ModeRNN: Harnessing Spatiotemporal Mode Collapse in Unsupervised Predictive Learning

Zhiyu Yao, Yunbo Wang, Haixu Wu, Jianmin Wang, and Mingsheng Long, *Member, IEEE*

***Abstract*—Learning predictive models for unlabeled spatiotemporal data is challenging in part because visual dynamics can be highly entangled, especially in real scenes. In this paper, we refer to the multi-modal output distribution of predictive learning as *spatiotemporal modes*. We find an experimental phenomenon named *spatiotemporal mode collapse* (STMC) on most existing video prediction models, that is, features collapse into invalid representation subspaces due to the ambiguous understanding of mixed physical processes. We propose to quantify STMC and explore its solution for the first time in the context of unsupervised predictive learning. To this end, we present ModeRNN, a *decoupling-aggregation* framework that has a strong inductive bias of discovering the compositional structures of spatiotemporal modes between recurrent states. We first leverage a set of *dynamic slots* with independent parameters to extract individual building components of spatiotemporal modes. We then perform a weighted fusion of slot features to adaptively aggregate them into a unified hidden representation for recurrent updates. Through a series of experiments, we show high correlation between STMC and the fuzzy prediction results of future video frames. Besides, ModeRNN is shown to better mitigate STMC and achieve the state of the art on five video prediction datasets.**

***Index Terms*—Predictive learning, mode collapse, spatiotemporal modeling, recurrent neural networks.**

I. Introduction

PREDICTIVE learning is to discover reasonable patterns of unlabeled spatiotemporal data with unsupervised learning approaches [1]. However, it is challenging in complex real-world environments, where the underlying physical processes of the entire dataset can be highly mixed and cannot be easily distinguished from high-dimensional visual observations. The entanglement of the learned representations usually leads to ambiguous future predictions.

Manuscript received 28 May 2022; revised 27 April 2023; accepted 23 June 2023. Date of publication 10 July 2023; date of current version 3 October 2023. This work was supported in part by the National Key R&D Project under Grant 2021YFC3000905, in part by the National Natural Science Foundation of China under Grants 62022050, 62021002, 62250062, and 62106144, in part by Beijing Nova Program under Grant Z201100006820041, and in part by BNRist Innovation Fund under Grant BNR2021RC01002. The work of Yunbo Wang was supported in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, and in part by Shanghai Sailing Program under Grant 21Z510202133. Recommended for acceptance by J. Li. *(Zhiyu Yao and Yunbo Wang contributed equally to this work.) (Corresponding author: Mingsheng Long.)*

Zhiyu Yao, Haixu Wu, Jianmin Wang, and Mingsheng Long are with the School of Software, BNRist, Tsinghua University, Beijing 100084, China (e-mail: yaozy15@gmail.com; whx20@mails.tsinghua.edu.cn; jimwang@tsinghua.edu.cn; mingsheng@tsinghua.edu.cn).

Yunbo Wang is with MoE Key Lab of AI, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yunbow@sjtu.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2023.3293145

Inspired by the definition of "*mode*" in generative modeling approaches such as GANs [2], in predictive learning, we assume that the video dataset has a multimodal data distribution (termed as "***spatiotemporal modes***") due to different physical dynamics, which needs to be represented accordingly in the feature space with multiple peaks. An insufficient understanding of the spatiotemporal modes is prone to result in fuzzy predictions. *Spatiotemporal modes* are considered to have the following properties:

- *A mixture of spatiotemporal modes naturally exists in video datasets. They can be viewed as (unlabeled) subsets of data samples with similar global representations in spacetime.*
- *Multiple modes can be represented by different compositions based on the same set of feature subspaces, which are modeled with a new set of recurrent structures named "***dynamic slots***"*[1].

For example, when we train a robot arm to perform multiple visual control tasks, we often need to simulate the environment changes. Notably, various tasks can share the same set of fundamental dynamics components (such as rotations and local movements of the robotic arm). But from a global view, they present entirely different spatiotemporal modes in global dynamics and visual appearance (such as pushing, grasping, picking-and-placing, etc.).

A. Spatiotemporal Mode Collapse

However, it is difficult to discover the hierarchy of local and global representations of different spatiotemporal modes. For most existing video prediction models, we observe an experimental phenomenon named ***spatiotemporal mode collapse*, (STMC)**, that is, the predictive models cannot effectively fit a wide variety of spatiotemporal modes. Due to the complexity of real-world data, STMC occurs when the model cannot effectively decouple mixed spatiotemporal modes and infer their underlying structures. With a limited number of model parameters, the learned features responding to different modes may interfere and compete in the training process, thus easily losing diversity and collapsing to a meaningless average of multiple valid modes.

When STMC occurs, we will have two key experimental findings. First, if the models are trained with complex video sequences with diverse dynamics patterns, *e.g.*, the examples in

[1]The concept of "*slot*" was initially introduced by Locatello et al. [3] to denote the object-centric features in static scene understanding. We borrow this term for the representation subspace in spacetime.

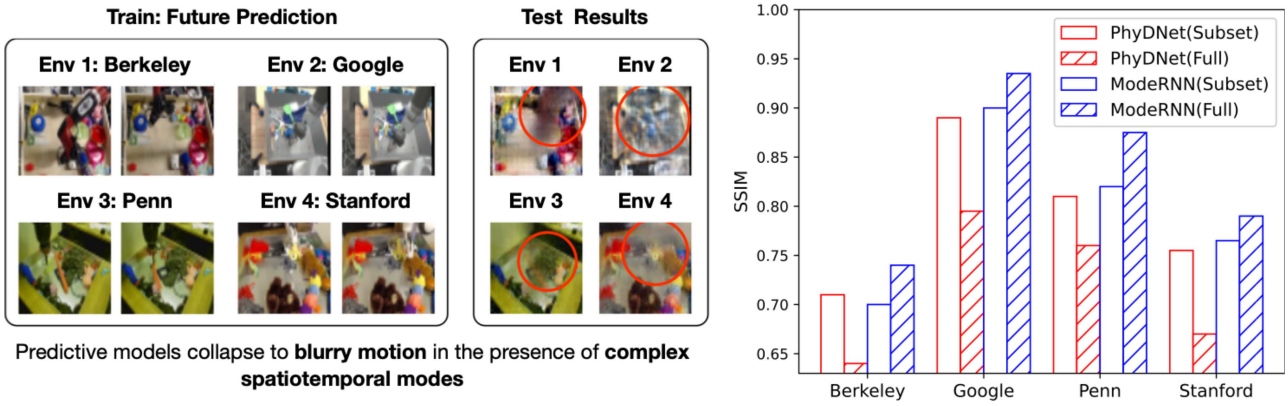


Fig. 1. Illustrations of *spatiotemporal mode collapse* on the multi-source RoboNet dataset [4]. **(Left)** RoboNet contains complex dynamics modes in videos collected in various environments. The predicted images collapse to blurry motions due to the incompatibility of learning various dynamics modes. **(Right)** The prediction results of most previous approaches (*e.g.*, PhyDNet [6]) drop a lot when being trained with data from multiple environments as shown by the red bars ("Subset" here indicates the models are separately trained only with data from individual environments; "Full" means that the model is trained on the entire dataset). In contrast, the proposed ModeRNN benefits from large-scale training across different environments as shown by the dashed blue bars.

Fig. 1 (Left), they may tend to produce fuzzy prediction results of future frames. Similar to the well-known mode collapse problem in GANs, the prediction results based on different input sequences collapse into one or a small subset of the true multimodal data distribution. Second, as shown in Fig. 1 (Right), we find that training individual models separately for each data environment[2] can obtain better performance (indicated by the hollow red bars), while training the same model across various environments (with $4\times$ training data) leads to a significant decline in the prediction results (indicated by the dashed red bars). These empirical findings can be counter-intuitive, because the model fails to further benefit from scalable training on large-scale multimodal video datasets due to STMC.

To quantify the influence of STMC, we propose a new evaluation metric based on the *silhouette coefficient*. Specifically, we use DBSCAN [5] to identify clusters of features throughout the entire dataset and then measure their purity with the silhouette value to assess the disentanglement of the learned spatiotemporal modes.

B. Key Idea of Our Approach

We explore STMC for the first time in *unsupervised* predictive learning. The core idea is to discover the compositional structures of latent modes. To this end, we propose ModeRNN (see Fig. 2), a new modular recurrent architecture that learns structured visual dynamics through a set of *dynamic slots*, where each of them responds to the representation subspace of a single spatiotemporal mode. ModeRNN also introduces a *decoupling-aggregation* framework to process the slot features in three stages, which is completely different from existing predictive models with modular architectures [7], [8].

[2]To demonstrate the existence of STMC, we roughly regard each data environment in the RoboNet dataset [4] as a spatiotemporal mode.

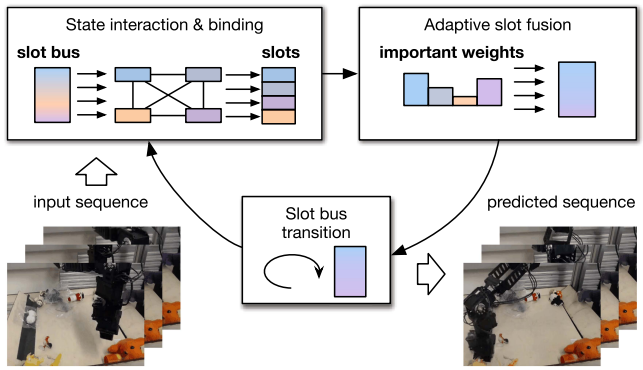


Fig. 2. Overview of the recurrent transitions in our approach. The global slot bus features are separated into multiple dynamic slots, corresponding to different representation subspaces. The dynamic slots are then re-combined into the slot bus with sample-specific importance weights and transit to the next time step. Using the hierarchical representations, the model learns to separate mixed spatiotemporal modes.

The first stage of ModeRNN is recurrent state interaction and slot binding, in which we use the multi-head attention mechanism [9] to enable the memory state to interact with the input state and previous hidden state of RNNs. We name the memory state "*slot bus*", because for each sequence, it is initialized from a multi-variate Gaussian distribution with learnable parameters, and thereafter refined using the slot features at each time step. By using the slot bus features as the queries, multi-head attention can naturally decouple modular components from hidden representations and bind them to particular slots. Features in each slot are then independently modeled using per-slot convolutional parameters. The second stage in each ModeRNN unit is slot fusion, motivated by the assumption that there can be multiple spatiotemporal modes in a single video and similar videos can be represented by similar compositional structures over the slot features. Therefore, we assign slot features with learnable importance weights and aggregate them into a unified hidden

representation, which is then used in the third stage to update the slot bus and generate the output state of the ModeRNN unit.

Experiments are conducted on five challenging datasets, including (a) the large-scale, real-world RoboNet dataset with diverse data collection environments and multiple robot control tasks, (b) the KTH dataset with different action categories, (c) the radar echo dataset with various dynamics of seasonal climate, (d) the Human3.6M dataset, and (e) the synthetic Mixed Moving MNIST dataset.

Our contributions can be summarized as follows:

- We witness a new experimental phenomenon in video prediction, called STMC, which affects the prediction quality due to the entanglement of features corresponding to different spatiotemporal modes. We also propose to quantify STMC with a new metric.
- We propose a modular and hierarchical architecture to mitigate STMC, which decouples the mixed visual dynamics based on the learned slot patterns, and then re-combines them adaptively to present higher-level semantics.
- Through a series of experiments, we demonstrate the existence of STMC quantitatively and qualitatively. We also validate the effectiveness of ModeRNN in mitigating STMC in the feature space and thus improving future prediction in the pixel space.

II. Related Work

RNN-Based Video Prediction Models. Many deep learning models based on RNNs have been proposed for spatiotemporal prediction [10], [11], [12], [13], [14], [15]. ConvLSTM [12] integrated 2D convolutions into the recurrent state transitions of standard LSTM and proposed the convolutional LSTM (ConvLSTM) network, which can model the spatial correlations and temporal dynamics in a unified recurrent unit. More recent approaches have extended the prediction ability of ConvLSTM in different aspects [6], [16], [17], [18], [19], [20], [21], [22], [23], [24]. For example, as an important compared model of our approach, SA-ConvLSTM [23] incorporates self-attention in the recurrent state transitions in ConvLSTM to obtain more global context information across time. However, unlike our approach, it does not learn decoupled representations to understand individual components in complex visual dynamics. In Section IV, ModeRNN is compared with the previous art including SA-ConvLSTM [23], CrevNet [21], E3D-LSTM [19], STMFANet [25], and LMC [24].

Probabilistic Video Prediction Models. Besides the deterministic models, probabilistic models were proposed to explicitly consider the uncertainty in future prediction [26], [27], [28], [29], [30], [31], [32], [33], [34]. Although many probabilistic prediction models [32], [35], [36] are also designed to decouple the multimodal distributions of future frames, they mainly focus on uncertainty modeling and do not share the same basic ideas as ModeRNN for disentanglement learning of spatiotemporal representation subspaces. In experiments, we mainly compare our model with cutting-edge probabilistic approaches, including SVG [31], SAVP [37], hierarchical VRNN [32], and SRVP [38]. We find that the use of explicitly separated slot features leads to better decoupling ability than the implicit Gaussian representations. It derives a better feature clustering result that is closer to the prior knowledge.

Unsupervised Predictive Learning for Spatiotemporal Disentanglement. Previous work has focused on learning to disentangle the spatial and temporal features from visual dynamics [6], [39], [40], [41]. These methods factorize spatiotemporal data into feature subspaces with strong priors, *e.g.*, assuming that the spatial information is temporally invariant. Another line of work is to learn predictive models for unsupervised scene decomposition such as [7], [40]. Unlike the above models, our approach uses a set of modular architectures in the recurrent unit to represent the mixed spatiotemporal dynamics. The most relevant work to our method is the *Recurrent Independent Mechanism* (RIM) [8], which consists of largely independent recurrent modules that are sparsely activated and interact via soft attention. ModeRNN is different from RIM in three aspects. First, it is specifically designed to tackle STMC in real-world environments. Second, it learns modular features by incorporating multi-head attention in the recurrent unit, and performs state transitions on compositional features with learnable importance weights. Third, the modular structures in ModeRNN are frequently activated responding to the mixed visual dynamics. We mainly use DDPAE [40] and PhyDNet [6] as the baselines of ModeRNN for spatiotemporal disentanglement learning in the following sections.

III. ModeRNN

We propose ModeRNN to reduce *spatiotemporal mode collapse* (STMC) in unsupervised predictive learning. The key idea is to build a *decoupling-aggregation* framework to model the recurrent state transitions of mixed spatiotemporal modes. We propose a novel recurrent unit named ModeCell as the building block of ModeRNN. It consists of three modules: (i) the state interaction and slot binding module, (ii) the adaptive slot fusion module, and (iii) the slot bus transition module.

In this section, we first discuss the basic network components and the overall architecture of ModeRNN. Following this, we will delve into the details of the decoupling-aggregation recurrent unit.

A. Dynamic Slots & Slot Bus

Dynamic slots. The decoupling-aggregation framework is built upon a set of hidden representations named *dynamic slots*. The term *slot* is in part borrowed from previous work for unsupervised scene decomposition [3]. We use it here to respond to a family of similar visual dynamics, that is, to bind each dynamic slot to the feature subspace of each spatiotemporal mode one-to-one. Slot features can be viewed as latent factors that can explicitly improve the unsupervised decoupling of mixed dynamics across the dataset.

Slot Bus. Assuming that the spatiotemporal mode consists of multiple slot patterns, so we dynamically combine all slot features with different importance weights into a unified feature representation for updating long-term memory states, termed *slot bus*. The hierarchical structure of dynamic slots and the slot

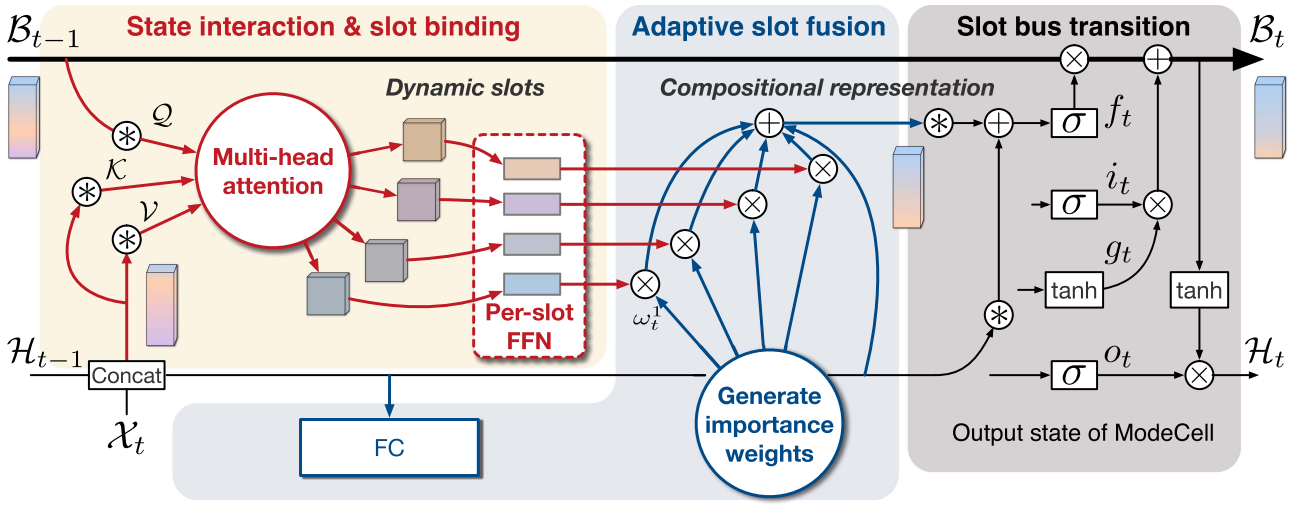


Fig. 3. The architecture of ModeCell. It tackles spatiotemporal mode collapse via a decoupling-aggregation framework based on dynamic slots $\{\text{slot}_t^1, \ldots, \text{slot}_t^N\}$. $\mathcal{B}_{t-1}$ denotes the slot bus of ModeRNN. The final model, ModeRNN, consists of multiple stacked ModeCells.

bus leads to a better understanding of complex and highly mixed dynamic patterns. The model is allowed to learn similar compositional structures over the slots from similar data samples with similar spatiotemporal modes. On the contrary, data samples with distinct visual dynamics may represent significant differences in the learned importance weights for updating the slot bus features, which provides a solution to STMC. Specifically, the slot bus is initialized from a learnable, multi-variate Gaussian distribution, whose mean and variance encode the global priors for the entire dataset.

B. Architecture of ModeRNN

ModeRNN consists of multiple stacked ModeCells. In the framework, the output state $\mathcal{H}_t^l$ of the l-th ModeCell transits between predictive blocks as it is in the ConvLSTM network [12]. Besides, the slot bus $\mathcal{B}_t^l$ of the l-th ModeCell transits recurrently within each ModeCell:

$$\mathcal{H}_t^l, \mathcal{B}_t^l = \text{ModeCell}_l\left(\mathcal{X}_t^l, \mathcal{H}_{t-1}^l, \mathcal{B}_{t-1}^l\right), \quad l \in [1, L], \tag{1}$$

where $\mathcal{X}_t^l$ is the input state. It is the input image for $l = 1$; Otherwise, $\mathcal{X}_t^l = \mathcal{H}_t^{l-1}$, which is the output state from the previous ModeCell unit, representing the connections between the stacked ModeCells. We present the internal details within ModeCell in the next section. We omit the layer index for brevity in the following notations and equations.

C. ModeCell

To learn and leverage the dynamic slots, we introduce a novel recurrent unit named ModeCell, which follows a decoupling-aggregation framework with three modules, *i.e.*, the state interaction and slot binding module p_θ, the adaptive slot fusion module p_ω, and the slot bus transition module p_ϕ. For each ModeCell, (1) can be unrolled as follows:

$$\begin{aligned} \text{Slot binding:} \quad & p_\theta(\text{slot}_t^1, \ldots, \text{slot}_t^N \mid \mathcal{X}_t, \mathcal{H}_{t-1}, \mathcal{B}_{t-1}) \\ \text{Adaptive slot fusion:} \quad & p_\omega(\mathcal{F}_t \mid \text{slot}_t^1, \ldots, \text{slot}_t^N, \mathcal{I}_t) \\ \text{Slot bus transition:} \quad & p_\phi(\mathcal{H}_t, \mathcal{B}_t \mid \mathcal{F}_t, \mathcal{I}_t, \mathcal{B}_{t-1}), \end{aligned} \tag{2}$$

where $\{\text{slot}_t^1, \ldots, \text{slot}_t^N\}$ represent N dynamic slots, $\mathcal{I}_t$ denotes the concatenation of input state and hidden state (*i.e.*, $\mathcal{I}_t = [\mathcal{X}_t, \mathcal{H}_{t-1}]$), and $\mathcal{F}_t$ is the fused representation of the slot features. The slot bus $\mathcal{B}_t$ and the output state $\mathcal{H}_t$ are updated recurrently through the slot bus transition module. Next, we discuss the detailed structures of these modules.

1) State Interaction and Slot Binding: This module decouples the mixed spatiotemporal modes from raw video frames to dynamic slots. To achieve this, as shown in Fig. 3, it first uses multi-head attention to allow the slot bus to interact with the input and hidden states of the unit, and thereby divides them into separate subspaces. It then binds the features to each dynamic slot using neural networks with per-slot independent parameters.

Multi-head attention [9] is widely used in neural language and image processing, and in this work, it is incorporated in the state transitions of ModeRNN. This attention mechanism allows interactions between the previous slot bus $\mathcal{B}_{t-1}$, the current input state $\mathcal{X}_t$, and the previous hidden state $\mathcal{H}_{t-1}$ (see Fig. 3). It can naturally decouple modular components from hidden representations and bind them to particular dynamic slots. Formally, at each time step, we first apply 2D convolution projections parameterized by $W_\mathcal{Q}$ to $\mathcal{B}_{t-1}$. Like the slot attention method previously designed for static scene understanding [3], we flatten these features to 1D to facilitate the efficiency of the subsequent multi-head attention. We then split the features into N dynamic slots along the channel dimension, such that $\{\text{slot}_{t-1}^1, \ldots, \text{slot}_{t-1}^N\} = \text{Split}(\text{Reshape}(W_\mathcal{Q} * \mathcal{B}_{t-1}))$. Note that slot bus $\mathcal{B}_{t-1} \in \mathbb{R}^{d_h \times d_w \times (d_x + d_s)}$ and each $\text{slot}_{t-1}^n \in \mathbb{R}^{d_h d_w (d_x + d_s)/N}$, where d_x is the channel number of input state,

d_s is that of hidden state, and $d_h \times d_w$ indicates the spatial resolution of the slot bus tensor. To improve efficiency, we use two 3×3 depth-wise separable convolutions [42] for $W_{\mathcal{Q}}$. We use $\{\text{slot}_{t-1}^1, \ldots, \text{slot}_{t-1}^N\}$ as the queries $\{\mathcal{Q}_t^1, \ldots, \mathcal{Q}_t^N\}$ in multi-head attention, and apply similar operations to obtain keys $\{\mathcal{K}_t^1, \ldots, \mathcal{K}_t^N\}$ and values $\{\mathcal{V}_t^1, \ldots, \mathcal{V}_t^N\}$ based on $\mathcal{I}_t = [\mathcal{X}_t, \mathcal{H}_{t-1}]$. We then perform multi-head attention and reshape the N output slot features back to 3D tensors:

$$\text{slot}'^n_t = \text{Reshape}\left(\text{softmax}\left(\frac{\mathcal{Q}_t^n \mathcal{K}_t^{n\intercal}}{\sqrt{d_k}}\right)\mathcal{V}_t^n\right), \tag{3}$$

where d_k is the dimensionality of the key vectors used as a scaling factor and $n \in \{1, \ldots, N\}$. It is worth noting that the number of dynamic slots N is pre-defined and fixed across the video sequence. In Section IV-B4, we will show how to determine this hyperparameter.

The motivation for the use of multi-head attention is to decouple the hidden states into different subspaces of spatiotemporal dynamic slots. It brings two benefits to the forward modeling of spatiotemporal data. First, since $\mathcal{B}_{t-1}$ can be unrolled along the recurrent state transition path to be represented as the transformation of slot features at the previous time step, using $\mathcal{B}_{t-1}$ as attention queries allows the model to extract features from $\mathcal{X}_t$ and $\mathcal{H}_{t-1}$ by jointly attending to prior information at different slots. Second, the architecture with N attention heads can naturally help factorize the hidden representation into N subspaces, corresponding to N dynamic slots. The output at each attention head is then updated by a per-slot feed-forward network (FFN) with independent parameters ($n \in \{1, \ldots, N\}$):

$$\text{slot}_t^n = \text{FFN}^n\left(\text{slot}'^n_t\right) = \max\left(0, W_{\text{FFN}}^n * \text{slot}'^n_t\right), \tag{4}$$

where $*$ denotes the convolution operator and W_{FFN}^n are 3×3 convolution kernels. Through random parameter initialization and stochastic gradient descent, the independent networks $\{W_{\text{FFN}}^1, \ldots, W_{\text{FFN}}^N\}$ would most likely be optimized into parameter subspaces far from each other, thus forcing the slots to bind to various modes in mixed visual dynamics.

2) Adaptive Slot Fusion: We explicitly consider multiple modes sharing a set of hidden representation subspaces via the slots, and use the adaptive slot fusion module to aggregate the decoupled slot features through importance weights. The similarity of visual dynamics is reflected in the similar importance weights of dynamic slots, while different visual dynamics can be distinguished by different significance of each slot feature. This mechanism prevents ModeRNN from making ambiguous predictions

The implementation of this module is largely inspired by the *mixture of experts (MoE)* [43], which introduces the gated networks to control the information flow from base models in an ensemble. We improve the gated architecture to dynamically aggregate decoupled slots $\{\text{slot}_t^1, \ldots, \text{slot}_t^N\}$ with learnable importance weights $\{\omega_t^1, \ldots, \omega_t^N\}$, and finally have the compositional feature $\mathcal{F}_t$ based on the learned importance weights and corresponding slot features for $n \in \{0, \ldots, N\}$:

$$\omega_t^n = \text{FC}^n\left(\text{FC}\left(\sum_{i=1}^{d_h}\sum_{j=1}^{d_w}\frac{\mathcal{I}_t(i,j)}{d_h \times d_w}\right)\right), \tag{5}$$

$$\mathcal{F}_t = \omega_t^0 \cdot \mathcal{I}_t + \sum_{n=1}^{N} \omega_t^n \cdot \text{slot}_t^n. \tag{6}$$

In (5), we use the global average pooling to encode the contextual information, which is the concatenation $\mathcal{I}_t$ of the current input state and previous hidden state, into dimensionality $\mathbb{R}^{(d_x+d_s)}$. For memory efficiency, we use a simple fully connected (FC) layer to reduce the dimensionality and get the compact feature in $\mathbb{R}^{(d_x+d_s)/2}$. Then we introduce $(N+1)$ slot-independent FC layers $\{\text{FC}^0, \ldots, \text{FC}^N\}$ with the Sigmoid activation function to generate the importance weights ω_t^n for the original input $\mathcal{I}_t$ and each slot_t^n. In (6), we aggregate all dynamic slots as well as the input into a compositional representation $\mathcal{F}_t$ based on the learned importance weights. Intuitively, the shortcut connection from $\mathcal{I}_t$ to $\mathcal{F}_t$ with importance weight ω_t^0 plays an important role in integrating the input state with the slot features. We provide an ablation study to show that this gated shortcut is effective in the disentanglement of different modes.

3) Slot Bus Transition: The compositional state $\mathcal{F}_t$ builds a hierarchical representation on top of the slot features. We use four sets of $\mathcal{F}_t$ and $\mathcal{I}_t$ to form the input gate i_t, forget gate f_t, output gate o_t, and modulated slot bus input g_t. We then update the slot bus following an LSTM-style recurrent transition mechanism:

$$\begin{pmatrix} g_t \\ i_t \\ f_t \\ o_t \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} \circ \begin{pmatrix} W_g \\ W_i \\ W_f \\ W_o \end{pmatrix} * [\mathcal{F}_t, \mathcal{I}_t],$$

$$\mathcal{B}_t = f_t \odot \mathcal{B}_{t-1} + i_t \odot g_t,$$

$$\mathcal{H}_t = o_t \odot \tanh(\mathcal{B}_t). \tag{7}$$

Here, the output state of ModeCell is generated as $\mathcal{H}_t$, which is taken as input by the next ModeCell at the upper level when multiple ModeCells are being stacked in ModeRNN. In other words, ModeCell is to ModeRNN what LSTM is to the stacked LSTM network.

IV. Experiments

We quantitatively and qualitatively evaluate ModeRNN on four datasets widely used in the unsupervised predictive learning literature. Datasets and code are available at https://github.com/thuml/ModeRNN.

A. Experimental Setup

1) Datasets: RoboNet. The RoboNet dataset [4] includes more than 15 million video frames collected by 7 different robot arms from 4 environments. It thus contains a large diversity of spatiotemporal modes of rigid motions. We randomly select 4,000 videos for testing and use the others for training. We follow

the action-free and action-conditioned video prediction setups from the work of SV2P [36]. In the action-free setup, models are trained to predict the next 10 frames from the previous 5 frames. In the action-conditioned setup, models are trained to predict 10 frames based on 2 observations along with the robot action vectors at all time steps. All images are resized to 64×64.

KTH. The KTH dataset [44] contains 6 action categories and involves 25 subjects in 4 different scenarios. It thus naturally contains various modes responding to similar motion dynamics. We use person 1-16 for training and 17-25 for testing, resize each frame to the resolution of 128×128, and predict 20 frames from 10 observations.

Human3.6M. The Human3.6M dataset contains $2{,}220$ sequences for training, 300 for validation, and $1{,}056$ for testing, involving 17 different scenarios. This dataset contains a larger diversity and complexity in spatiotemporal modes. We follow the protocol from [18] to resize each RGB frame to the resolution of 128×128 and make the model predict 4 future frames based on 4 previous ones.

Radar Echo. This dataset contains $30{,}000$ sequences of radar echo maps for training, and $3{,}769$ for testing. It naturally contains various spatiotemporal modes of fluid dynamics due to seasonal climates. Models are trained to predict the next 10 radar echoes based on the previous 10 observations. All frames are resized to the resolution of 384×384.

Mixed Moving MNIST. We introduce the synthetic Mixed Moving MNIST dataset consisting of sequences of frames in the resolution of 64×64. It contains $30{,}000$ training sequences, $6{,}000$ validation sequences, and $5{,}000$ testing sequences. Each sequence consists of 20 consecutive frames (10 for input and 10 for future prediction). To approximate the multi-mode phenomenon of the natural world, we (i) randomly set the number of the flying digits in each sequence to 2 or 3 and (ii) assign each digit a random color from {red, green, blue}. The modified Moving MNIST dataset is more challenging than its previous convention used in [16], which requires the model to handle various spatiotemporal dynamics due to different frequencies of digit occlusions and different digit colors.

2) Implementation Details: We train the models with the L_2 reconstruction loss and use the ADAM optimizer [46] with a starting learning rate of 0.0003. The batch size is set to 8, and the training process is stopped after $80{,}000$ iterations. Empirically, we use 4 stacked recurrent units (*i.e.*, ModeCells) in ModeRNN with 64-channel hidden states (*i.e.*, $d_s = 64$ for the slot bus features in each ModeCell). We apply the same number of hidden states to the compared models based on recurrent units. Besides, in Section IV-B4, we discuss how to determine the number of dynamic slots. All experiments are implemented in PyTorch [47] and conducted on NVIDIA TITAN-RTX GPUs. We follow previous literature to train all the compared models from scratch without any pre-training phases. We run all experiments three times and use the average results for quantitative evaluation.

3) Compared Models: In the following experiments, we compare ModeRNN with

- Other deterministic models, including ConvLSTM [12], PredRNN [16], DDPAE [40], RIM [8], E3D-LSTM [19], CrevNet [21], SA-ConvLSTM [23], PhyDNet [6], LMC [24], STMFANet [25], and SimVP [45].
- Probabilistic video prediction approaches, including SVG [31], SAVP [37], hierarchical VRNN [32], and SRVP [38].

4) Evaluation Metrics: Measures of prediction quality. We adopt the evaluation metrics commonly used in previous literature [1] for different datasets. Specifically, we use the *mean squared error* (MSE) and the *structural similarity* (SSIM) [48] for RoboNet and Mixed Moving MNIST, and use SSIM and the *peak signal-to-noise ratio* (PSNR) for KTH. For Human3.6M, besides PSNR, we use the *learned perceptual image patch similarity* (LPIPS), and the *Fréchet video distance* (FVD) [49]. LPIPS and FVD are perceptual metrics that respectively employ static/sequential deep features to measure the quality of each generated video frame and the temporal coherence of video content. These metrics are particularly useful for evaluating the model on complex natural images, such as the human motion videos in the Human3.6M dataset. For the radar echo dataset, we further use the *critical success index* (CSI), which is defined as $\text{CSI} = \frac{\text{Hits}}{\text{Hits+Misses+FalseAlarms}}$. Given a threshold value of precipitation alarms, "Hits" corresponds to the true positive, "Misses" corresponds to the false positive, and "FalseAlarms" corresponds to the false negative. We here set the alarm threshold to 30 dBZ. Therefore, a higher CSI indicates a better weather forecasting result. Unlike MSE, CSI can better respond to high-intensity precipitation values.

Measures of STMC. To quantify the effect of spatiotemporal mode collapse (STMC), we measure the feature clustering results as follows:

- We first perform DBSCAN [5], a typical density-based clustering method, to determine clusters of test samples based on their latent features.
- We then calculate the *silhouette coefficient* (SC) to measure the goodness of the purity of the disentangled modes. Its value ranges from -1 to 1. A higher SC value indicates that clusters are well apart and clearly distinguished, while lower SC values indicate more severe STMC issues.

We also use the *gradient difference loss* (GDL) [26] to measure the blurry effect of the predicted frames. A lower GDL indicates that the sharpness of the generated image is close to that of the true image. In Section IV-B2, we evaluate the correlations between SC and GDL, which can reveal the relations between STMC and ambiguous prediction results.

B. Results on the RoboNet Dataset

1) Main Results: Action-free video prediction. In Table I, we show the per-frame quantitative results and computational efficiency for action-free video prediction. As we can see, ModeRNN achieves *state-of-the-art* overall performance with *fewer parameters* compared with existing approaches. It can consistently benefit from training with complex visual dynamics in the entire dataset (see the bar chart in Fig. 1). Further, as shown in the first example in Fig. 4, ModeRNN is the only method that captures the exact movement of the robot arm, while other models make collapsed predictions.

TABLE I
QUANTITATIVE RESULTS ON THE ROBONET DATASET FOR ACTION-FREE VIDEO PREDICTION

Model	SSIM (↑)	MSE (↓)	Param (MB)	Mem (GB)
ConvLSTM [12]	0.725	133.4	8.2	**2.6**
PredRNN [16]	0.773	119.5	11.8	3.5
E3D-LSTM [19]	0.793	108.7	20.4	8.9
SA-ConvLSTM [23]	0.753	116.5	10.5	3.4
PhyDNet [6]	0.742	122.5	14.4	4.5
PhyDNet [6] w/ environment label	0.750	116.9	14.4	4.5
RIM [8]	0.756	120.3	9.2	3.8
LMC [24]	0.783	113.4	12.4	5.9
CrevNet [21] w/ ST-LSTM	0.794	109.4	7.0	3.3
STMFANet [25]	0.793	107.4	-	-
SimVP [45]	0.804	101.9	-	-
SVG [31]	0.792	108.2	15.2	6.5
Hierarchical VRNN [32]	0.801	103.1	-	-
SRVP [38]	0.803	102.5	-	-
ModeRNN	**0.831**	**91.9**	**6.4**	3.2

We report the average results of multiple prediction samples for the stochastic models (Rows 2-4 from the bottom).

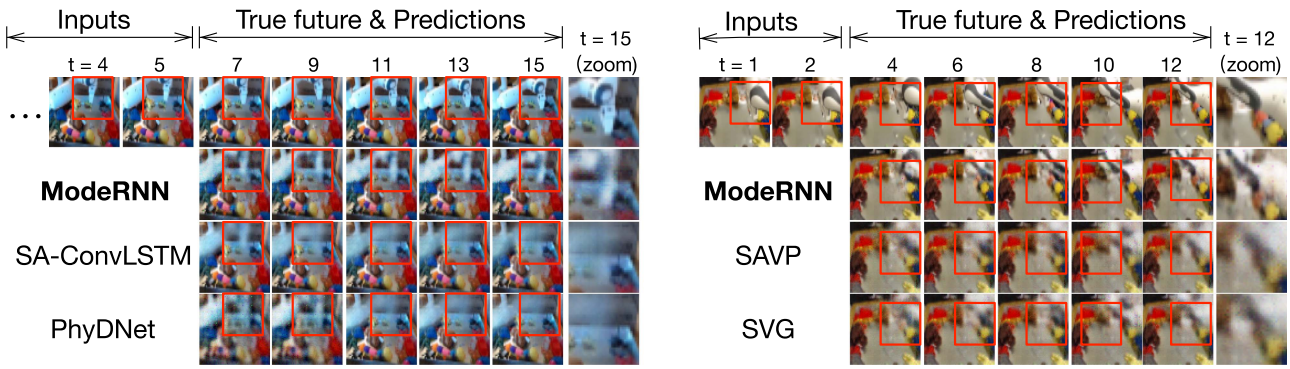
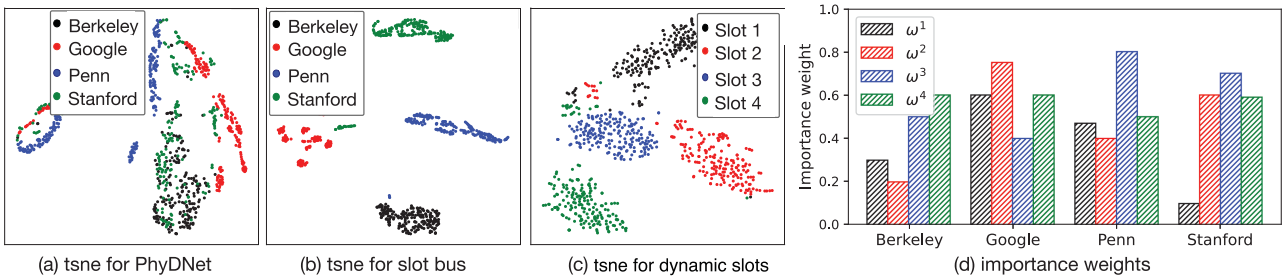


Fig. 4. Showcases of future predictions on RoboNet under the (**Left**) action-free and (**Right**) action-conditioned setups. These examples are randomly sampled from the **Stanford** environment. The red boxes highlight the positions of the robot arm. ModeRNN generates more accurate and sharper predictions about future moving trajectories of the robot arm, while SA-ConvLSTM [23], PhyDNet [6], SAVP [37], and SVG [31] suffer from severe blurry effects due to the collapse of learned dynamics.

Action-Conditioned Video Prediction. We also conduct experiments under the action-conditioned setup, encoding the inputs of robot action signals using the action fusion module from PredRNN-V2 [50]. We mainly compare the performance of ModeRNN with that of SVG [31], SRVP [38] and SAVP [37], which are strong baselines as their effectiveness on RoboNet has been well validated in the prior literature. For these models, we draw 100 prediction samples from the prior distribution given a testing sequence and report the results with the best SSIM scores. From Table II, ModeRNN achieves the best performance in *the shortest training time*. We show the qualitative results in the second case in Fig. 4. With the help of the action inputs, ModeRNN has more accurate predictions about the moving trajectories of the robot arm, while the compared models still suffer from motion collapse. Table II also gives the inference time per video sequence of the compared models, where ModeRNN is the second best in runtime efficiency. It is important to note that our approach achieves competitive results in both efficiency and prediction quality; It does not sacrifice one for the other. Unlike existing models, which mostly use *deep* image encoders and decoders (*e.g.*, SVG and SRVP employ a VGG16-based encoder

TABLE II
RESULTS ON ROBONET FOR ACTION-CONDITIONED VIDEO PREDICTION. "TRAIN": THE TOTAL TRAINING TIME IN HOURS. "TEST": THE INFERENCE TIME PER SEQUENCE

Model	SSIM (↑)	MSE (↓)	Train (h)	Test (ms)
PhyDNet [6]	0.813	106.2	20	23.9
SVG [31]	0.835	99.1	23	**18.7**
SAVP [37]	0.842	96.5	25	22.3
SRVP [38]	0.849	91.3	25	24.2
ModeRNN	**0.874**	**83.5**	**16**	19.6



(a) tsne for PhyDNet (b) tsne for slot bus (c) tsne for dynamic slots (d) importance weights

Fig. 5. (a) Demonstration of STMC on RoboNet using PhyDNet, an existing model for disentangling visual dynamics [6]. (b) The slot bus in ModeRNN learns distinct representations for samples from different environments. (c) The four slots in ModeRNN learn decoupled feature subspaces. (d) The importance weights of dynamic slots respond differently to different data environments in RoboNet, *i.e.*, families of similar video sequences.

and decoder architecture on the robot and KTH datasets), ModeRNN uses most of its parameters in the recurrent units.

2) Demonstration of Spatiotemporal Mode Collapse: STMC occurs on large real-world datasets. RoboNet naturally has the label of the data collection environments, including *Berkeley*, *Google*, *Penn*, and *Stanford*. To demonstrate that STMC does exist in real-world datasets, and our approach can overcome STMC, we assume that different environments correspond to different combinations of the spatiotemporal modes. From Fig. 1, training the existing models using data samples from all environments leads to ambiguous predictions of future dynamics; While training separate models on the subset of each environment leads to better overall performance. From these results, we may conclude that previous methods degenerate drastically when using all training samples with mixed visual dynamics. These results perfectly match the t-SNE [51] visualization in Fig. 5(a), where the cell output states of PhyDNet [6] are severely entangled and collapse to less discriminative subspaces. In contrast, from Fig. 5(b), the compositional slot bus features in ModeRNN show 4 clusters with clear boundaries, corresponding to four robot environments. For the configurations of t-SNE, we set the number of components to 2 and the perplexity to 30. The learning rate for t-SNE is 200, the maximum iteration is 250, and we use the euclidean distance as the metric. These parameters are applied for all t-SNE visualization experiments.

STMC Exists in Supervised Predictive Learning. One may concern that why not use the environment labels as input to help learn the environment-specific representations? There are two reasons. First, in reality, most spatiotemporal modes are implicit and cannot be pre-defined or annotated, even in RoboNet. Therefore, simply using the sparse labels for the environments or robot types cannot completely address the STMC problem. Second, from the 6-th line in Table I, using environment labels does not empirically significantly improve prediction results. We here follow the well-established practice in Conditional-GAN [52] to encode a one-hot environment label to PhyDNet [6].

Quantitative Analyses of STMC. As mentioned above, for each compared model, we quantify the effect of STMC by applying DBSCAN [5] to the latent features z of all test samples and then use *silhouette coefficient* to calculate the goodness of the clustering results. For each video sample i in the test set, we take the average slot bus features in ModeRNN over T prediction time horizon as $z^{(i)} = \frac{1}{T}\sum_{t=\tau+1}^{\tau+T} \mathcal{B}_t^{(i)}$, where τ is the number of input frames. Similarly, for other RNN-based predictive models, we use the average hidden state over the prediction time horizon. We then calculate the average SC value of all test samples to measure the disentanglement of multiple modes, which is defined as:

$$SC = \frac{b-a}{\max(a,b)}, \tag{8}$$

where a is the cluster cohesion, which refers to the average distance between a sample $z^{(i)}$ and all other data points within the same cluster; b represents the cluster separation, which refers to the average distance between $z^{(i)}$ and all other data points in the nearest cluster. That is to say when the average SC value is close to 1, the learned features are far away from the neighboring clusters, indicating clearly disentangled spatiotemporal modes. As shown in Table III, ModeRNN achieves the highest SC scores on the complex RoboNet dataset and significantly outperforms deterministic video prediction models that are designed for disentanglement learning, including RIM [8] and PhyDNet [6]. It also outperforms the state-of-the-art probabilistic model (*i.e.*, SRVP [38]) in mitigating STMC.

TABLE III
EVALUATION OF STMC ON ROBONET FOR ACTION-FREE VIDEO PREDICTION. "SC": THE AVERAGE SILHOUETTE COEFFICIENT OF ALL DATA. "GDL": THE GRADIENT DIFFERENCE LOSS THAT MEASURES THE BLURRY EFFECT OF THE GENERATED IMAGES

Model	SC (↑)	GDL (↓)
PhyDNet [6]	0.372	19.5
RIM [8]	0.348	20.1
E3D-LSTM [19]	0.432	18.3
SRVP [38]	0.537	16.7
ModeRNN	**0.659**	**13.4**

Relationships Between STMC and Ambiguous Predictions. Table III also presents the gradient difference loss to show the blurry effect of the generated future images. We can easily find the correlations between the SC and GDL scores, which indicates that mitigating STMC can keep the model from making ambiguous predictions. Besides, ModeRNN achieves the best results in the GDL scores.

TABLE IV
ABLATION STUDY OF EACH MODEL COMPONENT. EXPERIMENTS ARE CONDUCTED ON THE ACTION-FREE ROBONET. "MHA": MULTI-HEAD ATTENTION. "SC": THE AVERAGE SILHOUETTE COEFFICIENT OF ALL DATA

Model	MSE (↓)	SC (↑)
ModeRNN	**91.9**	**0.659**
ModeRNN w/o slot binding	132.5	0.347
ModeRNN w/o slot fusion	121.2	0.425
ModeRNN w/o per-slot FFN	110.7	0.483
ModeRNN w/o gated shortcut	128.3	0.379

3) Feature Visualization: Besides the visualization of slot bus in Fig. 5(b), we testify the mode decoupling ability of ModeRNN by visualizing the slot features in Fig. 5(c) We can see that features of the 4 dynamic slots are clustered into 4 groups, indicating the ability to disentangle various spatiotemporal modes. In Fig. 5(d), we use the averaged importance weights $\{\omega_t^n\}_{n=1}^4$ on each slot to analyze how the adaptive slot fusion module works. We can see that different robot environments lead to different significance over dynamic slots.

4) Ablation Study: Effectiveness of Each Model Component. In Table IV, we analyze the efficacy of each component in ModeRNN on the action-free RoboNet dataset in terms of prediction quality and feature disentanglement results (*i.e.*, silhouette coefficient values). We have the following observations. First, removing the slot binding module increases the prediction error by 44.2%, showing the necessity of learning to decouple the dynamics using separate dynamic slots. It also leads to a significant decrease of the SC value ($0.659 \rightarrow 0.347$), indicating that the slot binding module is effective for disentanglement learning of the spatiotemporal modes. Second, removing the adaptive slot fusion module increases the prediction error by 31.9%, which strongly demonstrates that it is crucial to learn the state transitions upon the compositional representations based on the slot features. Third, the per-slot FFN in the slot binding module and the gated shortcut ($\omega_t^0 \cdot \mathcal{I}_t$) in the slot fusion module also show a significant impact on the final performance, which verifies that parameter isolation is effective in mode decoupling. It reveals the positive effect of an adaptive fusion of rich appearance information and compact spatiotemporal dynamics. Finally, the entire decoupling-aggregation framework that integrates the above techniques in a unified modular model performs best in both prediction results in MSE and the feature disentanglement results in SC values.

Number of Dynamic Slots. The number of dynamic slots for different datasets is determined by the data complexity or our prior knowledge. On the RoboNet dataset, particularly, we observe that ModeRNN with 4–9 dynamic slots achieve competitive performance, as shown in Table V, and we finally set $N = 4$ for simplicity. Notably, using a single slot in each ModeCell achieves similar performance to SA-ConvLSTM [23] (Table I), which incorporates self-attention in the recurrent state transitions but does not have a mode decoupling framework.

Scaling-up the Model Size. We further study increasing the mode scale of ModeRNN. We separately experiment with an increased number of dynamic slots (as shown in the right column in Table V) and the increase of the feature channels for each slot (as shown in Fig. 6). The total number of model parameters grows in both cases. Experiments are still conducted on the action-free RoboNet dataset. Specifically, we fix the number of channels for each slot feature and gradually increase the number of slots N. From the results in the right part of Table V, we can see that ModeRNN achieves higher performance when N increases. Besides, in Fig. 6, we fix the number of dynamic slots $N = 4$, gradually increasing the hidden state's channel number d_s from 64 to 512. We find that, in the range of $d_s \in [64, 256]$, ModeRNN improves as d_s increases; While the performance of ModeRNN tends to be stable, when we continue to increase d_s from 256 to 512. All in all, ModeRNN achieves higher performance by increasing the model scale.

TABLE V
ABLATION STUDY ON THE ACTION-FREE ROBONET FOR THE NUMBER OF DYNAMIC SLOTS IN MODECELL. NOTE THAT ON THE LEFT SIDE, THE TOTAL NUMBER OF CHANNELS FOR THE SLOT BUS FEATURES (d_s) REMAINS ROUGHLY THE SAME; WHILE ON THE RIGHT SIDE, WE USE A FIXED NUMBER OF FEATURE CHANNELS 16 FOR EACH DYNAMIC SLOT, SO THAT THE TOTAL MODEL SIZE GROWS WITH THE INCREASE OF N

# Dynamic slots	MSE (↓)	# Dynamic slots	MSE (↓)
$N = 1$	118.1	$N = 1$	129.3
$N = 2$	103.4	$N = 2$	115.7
$N = 3$	94.5	$N = 3$	101.8
$N = 4$ **(Final)**	91.9	$N = 4$	91.9
$N = 5$	91.7	$N = 5$	90.2
$N = 6$	91.3	$N = 6$	89.1
$N = 7$	**90.8**	$N = 7$	88.3
$N = 8$	91.0	$N = 8$	**88.0**
$N = 9$	90.9	$N = 9$	88.2

TABLE VI
ABLATION STUDY ON THE ACTION-FREE ROBONET DATASET FOR THE USE OF MULTI-HEAD ATTENTION. "MHA" IS SHORT FOR *MULTI-HEAD ATTENTION*

Method	MSE (↓)	SC (↑)
ModeRNN w/o MHA	113.7	0.431
ModeRNN w/ GCN	107.4	0.493
ModeRNN w/ MHA (Final)	**91.9**	**0.659**

TABLE VII
INVESTIGATION OF LARGER MODELS FOR THE EXISTING APPROACHES ON THE ACTION-FREE ROBONET DATASET BY INCREASING THE CHANNEL NUMBER OF RECURRENT STATES FROM 64 TO 512

Model	# Channel	MSE (↓)	SC (↑)
PhyDNet	64	122.5	0.372
PhyDNet	512	120.3	0.371
RIM	64	120.3	0.348
RIM	512	118.1	0.345
ModeRNN	64	**91.9**	**0.659**

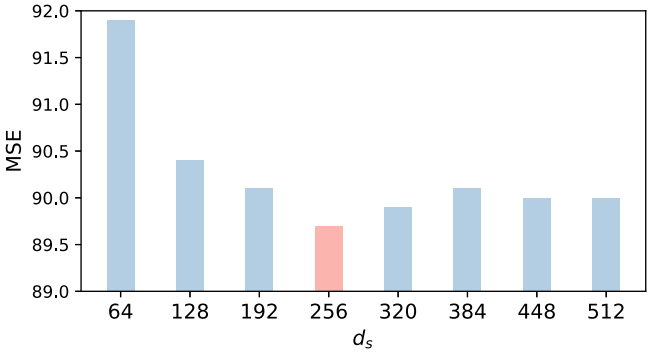


Fig. 6. The parameter analysis on the action-free RoboNet for the channel number d_s of the hidden state in ModeCell. We use a fixed number of dynamic slots $N = 4$. Note that in the final ModeRNN, we set the default value of d_s to 64 for higher training and testing efficiency.

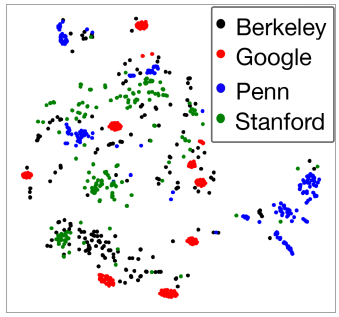


Fig. 7. The t-SNE results of the hidden states in SA-ConvLSTM on RoboNet, which can be compared with that of ModeRNN in Fig. 5(b).

Scaling-Up the Compared Models. To investigate whether existing models can also alleviate the problem of spatiotemporal mode collapse with larger model size, we train larger PhyDNet [6] and RIM [8] models by increasing the channel number of their recurrent states from 64 to 512. As shown in Table VII, the scaled-up models produce slightly better prediction results (in MSE), but their disentanglement ability (in SC scores) remains largely unchanged. These findings suggest that increasing the model size may not be an effective solution for the spatiotemporal mode collapse issue.

The Efficacy of Multi-Head Attention. The multi-head attention (MHA) is originally used in Transformer to learn diverse representation via the divided multiple heads. In ModeRNN, we borrow the MHA mechanism to decouple the hidden states into different subspaces of spatiotemporal dynamic slots. As shown in Table VI, we validate the effectiveness of such a design on RoboNet under the action-free setup. Specifically, we first remove MHA and find that ModeRNN deteriorates for prediction quality and disentanglement performance (SC: $0.659 \rightarrow 0.431$). Furthermore, we replace MHA with GCN [53], another typical architecture for disentanglement learning treating each slot as a node. We find that it performs much worse the final ModeRNN with MHA. These results show that multi-head attention is an effective way to factorize spatiotemporal features to overcome STMC.

5) Comparison With SA-ConvLSTM: To better position our ModeRNN, we provide a further comparison with the competitive baseline SA-ConvLSTM [23] as follows, which combines the self-attention and ConvLSTM to capture the global context information. We demonstrate STMC with extensive visualization and further propose the ModeRNN with full insights to tackle the STMC problem. Technically, there are two core differences between ModeRNN and SA-ConvLSTM.

First, SA-ConvLSTM uses self-attention only for representation aggregation, leading to an inherent lack of the ability to decouple the mixed visual dynamics into several modes. On the contrary, ModeRNN separates the learned representations in several subspaces as *dynamic slots* and adopts the *slot bus* to connect the decoupled slots along the temporal dimension. This design is highly motivated by the observation of STMC. As shown in Fig. 5, the learned dynamic slots from ModeRNN are clustered into 4 distinct groups. This observation indicates that ModeRNN has the capability to disentangle different spatiotemporal modes and effectively manage mixed dynamics.

Second, SA-ConvLSTM could not dynamically capture the mixed visual dynamics and adjust to different environments effectively. It only uses the self-attention between recurrent states to capture the global context regardless of various spatiotemporal mode information across different environments. In contrast, ModeRNN develops a modular structure, which adaptively produces compositional features via the *dynamic slots* and *adaptive slot fusion*.

As shown in Fig. 7, we conduct the t-SNE visualization on RoboNet, where the memory states of SA-ConvLSTM are entangled and collapse to the ambiguous representation subspaces, leading to the severe STMC. The quantitative results also show that SA-ConvLSTM does not behave well compared with ModeRNN (SSIM: 0.753versus 0.831, MSE: 116.5versus 91.9) on the complicated real-world dataset.

All in all, ModeRNN is different from SA-ConvLSTM in both motivation and technical design, which is compared distinctly from the combination of the multi-head attention and ConvLSTM. The carefully designed *dynamic slots*, *slot bus* and *adaptive slot fusion* form a decoupling-aggregation framework, directly aiming at the STMC, which is the key problem of unsupervised predictive learning. Benefiting from this compact connection between the STMC and model design, ModeRNN achieves the state-of-the-art performance and explainable slot features on extensive datasets with complex spatiotemporal modes.

C. Results on the KTH Dataset

The KTH dataset [44] contains 6 action categories and involves 25 subjects in 4 different scenarios. It thus naturally contains various modes responding to similar motion dynamics. We use person 1-16 for training and 17-25 for testing, resize each frame to the resolution of 128×128, and predict 20 frames from 10 observations.

Main Results. On this dataset, we use the frame-wise peak signal-to-noise ratio (PSNR) and Structural Similarity (SSIM) [48] as evaluation metrics following previous literature [1]. We use 6 dynamic slots in each ModeCell. In Table VIII, we show the quantitative results and find that ModeRNN

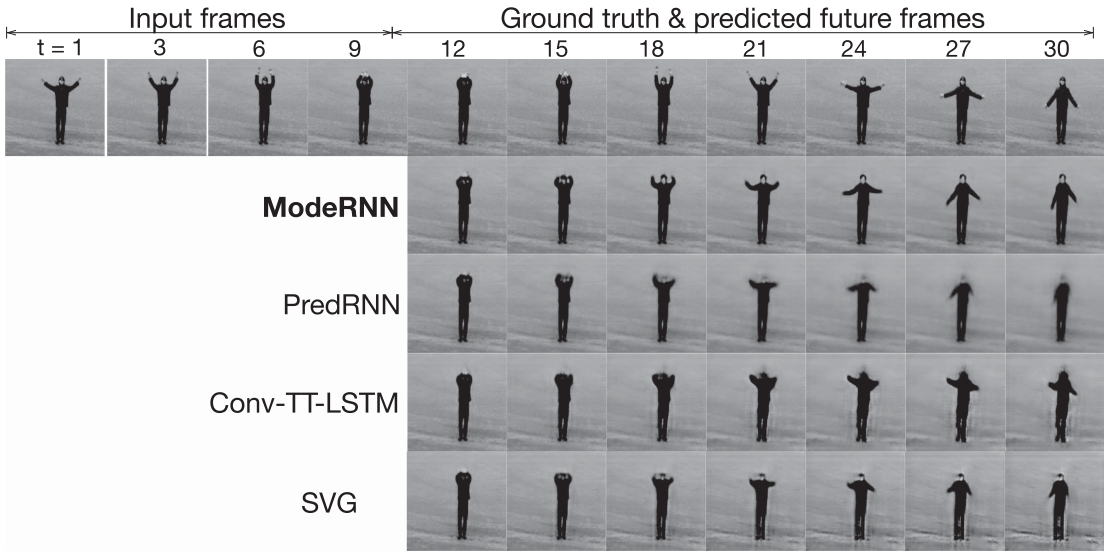


Fig. 8. Examples of predicted future frames on the KTH action dataset.

TABLE VIII
COMPARISONS WITH EXISTING PROBABILISTIC AND DETERMINISTIC VIDEO PREDICTION MODELS ON THE KTH DATASET

Model	PSNR (↑)	SSIM (↑)
SVG [31]	27.73	0.863
SRVP [38]	28.41	0.873
ConvLSTM [12]	24.12	0.712
TrajGRU [54]	*26.97	*0.814
PredRNN [16]	*27.47	*0.839
SA-ConvLSTM [23]	*29.33	0.897
E3D-LSTM [19]	*29.31	*0.879
PhyDNet [6]	28.69	0.879
CrevNet [21]	28.82	0.883
LMC [24]	*28.61	*0.894
RIM [8]	27.01	0.817
SimVP [45]	29.38	0.898
ModeRNN	**29.45**	**0.906**

For SVG, we report the best results from 100 output samples per input sequence. *Indicates the result directly copied from the original references.

performs best among all compared methods, including the state of the art proposed in recent two years [6], [21], [23], [24]. We provide the qualitative comparisons in Fig. 8, where we observe that ModeRNN can predict the precise position of the moving person.

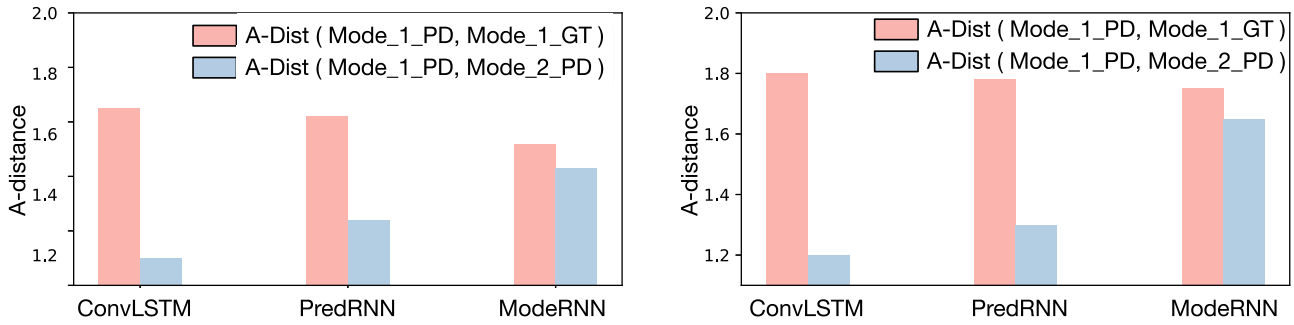
A-Distance. A-distance [55] is defined as $d_A = 2(1 - 2\epsilon)$ where ϵ is the error rate of a domain classifier trained to discriminate two visual domains. In Fig. 9, we use the A-distance to quantify the STMC in the real-world KTH action dataset. In this experiment, we divide the KTH dataset into two groups according to the visual similarities of human actions. According to the scale of the actions, we can simply group the existing six categories in the KTH dataset into two typical groups:

- The first group corresponds to the global movement of the torso, including running, walking, and jogging.
- The second group corresponds to the local movement of hands, including the categories of hand-clapping, hand-waving, and boxing.

We here use the memory state $\mathcal{C}_t$ in ConvLSTM and PredRNN, and the slot bus $\mathcal{B}_t$ in ModeRNN to calculate A-distance. As shown by the blue bars (higher is better), the lower A-distance between the two groups indicates that the learned representations from the two groups are highly entangled. The red bars (lower is better) show the domain distance between features taking as inputs the ground truth frames $\mathcal{X}_t$ and those taking the predictions $\widehat{\mathcal{X}}_t$. STMC happens when the A-distance between predictions of different groups (in blue) becomes much smaller than that between predictions and ground truth (in red).

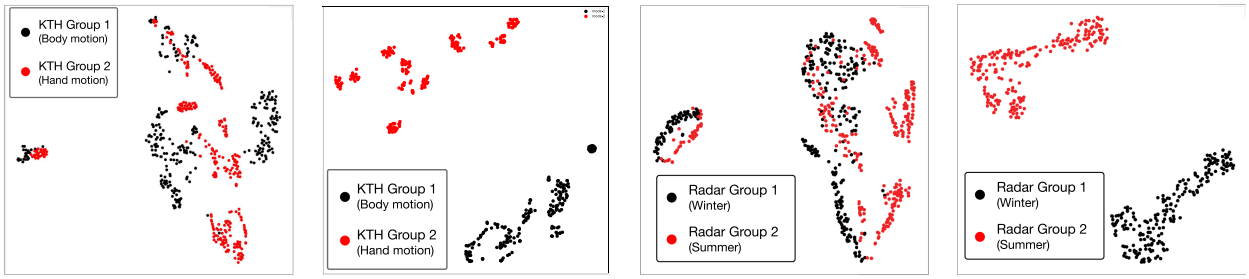
t-SNE. As shown in Fig. 10(a), we visualize the memory state of ConvLSTM using t-SNE [51]. It is observed that the learned cell states by ConvLSTM are entangled among different action groups. The t-SNE visualization result matches the PhyDNet visualization on the RoboNet dataset shown in Fig. 5(a). Thus, these results verify that the STMC also exists under the real-world human motion dataset. While in ModeRNN, we further visualize the learned features of the slot bus in Fig. 10(b), which shows 2 clusters with clear boundaries, corresponding to two action groups in the KTH dataset. According to these t-SNE results, we can find that directly training the previous methods on the mixed dynamics will lead to severe STMC in representation learning, shown as the entanglement of hidden representations. These entangled representations will make the model provide a poor ambiguous prediction. In contrast, ModeRNN can effectively overcome the STMC by learning an accurate decoupling for mixed dynamics.

Number of Dynamic Slots. In our definition, the dynamic slots correspond to feature subspaces that collectively constitute spatiotemporal modes, which can be thought of as data clusters with similar global representations. As stated earlier, the number of



(a) A-distance of memory states (b) A-distance of output states

Fig. 9. A-distance of the memory states ($\mathcal{B}_t$ for ModeRNN) and the output states ($\mathcal{H}_t$ for ModeRNN) on KTH.



(a) ConvLSTM cell state, KTH (b) ModeRNN slot bus, KTH (c) ConvLSTM cell state, Radar (d) ModeRNN slot bus, Radar

Fig. 10. (a, c) Illustration of STMC on the existing ConvLSTM model on KTH and radar echo dataset of Guangzhou (GZ). (b, d) The slot bus of ModeRNN shows discriminative representations on different groups of video dynamics. The two groups in KTH respectively correspond to subtle hand motion (*e.g.*, hand-waving, hand-clapping, and boxing) and more global body motion (*e.g.*, running, walking, and jogging). The two groups in Radar are divided by different seasons.

TABLE IX
ANALYSIS OF THE *NUMBER OF DYNAMIC SLOTS* (N) ON THE KTH DATASET, SIMILAR TO THE RESULTS PRESENTED IN TABLE V. WE KEEP THE MODEL SIZE APPROXIMATELY CONSTANT DESPITE VARYING VALUES OF N, AND WE STOP THE GRID SEARCH AT $N = 9$ WHEN THERE IS NO FURTHER IMPROVEMENT IN THE RESULTS

N	1	2	4	6 **(Final)**	8	9
PSNR	28.35	28.87	29.26	29.45	**29.52**	29.50

TABLE X
QUANTITATIVE RESULTS ON THE RADAR ECHO DATASET

Model	CSI-30 (↑)	MSE (↓)
SA-ConvLSTM [23]	0.362	86.1
TrajGRU [54]	0.357	89.2
PredRNN [16]	0.359	84.2
PhyDNet [6]	0.358	92.1
CrevNet [21]	0.381	81.5
LMC [24]	0.361	92.5
ModeRNN	**0.428**	**65.1**

dynamic slots used in a dataset is determined by the complexity and diversity of the data patterns. In the case of the KTH dataset, we examine the impact of varying the number of slots (N) in ModeRNN, while keeping the total model size approximately constant, as shown in Table IX. ModeRNN outperforms the previous state-of-the-art (which achieved a PSNR of 29.38) with N values ranging from 6 to 9 and achieves the best performance at $N = 8$. We empirically use $N = 6$ in our experiments for simplicity. In other words, the value of $N = 6$ is determined through a grid search and is NOT simply determined by the number of action categories.

D. Results on the Radar Echo Dataset

Main Results. As shown in Table X, ModeRNN achieves state-of-the-art overall performance and significantly outperforms the competitive precipitation method, TrajGRU [54] (CSI: **0.428** vs 0.357, MSE: **65.1** vs 89.2). We also show examples of predicted future frames on the radar echo dataset in Fig. 11. We find that the compared models fail in predicting the edges of the cyclone, and the predicted movement of the cloud even vanishes. On the contrary, ModeRNN provides more details about the cyclone and accurately predicts its center position indicated by the red box. Both the quantitative and qualitative results show that our ModeRNN can effectively capture the dynamic information from complex meteorological dynamic mode.

t-SNE. Considering the climate change among different seasons in Guangzhou, we can roughly consider the radar echo dataset into two typical meteorology groups:

- The first group: It corresponds to the windier part of the year, from March to May, with average wind speeds of

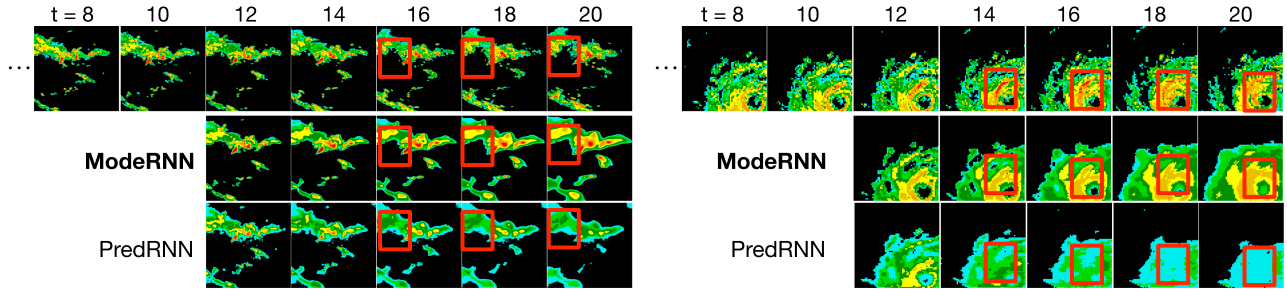


Fig. 11. Examples of predicted future frames on the radar echo dataset. ModeRNN produces more accurate predictions indicated by the red boxes.

TABLE XI
QUANTITATIVE RESULTS ON THE HUMAN3.6M DATASET. FOR MOTIONRNN [41], WE USE MIM [18] AS THE NETWORK BACKBONE

Model	PSNR (↑)	FVD (↓)	LPIPS (↓)
SA-ConvLSTM [23]	21.3	19.2	0.153
E3D-LSTM [19]	19.7	23.7	0.173
SRVP [38]	22.5	18.1	0.137
PhyDNet [6]	22.0	18.3	0.145
MotionRNN [41]	22.1	18.3	0.136
LMC [24]	21.5	18.7	0.151
CrevNet [21]	22.6	18.1	0.139
RIM [8]	20.1	21.3	0.168
SimVP [45]	23.0	17.3	0.134
ModeRNN	**24.2**	**16.4**	**0.123**

more than 7.5 miles per hour. There will be drizzles from time to time in these months. We use the radar maps from 2016/3 to 2016/5 and 2017/3 to 2017/4 for training, and use those in 2017/5 for testing.

- The second group: It corresponds to the summer in Guangzhou, which experiences heavier cloud cover, with the percentage of time that the sky is overcast or mostly cloudy is around 80%. We use the radar maps from 2016/6 to 2016/8 and 2017/6 to 2017/7 for training, and use those in 2017/8 for testing.

In Fig. 10(c), we visualize the cell state of ConvLSTM using t-SNE and find that the learned cell states are entangled under different climate groups. It shows that STMC exists under the real-world precipitation dataset. We further visualize the slot bus features in Fig. 10(d), which show 2 clusters with clear boundaries, corresponding to two climate groups.

E. Results on the Human3.6M Dataset

Main Results. As shown in Table XI, ModeRNN significantly outperforms the previous state-of-the-art method MotionRNN [41] (PSNR: 24.2versus 22.1, FVD: 16.4versus 18.3, LPIPS: 0.123versus 0.136). Note that our approach can also obtain great promotion on the FVD metric, which means the prediction results are better in terms of motion consistency. As for the qualitative results in Fig. 12, ModeRNN predicts the sharpest sequence compared with other methods and enriches the details for each part of the body, especially for the arms.

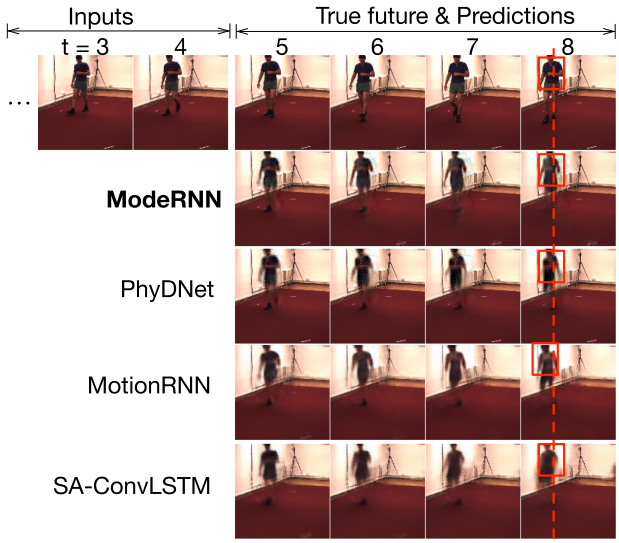


Fig. 12. Examples of predicted future frames on the Human3.6M dataset.

TABLE XII
RESULTS ON THE DETERMINISTIC MIXED MOVING MNIST DATASET

Model	SSIM (↑)	MSE (↓)
RIM [8]	0.874	57.5
E3D-LSTM [19]	0.901	47.5
SA-ConvLSTM [23]	0.880	70.2
PhyDNet [6]	0.871	73.8
LMC [24]	0.892	68.3
CrevNet [21]	0.891	51.5
DDPAE [40]	0.903	46.9
SimVP [45]	0.912	42.7
ModeRNN	**0.934**	**33.4**

These results validate the ability of ModeRNN to deal with complex spatiotemporal modes in a fully unsupervised way.

F. Results on the Mixed Moving MNIST Dataset

Main Results. In Table XII, we show the overall quantitative results as well as computational efficiency of the compared models on the Mixed Moving MNIST dataset. As we can see, ModeRNN achieves **state-of-the-art** overall performance compared with existing approaches, including the state-of-the-art

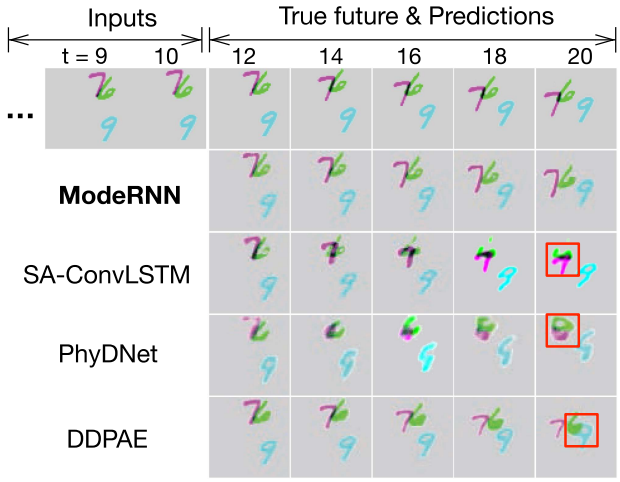


Fig. 13. Examples of predicted future frames on Mixed Moving MNIST.

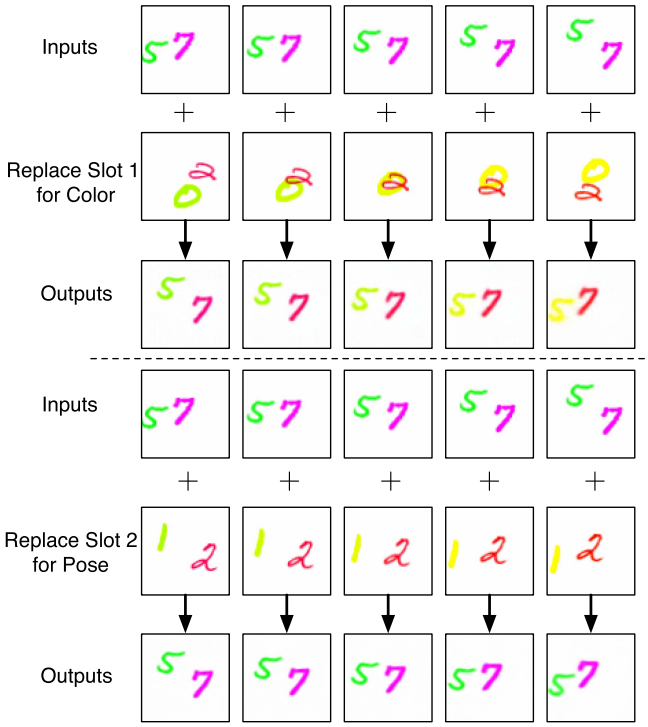


Fig. 14. *Slot swap tests* for the efficacy of modular structures in disentanglement learning.

approaches proposed in recent two years. DDPAE [40] is a strong baseline, which learns to decompose each digit and then disentangle its content and motion. It achieves good results on the synthetic dataset but can hardly generalize well to real-world data partly due to the strong inductive bias for plain background and rigid shapes of the moving objects. In contrast, ModeRNN achieves both competitive results on real and synthetic data (SSIM: **0.934** versus 0.903; MSE: **33.4** versus 46.9, compared with DDPAE). As we can see from Table XII, ModeRNN outperforms E3D-LSTM [19] and SimVP [45] remarkably.

As shown in Fig. 13, we can see that the predicted digits "7" and "6" from the compared models collapse across time. This is caused by STMC as the models are trained on the entire dataset with a variable number of flying digits. In contrast, ModeRNN is the only method that can capture the exact movement of each digit and keep its own visual mode of color. All in all, ModeRNN effectively overcomes STMC. It achieves the best performance on the synthetic data which is more difficult than the original Moving MNIST dataset [40] due to a larger variety of visual dynamics.

Further Evidence for Disentanglement. As shown in Fig. 14, we conduct the *slot swap tests*, in which we replace the features of one dynamic slot from the input sequence (1st row) with features from another sequence (2nd row) that belong to the same slot. Intuitively, there are two significant modes in this dataset: *the time-varying pose and the digit color that may also change over time.* In Fig. 14, ModeRNN has successfully learned to separate the mixed modes into multiple representation subspaces through the modular architecture, assigning different subspaces to different slots:

- Replacing features in Slot #1 makes the color change with the other sequence while maintaining the motions.
- Replacing features in Slot #2 is equivalent to borrowing the poses from the other one while maintaining the colors.

The swap tests validate the effectiveness of the proposed dynamic slots in disentangling mixed visual dynamics.

V. Conclusion

In this paper, we demonstrated a new phenomenon of spatiotemporal mode collapse (STMC) when training unsupervised predictive models on real-world datasets with highly mixed visual dynamics. Accordingly, we proposed ModeRNN that effectively learns modular features using a set of dynamic slots. To discover the compositional structures in spatiotemporal modes, ModeRNN adaptively aggregates the slot features with learnable importance weights. Compared with existing models, ModeRNN was shown to mitigate the collapse of future predictions, improving qualitative and quantitative results on five real-world or synthetic datasets.

References

[1] S. Oprea et al., "A review on deep learning techniques for video prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2806–2826, Jun. 2022.
[2] I. J. Goodfellow et al., "Generative adversarial networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[3] F. Locatello et al., "Object-centric learning with slot attention," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 11525–11538.
[4] S. Dasari et al., "RoboNet: Large-scale multi-robot learning," 2019, *arXiv: 1910.11215*.
[5] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.
[6] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 474–11 484.
[7] Z. Xu et al., "Unsupervised discovery of parts, structure, and dynamics," in *Proc. Int. Conf. Learn. Representations*, 2019.
[8] A. Goyal et al., "Recurrent independent mechanisms," in *Proc. Int. Conf. Learn. Representations*, 2021.
[9] A. Vaswani et al., "Attention is all you need," 2017, *arXiv: 1706.03762*.

[10] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: A baseline for generative models of natural videos," 2014, *arXiv:1412.6604*.
[11] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 843–852.
[12] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
[13] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in Atari games," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2863–2871.
[14] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 667–675.
[15] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3560–3569.
[16] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 879–888.
[17] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 716–731.
[18] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9154–9162.
[19] 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 *Proc. Int. Conf. Learn. Representations*, 2019.
[20] Z. Yao, Y. Wang, M. Long, and J. Wang, "Unsupervised transfer learning for spatiotemporal predictive networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10 778–10 788.
[21] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Efficient and information-preserving future frame prediction and beyond," in *Proc. Int. Conf. Learn. Representations*, 2019.
[22] J. Su, W. Byeon, F. Huang, J. Kautz, and A. Anandkumar, "Convolutional tensor-train LSTM for spatio-temporal learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 13714–13726.
[23] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention ConvLSTM for spatiotemporal prediction," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2020, pp. 11531–11538.
[24] S. Lee, H. G. Kim, D. H. Choi, H.-I. Kim, and Y. M. Ro, "Video prediction recalling long-term motion context via memory alignment learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3054–3063.
[25] B. Jin et al., "Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4554–4563.
[26] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. Int. Conf. Learn. Representations*, 2016.
[27] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 613–621.
[28] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1526–1535.
[29] J. Xu, B. Ni, and X. Yang, "Video prediction via selective sampling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1712–1722.
[30] T.-C. Wang et al., "Video-to-video synthesis," 2018, *arXiv: 1808.06601*.
[31] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1182–1191.
[32] L. Castrejon, N. Ballas, and A. Courville, "Improved conditional VRNNs for video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7608–7617.
[33] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1811–1820.
[34] S. Bhagat, S. Uppal, Z. Yin, and N. Lim, "Disentangling multiple features in video sequences using gaussian processes in variational autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 102–117.
[35] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall, "Probabilistic future prediction for video scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 767–785.
[36] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *Proc. Int. Conf. Learn. Representations*, 2018.
[37] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," 2018, *arXiv: 1804.01523*.
[38] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, "Stochastic latent residual video prediction," 2020, *arXiv: 2002.09219*.
[39] E. L. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4417–4426.
[40] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Learning to decompose and disentangle representations for video prediction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 515–524.
[41] H. Wu, Z. Yao, J. Wang, and M. Long, "MotionRNN: A flexible model for video prediction with spacetime-varying motions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 435–15 444.
[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[43] N. Shazeer et al., "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv: 1701.06538*.
[44] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. Int. Conf. Pattern Recognit.*, 2004, pp. 32–36.
[45] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "SimVP: Simpler yet better video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3170–3180.
[46] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[47] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[49] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," 2018, *arXiv: 1812.01717*.
[50] Y. Wang et al., "PredRNN: A recurrent neural network for spatiotemporal predictive learning," 2021, *arXiv:2103.09504*.
[51] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[52] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[53] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[54] X. Shi et al., "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5617–5627.
[55] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, pp. 151–175, 2010.

Zhiyu Yao received the BE degree in computer software from Tsinghua University, China, in 2019. He is working towards the PhD degree in computer software with Tsinghua University. His research interests include machine learning and computer vision.

Yunbo Wang received the BE degree from Xi'an Jiaotong University in 2012, and the ME and PhD degrees from Tsinghua University in 2015 and 2020. He received the CCF Outstanding Doctoral Dissertation Award in 2020, advised by Philip S. Yu and Mingsheng Long. He is now an assistant professor with the AI Institute, Shanghai Jiao Tong University. He does research in deep learning, especially predictive learning, model-based reinforcement learning, and intuitive physics.

Haixu Wu received the BE degree in computer software from Tsinghua University in 2020. He is working towards the PhD degree in computer software with Tsinghua University. His research interests include machine learning and computer vision.

Jianmin Wang received the BE degree from Peking University, China, in 1990, and the ME and PhD degrees in computer software from Tsinghua University, China, in 1992 and 1995, respectively. He is a full professor with the School of Software, Tsinghua University. His research interests include Big Data management systems and large-scale data analytics. He led to developing a product data and lifecycle management system, which has been deployed in hundreds of enterprises in China. He is leading the development of the Tsinghua DataWay Big Data platform in the National Engineering Lab for Big Data Software.

Mingsheng Long (Member, IEEE) received the BE and PhD degrees from Tsinghua University in 2008 and 2014 respectively. He was a visiting researcher with UC Berkeley from 2014 to 2015. He is currently a tenured associate professor with the School of Software, Tsinghua University. He serves as an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Artificial Intelligence Journal*, and as area chairs of major machine learning conferences, including ICML, NeurIPS, and ICLR. His research is dedicated to machine learning theory, algorithms, and models, with special interests in transfer learning and domain adaptation, deep learning and foundation models, scientific learning, and world models.