Appendix for
MOSO: Decomposing MOtion, Scene and Object for Video Prediction

A. Preprocessing Algorithm

We propose an efficient preprocessing algorithm for decomposing a video into motion, scene and object videos. The pseudo-code for the preprocessing algorithm is presented in Algorithm. 1. In particular, frame difference is calculated and employed as the motion video m_1^T. Then, a lower threshold c_{lb} and an upper threshold c_{ub} are set to filter pixels with modest differences to obtain the object video o_1^T. Finally, the left pixels are used to compose the scene video s_1^T. In Fig. 1, we show decomposed videos obtained by various combinations of c_{lb} and c_{ub}. When c_{lb} and c_{ub} are set to 0.1 and 0.9 respectively, the majority of object appearances can be separated from scenes.

Algorithm 1 Preprocessing algorithm.

Input: Video frames x_1^T
Parameter: c_{lb}, c_{ub} and channel dimension d_c
Output: Motion, scene and object videos

Let $t = 1$, $x_s = x_1$.
while t $\leq$ T **do**
  $x_{nxt} = x_T$ if $t == T$ else x_{t+1}
  $m_t = 2x_t - x_s - x_{nxt}$
  $d_{pixel} = max(abs(m_t), dim = d_c)$
  $mask = (d_{pixel} \geq c_{lb}) \odot (d_{pixel} \leq c_{ub})$
  $o_t = mask \odot x_t$
  $s_t = (1 - mask) \odot x_t$
end while
return m_1^T, s_1^T and o_1^T

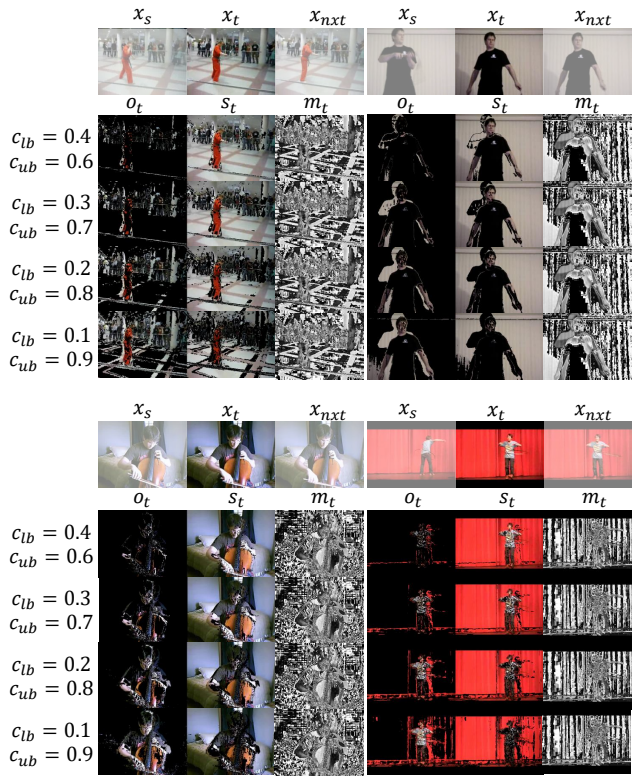

Figure 1. Visualizing the t-th frame in the decomposed motion video m_1^T, scene video s_1^T and object video o_1^T respectively through the preprocessing algorithm with different c_{lb} and c_{ub}.

B. A More General Situation of Eq. (12)

When obtaining motion tokens, several downsample layers, i.e., 2D convolutions with stride 2, first downsample videos by frame. The downsampled video frames are then concatenated in the temporal dimension and compose feature $z''_{mo} \in R^{H/f \times W/f \times T \times D}$. The temporal self-attention splits the temporal dimension into N_t working pools, obtains feature $z'_{mo} \in R^{H/f \times W/f \times N_t \times (T/N_t) \times D}$. Each working pool contains features of $\frac{T}{N_t}$ consecutive video frames and **exchange of temporal information only happens between features in the same working pool**. When $N_t = T$, no temporal information would be exchanged by the temporal self-attention, thus the t-th motion feature is obtained without the knowledge of other frames $x_k, k \neq t$. Accordingly, any change of video frames $x_k, k \neq t$ will not affect the value of the t-th motion feature. However, when $N_t = 1$, video features are obtained by interacting between each pair of video frames, thus changes in any single video frame would affect values of all video features. Considering the video prediction process of MOSO-Transformer, a pseudo video $\hat{x}_1^T$ is constructed through Eq. (10) and **has the same first K video frames** as the target video x_1^T. By partitioning the first K video frames and the others into different working pools, the first K motion tokens of x_1^K and

Table 1. Training settings of MOSO-VQVAE and MOSO-Transformer and quantitative results of video reconstruction on the UCF101, BAIR, KTH, RoboNet and KITTI datasets.

Dataset	UCF101	BAIR	KTH		RoboNet		KITTI	
Resolution	256	64	64	128	64	256	64	256
MOSO-VQVAE								
T	16	16	20	20	12	12	20	20
f_o	8	4	4	4	4	8	4	8
f_s	16	4	4	8	4	16	4	16
f_m	32	8	8	16	8	32	8	32
FPS	32	-	25	25	-	-	-	-
Batch size	2	24	16	6	24	3	24	2
Training steps	250K	250K	250K	250K	250K	250K	300K	300K
Learning rate	2e-4	2e-4	2e-4	2e-4	2e-4	2e-4	2e-4	2e-4
Scheduler	cosine	cosine	cosine	cosine	cosine	-	-	-
Discriminator Start Step	50K	50K	50K	50K	50K	50K	50K	50K
PSNR	26.9	34.2	36.3	36.1	34.1	27.8	28.4	23.3
SSIM	75.7	95.9	94.4	93.2	94.8	83.4	87.8	63.8
LPIPS	0.190	0.010	0.030	0.044	0.013	0.072	0.042	0.241
MOSO-Transformer								
Batch size	8	32	32	8	32	8	48	16
dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Transformer$_{SO/G}$ Blocks	16	16	16	16	16	16	7	12
Transformer$_M$ Blocks	8	8	8	8	8	8	7	7
Attention heads	8	8	8	8	8	8	8	8
Embedding dim.	758	758	758	758	758	758	758	758
Hidden dim.	1024	1024	1024	1024	1024	1024	1024	1024
Immediate dim.	2048	2048	2048	2048	2048	2048	2048	2048
Training steps	300K	90K	30K	85K	100K	110K	200K	250K

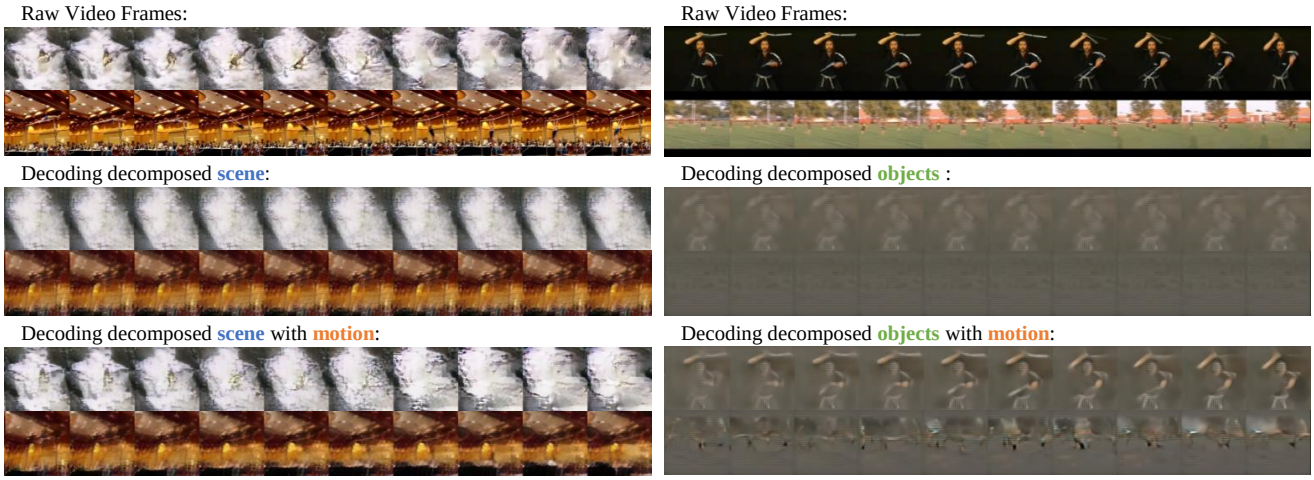

Figure 2. Visualizing decomposed objects and scenes with or without corresponding motions on UCF101.

$\hat{x}_1^K$ must be exactly **the same** as shown in Eq. (12).

A general solution to partition the given K given video frames and the subsequent ones to different working pools involves a constant hyper-parameter $c \in \{1, 2, .., K\}$, which satisfies that K can be exactly divided by c. Ensuring that T can be exactly divided by K, then N_t can be set as $\frac{cT}{K}$. When $c = 1$, the partition is the one stated in Eq. (12). When $c = K$, then non-temporal information will be exchanged.

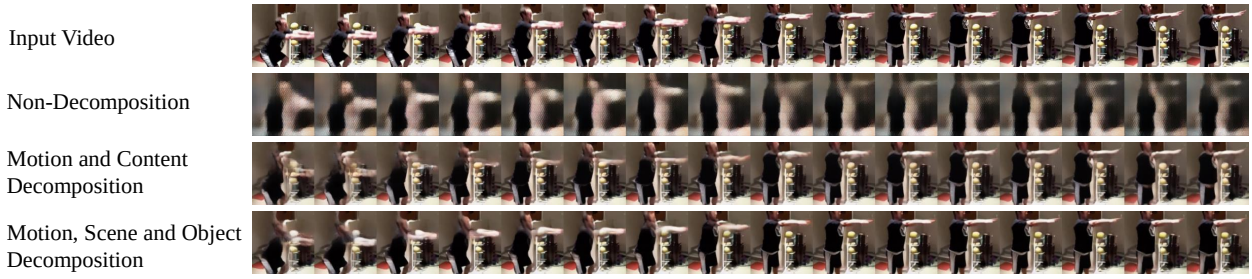

Figure 3. Qualitative comparison of ablated models on UCF101 for video reconstruction. The first row depicts the input video. The following three rows depict videos reconstructed by three ablated models.

C. Implementation Details and More Experimental Results

C.1. Hyperparameters and Optimizer

MOSO is implemented with PyTorch [9]. The specific training settings are given in Table 1, where we denote the downsample factor in the motion, scene and object encoders as f_m, f_s and f_o respectively. Adam optimizer [6] is used for both MOSO-VQVAE and MOSO-Transformer.

C.2. Ablation Experimental Settings and Qualitative Results

We conduct an ablation study to explore the necessity of decomposing object, scene and motion components. Specifically, we compare the quality of decoded videos from (a) non-decomposed features, (b) content and motion decomposed features, and (c) scene, object and motion decomposed features. To obtain non-decomposed video features, a frame-wise encoder is adopted to encode videos by frame and the merge module in the video decoder of MOSO-VQVAE is removed to reconstruct input videos. The frame-wise encoder has the same settings and architecture as the motion encoder but takes raw video frames as input. For content and motion decomposition encoding, a similar frame-wise encoder is used for encoding visual movements and a content encoder with the same structure as the scene encoder of MOSO-VQVAE is used to encode the content part. The frame-wise encoder takes frame difference as input and the content encoder is fed with raw video frames. The video decoder of MOSO-VQVAE is used to rebuild video details by summing frame-wise features with content features at multi scales. For motion, scene and object decomposition, MOSO-VQVAE is used to encode decomposed video components and decode video details. The total codebook size of all ablated models is 16384 and the dimension of all codebook entries is 256 for fair comparisons. We visualize videos reconstructed by three ablated models in Fig. 3, which demonstrates that our MOSO obtains more clear and more fidelity reconstruction results.

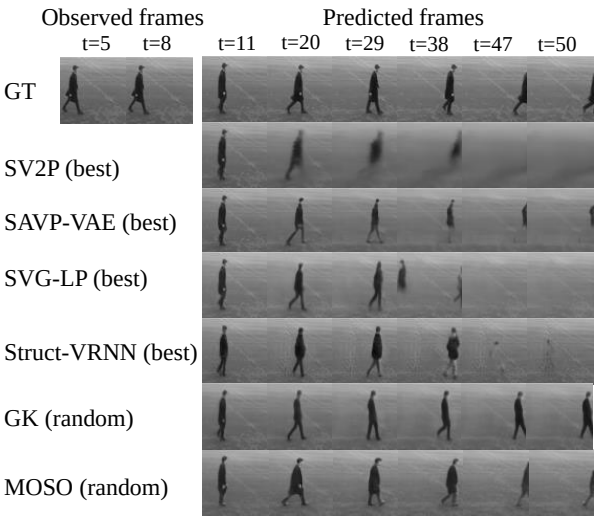

Figure 4. Qualitative comparison of MOSO and other models on KTH for video prediction.

C.3. Visualization of Decomposed Videos

As stated in the paper, the encoded video features (i.e., motion, scene and object) can be decoded by the video decoder of MOSO-VQVAE flexibly. Specifically, when decoding object features, we replace the scene and motion features with empty features filled with zeros and visualize the output of the video decoder. When only replacing the scene features with empty features, we can decode objects with motion and observe corresponding motion patterns. The decoding of scene features follows similar pipelines. Samples of visualized components are given in Fig. 2, which demonstrates that MOSO could well decompose scenes and objects and decouple different motion patterns.

C.4. More Ablation Studies

Video Decomposing We conduct an ablation study to explore the importance of motion, scene and object decomposition. Specifically, we compare the quality of reconstructed videos of MOSO-VQVAE from (a) non-

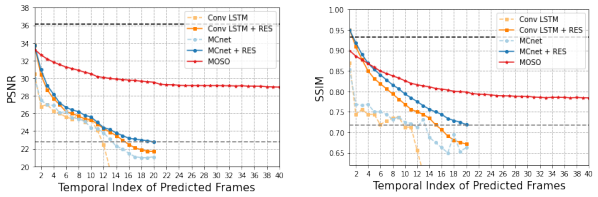

Figure 5. Quantitative comparison with prior work [10] on KTH 128^2 for video prediction. The performance of MOSO declines more slowly over the temporal index of predicted video frames. The black dashed line indicates the average reconstruction score.

Table 2. Ablation study on video decomposition methods on KTH and UCF101 for video reconstruction. *non decom.*: non-decomposition; *mo. co.*: motion and content decomposition; *mo. sc. ob.*: motion, scene and object decomposition. *pre. alg.* denotes the preprocessing algorithm.

Method	KTH			UCF101		
	PSNR↑	SSIM↑	FVD↓	PSNR↑	SSIM↑	FVD↓
non decom.	24.8	76.5	446.5	19.9	47.6	2487.2
mo. co.	32.6	86.4	238.8	28.5	75.8	1018.9
mo. sc. ob.	36.0	**95.9**	237.8	29.8	79.6	310.1
+ pre. alg.	**36.5**	**95.9**	**230.5**	**30.0**	**80.6**	**267.9**

Table 3. Ablate discriminators in MOCO-VQVAE on UCF101. $\mathcal{L}_{VD}$: loss for video discriminator; $\mathcal{L}_{ID}$: loss for image discriminator; *0.1/0.05*: loss weights.

Methods	SSIM↑	LPIPS↓	FID↓
w/o $\mathcal{L}_{VD}/\mathcal{L}_{ID}$	92.0	0.0294	24.8
$0.1\mathcal{L}_{VD}$	**92.1**	**0.0246**	**17.9**
$0.1\mathcal{L}_{VD} + 0.1\mathcal{L}_{ID}$	90.7	0.0308	19.5
$0.1\mathcal{L}_{VD} + 0.05\mathcal{L}_{ID}$	91.2	0.0277	19.7

decomposed signals, (b) content and motion decomposed signals, and (c) scene, object and motion decomposed signals on two benchmarks, i.e., KTH and UCF101. Settings for each ablated model are given in the appendix and the results are given in Table 2. By separating content from motion, the quality of rebuilt videos improves on all metrics and benchmarks. When further separating objects from scenes, the reconstruction quality further enhances. After adopting our simple but effective preprocess algorithm, our MOSO-VQVAE achieves the best reconstruction quality on both the UCF101 and KTH datasets.

Adversarial Training Inspired by VQGAN [3], we adopt video and/or image discriminators to train MOSO-VQVAE in an adversarial manner. The video and image discriminators respectively evaluate videos by clip and by frame. As shown in Table 3, using a video discriminator with a loss weight of 0.1 achieves the best LPIPS and FID and comparable SSIM. The image discriminator brings no improvement since it cannot preserve video consistency when optimizing a reconstructed video frame.

Table 4. Ablate codebooks in MOCO-VQVAE on UCF101. *sep. cb.*: each encoder adopts an independent codebook; *share cb.*: sharing codebooks used for three encoders; N: the codebook size.

Methods	N	SSIM↑	LPIPS↓	FID↓
sep. cb.	8192×3	88.5	0.0306	17.5
share cb.	8192	88.0	0.0335	17.9
	16384	**89.4**	**0.0294**	**16.5**

Codebook Sharing We conduct an ablation study on the shared codebook as reported in Table 4. When sharing codebooks to quantize different features (i.e. motion, scene and object features), we obtain better reconstruction performance with a smaller total codebook size. There exist two potential causes. Firstly, similarly to quantizing features with multi-scales, which has improved the performance of VQ-VAE on image reconstruction [5], quantizing features with multi-perspectives can make the codebook more diverse and informative. Second, the regions of features obtained by the three encoders may partially but not entirely overlap. Thus codebook sharing boosts performance with a one-third reduction in total codebook size, e.g., 16384 versus 8192×3. In contrast, when the shared codebook is the same size as each individual codebook, performance degrades due to codebook sharing.

C.5. More Experimental Results

We qualitatively compare MOSO with prior works on KTH at 64^2 resolution in Fig. 4. When SV2P [1], SAVP-VAE [7], SVG-LP [2] and Struct-VRNN [8] fail to synthesize consistent human objects in the last several frames, GK [4] and our MOSO could predict a long future video with consistent object identities and reasonable subsequent actions. Moreover, our MOSO generates more distinct object identities and more realistic actions. The better performance benefits from the decomposition of motion, scene and object, which helps to model varied motions and reduce disturbance of motion artifacts on object identities.

At 128^2 resolution, we compare MOSO with prior work [10] quantitatively in Fig. 5. The black dashed line represents the average reconstruction score of MOSO-VQVAE on PSNR and SSIM, which becomes the upper bound for MOSO-Transformer on video prediction. We train MOSO-VQVAE with negative SSIM loss on all videos and remove the discriminator loss on KTH 128^2. As shown in Fig. 5, MOSO outperforms prior work [10] on PSNR and SSIM after the 2nd and 5th predicted frames respectively, and the performance of MOSO declines much more slowly over time, demonstrating its potential on generating long videos.

C.6. Additional Samples

To facilitate visualization, we provide additional samples of MOSO via a website: https://iva-

mzsun.github.io/MOSO.

References

[1] 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. 4

[2] 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. 4

[3] 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. 4

[4] 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. 4

[5] Taehoon Kim, Gwangmo Song, Sihaeng Lee, Sangyun Kim, Yewon Seo, Soonyoung Lee, Seung Hwan Kim, Honglak Lee, and Kyunghoon Bae. L-verse: Bidirectional generation between image and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16505–16515, 2022. 4

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 3

[7] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, abs/1804.01523, 2018. 4

[8] 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. 4

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 3

[10] 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. 4