Unfolding Videos Dynamics via Taylor Expansion

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Taking inspiration from physical motion, we present a new self-supervised dynamics learning strategy for videos: **Vi**deo Time-**Di**fferentiation for Instance **Di**scrimination (ViDiDi). ViDiDi is a simple and data-efficient strategy, readily applicable to existing self-supervised video representation learning frameworks based on instance discrimination. At its core, ViDiDi observes different aspects of a video through various orders of temporal derivatives of its frame sequence. These derivatives, along with the original frames, support the Taylor series expansion of the underlying continuous dynamics at discrete times, where higher-order derivatives emphasize higher-order motion features. ViDiDi learns a single neural network that encodes a video and its temporal derivatives into consistent embeddings following a balanced alternating learning algorithm. By learning consistent representations for original frames and derivatives, the encoder is steered to emphasize motion features over static backgrounds and uncover the hidden dynamics in original frames. Hence, video representations are better separated by dynamic features. We integrate ViDiDi into existing instance discrimination frameworks (VICReg, BYOL, and SimCLR) for pretraining on UCF101 or Kinetics and test on standard benchmarks including video retrieval, action recognition, and action detection. The performances are enhanced by a significant margin without the need for large models or extensive datasets.

1 Introduction

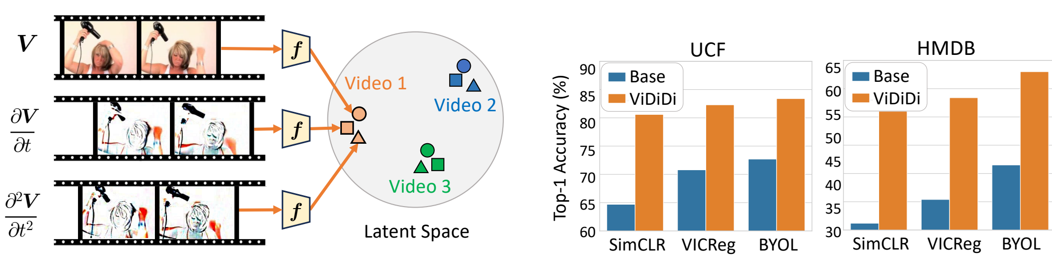


Figure 1: **Method Overview**. Left: Demonstration of ViDiDi. Right: ViDiDi enhances existing instance discrimination methods significantly on action recognition.

Learning video representations is central to various aspects of video understanding, such as action recognition [1, 2], video retrieval [3, 4], and action detection [5]. While supervised learning requires expensive video labeling [6], recent works highlight the strengths of self-supervised learning (SSL) from unlabeled videos [7, 8, 9] with a large number of training videos. One popular strategy for SSL on video representations uses instance discrimination objectives, such as SimCLR [10], initially demonstrated for images [11, 10, 12, 13, 14]. When adapting this approach to videos, previous methods often treat time directly as an additional spatial dimension. This may neglect the special

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

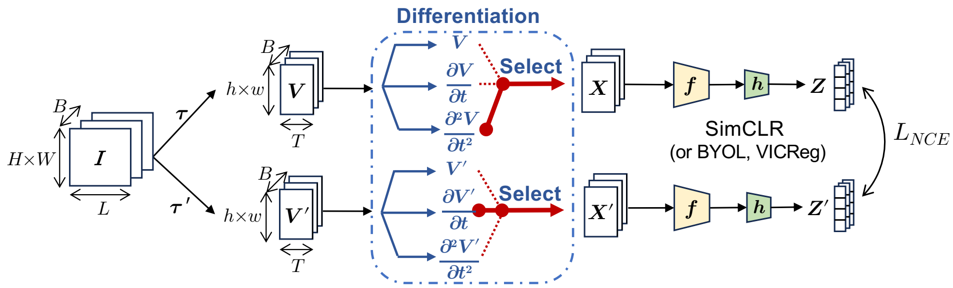


Figure 2: **Illustration of the ViDiDi framework.** For a batch of videos $\boldsymbol{I}$, we do two spatio-temporal augmentations to obtain two batches of clips: $\boldsymbol{V}$ and $\boldsymbol{V'}$. $\boldsymbol{X}$ and $\boldsymbol{X'}$ are the 0^{th}, 1^{st}, or 2^{nd} order derivatives of $\boldsymbol{V}$ and $\boldsymbol{V'}$, decided via a balanced alternating learning strategy (alg. 1). They are the inputs to the video encoder for learning the video encoder under SimCLR, BYOL, or VICReg.

feature of the temporal dimension in carrying dynamic information, causing models to prioritize static content (e.g., background scenes) over dynamic features (e.g., motion, action, and interaction), which are often essential to video understanding [1, 2, 6, 15, 16, 17, 18, 19].

In contrast, we utilize the unique role of time in "unfolding" continuous real-world dynamics. We introduce a generalizable and data-efficient method, applicable to self-supervised video representation learning through instance discrimination including VICReg [13], BYOL [11], and SimCLR [10]. We view a video as a continuous and dynamic process and use the Taylor series expansion to express this continuous process as a weighted sum of its temporal derivatives at each frame. Based on this, we evaluate temporal derivatives of videos as hidden views apart from the original frames. Following a *balanced alternating learning strategy*, we train models to align representations for the original video and its temporal derivatives, such that the learned representations are steered to dynamic information in the images (section 1). Herein, we refer to this approach as Video Time-Differentiation for Instance Discrimination (ViDiDi). Our method demonstrates excellent generalizability and data efficacy on standard benchmarks including action recognition (section 1), video retrieval, and action detection.

2 Approach

2.1 A Thought Experiment on Physical Motion

Our idea is analogous to and inspired by physical motion. Consider a 1-D toy example through a thought experiment. Imagine that we observe the free fall motion of a ball. The zeroth, first, and second derivatives are the position $\boldsymbol{y}(t)$, velocity $\boldsymbol{v}(t)$, and acceleration $\boldsymbol{a}(t)$, respectively. Given the physical law, the free fall motion is governed by $\boldsymbol{y}(t) = \boldsymbol{y_0} + \boldsymbol{v_0}t - \frac{1}{2}\boldsymbol{g}t^2$, including three latent factors: the initial position $\boldsymbol{y_0}$, the initial velocity $\boldsymbol{v_0}$, and the gravity $\boldsymbol{g}$. Inferring the representation shared across these different views reveals gravity $\boldsymbol{g}$, which is the defining feature of the dynamics.

2.2 The ViDiDi Framework

ViDiDi involves 1) creating multiple views from videos through augmentation and differentiation, 2) a balanced alternating learning strategy for pair-wise encoding of the different views into consistent representations, and 3) plugging this strategy into existing instance discrimination methods, including SimCLR [10], BYOL [11], and VICReg [13]. See fig. 2 for an overview.

Creating multiple views from videos. i) ***Augmentation***: Given a batch of videos, we sample each video by randomly cropping two clips. For each clip, we apply a random set of spatial augmentations such as random crops to create two batches of augmented clips, denoted as $\boldsymbol{V}$ and $\boldsymbol{V'}$. ii) ***Differentiation***: We further conduct temporal differentiation on clips. For every augmented clip within either $\boldsymbol{V}$ or $\boldsymbol{V'}$, we evaluate its 0^{th}, 1^{st}, or 2^{nd} temporal derivatives. We approximate temporal derivatives with finite forward differences. We make sure within one batch, the clips are all original frames or derivatives in the same order. Thus, we can now denote the three possible views for a batch of clips as $\boldsymbol{V}, \frac{\partial \boldsymbol{V}}{\partial t}$ and $\frac{\partial^2 \boldsymbol{V}}{\partial t^2}$.

Balanced alternating learning strategy. We design a paring schedule for learning representations among different views. Specifically, we define seven pairs: $(\boldsymbol{V}, \boldsymbol{V'})$, $(\boldsymbol{V}, \frac{\partial \boldsymbol{V'}}{\partial t})$, $(\frac{\partial \boldsymbol{V}}{\partial t}, \boldsymbol{V'})$, $(\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V'}}{\partial t})$, $(\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial^2 \boldsymbol{V'}}{\partial t^2})$, $(\frac{\partial^2 \boldsymbol{V}}{\partial t^2}, \frac{\partial \boldsymbol{V'}}{\partial t})$, $(\frac{\partial^2 \boldsymbol{V}}{\partial t^2}, \frac{\partial^2 \boldsymbol{V'}}{\partial t^2})$. $(\boldsymbol{X}, \boldsymbol{X'})$ will be chosen from one of the seven pairs above, and serve as inputs to the video encoder for learning using methods such as SimCLR. At each step, $(\boldsymbol{X}, \boldsymbol{X}')$ is decided following alg. 1. Intuitively we choose this strategy to let learning of derivatives guide original frames. We empirically verify that this balanced alternating learning strategy plays an important role in the learning process in the ablation study in appendix C.

Plug into existing instance discrimination methods. We plug ViDiDi into existing instance discrimination frameworks, namely ViDiDi-SimCLR, ViDiDi-BYOL, and ViDiDi-VIC. We describe ViDiDi-SimCLR below and refer to the appendix for details about other models. $(\boldsymbol{X}, \boldsymbol{X}')$ is input to the video encoder f and then the projection head h, yielding paired embeddings $(\boldsymbol{Z}, \boldsymbol{Z}')$ for evaluating the loss $\mathcal{L}_{NCE}$ and training the networks. We discuss more details in appendix D.

3 Experiments

3.1 Experiment Setup

We train and evaluate ViDiDi using human action video datasets. **UCF101** [1] includes 13k videos from 101 classes. **HMDB51** [2] contains 7k videos from 51 classes. In addition, we also use larger and more diverse datasets, **K400** [6], aka Kinetics400, including 240k videos from 400 classes, and **K200-40k**, including 40k videos from 200 classes, as a subset of Kinetics 400, helps verify data-efficiency. In our experiments, we pretrain models with UCF101, K400, or K200-40k and then test them with UCF101 or HMDB51, using split 1 for both datasets. **AVA** contains 280K videos from 60 action classes, each video is annotated with spatiotemporal localization of human actions. After pertaining on Kinetics or UCF101, we follow the evaluation protocol in previous works [3, 7, 8], including three types of downstream tasks. i) *video retrieval* on UCF101 and HMDB51, ii) *action recognition*, and iii) *Action detection* on AVA. More details are in appendix E.

3.2 Results

Table 1: **Video retrieval using different methods.**

Method	Pretrained	UCF101			HMDB51		
		1	5	10	1	5	10
SimCLR	UCF101	29.6	41.4	49.3	17.5	34.7	45.1
ViDiDi-SimCLR	UCF101	**38.3**	**54.6**	**64.5**	**17.5**	**38.9**	**52.4**
BYOL	UCF101	32.2	43.0	50.5	13.8	31.1	44.4
ViDiDi-BYOL	UCF101	**43.7**	**60.4**	**70.1**	**19.3**	**44.1**	**56.6**
VICReg	UCF101	31.1	43.6	50.9	15.7	33.7	44.5
ViDiDi-VIC	UCF101	**47.6**	**60.9**	**68.6**	**19.7**	**40.5**	**55.1**
VICReg	K400	41.9	56.5	64.8	21.7	44.1	56.1
ViDiDi-VIC	K400	**51.2**	**64.6**	**72.6**	**25.0**	**47.2**	**60.9**
ViDiDi-VIC	K200-40k	**49.5**	**63.4**	**71.0**	**24.7**	**45.4**	**56.0**

Table 2: **Video retrieval using different backbones backbones.**

Net	Method	UCF101			HMDB51		
		1	5	10	1	5	10
R(2+1)D-18	VICReg	30.2	44.1	51.4	15.8	33.6	45.5
	ViDiDi-VIC	**47.2**	**62.6**	**69.8**	**20.6**	**44.1**	**57.7**
MC3-18	VICReg	31.9	44.4	51.4	15.6	35.6	46.1
	ViDiDi-VIC	**44.1**	**59.8**	**68.0**	**20.3**	**40.3**	**53.4**
S3D	VICReg	29.2	41.9	49.2	12.8	29.8	40.9
	ViDiDi-VIC	**42.9**	**59.0**	**67.5**	**18.4**	**38.4**	**51.4**

Table 3: **Action detection using different methods.**

Method	VICReg	**ViDiDi-VIC**	SimCLR	**ViDiDi-SimCLR**	BYOL	**ViDiDi-BYOL**
mAP	0.089	**0.106**	0.079	**0.094**	0.087	**0.118**

Superior performance, generalizability, and data efficiency. ViDiDi learns effective video representations with limited data. It can be plugged into multiple existing frameworks based on instance discrimination, using multiple encoder architectures, and improving the performance of various downstream tasks significantly as shown in fig. 1, table 3, table 1, and table 2. Besides, as presented in table 5, table 4, pretrained on small dataset UCF101 or K200-40k, ViDiDi surpasses prior video representation learning works pretrained on large-scale K400 or K600 dataset. Further, ViDiDi-VIC pretrained on UCF101 or K200-40k outperforms its baseline method VICReg pretrained on K400, and also reaches compatible performance as ViDiDi-VIC pretrained on K400, as shown in table 1. In summary, ViDiDi is highly generalizable, and efficiently uncovers dynamic features using limited data. To gain insights into how ViDiDi works, we further analyze the features and attention of video encoders learned via ViDiDi.

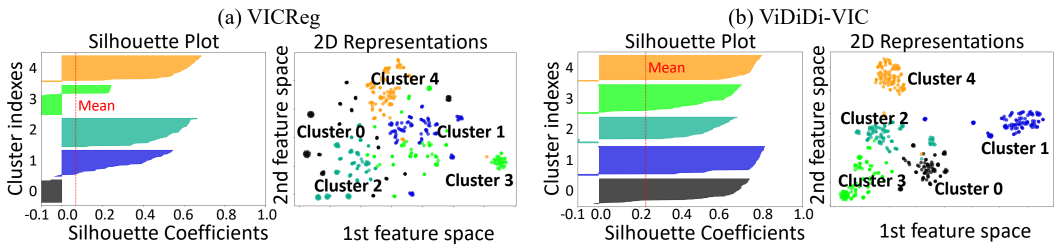


Figure 3: **Silhouette scores and t-SNE of top 5 classes** from VICReg (left) and ViDiDi-VIC (right).

Better separation based on dynamic features. We use t-SNE [20] to visualize the representations from five classes of videos. As shown in fig. 3, representations learned with ViDiDi are better clustered by action classes. We also use Silhouette score [21] in table 7 to quantify the degree of separation. To study whether such separation results from capturing better dynamic features, we visualize the spatiotemporal attention using Saliency Tubes [22]. ViDiDi leads the model to attend to dynamic aspects of the video, such as motions and interactions, rather than static backgrounds as shown in fig. 4. These results align with our intuition that ViDiDi attends to dynamic parts and avoids learning static content as a learning shortcut, resulting in efficient utilization of data. We attach more visualization and analysis results in appendix F.

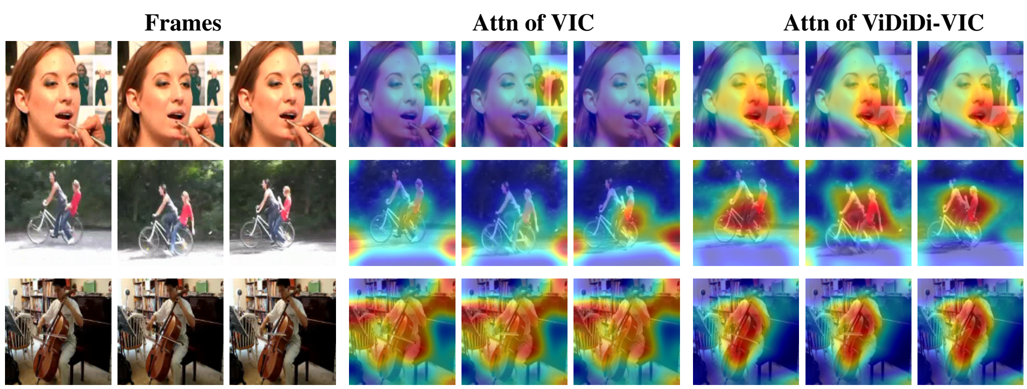


Figure 4: **Spatiotemporal attention on UCF and HMDB51.**

4 Conclusion

In this paper, we introduce ViDiDi, a novel, data-efficient, and generalizable framework for self-supervised video representation. We utilize the Taylor series to unfold multiple views from a video through different orders of temporal derivatives and learn representations among these views following a balanced alternating learning strategy. ViDiDi steers the video encoder to dynamic features instead of static shortcuts, enhancing performance on common video representation learning tasks significantly. We identify multiple future directions as well, such as applying ViDiDi to other modalities, and exploring other vision tasks that require a more fine-grained understanding of video dynamics [23, 24]. Furthermore, a potentially fruitful direction is to use this approach to learn intuitive physics, better supporting agents to understand, predict, and interact with the physical world.

References

[1] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[2] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[3] Sheng Guo, Zihua Xiong, Yujie Zhong, Limin Wang Wang, Xiaobo Guo, Bing Han, and Huang Weilin. Cross-architecture self-supervised video representation learning. In *CVPR*, June 2022.

[4] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding, 2021.

[5] Chunhui Gu, Chen Sun, Sudheendra Vijayanarasimhan, Caroline Pantofaru, David A. Ross, George Toderici, Yeqing Li, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. *CVPR*, pages 6047–6056, 2017.

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[7] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Neurips*, 2020.

[8] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning, 2021.

[9] Ishan Rajendrakumar Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Comput. Vis. Image Underst.*, 219:103406, 2021.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *NeurIPS*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735, 2020.

[13] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

[14] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[15] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *NeurIPS*, volume 33, pages 19545–19560. Curran Associates, Inc., 2020.

[16] Shengyi Qian, Linyi Jin, Chris Rockwell, Siyi Chen, and David F. Fouhey. Understanding 3d object articulation in internet videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1599–1609, June 2022.

[17] Zheyun Qin, Xiankai Lu, Xiushan Nie, Yilong Yin, and Jianbing Shen. Exposing the self-supervised space-time correspondence learning via graph kernels. *AAAI*, 37(2):2110–2118, Jun. 2023.

[18] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. *ICCV*, pages 13506–13515, 2021.

[19] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3888–3897, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[21] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[22] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Remco C. Veltkamp, and Ronald Poppe. Saliency tubes: Visual explanations for spatio-temporal convolutions. *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1830–1834, 2019.

[23] H. Zhang, D. Liu, Q. Zheng, and B. Su. Modeling video as stochastic processes for fine-grained video representation learning. In *CVPR*, pages 2225–2234, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society.

[24] Hilde Kuehne, Ali Bilgin Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014.

[25] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, pages 11205–11214, June 2021.

[26] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, pages 3298–3308, 2021.

[27] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency, 2020.

[28] S. Jenni and H. Jin. Time-equivariant contrastive video representation learning. In *ICCV*, pages 9950–9960, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

[29] Di Yang, Yaohui Wang, Quan Kong, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, and François Brémond. Self-supervised video representation learning via latent time navigation. In *AAAI*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[30] Jinpeng Wang, Yuting Gao, Ke Li, Xinyang Jiang, Xiao-Wei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. In *AAAI*, 2020.

[31] Minghao Zhu, Xiao Lin, Ronghao Dang, Chengju Liu, and Qijun Chen. Fine-grained spatiotemporal motion alignment for contrastive video representation learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 4725–4736, New York, NY, USA, 2023. Association for Computing Machinery.

[32] S. Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *ECCV*, 2020.

[33] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction, 2019.

[34] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, June 2020.

[35] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI*, 2020.

[36] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, page 504–521, Berlin, Heidelberg, 2020. Springer-Verlag.

[37] Y. Yao, C. Liu, D. Luo, Y. Zhou, and Q. Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, pages 6547–6556, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.

[38] Hyeon Cho, Taehoon Kim, Hyung Chang, and Wonjun Hwang. Self-supervised visual learning by variable playback speeds prediction of a video. *IEEE Access*, PP:1–1, 05 2021.

[39] Ishan Misra, C. Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, volume 9905, pages 527–544, 10 2016.

[40] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, pages 5729–5738, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[41] Tomoyuki Suzuki, Takahiro Itazuri, Kensho Hara, and Hirokatsu Kataoka. Learning spatiotemporal 3d convolution with video order self-supervision. In *ECCV Workshops*, 2018.

[42] Ishan Rajendrakumar Dave, Simon Jenni, and Mubarak Shah. No more shortcuts: Realizing the potential of temporal self-supervision, 2023.

[43] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequence. In *ICCV*, 2017.

[44] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, June 2019.

[45] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[46] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189, 2018.

[47] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[48] X. Sun, P. Chen, L. Chen, C. Li, T. H. Li, M. Tan, and C. Gan. Masked motion encoding for self-supervised video representation learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2235–2245, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society.

[49] David Fan, Jue Wang, Leo Liao, Yi Zhu, Vimal Bhat, Hector Santos, Rohith Mysore Vijaya Kumar, and Xinyu (Arthur) Li. Motion-guided masking for spatiotemporal representation learning. In *ICCV 2023*, 2023.

[50] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[51] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăaucean, Florent Altché, Michal Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning. In *ICCV*, pages 1235–1245, 2021.

[52] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, NeurIPS'18, page 7774–7785, Red Hook, NY, USA, 2018. Curran Associates Inc.

[53] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *ArXiv*, abs/2001.05691, 2020.

[54] A. Miech, J. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9876–9886, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society.

[55] Peng Jin, Jinfa Huang, Fenglin Liu, Xian Wu, Shen Ge, Guoli Song, David A. Clifton, and Jie Chen. Expectation-maximization contrastive learning for compact video-and-language representations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *NeurIPS*, 2022.

[56] Rui Li, Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Motion-focused contrastive learning of video representations*. *ICCV*, 2021.

[57] Jingcheng Ni, Nana Zhou, Jie Qin, Qianrun Wu, Junqi Liu, Boxun Li, and Di Huang. Motion sensitive contrastive learning for self-supervised video representation. In *ECCV*, 2022.

[58] X. Sui, S. Li, X. Geng, Y. Wu, X. Xu, Y. Liu, R. Goh, and H. Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *CVPR*, pages 17581–17590, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.

[59] Kecheng Zheng, Yang Cao, Kai Zhu, Ruijing Zhao, and Zhengjun Zha. Famlp: A frequency-aware mlp-like architecture for domain generalization. *ArXiv*, abs/2203.12893, 2022.

[60] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14378–14387, 2021.

[61] Ju-Hyeon Nam and Sang-Chul Lee. Frequency filtering for data augmentation in x-ray image classification. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 81–85, 2021.

[62] Jinhyung Kim, Taeoh Kim, Minho Shim, Dongyoon Han, Dongyoon Wee, and Junmo Kim. Frequency selective augmentation for video representation learning. In *AAAI*, 2022.

[63] Haoxin Li, Yuan Liu, Hanwang Zhang, and Boyang Li. Mitigating and evaluating static bias of action representations in the background and the foreground, 2023.

[64] Lei Wang, Xiuyuan Yuan, Tom Gedeon, and Liang Zheng. Taylor videos for action recognition. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[65] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *AAAI*, 2020.

[66] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1483–1492, 2019.

[67] Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised learning from video with deep neural embeddings. In *CVPR*, June 2020.

[68] Hanwen Liang, Niamul Quader, Zhixiang Chi, Lizhe Chen, Peng Dai, Juwei Lu, and Yang Wang. Self-supervised spatiotemporal representation learning by exploiting video continuity. *AAAI*, 36(2):1564–1573, Jun. 2022.

[69] Wei Li, Dezhao Luo, Bo Fang, Yu Zhou, and Weiping Wang. Video 3d sampling for self-supervised representation learning. *ArXiv*, abs/2107.03578, 2021.

[70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[71] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

[72] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017.

[73] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[74] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[75] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

Appendix

Algorithm 1: Differentiation at Each **Batch**

Data: $\texttt{epoch} \geq 0, (\boldsymbol{V}, \boldsymbol{V}')$
Result: $(\boldsymbol{X}, \boldsymbol{X}')$
`// Deterministic differentiation step`
if $\mathit{epoch}\%4 = 0$ **then** $(\boldsymbol{X}, \boldsymbol{X}') \leftarrow (\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V}'}{\partial t})$;
else if $\mathit{epoch}\%4 = 1$ **then** $(\boldsymbol{X}, \boldsymbol{X}') \leftarrow (\frac{\partial \boldsymbol{V}}{\partial t}, \boldsymbol{V}')$;
else if $\mathit{epoch}\%4 = 2$ **then** $(\boldsymbol{X}, \boldsymbol{X}') \leftarrow (\boldsymbol{V}, \frac{\partial \boldsymbol{V}'}{\partial t})$;
else $(\boldsymbol{X}, \boldsymbol{X}') \leftarrow (\boldsymbol{V}, \boldsymbol{V}')$;
`// Additional random differentiation step`
$\epsilon \leftarrow \texttt{rand}(0, 1)$;
if $\epsilon < 0.5$ **then** $(\boldsymbol{X}, \boldsymbol{X}') \leftarrow (\frac{\partial \boldsymbol{X}}{\partial t}, \frac{\partial \boldsymbol{X}'}{\partial t})$;

A Relationship to Prior Arts

Current methods for SSL of video representations mainly utilize instance discrimination, pretext tasks, multimodal learning, and other ones. In the following, we discuss these results, and highlight the advantages of our method over existing ones.

Instance discrimination. It is first applied to images [10, 11, 12, 14] and then to videos [25, 26, 8, 3, 27, 7, 28]. Given either images or videos as instances, models learn to discriminate different instances versus different "views" of the same instance, where the views are generated by spatio-temporal augmentations [25, 8]. The learning process is driven by contrastive learning [8], clustering [25], or teacher-student distillation [26]. Recognizing the rich dynamics in videos, some prior works have further modified the loss function to consider each video's temporal attributes, such as play speed [27], time differences [8, 29, 9, 26], frame order [30], and motion diversity [9, 31]. Such modifications do not fully capture the essence of videos as reflections of continuous real-world dynamics, and are usually designed for a specific instance discrimination method. Apart from being applicable to different frameworks of instance discrimination, our approach, is new for its extraction of continuous dynamics by unfolding a video's hidden views via Taylor expansion and temporal differentiation.

Pretext tasks. Another category of methods involves creating learning tasks from videos. These tasks have many possible variations, such as identifying transformations applied to videos [32, 33], predicting the speed of videos [34, 35, 36, 37, 38], identifying incorrect ordering of frames or clips [39, 40, 41, 42], resorting them in order [43, 44], and solving space-time puzzles [45, 46]. The above methods usually require a complex combination of different tasks to learn general representations, while some recent works utilize large transformer backbones and learn by reconstructing masked areas [47] and further incorporating motion guidance into masking or reconstruction [48, 49]. These tasks provide non-trivial challenges for models to learn but are unlikely to reflect the natural processes through which humans and machines alike may learn and interpret dynamic visual information. In contrast, ViDiDi uses simple learning objectives and models how the physical world can be intuitively processed and understood without the need for complex tasks.

Others. In addition, prior methods also learn to align videos with other modalities, such as audio tracks [50, 51, 52], video captions [50, 53, 54, 55], and optical flows [7, 51, 56, 57, 58]. Optical flow also models changes between frames and is related to our method. However, our method is easy to calculate and intuitively generalize to higher orders of motion and guides the learning of the original frames within the same encoder in contrast to an additional encoder for optical flow [7]. Besides, our temporal differentiation strategy may be flexibly adapted to other dynamic data such as audio while optical flow explicitly models the movements of pixels. Incidentally, our proposed *balanced alternating learning strategy* as a simple yet novel way of learning different types of data, may inspire multimodal learning. Some other existing works manipulate frequency content to create augmented views of images [59, 60, 61] or videos [62, 63, 56] to make models more robust to out-of-domain

Table 4: **ViDiDi surpasses previous works on action recognition after finetuning.**

Method	Net	Input	Pretrained	UCF	HMDB
VCOP [44]	R3D-18	16 × 112	UCF101	64.9	29.5
VCP [65]	R3D-18	16 × 112	UCF101	66.0	31.5
3D-RotNet [33]	R3D-18	16 × 112	K600	66.0	37.1
DPC [66]	R3D-18	25 × 128	K400	68.2	34.5
VideoMoCo [25]	R3D-18	32 × 112	K400	74.1	43.6
RTT [32]	R3D-18	16 × 112	K600	79.3	49.8
VIE [67]	R3D-18	16 × 112	K400	72.3	44.8
RSPNet [35]	R3D-18	16 × 112	K400	74.3	41.8
VTHCL [27]	R3D-18	8 × 224	K400	80.6	48.6
CPNet [68]	R3D-18	16 × 112	K400	80.8	52.8
CPNet [68]	R3D-18	16 × 112	UCF101	77.2	46.3
CACL [3]	T+C3D	16 × 112	K400	77.5	-
CACL [3]	T+R3D	16 × 112	UCF101	77.5	43.8
TCLR [9]	R3D-18	16 × 112	UCF101	82.4	52.9
ViDiDi-BYOL	R3D-18	16 × 112	UCF101	**83.4**	**58.0**
ViDiDi-VIC	R3D-18	16 × 112	UCF101	**82.3**	**53.4**
ViDiDi-VIC	R3D-18	16 × 112	K200-40k	**82.7**	**54.2**
ViDiDi-VIC	R3D-18	16 × 112	K400	**83.2**	**55.8**
VCOP [44]	R(2+1)D-18	16 × 112	UCF101	72.4	30.9
VCP [65]	R(2+1)D-18	16 × 112	UCF101	66.3	32.2
PacePred [36]	R(2+1)D-18	16 × 112	K400	77.1	36.6
VideoMoCo [25]	R(2+1)D-18	32 × 112	K400	78.7	49.2
V3S [69]	R(2+1)D-18	16 × 112	K400	79.2	40.4
RSPNet [35]	R(2+1)D-18	16 × 112	K400	81.1	44.6
RTT [32]	R(2+1)D-18	16 × 112	UCF101	81.6	46.4
CPNet [68]	R(2+1)D-18	16 × 112	UCF101	81.8	51.2
CACL [3]	T+R(2+1)D	16 × 112	UCF101	82.5	48.8
ViDiDi-VIC	R(2+1)D-18	16 × 112	UCF101	**83.0**	**54.9**

data. Related to but unlike these works, we seek a fundamental and computationally efficient strategy to construct views from videos that reflect the continuous nature of real-world dynamics. Besides, apart from a new way of processing data, we propose an alternating learning strategy that is pivotal to boosting learning and has not been explored in previous works. Upon wrapping up our work, we noticed a concurrent work [64] recovers views from videos inspired by the Taylor series expansion, but from a complementary perspective, where derivatives in different orders are combined together. Besides, they utilize the augmented input in a supervised way, while ours focuses on a generalizable self-supervised framework including creating new views and a learning strategy.

B Comparisons with Previous Works

Results on both video retrieval (table 5) and action recognition (table 4) suggest that ViDiDi outperforms prior models. Compared to other models trained on Kinetics, ViDiDi-VIC achieves the highest accuracy using K400, while also reaching compatible performance using UCF101 or K200-40k subset for pretraining. Besides, ViDiDi-BYOL achieves the best performance in action recognition on HMDB51, by a significant margin of 5.1% over the recent TCLR method [9]. Importantly, ViDiDi supports efficient use of data. Its performance gain is most significant in the scenario of training with smaller datasets. We discuss this with more details in the following section.

C Ablation Study

ViDiDi involves multiple methodological choices, including 1) the order of derivatives, 2) how to pair different orders of derivatives as the input to two-stream video encoders, and 3) how to prescribe the learning schedule over different pairings. We perform ablation studies to test each design choice.

For the order of derivatives, we consider up to the 2^{nd} derivative. For pairing, we consider pairing derivatives in the same order (1^{st} vs. 1^{st}, 2^{nd} vs. 2^{nd}, etc.) or between different orders (1^{st} vs. 0^{th}, 1^{st} vs. 2^{nd}, etc.), respectively. For scheduling, we consider either random vs. scheduled selection of input pairs. With random selection, temporal differentiation is essentially treated as additional data augmentation. In contrast, the scheduled selection (alg. 1) aims to provide a balanced and structured way for the model to learn from various orders of temporal derivatives, where higher order derivatives are intuitively used as guidance of learning the original frames.

Table 5: **ViDiDi surpasses previous SSL models on video retrieval.** T + C3D means training with an additional transformer.

Method	Net	Pretrained	UCF101			HMDB51		
			1	5	10	1	5	10
SpeedNet [34]	S3D-G	K400	13.0	28.1	37.5	-	-	-
RTT [32]	R3D-18	K600	26.1	48.5	59.1	-	-	-
RSPNet [35]	R3D-18	K400	41.1	59.4	68.4	-	-	-
CoCLR [7]	S3D	K400	46.3	62.8	69.5	20.6	43.0	54.0
CACL [3]	T+C3D	K400	44.2	63.1	71.9	-	-	-
ViDiDi-VIC	R3D-18	K200-40k	**49.5**	**63.4**	**71.0**	**24.7**	**45.4**	**56.0**
ViDiDi-VIC	R3D-18	K400	**51.2**	**64.6**	**72.6**	**25.0**	**47.2**	**60.9**
VCOP [44]	R3D-18	UCF101	14.1	30.3	40.0	7.6	22.9	34.4
VCP [65]	R3D-18	UCF101	18.6	33.6	42.5	7.6	24.4	33.6
PacePred [36]	R3D-18	UCF101	23.8	38.1	46.4	9.6	26.9	41.1
PRP [37]	R3D-18	UCF101	22.8	38.5	46.7	8.2	25.8	38.5
V3S [69]	R3D-18	UCF101	28.3	43.7	51.3	10.8	30.6	42.3
CACL [3]	T+R3D	UCF101	41.1	59.2	67.3	17.6	36.7	48.4
ViDiDi-VIC	R3D-18	UCF101	**47.6**	**60.9**	**68.6**	**19.7**	**40.5**	**55.1**

As shown in table 6, results demonstrate a progressive improvement in the model's performance in video retrieval, given a higher order of derivatives (from the 1^{st} to 2^{nd} order), given mixed pairing, and given scheduled selection of input pairs. Therefore, temporal differentiation is not merely another data augmentation trick. Invariance to different orders of temporal derivatives is a valuable principle for SSL of video representations that lead to better performance in downstream tasks. To leverage this principle, it is beneficial to design mixed pairing and prescribe a learning schedule that provides a balanced and holistic view of different orders of temporal dynamics inherent to videos. Details about how we design the groups of models in table 6 are summarized below:

- **Base**: The direct extension of VICReg.
- **+Random** 1^{st}: Add 1^{st} order derivatives as random augmentation.
- **+Random** 1^{st} **&** 2^{nd}: Add 1^{st} and 2^{nd} order derivatives as random augmentation.
- **Reverse ViDiDi-VIC**: Reverse the order of pair alternation by epoch in ViDiDi, i.e., line 1-9 in alg. 1.
- **+Schedule** 1^{st}: Alternate pairs across epochs in the order $(\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V}'}{\partial t}) \rightarrow (\boldsymbol{V}, \boldsymbol{V}') \rightarrow (\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V}'}{\partial t}) \rightarrow \ldots.$
- **+Schedule** 1^{st} **& Mix**: switch pairs by epoch in the order $(\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V}'}{\partial t}) \rightarrow (\frac{\partial \boldsymbol{V}}{\partial t}, \boldsymbol{V}') \rightarrow (\boldsymbol{V}, \boldsymbol{V}') \rightarrow (\frac{\partial \boldsymbol{V}}{\partial t}, \frac{\partial \boldsymbol{V}'}{\partial t}) \rightarrow \ldots.$
- **+Schedule** 1^{st} **&** 2^{nd} and **+Schedule** 1^{st} **&** 2^{nd} **& Mix**: Build upon +Schedule 1^{st} and +Schedule 1^{st} & Mix accordingly with random differentiation at each batch to utilize 2^{nd} order derivatives.

Table 6: **Ablation Study.** Video retrieval performance on UCF101 with different design choices.

Method	UCF101		
	1	5	10
Base (VICReg)	31.1	43.6	50.9
+Random 1^{st}	35.2	47.7	56.1
+Random 1^{st} & 2^{nd}	36.2	48.6	55.8
Reverse ViDiDi-VIC	39.1	54.7	62.9
+Schedule 1^{st}	37.1	50.3	58.2
+Schedule 1^{st} & Mix	39.3	53.2	60.9
+Schedule 1^{st} & 2^{nd}	40.7	56.5	64.0
+Schedule 1^{st} & 2^{nd} & Mix	43.0	59.2	66.6
ViDiDi-VIC	**47.6**	**60.9**	**68.6**

D Details of SimCLR, BYOL and VICReg

In this section, we provide more details on how we plug ViDiDi into different instance discrimination frameworks: SimCLR [10], BYOL [11], VICReg [13].

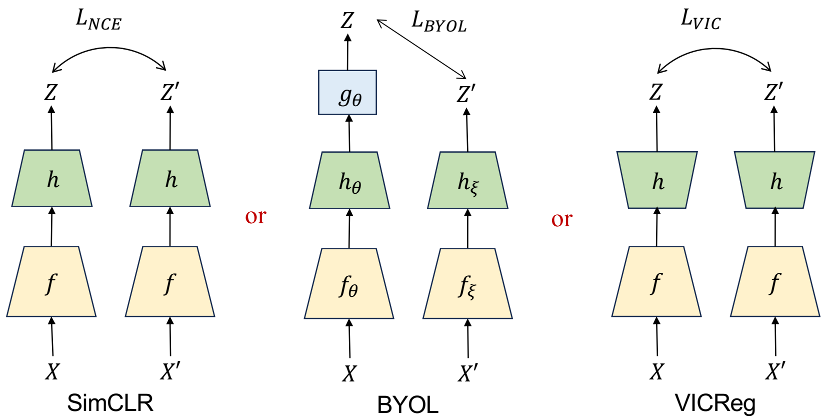


Figure 5: **The SimCLR, BYOL, VICReg details.**

D.1 Notation

The summary of the SimCLR, BYOL, and VICReg are shown in fig. 5. We begin by introducing the notations. (X, X') represents two batches of input to the discrimination framework, both in the shape $\mathbb{R}^{B\times C\times T\times h\times w}$, containing B clips (or derivatives) of length T, and size $h \times w$. (Z, Z') denotes two batches of latents encoded from (X, X'), in the shape of $\mathbb{R}^{B\times D}$, containing latents of dimension D for B clips. $Z = [z_1, \ldots, z_B]^T$ and $Z' = [z'_1, \ldots, z'_B]^T$, expressed as collects of column vectors. f represents the encoder, which is a 3D convolutional neural network in our experiments. h serves as the projector, either shrinking or expanding output dimensionality. g denotes the predictor. h and g are both realized as multi-layer perceptrons (MLPs). We also introduce the similarity function: $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}'_j / \left(\|\boldsymbol{z}_i\| \left\|\boldsymbol{z}'_j\right\|\right)$.

D.2 SimCLR

SimCLR [10] is a contrastive learning framework, whose key idea is to contrast dissimilar instances in the latent space. As shown in fig. 5, SimCLR uses a shared encoder f to process (X, X'), and then project the output with an MLP projection head h into (Z, Z'). $Z = h(f(X))$, $Z' = h(f(X'))$. The InfoNCE loss is defined as:

$$\mathcal{L}_{NCE} = \frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(s_{i,i}/\alpha)}{\sum_{j=1}^{B}\exp(s_{i,j}/\alpha)} + \frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(s_{i,i}/\alpha)}{\sum_{j=1}^{B}\exp(s_{j,i}/\alpha)} \tag{1}$$

D.3 BYOL

BYOL [11] is a teacher-student approach. It has an online encoder f_θ, an online projector h_θ, and a predictor g_θ, learned via gradient descent. BYOL uses stop gradient for a target encoder f_ξ and a target projector h_ξ, which are updated only by exponential moving average of the online ones $\xi \leftarrow \tau\xi + (1-\tau)\theta$ after each training step, where $\tau \in [0, 1]$ is the target decay rate. $Z = g_\theta(h_\theta(f_\theta(X)))$, $Z' = \mathrm{sg}(h_\xi(f_\xi(X')))$, here sg means stop gradient. The loss is defined as:

$$\mathcal{L}_{BYOL} = \frac{1}{2B}\sum_{i=1}^{B}(2 - 2s_{i,j}) \tag{2}$$

D.4 VICReg

VICReg [13] learns to discriminate different instances using direct variance, invariance, and co-variance regularization in the latent space. It also has a shared encoder f and a shared projector h. $Z = h(f(X))$, $Z' = h(f(X'))$. The invariance term is defined as:

$$s\left(Z, Z'\right) = \frac{1}{B} \sum_{i=1}^{B} \left\|z_i - z_i'\right\|_2^2 \tag{3}$$

The variance term constraints variance along each dimension to be at least γ, γ is a constant:

$$v(Z) = \frac{1}{D} \sum_{j=1}^{D} \max\left(0, \gamma - S\left(z^j, \epsilon\right)\right) \tag{4}$$

where S is the regularized standard deviation $S(x, \epsilon) = \sqrt{\mathrm{Var}(x) + \epsilon}$, ϵ is a small constant, z^j is the j^{th} row vector of Z^T, containing the value at j^{th} dimension for all latents in Z.

The covariance term constraints covariance of different dimensions to be 0:

$$c(Z) = \frac{1}{D} \sum_{i \neq j} [C(Z)]_{i,j}^2 \tag{5}$$

$C(Z) = \frac{1}{B-1} \sum_{i=1}^{B} \left(z_i - \bar{z}\right)\left(z_i - \bar{z}\right)^T$, $\bar{z} = \frac{1}{B} \sum_{i=1}^{B} z_i$.

The total loss is a weighted sum of invariance, variance, and covariance terms:

$$\begin{aligned} \mathcal{L}_{VIC} = & \lambda s\left(Z, Z'\right) + \mu\left[v(Z) + v\left(Z'\right)\right] \\ & + \nu\left[c(Z) + c\left(Z'\right)\right] \end{aligned} \tag{6}$$

E Implementation Details

E.1 Augmentation Details

We apply clipwise spatial augmentations as introduced in [8]. All the augmentations are applied before differentiation. For example, for a clip sampled from one video, we do a random crop on the first frame and crop all the other frames in the clip to the same area as the first frame. If a second clip is sampled, we do random crop on its first frame and crop the other frames to the same area. The original frames are extracted and resized to have a shorter edge of 150 pixels. The list of augmentations is as follows:

- Random Horizontal Flip, with probability 0.5;
- Random Sized Crop, with area scale uniformly sampled in the range $(0.08, 1)$, aspect ratio in $(\frac{3}{4}, \frac{4}{3})$, BILINEAR Interpolation, and output size 112×112;
- Gaussian Blur, with probability 0.5, kernal size $(3, 3)$, sigma range $(0.1, 2.0)$;
- Color Jitter, with probability 0.8, brightness 0.2, contrast 0.2, saturation 0.2, hue 0.05;
- Random Gray, with probability 0.5;
- Normalize, mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225].

E.2 Network Architecture

The output feature dimension for R3D-18, R(2+1)D-18, and MC3-18 is 512, while 1024 for S3D. In terms of the projector architecture, we use a 2-layer MLP in BYOL, and a 3-layer MLP in SimCLR and VICReg, as proposed by [10, 11, 13]. The output dimension of the projector is $d_{BYOL} = 256, d_{SimCLR} = 128, d_{VICReg} = 2048$, and the hidden dimension is $d_{BYOL} = 4096, d_{SimCLR} = 2048, d_{VICReg} = 2048$. The predictor for BYOL is a 2-layer MLP, with output dimension $d = 256$, and hidden dimension $d = 4096$. Batch normalization [70] and Rectified Linear Unit (ReLU) are applied for all hidden layers of projectors and predictors.

E.3 Pretraining

UCF101, K400, or K200-40k is used as the pertaining dataset. We train the model for 400 epochs on UCF101 or K200-40k, and K400. We set $T = 8$, and select 1 frame every 3 frames. The learning rate follows a cosine decay schedule [71] for all frameworks. The learning rate at k_{th} iteration is $\eta \cdot 0.5 \left[\cos\left(\frac{k}{K}\pi\right) + 1\right]$, where K is the maximum number of iterations and η is the base learning rate. A 10-epoch warmup is only employed for BYOL. Weight decay is set as $1e-6$. We apply cosine-annealing of the momentum for BYOL as proposed in [11]: $\tau = 1 - (1 - \tau_{\text{base}}) \cdot (\cos(\frac{k}{K}\pi) + 1)/2$, and set $\tau_{base} = 0.99$. The temperature for SimCLR is $\alpha = 0.1$, and hyper-parameters for VICReg are $\lambda = 1.0, \mu = 1.0, \nu = 0.05$. We train all models with the LARS optimizer [72] utilizing a batch size of 64 for UCF101 or K200-40k, batch size of 256 for K400, and a base learning rate $\eta = 1.2$. The pretraining can be conducted on 8 GPUs, each having at least 12 GB of memory.

E.4 Video Retrieval

For the pretrained model without any fine-tuning, we test its performance on video retrieval using nearest-neighborhood in the feature space [3, 7]. Specifically, given a video, we uniformly sample 10 clips of length 16, apply random crop and normalization for data augmentation, encode each clip using the pretrained video encoder, and average the resulting representations into a single feature vector for encoding the given video. Through a nearest-neighborhood model that fits the training set, we use each video in the testing set as a query and retrieve the top-k ($k = 1, 5, 10$) closest videos in the training set. The retrieval is successful if at least one out of the k retrieved training videos is from the same class as the query video. We report the top-k retrieval recall on UCF101 and HMDB51. The retrieval can be conducted on 1 GPU, having at least 24 GB of memory.

E.5 Action Recognition

We also fine-tune the pretrained model to classify human actions. For this purpose, we add a linear classification head to the pretrained model, and fine-tune it end-to-end on UCF101 or HMDB51 for 100 epochs (see more details in the supplementary material). At training, we sample clips of length 16. We use the SGD optimizer [73] with a momentum value of 0.9. The model is tuned for 100 epochs. The batch size is set at 128, with an initial learning rate of 0.2 which is scaled by $\frac{1}{10}$ at the 60th and 80th epochs. We use a weight decay of $1e-4$. Furthermore, a dropout rate of 0.5 is applied. After fine-tuning, we sample 10 clips of length 16 from each testing video, apply random crop and normalization, feed the results as the input to the fine-tuned model, and average their resulting predictions for the final classification of the video. We report the top-1 action recognition accuracy on UCF101 and HMDB51. The finetuning can be conducted on 8 GPUs, each having at least 12 GB of memory. The testing can be conducted on 1 GPU.

E.6 Action Detection

We mainly follow the CVRL [8] testing pipeline, taking our pre-trained R3D-18 as the backbone and casting a Faster-RCNN [74] on top of it. To fit the time-sequential nature of the input, we extract region-of-interest (RoI) features using a 3D RoIAlign on the output from the final convolutional block. These features are then processed through temporal average pooling and spatial max pooling. The resulting feature is fed into a sigmoid-based classifier for multi-label prediction. We pretrain our R3D-18 with three different methods(VIC/BYOL/SimCLR) and two different inputs (with/without derivative). We use an AdamW[75] optimizer with a 0.01 learning rate, then shrink the learning rate to half after epoch 5. The dropout rate for Faster-RCNN is 0.5. We perform 20 epochs for our six pre-trained weights and run an evaluation after each epoch. We report the epoch with the highest mAP. Our clip length is eight frames with an interval of four frames. The finetuning can be conducted on 2 GPUs, each having at least 48 GB of memory. The testing can be conducted on 1 GPU.

F Auxilary Results

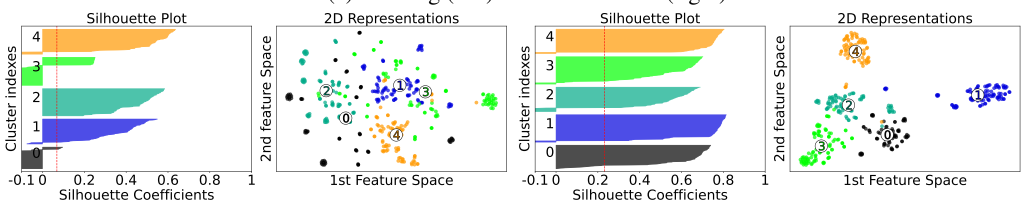
F.1 Silhouette Score

Apart from visualization of clustering in the latent space, we also quantify the clustering using Silhouette Score as illustrated in 7.

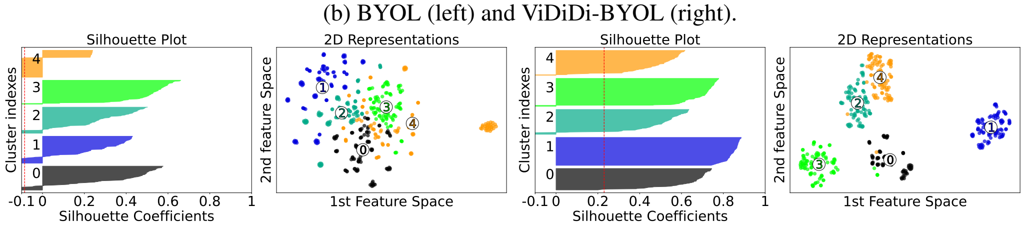
Table 7: **Silhouette Score** for Base and ViDiDi with 3, 5, . . . , 101 classes. ViDiDi improves the Score, showing better clustering in the latent space.

Method	Silhouette Score					
	3	5	10	15	20	101
SimCLR	0.136	0.081	0.048	0.034	0.022	-0.026
ViDiDi-SimCLR	**0.210**	**0.132**	**0.096**	**0.078**	**0.058**	**0.003**
BYOL	0.038	-0.086	-0.094	0.091	-0.080	-0.186
ViDiDi-BYOL	**0.185**	**0.230**	**0.128**	**0.107**	**0.070**	**0.004**
VICReg	0.110	0.069	0.044	0.036	0.017	-0.038
ViDiDi-VIC	**0.235**	**0.232**	**0.150**	**0.138**	**0.098**	**0.014**

(a) VIDReg (left) and ViDiDi-VIC (right).



(b) BYOL (left) and ViDiDi-BYOL (right).



(c) SimCLR (left) and ViDiDi-SimCLR (right).

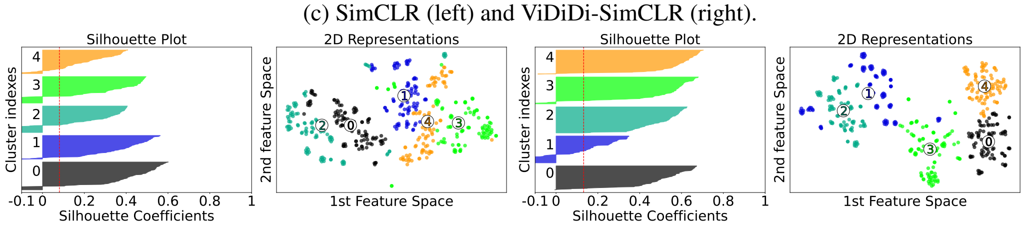
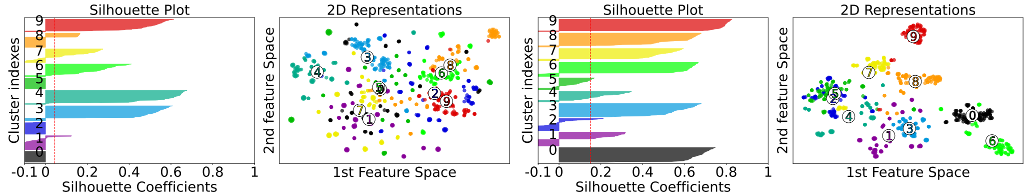


Figure 6: **Silhouette scores and t-SNE plots of top 5 classes in UCF101 train**.

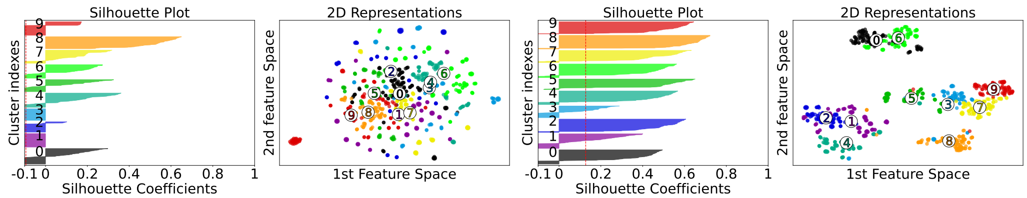
F.2 Clustering of Latent Space

We provide more visualization of the clustering phenomenon for VICReg, BYOL, and SimCLR, with or without the ViDiDi framework; on UCF101 train dataset or test dataset; utilizing 5 or 10 classes of videos. Here, for each model, we choose the top 5 or 10 classes of videos that are best retrieved during the video retrieval experiments. The results are shown in fig. 6, fig. 7, fig. 8, and fig. 9. ViDiDi provides consistently better clustering in the latent space for both train data and test data.

(a) VIDReg (left) and ViDiDi-VIC (right).



(b) BYOL (left) and ViDiDi-BYOL (right).



(c) SimCLR (left) and ViDiDi-SimCLR (right).

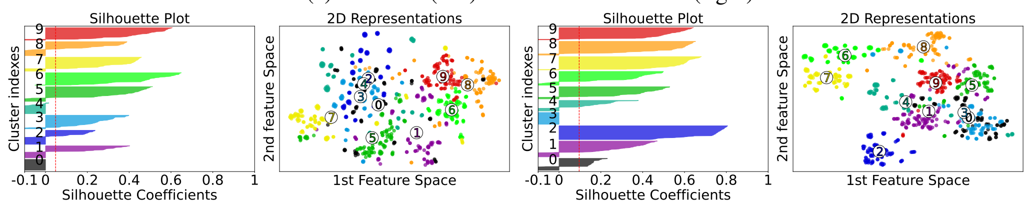
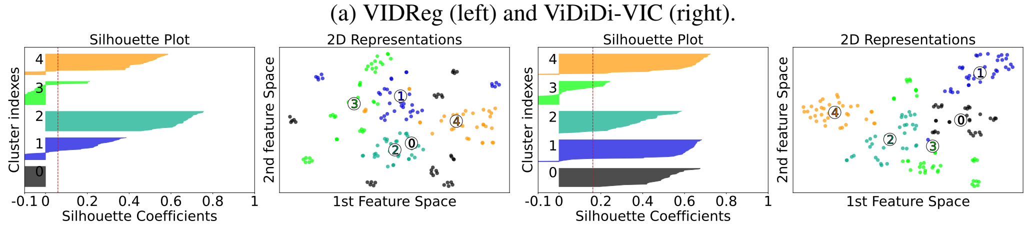
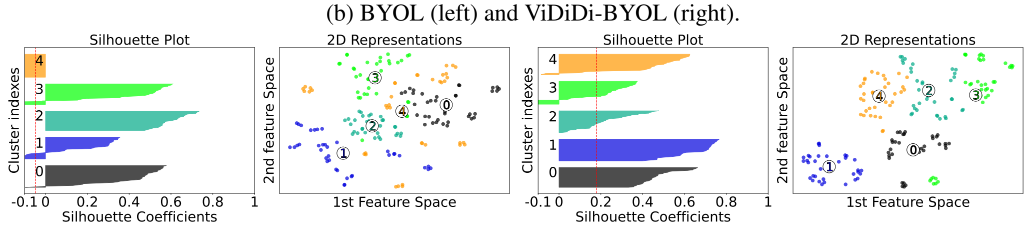


Figure 7: **Silhouette scores and t-SNE plots of top 10 classes in UCF101 train.**

(a) VIDReg (left) and ViDiDi-VIC (right).



(b) BYOL (left) and ViDiDi-BYOL (right).



(c) SimCLR (left) and ViDiDi-SimCLR (right).

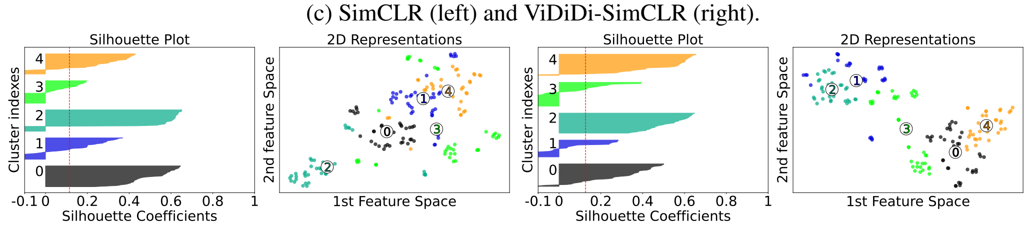
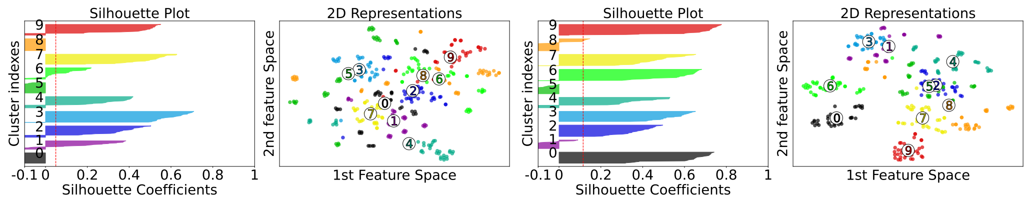
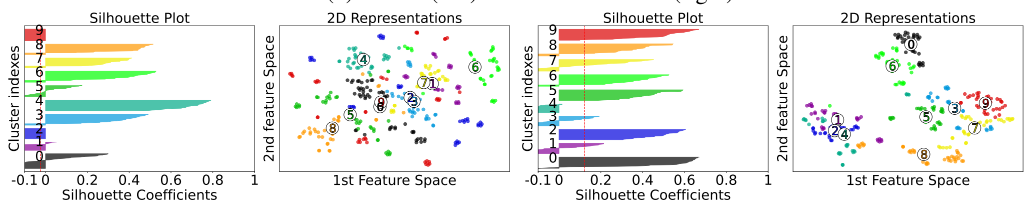


Figure 8: **Silhouette scores and t-SNE plots of top 5 classes in UCF101 test.**

(a) VIDReg (left) and ViDiDi-VIC (right).



(b) BYOL (left) and ViDiDi-BYOL (right).



(c) SimCLR (left) and ViDiDi-SimCLR (right).

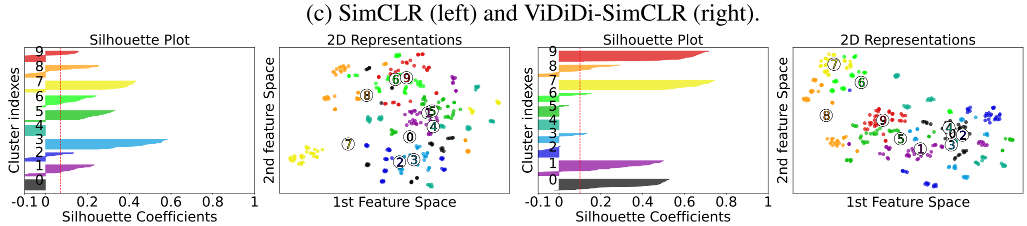


Figure 9: **Silhouette scores and t-SNE plots of top 10 classes in UCF101 test**.

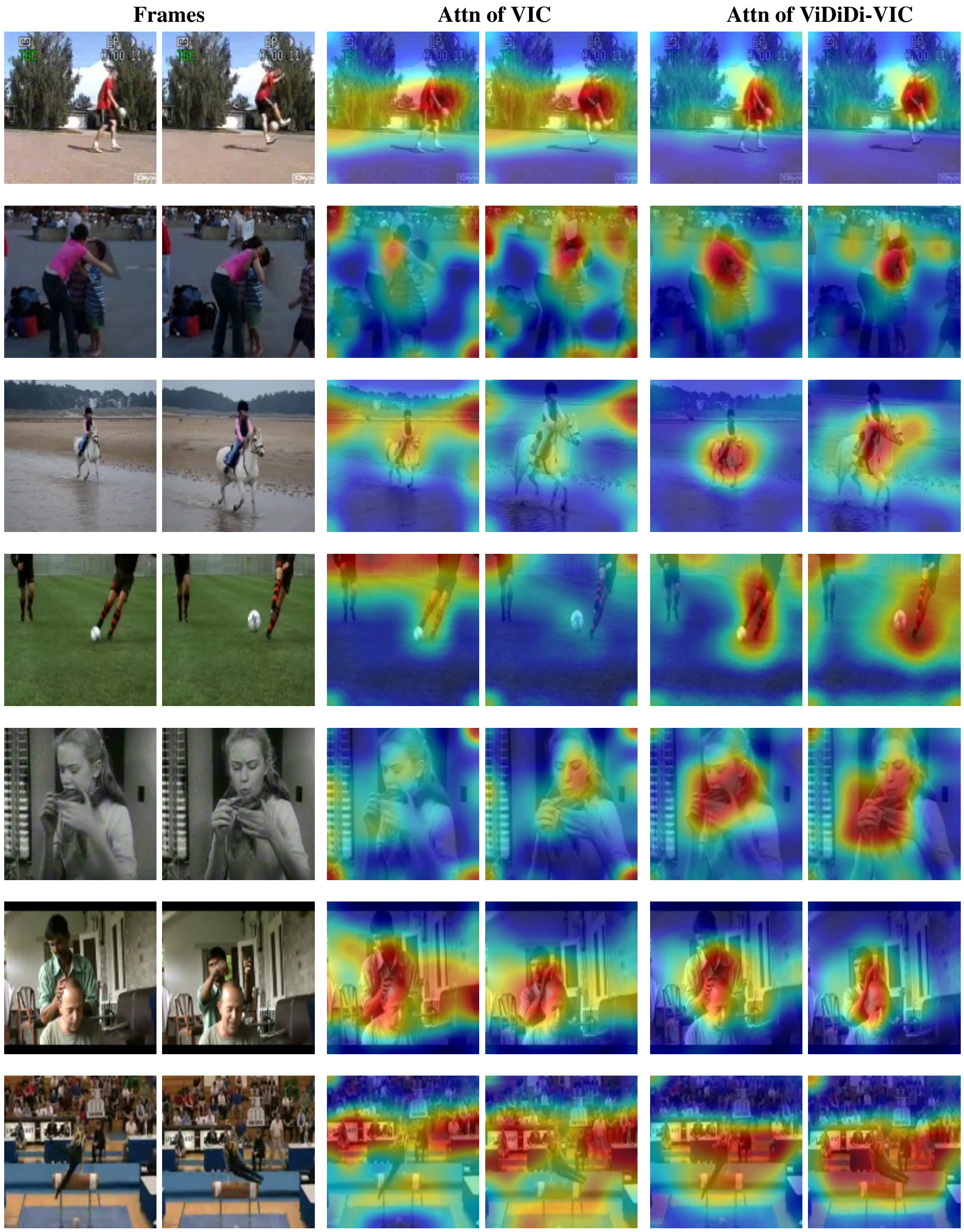


Figure 10: **More spatiotemporal attention for VICReg and ViDiDi-VIC.** Left: Original frames. Middle: Attention from VIC. Right: Attention from ViDiDi-VIC.

F.3 Spatio-temporal Attention

We provide more visualization of the attention for VICReg, BYOL, and SimCLR, with or without the ViDiDi framework; on UCF101 dataset or HMDB51 dataset. The results are presented in fig. 10, fig. 11, fig. 12, fig. 13, and fig. 14.

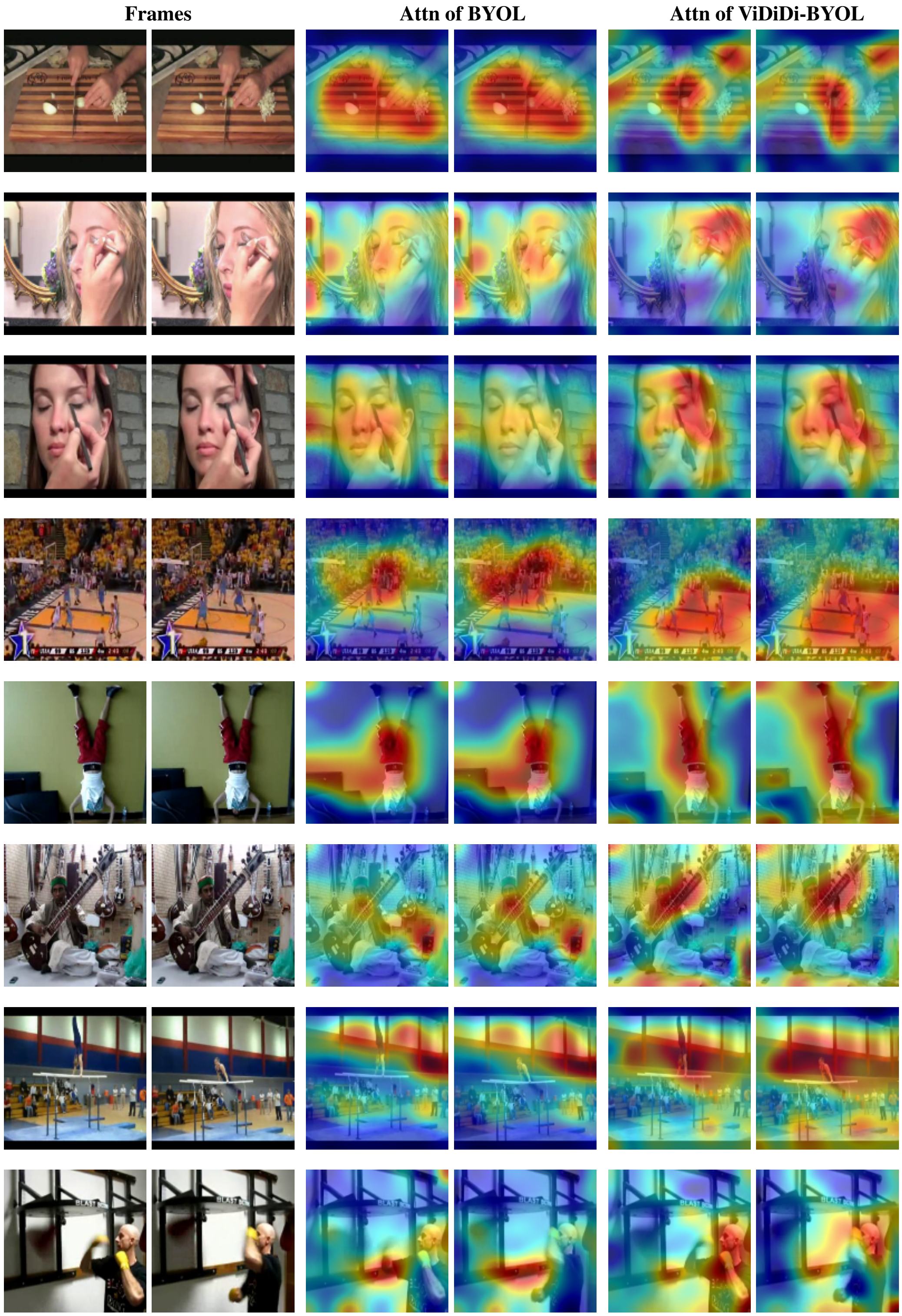


Figure 11: **Spatiotemporal attention on UCF101.** Left: Original frames. Middle: Attention from BYOL. Right: Attention from ViDiDi-BYOL.

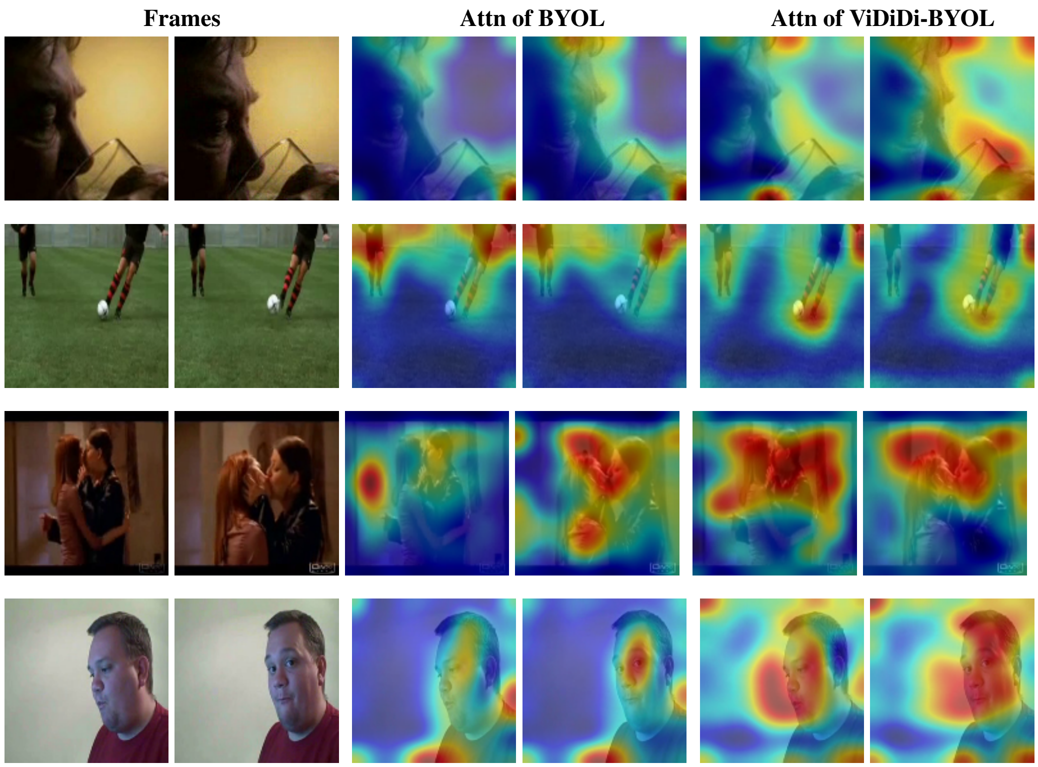


Figure 12: **Spatiotemporal attention on HMDB51.** Left: Original frames. Middle: Attention from BYOL. Right: Attention from ViDiDi-BYOL.

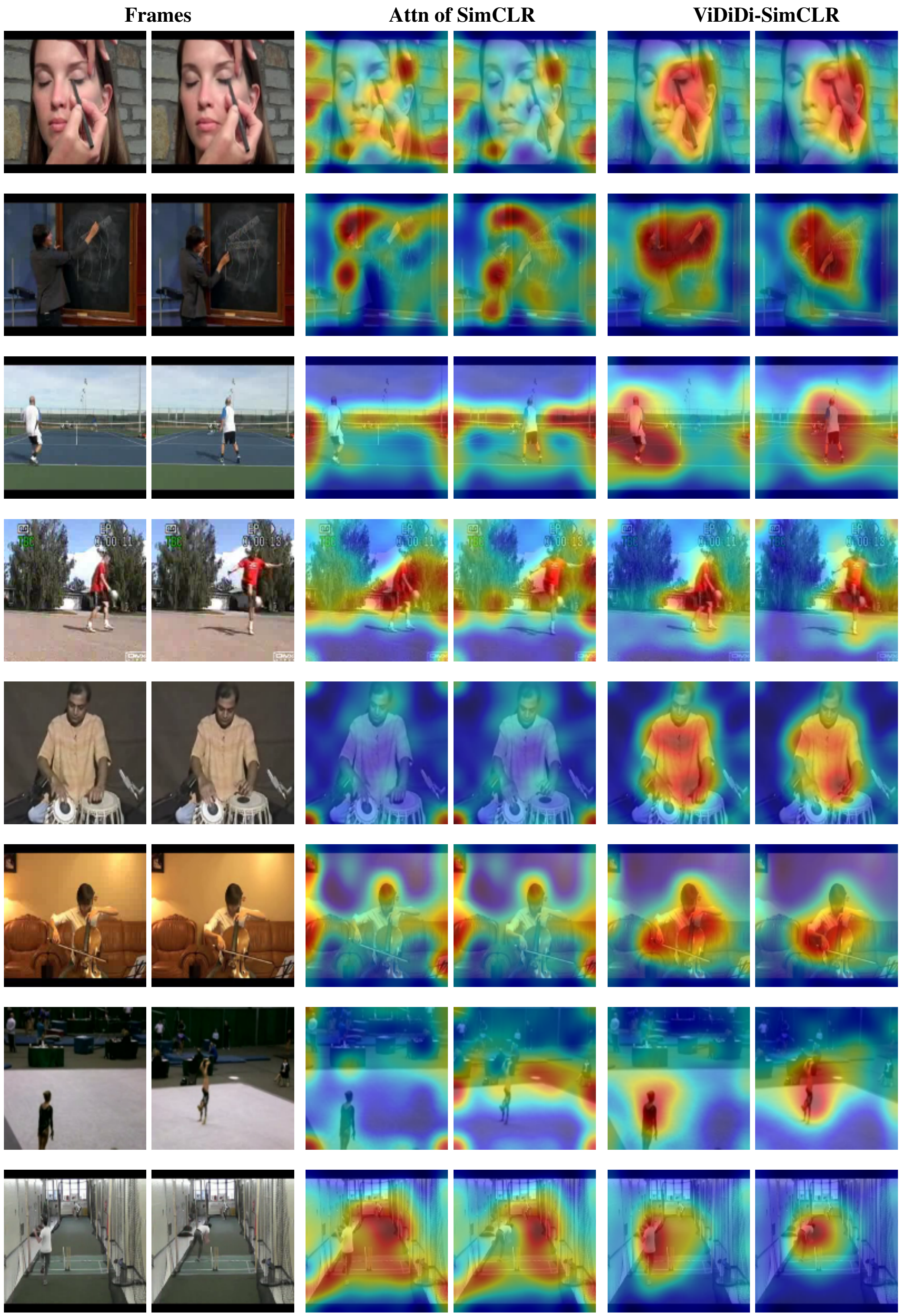


Figure 13: **Spatiotemporal attention on UCF101.** Left: Original frames. Middle: Attention from SimCLR. Right: Attention from ViDiDi-SimCLR.

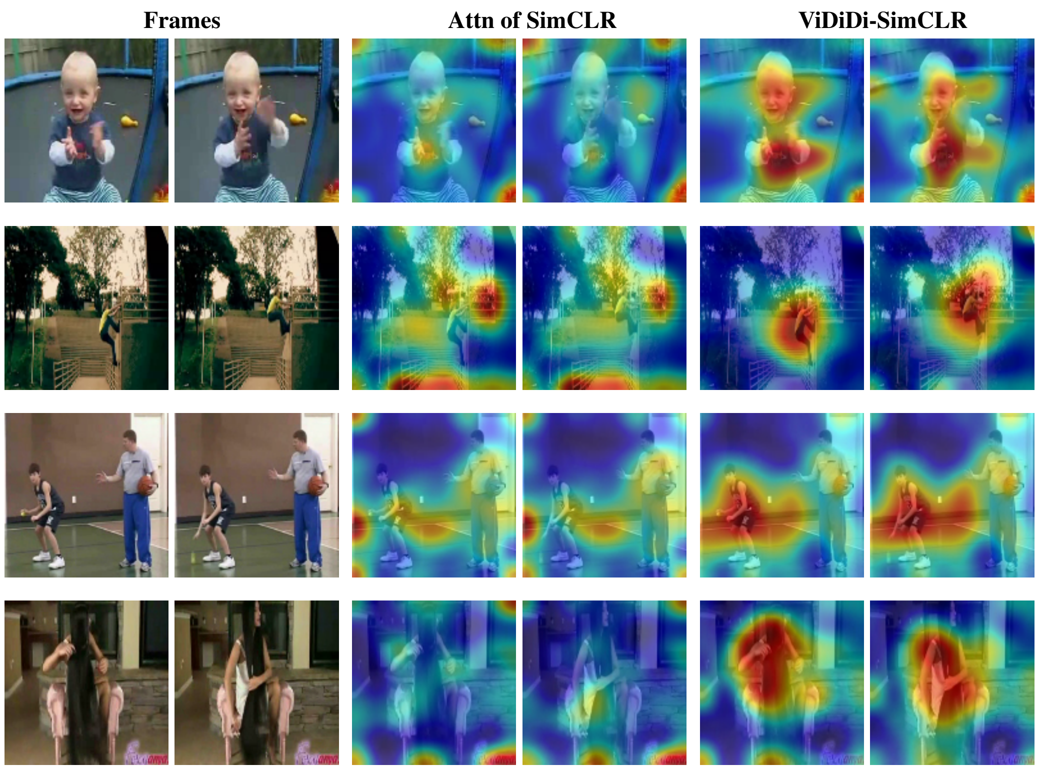


Figure 14: **Spatiotemporal attention on HMDB51.** Left: Original frames. Middle: Attention from SimCLR. Right: Attention from ViDiDi-SimCLR.