Unsupervised Video Domain Adaptation for Action Recognition: A Disentanglement Perspective

Pengfei Wei  
AI Lab, ByteDance  
pengfei.wei@bytedance.com

Lingdong Kong  
National University of Singapore  
lingdong@comp.nus.edu.sg

Xinghua Qu  
AI Lab, ByteDance  
quxinghua17@gmail.com

Yi Ren  
AI Lab, ByteDance  
ren.yi@bytedance.com

Zhiqiang Xu  
MBZUAI  
zhiqiang.xu@mbzuai.ac.ae

Jing Jiang  
University of Technology Sydney  
Jing.Jiang@uts.edu.au

Xiang Yin  
AI Lab, ByteDance  
yinxiang.stephen@bytedance.com

Abstract

Unsupervised video domain adaptation is a practical yet challenging task. In this work, for the first time, we tackle it from a disentanglement view. Our key idea is to handle the spatial and temporal domain divergence separately through disentanglement. Specifically, we consider the generation of cross-domain videos from two sets of latent factors, one encoding the static information and another encoding the dynamic information. A Transfer Sequential VAE (TranSVAE) framework is then developed to model such generation. To better serve for adaptation, we propose several objectives to constrain the latent factors. With these constraints, the spatial divergence can be readily removed by disentangling the static domain-specific information out, and the temporal divergence is further reduced from both frame- and video-levels through adversarial learning. Extensive experiments on the UCF-HMDB, Jester, and Epic-Kitchens datasets verify the effectiveness and superiority of TranSVAE compared with several state-of-the-art approaches.1

1 Code is publicly available at: https://github.com/ldkong1205/TranSVAE

1 Introduction

Over the past decades, unsupervised domain adaptation (UDA) has attracted extensive research attention [41]. Numerous UDA methods have been proposed and successfully applied to various real-world applications, e.g., object recognition [38, 42, 47], semantic segmentation [31, 19, 33, 24], and object detection [3, 15, 45]. However, most of these methods and their applications are limited to the image domain, while much less attention has been devoted to video-based UDA, where the latter is undoubtedly more challenging.

Compared with image-based UDA, the source and target domains also differ temporally in video-based UDA. Images are spatially well-structured data, while videos are sequences of images with both spatial and temporal relations. Existing image-based UDA methods can hardly achieve satisfactory performance on video-based UDA tasks as they fail to consider the temporal dependency of video frames in handling the domain gaps. For instance, in video-based cross-domain action recognition tasks, domain gaps are presented by not only the actions of different persons in different scenarios but also the actions that appear at different timestamps or last at different time lengths.

1 Code is publicly available at: https://github.com/ldkong1205/TranSVAE

Figure 1: Conceptual comparisons between the traditional *all-in-one view* and the proposed *disentanglement view*. Prior works often seek to compress implicit domain information to obtain domain-indistinguishable representations; while in this work, we pursue explicit decouplings of domain-specific information from other information via generative modeling.

Recently, few works have been proposed for video-based UDA. The key idea is to achieve *domain alignment* by aligning both frame- and video-level features through adversarial learning [7, 26], contrastive learning [37, 31], attention [10], or combination of these mechanisms, e.g., adversarial learning with attention [29, 8]. Though they have advanced video-based UDA, there is still room for improvement. Generally, existing methods follow an *all-in-one* way, where both spatial and temporal domain divergence are handled together, for adaptation (Fig. 1). However, cross-domain videos are highly complex data containing diverse mixed-up information, e.g., domain, semantic, and temporal information, which makes the simultaneous elimination of spatial and temporal divergence insufficient. This motivates us to handle the video-based UDA from a *disentanglement* perspective (Fig. 1) so that the spatial and temporal divergence can be well handled separately.

To achieve this goal, we first consider the *generation* process of cross-domain videos, as shown in Fig. 2, where a video is generated from two sets of latent factors: one set consists of a sequence of random variables, which are *dynamic* and incline to encode the semantic information for downstream tasks, e.g., action recognition; another set is *static* and introduces some domain-related spatial information to the generated video, e.g., style or appearance. Specifically, the blue / red nodes are the observed source / target videos $x^S / x^T$, respectively, over $t$ timestamps. Static latent variables $z^d_S$ and $z^d_T$ follow a joint distribution and combining either of them with dynamic latent variables $z_t$ constructs one video data of a domain.

With the above generative model, we develop a *Transfer Sequential Variational AutoEncoder (TranSVAE)* for video-based UDA. TranSVAE handles the cross-domain divergence in two levels, where the first level removes the spatial divergence by disentangling $z_d$ from $z_t$; while the second level eliminates the temporal divergence of $z_t$. To achieve this, we leverage appropriate constraints to ensure that the disentanglement indeed serves the adaptation purpose. Firstly, we enable a good decoupling of the two sets of latent factors by minimizing their *mutual dependence*. This encourages these two latent factor sets to be mutually independent. We then consider constraining each latent factor set. For $z^D_d$ with $D \in \{S, T\}$, we propose a *contrastive triplet loss* to make them static and domain-specific. This makes us readily handle spatial divergence by disentangling $z^D_d$ out. For $z_t$, we propose to align them across domains at both frame and video levels through *adversarial learning* so as to further eliminate the temporal divergence. Meanwhile, as downstream tasks use $z_t$ as input, we also add the *task-specific supervision* on $z_t$ extracted from source data (w/ ground-truth).

To the best of our knowledge, this is the first work that tackles the challenging video-based UDA from a domain disentanglement view. We conduct extensive experiments on popular benchmarks (UCF-HMDB, Jester, Epic-Kitchens) and the results show that TranSVAE consistently outperforms previous state-of-the-art methods by large margins. We also conduct comprehensive ablation studies.
and disentanglement analyses to verify the effectiveness of the latent factor decoupling. The main contribution of the paper is summarized as follows:

- We provide a generative perspective on solving video-based UDA problems. We develop a generative graphical model for the cross-domain video generation process and propose to utilize the sequential VAE as the base generative model.
- Based on the above generative view, we propose a TranSVAE framework for video-based UDA. By developing four constraints on the latent factors to enable disentanglement to benefit adaptation, the proposed framework is capable of handling the cross-domain divergence from both spatial and temporal levels.
- We conduct extensive experiments on several benchmark datasets to verify the effectiveness of TranSVAE. A comprehensive ablation study also demonstrates the positive effect of each loss term on video domain adaptation.

2 Related Work

Unsupervised Video Domain Adaptation. Despite the great progress in image-based UDA, only a few methods have recently attempted video-based UDA. In [7], a temporal attentive adversarial adaptation network (TA3N) is proposed to integrate a temporal relation module for temporal alignment. Choi et al. [10] proposed a SAVA method using self-supervised clip order prediction and clip attention-based alignment. Based on a cross-domain co-attention mechanism, the temporal co-attention network TCoN [29] focused on common key frames across domains for better alignment. Luo et al. [26] pay more attention to the domain-agnostic classifier by using a network topology of the bipartite graph to model the cross-domain correlations. Instead of using adversarial learning, Sahoo et al. [31] developed an end-to-end temporal contrastive learning framework named CoMix with background mixing and target pseudo-labels. Recently, Chen et al. [8] learned multi-level temporal attentive features to achieve better alignment, while Turrisi et al. [37] exploited two-headed deep architecture to learn a more robust target classifier by the combination of cross-entropy and contrastive losses. Although these approaches have advanced video-based UDA tasks, they all adopted to align features with diverse information mixed up from a compression perspective, which leaves room for further improvements.

Multi-Modal Video Adaptation. Most recently, there are also a few works integrating multiple modality data for video-based UDA. Although we only use the single modality RGB features, we still discuss this multi-modal research line for a complete literature review. The very first work exploring the multi-modal nature of videos for UDA is MM-SADA [28], where the correspondence of multiple modalities was exploited as a self-supervised alignment in addition to adversarial alignment. A later work, spatial-temporal contrastive domain adaptation (STCDA) [34], utilized a video-based contrastive alignment as the multi-modal domain metric to measure the video-level discrepancy across domains. [18] proposed cross-modal and cross-domain contrastive losses to handle feature spaces across modalities and domains. [43] leveraged both cross-modal complementary and cross-modal consensus to learn the most transferable features through a CIA framework. In [12], the authors proposed to generate noisy pseudo-labels for the target domain data using the source pre-trained model and select the clean samples in order to increase the quality of the pseudo-labels. Lin et al. [23] developed a cycle-based approach that alternates between spatial and spatiotemporal learning with knowledge transfer. Generally, all the above methods utilize the flow as the auxiliary modality input. Recently, there are also methods exploring other modalities, for instance, A3R [48] with audios and MixDANN [44] with wild data. It is worth noting that the proposed TranSVAE – although only uses single modality RGB features – surprisingly achieves better UDA performance compared with most current state-of-the-art multi-modal methods, which highlights our superiority.

Disentanglement. Feature disentanglement is a wide and hot research topic. We only focus on works that are closely related to ours. In the image domain, some works consider adaptation from a generative view. [4] learned a disentangled semantic representation across domains. A similar idea is then applied to graph domain adaptation [5] and domain generalization [16]. [13] proposed a novel informative feature disentanglement, equipped with the adversarial network or the metric discrepancy model. Another disentanglement-related topic is sequential data generation. To generate videos, existing works [22, 50, 1] extended VAE to a recurrent form with different recursive structures. In
this paper, we present a VAE-based structure to generate cross-domain videos. We aim at tackling video-based UDA from a new perspective: sequential domain disentanglement and transfer.

3 Technical Approach

Formally, for a typical video-based UDA problem, we have a source domain \( S \) and a target domain \( T \). Domain \( S \) contains sufficient labeled data, denoted as \( \{(V_i^S, y_i^S)\}_{i=1}^{N_S} \), where \( V_i^S \) is a video sequence and \( y_i^S \) is the class label. Domain \( T \) consists of unlabeled data, denoted as \( \{(V_i^T)_{i=1}^{N_T}\} \). For a sequence \( V_i^D \) from domain \( D \in \{S, T\} \), it contains \( T \) frames \( \{x_{i,t}^D\}_{t=1}^{T} \) in total\(^{2}\). We further denote \( N = N_S + N_T \). Domains \( S \) and \( T \) are of different distributions but share the same label space. The objective is to utilize both \( \{(V_i^S, y_i^S)\}_{i=1}^{N_S} \) and \( \{(V_i^T)_{i=1}^{N_T}\} \) to train a good classifier for domain \( T \). We present a table to list the notations in the Appendix.

Framework Overview. We adopt a VAE-based structure to model the cross-domain video generation process and propose to regulate the two sets of latent factors for adaptation purposes. The reconstruction process is based on the two sets of latent factors, i.e., \( z_{1,t}^S, ... , z_{T,t}^S \) and \( z_{1,t}^T, ... , z_{T,t}^T \) that are sampled from the posteriors \( q(z_{i,t}^S|\bar{x}_{i,t}^S) \) and \( q(z_{i,t}^T|\bar{x}_{i,t}^T) \), respectively. The overall architecture consists of five segments including the encoder, LSTM, latent spaces, sampling, and decoder, as shown in Fig. 3. The vanilla structure only gives an arbitrary disentanglement of latent factors. To make the disentanglement facilitate adaptation, we carefully constrain the latent factors as follows.

Video Sequence Reconstruction. The overall architecture of TranSVAE follows a VAE-based structure [22, 50, 1] with two sets of latent factors \( z_{1,t}^P, ... , z_{T,t}^P \) and \( z_{1,t}^D \). The generative and inference graphical models are presented in Fig. 2. Similar to the conventional VAE, we use a standard Gaussian distribution for static latent factors. For dynamic ones, we use a sequential prior \( z_{1,t}^P|z_{t-1}^D \sim \mathcal{N}(\mu_t, \text{diag}(\sigma_t^2)) \), that is, the prior distribution of the current dynamic factor is conditioned on the historical dynamic factors. The distribution parameters can be re-parameterized as a recurrent network, e.g., LSTM, with all previous dynamic latent factors as the input. Denoting \( Z^D = \{z_{1,t}^D, ... , z_{T,t}^D\} \) for simplification, we then get the prior as follows:

\[
p(z_{1,t}^P, Z^D) = p(z_{1,t}^D) \prod_{t=1}^{T} p(z_{t,t}^D|z_{t-1}^D). \tag{1}
\]

[^2]: Without confusion, we omit the subscript \( i \) for \( V^D \) and the corresponding notations, e.g., \( \{x_{i,t}^P, ... , x_{T,t}^D\} \).
Following Fig. 2 (the 1st subfigure), \(x^D_t\) is generated from \(z^D_t\) and \(z^P_t\), and we thus model 
\[p(x^D_t|z^D_t, z^P_t) = \mathcal{N}(\mu^d_t, \text{diag}(\sigma^d_t)).\]

The distribution parameters are re-parameterized by the decoder which can be flexible networks like the deconvolutional neural network. Using \(V^D = \{x^D_1, \ldots, x^D_T\}\), the generation can be formulated as follows:

\[p(V^D) = p(z^D_1) \prod_{t=1}^T p(x^D_t|z^D_t, z^P_t)p(z^P_t|z^D_{<t}). \quad (2)\]

Following Fig. 2 (the 2nd subfigure), we model the posterior distributions of the latent factors as 
another two Gaussian distributions, i.e., 
\[q(z^D_t|V^D) = \mathcal{N}(\mu^d_t, \text{diag}(\sigma^d_t))\]
and 
\[q(z^P_t|x^P_t) = \mathcal{N}(\mu^p_t, \text{diag}(\sigma^p_t)).\]

The parameters of these two distributions are re-parameterized by the encoder, which can be a convolutional or LSTM module. However, the network of the static latent factors uses the whole sequence as the input while that of the dynamic latent factors only uses previous frames. Then the inference can be factorized as:

\[q(z^D_t, Z^P|V^D) = q(z^D_t|V^D) \prod_{t=1}^T q(z^P_t|x^P_t). \quad (3)\]

Combining the above generation and inference, we obtain the VAE-related objective function as:

\[\mathcal{L}_{\text{svae}} = \mathbb{E}_{q(z^D_t, Z^P|V^D)}[-\sum_{t=1}^T \log p(x^D_t|z^D_t, z^P_t)] + \sum_{t=1}^T \text{KL}(q(z^D_t|V^D)||p(z^D_t)) + \sum_{t=1}^T \text{KL}(q(z^P_t|x^P_t)||p(z^P_t|z^D_{<t})). \quad (4)\]

which is a frame-wise negative variational lower bound. Only using the above vanilla VAE-based loss cannot guarantee that the disentanglement serves for adaptation, and thus we propose additional constraints on the two sets of latent factors.

**Mutual Dependence Minimization** (Fig. 3, □). We first consider explicitly enforcing the two sets of latent factors to be mutually independent. To do so, we introduce the mutual information [2] loss \(L_{\text{mi}}\) to regulate the two sets of latent factors. Thus, we obtain:

\[L_{\text{mi}}(z^D_t, Z^P) = \sum_{t=1}^T \text{KL}(q(z^D_t, z^P_t)||q(z^D_t)q(z^P_t)) = \sum_{t=1}^T [H(z^D_t) + H(z^P_t) - H(z^D_t, z^P_t)]. \quad (5)\]

To calculate Eq. (5), we need to estimate the densities of \(z^D_t, z^P_t\) and \(z^D_t, z^P_t\). Following the non-parametric way in [9], we use the mini-batch weighted sampling as follows:

\[H(z) = -\mathbb{E}_{q(z)}[\log q(z)] \approx -\log \frac{1}{M} \sum_{i=1}^M \log \frac{1}{MN} \sum_{j=1}^M q(z(x_i)|x_j)], \quad (6)\]

where \(z\) is \(z^D_t, z^P_t\) or \(z^D_t, z^P_t\), \(N\) denotes the data size and \(M\) is the mini-batch size.

**Domain Specificity & Static Consistency** (Fig. 3, ■). A characteristic of the domain specificity, e.g., the video style or the objective appearance, is its static consistency over dynamic frames. With this observation, we enable the static and domain-specific latent factors so that we can remove the spatial divergence by disentangling them out. Mathematically, we hope that \(z^D_t\) does not change a lot when \(z^D_t\) varies over time. To achieve this, given a sequence, we randomly shuffle the temporal order of frames to form a new sequence. The static latent factors disentangled from the original sequence and the shuffled sequence should be ideally equal or be very close at least. This motivates us to minimize the distance between these two static factors. Meanwhile, to further enhance the domain specificity, we enforce the dynamic latent factors from different domains to have a large distance. To this end, we propose the following contrastive triplet loss:

\[L_{\text{ctc}} = \max \left( D(z^D_t^+, z^P_t^+) - D(z^D_t^+, z^P_t^-) + m, 0 \right), \quad (7)\]

where \(D(\cdot, \cdot)\) is Euclidean distance. \(m\) is a margin set to 1 in the experiments, \(z^D_t^+, z^P_t^+\), and \(z^D_t^-, z^P_t^-\) are static latent factors of the anchor sequence from domain \(D^+\), the shuffled sequence, and a randomly selected sequence from domain \(D^-\), respectively. \(D^+\) and \(D^-\) represent two different domains.
Domain Temporal Alignment (Fig. 3, □). We now consider to reduce the temporal divergence of the dynamic latent factors. There are several ways to achieve this, and in this paper, we take advantage of the most popular adversarial-based idea [14]. Specifically, we build a domain classifier to discriminate whether the data is from $S$ or $T$. When back-propagating the gradients, a gradient reversal layer (GRL) is adopted to invert the gradients. Like existing video-based UDA methods, we also conduct both frame-level and video-level alignments. Moreover, as TA$^3$N [7] does, we exploit the temporal relation network (TRN) [49] to discover the temporal relations among different frames, and then aggregate all the temporal relation features into the final video-level features. This enables another level of alignment on the temporal relation features. Thus, we have:

$$
L_f = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T} \sum_{t=1}^{T} CE \left[ G_f (Z^D_i) , d_i \right],
$$

(8)

$$
L_r = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T-1} \sum_{t=2}^{T} CE \left[ G_r (TrN_i(Z^D_i)) , d_i \right],
$$

(9)

$$
L_v = \frac{1}{N} \sum_{i=1}^{N} CE \left[ G_v \left( \frac{1}{T-1} \sum_{n=2}^{T} TrN_i(Z^D_i) \right) , d_i \right],
$$

(10)

where $d_i$ is the domain label, $CE$ denotes the cross-entropy loss function, $Z^D = \{ Z^D_1, ..., Z^D_T \}$, $TrN_i$ is the $n$-frame temporal relation network, $G_f$, $G_r$, and $G_v$ are the frame feature level, the temporal relation feature level, and the video feature level domain classifiers, respectively. To this end, we obtain the domain adversarial loss by summing up Eqs. (8-10):

$$
L_{adv} = L_f + L_r + L_v.
$$

(11)

We assign equal importance to these three levels of losses to reduce the overhead of the hyperparameter search. To this end, with $L_{mi}$, $L_{ac}$, and $L_{adv}$, the learned dynamic latent factors are expected to be domain-invariant (the three constraints are interactive and complementary to each other for obtaining the domain-invariant dynamic latent factor), and then can be used for downstream UDA tasks. In this paper, we specifically focus on action recognition as the downstream task.

Task Specific Supervision (Fig. 3, ■). We further encourage the dynamic latent factors to carry the semantic information. Considering that the source domain has sufficient labels, we accordingly design the task supervision as the regularization imposed on $Z^S$. This gives us:

$$
L_{cls} = \frac{1}{N^S} \sum_{i=1}^{N^S} L (\mathcal{F}(Z^S_i), y^S_i),
$$

(12)

where $\mathcal{F}(\cdot)$ is a feature transformer mapping the frame-level features to video-level features, specifically a TRN in this paper, and $L(\cdot, \cdot)$ is either cross-entropy or mean squared error loss according to the targeted task.

Although the dynamic latent factors are constrained to be domain-invariant, we do not completely rely on source semantics to learn features discriminative for the target domain. We propose to incorporate target pseudo-labels in task-specific supervision. During the training, we use the prediction network obtained in the previous epoch to generate the target pseudo-labels of the unlabelled target training data for the current epoch. However, to increase the reliability of target pseudo-labels, we let the prediction network be trained only on the source supervision for several epochs and then integrate the target pseudo-labels in the following training epochs. Meanwhile, a confidence threshold is set to determine whether to use the target pseudo-labels or not. Thus, we have the final task-specific supervision as follows:

$$
L_{cls} = \frac{1}{N} \left( \sum_{i=1}^{N^S} L(\mathcal{F}(Z^S_i), y^S_i) + \sum_{i=1}^{N^T} L(\mathcal{F}(Z^T_i), \tilde{y}^T_i) \right),
$$

(13)

where $\tilde{y}^T_i$ is the pseudo-label of $Z^T_i$.

Summary. To this end, we reach the final objective function of our TransVAE framework as follows:

$$
L' = L_{svae} + \lambda_1 L_{mi} + \lambda_2 L_{adv} + \lambda_3 L_{cls} + \lambda_4 L_{cls},
$$

(14)

where $\lambda_i$ with $i = 1, 2, 3, 4$ denotes the loss balancing weight.
4 Experiments

In this section, we conduct extensive experimental studies on popular video-based UDA benchmarks to verify the effectiveness of the proposed TranSVAE framework.

4.1 Datasets

UCF-HMDB is constructed by collecting the relevant and overlapping action classes from UCF101 [35] and HMDB51 [20]. It contains 3,209 videos in total with 1,438 training videos and 571 validation videos from UCF101, and 840 training videos and 360 validation videos from HMDB51. This in turn establishes two video-based UDA tasks: U → H and H → U.

Jester [27] consists of 148,092 videos of humans performing hand gestures. Pan et al. [29] constructed a large-scale cross-domain benchmark with seven gesture classes, and form a single transfer task \( J_S \rightarrow J_T \), where \( J_S \) and \( J_T \) contain 51,498 and 51,415 video clips, respectively.

Epic-Kitchens [11] is a challenging egocentric dataset consisting of videos capturing daily activities in kitchens. Pan et al. [28] constructs three domains across the eight largest actions. They are \( D_1 \), \( D_2 \), and \( D_3 \) corresponding to P08, P01, and P22 kitchens of the full dataset, resulting in six cross-domain tasks.

Sprites [22] contains sequences of animated cartoon characters with 15 action categories. The appearances of characters are fully controlled by four attributes, \( i.e. \), body, top wear, bottom wear, and hair. We construct two domains, \( P_1 \) and \( P_2 \). \( P_1 \) uses the human body with attributes randomly selected from 3 top wears, 4 bottom wears, and 5 hairs, while \( P_2 \) uses the alien body with attributes randomly selected from 4 top wears, 3 bottom wears, and 5 hairs. The attribute pools are non-overlapping across domains, resulting in completely heterogeneous \( P_1 \) and \( P_2 \). Each domain has 900 video sequences, and each sequence is with 8 frames.

4.2 Implementation Details

Architecture. Following the latest works [31, 37], we use I3D [6] as the backbone\(^3\). However, different from CoMix which jointly trains the backbone, we simply use the pretrained I3D model on Kinetics [17], provided by [6], to extract RGB features. For the first three benchmarks, RGB features are used as the input of TranSVAE. For Sprites, we use the original image as the input, for the purpose of visualizing the reconstruction and disentanglement results. We use the shared encoder and decoder structures across the source and target domains. For RGB feature inputs, the encoder and decoder are fully connected layers. For original image inputs, the encoder and decoder are the convolution and deconvolution layers (from DCGAN [46]), respectively. For the TRN model, we directly use the one provided by [7]. Other details on this aspect are placed in Appendix.

Configurations. Our TranSVAE is implemented with PyTorch [30]. We use Adam with a weight decay of \( 1 \times 10^{-4} \) as the optimizer. The learning rate is initially set to be \( 1 \times 10^{-3} \) and follows a commonly used decreasing strategy in [14]. The batch size and the learning epoch are uniformly set to be 128 and 1,000, respectively, for all the experiments. We uniformly set 100 epochs of training under only source supervision and involved the target pseudo-labels afterward. Following the common protocol in video-based UDA [37], we perform hyperparameter selection on the validation set. The specific hyperparameters used for each task can be found in the Appendix. NVIDIA A100 GPUs are used for all experiments. Kindly refer to our Appendix for all other details.

---

\(^3\)The existing widely adopted backbones for video-based UDA include ResNet-101 and I3D. However, more recent backbones, e.g. Transformer-based or VideoMAE-based [36, 40], are promising to be used in this task.
Table 2: UDA performance comparisons to approaches using multi-modality data as the input.

<table>
<thead>
<tr>
<th>Task</th>
<th>$s_{only}$</th>
<th>MM-SADA</th>
<th>STCDA</th>
<th>CMCD</th>
<th>A3R</th>
<th>CleanAdapt</th>
<th>CycDA</th>
<th>MixDANN</th>
<th>CIA</th>
<th>TranSVAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>U → H</td>
<td>86.1</td>
<td>84.2</td>
<td>83.1</td>
<td>84.7</td>
<td>/</td>
<td>89.8</td>
<td>88.1</td>
<td>82.2</td>
<td>88.3</td>
<td>87.8 (±3.7)</td>
</tr>
<tr>
<td>H → U</td>
<td>92.5</td>
<td>91.1</td>
<td>92.1</td>
<td>92.8</td>
<td>/</td>
<td>99.2</td>
<td>98.0</td>
<td>92.8</td>
<td>94.1</td>
<td>99.0 (±0.5)</td>
</tr>
<tr>
<td>Average ↑</td>
<td>89.3</td>
<td>87.7</td>
<td>87.6</td>
<td>88.8</td>
<td>/</td>
<td>94.5</td>
<td>93.1</td>
<td>87.5</td>
<td>91.2</td>
<td>93.4 (±3.1)</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_2$</td>
<td>43.2</td>
<td>49.5</td>
<td>52.0</td>
<td>50.3</td>
<td>53.2</td>
<td>52.7</td>
<td>/</td>
<td>56.0</td>
<td>52.5</td>
<td>50.5 (±3.7)</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_3$</td>
<td>42.5</td>
<td>44.1</td>
<td>45.5</td>
<td>46.3</td>
<td>52.1</td>
<td>47.0</td>
<td>/</td>
<td>47.3</td>
<td>47.8</td>
<td>50.3 (±6.8)</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_1$</td>
<td>43.0</td>
<td>48.2</td>
<td>49.0</td>
<td>49.5</td>
<td>51.9</td>
<td>46.2</td>
<td>/</td>
<td>50.3</td>
<td>49.8</td>
<td>50.3 (±1.3)</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_3$</td>
<td>48.0</td>
<td>52.7</td>
<td>52.5</td>
<td>52.0</td>
<td>55.5</td>
<td>52.7</td>
<td>/</td>
<td>52.4</td>
<td>53.2</td>
<td>58.6 (±10.6)</td>
</tr>
<tr>
<td>$D_3 \rightarrow D_1$</td>
<td>43.0</td>
<td>50.9</td>
<td>52.6</td>
<td>48.7</td>
<td>51.5</td>
<td>47.6</td>
<td>/</td>
<td>51.9</td>
<td>52.2</td>
<td>48.0 (±5.0)</td>
</tr>
<tr>
<td>$D_3 \rightarrow D_2$</td>
<td>55.5</td>
<td>56.1</td>
<td>55.6</td>
<td>56.3</td>
<td>63.2</td>
<td>54.4</td>
<td>/</td>
<td>54.7</td>
<td>57.6</td>
<td>58.0 (±2.5)</td>
</tr>
<tr>
<td>Average ↑</td>
<td>45.9</td>
<td>50.3</td>
<td>51.2</td>
<td>51.0</td>
<td>54.1</td>
<td>50.3</td>
<td>/</td>
<td>52.0</td>
<td>52.2</td>
<td>52.6 (±1.7)</td>
</tr>
</tbody>
</table>

Competitors. We compared methods from three lines. We first consider the source-only ($s_{only}$) and supervised-target ($T_{sup}$) which uses only labeled source data and only labeled target data, respectively. These two baselines serve as the lower and upper bounds for our tasks. Secondly, we consider five popular image-based UDA methods by simply ignoring temporal information, namely DANN [14], JAN [25], ADDA [38], AdaBN [21], and MCD [32]. Lastly and most importantly, we compare recent SoTA video-based UDA methods, including TA$^3$N [7], SAVA [10], TCoN [29], ABG [26], CoMix [31], CO$^2$A [37], and MA$^2$L-TD [8]. All these methods use single modality features. We directly quote numbers reported in published papers whenever possible. There exist recent works conducting video-based UDA using multi-modal data, e.g., RGB + Flow. Although TranSVAE solely uses RGB features, we still take this set of methods into account. Specifically, we consider MM-SADA [28], STCDA [34], CMCD [18], A3R [48], CleanAdapt [12], CycDA [23], MixDANN [44] and CIA [43].

4.3 Comparative Study

Results on UCF-HMDB. Tab. 1 shows comparisons of TranSVAE with baselines and SoTA methods on UCF-HMDB. The best result among all the baselines is highlighted using bold. Overall, methods using the I3D backbone [6] achieve better results than those using ResNet-101. Our TranSVAE consistently outperforms all previous methods. In particular, TranSVAE achieves 93.37% average accuracy, improving the best competitor CO$^2$A [37], with the same I3D backbone [6], by 1.38%. Surprisingly, we observe that TranSVAE even yields better results (by a 2.1% improvement) than the supervised-target ($T_{sup}$) baseline. This is because the H → U task already has a good performance without adaptation, i.e., 88.79% for the source-only ($s_{only}$) baseline, and thus the target pseudo-labels used in TranSVAE are almost correct. By further aligning domains and equivalently augmenting training data, TranSVAE outperforms $T_{sup}$ which is only trained with target data.

Results on Jester & Epic-Kitchens. Tab. 3 shows the comparison results on the Jester and Epic-Kitchens benchmarks. We can see that our TranSVAE is the clear winner among all the methods on all the tasks. Specifically, TranSVAE achieves a 1.4% improvement and a 9.4% average improvement over the runner-up baseline CoMix [31] on Jester and Epic-Kitchens, respectively. This verifies the superiority of TranSVAE over others in handling video-based UDA. However, we also notice that the accuracy gap between CoMix and $T_{sup}$ is still significant on Jester. This is because the large-scale Jester dataset contains highly heterogeneous data across domains, e.g., the source domain contains videos of the rolling hand forward, while the target domain only consists of videos of the rolling hand backward. This leaves much room for improvement in the future.

Compare to Multi-Modal Methods. We further compare with four recent video-based UDA methods that use multi-modalities, e.g., RGB features, and optical flows, although TranSVAE only uses RGB features. Surprisingly, TranSVAE achieves better average results than seven out of eight multi-modal methods and is only worse than CleanAdapt [12] on UCF-HMDB and A3R [48] on Epic-Kitchens. Considering TranSVAE only uses single-modality data, we are confident that there exists great potential for further improvements of TranSVAE with multi-modal data taken into account.

Table 3: Comparison results on Jester and Epic-Kitchens.

<table>
<thead>
<tr>
<th>Task</th>
<th>$s_{only}$</th>
<th>DANN</th>
<th>ADDA</th>
<th>TA$^3$N</th>
<th>CoMix</th>
<th>TranSVAE</th>
<th>$T_{sup}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_0 \rightarrow J_F$</td>
<td>51.5</td>
<td>55.4</td>
<td>52.3</td>
<td>55.5</td>
<td>64.7</td>
<td>66.1 (±14.6)</td>
<td>95.6</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_2$</td>
<td>32.8</td>
<td>37.7</td>
<td>35.4</td>
<td>34.2</td>
<td>42.9</td>
<td>50.5 (±19.7)</td>
<td>64.0</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_3$</td>
<td>34.1</td>
<td>36.6</td>
<td>34.9</td>
<td>37.4</td>
<td>40.9</td>
<td>50.3 (±16.2)</td>
<td>63.7</td>
</tr>
<tr>
<td>$D_1 \rightarrow D_3$</td>
<td>35.4</td>
<td>38.3</td>
<td>36.3</td>
<td>40.9</td>
<td>38.6</td>
<td>50.3 (±14.6)</td>
<td>57.0</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_1$</td>
<td>39.1</td>
<td>41.9</td>
<td>40.8</td>
<td>42.8</td>
<td>45.2</td>
<td>58.6 (±14.5)</td>
<td>64.7</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_3$</td>
<td>34.6</td>
<td>38.8</td>
<td>36.1</td>
<td>39.9</td>
<td>42.3</td>
<td>48.0 (±13.4)</td>
<td>57.0</td>
</tr>
<tr>
<td>$D_2 \rightarrow D_3$</td>
<td>35.8</td>
<td>42.1</td>
<td>41.4</td>
<td>44.2</td>
<td>49.2</td>
<td>58.0 (±22.2)</td>
<td>64.0</td>
</tr>
<tr>
<td>Average ↑</td>
<td>35.1</td>
<td>39.2</td>
<td>37.4</td>
<td>39.9</td>
<td>43.2</td>
<td>52.6 (±17.0)</td>
<td>61.5</td>
</tr>
</tbody>
</table>
We further conduct another ablation study by sequentially integrating the respective losses. Table 5 shows the ablation results on UCF-HMDB, Jester, and Epic-Kitchens. As can be seen, removing each loss sequentially requires less trainable parameters than CO with existing baselines in terms of FLOPs, and inference frame-per-second (FPS).

### 4.4 Property Analysis

**Disentanglement Analysis.** We analyze the disentanglement effect of TranSVAE on Sprites [22] and show results in Fig. 4. The left subfigure shows the original sequences of the two domains. The “Human” and “Alien” are completely different appearances and the former is casting spells while the latter is slashing. The middle subfigure shows the sequences reconstructed only using \{z^P_1, \ldots, z^P_d\}. It can be clearly seen that the two sequences keep the same action as the corresponding original ones. However, if we only focus on the appearance characteristics, it is difficult to distinguish the domain to which the sequences belong. This indicates that \{z^P_1, \ldots, z^P_d\} are domain-invariant and well encode the semantic information. The right subfigure shows the sequences reconstructed by exchanging \(z_d^P\), which results in two sequences with the same actions but exchanged appearance. This verifies that \(z_d^P\) is representing the appearance information, which is actually the domain-related information in this example. This property study sufficiently supports that TranSVAE can successfully disentangle the domain information from other information, with the former embedded in \(z_d^P\) and the latter embedded in \{z^P_1, \ldots, z^P_{d-1}\}.

**Complexity Analysis.** We further conduct complexity analysis on our TranSVAE. Specifically, we compare the number of trainable parameters, multiply-accumulate operations (MACs), floating-point operations (FLOPs), and inference frame-per-second (FPS) with existing baselines including TA^N [7], CO^2A [37], and CoMix [31]. All the comparison results are shown in Tab. 4. From the table, we observe that TranSVAE requires less trainable parameters than CO^2A and CoMix. Although more trainable parameters are used than TA^N, TranSVAE achieves significant adaptation performance improvement than TA^N (see Tab. 1 and Tab. 3). Moreover, the MACs, FLOPs, and FPS are competitive among different methods. This is reasonable since all these approaches adopt the same I3D backbone.

**Ablation Study.** We now analyze the effectiveness of each loss term in Eq. (14). We compare with four variants of TranSVAE, each removing one loss term by equivalently setting the weight \(\lambda\) to 0. The ablation results on UCF-HMDB, Jester, and Epic-Kitchens are shown in Tab. 5. As can be seen, removing \(L_{cbs}\) significantly reduces the transfer performance in all the tasks. This is reasonable as \(L_{cbs}\) is used to discover the discriminative features. Removing any of \(L_{adv}, L_{mi},\) and \(L_{ctc}\) leads to an inferior result than the full TranSVAE setup, and removing \(L_{adv}\) is the most influential. This is because \(L_{adv}\) is used to explicitly reduce the temporal domain gaps. All these results demonstrate that every proposed loss matters in our framework.

We further conduct another ablation study by sequentially integrating \(L_{cbs}, L_{adv}, L_{mi},\) and \(L_{ctc}\) into our sequential VAE structure using UCF-HMDB. We use this integration order based on the average positive improvement that a loss brings to TranSVAE as shown in Tab. 5. We also take advantage of t-SNE [39] to visualize the features learned by these different variants. We plot two sets of t-SNE
We further discuss the limitations of the TranSVAE framework. These are also promising future directions. Firstly, the empirical evaluation is mainly on the action recognition task. The performance of TranSVAE on other video-related tasks, e.g., video segmentation, is not tested. Secondly, TranSVAE is only evaluated on the typical two-domain transfer scenario. The multi-source transfer case is not considered but is worthy of further study. Thirdly, although TranSVAE exhibits better performance than multi-modal transfer methods, its current version does not consider multi-modal data functionally and structurally. An improved TranSVAE with the capacity of using multi-modal data is expected to further boost the adaptation performance. Fourthly, current empirical evaluations are mainly based on the I3D backbone, more advanced backbones, e.g., [36, 40], are expected to be explored for further improvement. Lastly, TranSVAE handles the spatial divergence by disentangling the static domain-specific latent factors out. However, it may happen that spatial divergence is not completely captured by the static latent factors due to insufficient disentanglement.
References


[29] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In AAAI Conference on Artificial Intelligence, pages 11815–11822, 2020. 2, 3, 7, 8


[34] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9787–9795, 2021. 3, 8


[38] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7167–7176, 2017. 1, 8


[40] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14549–14560, 2023. 7, 10


Appendix

In this appendix, we supplement the following materials to support the findings and conclusions drawn in the main body of this paper:

- Sec. A provides additional implementation details to assist reproductions.
- Sec. B contains additional comparative and ablation results of the TranSVAE framework.
- Sec. C elaborates on the broader impact of this work.
- Sec. D acknowledges the public resources used during the course of this work.

A Additional Implementation Details

In this section, we provide more implementation details including the I3D feature extraction procedure, the concrete model architecture, and the hyperparameter selection in our proposed TranSVAE framework. We also provide detailed instructions for our live demo.

A.1 Notations

We present Tab. A to summarize the notations used in this paper.

A.2 I3D Feature Extractions

We extract the I3D RGB features following the routine described in SAVA [10]. Given a video sequence, 16 frames along clips are sampled by sliding a temporal window with a temporal stride of 1. Specifically, for each frame in the video, the temporal window consists of its previous seven frames and the following eight frames. Zero padding is used for the beginning and the end of the video. We then feed the sliding windows to the I3D backbone to extract features, which results in a 1024-dimensional feature vector for each frame of the video.

A.3 Model Architecture

We now provide the detailed model architecture of our TranSVAE. In Fig. A, we show the model with the image as the input, where the encoder and decoder are more complex convolutional and deconvolutional layers. For the model with the RGB features as the input, we can simply replace the encoder and decoder with fully connected linear layers. Note that the dimensionality of all the modules shown above is uniformly applied in all the experiments.

A.4 Hyperparameter Selection

There are several hyperparameters used in TranSVAE, including the balancing weights $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$, the number of the video frames $T$, and the confidence threshold $\eta$ for generating target pseudo-labels. For $\lambda_1$ to $\lambda_4$, we select from the value set $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 0.5, 1, 5, 10, 50, 100, 1000\}$. For $T$, we select from $\{5, 6, 7, 8, 9, 10\}$. For $\eta$, we set its value range from 0.9 to 1.0 with a step of 0.01.

We set a high-value range of $\eta$ to ensure a high confidence score on the correctness of the target pseudo-labels. Following the common protocol used in video-based UDA, we conduct an extensive grid search regarding these hyperparameters on the validation set of each transfer task. Tab. B summarizes the exact used values of these hyperparameters for the UCF-HMDB [35, 20], Jester [27], and Epic-Kitchens [11] UDA benchmarks. For the Sprites [22] dataset, we do not do a hyperparameter search as the data is quite simple. We simply set $T$ as 8, which is the original length of the video sequence. The confidence threshold is set to be 0.99, and $\lambda_1$ to $\lambda_4$ are all set to be 1.

A.5 Demo Instruction

As mentioned in the main body, we include a live demo for our TranSVAE framework. This demo can be accessed at: https://huggingface.co/spaces/ldkong/TranSVAE. Here we include the detailed instructions for playing with this demo.
This demo is built upon Hugging Face Spaces\(^4\), which provides concise and easy-to-use live demo interfaces. Our demo consists of one input interface and one output interface as shown in Fig. B. Specifically, the appearances of the Sprites avatars are fully controlled by four attributes, i.e., body, hair color, top wear, and bottom wear. We construct two domains, \(P_1\) and \(P_2\). \(P_1\) uses the “Human” body while \(P_1\) uses the “Alien” body. The attribute pools of “Human” and “Alien” are non-overlapping across domains, resulting in completely heterogeneous \(P_1\) and \(P_2\). Each video sequence contains eight frames in total.

For conducting domain disentanglement and transfer with TranSVAE, users are free to choose the action and the appearance of the avatars on the left-hand side of the interface. Next, simply click the “Submit” button and the adaptation results will display on the right-hand side of the interface in a few seconds. The outputs include:

- The 1st column: The original input of the “Human” and “Alien” avatars, i.e., \(\{x_{P_1}^1, ..., x_{P_1}^8\}\) and \(\{x_{P_2}^1, ..., x_{P_2}^8\}\);
- The 2nd column: The reconstructed “Human” and “Alien” avatars \(\{\tilde{x}_{P_1}^1, ..., \tilde{x}_{P_1}^8\}\) and \(\{\tilde{x}_{P_2}^1, ..., \tilde{x}_{P_2}^8\}\);
- The 3rd column: The reconstructed “Human” and “Alien” avatars using only \(\{z_{D}^1, ..., z_{D}^8\}\), \(D \in \{P_1, P_2\}\), which are domain-invariant;

\(^4\)https://huggingface.co
Figure A: The neural network structure (left) and a Netron graph (right) of the proposed TransSVAE framework. Zoom-ed in for the details.
Figure B: The input (left) and output (right) interfaces of our live demo. Users are free to customize the actions and appearances of the source and target inputs (i.e., the “Human” and “Alien” avatars) in the left-hand side and use them for disentanglement and transfer as shown in the right-hand side.

Table C: Ablation results without and with disentanglement term.

<table>
<thead>
<tr>
<th>$\mathcal{L}_{\text{svae}}$</th>
<th>$\mathcal{L}_{\text{cls}}$</th>
<th>$\mathcal{L}_{\text{adv}}$</th>
<th>$\mathcal{L}_{\text{mi}}$</th>
<th>$\mathcal{L}_{\text{ctc}}$</th>
<th>$U \rightarrow H$</th>
<th>$H \rightarrow U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>81.67</td>
<td>90.54</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>79.44</td>
<td>90.37</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>81.11</td>
<td>90.37</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>82.22</td>
<td>91.07</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>84.44</td>
<td>93.52</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>83.06</td>
<td>93.70</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>83.61</td>
<td>91.42</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td><strong>87.78</strong></td>
<td><strong>98.95</strong></td>
</tr>
</tbody>
</table>

- The 4th column: The reconstructed “Human” and “Alien” avatars by exchanging $z_D$, which results in two sequences with the same actions but exchanged appearance, i.e., domain disentanglement and transfer.

B Additional Experimental Results

In this section, we provide additional quantitative and qualitative results for our TranSVAE framework.

B.1 Additional Ablation Studies

Component Integration Study. We further analyze the effect of each loss term without disentanglement, i.e., not adding $\mathcal{L}_{\text{svae}}$, on the UCF-HMDB dataset, as shown in Tab. C. Without $\mathcal{L}_{\text{svae}}$, the framework is to learn representations with corresponding constraints as many existing video-based UDA methods do. Precisely, we have the following remarks for the top-half table:

- Only with $\mathcal{L}_{\text{adv}}$, the framework is equivalent to TA$_3$N [7], which learns the domain-invariant temporal representation. We can see the adaptation results are similar to that of [7].
- $\mathcal{L}_{\text{mi}}$ is a term specifically designed for disentanglement purposes. Thus, the effect of using $\mathcal{L}_{\text{mi}}$ alone for adaptation is random as verified in the above table, achieving worse results.
in the U → H task and better results in the H → U tasks compared with the source-only baseline.

• $L_{ctc}$ enforces the learned static representation to be domain-specific and brings indirect benefit to adaptation, thus obtaining slightly better results than the source-only baseline.

In sum, without $L_{svae}$, the adaptation performance of each loss term alone is far from optimal. Moreover, combining all the terms can slightly improve the adaptation performance over the individual ones, however, is still obviously worse than TranSVAE results. The above ablation study shows the necessity of the disentanglement in the TranSVAE framework and proves that $L_{svae}$ is an indispensable loss term.

Regarding the bottom half table showing the results with $L_{adv}$, we have the following remarks:

• Each individual term with $L_{adv}$ achieves better adaptation results than the source-only baseline.

• Each individual term with $L_{adv}$ consistently outperforms the corresponding ones without $L_{adv}$.

• The final TranSVAE framework combined all the loss terms yields much better results than each individual case.

This set of ablation studies indicates that 1) each term in TranSVAE brings positive effects to adaptation, and 2) disentanglement and adaptation are interactive and complementary in our framework to obtain the best possible UDA results.

**Number of Frames $T$.** Tab. D shows the transfer performance with the variation of the number of frames $T$ on the UCF-HMDB dataset. As can be seen, the two tasks achieve the optimal performance with different $T$, specifically $T = 8$ for U → H and $T = 9$ for H → U. Based on this observation, we apply a grid search on the validation set to obtain the optimal $T$ for each task in our experiments.

**Target Pseudo-Label Threshold $\eta$.** We show the sensitivity analyses of the transfer performance with respect to the target pseudo label threshold $\eta$ on the UCF-HMDB dataset in Tab. E. The results show that different tasks yield the best transfer result with different $\eta$, specifically $\eta = 0.93$ for U → H and $\eta = 0.96$ for H → U. Thus, we also apply the grid search on the validation set to obtain the optimal $\eta$ for each task.

### B.2 Additional Qualitative Results

For those who cannot access our live demo, we have included more qualitative examples for domain disentanglement and transfer in Fig. C. We also provide a link for GIFs demonstrating various disentanglement and reconstruction results.

In the GIFs link, we show two cases, the first two columns for the first one and the last two columns for the second one. Each case contains a source and a target cartoon character performing an action. For each row, we have the following remark:

5https://github.com/ldkong1205/TranSVAE#ablation-study

---

**Table D: Ablation results for the frame number $T$.**

<table>
<thead>
<tr>
<th>Task</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>U → H</td>
<td>84.44</td>
<td>86.11</td>
<td>84.17</td>
<td><strong>87.78</strong></td>
<td>84.72</td>
<td>85.28</td>
<td>83.89</td>
<td>84.44</td>
<td>82.78</td>
<td>85.00</td>
</tr>
<tr>
<td>H → U</td>
<td>96.85</td>
<td>94.22</td>
<td>98.42</td>
<td>94.75</td>
<td><strong>98.95</strong></td>
<td>93.70</td>
<td>93.87</td>
<td>94.40</td>
<td>94.05</td>
<td>94.40</td>
</tr>
</tbody>
</table>

**Table E: Ablation results for the pseudo-label threshold $\eta$.**

<table>
<thead>
<tr>
<th>Task</th>
<th>w/o</th>
<th>0.90</th>
<th>0.91</th>
<th>0.92</th>
<th>0.93</th>
<th>0.94</th>
<th>0.95</th>
<th>0.96</th>
<th>0.97</th>
<th>0.98</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>U → H</td>
<td>87.22</td>
<td>86.94</td>
<td>87.22</td>
<td>87.50</td>
<td><strong>87.78</strong></td>
<td>86.11</td>
<td>86.67</td>
<td>86.94</td>
<td>86.39</td>
<td>86.11</td>
<td>86.39</td>
</tr>
<tr>
<td>H → U</td>
<td>94.40</td>
<td>93.45</td>
<td>98.42</td>
<td>97.37</td>
<td>98.77</td>
<td>98.60</td>
<td>98.25</td>
<td>98.25</td>
<td><strong>98.95</strong></td>
<td>97.90</td>
<td>97.72</td>
</tr>
</tbody>
</table>
• The second row shows the reconstructed results of TranSVAE, i.e. $z^D_d$ and $z^D_t$. As can be seen, TranSVAE reconstructs the image with a high quality.

• The third row shows the reconstructed results only using the static latent factors, i.e., $z^D_d$ and $0^D_t$, where we replace $z^D_t$ with zero vectors. As can be seen, the reconstructed results are basically static containing the appearance of the character which is the main domain difference in the Sprites dataset [22]. Specifically, we find they generally lack arms. This is reasonable as the target action is slashing or spelling cards with arms moving, and such dynamic information on the arm is captured by $z^D_t$.

• The fourth row shows the reconstructed results only using the dynamic latent factors, i.e., $0^D_d$ and $z^D_t$, where we replace $z^D_d$ with zero vectors. As can be seen, the reconstructed results are performing the right action but the appearance is mixed up. This shows that $z^D_t$ are indeed domain-invariant and contain the semantic information.

• The last row shows the reconstructed results of exchanging the dynamic latent factors between domains, i.e., $(0^S_d, z^T_t)$ and $(0^T_d, z^S_t)$. As can be seen, the reconstructed results are with the original appearance but perform the transferred action. This indicates the potential of TranSVAE for some style-transfer tasks.

C Broader Impact

This paper provides a novel transfer method to use cross-domain video data, which effectively helps reduce the annotation efforts in related video applications. Although the main empirical evaluation is on the video action recognition task, the model structure proposed in this paper is also applicable to other video-related tasks, such as action segmentation, video semantic segmentation, etc. More generally, the idea of disentangling domain information sheds light on other data modality style transfer tasks, e.g., voice conversion. The negative impacts of this work are difficult to predict. However, as a deep model, our method shares some common pitfalls of the standard deep learning models, e.g., demand for powerful computing resources, and vulnerability to adversarial attacks.
D  Public Resources Used

We acknowledge the use of the following public resources, during the course of this work:

- UCF101\textsuperscript{6} .................................................. Unknown
- HMDB51\textsuperscript{7} .................................................. CC BY 4.0
- Jester\textsuperscript{8} .................................................. Unknown
- Epic-Kitchens\textsuperscript{9} ................................................. CC BY-NC 4.0
- Sprites\textsuperscript{10} .................................................. CC-BY-SA-3.0
- I3D\textsuperscript{11} .................................................. Apache License 2.0
- TRN\textsuperscript{12} .................................................. BSD 2-Clause License
- CoMix\textsuperscript{13} .................................................. Apache License 2.0
- TA\textsuperscript{3N}\textsuperscript{14} ............................................. MIT License
- CO\textsuperscript{2A}\textsuperscript{15} ............................................. Unknown
- PyTorch\textsuperscript{16} .................................................. PyTorch Custom License
- Netron\textsuperscript{17} .................................................. MIT License
- flops-counter.pytorch\textsuperscript{18} ........................................... MIT License

\textsuperscript{6}https://www.crcv.ucf.edu/data/UCF101.php
\textsuperscript{7}https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database
\textsuperscript{8}https://2bn.com/datasets/jester
\textsuperscript{9}https://epic-kitchens.github.io/2021
\textsuperscript{10}https://github.com/YingzhenLi/Sprites
\textsuperscript{11}https://github.com/piergiaj/pytorch-i3d
\textsuperscript{12}https://github.com/zhoubolei/TRN-pytorch
\textsuperscript{13}https://github.com/CVIR/CoMix
\textsuperscript{14}https://github.com/cmhungsteve/TA3N
\textsuperscript{15}https://github.com/vturrisi/CO2A
\textsuperscript{16}https://github.com/pytorch/pytorch
\textsuperscript{17}https://github.com/lutzroeder/netron
\textsuperscript{18}https://github.com/sovrasov/flops-counter.pytorch