Video Prediction Recalling Long-term Motion Context via Memory Alignment Learning

Sangmin Lee[1] Hak Gu Kim[2] Dae Hwi Choi[1] Hyung-Il Kim[1,3] Yong Man Ro[1*]
[1] Image and Video Systems Lab, KAIST, South Korea
[2] EPFL, Switzerland [3] ETRI, South Korea
{sangmin.lee,sinjh1796,ymro}@kaist.ac.kr hakgu.kim@epfl.ch hikim@etri.re.kr

Abstract

Our work addresses long-term motion context issues for predicting future frames. To predict the future precisely, it is required to capture which long-term motion context (e.g., walking or running) the input motion (e.g., leg movement) belongs to. The bottlenecks arising when dealing with the long-term motion context are: (i) how to predict the long-term motion context naturally matching input sequences with limited dynamics, (ii) how to predict the long-term motion context with high-dimensionality (e.g., complex motion). To address the issues, we propose novel motion context-aware video prediction. To solve the bottleneck (i), we introduce a long-term motion context memory (LMC-Memory) with memory alignment learning. The proposed memory alignment learning enables to store long-term motion contexts into the memory and to match them with sequences including limited dynamics. As a result, the long-term context can be recalled from the limited input sequence. In addition, to resolve the bottleneck (ii), we propose memory query decomposition to store local motion context (i.e., low-dimensional dynamics) and recall the suitable local context for each local part of the input individually. It enables to boost the alignment effects of the memory. Experimental results show that the proposed method outperforms other sophisticated RNN-based methods, especially in long-term condition. Further, we validate the effectiveness of the proposed network designs by conducting ablation studies and memory feature analysis. The source code of this work is available[†].

1. Introduction

Video prediction in computer vision is to estimate upcoming future frames at pixel-level from given previous frames. Since predicting the future is an important basement for intelligent decision-making systems, the video prediction has attracted increasing attention in industry and research fields. It has the potential to be applied to various tasks such as weather forecasting [40], traffic situation prediction [5], and autonomous driving [4]. However, the pixel-level video prediction is still challenging mainly due to the difficulties of capturing high-dimensionality and long-term motion dynamics [11, 33, 34, 36].

Recently, several studies with deep neural networks (DNNs) have been proposed to capture the high-dimensionality and the long-term dynamics of video data in the video prediction field [7, 11, 29, 33–36]. The models considering the high-dimensionality of videos tried to simplify the problem by constraining motion and disentangling components [7, 11, 33]. However, these methods did not consider the long-term frame dynamics, which leads to predicting blurry frames or wrong motion trajectories. Recurrent neural networks (RNNs) have been developed to capture the long-term dynamics with consideration for long-term dependencies in the video prediction [34–36]. The long-term dependencies in the RNNs is about remembering past step inputs. The RNN-based methods exploited the memory cell states in the RNN unit. The cell states are recurrently changed according to the current input sequence to remember the previous steps of the sequence. However, it is difficult to capture the long-term motion dynamics for the input sequence with limited dynamics (*i.e.*, short-term motion) because such cell states mainly depend on revealing relations within the current input sequence. For example, given short-length input frames for a walking motion, the leg movement from the input is limited itself. Therefore, it is difficult to grasp what will happen to the leg in the future through the cell states of the RNNs. In this case, the long-term motion context of the partial action may not be properly captured by the RNN-based methods.

Our work addresses *long-term motion context issues* for predicting future frames, which have not been properly dealt with in previous video prediction works. To predict the future precisely, it is required to capture which long-

*Corresponding author
[†]https://github.com/sangmin-git/LMC-Memory

term motion context the input motion belongs to. For example, in order to predict the future of leg movement, we need to know such partial leg movement belongs to either walking or running (*i.e.*, long-term motion context). The bottlenecks arising when dealing with long-term motion context are as follows: *(i) how to predict the long-term motion context naturally matching input sequences with limited dynamics, (ii) how to predict the long-term motion context with high-dimensionality*.

In this paper, we propose novel motion context-aware video prediction to address the aforementioned issues. To solve the bottleneck *(i)*, we introduce a long-term motion context memory (LMC-Memory) with memory alignment learning. Contrary to the internal memory cells of the RNNs, the LMC-Memory externally exists with its own parameters to preserve various long-term motion contexts of training data, which are not limited to the current input. Memory alignment learning is proposed to effectively store long-term motion contexts into the LMC-Memory and recall them even with inputs having limited dynamics. Memory alignment learning contains two training phases to align long-term and short-term motions: $\langle$*Phase 1*$\rangle$ storing long-term motion context from long-term sequences into the memory, $\langle$*Phase 2*$\rangle$ matching input short-term sequences with the stored long-term motion contexts in the memory. As a result, the long-term motion context (*e.g.*, long-term walking dynamics) can be recalled from the input short-term sequence alone (*e.g.*, short-term walking clip).

Furthermore, to resolve the bottleneck *(ii)*, we propose decomposition of a memory query that is used to store and recall the motion context. Even if various motion contexts of training data are stored in the LMC-Memory, it is difficult to capture the motion context that is exactly matched with the input. This is because motions of video sequences have high-dimensionality (*e.g.*, complex motion with local motion components). The dimensionality indicates the number of pixels in a video sequence. Since each motion is slightly different from one another in a global manner even for the same category, the proposed memory query decomposition is useful in that it enables to store local context (*i.e.*, low-dimensional dynamics) and recall the suitable local context for each local part of the input individually. It can boost the alignment effects between the input and the stored long-term motion context in the LMC-Memory.

The major contributions of the paper are as follows.

- We introduce novel motion context-aware video prediction to solve the inherent problem of the RNN-based methods in capturing long-term motion context. We address the arising long-term motion context issues in the video prediction.
- We propose the LMC-Memory with memory alignment learning to address storing and recalling long-term motion contexts. Through the learning, it is possible to recall long-term motion context corresponding to an input sequence even with limited dynamics.
- To address the high-dimensionality of motions, we decompose memory query to separate an overall motion into local motions with low-dimensional dynamics. It makes it possible to recall suitable local motion context for each local part of the input individually.

2. Related Work

2.1. Video Prediction

In video prediction, errors for predicting future frames can be divided into two factors [36]. The first one is about systematic errors due to the lack of modeling capacity for deterministic changes. The second one is related to modeling the intrinsic uncertainty of the future. There have been several works to address the second factor [1, 6, 19, 42]. These methods utilized stochastic modeling to generate plausible multiple futures. Contrary to these, our paper addresses the video prediction focusing on the first factor.

Recently, deep learning-based video prediction methods have been proposed to deal with the first factor. They considered the problems leading to prediction difficulty such as capturing high-dimensionality and long-term dynamics in video data [2,7,8,13,15,16,21,24,29,31–37,39,41,43,44]. Finn *et al.* [7] incorporated appearance information in the previous frames with the predicted pixel motion information for long-range video prediction. Villegas *et al.* [34] introduced a hierarchical prediction model that generates the future image from the predicted high-level structure. A predictive recurrent neural network (PredRNN) model was presented by Wang *et al.* [37] to model and memorize both spatial and temporal representations simultaneously. Wang *et al.* [35] further extended this model, named PredRNN++ to solve the vanishing gradient problem in deep-in-time prediction by building adaptive learning between long-term and short-term frame relation. Recently, eidetic 3D LSTM (E3D-LSTM) [36] was proposed to integrate 3D convolutions into the RNNs for effectively addressing memories across long-term periods. Jin *et al.* [13] introduced spatial-temporal multi-frequency analysis for high-fidelity video prediction with temporal-consistency . Su *et al.* [29] proposed convolutional tensor-train decomposition to learn long-term spatio-temporal correlations. However, these works still have a limitation in encoding long-term dynamics in that they mainly rely on the input sequence to find frame relations. Therefore, it is difficult to capture the long-term motion context for predicting the future from the input sequence with limited dynamics.

2.2. Memory Network

Memory augmented networks have recently been introduced for solving various problems in computer vision tasks

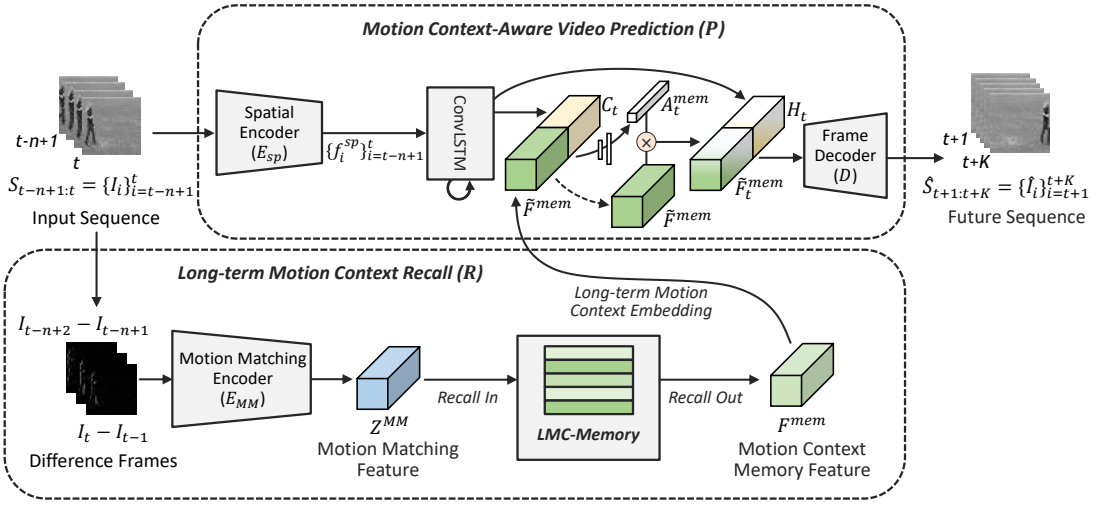

Figure 1: Overall framework with the proposed LMC-memory for video prediction at testing phase. The lower path is for recalling long-term motion context from the external memory, named LMC-Memory. The upper path is for predicting future frames with recurrent manner considering the recalled long-term motion context.

[3, 9, 10, 14, 18, 20, 23, 25, 46, 47]. Such computer vision tasks include anomaly detection [9, 23], few-shot learning [3, 14, 46], image generation [47], and video summarization [20]. Kaiser *et al.* [14] presented a large scale long-term memory module for life-long learning. Memory-attended recurrent network was proposed by Pei *et al.* [25] to capture the full-spectrum correspondence between the word and its visual contexts across video sequences in training data. To utilize the external memory network for our purposes, we introduce novel memory alignment learning that enables to store the long-term motion contexts into the memory and to recall them with limited input sequences. In addition, we separate overall motion into low-dimensional dynamics and utilize them as an individual memory query to recall proper long-term motion context for each local part of inputs.

3. Proposed Method

3.1. Motion Context-Aware Video Prediction

Video prediction task can be formulated as follows. Let $I_t \in \mathbb{R}^{W\times H\times C}$ denote the (t)-th frame in the video, and $S_{t-n+1:t} = \{I_i\}_{i=t-n+1}^{t}$ denote the video sequence containing $(t\text{-}n\text{+}1)$-th to (t)-th frames. The goal is to optimize the predictive function $\mathcal{F}$ for making generated next sequence $\hat{S}_{t+1:t+K}=\mathcal{F}(S_{t-n+1:t})$ be similar with actual next sequence $S_{t+1:t+K}$ for given previous sequence $S_{t-n+1:t}$. Figure 1 shows the overall framework of the proposed video frame prediction at inference phase. The input sequence goes through two paths to predict the future frames. One (lower path of Figure 1) is for recalling long-term motion context from the memory. The other (upper path of Figure 1) is for predicting frames recurrently with the recalled long-term motion context.

First, in the lower path of Figure 1, the differences between the consecutive frames (*i.e.*, difference frames) are used as inputs of motion matching encoder E_{MM}. Then, a motion matching feature Z^{MM} is extracted to recall the motion context memory feature F^{mem} from the external memory, named LMC-Memory. This LMC-Memory contains various long-term motion contexts of training data. Thus, F^{mem} from the memory can be considered as long-term information corresponding to the input sequence $S_{t-n+1:t}$ (described in Section 3.2 in detail). It is then embedded in the upper part P. This long-term motion context embedding contains 2D-DeConvs to match the spatial size with the upper part, which results in $\widetilde{F}^{mem}$.

The upper part of Figure 1 demonstrates long-term motion context-aware video prediction P scheme. In this path, the required motion context is refined through attention-based encoding to effectively embed it in predicting future frames. Each frame of the input sequence is independently fed to spatial encoder E_{sp} with 2D-Convs to extract appearance characteristics. The ConvLSTM [40] receives each extracted spatial feature $f_t^{sp} = E_{sp}(I_t)$ as inputs in time step order. A cell state C_t and an output state H_t are obtained from recurrent processing of the ConvLSTM. Since C_t contains the information from the past to the present of the input sequence, we use C_t to refine $\widetilde{F}^{mem}$ for embedding the required motion context at the current step. C_t and $\widetilde{F}^{mem}$ are concatenated and pass through fully connected layers to make channel-wise attention A_t^{mem} for $\widetilde{F}^{mem}$. The channel-wise refined feature $\widetilde{F}_t^{mem}=A_t^{mem}\otimes\widetilde{F}^{mem}$ and output state H_t from the ConvLSTM are concatenated to embed long-term context to the ConvLSTM output (*i.e.*, spatio-temporal information of the input). The concatenated feature is fed to a frame decoder

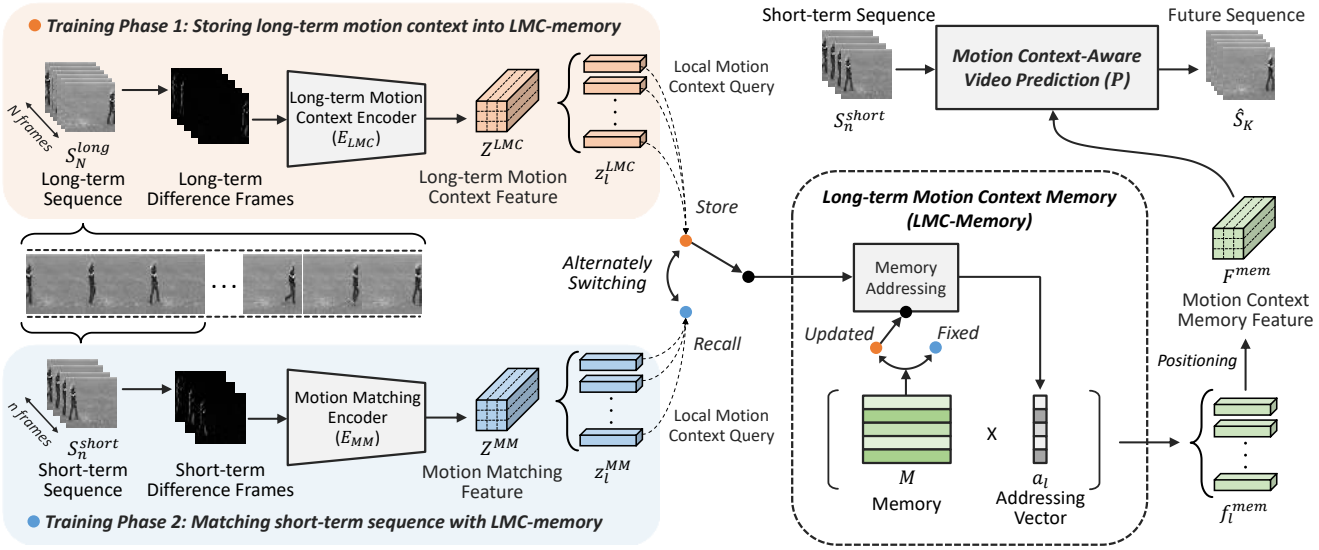

Figure 2: Training scheme of LMC-memory. To align the long-term and short-term in the memory, the networks are trained with two phases: (1) storing long-term motion context, (2) matching limited short-term sequence with the long-term context.

D with 2D-DeConvs to generate corresponding next frame $\hat{I}_{t+1} = D([H_t; \tilde{F}_t^{mem}])$. The embedded motion context memory feature can provide the prior of long-term motion context for the current input sequence. Note that the generated next frame $\hat{I}_{t+1}$ enters P as a new input to create the further future frame.

3.2. LMC-Memory with Alignment Learning

The long-term motion context memory, named LMC-Memory is to provide the long-term motion context for current input sequences to predict future frames. To effectively recall the long-term motion context even for the input sequence with limited dynamics, we propose novel memory alignment learning. Figure 2 shows the training scheme of the LMC-Memory. The memory is trained alternately with two phases: ⟨*Phase 1*⟩ storing long-term motion context into the memory and ⟨*Phase 2*⟩ matching a limited sequence with the corresponding long-term context in the memory.

During the storing phase ⟨*Phase 1*⟩, we take a long-term sequence S_N^{long} with length N from the training data. After obtaining the difference frames, the motion context of the long-term sequence is extracted by a long-term motion context encoder E_{LMC}. We adopt typical motion extractor, C3D [30] with 3D-Convs for E_{LMC}. The resulting long-term motion context feature $Z^{LMC} = \{z_l^{LMC}\}_{l=1}^{w\times h} \in \mathbb{R}^{w\times h\times c}$ is divided into local parts to exploit decomposed dynamics. The local feature $z_l^{LMC} \in \mathbb{R}^c$ is used as a memory query individually.

The parameters of the LMC-Memory have a matrix form, $M = \{m_i\}_{i=1}^s \in \mathbb{R}^{s\times c}$ with s slot size and c channels. A row vector $m_i \in \mathbb{R}^c$ denotes a memory item of M. An addressing vector $a_l = \{a_{l\_i}\}_{i=1}^s \in \mathbb{R}^s$ for query z_l^{LMC} is used to address the location of the memory M. Each scalar value a_{l_i} of a_l can be considered as an attention weight for the corresponding memory slot m_i. Memory addressing procedure can be formulated as

$$a_{l\_i} = \frac{\exp(d(z_l^{LMC}, m_i))}{\sum_{j=1}^{s} \exp(d(z_l^{LMC}, m_j))}, \tag{1}$$

where $d(\cdot,\cdot)$ indicates cosine similarity function and $\exp(\cdot)/\sum\exp(\cdot)$ denotes softmax function. With M and $a_l = \{a_{l\_i}\}_{i=1}^s$, the memory outputs a local motion context memory feature $f_l^{mem} \in \mathbb{R}^c$ $(l = 1, 2, ..., w \times h)$ for each location l as follows

$$f_l^{mem} = \sum_{i=1}^{s} a_{l\_i} m_i. \tag{2}$$

Finally, a motion context memory feature $F^{mem} = \{f_l^{mem}\}_{l=1}^{w\times h} \in \mathbb{R}^{w\times h\times c}$ is obtained by positioning each local feature f_l^{mem} as shown in Figure 2. As addressed in Section 3.1, F^{mem} is embedded to motion context-aware video prediction P. During the training phase 1, the weights of the memory M are updated through backpropagation as [9]. We train the networks to generate long-term future from long-term input so that long-term motion context can be stored in the memory at this phase.

At the matching phase ⟨*Phase 2*⟩, the model receives a short-term sequence S_n^{short} with length n (long-term length $N >$ short-term length n). The matching phase allows long-term information in the memory to be recalled by a limited short-term sequence. Similar to the long-term encoding process, the difference frames are utilized for motion encoding. Then, motion matching feature Z^{MM} is extracted by a motion matching encoder E_{MM}. Same as

Algorithm 1 Memory Alignment Learning

Inputs: short-term sequence $S_n^{short} = S_{t-n+1:t}$, long-term sequence $S_N^{long} = S_{t-n+1+r:t-n+r+N}$ (random integer $r \sim \mathcal{U}\{0, n\}$), and learning rate α.
Initialize parameters of PHASE 1 networks (θ), PHASE 2 networks (ϕ), and LMC-memory (M).

for each iteration **do**
  ⟨**PHASE 1**: STORING PHASE⟩
  Get $Z^{LMC} = E_{LMC}(S_N^{long})$
  Get $F^{mem} = LMC\text{-}Memory(Z^{LMC})$
  for $i = 0, 1, ..., K$-1 **do**
    Get $\hat{I}_{t+i+1} = P(S_{t-n+1:t}, \hat{S}_{t+1:t+i}, F^{mem})$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}^{\texttt{pred}}(\hat{S}_{t+1:t+K}, S_{t+1:t+K})$
  Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
  ⟨**PHASE 2**: MATCHING PHASE⟩
  Get $Z^{MM} = E_{MM}(S_n^{short})$
  Get $F^{mem} = LMC\text{-}Memory(Z^{MM})$
  for $i = 0, 1, ..., K$-1 **do**
    Get $\hat{I}_{t+i+1} = P(S_{t-n+1:t}, \hat{S}_{t+1:t+i}, F^{mem})$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}^{\texttt{pred}}(\hat{S}_{t+1:t+K}, S_{t+1:t+K})$
  Update ϕ $(except\ M) \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}$
end for

E_{LMC}, E_{MM} has the C3D [30] structure, but does not share parameters with E_{LMC}. The local feature z_l^{MM} of $Z^{MM} = \{z_l^{MM}\}_{l=1}^{w \times h} \in \mathbb{R}^{w \times h \times c}$ is used to recall the corresponding long-term motion context from the memory. The memory addressing procedures are the same as the first storing phase (Eq. 1 and 2). However, unlike the phase 1, the weights of the memory M are not optimized and only used to recall the motion context during this matching phase. This is to preserve the stored long-term motion context in M. Except for the memory M, overall network weights are trained to predict the long-term future frames with the memory feature. Thus, it enables E_{MM} to extract Z^{MM} that properly recalls the corresponding long-term motion context in the given LMC-memory. Optimization is performed with the prediction framework P (see Figure 2). Only the short-term sequence $S_n^{short} = S_{t-n+1:t}$ is fed as a main input of P. The memory path receives the long-term S_N^{long} and short-term S_n^{short} alternately. Two training phases are alternately performed in each iteration. In both phases, according to [29, 35–37], we exploit a prediction loss function $\mathcal{L}^{\texttt{pred}}$ as follows

$$\begin{aligned} \mathcal{L}^{\texttt{pred}} = & \|\hat{S}_{t+1:t+K} - S_{t+1:t+K}\|_2^2 \\ & + \|\hat{S}_{t+1:t+K} - S_{t+1:t+K}\|_1, \end{aligned} \tag{3}$$

where $\hat{S}_{t+1:t+K}$ denotes K predicted future frames while $S_{t+1:t+K}$ denotes K ground truth future frames. Note that the proposed method only takes short-term sequences at inference time as shown in Figure 1. Training procedure is further described in Algorithm 1.

4. Experiments

4.1. Datasets

To validate the proposed method, we utilize both synthetic and natural video datasets. We use a synthetic Moving-MNIST dataset [28] that is mainly used in video prediction. In addition, we use a KTH Action [26] and a Human 3.6M [12] datasets including natural videos with human action scenarios.

Moving-MNIST. The Moving-MNIST [28] contains the moving of two randomly sampled digits from the original MNIST dataset. Each digit moves in a random direction within a 64×64 size image with a gray scale. The constructed Moving-MNIST dataset consists of 10,000 sequences for training and 5,000 sequences for testing as [35].

KTH Action. KTH Action dataset [26] consists of 6 types of action videos for 25 subjects. It includes indoor, outdoor, scale variations, and different clothes. Each frame is resized to 128×128 with a gray scale. The videos of 1-16 subjects are used as the training set while the videos of 17-25 subjects are used as the test set. We follow the experimental setting [33] of video prediction for the KTH Action dataset.

Human 3.6M. The Human 3.6M [12] includes 17 human action scenarios with total 11 actors. It contains 4 different camera views. Each frame is resized to 64×64 with RGB color channels. Videos of subjects 1, 5, 6, 7, and 8 are used to train the model while videos of subjects 9 and 11 are used to test the model. We follow the experimental setting [34].

4.2. Implementation

The video frames are normalized to intensity of [0, 1] and resized to 64 × 64 (MNIST and Human 3.6M) or 128 × 128 (KTH) as [33–35]. The proposed model is trained by Adam optimizer [17] with a learning rate of 0.0002. Memory slot size s is fixed as 100 for all experiments. Input short-term sequence length n is set as 10. Long-term sequence length N is set as 30 (MNIST) or 40 (KTH and Human 3.6M). Our model is trained to predict corresponding N future frames. We use 4-layer ConvLSTMs for frame prediction. The overall detailed network structures are described in the supplementary material.

4.3. Evaluation

We use MSE, PSNR, SSIM [38], and LPIPS [45] to measure the performances. MSE and PSNR are calculated by the pixel-wise difference between the actual frame and the predicted frame. We also evaluate the performance using SSIM that considers the structural similarity between

Prediction Method	Performance (10 → 10)			Performance (10 → 30)			Computational Cost (10 → 30)
	MSE	SSIM	LPIPS	MSE	SSIM	LPIPS	Inference Time (s)
TRAJGRU [27]	106.9	0.713	-	163.0	0.588	-	-
CDNA [7]	97.4	0.721	-	142.3	0.609	-	-
VPN [15]	64.1	0.870	-	129.6	0.620	-	-
PredRNN [37]	56.8	0.867	-	112.2	0.645	-	-
PredRNN++ [35]	**42.1**	0.913	59.5	**84.0**	0.834	139.9	0.308
E3D-LSTM [36]	50.9	0.912	86.7	102.2	**0.849**	156.3	**0.299**
Conv-TT-LSTM [29]	53.0	**0.915**	**40.5**	105.7	0.840	**90.3**	0.378
Proposed Method	**41.5**	**0.924**	**46.9**	**73.2**	**0.879**	**71.6**	**0.099**

Table 1: Results on the Moving-MNIST. Higher SSIM values are better while lower MSE and LPIPS values are better. **Red** and **Blue** indicate the best and the second best, respectively. Ours outperforms the others especially in long-term condition.

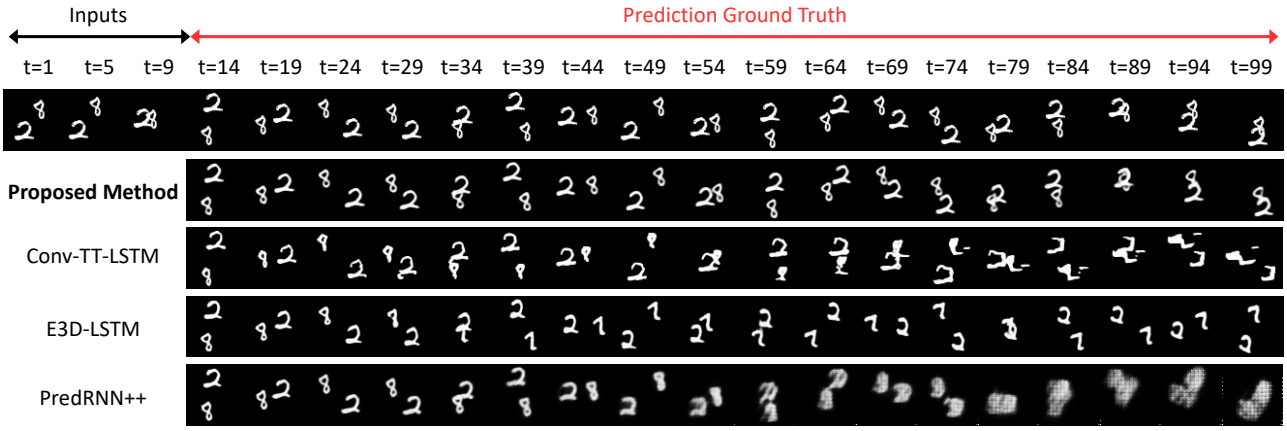

Figure 3: Qualitative results with given 10 frames on the Moving-MNIST. The other results are obtained from official sources.

frames. Furthermore, we utilize LPIPS as a perceptual metric, which tends to be similar to the human recognition system [45]. Higher values are better for PSNR and SSIM while lower values are better for MSE and LPIPS. LPIPS results are represented in 10^{-3} scale. Single TITAN XP is used to evaluate computational costs for all models. Note that official source codes are used for other methods.

Results on Moving-MNIST. Table 1 shows the performance comparisons with the state-of-the-art methods on the Moving-MNIST. The left part of the table shows the experimental results of 10 frames prediction with the input 10 frames. The right part of the table shows the experimental results for 30 frames prediction with 10 input frames. The proposed method outperforms the other state-of-the-art methods. In particular, our method far surpasses the others in predicting 30 frames in terms of the LPIPS metric. In addition, the proposed method shows much better results on the computational cost compared to the other methods. Compared to other complex RNN-based methods, we adopt simple ConvLSTMs. Further, memory feature F^{mem} is extracted only once at the beginning, which is advantageous in computational cost. Figure 3 shows examples of frames predicted by the proposed method and other video prediction methods. As shown in the figure, the predicted frames of the proposed method show convincingly similar results to the ground truth. However, the prediction results by other methods show that they lose the trajectories or the shape of digits, especially in long-term condition.

Results on KTH Action. Table 2 shows the quantitative results of the proposed method and other state-of-the-art methods on the KTH Action dataset. The left part of the table shows the experimental results for predicting 20 frames with 10 input frames. The right part of the table indicates performances for predicting the next 40 frames. As shown in the table, the proposed method mostly surpasses the other state-of-the-art methods in predicting 40 frames. Especially, it is significant in the human perceptual metric (*i.e.*, LPIPS). In addition, the proposed method shows a much faster inference speed compared to the other methods also on the KTH. Figure 4 shows qualitative long-term prediction results for input sequence with limited dynamics on the KTH. This input motion is limited because motion

Prediction Method	Performance (10 → 20)			Performance (10 → 40)			Computational Cost (10 → 40)
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	Inference Time (s)
MCNET [33]	25.95	0.804	-	-	-	-	-
FRNN [22]	26.12	0.771	-	23.77	0.678	-	-
PredRNN [37]	27.55	0.839	-	24.16	0.703	-	-
PredRNN++ [35]	**28.62**	0.888	228.9	**26.94**	0.865	279.0	**0.411**
E3D-LSTM [36]	27.92	0.893	298.4	26.55	0.878	328.8	0.422
Conv-TT-LSTM [29]	28.36	**0.907**	**133.4**	26.11	**0.882**	**191.2**	1.188
Proposed Method	**28.61**	**0.894**	**133.3**	**27.50**	**0.879**	**159.8**	**0.147**

Table 2: Results on the KTH. Higher values are better for PSNR and SSIM while lower values are better for LPIPS. **Red** and **Blue** indicate the best and the second best, respectively. Ours outperforms the others especially in long-term condition.

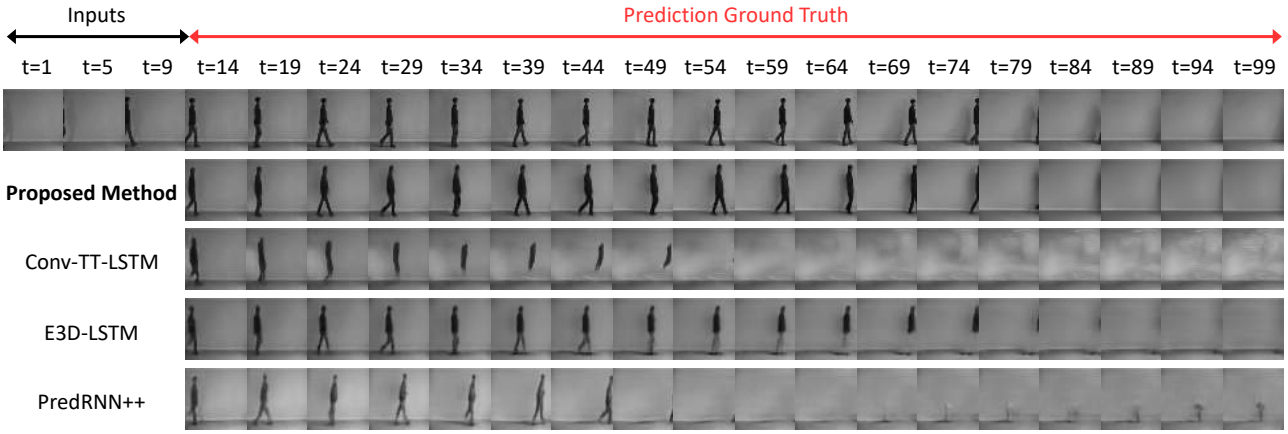

Figure 4: Qualitative results with given 10 frames on the KTH Action. The other results are obtained from official sources.

Prediction Method	Performance (10 → 40)			Performance (10 → 40, Last 10)		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
PredRNN++ [35]	23.23	0.876	106.6	22.10	0.862	123.3
E3D-LSTM [36]	22.33	0.850	113.3	21.11	0.825	140.1
Propose Method	**24.97**	**0.919**	**63.2**	**23.46**	**0.902**	**80.1**

Table 3: Results on the Human 3.6M. Higher PSNR and SSIM are better while lower LPIPS is better.

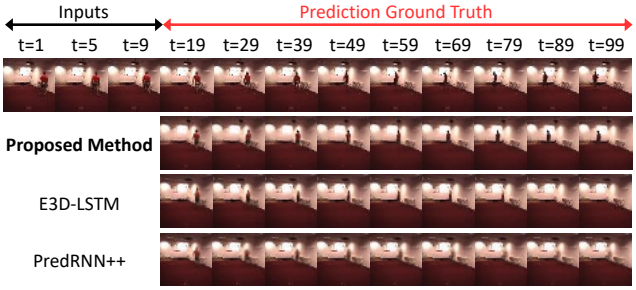

Figure 5: Qualitative results with given 10 frames on the Human 3.6M. Results of the others are from official sources.

actually starts from the middle. As shown in the figure, the other methods fail to capture the detailed leg movement, especially in long-term condition. Whereas, our predicted frames are very similar to the ground truth frames. The proposed method maintains a clear shape of the legs while following the long-term trajectories even in such a challenging condition (*i.e.*, limited dynamics).

Results on Human 3.6M. Table 3 shows the performance comparisons with the other methods on the Human 3.6M. The left part of the table shows the experimental results for predicting 40 frames with given 10 input frames. The right part of the table indicates performances for the last 10 frames among the future 40 frames. The proposed method outperforms other state-of-the-art video prediction methods both in predicting 40 future frames and predicting the last 10 frames. Figure 5 shows qualitative results for the long-term prediction on the Human 3.6M. The proposed method captures direction changing in the long-term while the other methods show the disappearance of a person at the corner. Compared to the others, the proposed method properly captures the long-term motion context with redirection.

Prediction Method	Performance (10 → 40, Last 10)			Computational Cost (10 → 40)
	PSNR	SSIM	LPIPS	Inference Time (s)
Model w/o LMC-Memory	25.29	0.851	321.1	0.118
Model w/ LMC-Memory (Non-local Motion Context)	25.57	0.854	298.8	0.136
Model w/ LMC-Memory (Local Motion Context)	**26.21**	**0.862**	**195.6**	0.147

Table 4: Effects of the network designs on the performance and the computational cost. Performance evaluations are conducted on KTH Action dataset.

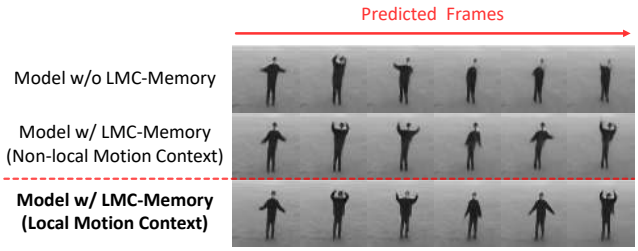

Figure 6: Video prediction qualitative results for the different network designs on the KTH Action dataset.

4.4. Ablation Study

We analyze the effects of network designs by ablating them as shown in Table 4. In detail, we investigate the effectiveness of the LMC-Memory (*i.e.*, memory alignment learning) and the local motion context (*i.e.*, memory query decomposition). The baseline, '*Model w/o LMC-Memory*' consists of the spatial encoder, the ConvLSTMs, and the frame decoder. The second one, '*Model w/ LMC-Memory (Non-local motion context)*' contains LMC-Memory but it does not adopt memory query decomposition to use local motion context as memory queries. This model uses a globally pooled motion context feature as a query. The last one indicates our final proposed model of this paper. As shown in the table, each component contributes to the performances in predicting the last 10 among 40 frames. The final model outperforms the other models, especially in terms of perceptual metric LPIPS. These results show that locally manipulated query boosts the effects of the memory since it is more accessible to store and recall the motion context with low-dimensional dynamics. Further, the additional computational cost to use the LMC-Memory is marginal.

Figure 6 shows qualitative results for different network designs. The first model does not properly capture the long-term motion. The second one predicts long-term motion to some extent. However, the detailed local parts are distorted because it addresses the motion context in an only global manner. The final model effectively predicts future frames by properly capturing the context of long-term motion.

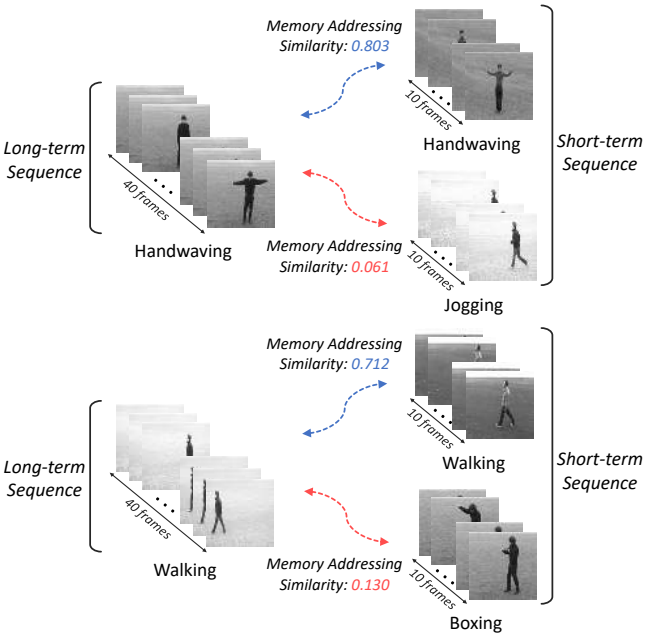

Figure 7: Examples of similarity between memory addressing vectors from long-term and short-term sequences in the KTH Action dataset.

4.5. Memory Addressing

We analyze the memory addressing for different sequences. Figure 7 shows the cosine similarity values between addressing vectors from long-term and short-term sequences including different subjects. The areas addressed in the memory are more comparable (similarity between addressing vectors is high) in the case of the same action scenario than in the case of the different actions. It shows that the long-term and short-term features that belongs to the similar action are convincingly aligned in the memory.

5. Conclusion

The objective of the proposed work is to predict future frames being aware of the long-term motion context. To this end, we propose the LMC-Memory with the alignment learning scheme to effectively store abundant long-term contexts of training data and recall suitable motion context even from limited inputs. In addition, we utilize memory query decomposition to separate overall motion into low-dimensional dynamics. It enables to cope with the high-dimensionality in terms of utilizing motion contexts in the memory. As a result, the proposed method outperforms the state-of-the-art methods with sophisticated RNNs. In particular, it is significantly noticeable in long-term condition. Further, the effectiveness of the proposed method is analyzed in both quantitative and qualitative ways.

Acknowledgement. This work was partly supported by the IITP grant (No. 2020-0-00004) and BK 21 Plus project.

References

[1] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[2] W. Byeon, Q. Wang, R. Kumar Srivastava, and P. Koumoutsakos. Contextvp: Fully context-aware video prediction. In *European Conference on Computer Vision (ECCV)*, pages 753–769, 2018. 2

[3] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei. Memory matching networks for one-shot image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4080–4088, 2018. 3

[4] L. Castrejon, N. Ballas, and A. Courville. Improved conditional vrnns for video prediction. In *International Conference on Computer Vision (ICCV)*, pages 7608–7617, 2019. 1

[5] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8483–8492, 2019. 1

[6] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning (ICML)*, 2018. 2

[7] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Neural Information Processing Systems (NeurIPS)*, pages 64–72, 2016. 1, 2, 6

[8] H. Gao, H. Xu, Q. Cai, R. Wang, F. Yu, and T. Darrell. Disentangling propagation and generation for video prediction. In *International Conference on Computer Vision (ICCV)*, pages 9006–9015, 2019. 2

[9] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, and A. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *International Conference on Computer Vision (ICCV)*, pages 1705–1714, 2019. 3, 4

[10] T. Han, W. Xie, and A. Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[11] J. Hsieh, B. Liu, D. Huang, L.F. Fei-Fei, and J.C. Niebles. Learning to decompose and disentangle representations for video prediction. In *Neural Information Processing Systems (NeurIPS)*, pages 517–526, 2018. 1

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013. 5

[13] B. Jin, Y. Hu, Q. Tang, J. Niu, Z. Shi, Y. Han, and X. Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4554–4563, 2020. 2

[14] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to remember rare events. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[15] N. Kalchbrenner, Aä. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning (ICML)*, pages 1771–1779. JMLR. org, 2017. 2, 6

[16] Y. Kim, S. Nam, I. Cho, and S.J. Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *Neural Information Processing Systems (NeurIPS)*, pages 3809–3819, 2019. 2

[17] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[18] Z. Lai, E. Lu, and W. Xie. Mast: A memory-augmented self-supervised tracker. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6479–6488, 2020. 3

[19] A. X Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2

[20] S. Lee, J. Sung, Y. Yu, and Gu. Kim. A memory network approach for story-based temporal summarization of 360 videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1419, 2018. 3

[21] M. Minderer, C. Sun, R. Villegas, F. Cole, K.P. Murphy, and H. Lee. Unsupervised learning of object structure and dynamics from videos. In *Neural Information Processing Systems (NeurIPS)*, pages 92–102, 2019. 2

[22] M. Oliu, J. Selva, and S. Escalera. Folded recurrent neural networks for future video prediction. In *European Conference on Computer Vision (ECCV)*, pages 716–731, 2018. 7

[23] H. Park, J. Noh, and B. Ham. Learning memory-guided normality for anomaly detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14372–14381, 2020. 3

[24] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *International Conference on Learning Representations Workshop (ICLRW)*, 2016. 2

[25] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y. Tai. Memory-attended recurrent network for video captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8347–8356, 2019. 3

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36. IEEE, 2004. 5

[27] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Yeung, W. Wong, and W. Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Neural Information Processing Systems (NeurIPS)*, pages 5617–5627, 2017. 6

[28] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, pages 843–852, 2015. 5

[29] J. Su, W. Byeon, F. Huang, J. Kautz, and A. Anandkumar. Convolutional tensor-train lstm for spatio-temporal learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 5, 6, 7

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 4, 5

[31] R. Villegas, D. Erhan, and H. Lee. Hierarchical long-term video prediction without supervision. In *International Conference on Machine Learning (ICML)*, pages 6038–6046, 2018. 2

[32] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Quoc V. Le, and H. Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *Neural Information Processing Systems (NeurIPS)*, pages 81–91, 2019. 2

[33] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 5, 7

[34] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. In *International Conference on Machine Learning (ICML)*, pages 3560–3569. JMLR. org, 2017. 1, 2, 5

[35] Y. Wang, Z. Gao, M. Long, J. Wang, and S.Y. Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning (ICML)*, pages 5123–5132, 2018. 1, 2, 5, 6, 7

[36] Y. Wang, L. Jiang, M.H. Yang, L.J. Li, M. Long, and L. Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations (ICLR)*, 2019. 1, 2, 5, 6, 7

[37] Y. Wang, M. Long, J. Wang, Z. Gao, and S.Y. Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Neural Information Processing Systems (NeurIPS)*, pages 879–888, 2017. 2, 5, 6, 7

[38] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[39] Y. Wu, R. Gao, J. Park, and Q. Chen. Future video synthesis with object motion prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5539–5548, 2020. 2

[40] S. Xingjian, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Neural Information Processing Systems (NeurIPS)*, pages 802–810, 2015. 1, 3

[41] J. Xu, B. Ni, and X. Yang. Video prediction via selective sampling. In *Neural Information Processing Systems (NeurIPS)*, pages 1705–1715, 2018. 2

[42] Q. Xu, H. Zhang, W. Wang, P. Belhumeur, and U. Neumann. Stochastic dynamics for video infilling. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 2714–2723, 2020. 2

[43] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani. Compositional video prediction. In *International Conference on Computer Vision (ICCV)*, pages 10353–10362, 2019. 2

[44] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations (ICLR)*, volume 2020, 2020. 2

[45] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 5, 6

[46] L. Zhu and Y. Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4344–4353, 2020. 3

[47] M. Zhu, P. Pan, W. Chen, and Y. Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5802–5810, 2019. 3