Unsupervised Transfer Learning for Spatiotemporal Predictive Networks

Zhiyu Yao [*1] **Yunbo Wang** [*1] **Mingsheng Long** [1] **Jianmin Wang** [1]

Abstract

This paper explores a new research problem of unsupervised transfer learning across multiple spatiotemporal prediction tasks. Unlike most existing transfer learning methods that focus on fixing the discrepancy between supervised tasks, we study how to transfer knowledge from a zoo of unsupervisedly learned models towards another predictive network. Our motivation is that models from different sources are expected to understand the complex spatiotemporal dynamics from different perspectives, thereby effectively supplementing the new task, even if the task has sufficient training samples. Technically, we propose a differentiable framework named *transferable memory*. It adaptively distills knowledge from a bank of memory states of multiple pretrained RNNs, and applies it to the target network via a novel recurrent structure called the *Transferable Memory Unit* (TMU). Compared with finetuning, our approach yields significant improvements on three benchmarks for spatiotemporal prediction, and benefits the target task even from less relevant pretext ones.

1. Introduction

Existing transfer learning methods mainly focus on how to fix the discrepancy between supervised tasks. However, unsupervised learning has achieved remarkable advances in recent years and has become a hot topic in the deep learning community. Then new questions arise: *Is it necessary to do transfer learning between unsupervised tasks, and how to do it?*

As a typical unsupervised learning paradigm, predictive learning has shown great research significance in discovering the underlying structure of unlabeled spatiotemporal data without human supervision and learning generalizable deep representations from the consequences of complex video events. The studies of the spatiotemporal predictive learning can benefit many practical applications and downstream tasks, such as precipitation nowcasting (Shi et al., 2015), traffic flow prediction (Xu et al., 2018), physical scene understanding (Wu et al., 2017), early activity recognition (Wang et al., 2019a), deep reinforcement learning (Ha & Schmidhuber, 2018), and vision-based model predictive control (Finn & Levine, 2017). Different from all the above work, in this paper, we explore how to transfer knowledge from a zoo of pretrained models towards a novel predictive learning task. Models from both the source and target domains are trained to predict sequences of future frames.

Transferring knowledge across these tasks is yet to be explored, but important. In many scenarios, deep networks may suffer from the serious problem of long-tail data distribution in the target domain. A natural solution is to finetune another model that was well-pretrained with large-scale and more effective training data. For example, when we train precipitation forecasting models for arid areas, we may exploit the laws of weather changes that are learned from other areas with abundant rainfall. However, it is a challenging transfer learning problem, because, in the first place, not all knowledge of the pretrained models can be directly applied to the target task due to the discrepancy of various domains. After all, different areas may have their unique climate characteristics. We have to explore how to distill the transferable representations from the pretrained models without labeled data. In the second place, with a zoo of source models, we need to dynamically adjust their impact on the training process of the target network.

To solve these problems, we propose a novel differentiable framework named *transferable memory* along with the new *Transferable Memory Unit* (TMU). Different from finetuning, our approach enables the target model to adaptively learn from a zoo of source models. It provides diverse understandings of the underlying, complex data structure of the target domain. Technically, we perform unsupervised knowledge distillation on the memory states of multiple pretrained recurrent networks, and then introduce a new gating mechanism to dynamically find the transferable part of the distilled representations. In this way, we use the spatiotemporal dynamics of source domains as the prior knowledge

[*]Equal contribution [1]School of Software, BNRist, Research Center for Big Data, Tsinghua University. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

of the final predictive model, so it can focus more on the domain-specific data structure on the target dataset. The *transferable memory* framework significantly outperforms previous transfer learning methods on three benchmarks with nine sub-datasets: a synthetic flying digits benchmark, a real-world human motion benchmark, and a precipitation nowcasting benchmark.

The contributions of this paper are summarized as follows:

- We introduce a new research problem of unsupervised transfer learning across multiple spatiotemporal prediction tasks. It is challenging as the data distribution of different domains can be distant, *e.g.*, from various data sources, or from synthetic data to real data.
- We propose a deep learning solution which features the *transferable memory* and is shown effective for a wide range of RNNs, including ConvLSTM (Shi et al., 2015), PredRNN (Wang et al., 2017), MIM (Wang et al., 2019b), and SAVP (Lee et al., 2019), covering both deterministic and stochastic models.
- We validate the effectiveness of the proposed approach on three benchmarks with a variety of data sources and have a series of empirical findings. Unlike supervised transfer learning where irrelevant source data may lead to negative transfer learning effects, the proposed approach can adaptively transfer temporal dynamics from source videos even if the content seems less relevant.

2. Related Work

2.1. Spatiotemporal Prediction

Due to the modeling capability of temporal dependencies, the early literature suggested using RNN-based models for spatiotemporal predictive learning (Ranzato et al., 2014; Srivastava et al., 2015; Oh et al., 2015; De Brabandere et al., 2016). Shi et al. (2015) proposed the convolutional LSTM (ConvLSTM) that combines the advantages of the convolutions and the LSTMs to capture the spatial and temporal correlations simultaneously. Wang et al. (2017) introduced the ST-LSTM that allows the memory state to be updated across the stacked recurrent layers along a zigzag state transition path. Villegas et al. (2018) proposed a framework for long-term video generation with a combination of LSTMs and a pose estimation model. Wichers et al. (2018) extended the work from Villegas et al. (2018) by learning hierarchical video representations in an unsupervised manner. Finn et al. (2016) presented a recurrent model based on ConvLSTM to predict how the content of the pixels moves instead of estimating the variations of the pixel values. Wang et al. (2019a) introduced the E3D-LSTM that combines ST-LSTM, the 3D convolution, and a memory attentive module. It builds a memory-augmented recurrent network that can capture long-term video dynamics. Wang et al. (2019b) treated the predictive learning task as a spatiotemporal non-stationary process and proposed to reduce the non-stationarity by replacing the forget gate of ST-LSTM with an inner recurrent structure. There are many other methods focusing on improving the RNN-based predictive models for spatiotemporal data (Kalchbrenner et al., 2017; Liu et al., 2017; Villegas et al., 2017). Besides these deterministic video prediction models, some recent literature explored the video prediction problem by modeling the future uncertainty. These models are either based on adversarial training (Mathieu et al., 2016; Vondrick et al., 2016; Tulyakov et al., 2018) or variational autoencoders (VAEs) (Babaeizadeh et al., 2018; Tulyakov et al., 2018; Denton & Fergus, 2018), or both (Lee et al., 2019; Villegas et al., 2019).

Note that most of the above models, including both stochastic and non-stochastic models, are based on recurrent architectures such as LSTMs. Thus, in this paper, we focus on finding a transfer learning approach particularly designed for LSTM-based predictive networks, while most existing transfer learning techniques are designed for CNNs.

2.2. Transfer Learning

Transfer learning focuses on storing knowledge while solving one problem and applying it to a different but related problem (Long et al., 2015). The ImageNet (Deng et al., 2009) pretrained CNNs have greatly benefited many computer vision tasks such as image classification, object detection, and segmentation. Donahue et al. (2014) proposed a method to leverage the pretrained models, which directly trains a classifier upon the fixed, pretrained CNNs on the target dataset. Apart from the initialization with the pretrained model, Li et al. (2018; 2019) presented several regularization techniques to retain the features learned on the source task, explicitly enhancing the similarity of the final model and the initial one. Rebuffi et al. (2017; 2018) introduced convolutional adapter modules upon pretrained ResNet (He et al., 2016) or VGGNet (Simonyan & Zisserman, 2015) that can adapt the domain-specific knowledge from novel tasks. Liu et al. (2019) developed a model transfer framework named knowledge flow, in which the knowledge is transferred by intermediate features flowing from multiple pretrained teacher CNNs to a randomly initialized student CNN. To make the student CNN independent, it uses a curriculum learning strategy and gently increases the weights of features by the student compared to those by the teachers.

This work is also inspired by the idea of knowledge distillation (Li et al., 2014; Hinton et al., 2015), which transfers knowledge from larger models into smaller, faster models without losing too much generalization ability. Romero et al. (2015) and Zagoruyko & Komodakis (2017) proposed to explicitly produce similar response patterns in the teacher

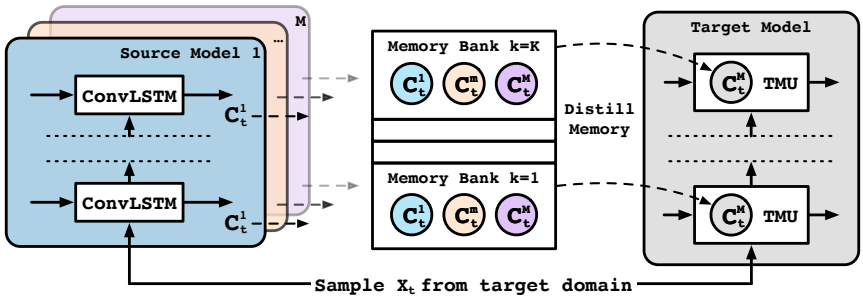

Figure 1. An overview of the *transferable memory* framework, which learns a predictive network on the target dataset from M pretrained networks that were collected from different sources. K is the number of the recurrent layers. Without loss of generality, we use the ConvLSTM (Shi et al., 2015) for the source models, yet this framework can be applied to any variants of LSTMs.

and student feature maps. The above work and many other papers (Huang & Wang, 2017; Yim et al., 2017; Kim et al., 2018; Koratana et al., 2019; Ahn et al., 2019) mainly focus on distilling knowledge to solve the model compression problem within the same dataset. These methods were not designed for cross-domain transfer learning and are therefore different from our approach. Gupta et al. (2016) introduced a cross-modal knowledge distillation technique to transfer supervision between images from different modalities, while our transfer learning approach is unsupervised.

In contrast with all the above transfer learning methods designed for CNNs, we focus on the transfer learning problem for predictive RNNs. This problem is under-explored, especially in spatiotemporal scenarios. In the field of natural language processing, Cui et al. (2019) proposed a recurrent transfer learning framework that transfers hidden states from the teacher RNN to the student RNN. However, upon training, this method still relies on the pretrained teacher models, and thus requires extra memory footprint. Different from this work, our paper presents a novel framework for a new problem, *i.e.*, transferring knowledge across multiple unsupervised prediction tasks for spatiotemporal data.

3. Method

In this section, we provide a solution to the problem of distilling knowledge from unsupervisedly pretrained predictive networks, and transferring it to a new spatiotemporal prediction task. Different from most previous work in transfer learning, our approach is specifically designed for RNN models and unlabelled sequential data. Below we introduce the overall *transferable memory* framework, a new recurrent unit named TMU, and the multi-task training objective for knowledge distillation and sequence prediction.

3.1. Transferable Memory Framework

Why transfer memory representations? The memory state of the LSTM unit (Hochreiter & Schmidhuber, 1997) can latch the gradients during the training process of the recurrent networks, to alleviate the gradient vanishing problem, thereby storing valuable information about the underlying temporal dynamics. In spatiotemporal predictive learning scenarios, the effectiveness of the memory states has also been explored and validated (Wang et al., 2017). They are important for multi-step future prediction as they convey long-term features of the spatiotemporal data. Besides, training an LSTM-based model in the predictive learning manner, *i.e.*, one of the unsupervised learning paradigms, has been empirically proved to successfully learn concept-level representations that can benefit downstream supervised tasks (Wang et al., 2019a). Therefore, we assume that the predictive networks that were pretrained on different unlabeled datasets can provide knowledge of their source domains, and understand the spatiotemporal dynamics of a new task from different perspectives. Now, the question is how to effectively leverage the memory representations of multiple pretrained models. Figure 1 shows our proposed *transferable memory* framework, which enables the student recurrent network to learn from M existing teacher models.

Memory bank. Without loss of generality, we use ConvLSTM (Shi et al., 2015) as the building block of the source models. Note that the proposed framework can be easily applied to other forms of future frames prediction models, such as the Spatiotemporal LSTM (Wang et al., 2017), the Video Pixel Network (Kalchbrenner et al., 2017), the Eidetic LSTM (Wang et al., 2019a), etc. In this paper, we do not focus on discussing how to pretrain the source models. During the training process of the target network on a new dataset, the parameters of the source models are frozen, and they are not taken as the initialization of the target model. In other words, the target model is trained from scratch. It gradually obtains knowledge from the pretrained networks via knowledge distillation. Both the source models and the target one take the same input sequences. Formally, at time step t, each unit of the source model computes

$$\mathcal{H}_t^m, \mathcal{C}_t^m = \text{ConvLSTM}\left(\mathcal{X}_t, \mathcal{H}_{t-1}^m, \mathcal{C}_{t-1}^m\right), \quad (1)$$

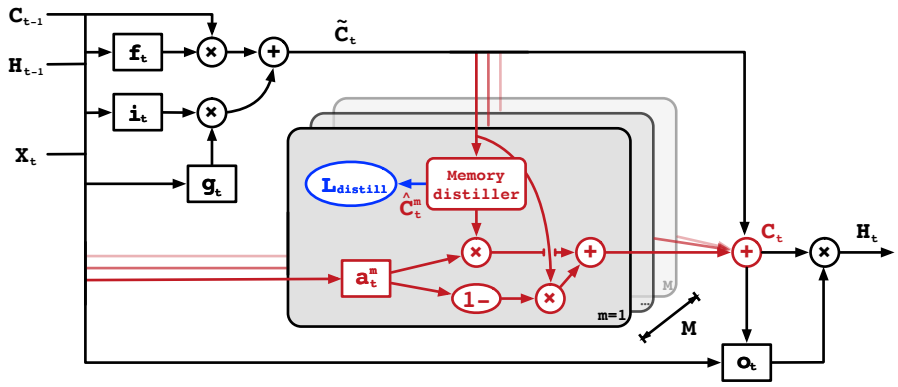

Figure 2. The architecture of the TMU that is used in the target predictive network. It unsupervisedly distills knowledge (in terms of diverse representations of the spatiotemporal dynamics) from a bank of memory states of a zoo of pretrained models. Here, M is the number of pretrained models, and a_t^m is the transfer gate that corresponds to the m-th pretrained model.

where $\mathcal{X}_t$ is the input state that can be an input frame or the hidden state from the lower layer. $\mathcal{H}_t^m, \mathcal{C}_t^m$ are respectively the hidden state and memory state of the m-th pretrained networks, where $m \in \{1, \dots, M\}$. Then we obtain the *memory bank* in forms of $\{\mathcal{C}_t^1, \dots, \mathcal{C}_t^M\}$, which contains diverse representations of the spatiotemporal dynamics, part of which can contribute to the target task.

3.2. Transferable Memory Unit

The *Transferable Memory Unit* (TMU) is the basic building block of the target network (see Figure 1). It is designed to distill transferable features from the memory bank and dynamically adjust the influence of all source networks. As shown in Figure 2, the main architecture of TMU has three components: a *memory distiller* module, a set of *transfer gate*, and the basic operations following the ConvLSTM (Shi et al., 2015), which are specified as follows:

$$\begin{aligned} g_t &= \tanh\left(W_{xg} * \mathcal{X}_t + W_{hg} * \mathcal{H}_{t-1} + b_g\right) \\ i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \odot \mathcal{C}_{t-1} + b_s\right) \\ f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \odot \mathcal{C}_{t-1} + b_f\right) \\ \widetilde{\mathcal{C}}_t &= f_t \odot \mathcal{C}_{t-1} + i_t \odot g_t, \end{aligned} \tag{2}$$

where σ is sigmoid activation function, $*$ and $\odot$ denote the convolution operator and the Hadamard product respectively. Unless otherwise mentioned, all through this text, the convolutional filters are 5×5. The use of the input gate i_t, forget gate f_t, and input-modulation gate g_t controls the information flow towards the intermediate memory state $\widetilde{\mathcal{C}}_t$. Here we build TMU upon ConvLSTM for the sake of convenience, yet the proposed memory distiller and transfer gate, being displayed as a whole by the gray box in Figure 2, can be seamlessly integrated into any forms of LSTM-like recurrent units.

Memory distiller module. The memory distiller module is largely inspired by recent advances on *compressing many visual domains in relatively small networks, with substantial parameter sharing between them* (Rebuffi et al., 2017; 2018), which have also been shown to mitigate the forgetting problem of finetuning. However, different from these existing methods that were particularly designed for transferring knowledge from a single source model to multiple target models, our memory distiller is used for vice-versa. As shown in Figure 2, TMU contains M memory distiller modules, corresponding to the number of source models. Each memory distiller takes $\widetilde{\mathcal{C}}_t$ as input, and employs a 1×1 convolutional layer parametrized as W_{distill}^m for each pretext task, followed by layer normalization (Ba et al., 2016):

$$\widehat{\mathcal{C}}_t^m = \text{LayerNorm}\left(W_{\text{distill}}^m * \widetilde{\mathcal{C}}_t\right). \tag{3}$$

We then use the generated features $\{\widehat{\mathcal{C}}_t^1, \dots, \widehat{\mathcal{C}}_t^M\}$ to distill knowledge from the memory bank mentioned above $\{\mathcal{C}_t^1, \dots, \mathcal{C}_t^M\}$. Over all pretext tasks and across the time horizon, we minimize the Euclidean distance between pairs of memory states:

$$\mathcal{L}_{\text{distill}} = \sum_{s=1}^{M} \sum_{t=1}^{T} \|\widehat{\mathcal{C}}_t^m - \mathcal{C}_t^m\|_2^2. \tag{4}$$

The distillation loss enables TMU to learn separately from multiple teachers, thereby gaining substantial prior knowledge of the complex spatiotemporal dynamics. In this way, throughout the training process, the student network can focus on more domain-specific patterns of the target dataset. Noticeably, the target memory $\widehat{\mathcal{C}}_t^m$ would not converge to the mean of source memories as in Eq. (3) we have M sets of parameters W_{distill}^m to match each $\widehat{\mathcal{C}}_t^m$ with the corresponding source memory.

Transfer gate. However, two problems remain for learning from multiple pretext domains. First, not all memory representations by the source models are transferable and yield a positive effect to the target task. Second, the source models should not equally contribute to the target one. To further solve these problems, we propose to learn a set of transfer gates $\{a_t^1, \ldots, a_t^M\}$ to adaptively control the information flow from the previously distilled memory representations $\{\widehat{\mathcal{C}}_t^1, \ldots, \widehat{\mathcal{C}}_t^M\}$ to the final memory state $\mathcal{C}_t$. Finally, TMU obtains the output hidden state as follows.

$$\begin{aligned} a_t^m &= \sigma\left(W_{xm} * \mathcal{X}_t + W_{hm} * \mathcal{H}_{t-1} + b_m\right) \\ \mathcal{C}_t &= \widetilde{\mathcal{C}}_t + \sum_{m=1}^{M} \left(a_t^m \odot \widehat{\mathcal{C}}_t^m + (1 - a_t^m) \odot \widetilde{\mathcal{C}}_t\right) \\ o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \odot \mathcal{C}_t + b_o\right) \\ \mathcal{H}_t &= o_t \odot \tanh\left(\mathcal{C}_t\right). \end{aligned} \tag{5}$$

The complete computation of TMU consists of Eq. (2), Eq. (3), and Eq. (5). When a_t^m approaches 1, more pretext knowledge is distilled from the m-th source domain to the learned model. By controlling the states in $\{a_t^1, \ldots, a_t^M\}$, TMU can dynamically adjust the influence of M sources.

3.3. Unsupervised Training Objective

All the training procedures of the transfer memory framework are unsupervised. The final training objective is:

$$\mathcal{L}_{\text{final}} = \sum_{t=2}^{T} \|\widehat{\mathcal{X}}_t - \mathcal{X}_t\|_2^2 + \beta \sum_{k=1}^{K} \mathcal{L}_{\text{distill}}^k, \tag{6}$$

where $\widehat{\mathcal{X}}_t$ is the generated frame, $k \in \{1, \ldots, K\}$ is the index of the TMU layer, $\mathcal{L}_{\text{distill}}^k$ is defined in Eq. (4), and β is a hyper-parameter tuned on the target validation set. It is worth noting that we do not use any parameters to the distillation loss terms of different source domains in Eq. (4). It is because that due to Eq. (3), $\{\widehat{\mathcal{C}}_t^1, \ldots, \widehat{\mathcal{C}}_t^M\}$ can learn domain-specific patterns so that $\widetilde{\mathcal{C}}_t$ can focus on common ones. Further, in Eq. (5), the transfer gates $\{a_t^1, \ldots, a_t^M\}$ dynamically adjust the significance of all source domains.

4. Experiments

We study unsupervised transfer learning performed between different spatiotemporal prediction tasks, within or across the following three benchmarks:

Flying digits. This synthetic benchmark has three Moving MNIST datasets with respectively 1, 2, or 3 flying digits randomly sampled from the static MNIST dataset. Each dataset contains 10,000 training sequences, 2,000 validation sequences, and 3,000 testing sequences. Each sequence consists of 20 consecutive frames, 10 for the input, and 10 for the prediction. Each frame is of the resolution of 64×64.

Human motion. This benchmark is built upon three human action datasets with real-world videos: Human3.6M (Ionescu et al., 2013), KTH (Schuldt et al., 2004), and Weizmann (Blank et al., 2005). Specifically, we use the Human3.6M dataset as the target domain, which has 2,220 sequences for training, 300 for validation, and 1,056 for testing. We follow (Wang et al., 2019b) to resize each RGB frame to the resolution of 128×128, and make the model predict 4 future frames based on 4 previous ones.

Precipitation nowcasting. Precipitation nowcasting is a meaningful application of spatiotemporal prediction. This benchmark consists of three radar echo datasets[1]: two of them are from separate years of Guangzhou, and the other one is from Beijing, which is a more arid place. The Guangzhou2016 dataset has 33,769 consecutive radar observations, collected every 6 minutes. The Guangzhou2014 dataset has 9,998 observations. Though the data sources are different, these two Guangzhou datasets both contain the rainy seasons of the city. We use the Beijing dataset as the target domain, which suffers from a large amount of ineffective training data due to the lack of rain. The Beijing dataset has 55,466 observations for training, 3,000 for validation, and 12,711 for testing. All frames are resized to the resolution of 256×256.

Implementation. On all benchmarks, our final model has four stacked TMU layers with 64-channel hidden states. We use the ADAM optimizer (Kingma & Ba, 2015) with a starting learning rate of 0.001 for training the TMU network. Unless otherwise mentioned, the batch size is set to 8, and the training process is stopped after 80,000 iterations. All experiments are implemented in PyTorch (Paszke et al., 2019) and conducted on NVIDIA TITAN-RTX GPUs. We run all experiments three times and use the average results for quantitative evaluation. As for the dimensionality of the tensors, all the dimensions of the source and target states should be matched, including the number of channels (P), width (W), and height (H). For example, on the human motion benchmark, both C_t^m (source, KTH/WEI) and C_t (target, Human3.6M) are 3D tensors of $64 \times 32 \times 32$. We use a standard frame sub-scaling method (Shi et al., 2015) to transform the input images from $P \times W \times H$ to $(P \cdot K \cdot K) \times (W/K) \times (H/K)$ and control K to make W/K and H/K constant across source and target domains.

4.1. Flying Digits Benchmark

Setups. We take the 3-digits Moving MNIST dataset as the target domain. Due to frequent occlusions and complex motions, it is challenging to accurately predict the trajecto-

[1] Predicting the shapes and trajectories of future radar echoes is the foundation of accurate precipitation nowcasting (Shi et al., 2015; 2017; Wang et al., 2017).

Table 1. Quantitative results on the flying digits benchmark. We use the 3-digits subdataset as the target domain. A lower MSE or a higher SSIM per frame indicates better prediction results. All compared models are built upon the same ConvLSTM architecture.

METHOD	SOURCES	MSE	SSIM	#PARAMETERS		RUNTIME	
				TRAIN	TEST	TRAIN	TEST
CONVLSTM (SHI ET AL., 2015)	NONE	120.5	0.712	3.0M	3.0M	0.35S/BATCH	16.3MS/SEQ
TMU (TRAIN FROM SCRATCH)	NONE	120.6	0.715	-	-	-	-
TMU (FINETUNE)	1 DIGIT	114.8	0.720	3.0M	3.0M	0.35S/BATCH	16.3MS/SEQ
TMU (FINETUNE)	2 DIGITS	110.0	0.732	-	-	-	-
L2SP (LI ET AL., 2018)	1 DIGIT	118.5	0.703	3.0M	3.0M	0.39S/BATCH	16.3MS/SEQ
L2SP (LI ET AL., 2018)	2 DIGITS	116.4	0.705	-	-	-	-
KNOWLEDGE FLOW (LIU ET AL., 2019)	BOTH	107.2	0.748	10.1M	4.1M	0.54S/BATCH	21.7MS/SEQ
ART (CUI ET AL., 2019)	BOTH	105.0	0.734	10.1M	10.1M	0.73S/BATCH	25.3MS/SEQ
TMU (MEMORY TRANSFER)	1 DIGIT	96.1	0.762	-	-	-	-
TMU (MEMORY TRANSFER)	2 DIGITS	97.3	0.756	-	-	-	-
TMU (MEMORY TRANSFER)	BOTH	**94.7**	**0.777**	9.5M	3.6M	0.43S/BATCH	17.7MS/SEQ

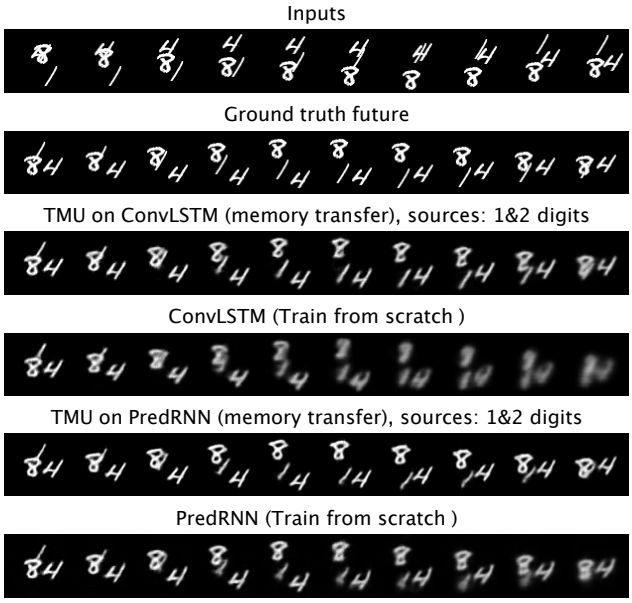

Figure 3. Predicted frames on the flying digits benchmark. Our transfer learning approach can consistently outperform the baseline predictive networks being trained from scratch.

ries of all three digits. We expect to improve this task by transferring the understandings of the digit's motion from the existing models that were pretrained with fewer digits.

Comparing with training from scratch. Table 1 and Figure 3 respectively give the quantitative and qualitative results of our approach. Compared with training a model on the target dataset from scratch, it gains significant improvements by learning from pretrained models on 1&2-digits Moving MNIST. Besides, by comparing the training-from-scratch TMU model without any pretrained models with the training-from-scratch ConvLSTM network (the first two rows in Table 1), we may conclude that it is the *memory transfer* mechanism that improves the final results, instead of the engineering on the network architecture or the increased number of model parameters.

Comparing with previous transfer learning methods. Also shown in Table 1, our approach outperforms finetuning by 17.5% in MSE. It also achieves better results than existing transfer learning approaches, including L2SP (Li et al., 2018), Knowledge Flow (Liu et al., 2019), and ART (Cui et al., 2019). Furthermore, compared with finetuning, TMU with two sources only increases the number of parameters slightly at test time but improves MSE and SSIM remarkably. Compared with ART (Cui et al., 2019), which is also particularly designed for RNNs, our approach only requires about one-third of the number of model parameters at test time. Thus, it does not increase the memory usage linearly with the growth of sources. As for the training stage, all multi-source transfer learning models are forced to yield more parameters. A TITAN-RTX GPU can hold up to 41 source ConvLSTM models and a target TMU model, which is sufficient for most practical application scenarios.

Backbones. Our TMU can also be applied to other LSTM-based predictive models. We use PredRNN (Wang et al., 2017) and MIM (Wang et al., 2019b) to take the place of the ConvLSTM network, covering both deterministic and stochastic models. Quantitative results and prediction examples are respectively shown in Table 2 and Figure 3. The proposed TMU network achieves better results than directly finetuning the pretrained PredRNN or MIM on the 3-digits dataset. It significantly improves the state-of-the-art MIM model in all metrics.

Hyper-parameters. Last but not least, we show the sensitivity analysis of the training hyper-parameter β in Figure 4. It achieves the best results at 0.1 on the flying digits bench-

Table 2. MSE/SSIM results of TMU upon different network backbones. We take the 3-digits Moving MNIST as the target domain.

METHOD	SOURCES	PREDRNN	MIM
FROM SCRATCH	NONE	93.4/0.802	89.0/0.783
FINETUNE	1 DIGIT	91.1/0.811	84.5/0.794
FINETUNE	2 DIGITS	89.4/0.816	83.2/0.801
MEM. TRANSFER	BOTH	**84.9/0.828**	**75.3/0.838**

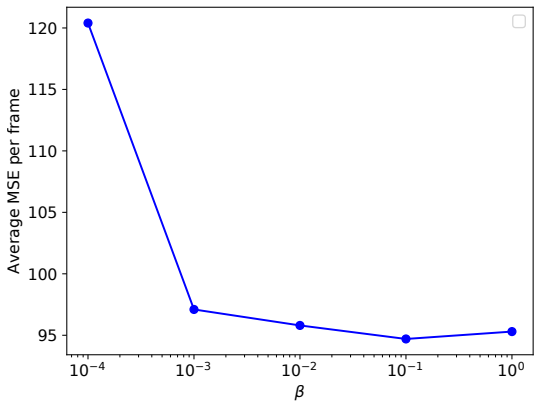

Figure 4. Sensitivity analysis of the hyper-parameter (β) for the unsupervised training objective on the flying digits benchmark.

mark and is robust and easy to tune in the range of 10^{-3} to 1. We have similar results on the other two benchmarks and thus set β to 0.1 throughout this paper.

4.2. Human Motion Benchmark

Setups. Compared with the KTH and Weizmann datasets with limited variability (they are small sets of backgrounds and actions, performed by a small group of individuals), the Human3.6M dataset contains larger amounts of data and more complex human motions, which makes it difficult to predict the future frames. On this benchmark, we take Human3.6M as the target domain and the other two datasets as the source domains. We use ConvLSTM (Shi et al., 2015), MIM (Wang et al., 2019b), and SAVP (Lee et al., 2019) as the network backbone of TMU, covering both deterministic and stochastic models.

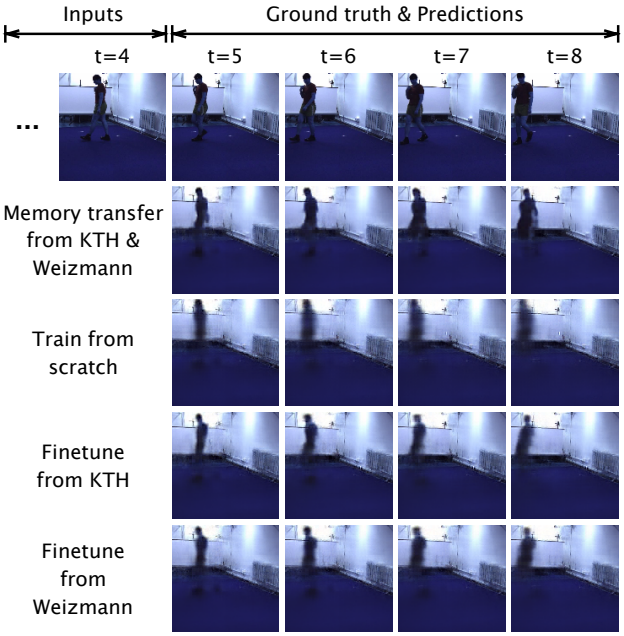

Figure 5. Prediction examples on Human3.6M by TMU networks based on ConvLSTM. Our method obtains the sharpest predictions.

Results. We show the quantitative evaluations in Table 3. The baseline TMU network, which takes either of the KTH or Weizmann datasets as the source domain, consistently outperforms the finetuning counterpart by large margins. The final TMU network that learns from both pretrained models further improves the prediction quality on the target task, which is because of the effectiveness of the *transfer gates*. Besides, by using MIM and SAVP as the network backbones, we validate that TMU can outperform strong competitors that are pretrained well on the source datasets. Moreover, we can see from Figure 5 that the generated frames of the finetuning models largely suffer from blur effect, indicating that they are unable to capture a clear trend of motion. By contrast, the TMU network provides the sharpest results. We may conclude that the pretrained models from domains of plain backgrounds and simple actions can facilitate the training process of the model on a more challenging task, and the proposed *transferable memory* framework can enhance this positive effect.

Table 3. Quantitative results averaged per frame on Human3.6M using different network backbones, including ConvLSTM (Shi et al., 2015), MIM (Wang et al., 2019b), and SAVP (Lee et al., 2019). For SAVP, we take the best one in SSIM from 100 prediction samples.

MODEL	METHOD	SOURCES	CONVLSTM		MIM		SAVP	
			MSE	SSIM	MSE	SSIM	MSE	SSIM
TMU	TRAIN FROM SCRATCH	NONE	504.2	0.762	430.5	0.790	465.2	0.792
	FINETUNE	KTH	472.0	0.778	420.1	0.796	453.7	0.808
	FINETUNE	WEIZMANN	476.4	0.774	422.9	0.793	458.1	0.805
	MEMORY TRANSFER	KTH & WEIZMANN	**442.5**	**0.794**	**394.2**	**0.813**	**430.2**	**0.831**

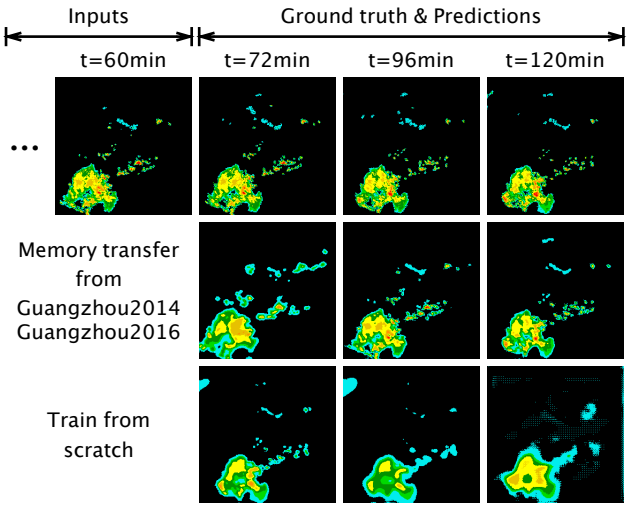

Figure 6. Prediction examples on the Beijing radar echo dataset by TMU networks based on ConvLSTM. We mainly compare the predicted high-intensity areas in yellow and red.

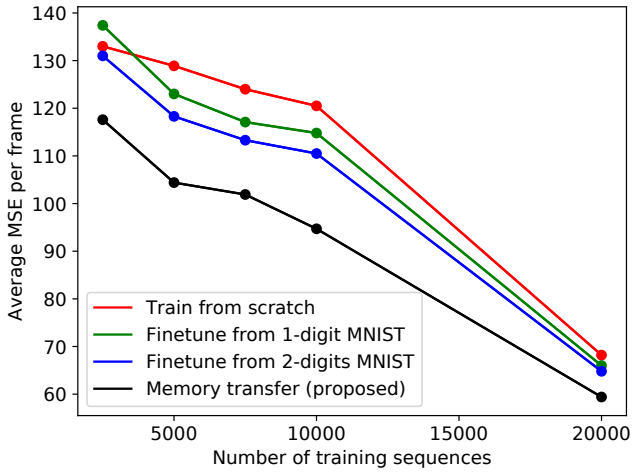

Figure 7. Averaged MSE per frame with respect to different numbers of training sequences of the target domain of the flying digits.

4.3. Precipitation Nowcasting Benchmark

Setups. We forecast the next 10 radar echo frames from the previous 10 observations, covering weather conditions in the next hour. Due to the lack of effective training data in the Beijing dataset, we take Guangzhou2014 and Guangzhou2016 as the source domains and pretrain models on these two datasets. Different from the previous experiments, the convolutional filters inside both the source ConvLSTM networks and the target TMU network are 3×3.

Results. In addition to MSE and MAE, here we also evaluate the predicted radar echoes using the Critical Success Index (CSI), which is defined as $\frac{\text{Hits}}{\text{Hits} + \text{Misses} + \text{FalseAlarms}}$. Here, hits correspond to the true positive, misses correspond to the false positive, and false alarms correspond to the false negative. We set the alarm threshold to 20 dBZ. Compared with MSE and MAE, this metric is particularly sensitive to the high-intensity echoes. A higher CSI indicates better prediction results. As shown in Table 4, the TMU network remarkably outperforms the finetuning method in all evaluation metrics. Figure 6 provides showcases of predictions taking the Guangzhou radar echo datasets as source domains. Note that the ConvLSTM network without the help of the transferable memory framework makes fuzzy predictions, while the final TMU model forecasts the positions of high-intensity echoes (areas in red and yellow) more accurately.

4.4. Further Analysis and Empirical Findings

What if the target domain has sufficient training data? Based on the previous studies of transfer learning performed between supervised tasks, someone may concern that the effect of *transferable memory* mechanism will degenerate when the number of training samples increases for the target domain. We explore this problem by training target models with respectively 25%, 50%, 75%, 100%, and 200% training sequences for the 3-digits Moving MNIST, and evaluate all the models on the same set (see Figure 7). We observe the TMU network consistently outperforms the ones trained from scratch or finetuned upon the source models. Specifically, in the case that the target set has twice the training samples (20,000) than those in the standard settings, the finetuning method fails to remarkably improve the training-from-scratch baseline, while our approach achieves larger improvements. We may conclude that the main cause is that TMU enables successful distillation of the diverse understandings about the complex spatiotemporal dynamics, which can be a meaningful supplement to the target domain.

Finding favorable transfer schemes. We then explore the transfer schemes that can maximize the effectiveness of our method. What assumptions does our method make on the pretrained models? Table 6 provides the results of different transfer schemes towards the KTH dataset. We observe that although the pretrained model on the KTH dataset can greatly help the training process on the Human3.6M dataset (SSIM: $0.762 \rightarrow 0.790$, Table 3), conversely, the pretrained model from Human3.6M only has a slight effect on the KTH result (SSIM: $0.771 \rightarrow 0.774$, Table 6). There are two possible causes: the first is that representations learned from the complex Human3.6M dataset do not have strong transferability; the second is that the training-from-scratch model on the KTH dataset is strong enough and cannot be further improved. To find the reason, we pretrain a model on the 2-digits Moving MNIST dataset and apply it to the training process on KTH. We observe that such a transfer

Table 4. Comparisons of transfer learning schemes within or across benchmarks using the Beijing radar echo dataset as the target domain.

MODEL	METHOD	SOURCES	MSE	MAE	CSI
TMU ON CONVLSTM	TRAIN FROM SCRATCH	NONE	110.5	219.9	0.348
	FINETUNE	GUANGZHOU2014	96.6	198.3	0.368
	FINETUNE	GUANGZHOU2016	94.9	197.3	0.375
	MEMORY TRANSFER	2-DIGITS MNIST & KTH	91.8	192.8	0.382
	MEMORY TRANSFER	GUANGZHOU2014	87.4	193.3	0.374
	MEMORY TRANSFER	GUANGZHOU2016	84.7	191.6	0.384
	MEMORY TRANSFER	GUANGZHOU2014 & 2016	77.3	184.2	0.403
	MEMORY TRANSFER	GUANGZHOU & MNIST & KTH	**77.1**	**181.3**	**0.408**

Table 5. The averages of transfer gate on the Beijing radar echo dataset. The model corresponds to the last one in Table 4, where two relevant sources and two less relevant sources are used. The results show the significance of each source domain to the target domain.

METRIC	GUANGZHOU2014	GUANGZHOU2016	2-DIGITS MNIST	KTH
VALUES OF TRANSFER GATE (a_t^m)	0.60	0.61	0.43	0.39

Table 6. Results of different transfer schemes using KTH as the target domain. All models are built upon the ConvLSTM network.

METHOD	SOURCE	SSIM
TRAIN FROM SCRATCH	NONE	0.771
MEMORY TRANSFER	HUMAN3.6M	0.774
MEMORY TRANSFER	2-DIGITS MNIST	0.808

scheme obtains remarkable improvements over the baseline (SSIM: $0.771 \rightarrow 0.808$). Therefore, we can rule out the validity of the second hypothesis. Since the Moving MNIST dataset only contains deterministic motions, the pretrained model yields less uncertainty about the future spatiotemporal dynamics. We may conclude that *a favorable transfer scheme is to use the knowledge of better pretrained, more deterministic source models, and thus the final model can focus more on the domain-specific mode of target data.*

Will content-irrelevant source domains benefit the target predictive learning task? We take the radar echo dataset from the city of Beijing as the target domain, and the seemingly irrelevant 2-digits Moving MNIST dataset and KTH dataset as the source domains. From Table 4, we find that using KTH and Moving MNIST pretrained models can greatly help the prediction results, which outperforms the training-from-scratch TMU baseline (CSI: $0.348 \rightarrow 0.382$). Such results might be counter-intuitive, yet important to our understandings of the transferability of spatiotemporal modeling. Further, we take both the relevant Guangzhou2014 and Guangzhou2016 datasets as well as the seemingly irrelevant Moving MNIST dataset and KTH dataset as source domains. As opposed to our common sense for supervised transfer learning, TMU benefits from the less relevant sources, which are unrelated in image appearance but related in temporal dynamics (can be transferable). We then use the averages of a_t^m to analyze the significance of each source domain. As shown in Table 5, TMU has higher a_t^m for the Guangzhou radar echo datasets, indicating that it can adaptively control the influence of different sources via the transfer gates. Unlike supervised transfer learning, where irrelevant source data may cause negative effects, our approach can associate a variety of source domains and transfer temporal dynamics even if the content of source videos seems irrelevant.

5. Conclusion and Discussion

In this paper, we studied a new unsupervised transfer learning problem of using multiple pretrained models to improve the performance of a new spatiotemporal predictive learning task. We used the term unsupervised for two reasons. First, we only explored the transfer learning cases between multiple unsupervised tasks. Second, the proposed method does not require any labels. We proposed the transferable memory framework, which transfers knowledge from multi-source RNNs and yielded better results than finetuning. Our approach was shown effective even in the case that there is adequate data for the target domain, or the pretrained models were collected from less relevant domains. Code and datasets are made available at `https://github.com/thuml/transferable-memory`.

One potential work in the future is to explore how to transfer knowledge between unsupervised tasks beyond predictive learning. Another one is that how we can transfer knowledge from unsupervisedly learned RNN models to CNN models of the downstream supervised task. Our approach is also likely to be effective for supervised tasks, though it may not be the best choice when labels are available. It is worth exploring, but beyond the scope of the paper.

Acknowledgements

This work was supported by the Natural Science Foundation of China (61772299, 71690231), and China University S&T Innovation Plan Guided by the Ministry of Education.

References

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., and Dai, Z. Variational information distillation for knowledge transfer. In *CVPR*, pp. 9163–9171, 2019.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. In *ICLR*, 2018.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. Actions as space-time shapes. In *ICCV*, pp. 1395–1402, 2005.

Cui, W., Zheng, G., Shen, Z., Jiang, S., and Wang, W. Transfer learning for sequences via learning to collocate. In *ICLR*, 2019.

De Brabandere, B., Jia, X., Tuytelaars, T., and Van Gool, L. Dynamic filter networks. In *NeurIPS*, pp. 667–675, 2016.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. ImageNet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Denton, E. and Fergus, R. Stochastic video generation with a learned prior. In *ICML*, pp. 1182–1191, 2018.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, pp. 647–655, 2014.

Finn, C. and Levine, S. Deep visual foresight for planning robot motion. In *ICRA*, pp. 2786–2793, 2017.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, pp. 64–72, 2016.

Gupta, S., Hoffman, J., and Malik, J. Cross modal distillation for supervision transfer. In *CVPR*, pp. 2827–2836, 2016.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In *NeurIPS*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, pp. 1735–1780, 1997.

Huang, Z. and Wang, N. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, pp. 1325–1339, 2013.

Kalchbrenner, N., Oord, A. v. d., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. In *ICML*, pp. 1771–1779, 2017.

Kim, J., Park, S., and Kwak, N. Paraphrasing complex network: Network compression via factor transfer. In *NeurIPS*, pp. 2765–2774, 2018.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Koratana, A., Kang, D., Bailis, P., and Zaharia, M. LIT: Learned intermediate representation training for model compression. In *ICML*, pp. 3509–3518, 2019.

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., and Levine, S. Stochastic adversarial video prediction. In *ICLR*, 2019.

Li, J., Zhao, R., Huang, J.-T., and Gong, Y. Learning small-size DNN with output-distribution-based criteria. In *INTERSPEECH*, pp. 1910–1914, 2014.

Li, X., Grandvalet, Y., and Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, pp. 2830–2839, 2018.

Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., and Huan, J. DELTA: Deep learning transfer using feature map with attention for convolutional networks. In *ICLR*, 2019.

Liu, I., Peng, J., and Schwing, A. G. Knowledge flow: Improve upon your teachers. In *ICLR*, 2019.

Liu, Z., Yeh, R., Tang, X., Liu, Y., and Agarwala, A. Video frame synthesis using deep voxel flow. In *ICCV*, pp. 4473–4481, 2017.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In *ICML*, pp. 97–105, 2015.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. In *NeurIPS*, pp. 2863–2871, 2015.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8024–8035, 2019.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. Learning multiple visual domains with residual adapters. In *NeurIPS*, pp. 506–516, 2017.

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pp. 8119–8127, 2018.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. FitNets: Hints for thin deep nets. In *ICLR*, 2015.

Schuldt, C., Laptev, I., and Caputo, B. Recognizing human actions: a local SVM approach. In *ICPR*, pp. 32–36, 2004.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, pp. 802–810, 2015.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NeurIPS*, pp. 5617–5627, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Srivastava, N., Mansimov, E., and Salakhutdinov, R. Unsupervised learning of video representations using LSTMs. In *ICML*, pp. 843–852, 2015.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, pp. 1526–1535, 2018.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.

Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. Learning to generate long-term future via hierarchical prediction. In *ICML*, pp. 3560–3569, 2018.

Villegas, R., Pathak, A., Kannan, H., Erhan, D., Le, Q. V., and Lee, H. High fidelity video prediction with large stochastic recurrent neural networks. In *NeurIPS*, pp. 81–91, 2019.

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. In *NeurIPS*, pp. 613–621, 2016.

Wang, Y., Long, M., Wang, J., Gao, Z., and Philip, S. Y. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NeurIPS*, pp. 879–888, 2017.

Wang, Y., Jiang, L., Yang, M.-H., Li, L.-J., Long, M., and Fei-Fei, L. Eidetic 3D LSTM: A model for video prediction and beyond. In *ICLR*, 2019a.

Wang, Y., Zhang, J., Zhu, H., Long, M., Wang, J., and Yu, P. S. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *CVPR*, pp. 9154–9162, 2019b.

Wichers, N., Villegas, R., Erhan, D., and Lee, H. Hierarchical long-term video prediction without supervision. In *ICML*, pp. 6033–6041, 2018.

Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J. Learning to see physics via visual de-animation. In *NeurIPS*, pp. 152–163, 2017.

Xu, Z., Wang, Y., Long, M., and Wang, J. PredCNN: Predictive learning with cascade convolutions. In *IJCAI*, pp. 2940–2947, 2018.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, pp. 7130–7138, 2017.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.