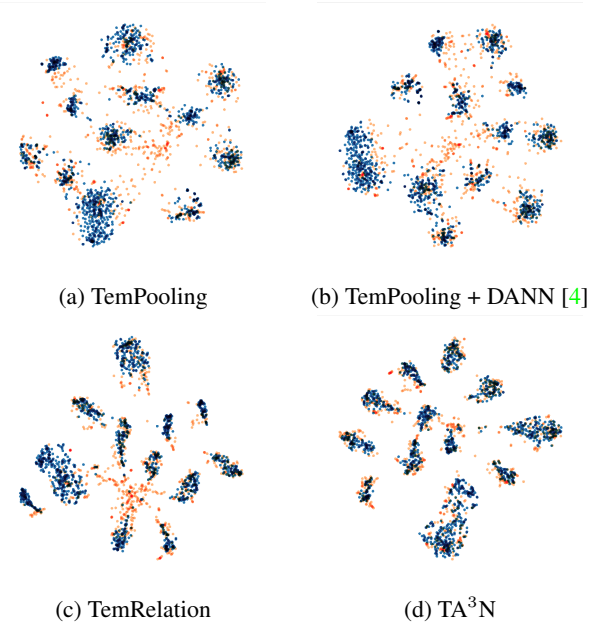
Supplementary Material for "Temporal Attentive Alignment for Large-Scale Video Domain Adaptation"

Min-Hung Chen[1]* Zsolt Kira[1] Ghassan AlRegib[1] Jaekwon Yoo[2] Ruxin Chen[2] Jian Zheng[3]*
[1]Georgia Institute of Technology [2]Sony Interactive Entertainment LLC [3]Binghamton University

In the supplementary material, we would like to show more detailed ablation studies, more implementation details, and a complete introduction of the datasets.

1. Visualization of distribution

We visualize the distribution of both domains using t-SNE [11] to investigate how our approaches bridge the gap between the source and target domains. Figures 1a and 1b show that the models using the TemPooling architecture poorly align the distribution between different domains, even with the integration of image-based DA approaches. Figure 1c shows the temporal relation module helps to group source data (blue) into denser clusters but is still not able to generalize the distribution into the target domains (orange). Finally, with TA^{3}N, data from both domains are clustered and aligned with each other (Figure 1d).



(a) TemPooling (b) TemPooling + DANN [4]

(c) TemRelation (d) TA^{3}N

Figure 1: The comparison of t-SNE visualization with source (blue) and target (orange) distributions.

2. Domain Attention Mechanism

We also apply the domain attention mechanism to TemPooling by attending to the raw frame features, as shown in Figure 2. Tables 1 and 2 show that the domain attention mechanism improves the performance for both TemPooling and TemRelation architectures, including all types of adversarial discriminators. This implies that video DA can benefit from domain attention even if the backbone architecture does not encode temporal dynamics.

Temporal Module	TemPooling	TemPooling + Attn.	TemRelation	TemRelation + Attn.
Target only	80.56 (-)		82.78 (-)	
Source only	70.28 (-)		71.67 (-)	
$\hat{G}_{sd}$	71.11 (0.83)	71.94 (1.66)	74.44 (2.77)	75.00 (3.33)
$\hat{G}_{td}$	71.11 (0.83)	72.78 (2.50)	74.72 (3.05)	76.94 (5.27)
$\hat{G}_{rd}$	- (-)	- (-)	76.11 (4.44)	76.94 (5.27)
All $\hat{G}_d$	71.11 (0.83)	**73.06 (2.78)**	77.22 (5.55)	**78.33 (6.66)**

Table 1: The evaluation of accuracy (%) for integrating $\hat{G}_d$ in different positions on "U → H" . Gain values are in ().

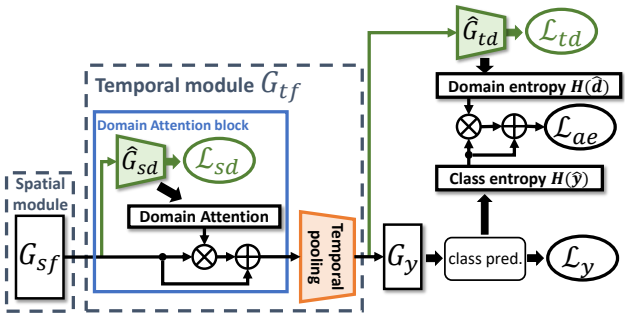


Figure 2: Baseline architecture (TemPooling) equipped with the domain attention mechanism (ignoring the input feature parts to save space).

*Work partially done as a SIE intern

Temporal Module	TemPooling	TemPooling + Attn.	TemRelation	TemRelation + Attn.
Target only	92.12 (-)		94.92 (-)	
Source only	74.96 (-)		73.91 (-)	
$\hat{G}_{sd}$	75.13 (0.17)	77.58 (2.62)	74.44 (1.05)	78.63 (4.72)
$\hat{G}_{td}$	75.13 (0.17)	78.46 (3.50)	75.83 (1.93)	81.44 (7.53)
$\hat{G}_{rd}$	- (-)	- (-)	75.13 (1.23)	78.98 (5.07)
All $\hat{G}_d$	75.13 (0.17)	**78.46 (3.50)**	80.56 (6.66)	**81.79 (7.88)**

Table 2: The evaluation of accuracy (%) for integrating $\hat{G}_d$ in different positions on "H → U" . Gain values are in ().

3. Implementation Details

3.1. Detailed architectures

The architecture with detailed notations for the baseline is shown in Figure 3. For our proposed TA^{3}N, after generating the n-frame relation features R_n by the temporal relation module, we calculate the domain attention value w^n using the domain prediction $\hat{d}$ from the relation discriminator G^n_{rd}, and then attend to R_n using w^n with a residual connection. To calculate the attentive entropy loss $\mathcal{L}_{ae}$, since the videos with low domain discrepancy are what we only want to focus on, we attend to the class entropy loss $H(\hat{y})$ using the domain entropy $H(\hat{d})$ as the attention value with a residual connection, as shown in Figure 4.

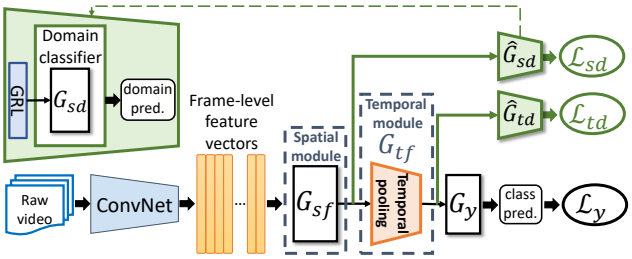


Figure 3: The detailed baseline architecture (TemPooling) with the adversarial discriminators $\hat{G}_{sd}$ and $\hat{G}_{td}$.

3.2. Optimization

Our implementation is based on the PyTorch [12] framework. We utilize the ResNet-101 model pre-trained on ImageNet as the frame-level feature extractor. We sample a fixed number K of frame-level feature vectors with equal spacing in the temporal direction for each video (K is equal to 5 in our setting to limit computational resource requirements). For optimization, the initial learning rate is 0.03, and we follow one of the commonly used learning-rate-decreasing strategies shown in DANN [4]. We use stochastic gradient descent (SGD) as the optimizer with the momentum and weight decay as 0.9 and 1×10^{-4}, respectively. The ratio between the source and target batch size is proportional to the scale between the source and target datasets. The source batch size depends on the scale of the dataset, which is 32 for UCF-Olympic and UCF-HMDB$_{small}$, 128 for UCF-HMDB$_{full}$ and 512 for Kinetics-Gameplay. The optimized values of λ^s, λ^r and λ^t are found using the coarse-to-fine grid-search approach. We first search using a coarse-grid with the geometric sequence [0, 10^{-3}, 10^{-2}, ..., 10^0, 10^1]. After finding the optimized range of values, [0, 1], we search again using a fine-grid with the arithmetic sequence [0, 0.25, ..., 1]. The final values are 0.75 for λ^s, 0.5 for λ^r and 0.75 for λ^t, respectively. We search γ only by a coarse-grid, and the best value is 0.3. For future work, we plan to adopt adaptive weighting techniques used for multi-task learning, such as uncertainty weighting [7] and GradNorm [2], to replace the manual grid-search method.

3.3. Comparison with other work

As mentioned in the experimental setup, we compare our proposed TA^{3}N with other approaches by extending several state-of-the-art image-based DA methods [4, 10, 9, 14] for video DA with our TemPooling and TemRelation architectures, which are shown as follows:

1. *DANN [4]*: we add one adversarial discriminator $\hat{G}_{sd}$ right after the spatial module and add another one $\hat{G}_{td}$ right after the temporal module. We do not add one more discriminator for relation features for the fair comparison between TemPooling and TemRelation.

2. *JAN [10]*: we add Joint Maximum Mean Discrepancy (JMMD) to the final video representation and the class prediction.

3. *AdaBN [9]*: we integrate an adaptive batch-normalization layer into the feature generator G_{sf}. In the adaptive batch-normalization layer, the statistics (mean and variance) for both source and target domains are calculated, but only the target statistics are used for validating the target data.

4. *MCD [14]*: we add another classifier G'_y and follow the adversarial training procedure of Maximum Classifier Discrepancy to iteratively optimize the generators (G_{sf} and G_{tf}) and the classifier (G_y).

4. Datasets

The full summary of all four datasets investigated in this paper is shown in Table 3.

4.1. UCF-HMDB$_{full}$

We collect all of the relevant and overlapping categories between UCF101 [15] and HMDB51 [8], which results in 12 categories: *climb, fencing, golf, kick_ball, pullup, punch, pushup, ride_bike, ride_horse, shoot_ball, shoot_bow,* and *walk*. Each category may correspond to multiple categories in the original UCF101 or HMDB51 dataset, as shown in

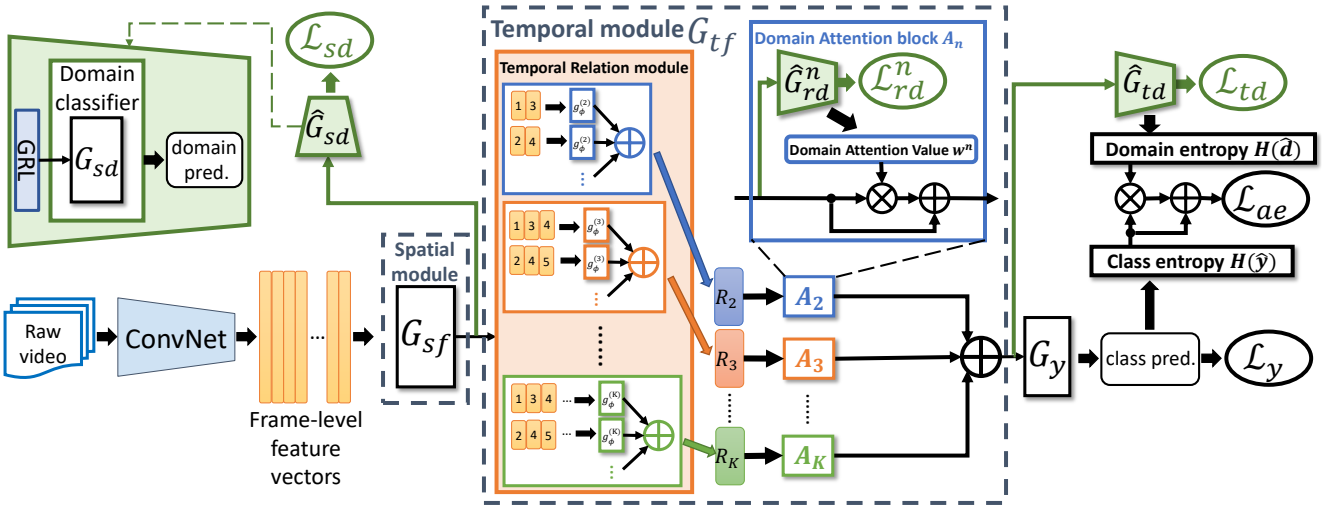


Figure 4: The detailed architecture of the proposed TA^{3}N.

	UCF-HMDB$_{small}$	UCF-Olympic	UCF-HMDB$_{full}$	Kinetics-Gameplay
length (sec.)	1 - 21	1 - 39	1 - 33	1 - 10
resolution	UCF: 320 × 240 / Olympic: vary / HMDB: vary×240 / Kinetics: vary / Gameplay: 1280 × 720			
frame rate	UCF: 25 / Olympic: 30 / HMDB: 30 / Kinetics: vary / Gameplay: 30			
class #	5	6	12	30
training video #	UCF: 482 / HMDB: 350	UCF: 601 / Olympic: 250	UCF: 1438 / HMDB: 840	Kinetics: 43378 / Gameplay: 2625
validation video #	UCF: 189 / HMDB: 150	UCF: 240 / Olympic: 54	UCF: 571 / HMDB: 360	Kinetics: 3246 / Gameplay: 749

Table 3: The summary of the cross-domain video datasets.

Table 4. This dataset, **UCF-HMDB$_{full}$**, includes 1438 training videos and 571 validation videos from UCF, and 840 training videos and 360 validation videos from HMDB, as shown in Table 3. Most videos in UCF are from certain scenarios or similar environments, while videos in HMDB are in unconstrained environments and different camera angles, as shown in Figure 5.

4.2. Kinetics-Gameplay

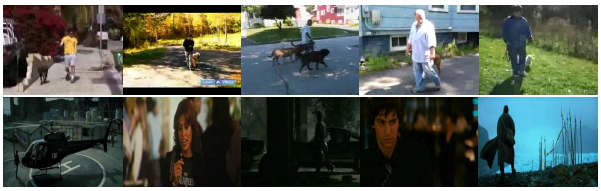
We create the *Gameplay* dataset by first collecting gameplay videos from two video games, *Detroit: Become Human* and *Fortnite*, to build our own action dataset for the virtual domain. The total length of the videos is 5 hours and 41 minutes. We segment all of the raw, untrimmed videos into video clips according to human annotations, which results in 91 categories: *argue, arrange_object, assemble_object, break, bump, carry, carve, chop_wood, clap, climb, close_door, close_others, crawl, cross_arm, crouch, crumple, cry, cut, dance, draw, drink, drive, eat, fall_down, fight, fix_hair, fly_helicopter, get_off, grab, haircut, hit, hit_break, hold, hug, juggle_coin, jump, kick, kiss, kneel, knock, lick, lie_down, lift, light_up, listen, make_bed, mop_floor, news_anchor, open_door, open_others, paint_brush, pass_object, pet, poke, pour, press, pull, punch, push, push_object, put_object, raise_hand, read, row_boat, run, shake_hand, shiver, shoot_gun, sit, sit_down, slap, sleep, slide, smile, stand, stand_up, stare, strangle, swim, switch, take_off, talk, talk_phone, think, throw, touch, walk, wash_dishes, wa-*

UCF-HMDB$_{full}$	UCF	HMDB
climb	RockClimbingIndoor, RopeClimbing	climb
fencing	Fencing	fencing
golf	GolfSwing	golf
kick_ball	SoccerPenalty	kick_ball
pullup	PullUps	pullup
punch	Punch, BoxingPunchingBag, BoxingSpeedBag	punch
pushup	PushUps	pushup
ride_bike	Biking	ride_bike
ride_horse	HorseRiding	ride_horse
shoot_ball	Basketball	shoot_ball
shoot_bow	Archery	shoot_bow
walk	WalkingWithDog	walk

Table 4: The lists of all collected categories in UCF and HMDB.

(a) fencing

(b) kick_ball



(c) walk

Figure 5: Snapshots of some example categories on UCF-HMDB$_{full}$. For each category, the snapshots from UCF are shown in the upper row, and the snapshots from HMDB are shown in the lower row.

ter_plant, wave_hand, and *weld*. The maximum length for each video clip is 10 seconds, and the minimum is 1 second. We also split the dataset into training, validation, and testing sets by randomly selecting videos in each category with the ratio 7:2:1. We build the **Kinetics-Gameplay** dataset by selecting 30 overlapping categories between Gameplay and one of the largest public video datasets *Kinetics-600* [6, 1]: *break, carry, clean_floor, climb, crawl, crouch, cry, dance, drink, drive, fall_down, fight, hug, jump, kick, light_up, news_anchor, open door, paint_brush, paraglide, pour, push, read, run, shoot_gun, stare, talk, throw, walk,* and *wash_dishes*. Each category may also correspond to multiple categories in both datasets, as shown in Table 5. Kinetics-Gameplay includes 43378 training videos and 3246 validation videos from Kinetics, and 2625 training videos and 749 validation videos from Gameplay, as shown in Table 3. Kinetics-Gameplay is much more challenging than UCF-HMDB$_{full}$ due to the significant domain shift between the distributions of virtual and real data. Furthermore, The alignment between imbalanced-scaled source and target data is also another challenge. Some example snapshots are shown in Figure 6.

Figure 6: Some example screenshots from YouTube videos in Kinetics-Gameplay (left two: Gameplay, right two: Kinetics)

5. More Details

5.1. JAN on Kinetics-Gameplay

JAN [10] does not perform well on Kinetics-Gameplay compared to the performance on UCF-HMDB$_{full}$. The main reason is the imbalanced size between the source and target data in Kinetics-Gameplay. The discrepancy loss MMD is calculated using the same number of source and target data (not the case for other types of DA approaches). Therefore, in each iteration, MMD is calculated using parts of the source batch and the whole target batch. This means that the domain discrepancy is reduced only between part of source data and target data during training, so the learned model is still overfitted to the source domain. The discrepancy loss MMD works well when the source and target data are balanced, which is the case for most image DA datasets and UCF-HMDB$_{full}$, but not for Kinetics-Gameplay.

5.2. Comparison with AMLS [5]

When evaluating on UCF-HMDB$_{small}$, AMLS [5] fine-tunes their networks using UCF and HMDB, respectively, before applying their DA approach. Here we only show their results which are fine-tuned with source data, because the target labels should be unseen during training in unsupervised DA settings. For example, we don't compare their results which test on HMDB data using the models fine-tuned with HMDB data since it is not unsupervised DA.

5.3. Other baselines

3D ConvNets [16] have also been used for extracting video-level feature representations. However, 3D ConvNets consume a great deal of GPU memory, and [17] also shows that 3D ConvNets are limited by efficiency and effectiveness issues when extracting temporal information.

Optical-flow extracts the motion characteristics between neighbor frames to compensate for the lack of temporal information in raw RGB frames. In this paper, we focus on attending to the temporal dynamics to effectively align domains even with only RGB frames. We consider optical-flow to be complementary to our method.

Kinetics-Gameplay	Kinetics	Gameplay
break	breaking boards, smashing	break, bump, hit_break
carry	carrying baby	carry
clean_floor	mopping floor	mop_floor
climb	climbing a rope, climbing ladder, climbing tree, ice climbing, rock climbing	climb
crawl	crawling baby	crawl
crouch	squat, lunge	crouch, kneel
cry	crying	cry
dance	belly dancing, krumping, robot dancing	dance
drink	drinking shots, tasting beer	drink
drive	driving car, driving tractor	drive
fall_down	falling off bike, falling off chair, faceplanting	fall_down
fight	pillow fight, capoeira, wrestling, punching bag, punching person (boxing)	fight, strangle, punch, hit
hug	hugging (not baby), hugging baby	hug
jump	high jump, jumping into pool, parkour	jump
kick	drop kicking, side kick	kick
light_up	lighting fire	light_fire
news_anchor	news anchoring	news_anchor
open_door	opening door, opening refrigerator	open_door
paint_brush	brush painting	paint_brush
paraglide	paragliding	paraglide
pour	pouring beer	pour
push	pushing car, pushing cart, pushing wheelbarrow, pushing wheelchair, push up	push, push_object
read	reading book, reading newspaper	read
run	running on treadmill, jogging	run
shoot_gun	playing laser tag, playing paintball	shoot_gun
stare	staring	stare
talk	talking on cell phone, arguing, testifying	talk, argue, talk_phone
throw	throwing axe, throwing ball (not baseball or American football), throwing knife, throwing water balloon	throw
walk	walking the dog, walking through snow, jaywalking	walk
wash_dishes	washing dishes	wash_dishes

Table 5: The lists of all collected categories in Kinetics and Gameplay.

5.4. Comparison with literature in other fields

Cycle-consistency. Some papers related to cycle-consistency [18, 3] introduce self-supervised methods for learning visual correspondence between images or videos from unlabeled videos. They use cycle-consistency as free supervision to learn video representations. The main difference from our approach is that we explicitly align the feature spaces between source and target domains, while these self-supervised methods aim to learn general representations using only the source domain. We see cycle-consistency as a complementary method that can be integrated into our approach to achieve more effective domain alignment.

Robotics. In Robotics, it is a common trend to transfer the models trained in simulation to real world. One of the effective method to bridge the domain gap is randomizing the dynamics of the simulator during training to improve the robustness for different environments [13]. The setting is different from our task because we focus on feature learning rather than policy learning, and we see domain randomization as a complementary technique that can extend our approach to a more generalized version.

5.5. Failure cases for TemRelation

TemRelation shows limited improvement over TemPooling for some categories with consistency across time. For example, with the same DA method (DANN), TemRelation has the same accuracy with TemPooling for *ride_bike* (97%), and has lower accuracy for *ride_horse* (93% and 97%). The possible reason is that temporal pooling can already model temporally consistent actions well, and it may be redundant to model these actions with multiple timescales like TemRelation.

5.6. Testing time for TA^{3}N

Different from TA^{2}N, TA^{3}N passes data to all the domain discriminators during testing. However, since all our domain discriminators are shallow, the testing time is similar. In our experiment, TA^{3}N only computes 10% more time than TA^{2}N.

References

[1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 4

[2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning (ICML)*, 2018. 2

[3] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 2

[5] Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. Deep domain adaptation in action space. In *British Machine Vision Conference (BMVC)*, 2018. 4

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[7] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[8] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[9] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 2

[10] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning (ICML)*, 2017. 2, 4

[11] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 1

[12] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshop (NeurIPSW)*, 2017. 2

[13] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 5

[14] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[17] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[18] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5