Supplementary Materials: Unsupervised Video Domain Adaptation: A Disentanglement Perspective

Pengfei Wei[1], Lingdong Kong[1,2], Xinghua Qu[1], Xiang Yin[1], Zhiqiang Xu[3], Jing Jiang[4], Zejun Ma[1]

[1]ByteDance AI Lab [2]National University of Singapore [3]MBZUAI [4]University of Technology Sydney

{pengfei.wei,xinghua.qu,yinxiang.stephen,mazejun}@bytedance.com

lingdong@comp.nus.edu.sg, zhiqiang.xu@mbzuai.ac.ae, jing.jiang@uts.edu.au

In this file, we supplement more materials to support the main body of this paper. Specifically:

- Sec. A provides additional implementation details to help readers reproduce our reported results.
- Sec. B contains more ablation results and visualizations for our *TranSVAE* framework.
- Sec. C discusses the broader impact as well as the potential limitation of this work.
- Sec. D acknowledges the public resources used during the course of this work.

A 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 anonymous live demo[1].

A.1 I3D Feature Extractions

We extract the I3D RGB features following the routine described in SAVA (Choi et al. 2020). 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 7 frames and the following 8 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 1,024-dimensional feature vector for each frame of the video.

A.2 Model Architecture

We now provide the detailed model architecture of our *TranSVAE*. In Fig. A.1, 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.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Anonymous demo for *TranSVAE* at: https://huggingface.co/spaces/anonymous-demo/Anonymous-TranSVAE-Demo.

Table A.1: Summary of the best-possible hyperparameter values for each UDA task after extensive grid search.

Task	Backbone	T	η	λ_1	λ_2	λ_3	λ_4
$\mathbf{U} \rightarrow \mathbf{H}$	I3D	8	0.93	50	1	1	1
$\mathbf{H} \rightarrow \mathbf{U}$	I3D	9	0.96	0.5	0.1	10	1
$\mathbf{J}_{\mathcal{S}} \rightarrow \mathbf{J}_{\mathcal{T}}$	I3D	6	0.95	0.001	10	100	10
$\mathbf{D}_1 \rightarrow \mathbf{D}_2$	I3D	9	0.96	50	10	5	100
$\mathbf{D}_1 \rightarrow \mathbf{D}_3$	I3D	10	1	1	0.5	0.1	100
$\mathbf{D}_2 \rightarrow \mathbf{D}_1$	I3D	8	0.93	100	10	5	100
$\mathbf{D}_2 \rightarrow \mathbf{D}_3$	I3D	10	0.91	0.5	10	50	100
$\mathbf{D}_3 \rightarrow \mathbf{D}_1$	I3D	8	0.91	100	10	50	100
$\mathbf{D}_3 \rightarrow \mathbf{D}_2$	I3D	9	0.91	1000	1	0.1	100

A.3 Hyperparameter Selection

There are several hyperparameters used in *TranSVAE*, including the balancing weights λ_1, λ_2, λ_3, λ_4, the number of the video frames T, and the confidence threshold η for generating target pseudo-labels. For λ_1 to λ_4, we select from the value set $\{1e^{-3}, 1e^{-2}, 1e^{-1}, 0.5, 1, 5, 10, 50, 100, 1000\}$. For T, we select from $\{5, 6, 7, 8, 9, 10\}$. For η, we set its value range from 0.9 to 1.0 with a step of 0.01.

We set a high value range of η 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 a extensive grid search regarding these hyperparameters on the validation set of each transfer task. Tab. A.1 summarizes the exact used values of these hyperparameters for the UCF-HMDB (Soomro, Zamir, and Shah 2012; Kuehne et al. 2011), Jester (Materzynska et al. 2019), and Epic-Kitchens (Damen et al. 2018) UDA benchmarks. For the Sprites (Li and Mandt 2018) dataset, we do not do 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 λ_1 to λ_4 are all set to be 1.

A.4 Demo Instruction

As mentioned in the main body, we include an anonymous live demo for our *TranSVAE* framework. This demo can

```
TranSVAE(
  (encoder): encoder(
    (c1): dcgan_conv(
      (main): Sequential(
        (0): Conv2d(3, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
        (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (c2): dcgan_conv( (main): Sequential(
        (0): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (c3): dcgan_conv( (main): Sequential(
        (0): Conv2d(128, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (c4): dcgan_conv( (main): Sequential(
        (0): Conv2d(256, 512, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (c5): Sequential(
        (0): Conv2d(512, 1024, kernel_size=(4, 4), stride=(1, 1))
        (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): Tanh())
  )
  (decoder): decoder_woSkip
    (upc1): Sequential(
      (0): ConvTranspose2d(1024, 512, kernel_size=(4, 4), stride=(1, 1))
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True) )
    (upc2): dcgan_upconv( (main): Sequential(
      (0): ConvTranspose2d(512, 256, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (upc3): dcgan_upconv( (main): Sequential(
      (0): ConvTranspose2d(256, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (upc4): dcgan_upconv( (main): Sequential(
      (0): ConvTranspose2d(128, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): LeakyReLU(negative_slope=0.2, inplace=True) ) )
    (upc5): Sequential(
      (0): ConvTranspose2d(64, 3, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
      (1): Sigmoid() )
  )
  (relu): LeakyReLU(negative_slope=0.1)
  (dropout_f): Dropout(p=0.5, inplace=False)
  (dropout_v): Dropout(p=0.5, inplace=False)
  (z_prior_lstm_ly1): LSTMCell(512, 512)
  (z_prior_lstm_ly2): LSTMCell(512, 512)
  (z_prior_mean): Linear(in_features=512, out_features=512, bias=True)
  (z_prior_logvar): Linear(in_features=512, out_features=512, bias=True)
  (z_lstm): LSTM(1024, 512, batch_first=True, bidirectional=True)
  (f_mean): Linear(in_features=1024, out_features=512, bias=True)
  (f_logvar): Linear(in_features=1024, out_features=512, bias=True)
  (z_rnn): RNN(1024, 512, batch_first=True)
  (z_mean): Linear(in_features=512, out_features=512, bias=True)
  (z_logvar): Linear(in_features=512, out_features=512, bias=True)
  (fc_feature_domain_frame): Linear(in_features=512, out_features=512, bias=True)
  (fc_classifier_domain_frame): Linear(in_features=512, out_features=2, bias=True)
  (TRN)
  (bn_trn_S): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (bn_trn_T): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc_feature_domain_video): Linear(in_features=256, out_features=256, bias=True)
  (fc_classifier_domain_video): Linear(in_features=256, out_features=2, bias=True)
  (relation_domain_classifier_all): ModuleList(
    (0): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (1): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (2): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (3): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (4): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (5): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
    (6): Sequential(
      (0): Linear(in_features=256, out_features=256, bias=True)
      (1): ReLU()
      (2): Linear(in_features=256, out_features=2, bias=True) )
   )
  (pred_classifier_video): Linear(in_features=256, out_features=15, bias=True)
  (fc_feature_domain_latent): Linear(in_features=512, out_features=512, bias=True)
  (fc_classifier_doamin_latent): Linear(in_features=512, out_features=2, bias=True)
)
```

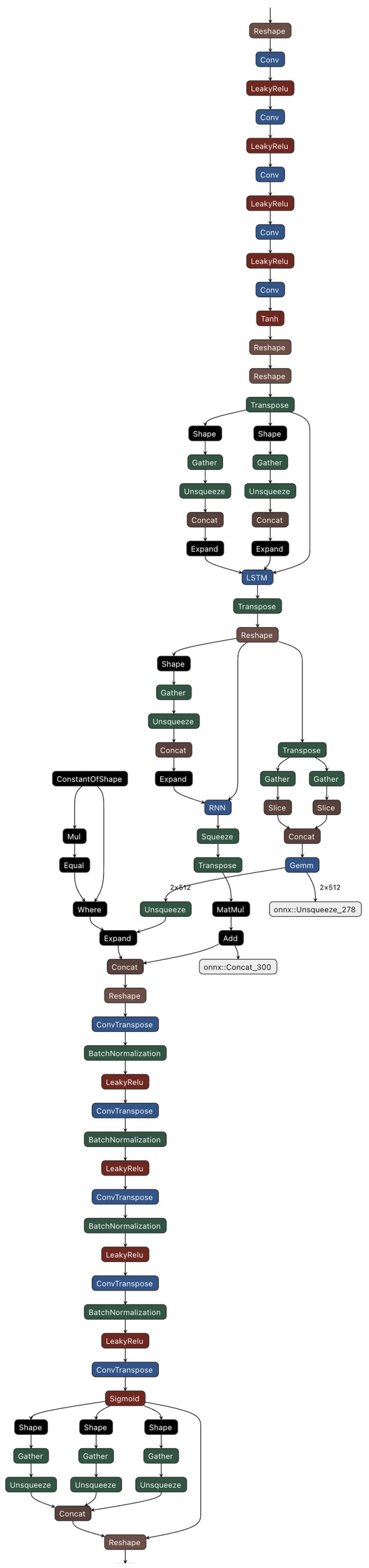
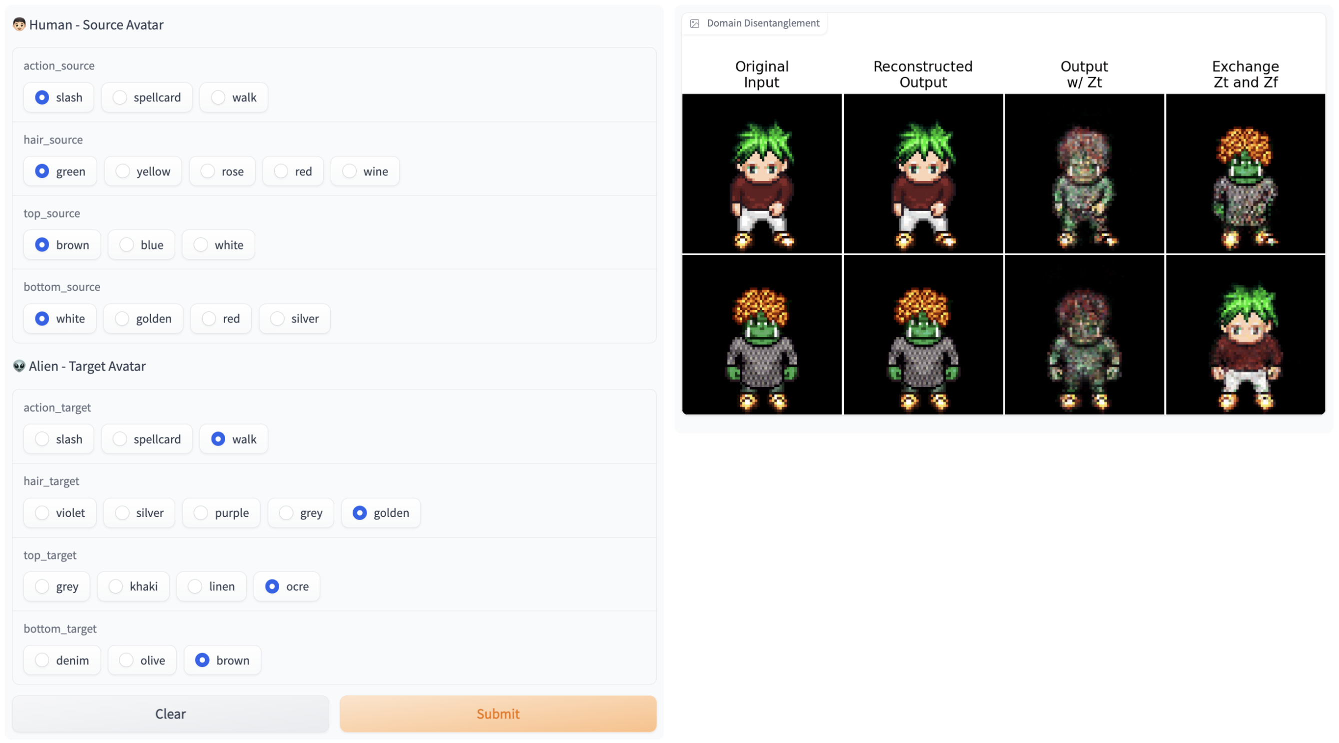


Figure A.1: The neural network structure and a Netron graph of the proposed *TranSVAE* framework. Zoom-ed in for the details.


Human - Source Avatar

action_source: [✓ slash] spellcard walk

hair_source: [✓ green] yellow rose red wine

top_source: [✓ brown] blue white

bottom_source: [✓ white] golden red silver

Alien - Target Avatar

action_target: slash spellcard [✓ walk]

hair_target: violet silver purple grey [✓ golden]

top_target: grey khaki linen [✓ ocre]

bottom_target: denim olive [✓ brown]

Figure A.2: 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 domain disentanglement and transfer as shown in the right-hand side.

be accessed via: https://huggingface.co/spaces/anonymous-demo/Anonymous-TranSVAE-Demo. Here we include the detailed instructions for playing with this demo.

This demo is built upon Hugging Face Spaces[2], 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. A.2. 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, $\mathbf{P}_1$ and $\mathbf{P}_2$. $\mathbf{P}_1$ uses the "Human" body while $\mathbf{P}_1$ uses the "Alien" body. The attribute pools of "Human" and "Alien" are non-overlapping across domains, resulting in completely heterogeneous $\mathbf{P}_1$ and $\mathbf{P}_2$. Each video sequence contains 8 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.*, $\{\mathbf{x}_1^{\mathbf{P}_1}, ..., \mathbf{x}_8^{\mathbf{P}_1}\}$ and $\{\mathbf{x}_1^{\mathbf{P}_2}, ..., \mathbf{x}_8^{\mathbf{P}_2}\}$;
- The **2nd** column: The reconstructed "Human" and "Alien" avatars $\{\widetilde{\mathbf{x}}_1^{\mathbf{P}_1}, ..., \widetilde{\mathbf{x}}_8^{\mathbf{P}_1}\}$ and $\{\widetilde{\mathbf{x}}_1^{\mathbf{P}_2}, ..., \widetilde{\mathbf{x}}_8^{\mathbf{P}_2}\}$;
- The **3rd** column: The reconstructed "Human" and "Alien" avatars using only $\{\mathbf{z}_1^{\mathcal{D}}, ..., \mathbf{z}_8^{\mathcal{D}}\}$, $\mathcal{D} \in \{\mathbf{P}_1, \mathbf{P}_2\}$, which are domain-invariant;
- The **4th** column: The reconstructed "Human" and "Alien" avatars by exchanging $\mathbf{z}_d^{\mathcal{D}}$, which results in two sequences with the same actions but exchanged appearance, *i.e.*, domain disentanglement and transfer.

[2]https://huggingface.co.

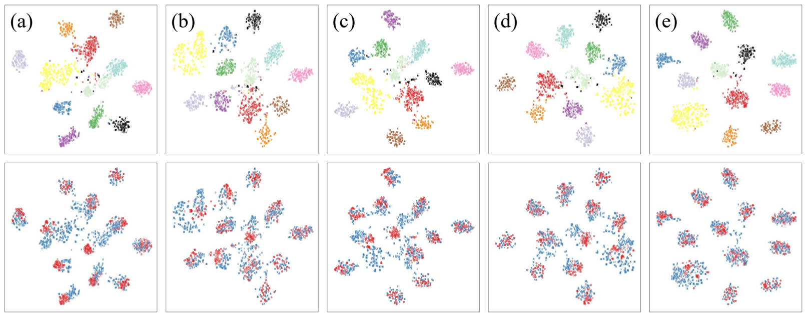
B Additional Experimental Results

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

B.1 Additional Ablation Studies

Component Integration Study. We further report the results of gradually adding each loss term to *TranSVAE* on the $\mathbf{H} \rightarrow \mathbf{U}$ task, as shown in Fig. A.3. Similar trend as reported in the main body (*i.e.*, Fig. 5 in Sec. 4.4) is observed in the feature visualization; both the domain and semantic alignments become better with the integration of the loss terms. The quantitative results also demonstrate the positive benefits brought by each loss term in *TranSVAE*.

Number of Frames T. Tab. A.2 shows the transfer performance with the variation of the number of frames T on UCF-HMDB dataset. As can be seen, the two tasks achieve the optimal performance with different T, specifically $T = 8$ for $\mathbf{U} \rightarrow \mathbf{H}$ and $T = 9$ for $\mathbf{H} \rightarrow \mathbf{U}$. Based on this observation, we apply a grid search on the validation set to obtain the optimal T for each task in our experiments.



#	$\mathcal{L}_{\text{svae}}$	$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{adv}}$	$\mathcal{L}_{\text{mi}}$	$\mathcal{L}_{\text{ctc}}$	PL	Accuracy (%)
$\mathcal{S}_{\text{only}}$							88.79
(a)	✓	✓					92.29 (+3.50)
(b)	✓	✓	✓				93.52 (+4.73)
(c)	✓	✓	✓	✓			93.70 (+4.91)
(d)	✓	✓	✓	✓	✓		94.40 (+5.61)
(e)	✓	✓	✓	✓	✓	✓	98.95 (+10.16)
$\mathcal{T}_{\text{sup}}$							96.85

Figure A.3: Loss integration studies on $\mathbf{H} \rightarrow \mathbf{U}$. **Left:** The t-SNE plots for class-wise (top row) and domain (bottom row, red source & blue target) features. **Right:** Ablation results (%) by adding each loss sequentially, *i.e.*, row (a) - row (e).

Table A.2: Ablation results for the frame number T.

Task	5	6	7	8	9	10	12	14	16	20
$\mathbf{U} \rightarrow \mathbf{H}$	84.44	86.11	84.17	**87.78**	84.72	85.28	83.89	84.44	82.78	85.00
$\mathbf{H} \rightarrow \mathbf{U}$	96.85	94.22	98.42	94.75	**98.95**	93.70	93.87	94.40	94.05	94.40

Table A.3: Ablation results for the pseudo-label threshold η.

Task	*w/o*	0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
$\mathbf{U} \rightarrow \mathbf{H}$	87.22 (−0.56)	86.94	87.22	87.50	**87.78**	86.11	86.67	86.94	86.39	86.11	86.39
$\mathbf{H} \rightarrow \mathbf{U}$	94.40 (−4.55)	98.42	97.37	98.77	98.60	98.25	98.25	**98.95**	97.90	97.72	95.27

Target Pseudo-Label Threshold η. We show the sensitivity analyses of the transfer performance with respect to the target pseudo label threshold η on UCF-HMDB dataset in Tab. A.3. The results show that different tasks yield the best transfer result with different η, specifically $\eta = 0.93$ for $\mathbf{U} \rightarrow \mathbf{H}$ and $\eta = 0.96$ for $\mathbf{H} \rightarrow \mathbf{U}$. Thus, we also apply the grid search on the validation set to obtain the optimal η 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. A.4.

C Impact & Limitation

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 the 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.

Potential Limitation. We discuss three limitations of the *TranSVAE* framework proposed in this work. They are also promising future directions.

- The empirical evaluation in this work is mainly on the action recognition task. The performance of *TranSVAE* on other video-related tasks, *e.g.*, semantic segmentation, is not tested.
- In this paper, *TranSVAE* is only evaluated on the typical two-domain transfer scenarios. The multi-source transfer case is not considered, but is worthy of further study.
- 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.

D Public Resources Used

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

• UCF_{101}[3]	Unknown
• HMDB_{51}[4]	CC BY 4.0
• Jester[5]	Unknown

[3]https://www.crcv.ucf.edu/data/UCF101.php
[4]https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database
[5]https://20bn.com/datasets/jester

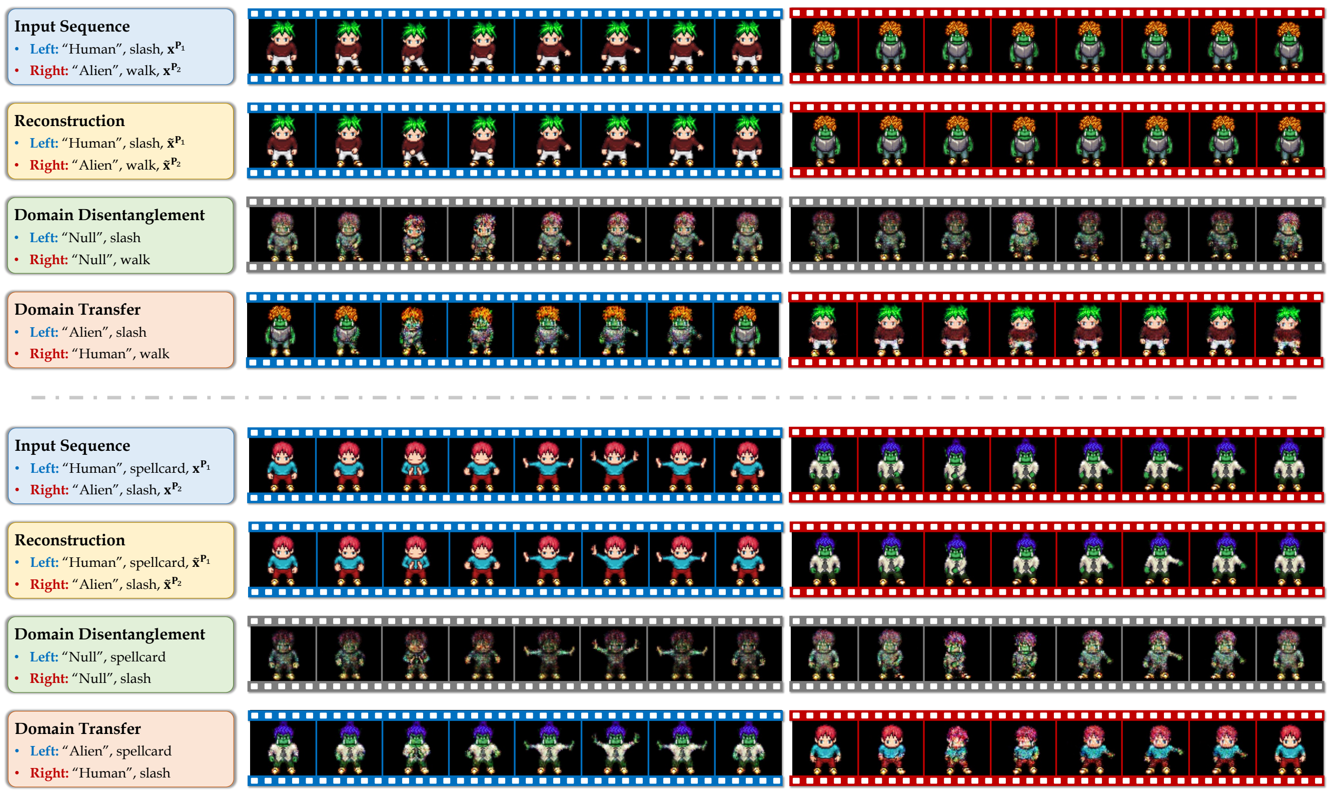

Input Sequence
Left: "Human", slash, $\mathbf{x}^{\mathbf{P}_1}$
Right: "Alien", walk, $\mathbf{x}^{\mathbf{P}_2}$
Reconstruction
Left: "Human", slash, $\tilde{\mathbf{x}}^{\mathbf{P}_1}$
Right: "Alien", walk, $\tilde{\mathbf{x}}^{\mathbf{P}_2}$
Domain Disentanglement
Left: "Null", slash
Right: "Null", walk
Domain Transfer
Left: "Alien", slash
Right: "Human", walk
Input Sequence
Left: "Human", spellcard, $\mathbf{x}^{\mathbf{P}_1}$
Right: "Alien", slash, $\mathbf{x}^{\mathbf{P}_2}$
Reconstruction
Left: "Human", spellcard, $\tilde{\mathbf{x}}^{\mathbf{P}_1}$
Right: "Alien", slash, $\tilde{\mathbf{x}}^{\mathbf{P}_2}$
Domain Disentanglement
Left: "Null", spellcard
Right: "Null", slash
Domain Transfer
Left: "Alien", spellcard
Right: "Human", slash

Figure A.4: Additional qualitative results for domain disentanglement and transfer in our *TranSVAE* framework.

• Epic-Kitchens[6]	CC BY-NC 4.0
• Sprites[7]	CC-BY-SA-3.0
• I3D[8]	Apache License 2.0
• TRN[9]	BSD 2-Clause License
• Netron[10]	MIT License

References

Choi, J.; Sharma, G.; Schulter, S.; and Huang, J.-B. 2020. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, 678–695. Springer.

Damen, D.; Doughty, H.; Farinella, G. M.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision*, 720–736.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *IEEE/CVF International Conference on Computer Vision*, 2556–2563. IEEE.

Li, Y.; and Mandt, S. 2018. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, 5670–5679. PMLR.

Materzynska, J.; Berger, G.; Bax, I.; and Memisevic, R. 2019. The jester dataset: A large-scale video dataset of human gestures. In *IEEE/CVF International Conference on Computer Vision Workshops*, 1–12.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[6]https://epic-kitchens.github.io/2021
[7]https://github.com/YingzhenLi/Sprites
[8]https://github.com/piergiaj/pytorch-i3d
[9]https://github.com/zhoubolei/TRN-pytorch
[10]https://github.com/lutzroeder/netron