MOSO: Decomposing MOtion, Scene and Object for Video Prediction

Mingzhen Sun [1,2]   Weining Wang [1]   Xinxin Zhu [1]   Jing Liu [1,2,*]

[1]The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences (CASIA)
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)

sunmingzhen2020@ia.ac.cn {weining.wang, xinxin.zhu, jliu}@nlpr.ia.ac.cn

Abstract

Motion, scene and object are three primary visual components of a video. In particular, objects represent the foreground, scenes represent the background, and motion traces their dynamics. Based on this insight, we propose a two-stage MOtion, Scene and Object decomposition framework (MOSO)[1] for video prediction, consisting of MOSO-VQVAE and MOSO-Transformer. In the first stage, MOSO-VQVAE decomposes a previous video clip into the motion, scene and object components, and represents them as distinct groups of discrete tokens. Then, in the second stage, MOSO-Transformer predicts the object and scene tokens of the subsequent video clip based on the previous tokens and adds dynamic motion at the token level to the generated object and scene tokens. Our framework can be easily extended to unconditional video generation and video frame interpolation tasks. Experimental results demonstrate that our method achieves new state-of-the-art performance on five challenging benchmarks for video prediction and unconditional video generation: BAIR, RoboNet, KTH, KITTI and UCF101. In addition, MOSO can produce realistic videos by combining objects and scenes from different videos.

1. Introduction

Video prediction aims to generate future video frames based on a past video without any additional annotations [6, 18], which is important for video perception systems, such as autonomous driving [25], robotic navigation [16] and decision making in daily life [5], etc. Considering that video is a spatio-temporal record of moving objects, an ideal solution of video prediction should depict visual content in the spatial domain accurately and predict motions in the temporal domain reasonably. However, easily distorted object identities and infinite possibilities of motion trajecto-

---

* Corresponding Author
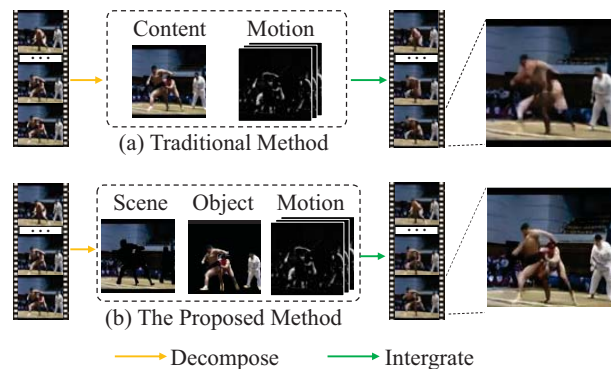[1]Codes have been released in *https://github.com/iva-mzsun/MOSO*

Figure 1. Rebuilding video signals based on (a) traditional decomposed content and motion signals or (b) our decomposed scene, object and motion signals. Decomposing content and motion signals causes blurred and distorted appearance of the wrestling man, while further separating objects from scenes resolves this issue.

ries make video prediction a challenging task.

Recently, several works [15, 44] propose to decompose video signals into content and motion, with content encoding the static parts, i.e., scene and object identities, and motion encoding the dynamic parts, i.e., visual changes. This decomposition allows two specific encoders to be developed, one for storing static content signals and the other for simulating dynamic motion signals. However, these methods do not distinguish between foreground objects and background scenes, which usually have distinct motion patterns. Motions of scenes can be caused by camera movements or environment changes, e.g., a breeze, whereas motions of objects such as jogging are always more local and routine. When scenes and objects are treated as a unity, their motion patterns cannot be handled in a distinct manner, resulting in blurry and distorted visual appearances. As depicted in Fig. 1, it is obvious that the moving subject (i.e., the wrestling man) is more clear in the video obtained by separating objects from scenes than that by treating them as a single entity traditionally.

Based on the above insight, we propose a two-stage MOtion, Scene and Object decomposition framework (MOSO) for video prediction. We distinguish objects from scenes and utilize motion signals to guide their integration. In the first stage, MOSO-VQVAE is developed to learn motion, scene and object decomposition encoding and video decoding in a self-supervised manner. Each decomposed component is equipped with an independent encoder to learn its features and to produce a distinct group of discrete tokens. To deal with different motion patterns, we integrate the object and scene features under the guidance of the corresponding motion feature. Then the video details can be decoded and rebuilt from the merged features. In particular, the decoding process is devised to be time-independent, so that a decomposed component or a single video frame can be decoded for flexible visualization.

In the second stage, MOSO-Transformer is proposed to generate a subsequent video clip based on a previous video clip. Motivated by the production of animation, which first determines character identities and then portrays a series of actions, MOSO-Transformer firstly predicts the object and scene tokens of the subsequent video clip from those of the previous video clip. Then the motion tokens of the subsequent video clip are generated based on the predicted scene and object tokens and the motion tokens of the previous video clip. The predicted object, scene, and motion tokens can be decoded to the subsequent video clip using MOSO-VQVAE. By modeling video prediction at the token level, MOSO-Transformer is relieved from the burden of modeling millions of pixels and can instead focus on capturing global context relationships. In addition, our framework can be easily extended to other video generation tasks, including unconditional video generation and video frame interpolation tasks, by simply revising the training or generation pipelines of MOSO-Transformer.

Our contributions are summarized as follows:

• We propose a novel two-stage framework MOSO for video prediction, which could decompose videos into motion, scene and object components and conduct video prediction at the token level.

• MOSO-VQVAE is proposed to learn motion, scene and object decomposition encoding and time-independently video decoding in a self-supervised manner, which allows video manipulation and flexible video decoding.

• MOSO-Transformer is proposed to first determine the scene and object identities of subsequent video clips and then predict subsequent motions at the token level.

• Qualitative and quantitative experiments on five challenging benchmarks of video prediction and unconditional video generation demonstrate that our proposed method achieves new state-of-the-art performance.

2. Related Work

Video Prediction The video prediction task has received increasing interest in the computer vision field. ConvLSTM [37] combines CNN and LSTM architectures and adopts an adversarial loss. MCnet [44] models pixel-level future video prediction with motion and content decomposition for the first time. GVSD [46] proposes a spatio-temporal CNN combined with adversarial training to untangle foreground objects from background scenes, while severe distortion of object appearances exists in their predicted video frames. MCVD [45] adopts a denoising diffusion model to conduct several video-related tasks conditioned on past and/or future frames. Although previous models can predict consistent subsequent videos, they still suffer from indistinct or distorted visual appearances since they lack a stable generator or fail to decouple different motion patterns. SLAMP [1] and vid2vid [47] decomposes video appearance and motion for video prediction with the help of optical flow. SADM [4] proposes a semantic-aware dynamic model that predicts and fuses the semantic maps (content) and optical flow maps (motion) of future video frames. In addition to optical flow and semantic maps, Wu et al. [54] further utilizes instance maps to help separate objects from backgrounds. Although these works also decompose video components, they are more complicated than MOSO since they require much more additional information. Furthermore, these previous works are primarily based on generative adversarial networks or recurrent neural networks, while MOSO follows a recently developed two-stage autoregressive generation framework, which demonstrates greater potential on open domain visual generation tasks.

Two-stage Visual Generation The two-stage framework is first proposed for image generation [11, 14, 33] and demonstrates excellent generation ability. Motivated by the success, several attempts have been made to extend the two-stage framework to video generation tasks [14, 52, 55, 55]. For video prediction, MaskViT [20] encodes videos by frame though VQ-GAN [14] and models video tokens with a bidirectional Transformer through window attention. For unconditional video generation, VideoGPT [55] encodes videos by employing 3D convolutions and axial attention, and then models video tokens in an auto-regressive manner. However, existing two-stage works for video tasks do not consider video component decomposition and are affected by flicker artifacts and expensive computation costs.

3. MOSO

In this section, we present our proposed framework MOSO in detail. MOSO is a novel two-stage framework for video prediction and consists of MOSO-VQVAE and MOSO-Transformer, where MOSO-VQVAE encodes decomposed video components to tokens and MOSO-

Transformer conducts video prediction at the token level. The overall framework of MOSO is drawn in Fig. 2.

We denote a T-frame video as x_1^T, which is an abbreviation for $\{x_t\}_{t=1}^T$, where $x_t \in R^{H \times W \times C}$ is the t-th video frame, H is the height, W is the weight and C is the number of channels. Video prediction requires predicting a subsequent video clip x_{K+1}^T based on a previous one x_1^K.

3.1. Stage I: MOSO-VQVAE

MOSO-VQVAE consists of a codebook $\mathbb{E}$, three video encoders (i.e., motion, scene and object encoders) and a time-independent video decoder. The codebook is denoted as $\mathbb{E} = \{e_i\}_{i=1}^N$, where $e_i \in R^D$ is the i-th codebook entry, N is the codebook size and D is the dimension. MOSO-VQVAE first roughly decomposes the motion, scene and object components of the input video x_1^T with a preprocessing algorithm, obtaining the motion video m_1^T, the scene video s_1^T and the object video o_1^T. Then the video encoder of each component takes its corresponding video as input and learns its feature. Each feature is then quantized using the codebook and a distinct group of tokens is available as a by-product. Finally, the video decoder integrates the quantized features to reconstruct the input video.

3.1.1 Decomposing

Considering that most of the scene motion is caused by camera movements or environment changes, scene motion is typically either tiny or extremely dramatic. Based on this observation, we propose an efficient preprocessing algorithm to separate one video into three distinct videos (i.e., motion, scene and object videos) without additional annotations. Specifically, frame difference is calculated and used as the motion video m_1^T. Then a lower threshold c_{lb} and an upper threshold c_{ub} are adopted to filter pixels with middle differences to obtain the object video o_1^T. The scene video s_1^T is composed of the left pixels with differences less than c_{lb} or higher than c_{ub}, corresponding to tiny or dramatic movements. More details are presented in the appendix.

3.1.2 Encoding

The scene encoder consists of several downsample layers, a temporal compressing layer and a residual stack. Given a scene video, the downsample layers, i.e., 2D convolutions with stride 2, downsample each frame by a factor of f. Then the frame features are concatenated in the channel dimension and compose feature $z_s' \in R^{H/f \times W/f \times TD}$, where TD is the number of channels. The temporal compressing layer, i.e., linear projection $R^{TD \to D}$, reduces the number of channels to D. The residual stack, composed of several residual layers [21], then learns the scene feature z_s. The object encoder has the same structure as the scene encoder but takes an object video as input and outputs an object feature z_o.

The motion encoder replaces the temporal compressing layer in the scene encoder with a temporal self-attention layer and outputs motion feature z_m. In particular, the downsampled frame features are concatenated in the temporal dimension and split into N_t parts, composing feature $z_m' \in R^{H/f \times W/f \times N_t \times (T/N_t) \times D}$. Then the temporal self-attention is conducted on each $Z \in \{z_m'^{h,w,n}\}$:

$$Q = ZW_Q, K = ZW_K, V = ZW_V$$
$$Y = softmax(QK^\top/\sqrt{D})V \qquad (1)$$

where $h \in \{1, ..., \frac{H}{f}\}, w \in \{1, ..., \frac{W}{f}\}, n \in \{1, ..., N_t\}$ and W_Q, W_K and W_V are learnable $D \times D$ projection matrices.

The codebook is used to quantize each feature $z \in \{z_s, z_o, z_m\}$ and obtain discrete tokens as:

$$\mathcal{VQ}(x_1^T)^{h,w,(t)} = e_r, \qquad \mathcal{Q}(x_1^T)^{h,w,(t)} = r$$
$$r = \arg\min_{1 \le i \le N} \|e_i - z^{h,w,(t)}\|_2^2 \qquad (2)$$

where $h \in \{1, ..., \frac{H}{f}\}$, $w \in \{1, ..., \frac{W}{f}\}$, $t \in \{1, ..., T\}$, $z_s, z_o \in R^{H/f \times W/f \times D}$, $z_m \in R^{H/f \times W/f \times T \times D}$, $\mathcal{VQ}$ obtains quantized features, $\mathcal{Q}$ obtains maps of tokens, e_i is the i-th codebook entry and $\|*\|^2$ denotes the calculation of L2 norm. The quantized features have the same shape as the raw features. Note that the temporal dimension of the scene and object features is 1, which is omitted for conciseness.

3.1.3 Decoding

The video decoder consists of a merge module, a residual stack and several upsample layers, i.e., transposed 2D convolutions with stride 2. To allow gradients to backpropagate to the encoders, the video decoder stops gradients of each quantized feature, obtaining $\tilde{z} = sg(\mathcal{VQ}(x_1^T) - z) + z$, where sg denotes the operator of stop-gradient.

The merge module dynamically integrates the object feature $\tilde{z}_o$ and the scene feature $\tilde{z}_s$ according to the t-th motion feature $\tilde{z}_m^t$ to obtain the video feature z^t for the t-th video frame. It first obtains three multi-scale weight features:

$$w_1^t = \mathcal{F}_1([\tilde{z}_o, \tilde{z}_s, \tilde{z}_m^t]) \qquad (3)$$
$$w_2^t = \mathcal{F}_2([\downarrow_2 (\tilde{z}_o), \downarrow_2 (\tilde{z}_s), \downarrow_2 (\tilde{z}_m^t)]) \qquad (4)$$
$$w_3^t = \mathcal{F}_3([\downarrow_4 (\tilde{z}_o), \downarrow_4 (\tilde{z}_s), \downarrow_4 (\tilde{z}_m^t)]) \qquad (5)$$

where $[*]$ concatenates input features along the channel dimension and $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ are linear functions $R^{3D \to D}$. $\downarrow_f$ denotes a downsample function implemented by 2D convolutions with stride 2, where f denotes the sample factor. The weight features are then merged by:

$$w_4^t = w_2^t + \uparrow_2 (w_3^t) \qquad (6)$$
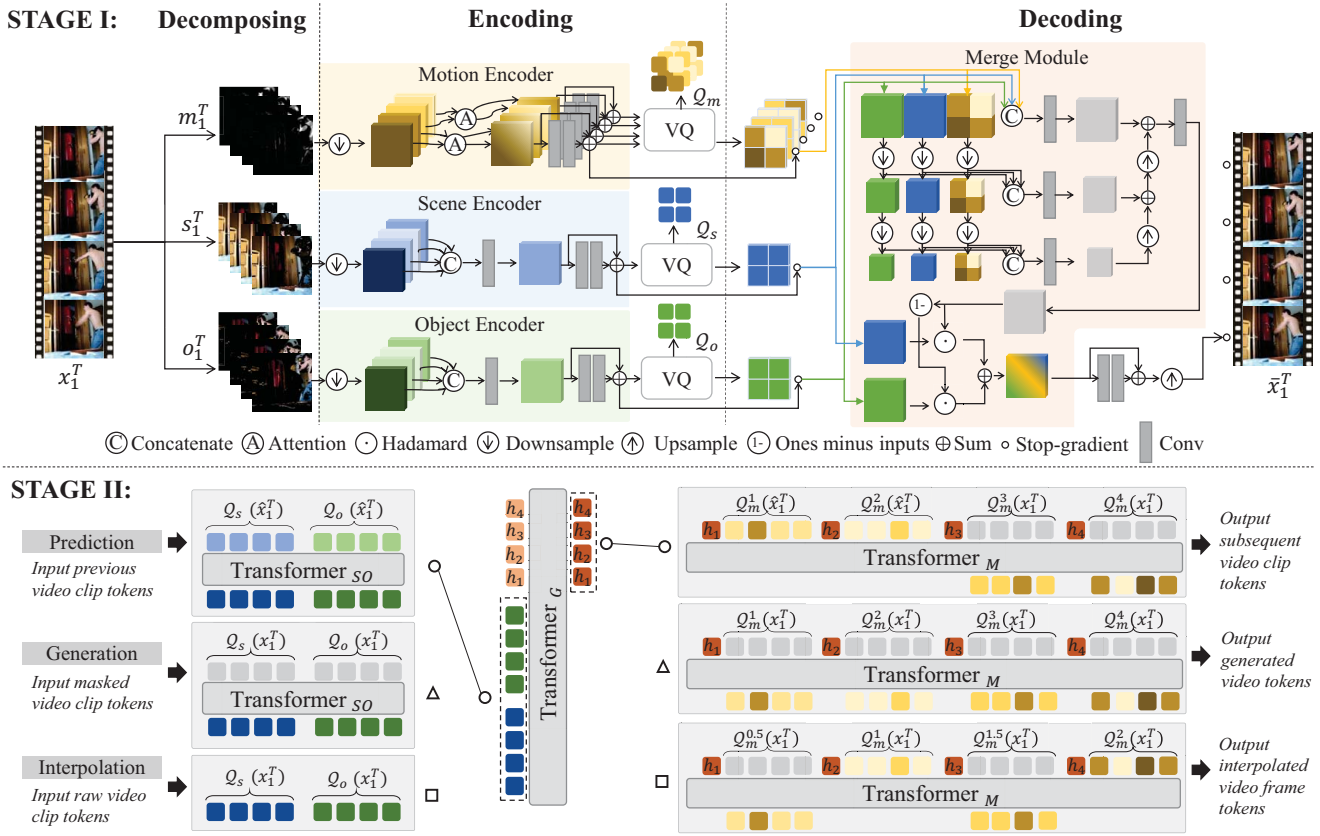$$w^t = \sigma(\mathcal{F}(w_1^t + \uparrow_2 (w_4^t))) \qquad (7)$$

Figure 2. The overall framework of MOSO. Above the dashed line shows the architecture of MOSO-VQVAE, which decomposes and encodes the motion, scene and object components of a video, produces token groups and reconstructs the input video from features. Below the dashed line shows the generation process of MOSO-Transformer, where the small gray grid represents the mask token. For video prediction, Transformer$_{SO}$ generates the scene and object tokens of x_1^T based on those of the pseudo video $\hat{x}_1^T$, which is composed of the given video frames. Then Transformer$_G$ outputs the guidance embeddings $\{h_1, ..., h_T\}$ and Transformer$_M$ predicts the subsequent motion tokens, where K is 2, T is 4 and the generation iteration S is 1 for illustration. MOSO-Transformer can perform unconditional video generation by feeding all mask tokens and video frame interpolation by generating only motion tokens of interpolated video frames.

where $\sigma(*)$ denotes the sigmoid activation function and $\mathcal{F} : R^D \to R^D$ is a linear function. $\uparrow_f$ denotes an upsampling function implemented by bilinear interpolation, where f denotes the sample factor. The t-th video frame feature is then obtained by:

$$z^t = \tilde{z}_s \odot w^t + \tilde{z}_o \odot (1 - w^t) \tag{8}$$

where $\odot$ denotes Hadamard product.

The residual stack has the same structure as that in the encoders. It operates on video frame features and is blind to their temporal index. Finally, several upsample layers spatially upsample video features to reconstruct video frames. The reconstructed t-th video frame is denoted as $\bar{x}_t$.

Notably, our video decoder is time-independent since the merge module, the residual stack and the upsample layers are agnostic to the temporal position of each video frame

and have no access to other motion features $\tilde{z}_m^k, k \neq t$ when reconstructing the t-th video frame. In other words, we only need to obtain the scene, object and t-th motion features when decoding the t-th video frame. It not only allows flexible visualization but also facilitates the modeling process in the second stage. In particular, the decomposed scenes and objects can be decoded and visualized by the video decoder based on their tokens, as discussed in the appendix.

3.1.4 Training

The overall loss function $\mathcal{L}$ for training MOSO-VQVAE consists of the reconstruction loss $\mathcal{L}_{rec}$ and the commitment loss $\mathcal{L}_{com}$. The reconstruction loss $\mathcal{L}_{rec}$ optimizes the video encoders and the video decoder by minimizing the L2 distance and the perceptual loss [57] between each pair
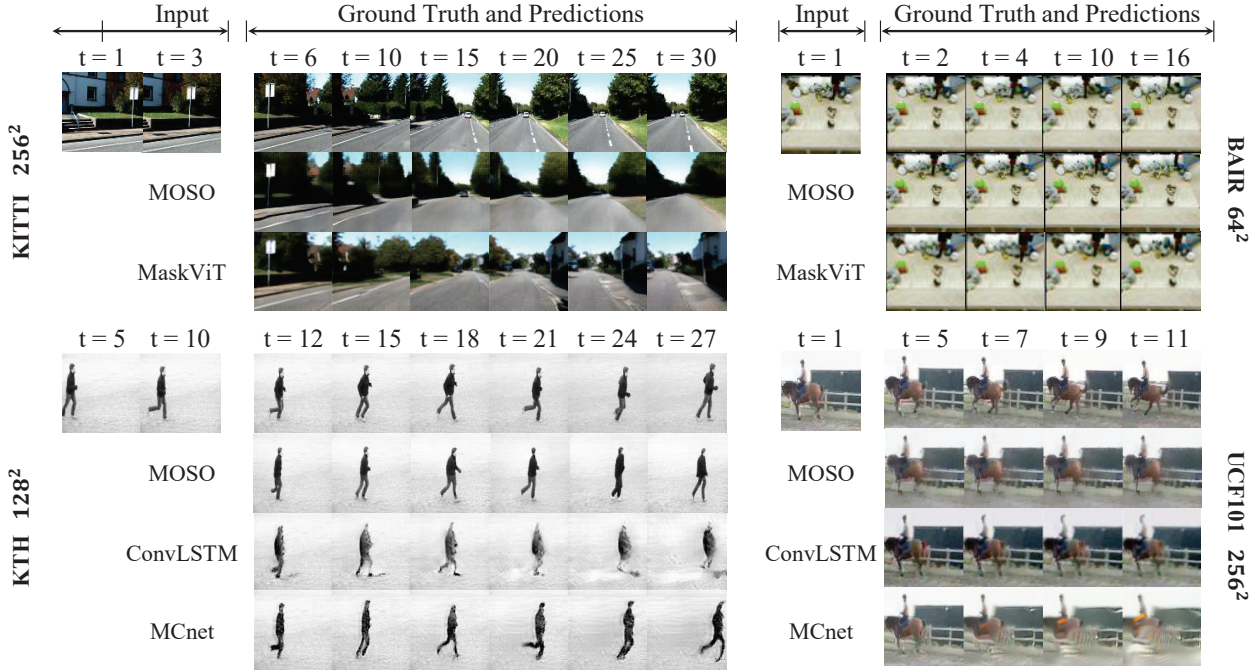
Figure 3. Qualitative comparison of MOSO with MaskViT [20], ConvLSTM [37] and MCnet [44] on test set of KITTI 256^2, BAIR 64^2, KTH 128^2 and UCF101 256^2 for video prediction.

of input and reconstructed video frames. The commitment loss $\mathcal{L}_{com}$ optimizes the encoders to ensure the training of the encoders keeps in step with that of the codebook, thus preventing divergence:

$$\mathcal{L}_{com} = \sum_{h=1}^{H/f} \sum_{w=1}^{W/f} \left[\sum_{t=1}^{T} \left\| z^{h,w,(t)} - sg\left(\mathcal{VQ}(x_1^T)^{h,w,(t)}\right) \right\|^2 \right] \tag{9}$$

where sg denotes the operator of stop-gradient and $\|*\|^2$ denotes the calculation of L2 norm. The exponential moving average (EMA) optimization method is used to optimize the codebook entries as in [35].

3.2. Stage II: MOSO-Transformer

MOSO-Transformer is comprised of three bidirectional Transformers: Transformer$_{SO}$, Transformer$_G$ and Transformer$_M$. Given a previous video clip x_1^K, MOSO-Transformer predicts the subsequent video clip x_{K+1}^T at the token level though pretrained MOSO-VQVAE. Since MOSO-VQVAE takes fixed length T-frame videos as input, we first pad the previous video clip x_1^K to T frames by duplicating the last frame, obtaining a pseudo video $\hat{x}_1^T$:

$$\hat{x}_1^T = \{x_1, ..., x_{K-1}, x_K, \underbrace{x_K, ..., x_K}_{T-K}\}, \quad K \leq T \tag{10}$$

Then the motion, scene and object tokens of $\hat{x}_1^T$ can be produced by MOSO-VQVAE. Based on the scene and object tokens of $\hat{x}_1^T$, Transformer$_{SO}$ is trained to generate the scene and object tokens of x_1^T:

$$P(\mathcal{Q}_s(x_1^T), \mathcal{Q}_o(x_1^T); \mathcal{Q}_s(\hat{x}_1^T), \mathcal{Q}_o(\hat{x}_1^T)) \tag{11}$$

To provide guidance for the generation of motion tokens, Transformer$_G$ is used to learn the guidance embeddings $\{h_1, ..., h_T\}$ based on the scene and object tokens of x_1^T. Considering that both Transformer$_{SO}$ and Transformer$_G$ require learning the scene and object content at the token level, we train them with shared parameters.

Rethinking the temporal split operation demonstrated in Eq. (1), when T can be exactly divided by K and the temporal dimension is $N_t = \frac{T}{K}$ parts, the motion tokens of the target video x_1^T and the pseudo video $\hat{x}_1^T$ satisfy:

$$\mathcal{Q}(x_1^T)^t == \mathcal{Q}(\hat{x}_1^T)^t, \quad 1 \leq t \leq K \tag{12}$$

Namely, the motion tokens for the first K video frames in the target and pseudo videos are totally the same. A detailed explanation of this property and a more general situation are presented in the appendix.

Based on the above property, Transformer$_M$ only needs to generate motion tokens of the last $T - K$ frames of x_1^T based on the guidance embeddings $\{h_1, ..., h_T\}$ and the motion tokens of the first K given video frames of $\hat{x}_1^T$. Following the training pipeline in [7], we first randomly mask $\gamma(r)$ proportion of motion tokens with mask tokens $[M]$, where r is a scalar and $\gamma(r)$ is a monotonically declining function with $\gamma(0) = 1$ and $\gamma(1) = 0$. Then Transformer$_M$ is

trained through a cross-entropy loss to model the distribution of masked motion tokens based on unmasked motion tokens, the guidance embeddings and the motion tokens of the first K video frames of $\hat{x}_1^T$:

$$P(\{\mathcal{Q}_m(x_1^T)_{=[M]}^n\}; \{\mathcal{Q}_m(x_1^T)_{\neq[M]}^n\}, \{\mathcal{Q}_m(\hat{x}_1^T)^r\}, \{h_t\})$$
$$K+1 \leq n \leq T, \quad 1 \leq r \leq K, \quad 1 \leq t \leq T$$
(13)

3.2.1 Predicting

Inspired by the production of animation, which first creates figure characters and then depicts a series of actions, we predict the subsequent video clip by first determining its scene and object identities and then producing dynamic motions at the token level. In particular, Transformer$_{SO}$ firstly outputs the distribution of the scene and object tokens of x_1^T as in Eq. (11). Then the scene and object tokens are randomly sampled and the guidance embeddings can be obtained through the Transformer$_G$. Given the guidance embeddings and a template filled with mask tokens $[M]$, Transformer$_M$ performs the following two steps for S iterations to predict the motion tokens: (1) outputting distribution of motion tokens as specified in Eq. (13) and sampling motion tokens in the masked positions; (2) randomly masking $\gamma(s/S)$ proportion of all motion tokens with previous unmasked tokens fixed, where s denotes the s-th iteration. Finally, the predicted video frames x_{K+1}^T are decoded from the generated tokens by MOSO-VQVAE. In addition, our MOSO-Transformer can be easily extended to other tasks involving unconditional video generation and video frame interpolation as shown in Fig. 2.

4. Experiments

In this section, we compare our method with prior works for video prediction and unconditional video generation tasks on five challenging benchmarks, including BAIR [13], RoboNet [9], KTH [36], KITTI [19] and UCF101 [39]. Fig. 3 shows the qualitative comparison of video prediction. We start with introducing evaluation metrics and experimental implementations.

Metrics We adopt five evaluation metrics: Fréchet Video Distance (FVD), Fréchet Instance Distance (FID), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). FVD [42] measures the distance between distributions of predicted videos and real-world videos. We calculate FVD with codes released by StyleGAN-V[2] [38], which has been proven to be a precise implementation of the official one. To obtain the FVD score, we pass those given previous video frames and calculate between the predicted frames and corresponding ground truths following MaskViT [20]. For unconditional video generation,

we follow StyleGAN-V [38] and calculate FVD on 2048 randomly sampled videos. FID [22] evaluates the generated videos by frame. PSNR [24], SSIM [48] and LPIPS [57] measure frame-wise similarity between predicted and ground truth videos. We conduct one trial per video on the BAIR datasets. On the KTH, KITTI and RoboNet datasets, we follow [3, 20] and report the best SSIM, PSNR and LPIPS scores over 100 trials per video to account for the stochastic nature of video prediction. FVD is calculated over all 100 trials with batch size being 256 following [20].

Implementation MOSO-VQVAE encodes videos by clip. When training MOSO-VQVAE, each video clip has a fixed length T, which is set as 16 for BAIR, 12 for RoboNet and 20 for KTH and KITTI. N_t is set as 1 for BAIR, 6 for RoboNet, 2 for KTH and 4 for KITTI. For UCF101, T and N_t are set as 12 and 3 for video prediction or 16 and 1 for unconditional video generation. The batch size is 32 for 64^2 resolution and 4 for others. The codebook size N is 16384. The preprocessing algorithm is used for the first 50k iterations. After that, the scene and object encoders take raw video frames as input and learn scene and object identities in a self-supervised manner, and the motion encoder still takes the frame difference as input. For MOSO-Transformer, the batch size is 16 for 64^2 resolution and 4 for others. $gamma(*)$ is set to decay as cosine and S is set as 16. For long video prediction, tokens of the last video clip are used as previous tokens for iterative prediction. Deepspeed [34] and mixed-precision FP16/FP32 [26] are utilized for fast training. Experiments are conducted on 4 A100s.

4.1. Video Prediction

BAIR. The BAIR robot pushing dataset [13] records random movements of robotic arms. Following prior work [53], we predict subsequent 15 video frames given only 1 previous video frame, and all videos are resized to 64×64 resolution. Qualitative comparison of MOSO with MaskViT is given in Fig. 3. Our MOSO achieves new state-of-the-art performance on this dataset as reported in Table 1, outperforming prior best work by 3.3 FVD.

RoboNet. The RoboNet dataset [9] contains more than 15 million videos of 7 robotic arms pushing things in different bins. Following prior works [3, 51], we randomly select 256 videos for testing and predict 10 subsequent video frames given 2 previous video frames. As shown in Table 2, MOSO achieves significant improvements on all metrics than prior works at both 64^2 and 256^2 resolution.

Both MOSO and MaskViT follow the pipeline of VQ-VAE and Transformer, while MOSO outperforms MaskViT by a large margin on the RoboNet dataset. We attribute this phenomenon to two reasons. Firstly, MaskViT models videos by frame, which can lead to flicker artifacts in videos with static backgrounds like RoboNet [20], while MOSO models videos by clip and thus obtains better con-

---

[2]https://github.com/universome/stylegan-v

Table 1. Quantitative comparison with other methods on BAIR for video prediction.

Method	Params	FVD ↓
LVT [32]	-	125.8
SAVP [27]	-	116.4
DVD-GAN-FP [8]	-	109.8
VT (S) [49]	46M	106.0
TrIVD-GAN-FP [28]	-	103.3
VideoGPT [55]	-	103.3
CCVS [30]	-	99.0
VT (L) [49]	373M	94.0
MaskViT [20]	189M	93.7
FitVid [3]	302M	93.6
MCVD [45]	251M	89.5
RaMViD [23]	-	89.2
NUWA [53]	870M	86.9
MOSO	265M	**83.6**

Table 2. Comparison with other methods on RoboNet for video prediction.

Method	Params	$H \times W$	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MaskViT [20]	257M	64^2	133.5	23.2	80.5	0.042
SVG [43]	298M	64^2	123.2	23.9	87.8	0.060
GHVAE [50]	599M	64^2	95.2	24.7	89.1	0.036
FitVid [3]	302M	64^2	62.5	28.2	89.3	0.024
MOSO	265M	64^2	**53.3**	**31.2**	**92.1**	**0.017**
MaskViT [20]	228M	256^2	211.7	20.4	67.1	0.170
MOSO	265M	256^2	**91.5**	**26.3**	**79.9**	**0.096**

Table 3. Comparison with other methods on KITTI for video prediction.

Method	Params	$H \times W$	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
SVG [43]	298M	64^2	1217.3	15.0	41.9	0.327
FitVid [3]	302M	64^2	884.5	17.1	49.1	0.217
GHVAE [50]	599M	64^2	552.9	15.8	51.2	0.286
MaskViT [20]	181M	64^2	401.9	**27.2**	58.1	0.089
MOSO	177M	64^2	**395.3**	25.6	**74.8**	**0.086**
MaskViT [20]	228M	256^2	**446.1**	**26.2**	40.7	0.270
MOSO	219M	256^2	516.6	21.1	**59.2**	**0.265**

Table 4. Comparison with prior works on KTH for video prediction.

Method	$H \times W$	PSNR ↑	SSIM ↑	LPIPS ↓
SVG-LP [10]	64^2	23.9	80.0	0.129
Struct-VRNN [29]	64^2	24.3	76.6	0.124
SV2P [2]	64^2	25.9	78.2	0.232
SAVP [27]	64^2	26.0	80.6	0.116
MCVD [45]	64^2	26.4	81.2	-
GK [17]	64^2	27.1	**83.7**	0.092
MOSO	64^2	**29.8**	82.2	**0.083**

tent consistency in videos. Secondly, videos in RoboNet are easy to distinguish between foreground objects and background scenes, which facilitates object and scene decomposition and hence favors MOSO. Quantitative results of MOSO-VQVAE video reconstruction are given in the appendix, which demonstrate that MOSO-VQVAE can handle videos with static and tiny-motion backgrounds quite well.

KTH. The KTH dataset [36] contains videos of 25 people performing 6 types of actions. Following [17, 44], we adopt videos of persons 1-16 for training and 17-25 for testing. When predicting videos, 10 subsequent video frames are generated based on 10 previous video frames during training, and 40 subsequent video frames are required to be predicted during testing. Following [44], we manually trim the conditional video clips to ensure humans are always present. We report the quantitative results in Table 4 and the qualitative results in Fig. 3. More results are available in the appendix. We do not calculate FVD since its batch size setting is ambiguously stated in [17], which may dramatically impact the evaluation result. As reported in Table 4, MOSO outperforms GK [17] by 0.27 on PSNR and 0.09 on LPIPS, and obtains a comparable SSIM score. As depicted in Fig. 3, our MOSO outperforms the previous motion and content decomposition method MCnet [44] with more distinct object appearances and less artifacts.

KITTI. The KITTI dataset [19] is a challenging dataset with only 57 training videos and dramatic moving scenes. MaskViT [20] outperforms prior works by a large margin on this dataset, and we achieve comparable performance with MaskViT at both 64^2 and 128^2 resolutions as reported in Table 3. Different from the VQ-GAN adopted in MaskViT, which sets the weight of the adversarial loss as 1, MOSO-VQVAE adopts a smaller loss weight 0.1 as specified in the ablation study. Despite such loss weight helps MOSO-VQVAE achieve outstanding performance on other datasets, it seems to be too small for KITTI to compete with the perceptual loss with weight 1, leading to some checkboard artifacts in the predicted video frames as shown in Fig. 3. These checkboard artifacts are produced by the ResNet-50 network used in the perceptual loss as indicated in [12, 31].

4.2. Other Video Generation Tasks

Unconditional Video Generation. MOSO-Transformer can be trained for unconditional video generation by replacing the scene and object tokens of the given video clip with mask tokens and removing given motion tokens. We quantitatively compare MOSO with other models on the UCF101 dataset for unconditional video generation as reported in Table 5. The results demonstrate that our MOSO outperforms the previous method [38] by 219.1 on FVD.

Video Frame Interpolation. MOSO can directly perform video frame interpolation after being trained for video prediction. Based on raw motion, scene and object tokens, MOSO-Transformer generates interpolated video frames by initializing their motion tokens with mask tokens and then removing mask tokens through S steps, where S is 16. Samples on the RoboNet, KTH and KITTI datasets are given in Fig. 4, which shows that MOSO could interpolate consistent video frames.

Video Manipulation. By separating object identities

Table 5. Quantitative comparison with other methods on UCF101 for unconditional video generation.

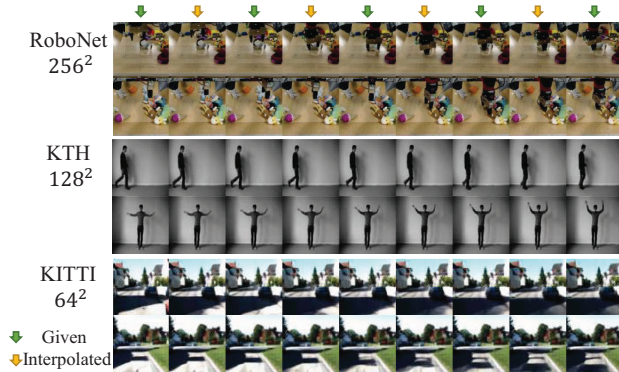| Method	FVD ↓
MoCoGAN [41]	2886.9
+StyleGAN2 backbone	1821.4
MoCoGAN-HD [40]	1729.6
VideoGPT [55]	2880.6
DIGAN [56]	1630.2
StyleGAN-V [38]	1431.0
MOSO	**1202.6**

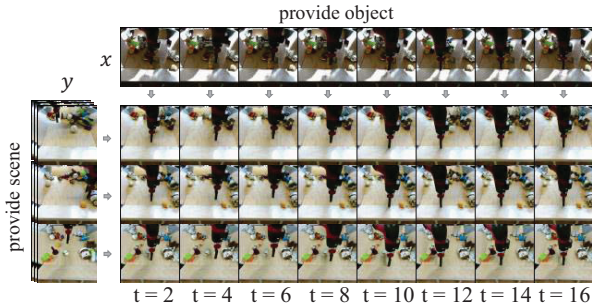Figure 4. Samples of video frame interpolation on RoboNet, KTH and KITTI.

Figure 5. Samples of video manipulation on BAIR. Videos with objects from x and scenes from y are created by decoding object and motion tokens of x combined with scene tokens of y.

from scenes, MOSO-VQVAE can decode realistic videos with objects from one video and scenes from other videos. In particular, given two series of video frames x_1^T and y_1^T, motion, scene and object tokens are obtained through the video encoders. By combining the object and motion tokens of x_1^T and the scene tokens of y_1^T, a new video with objects from x_1^T and scene from y_1^T can be decoded by the video decoder as shown in Fig. 5.

4.3. Ablation Study

We conduct ablation studies on the KTH 64^2 dataset for video prediction and report the results in Table 6. The number of tokens for the non-decomposition or content and motion decomposition encoders are shown in Table 7. For a

Table 6. Ablation studies on KTH 64^2 for video prediction. *non decom.*: non-decomposition. *mo. co.*: motion and content decomposition. *mo. sc. ob.*: motion, scene and object decomposition. *pre. alg.* denotes the preprocessing algorithm. *Trans.* denotes Transformer.

Method	Parameters (VAE+Trans.)	PSNR↑	SSIM↑	LPIPS↓
non decom. (f_{non} : 8)	605M+308M	21.5	62.8	0.202
mo. co. (f_{mo}:8,f_{co}:4)	611M+265M	25.4	68.8	0.147
mo. co. (f_{mo}:8,f_{co}:2)	535M+265M	27.1	75.5	0.129
mo. sc. ob. (MOSO)	593M+265M	**29.8**	**82.2**	**0.083**
MOSO - pre. alg.	593M+265M	29.2	82.0	0.086
MOSO - merge Module	557M+265M	28.3	79.3	0.125
MOSO + single Trans.	593M+286M	28.9	80.1	0.116

Table 7. Number of tokens changes with downsample factors.

Method	f_{non}		(f_m,f_c)		(f_m,f_s,f_o)
Down. factor	8	4	(8,4)	(8,2)	(8,4,4)
Token length	1280	5120	1536	2304	1792
FLOPs($\times 10^9$)	389	2382	481	796	581

fair comparison, we ensure the parameters of both VQVAE and the Transformer of all involved models are comparable. As reported in Table 6, the performance of MOSO outperforms the non-decomposition method and the motion and content decomposition method by a large margin.

5. Conclusion and Discussions

In this paper, we propose a novel two-stage motion, scene and object decomposition framework for video prediction. Extensive experiments show that our method achieves new state-of-the-art performance on several challenging benchmarks, demonstrating the importance of decomposing the motion, scene and object video components.

Our research has two main limitations. First, although the preprocessing algorithm is efficient and effective, it is not delicate enough. More expressive and powerful tools, e.g. optical flow, may help decompose the motion, scene and object components better. Second, we do not enlarge the model and dataset for video prediction, while several works [33, 53] have revealed the potential of a huge Transformer for the open-domain visual generation. Future works are highly-encouraged for further exploration.

6. Acknowledgments

References

[1] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Güney. Slamp: Stochastic latent appearance and motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14728–14737, 2021. 2

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proceedings of the International Conference on Learning Representations*, 2018. 7

[3] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *CoRR*, abs/2106.13195, 2021. 6, 7

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

[5] Andreja Bubic, D Yves Von Cramon, and Ricarda I Schubotz. Prediction, cognition and the brain. *Frontiers in human neuroscience*, 4:25, 2010. 1

[6] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision*, pages 753–769, 2018. 1

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 5

[8] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 7

[9] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, volume 100, pages 885–897, 2019. 6

[10] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the International Conference on Machine Learning*, volume 80, pages 1182–1191, 2018. 7

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems*, pages 19822–19835, 2021. 2

[12] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for BERT pre-training of vision transformers. *CoRR*, abs/2111.12710, 2021. 7

[13] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *Conference on Robot Learning*, volume 78, pages 344–356, 2017. 6

[14] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 2

[15] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016. 1

[16] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *International Conference on Robotics and Automation*, pages 2786–2793, 2017. 1

[17] Xiaojie Gao, Yueming Jin, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. Accurate grid keypoint learning for efficient video prediction. In *International Conference on Intelligent Robots and Systems*, pages 5908–5915, 2021. 7

[18] Xiaojie Gao, Yueming Jin, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. Accurate grid keypoint learning for efficient video prediction. In *International Conference on Intelligent Robots and Systems*, pages 5908–5915, 2021. 1

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6, 7

[20] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *CoRR*, abs/2206.11894, 2022. 2, 5, 6, 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[23] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *CoRR*, abs/2206.07696, 2022. 7

[24] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6

[25] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *International Conference on Intelligent Robots and Systems*, pages 5801–5806, 2018. 1

[26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020. 6

[27] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018. 7

[28] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *CoRR*, abs/2003.04035, 2020. 7

[29] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P. Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *Advances in Neural Information Processing Systems*, pages 92–102, 2019. 7

[30] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. CCVS: context-aware controllable video synthesis. In *Advances in Neural Information Processing Systems*, pages 14042–14055, 2021. 7

[31] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 7

[32] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 101–112, 2021. 7

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning*, volume 139, pages 8821–8831, 2021. 2, 8

[34] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3505–3506, 2020. 6

[35] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019. 5

[36] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *International Conference on Pattern Recognition*, pages 32–36, 2004. 6, 7

[37] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015. 2, 5

[38] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6, 7, 8

[39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. 6

[40] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *Proceedings of the International Conference on Learning Representations*, 2021. 8

[41] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018. 8

[42] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717, 2018. 6

[43] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 81–91, 2019. 7

[44] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *Proceedings of the International Conference on Learning Representations*, 2017. 1, 2, 5, 7

[45] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: masked conditional video diffusion for prediction, generation, and interpolation. *CoRR*, abs/2205.09853, 2022. 2, 7

[46] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems*, pages 613–621, 2016. 2

[47] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 2

[48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[49] Dirk Weissenborn, Oscar Tackstrom, and Jakob Uszkoreit. Scaling autoregressive video models. In *Proceedings of the International Conference on Learning Representations*, 2020. 7

[50] Bohan Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2318–2328, 2021. 7

[51] Bohan Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2318–2328, 2021. 6

[52] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. GODIVA: generating open-domain videos from natural descriptions. *CoRR*, abs/2104.14806, 2021. 2

[53] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pretraining for neural visual world creation. In *European Conference on Computer Vision*, volume 13676, pages 720–736, 2022. 6, 7, 8

[54] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5539–5548, 2020. 2

[55] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using VQ-VAE and transformers. *CoRR*, abs/2104.10157, 2021. 2, 7, 8

[56] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2022. 8

[57] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 4, 6