Supplementary: SPAct: Self-supervised Privacy Preservation for Action Recognition

Ishan Rajendrakumar Dave, Chen Chen, Mubarak Shah
Center for Research in Computer Vision, University of Central Florida, Orlando, USA
ishandave@knights.ucf.edu, {chen.chen, shah}@crcv.ucf.edu

A. Overview

The supplementary material is organized into the following sections:

- Section B: Dataset details
- Section C: Implementation details such as network architectures, data augmentations, training setup, baseline implementation details, performance metrics.
- Section D: Evaluating learned anonymization function in various target settings.
- Section E: Additional ablation experiments.
- Section F: Qualitative results of the learned anonymization
- Section G: Visual Aid to understand the training and evaluation protocol

B. Datasets

UCF101 [14] has around 13,320 videos representing 101 different human activities. All results in this paper are reported on split-1, which has 9,537 train videos and 3,783 test videos.
HMDB51 [10] is a relatively smaller action recognition dataset having 6,849 total videos collected from 51 different human actions. All results in this paper are reported on split-1, which has 3,570 train videos and 1,530 test videos.
VISPR [12] is an image dataset with a diverse set of personal information in an image like skin color, face, gender, clothing, document information etc. We use two subsets of privacy attributes of VISPR dataset as shown in Table 1. Each of the privacy attribute is a binary label, where 0 indicates absence of the attribute and 1 indicates presence of the attribute in the image. An image can have multiple privacy attributes, hence it is as a multi-label classification problem.

PA-HMDB51 [17] is subset of HMDB51 dataset with 51 action labels and 6 human privacy attributes which are annotated temporally. The privacy attributes are the same as VISPR-1 subset shown in Table 1 except `a65_rel_soci` attribute. Each privacy attribute has a fine-grained class assigned as well, however, it is not considered in this paper. Following [17], we use binary label for each privacy attribute i.e. if the privacy attribute is present in the image or not.

VISPR1 [17]	VISPR2
a17_color	a6_hair_color
a4_gender	a16_race
a9_face_complete	a59_sports
a10_face_partial	a1_age_approx
a12_semi_nudity	a2_weight_approx
a64_rel_personal	a73_landmark
a65_rel_soci	a11_tattoo

Table 1. Privacy attributes of VISPR [12] subsets.

P-HVU is a selected subset of LSHVU [4], which is a large-scale dataset of multi-label human action with a diverse set of auxiliary annotations provided for objects, scenes, concepts, events etc. We consider using this dataset to understand privacy leakage in terms of object or scene. P-HVU is prepared from LSHVU dataset such that each video has object and scene annotations along with the action label. A video of the LSHVU always has action labels, however, it does not necessarily have scene and object label. We consider following steps to prepare P-HVU dataset:

- Select all LSHVU *validation* set videos such that each video has object and scene annotation and call it *P-HVU test set*.
- Select LSHVU *train* set videos which has action, object and privacy class from the P-HVU test set, and filter out videos if either of the object or scene annotations are missing in the video and call it *P-HVU train set*.

Each video of the P-HVU dataset has multi-label action, object and scene annotation. The dataset consists of 739 action

classes, 1678 objects, and 248 scene categories. Train/test split of P-HVU consists of 245,212/16,012 videos to provide a robust evaluation.

C. Implementation Details

C.1. Architectural details

For anonymization function we utilize PyTorch implementation[1] of UNet [13] with three output channels. For 2D-CNN based ResNet [7], 3D-CNN models R3D-18 [6], and R2plus1D-18 [16], we utilize `torchvision.models` implementation[2]. Multi-layer projection head $\mathbf{g}(\cdot)$ of self-supervised privacy removal branch consists of 2 layers: Linear(2048, 2048) with ReLU activation and Linear(2048, 128) followed by L2-Normalization.

C.2. Augmentations

We apply two different sets of augmentation depending upon the loss function: (1) For supervised losses, we use standard augmentations like random crop, random scaling, horizontal flip and random gray-scale conversion with less strength. (2) For self-supervised loss, in addition to the standard augmentations with with more strength, we use: random color jitter, random cut-out and random color drop. For more details on augmentation strengths in supervised and self-supervised losses refer SimCLR [2]. In order to ensure temporal consistency in a clip, we apply the exact same augmentation on all frames of the clip. All video frames or images are resized to 112×112. Input videos are of 16 frames with skip rate of 2.

C.3. Hyperparameters

We use a base learning rate of 1e-3 with a learning rate scheduler which drops learning rate to its 1/10th value on the loss plateau.

For self-supervised privacy removal branch, we use the 128-D output as representation vector to compute contrastive loss of temperature $\tau = 0.1$. For RotNet [5] experiment we use 4 rotations: $\{0, 90, 180, 270\}$.

C.4. Training details

To optimize parameters of different neural networks we use Adam optimizer [9]. For initialization, we train f_A for 100 epochs using $\mathcal{L}_1$ reconstruction loss, action recognition auxiliary model f_T using cross-entropy loss for 150 epochs, and privacy auxiliary model f_B using NT-Xent loss for 400 epochs. Training phase of anonymization function f_A is carried out for 100 epochs, whereas target utility model f'_T and target privacy model f'_B are trained for 150 epochs.

[1] https://github.com/milesial/Pytorch-UNet

[2] https://github.com/pytorch/vision/tree/main/torchvision/models

C.5. Performance Metrics

To evaluate the performance of target privacy model f'_B we use macro-average of classwise mean average precision (cMAP). The results are also reported in average F1 score across privacy classes. F1 score for each class is computed at confidence 0.5. For action recognition, we use top-1 accuracy computed from video-level prediction from the model and groundtruth. A video-level prediction is average prediction of 10 equidistant clips from a video.

C.6. Baselines

Supervised adversarial framework [17]: we refer to official github repo[3] and with the consultation of authors we reproduce their method. For fair comparison, we use exact same model architectures and training augmentations. For more details on hyperparameters refer [17].

Blurring based obfuscation baselines: we first detect the person using MS-COCO [11] pretrained yolov5x [8] model in each frame of the video. After detecting the person bounding boxes, we apply Gaussian blur filter on the bounding boxes regions. We utilize `torchvision.transforms.GaussianBlur` function with kernel size = 21 and sigma = 10.0 for Strong blur, and kernel size = 13, sigma = 10.0 for the Weak blur baselines. For VISPR dataset, we first downsample images such that smaller side of image = 512.

Blackening based obfuscation baselines: we first detect person bounding boxes using yolov5x model and assign zero value to all RGB channels of the bounding box regions.

Blackening based obfuscation baselines: we first detect person bounding boxes using yolov5x model and assign zero value to all RGB channels of the bounding box regions.

Ablation with spatio-temporal privacy removal branch: For ablation of Table 3 of the main paper, we use naive extension of SimCLR [2] to the domain of video, where we consider two clips from the same video as positive and clips from other videos as negatives in the contrastive loss. R3D-18 is chosen as 3D-CNN backbone and MLP $\mathbf{g}(\cdot)$ consist of Linear(512, 512) with ReLU activation and Linear(512, 128) followed by L2-Normalization.

Noisy Features baseline [19]: Zhang *et al*. [19] proposed non-visual privacy preservation in wearable device from 1D singal of mobile sensors. We extended this work to video privacy by replacing LossNet to R3D-18, TransNet to UNet and extended similarity losses to handle video input.

Method	VISPR1		VISPR2		PA-HMDB	
	cMAP (%)(↓)	F1(↓)	cMAP (%)(↓)	F1(↓)	cMAP (%)(↓)	F1(↓)
Raw data	64.40	0.5553	57.60	0.4980	70.10	0.4010
Downsample-2×	51.23	0.4627	46.39	0.4330	60.04	0.2403
Downsample-4×	38.82	0.3633	33.42	0.3055	0.59	0.2630
Obf-Blackening	48.38	0.3493	44.01	0.3134	55.66	0.0642
Obf-StrongBlur	54.44	0.4440	50.31	0.3990	60.13	0.2830
Supervised [17]	22.81↓**65%**	0.2437↓56%	26.61↓54%	0.1840↓63%	57.01↓**19%**	0.2310↓42%
Ours	27.44↓57%	0.0760↓**86%**	20.02↓**65%**	0.0460↓**91%**	58.90↓16%	0.0940 ↓**77%**

Table 2. Evaluating learned anonymization function f_A^* to measure its privacy leakage from a **raw-data pretrained privacy target model** f'_B. Lower privacy classification score is better, ↓% denotes relative drop from raw data. Our self-supervised gets a competitive performance to the supervised method [17].

D. Additional results

D.1. Evaluating f_A^* privacy target model with f'_B pretrained on a raw data

In a practical scenario, learned anonymization f_A^* is not accessible to a intruder, hence one can try to extract privacy information using a pretrained privacy classifier of raw data. In this protocol, instead of learning a target privacy model f'_B from the anonymized version of the training data, we directly evaluate f_A^* using a privacy target model which is pretrained on raw data. Results are shown in Table 2. We use ResNet-50 model as privacy target model, which is pretrained on raw training data of the the respective evaluation set. There are two main observations in in this protocol: (1) Compared to other methods, supervised [17] and our self-supervised method gets a remarkable amount of privacy classification drop, which is desired to prevent privacy leakage. (2) Our method gets a competitive cMAP performance to [17], and greatly outperforms it in terms of F1 score.

D.2. Evaluating learned f_A^* on different utility target model f'_T

A learned anonymization function, f_A^*, should allow learning any action recognition target model, f'_T, over the anonymized version of training data without significant drop in the performance. Using the R3D-18 as a auxiliary action recognition model, f_T, in the training of anonymization function, we evaluate the learned f_A^* to train different action recognition (utility) target models like R3D-18, C3D [15], and R2plus1D-18 from scratch and Kinetics-400 [1] pretraining. Results are shown in Table 3. We can observe that our method maintains the action recognition performance on any utility action recognition model. Also, it is interesting to notice that the learned anonymization by our method and method in [17] get benefit from a large-scale raw data pretraining of Kinetics-400.

[3] https://github.com/VITA-Group/Privacy-AdversarialLearning

Method	R3D-18	R2Plus1D	R2Plus1D K400 pretraining	C3D
Raw data	62.3	64.33	88.76	58.51
Supervised [17]	62.1	62.58	85.33	56.30
Ours	62.03	62.71	85.14	56.10

Table 3. Evaluation with different architectures of **action recognition utility target model** f'_T. Results shows Top-1 Accuracy (%) on UCF101. Goal of this evaluation is to maintain the action recognition performance close to the raw data baseline regardless of choice of model f'_T. Our self-supervised method achieves **model-agnostic action recognition performance** which is also comparable to the supervised method [17].

D.3. Evaluating on different privacy target model f'_B

A learned anonymization function f_A^* is expected to provide protection against privacy leakage from any privacy target model f'_B. In training of anonymizatoin function , we use ResNet50 as the auxiliary privacy model f_B and evaluate the learned anonymization f_A^* on target privacy classifiers f'_B like ResNet18/50/34/101/152 and MobileNet-V1 with and without ImageNet [3] pretraining. From Table 4, we can observe that our method protects privacy leakage regardless of choice of target privacy model. Using ImageNet pretraining as shown in Table 5, privacy leakage increases in all methods, however, the relative drop to the raw data baseline is improved.

D.4. Evaluation protocol: Pretrained Action classifier and fixed privacy classifier

In a practical scenario, we can initialize an action recognition target model f'_T from the Kinetics400 raw data pretrained checkpoint. Also, an intruder has no direct access to the learned anonymization function in a practical setting, hence we can consider the raw-data pretrained privacy clas-

Method	ResNet18		ResNet34		ResNet50		ResNet101		ResNet152		MobileNet-V1	
	cMAP (%)(↓)	F1(↓)	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1
Raw data	64.38	0.5385	65.30	0.5554	64.40	0.5553	60.70	0.5269	58.83	0.4852	61.21	0.5056
Supervised	53.84	0.4402	53.22	0.4283	53.97	0.4459	53.55	0.4257	51.05	0.4030	52.48	0.4013
Ours	54.83	0.4574	54.09	0.4226	57.43	0.4732	52.94	0.4096	53.27	0.4322	53.41	0.3974

Table 4. Evaluating f_A^* for privacy leakage against different architectures of **privacy target model** f'_B. Results shown on VISPR-1 dataset. Lower privacy classification score is better. Our self-supervised method gets a **model-agnostic privacy anonymization performance** which is also comparable to the supervised method [17].

Method	ResNet18		ResNet34		ResNet50		ResNet101		ResNet152	
	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1	cMAP (%)	F1
Raw data	69.82	0.6041	69.55	0.6447	70.66	0.6591	71.09	0.6330	69.50	0.6130
Supervised	58.05	0.5367	58.02	0.5463	62.01	0.5281	61.44	0.5553	61.88	0.5711
Ours	59.10	0.5302	59.71	0.5227	60.73	0.5689	59.24	0.5601	60.51	0.5352

Table 5. Similar setting as Table 4, but f'_B is initialized with **ImageNet Pretraining**.

Method	Top-1 Acc (%) (↑)	cMAP (%) (↓)		
		ResNet18	**ResNet50**	**ResNet101**
Raw data	88.76	64.38	64.40	60.70
Downsample-2x	77.45	49.37	51.23	50.72
Downsample-4x	63.53	36.22	38.82	40.68
Obf-Blackening	72.11	46.48	48.38	47.92
Obf-StrongBlur	74.10	53.30	54.44	52.39
Supervised	85.33	19.23↓70%	22.81↓64%	22.01↓64%
Ours	85.01	22.16↓66%	23.44↓64%	22.64↓63%

Table 6. Trade-off between action classification and privacy classifier in a **practical scenario** where target utility model is taken from Kinetics400 checkpoint and target privacy model is raw-data pretrained. UCF101 is used as action classification dataset and VISPR is used as privacy dataset. ↓% denotes relative drop from raw data. With a small drop in action recognition performance our method greatly reduce privacy leakage.

sifier as a target privacy model f'_B. Results are shown in Table 6. We use Kinetics400 pretrained R2Plus1D-18 model as the action recognition target model f'_T, and ResNet models with varying capacity as the target privacy model f'_B. Plotting the trade-off of Table 6 in Fig. 1, we can observe that at the cost of a small drop in action recognition performance our method obtains about **66% reduction in privacy leakage** from the raw data baseline. This highlights the potential of our self-supervised privacy preserving framework in a practical scenario without adding cost of privacy annotation in training.

f_T architecture	UCF101	VISPR1	
	Top-1(%) (↑)	cMAP(%) (↓)	F1 (↓)
R3D-18	62.03	57.43	0.4732
R2+1D-18	62.37	57.37	0.4695
R3D-50	62.58	57.51	0.4707

Table 7. **Auxiliary utility model** f_T architecture has no significant effect on final action-privacy measures. Auxiliary models are just used to train the anonymization function and discarded after that. All results are reported on ResNet50 privacy target model f'_B and R3D-18 action recognition target model f'_T.

D.5. Plots for known and novel action and privacy attributes protocol

A trade-off plot for evaluating learned f_A^* for novel action-privacy attributes is shown in Fig. 2 and known action-privacy attributes is shown in Fig 3, for more details see Sec. 5 of main paper.

E. Additional ablations

E.1. Effect of different f_T architectures

To understand the effect of auxiliary model f_T in the training process of f_A, we experiment with different utility auxiliary model f_T, and report the performance of their learned f_A^* in the same evaluation setting as shown in Table 7. We can observe that there is no significant effect of f_T in learning the f_A.

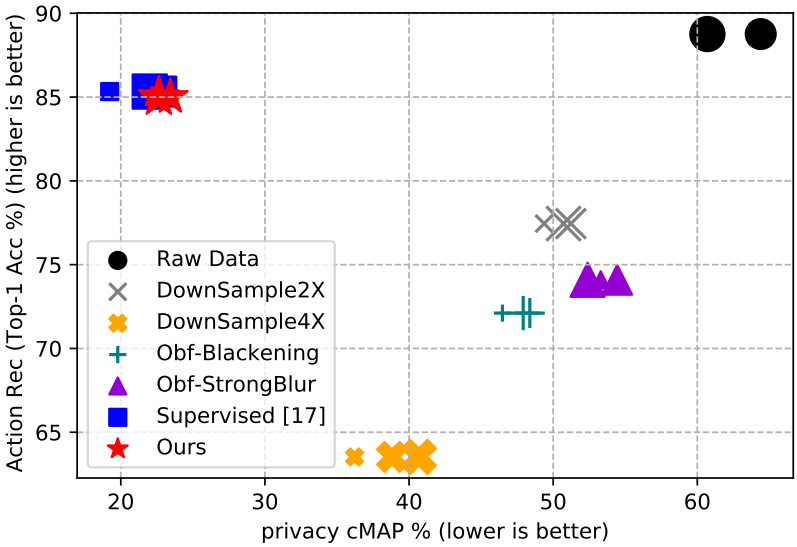
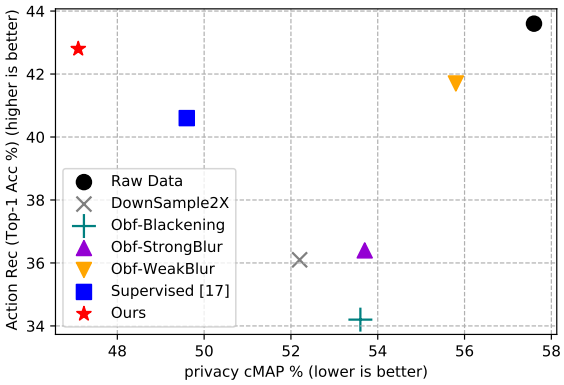
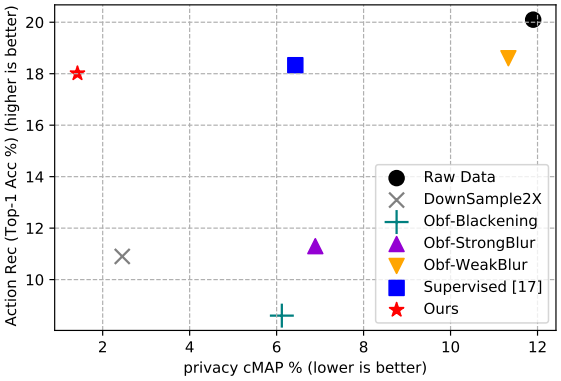


Figure 1. Trade-off between action classification using **pretrained** action classifier and **raw-data frozen** privacy classifier. UCF101 is used as action classification dataset and VISPR is used as privacy dataset. Increasing size of the marker shows increasing size of privacy classifiers: ResNet18, ResNet50, ResNet101.



(a) Trade-off between action classification and privacy removal while generalizing from **UCF101→PA-HMDB** for action and **VISPR1→VISPR2** for privacy attributes.



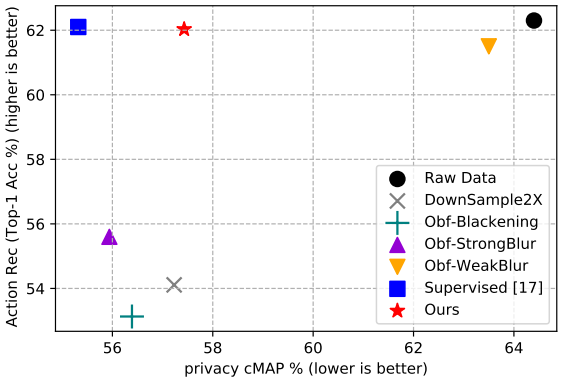
(b) Trade-off between action classification and privacy removal while generalizing from **Scenes→Objects** for privacy attributes on **P-HVU** dataset.

Figure 2. Evaluating learned anonymization for **novel action-privacy attributes**. Our framework outperforms the supervised method [17] and achieves **robust generalization** across novel action-privacy attributes. For more details refer Sec. 5.4 of main paper.

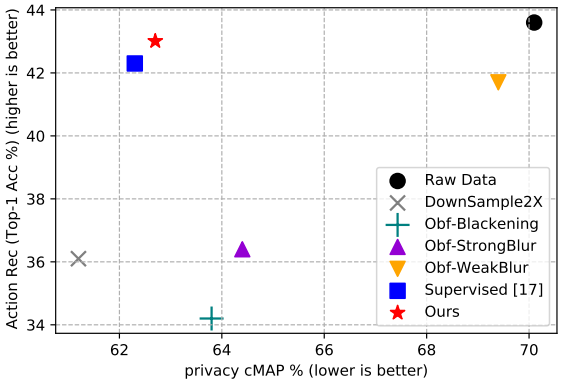
F. Qualitative Results

F.1. Visualization of learned anonymization f_A^* at different stages of training

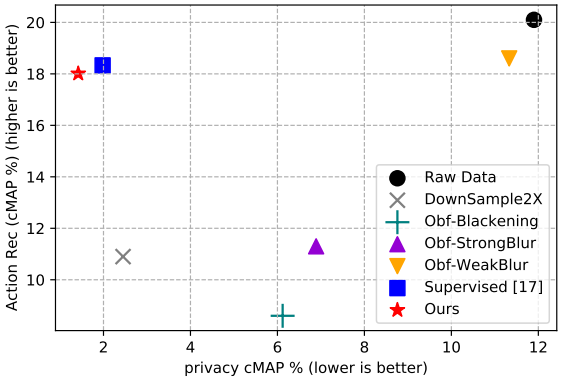
In order to visualize the transformation due to learned anonymization function f_A^*, we experiment with various test set videos of UCF101. The sigmoid function after the f_A^* ensure (0,1) range of the output image. We visualize output at different stages of anonymization training as shown in Fig. 4, 5, 6. We can see our self-supervised framework is successfully able to achieve anonymization as the training progresses.



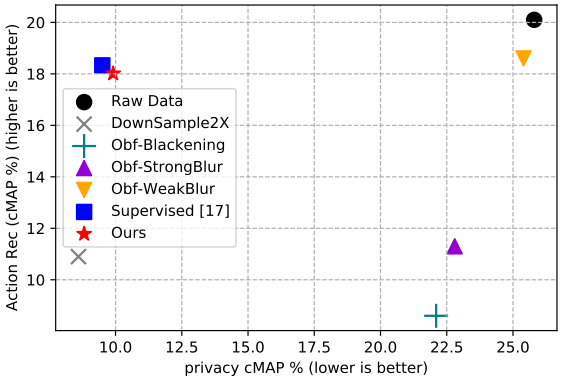
(a) Trade-off between action classification on **UCF101** vs privacy classification on **VISPR-1**.



(b) Trade-off between action classification vs privacy classification on **PA-HMDB**.



(c) Trade-off between action classification vs **privacy-object** classification on **P-HVU**.



(d) Trade-off between action classification vs **privacy-scene** classification on **P-HVU**.

Figure 3. Evaluating learned anonymization for **known action-privacy attributes**. Our framework achieves comparable performance to the supervised method [17]. For more details refer Sec. 5.3 of main paper.

F.2. Visualization of learned anonymization f_A^* for different methods

Apart from Fig. 4, 5, 6 visualization of our method, we show visualization for all methods, attached in the form of videos in the supplementary zip file.

F.3. Attention map for supervised vs self-supervised privacy removal branch

A self-supervised model focuses on **holistic spatial semantics**, whereas a supervised privacy classifier focuses on specific semantics of the privacy attributes. To bolster this observation, we visualize the attention map of ResNet50 model which is trained in (1) Supervised manner using binary cross entropy loss using VISPR-1. (2) Self-supervised manner using NT-Xent loss. We use the method of Zagoruyko and Komodakis [18] to generate model attention from the third convolutional block of the ResNet model. As can be observed from the attention map visualization of Fig. 7 that a self-supervised model focuses on semantics related to human and its surrounding **scene**, whereas, the supervised privacy classifier mainly focuses on the human semantics. In Fig. 8, we can see that the self-supervised model attends to the semantics of **object** along with human, and supervised privacy classifier mainly learns semantics of human only.

G. Visual Aid for training and evaluation protocols

In order to better understand protocols of Sec. 4 of main paper, we provide here some visual aids in Fig 9, 10, and 11.

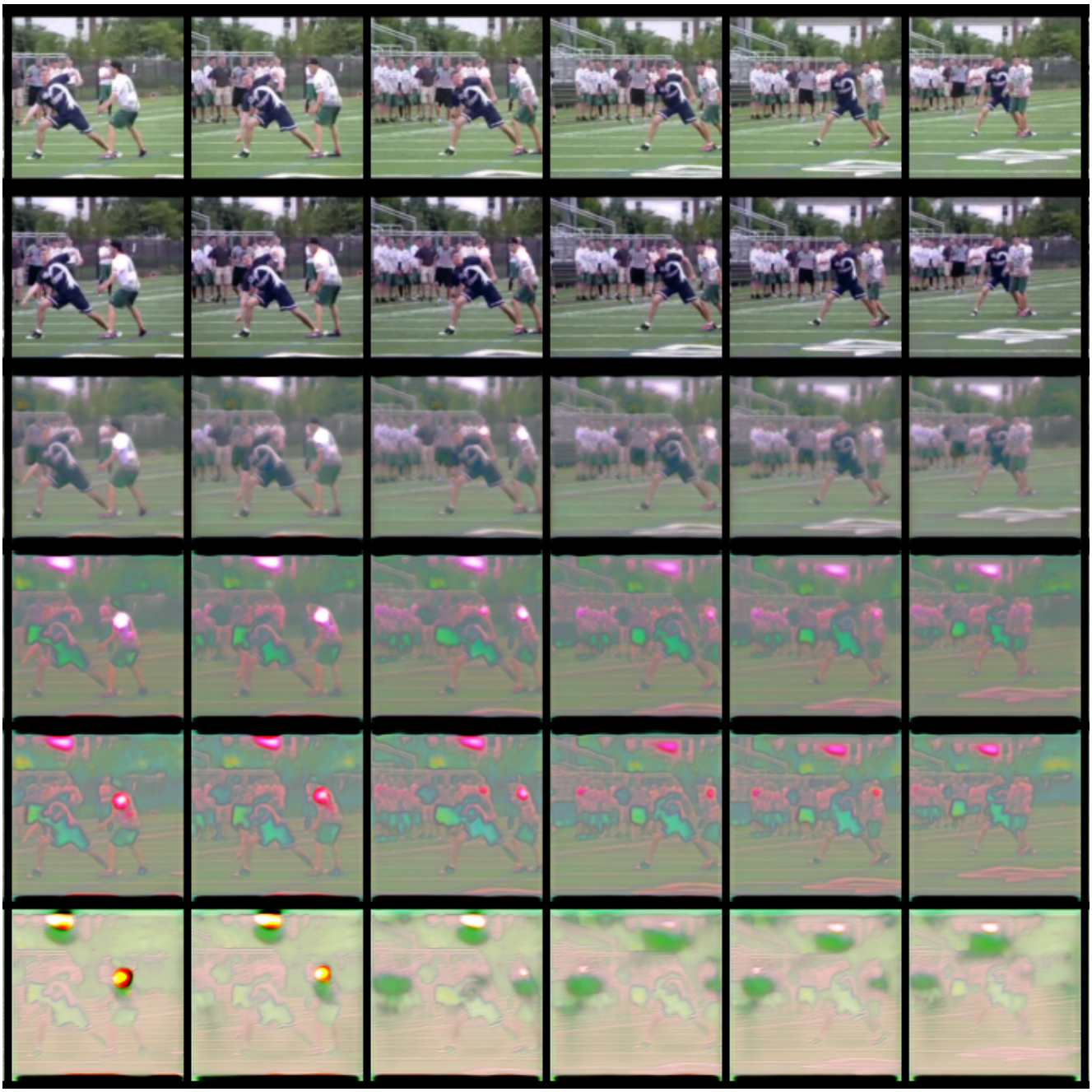


Figure 4. Learned anonymization using our self-supervised privacy preservation framework on test set of UCF101. Groundtruth action label: `FrisbeeCatch`. First row: original video, from second to last row: anonymized version of video at epoch 1, 3, 6, 9, 30.

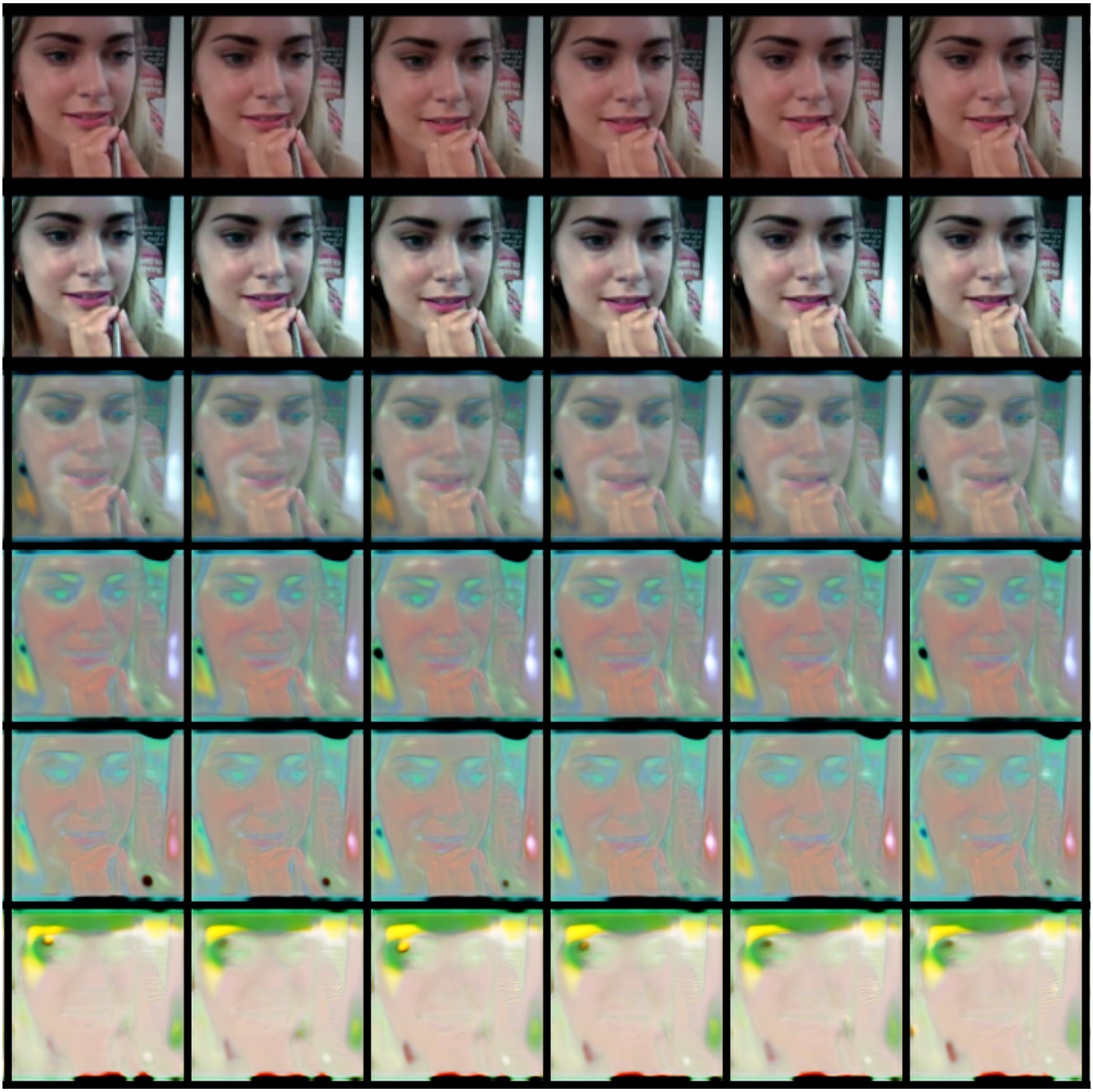


Figure 5. Learned anonymization using our self-supervised privacy preservation framework on test set of UCF101. Groundtruth action label: `ApplyLipstick`. First row: original video, from second to last row: anonymized version of video at epoch 1, 3, 6, 9, 30.

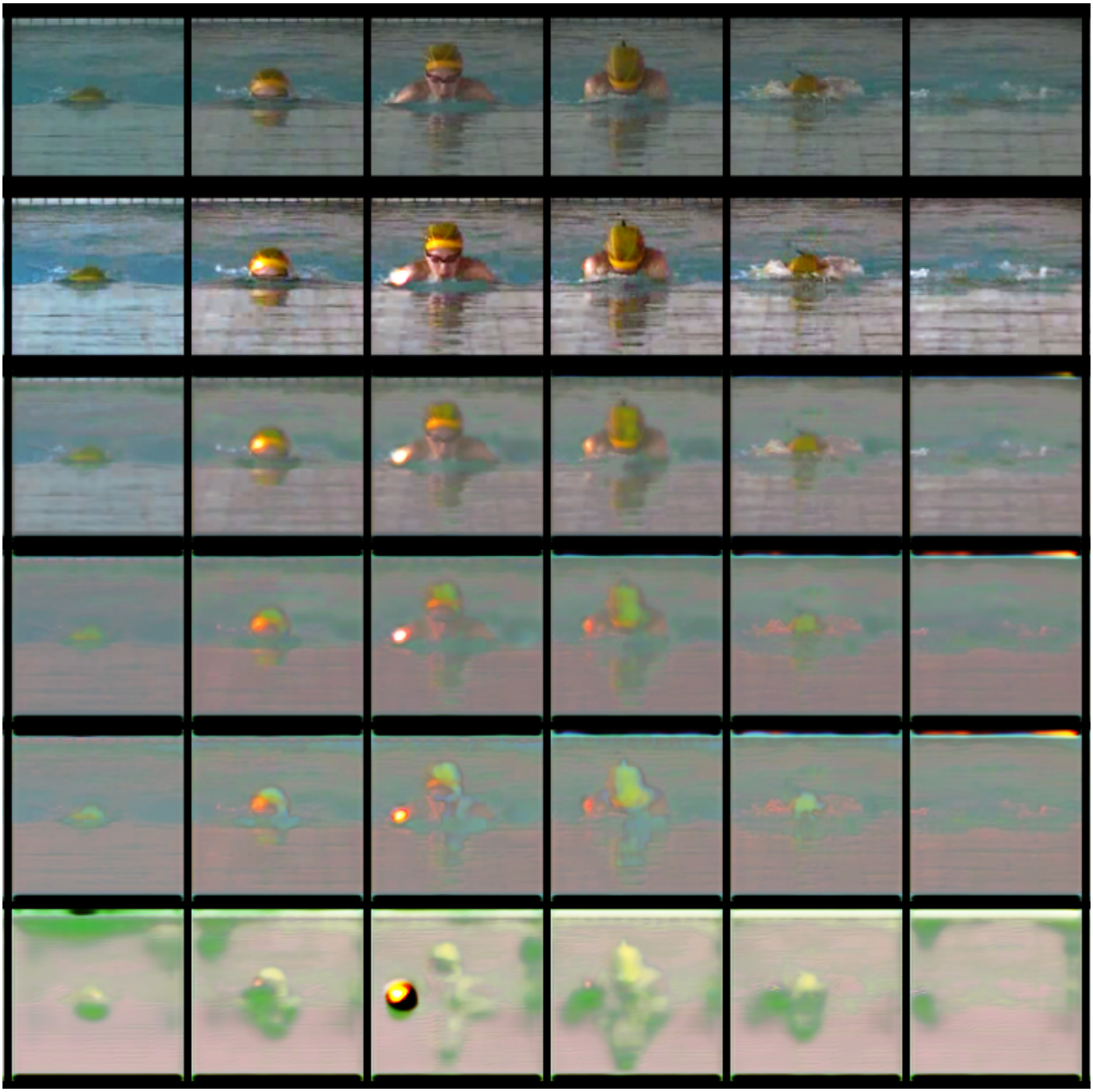
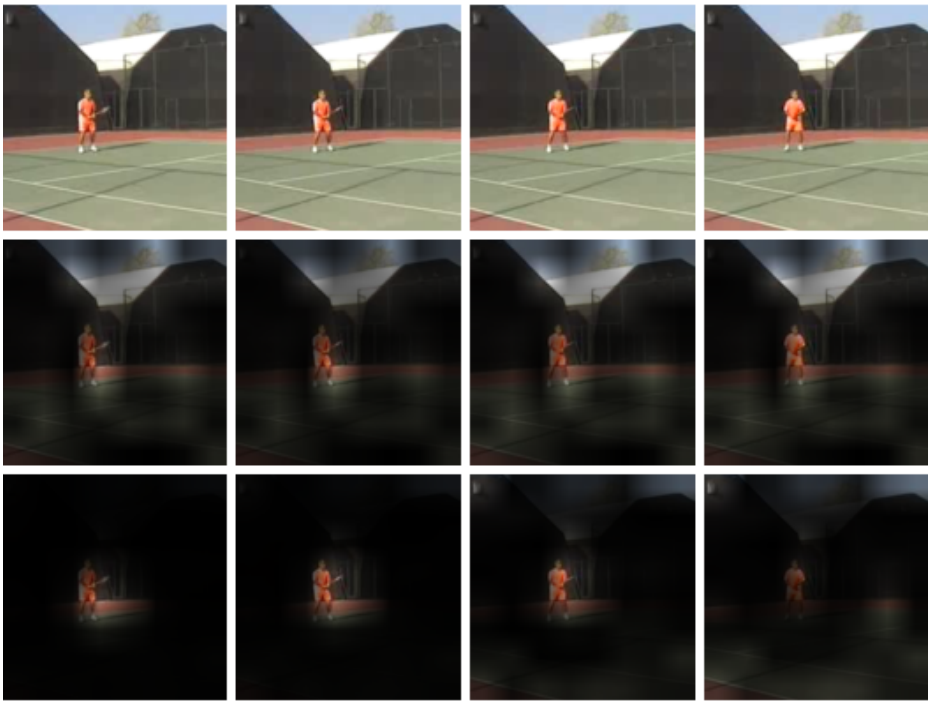


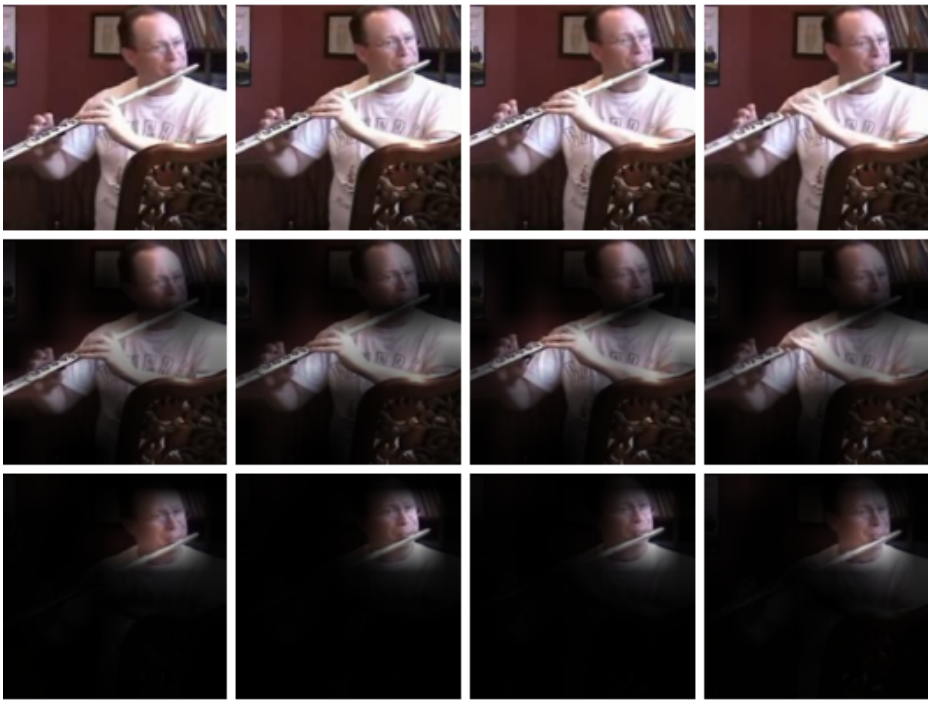
Figure 6. Learned anonymization using our self-supervised privacy preservation framework on test set of UCF101. Groundtruth action label: `BreastStroke`. First row: original video, from second to last row: anonymized version of video at epoch 1, 3, 6, 9, 30.

(a) `Lunges`



(b) `TennisSwing`

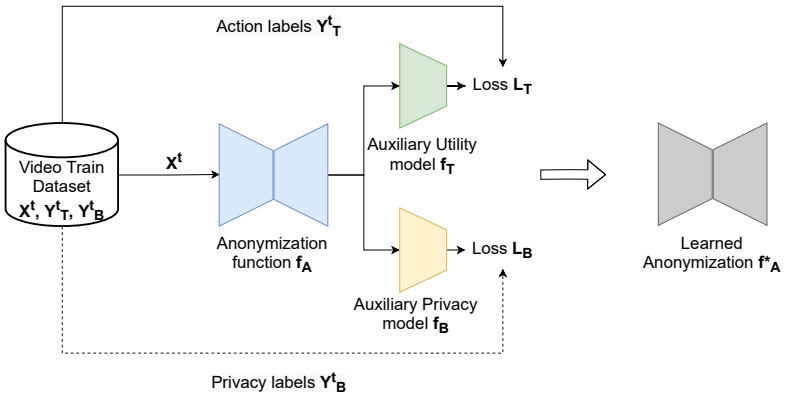
Figure 7. **Attention map visualization**: Top row: original video, middle-row: attention of a self-supervised model, bottom-row: attention of supervised privacy classifier. It can be observed that supervised privacy classifier mainly focuses on the semantics of human, whereas self-supervised model learns holistic spatial semantic features related to the **scene** (eg. **track-field** in (a) and **tennis court** in (b)) as well.



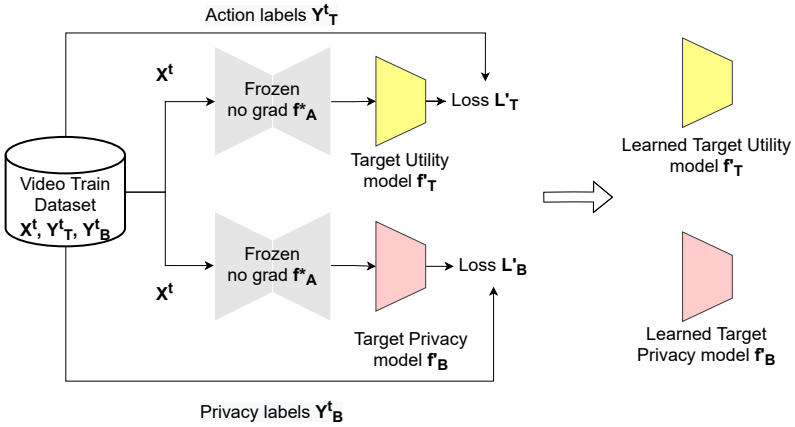
(a) `PlayingFlute`

(b) `Skijet`

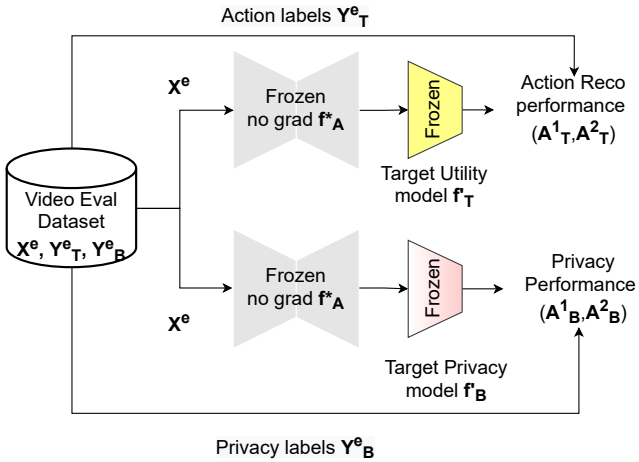
Figure 8. **Attention map visualization**: Top row: original video, middle-row: attention of a self-supervised model, bottom-row: attention of supervised privacy classifier. It can be observed that supervised privacy classifier mainly learns semantics of human, whereas self-supervised model learns holistic semantic spatial features related to the **objects** (eg. **Flute** in (a) and **SkiJet** in (b)) as well.



(a) **First phase: Training of anonymization function** f_A. For our self-supervised method we do not require privacy labels Y^t_B. At the end of training, f_A is frozen call it f^*_A, and auxiliary models f_B and f_T are discarded.

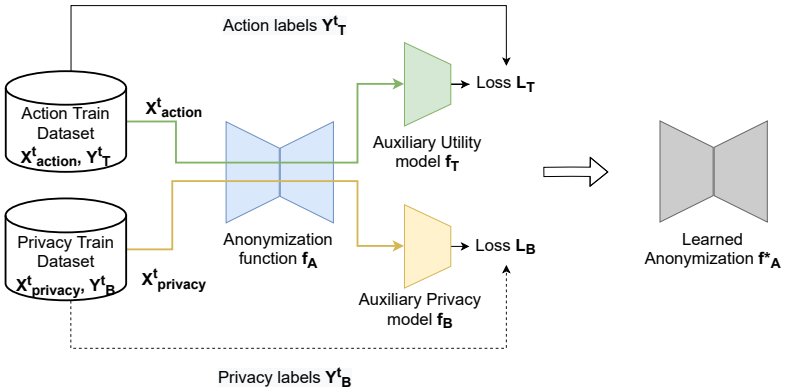


(b) **Second phase :Target models training** Target models are used to evaluate the performance of learned anonymization function f^*_A and are different from auxiliary models. Target utility model i.e. action classifier f'_T and Target privacy classifier f'_B are learned in supervised manner on the anonymized version of training data X^t.

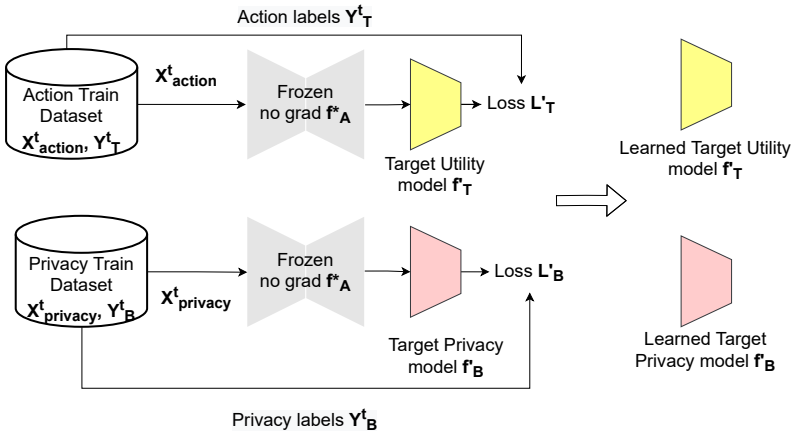


(c) **Third phase: Target models testing**: Once target models are trained from anonymized version of X^t, they are are frozen and evaluated on anonymized version of test/evaluation set X^e.

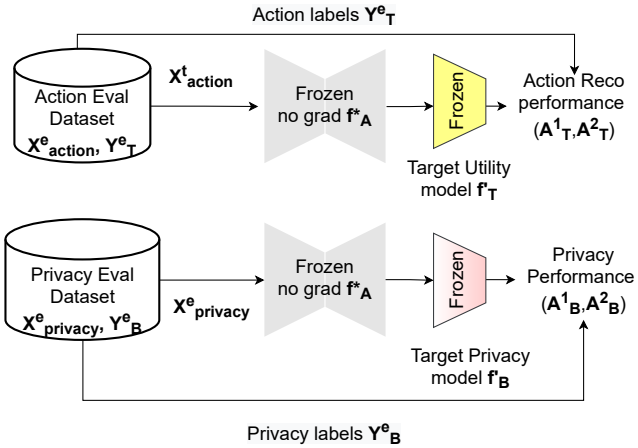
Figure 9. Visual Aid for **Same-dataset** training and evaluation protocol Sec. 4.1 of main paper.



(a) **First phase: Training of anonymization function** f_A. For our self-supervised method we do not require privacy labels Y^t_B. At the end of training, f_A is frozen call it f^*_A, and auxiliary models f_B and f_T are discarded.

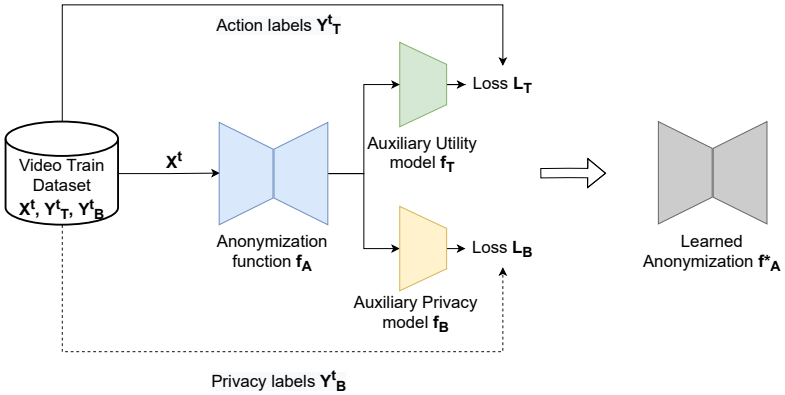


(b) **Second phase :Target models training** Target models are used to evaluate the performance of learned anonymization function f^*_A and are different from auxiliary models. Target utility model (action classifier) f'_T and Target privacy classifier f'_B are learned in supervised manner on the anonymized version of training data X^t_{action} and $X^t_{privacy}$.

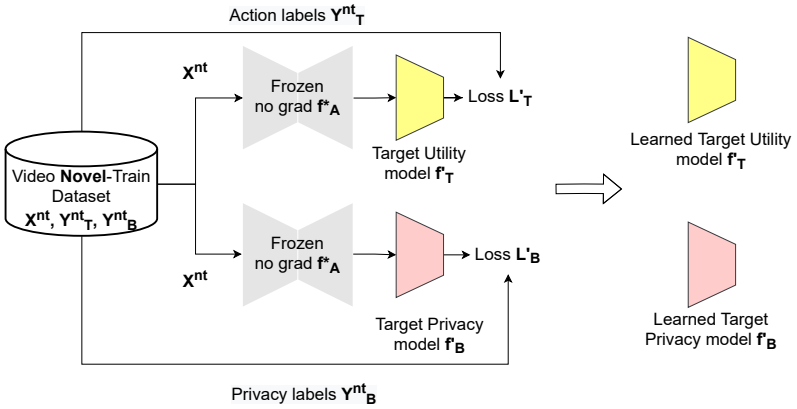


(c) **Third phase: Target models testing**: Once target models are trained from anonymized version of X^t_{action}, $X^t_{privacy}$, they are are frozen and evaluated on anonymized version of test/eval set X^e_{action} , $X^e_{privacy}$.

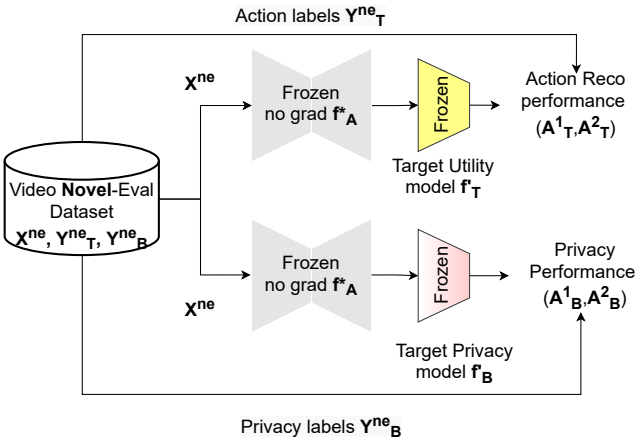
Figure 10. Visual Aid for **Cross-dataset** training and evaluation protocol Sec. 4.2 of main paper.



(a) **First phase: Training of anonymization function** f_A. For our self-supervised method we do not require privacy labels Y^t_B. At the end of training, f_A is frozen call it f^*_A, and auxiliary models f_B and f_T are discarded.



(b) **Second phase :Target models training** Target models are used to evaluate the performance of learned anonymization function f^*_A and are different from auxiliary models. Target utility model (action classifier) f'_T and Target privacy classifier f'_B are learned in supervised manner on the anonymized version of **novel training data** X^{nt} which has action and privacy labels such that $Y^{nt}_T \cap Y^t_T = \phi$ and $Y^{nt}_B \cap Y^t_B = \phi$



(c) **Third phase: Target models testing**: Once target models are trained from anonymized version of **novel training data** X^{nt}, they are are frozen and evaluated on anonymized version of **novel test/eval set** X^{ne}.

Figure 11. Visual Aid for **Novel Action and privacy attribution** protocol Sec. 4.3 of main paper.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[4] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jürgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large scale holistic video understanding. In *European Conference on Computer Vision*, pages 593–610. Springer, 2020. 1

[5] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[6] K. Hara, H. Kataoka, and Y. Satoh. Towards good practice for action recognition with spatiotemporal 3d convolutions. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2516–2521, 2018. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[8] Glenn Jocher. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. https://github.com/ultralytics/yolov5, Oct. 2020. 2

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 1

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[12] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[14] 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. 1

[15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 3

[16] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[17] Zhenyu Wu, Haotao Wang, Zhaowen Wang, Hailin Jin, and Zhangyang Wang. Privacy-preserving deep action recognition: An adversarial learning framework and a new dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2, 3, 4, 5, 6

[18] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 6

[19] Dalin Zhang, Lina Yao, Kaixuan Chen, Guodong Long, and Sen Wang. Collective protection: Preventing sensitive inferences via integrative transformation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1498–1503. IEEE, 2019. 2