Supplementary Material of Motion-Excited Sampler: Video Adversarial Attack with Sparked Prior

Hu Zhang[1], Linchao Zhu[1], Yi Zhu[2], and Yi Yang[1]

[1] ReLER, University of Technology Sydney, NSW
[2] Amazon Web Services
Hu.Zhang-1@student.uts.edu.au; zhulinchao7@gmail.com;
yzaws@amazon.com; Yi.Yang@uts.edu.au

A Implementation details

In terms of the accumulated motion vector, we follow the setting in [5] and set each interval T within a video to 12 frames. There is no overlap between two adjacent intervals. For each interval, we generate one accumulated motion map. In adversarial attacking, we impose noises on original video frames which are normalized to [0,1]. The modified videos are processed via standard 'mean' and 'std' normalization before inputting to the black-box model for gradient estimation. Each iteration consumes three queries that two queries for Δ in Algorithm 1 and one to determine whether the updated video $\boldsymbol{x}_t$ is successful in attacking. In the untargeted attack setting, we set the query limit to 60,000 and the maximal iteration for updating adversarial video is 20,000. In the targeted attack setting, we set the query limit to 200,000 and the maximal iteration is 66,667 (200,000/3).

In Algorithm 1, the interval t for sampling new motion maps is 10, δ for adjusting the magnitude of loss variation is 0.1, ϵ for approximation is 0.1. In Algorithm 2, learning rate η for updating estimated gradient $\boldsymbol{g}_t$ is 0.1 and learning rate h for updating adversarial video $\boldsymbol{x}_t$ is 0.025.

B Attack transferability on flow stream

In this section, we investigate the attack transferability of generated adversarial video on motion stream. We first train an optical flow model on original videos. We then extract new optical flows from adversarial videos generated in our attack and evaluate the performance of obtained flow model on new flows. We measure the attack transferability of our adversarial video in terms of recognition accuracy. In the experiment, we train the optical flow model on Something-Something V2, where motion information is especially important for accurate recognition.

The above table shows that using flow extracted from original videos achieves 40.59% recognition accuracy. However, for the new flow extracted from our generated adversarial video, its recognition accuracy goes down to 12.35%. This result clearly demonstrates that our adversarial perturbation designed for RGB stream can be transferred to attack the flow model.

Table 1. Transferability of adversarial video on motion stream

Type of flow	Recognition accuracy (%)
Flow extracted from original video	40.59
Flow extracted from our adversarial video	12.35

C More visualization

Demo video We first put together a video to showcase our generated adversarial samples. The video can be found in *MESampler-demo-video.mp4*.

Video and motion map visualization We also show more results of adversarial video frames and the adopted motion vector on four datasets. Visualizations on SthSth-V2 [1] and HMDB-51 [3] are shown in Fig. 1 and the results of Kinetics-400 [2] and UCF-101 [4] are in Fig. 2.

From the demo video and the frame visualizations on four datasets, we can see that our generated adversarial samples can successfully fail the video classification model. Even though the generated video samples look the same as the original videos and the true labels can still be recognized by human without any difficulties.

Failure cases To have a better understanding of our model, we show several failure cases of our method. Examples on Kinetics-400 and UCF-101 are shown in Fig. 3 and Fig. 4, respectively.

There are two potential reasons behind failures in adversarial videos. The first one is about the confidence of the attacked video model. On certain videos, the model is confident about its prediction. Take the first video in Fig. 3 as an example, the black-box model outputs its true label 'golf driving' with confidence 0.9999. This confidence score is so high that the perturbation posed on the video is likely to have little consequences on the final results. Secondly, it is about motion quality in videos. We notice that for videos that we fail to attack, their motion map is rather obscure and unrecognizable. Under such circumstances, the advantage of motion information can not be fully utilized.

D Better motion information

Here, we justify another assumption that clearer and more completed motion maps will lead to better attack performance. Rather than fixing the starting point and length of each interval for generating motion map, the starting point and the length of interval for generating motion maps is modified according to the trajectory of given video to get a clearer and more complete description of movement. Such new map is termed as 'improved motion map'. We show two

samples on Kinetics-400 in Fig. 5, that are from class 'zumba' and 'vault'. The left column are the video frames, 'improved motion map' and original motion map are in the middle and right respectively. Clearly, improved motion map in the middle is more consistent and clearer. For sample (a), it saves more than 10,000 queries by applying 'improved motion map' instead of original one. For sample (b), 30,000 queries are also saved by using the 'improved motion map'.

However, it is still difficult to automatically determine the starting point and the interval length to generate much clearer maps. We will leave the study as the future work.

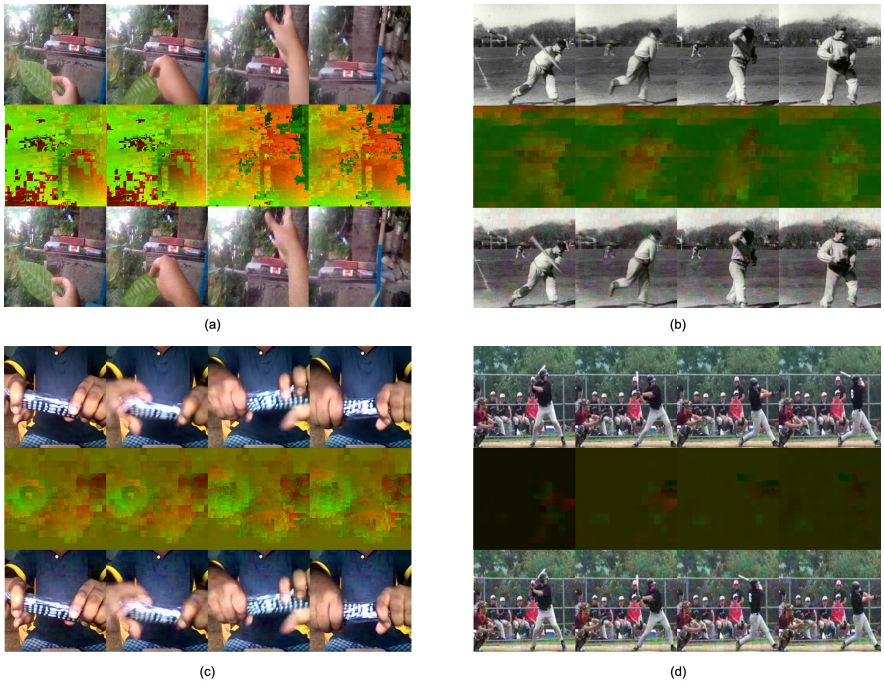

Fig. 1. Examples of motion vectors used in attacking and the generated adversarial samples. In (a)-(d), the first row is the original video frame, the second row is the motion vector and the third row is generated adversarial video frame. a) SthSth-V2 on I3D: throwing a leaf in the air and letting it fall → throwing tooth paste; b) HMDB-51 on I3D: throw → fencing; c) SthSth-V2 on TSN2D: pretending or trying and failing to twist remote-control → pretending to open something without actually opening it; d) HMDB-51 on TSN2D: swing-baseball → throw.

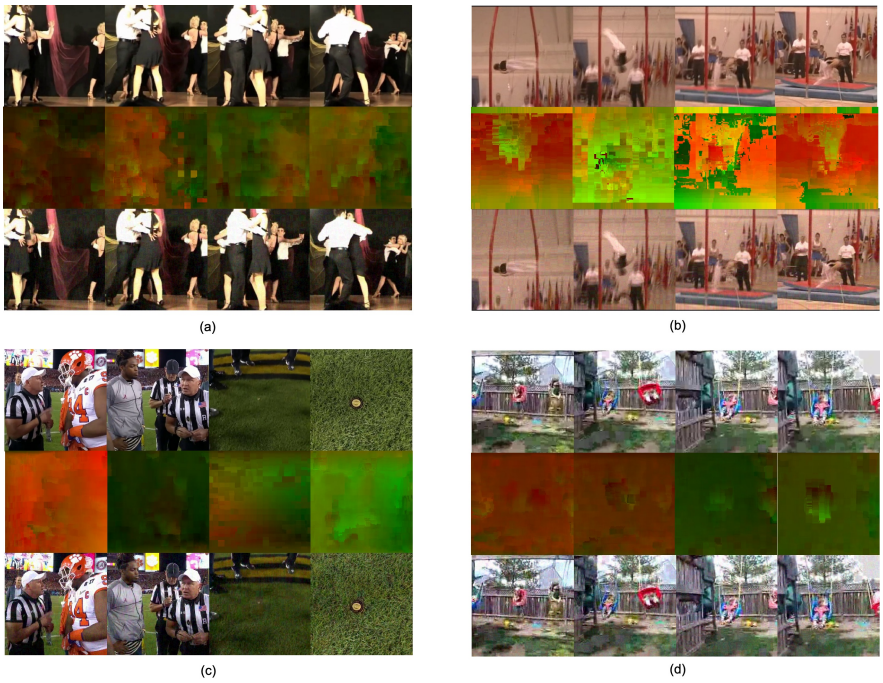

Fig. 2. Examples of motion vectors used in attacking and generated adversarial samples. In (a)-(d), the first row is the original video frame, the second row is the motion vector and the third row is generated adversarial video frame. a) Kinetics-400 on I3D: tango dancing → salsa dancing; b) UCF-101 on I3D: StillRings → PoleVault; c) Kinetics-400 on TSN2D: tossing coin → scissors paper; d) UCF-101 on TSN2D: Swing → TrampolineJumping.

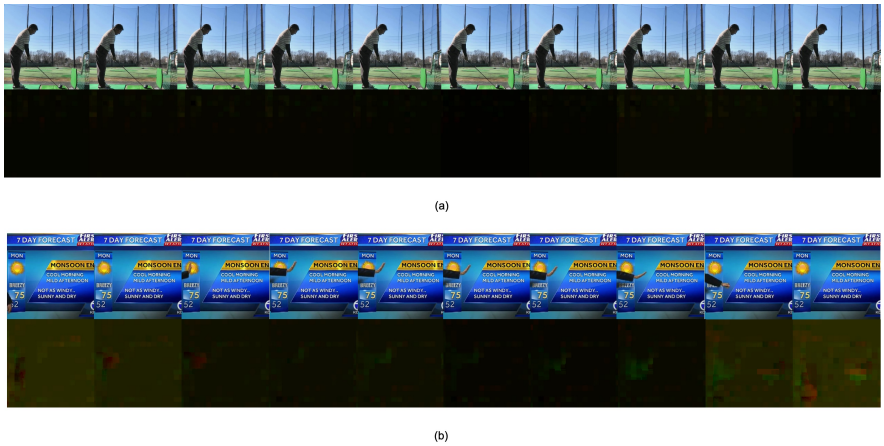

Fig. 3. Failed samples from Kinetics-400 against I3D and TSN2D. a) Sample from class 'golf driving' against I3D; b) Sample from class 'presenting weather forecast' against TSN2D. The first row are the frames of original video and the second row are the motion vectors generated between frames. The movements between video frames seem to change little and the generated motion vectors are very obscure.

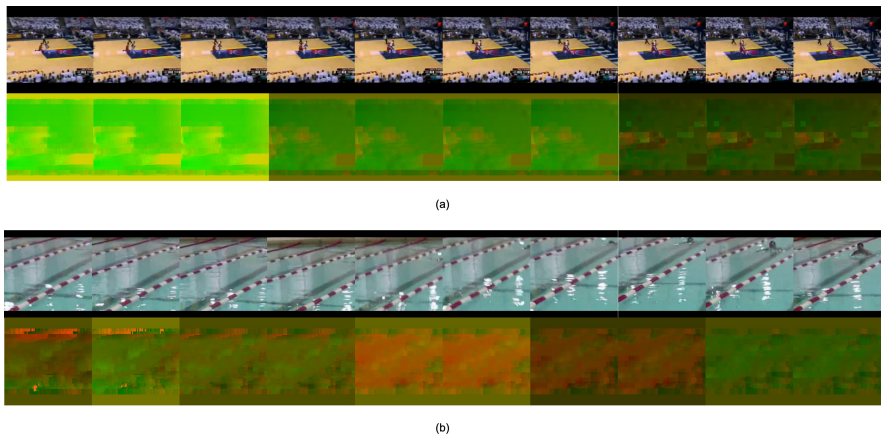

Fig. 4. Failed samples from UCF-101 against I3D and TSN2D. a) Sample from class 'BasketballDunk' against I3D; b) Sample from class 'BreastStroke' against TSN2D. The first row are the frames of original video and the second row are the motion vectors generated between frames. The movement between video frames changes little and the target object in the video is very small.

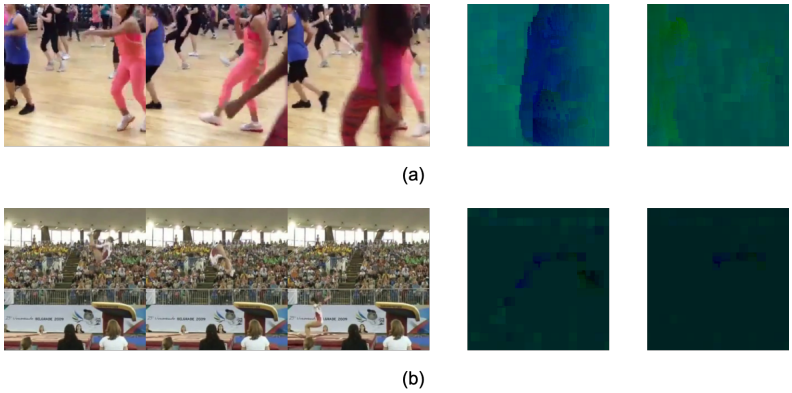

Fig. 5. Rather than fixing the starting point and length of each interval for generating motion map, the start point and the length of interval for generating motion maps is modified according to the trajectory of the given video to get clearer and more complete description of movement. Two samples from Kinetics-400: **a)** Sample from class 'zumba'; **b)** Sample from class 'vault'. **Left**: the frames of original video; **Middle**: 'improved motion map'; **Right**: original motion map in attacking. Clearly, 'improved motion map' is more complete and clearer than original motion map in the right. The attacking results are also better by using the 'improved motion map'.

References

1. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV. vol. 1, p. 3 (2017)
2. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
3. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision. pp. 2556–2563. IEEE (2011)
4. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
5. Wu, C.Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A.J., Krähenbühl, P.: Compressed video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6026–6035 (2018)