Breaking Temporal Consistency: Generating Video Universal Adversarial Perturbations Using Image Models

Hee-Seon Kim, Minji Son, Minbeom Kim, Myung-Joon Kwon, Changick Kim
Korea Advanced Institute of Science and Technology (KAIST)
{hskim98, ming0103, alsqja1754, kwon19, changick}@kaist.ac.kr

Abstract

As video analysis using deep learning models becomes more widespread, the vulnerability of such models to adversarial attacks is becoming a pressing concern. In particular, Universal Adversarial Perturbation (UAP) poses a significant threat, as a single perturbation can mislead deep learning models on entire datasets. We propose a novel video UAP using image data and image model. This enables us to take advantage of the rich image data and image model-based studies available for video applications. However, there is a challenge that image models are limited in their ability to analyze the temporal aspects of videos, which is crucial for a successful video attack. To address this challenge, we introduce the Breaking Temporal Consistancy (BTC) method, which is the first attempt to incorporate temporal information into video attacks using image models. We aim to generate adversarial videos that have opposite patterns to the original. Specifically, BTC-UAP minimizes the feature similarity between neighboring frames in videos. Our approach is simple but effective at attacking unseen video models. Additionally, it is applicable to videos of varying lengths and invariant to temporal shifts. Our approach surpasses existing methods in terms of effectiveness on various datasets, including ImageNet, UCF-101, and Kinetics-400.

1. Introduction

Deep learning models have achieved remarkable performance in various computer vision tasks [5, 2, 33, 13, 1], including image and video recognition. However, there is growing concern about the robustness and reliability of these models, as they have been shown to be vulnerable to adversarial attacks [34, 6, 42]. Adversarial attacks use imperceptible perturbations to manipulate the inputs to produce inaccurate predictions. These attacks can have serious consequences in various applications of deep neural networks, such as autonomous vehicles and surveillance cameras [35] where false activity detection [19] can cause serious consequences. Despite these concerns, the problem of adversarial attacks on video models remains largely unsolved.

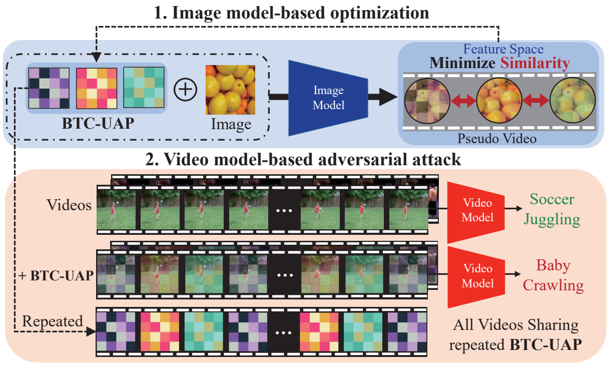

Figure 1: **Overall illustration of Breaking Temporal Consistency Method.** We propose a novel approach to minimize the similarity between features of consecutive frames in video adversarial attacks. Please note that the illustrated BTC-UAP is not a real representation, but rather serves as a visual aid. The different colors represent the low similarity between features.

Adversarial attacks can be broadly categorized into white-box [37, 14, 29] and black-box [39, 40] attacks. White-box attacks exploit model information to generate adversarial examples, while black-box attacks are more challenging due to the lack of model access. In real-world scenarios, accessing the target model is often difficult or impossible, so black-box attacks are more practical. One way to launch black-box attacks is by leveraging the transferability of adversarial examples [39, 40, 42, 7, 24], applying adversarial examples crafted using accessible source models to the target models. Transfer-based attacks can also be cross-modal [40], which enables attackers to transfer adversarial examples between different modalities, such as image to video. For most cases, crafting adversarial examples still requires optimization for each individual adversarial example. On the other hand, Universal Adversarial Perturbations (UAPs) [26, 37, 14, 43, 23] poses a powerful threat as a sin-

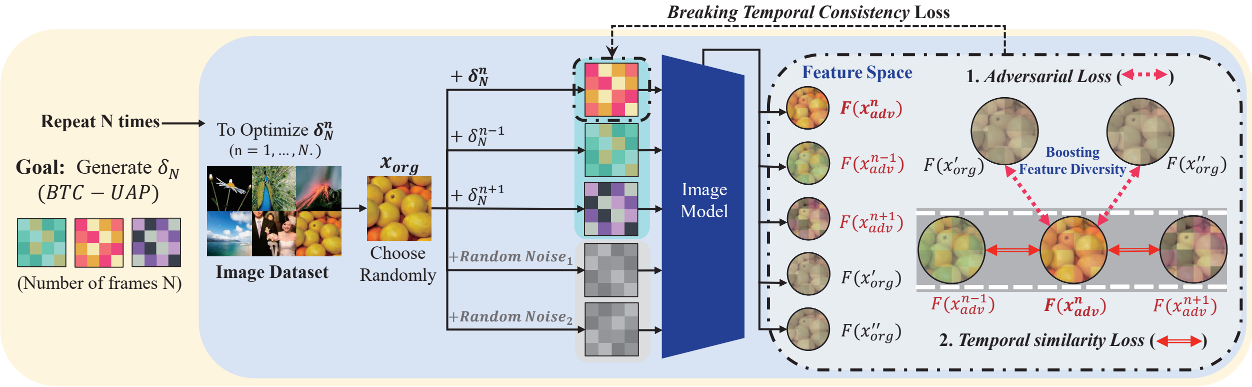

Figure 2: **Details of Breaking Temporal Consistancy Method.** Our goal is to create BTC-UAP for video attacks composed of N frames. We treat each frame of the UAP as an individual image, and add it to the original image to generate corresponding adversarial images. To ensure that these images are adversarial, we use an Adversarial Loss and prevent overfitting with the Feature Diversity method. Additionally, while treating the adversarial images as a pseudo video, we apply the Temporal Similarity Loss to the video frames and make each frame distinct from one another.

gle perturbation can mislead deep learning models on entire datasets. This is considered a highly practical attack method in scenarios where it may be difficult or impossible to optimize adversarial perturbations for each individual dataset every time, such as real-time systems.

Our study aims to extend the applicability of UAPs generated using image data and models, to the domain of video data and models. The overall scheme is illustrated in Fig. 1. This extension allows significant benefits as it allows us to leverage the wealth of image data [4] and image model-based studies [6, 42, 7, 24, 3] available for video applications. Furthermore, generating UAPs using image data requires relatively less computation compared to using video data. However, we face significant challenges due to the lack of access to video data [32, 17] and video models [9, 45, 36]. There are two main challenges in generating adversarial videos using image models only [12, 30, 15]. Firstly, image models have limited capability in effectively analyzing the passage of time, which is a crucial aspect for videos. Secondly, UAPs should be applicable to unseen videos of varying lengths. Despite the importance of temporal information, prior research has not been able to address these challenges.

As the first paper to consider temporal information in video attacks using image models and data, our study addresses this issue with the **Breaking Temporal Consistancy (BTC)** method, as illustrated in Fig. 2. Our target UAP is a video consisting of N frames. Motivated by the high similarity pattern between neighboring frames in the original video, our UAP aims to generate adversarial videos that have opposite patterns to the original. To achieve this, we jointly optimize the adversarial and temporal aspects of the UAPs. First, to make the UAPs adversarial, we minimize the feature similarity between the original and adversarial images in the feature space using the *Adversarial Loss*. We treat the frames of the UAPs as images, and add them to the original to create corresponding adversarial images. To ensure universality across unseen datasets and prevent overfitting, we incorporate randomness using the Feature Diversity method. Second, we minimize the similarity between each frame of the UAPs using the *Temporal Similarity Loss*. To achieve this, we treat the adversarial images as a pseudo-video sequence and minimize the similarity among them.

We named our proposed UAP as BTC-UAP, which stands for Breaking Temporal Consistancy Universal Adversarial Perturbation. To ensure length-agnosticity of BTC-UAP, we apply it repeatedly until it covers entire frames of the video. Moreover, our approach is temporal shift invariant, meaning that the starting point of the UAP is irrelevant. Through extensive experiments on various datasets, including ImageNet, UCF-101, and Kinetics-400, we demonstrate that our simple but effective approach achieves superior performance compared to existing methods.

To summarize our study:

- We propose a novel video UAP using image data and image models, which allows us to leverage the wealth of image data and image model-based studies available for video applications.

- Our study proposes the Breaking Temporal Consistancy method as the first attempt to incorporate temporal information into video attacks using image models. Our BTC-UAP makes adversarial videos with opposite patterns to the original by minimizing the feature similarity between neighboring frames in videos.

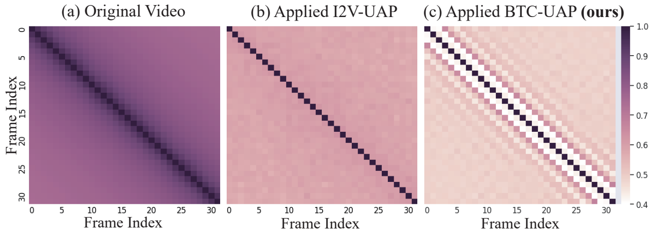

Figure 3: **The feature similarity of frames within videos.** This heatmap shows the average feature similarity between frames in the UCF-101 dataset, with brighter colors indicating lower levels of similarity.

- BTC-UAP is both temporal shift invariant and length-agnostic, making it a highly practical video attack method that can be applied to videos of varying lengths and datasets. We demonstrate the effectiveness of BTC-UAP through extensive experiments on various datasets, including ImageNet, UCF-101, and Kinetics-400, outperforming existing methods.

2. Related Work

2.1. Adversarial Attacks

Deep learning models are effective in computer vision tasks, but they can be easily fooled by adding imperceptible noise, which is known as adversarial perturbations. The adversarial perturbation is added to the original data to create an adversarial example, and using this adversarial example to attack a deep learning model is called an adversarial attack.

2.2. Image Classification Attacks

As studies on adversarial attacks began with tricking image classification models, various image classification attack methods have been developed [10, 20, 6, 42, 7, 24, 3, 31]. In the first stage, white-box image-specific adversarial attack methods were introduced. Fast Gradient Sign Method (FGSM) [10] creates adversarial examples by updating an input image with its gradient calculated to increase the classification loss. FGSM evolved into an iterative method called Iterative Fast Gradient Sign Method (I-FGSM) [20]. I-FGSM iteratively updates the input image with its gradients calculated in the same way as FGSM. Then, Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [6] achieved better performance by integrating momentum during the iterative updates of I-FGSM.

Afterward, the transfer-based black-box attack methods have emerged. Diverse Input (DI) method [42] increases the transferability of adversarial examples by performing random resizing and random padding to input images at each iteration. Translation-Invariant (TI) method [7] uses multiple translated images to generate an adversarial perturbation, rather than using a single input image. They efficiently approximate this process by applying a convolutional operation with a kernel to the gradient obtained from a single input image without any translation. Scale-Invariant (SI) attack method [24] improves the transferability of adversarial examples by using a scaled copy of the input image to compute the gradient.

[26] showed the existence of a single adversarial perturbation that can fool image classifier models when added to any input images. This single perturbation is called a Universal Adversarial Perturbation (UAP). There are many studies on UAP designed for deep learning models that deal with images [26, 27, 11, 18, 28, 25, 46, 21, 47].

2.3. Video Classification Attacks

There are several methods to create UAPs for video classification models [37, 14, 43, 23]. [23] trains a Generative Adversarial Network (GAN) to generate UAPs, and [43] optimizes a noise generator to create a UAP. [37] and [14] are optimization-based white-box UAPs. In white-box settings, [37] introduced an optimization-based algorithm for generating adversarial perturbations on the whole video, specifically on LSTM-based models. They proposed to regularization to concentrate perturbations on key frames. Similarly, a one-frame attack [14] only adds adversarial noise to one selected video frame. The researchers choose a vulnerable frame and perturbed it using the I-FGSM attack method. Similar to [14], there are other key frame selection attack methods [38, 44, 8] for both white-box and black-box settings.

In black-box settings, there are query-based video classification attacks [16, 22, 48, 41] and transfer-based video classification attacks [39, 40], similar to image classification attacks. [39] introduced a method called TT (Temporal Translation) to enhance the transferability of video adversarial examples. They prevent overfitting the source model by optimizing over a set of video clips that have been translated in time for each video. I2V (Images to Videos) method [40] achieved better transferability without relying on video models. I2V minimizes the similarity between the features of the original video frames and the adversarial video frames obtained by the ImageNet pre-trained image model. These perturbations optimized with the image model applied to the videos to attack video models. Both previous works (TT and I2V) have significantly improved transferability, but they have the limitation of requiring optimization for each individual video, which is not the case for UAP.

Algorithm 1 BTC-UAP Attack Method

```
Input      : Image dataset $X \subset \mathbb{R}^{C\times H\times W}$,
             image classification model $f(\cdot)$
Parameter: Perturbation budget $\epsilon$,
             number of layer $l$, step size $\alpha$,
             number of frames of BTC-UAP $N$,
             number of random noises $K$,
             set of temporal distance of neighbors $J$
Output     : BTC-UAP $\delta_N \in \mathbb{R}^{N\times C\times H\times W}$

Initialize $n \leftarrow 1$
Initialize all $\delta_N$ elements to $\frac{0.01}{255}$
for $x \in X$ do
    ▷ Compute BTC-Loss (7) with $l, J, K$ and $f$:
    $loss = \mathcal{L}_{BTC}(x, n, \delta_N)$
    ▷ Update $\delta_N^n \in \delta_N$ by Adam optimizer:
    $\delta_N^n \leftarrow Adam(loss, \alpha)$
    $\delta_N^n \leftarrow clip_\epsilon(\delta_N^n)$
    $n \leftarrow n + 1$
    if $n > N$ then
        $n \leftarrow 1$
    end if
end for
return $\delta_N = \{\delta_N^1, ..., \delta_N^N\}$.
```

3. Methodology

In this section, we describe the Breaking Temporal Consistancy method for generating the BTC-UAP using an image classification model that takes images or video frames as input. This approach does not require any prior knowledge about the target video data or model and can fool the video model into producing an incorrect prediction.

3.1. Problem Definition

We consider a video $V \in \mathbb{R}^{T\times C\times H\times W}$ and aim to generate an adversarial video V^{adv} by adding a BTC-UAP $\delta_N \in \mathbb{R}^{N\times C\times H\times W}$ to V. Here, T, C, H, W, and N denote the frames of the video, channels, height, width, and frames of the UAP, respectively. To represent each frame of the δ_N, we use $\delta_N^n \in \mathbb{R}^{C\times H\times W}$, where $n = 1, ..., N$ is the frame index. To ensure the imperceptibility of the perturbation, we constrain δ_N to have an l_∞-norm, as in previous works [40, 14].

The value of N is either less than or equal to T, and if $N < T$, we repeat the UAP in the frame dimension until it covers all T frames of the video. We define the repeated UAPs as $\delta_T \in \mathbb{R}^{T\times C\times H\times W}$, where $\delta_T = \{\delta_T^1, ..., \delta_T^T\}$ is obtained by repeating the original UAP $\delta_N = \{\delta_N^1, ..., \delta_N^N\}$ in the frame dimension until it covers entire T frames of the video. We can represent this operation as follows:

$$\delta_T = Repeat(\delta_N) = \{\underbrace{\delta_N^1, ..., \delta_N^N, \delta_N^1, ..., \delta_N^N, \delta_N^1, ...}_{\text{repeated until it covers } T \text{ frames}}\}. \tag{1}$$

Let $g(\cdot)$ be a video recognition model, and y be the true label of V. Our goal is to find a perturbation δ_N that misleads the video model's prediction:

$$g(V + \delta_T) \neq y, \quad s.t. \quad ||\delta_T||_\infty \leq \epsilon. \tag{2}$$

To achieve this, we optimize δ_N with $f(\cdot)$, which represents a image classification model.

3.2. Feature Similarity Analysis of Video Frames

In this section, we measure the average similarity of features obtained between video frames in the dataset. Since feature maps represent characteristics of an image, we use them to compare the similarity between frames. Figure 3 represents the feature similarity between two frames of the videos. For example, the diagonal represents the similarity between identical frames, so it always has the value of 1. To obtain the value, we input each frame of the video to an image model and measured the similarity Sim at a specific feature level using cosine similarity. The similarity between vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is expressed as follows:

$$Sim(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \, \|\mathbf{x}_2\|}. \tag{3}$$

To represent each frame of the video, we use $V^t \in \mathbb{R}^{C\times H\times W}$, where $t \in T$ is the frame index. We extract the feature map $F(\cdot)$ from a specific layer l of an image classification model $f(V^t)$ and denote this feature map by $F_l(V^t)$. We visualize the similarity of frames within an original video V in Figure 3-(a). We observe that the original videos tend to have high levels of similarity between consecutive frames.

Furthermore, we extend the non-UAP I2V method [40] to create an I2V-UAP. To make I2V universal, we optimize one perturbation for multiple videos within the dataset. To observe the effects of UAPs on the feature maps, we create an adversarial example V_{adv}^t by adding the UAP δ_T^t to the original frame V^t and extract the feature maps $F_l(V_{\text{adv}}^t)$ in the same way as for the original frame. Applying the I2V-UAP shown in Figure 3-(b) results in a reduction in similarity across all frames.

We further observe that adversarial videos disrupt the high similarity pattern of consecutive frames in the original videos. Based on this observation, we propose the BTC method to generate adversarial videos with opposite patterns to the original videos. Details of our method can be found in Section 3.3. Our proposed BTC-UAP, as shown in Figure 3-(c), generates a completely opposite pattern of similarity to the original video, with neighboring frames having low similarity. As we intentionally make the features of consecutive frames less similar to each other, the overall similarity between frames decreases when compared to the original video. These results indicate that neighboring frames are recognized as different images by image models. These effects are contrary to the original characteristics of the video, and our experiments in Section 4 demonstrates that BTC-UAP effectively confuses video models.

3.3. Breaking Temporal Consistancy Method

In this section, we focus on Breaking Temporal Consistancy method and discuss how to optimize the BTC-UAP using image data and models. Let $x \in \mathbb{R}^{C\times H\times W}$ be an image, which can be a frame of a video $V^t \in \mathbb{R}^{C\times H\times W}$. Our goal is to find a universal adversarial perturbation $\delta_N^n \in \delta_N$ using images. The overall optimization process is described in Algorithm 1.

Adversarial Loss. Feature maps represent characteristics and patterns of an image, which can be used to create adversarial examples. Therefore, decreasing the similarity between the feature representations $F(\cdot)$ of original images x and adversarial examples $x^n_{adv} = x + \delta^n_N$ will result in the UAP causing confusion in the information of the original image. To ensure that the BTC-UAP is effective against other data and prevent overfitting to the training dataset, we propose *Feature Diversity* method with a total of K random noises. This involves adding a random noise $\eta_k \in [-\epsilon, \epsilon]^{C\times H\times W}$ to each x to increase diversity to avoid overfitting. This simple method is highly effective in improving the performance of the UAP framework. The adversarial loss can be expressed mathematically as follows:

$$\mathcal{L}_{adv}(x,n,\delta_N) = \sum_{k=1}^{K} Sim(F_l(x+\eta_k), F_l(x^n_{adv})). \quad (4)$$

Temporal Similarity Loss. Our approach presents a novel solution to the issue that image models are unable to fully consider the temporal dimension, in contrast to video models. Our goal is to minimize the similarity between neighboring frames in videos using the optimized δ_N. To successfully deceive a video model, we introduce confusion in the temporal domain through the use of $f(\cdot)$, by decreasing similarity between the neighboring frames. To achieve this, we generate the adversarial image $x^{\text{adv}}_{n+j} = x + \delta_N^{n+j}$ and then treat the sequence of adversarial images as a pseudo video. Here, $j \in J$ and J represents the set of temporal distances of neighbors, such as $J = \{-2, -1, 1, 2\}$.

Dataset	Networks						
	SF-101	SF-50	TPN-50	TPN-101	NL-50	NL-101	AVG.
UCF-101	90.2	91.7	91.7	93.6	86.9	88.4	90.4
Kinetics	69.8	71.0	73.9	75.0	69.5	69.5	71.4

Table 1: Clean Accuracy

To reduces the similarity between x^n_{adv} and x^{n+j}_{adv}, we extract feature of adversarial images $F(x_{adv})$ using the image model $f(\cdot)$ and calculate the similarity between them, following Eq.3. The temporal similarity loss can effectively cause confusion in the temporal information when the perturbations δ_N is added to video along the temporal axis. This temporal similarity loss can be expressed mathematically as follows:

$$\mathcal{L}_{temp}(x,n,\delta_N) = \sum_{j\in J} Sim\left(F_l(x^n_{adv}), F_l(x^{n+j}_{adv})\right). \quad (5)$$

Compared to previous approaches, our method allows us to effectively minimize the temporal similarity between perturbed frames, enabling us to produce more robust adversarial examples.

By considering both adversarial and temporal aspects using image-based approaches, the proposed BTC-UAP can effectively perturb both types of information and successfully attack video models. To optimize δ_N, we utilized a Breaking Temporal Consistancy Loss that is the sum of the adversarial loss and temporal similarity loss, mathematically represented as follows:

$$\mathcal{L}_{BTC}(x,n,\delta_N) = \mathcal{L}_{adv} + \mathcal{L}_{temp}. \quad (6)$$

Finally, we can get optimized BTC-UAP δ_N^{n*} by minimizing BTC-Loss with l, J, K and f:

$$\delta_N^{n*} = arg\min_{\delta_N^n} \mathcal{L}_{BTC}(x,n,\delta_N). \quad (7)$$

4. Experiment

4.1. Experiment Settings

We evaluate the Attack Success Rates (ASR) of UAPs in the following settings. The ASR indicates the rate at which the target model misclassifies the adversarial examples into the wrong label. A higher ASR indicates that the UAPs achieve higher transferability.

Datasets. We refer to the data used to generate UAPs as the source data, and the data where the UAP is added to create adversarial examples as the target data. We conducted experiments using various datasets. ImageNet [4] is a large image dataset with 1,000 classes. We used the ImageNet train set as source data, selecting 10 images per one class. UCF-101 [32] and Kinetics-400 [17] are video classification datasets that label human action categories. UCF-101 has 13,320 videos with 101 action classes, and Kinetics-400 has 650,000 videos with 400 classes. We used the UCF-101

Source Dataset	Source Models	Attack	Target Models						
			SF-50 (Kinetics)	SF-101 (Kinetics)	TPN-50 (Kinetics)	TPN-101 (Kinetics)	NL-50 (Kinetics)	NL-101 (Kinetics)	AVG.
UCF-101	SF-101 (UCF-101)	All-UAP	76.12	*98.59*	49.76	45.90	48.41	48.76	61.26
		TT-UAP	75.92	*98.74*	66.59	58.97	62.55	62.55	70.88
	TPN-101 (UCF-101)	All-UAP	58.81	58.29	62.91	*79.95*	51.29	48.49	59.95
		TT-UAP	53.99	53.16	48.85	*44.68*	43.86	41.62	47.69
	NL-101 (UCF-101)	All-UAP	52.75	50.21	39.78	37.93	**69.80**	*81.32*	55.30
		TT-UAP	59.69	60.21	60.60	56.20	67.41	*80.52*	64.10
	Res-101 (ImageNet)	I2V-UAP	66.29	62.15	82.40	71.78	51.85	50.18	64.11
		BTC-UAP	82.49	77.30	93.35	**86.62**	67.40	**67.83**	79.16
ImageNet	Res-101 (ImageNet)	I2V-UAP	69.31	64.37	83.85	73.64	53.26	50.97	65.90
		BTC-UAP	**83.26**	**77.63**	**93.49**	86.23	68.27	66.97	**79.31**

Table 2: **Comparison with UAPs generated on video models.** UAPs are optimized on the source datasets UCF-101 and ImageNet, respectively. The generated UAPs are *repeated* and added to Kinetics-400 videos until they cover the entire video. The bold numbers indicate the highest attack success rates (%) in each column. The gray color represents the *white-box setting*, where the source and target models are identical.

Source Dataset	Source Models	Attack	Target Models						
			SF-50 (UCF-101)	SF-101 (UCF-101)	TPN-50 (UCF-101)	TPN-101 (UCF-101)	NL-50 (UCF-101)	NL-101 (UCF-101)	AVG.
UCF-101	SF-101 (UCF-101)	All-UAP	48.74	*98.93*	13.42	8.62	21.18	14.14	34.17
		TT-UAP	43.28	*96.81*	23.25	17.01	38.38	50.46	44.86
	TPN-101 (UCF-101)	All-UAP	17.62	12.88	19.93	*94.54*	26.83	27.00	33.13
		TT-UAP	14.41	8.11	9.88	*6.88*	17.01	15.48	11.96
	NL-101 (UCF-101)	All-UAP	19.20	10.23	8.17	5.52	**56.19**	*97.96*	32.88
		TT-UAP	20.84	18.02	23.73	21.34	51.47	*96.79*	38.70
	Res-101 (ImageNet)	I2V-UAP	24.24	16.71	39.21	27.02	24.18	40.01	28.56
		BTC-UAP	47.78	35.62	64.43	46.55	50.43	61.89	51.12
ImageNet	Res-101 (ImageNet)	I2V-UAP	25.60	18.45	42.55	29.16	25.82	41.27	30.48
		BTC-UAP	**49.01**	**36.98**	**65.37**	**47.67**	49.41	**63.34**	**51.96**

Table 3: **Comparison with UAPs generated on video models.** UAPs are optimized on the source datasets UCF-101 and ImageNet, respectively. Adversarial videos are generated by adding UAPs to UCF-101 videos. The bold numbers indicate the highest attack success rates (%) in each column for the UCF-101 dataset. The gray color represents the *white-box* setting, where the source model and target model are identical.

test set and Kinetics-400 validation set. For Kinetics-400, we randomly chose 5 videos per a class.

Models. We used three pre-trained image models on the ImageNet dataset: ResNet101 (Res-101) [12], SqueezeNet (Squeeze) [15], and VGG16 [30]. These models were used as source models to generate adversarial examples. We used six different video models: SlowFast-50 (SF-50), SlowFast-101 (SF-101)[9], Temporal Pyramid Network-50 (TPN-50), Temporal Pyramid Network-101 (TPN-101) [45], NonLocal-50 (NL-50), and NonLocal-101 (NL-101) [36]. Each six models trained on UCF-101[1] and Kinetics-400[2] datasets, for a total of 12 video models are used to evaluate the performance. UCF-101 models were tested on 32-frame videos, while Kinetics-400 models were tested on 64-frame videos. Table 1 shows the accuracy of the models on clean data.

Baselines. There is no UAP framework for transfer-based video attacks using image datasets and models. To compare performance, we adapted the cross-modal video attack method [40] and the transfer-based video attack method [39] to the UAP scenario (I2V-UAP and TT-UAP). ALL-UAP indicates UAP based on I-FGSM method [20]. We also compared our method with image transfer-based attack methods, including MI [6], DI [7], TI [42], and SI [24] in Table 4.

Hyperparameters. The perturbation budget ϵ was set to $16/255$, and the step size α was set to 0.004. We used the number of feature layers l to optimize BTC-UAP and I2V-UAP, following the previous cross-modal attack [40]. We randomly selected an image or one frame per a video for BTC-UAP optimization, and set the number of UAP frames N to 32. The number of random noise K was set to 4 and the set of temporal distance J to $\{-2, -1, +1, +2\}$.

In Section 4.4, we show how we selected the hyper-parameters for BTC-UAP. The implementation details for

[1] We used the trained models from https://github.com/zhipeng-wei/Image-to-Video-I2V-attack.

[2] We downloaded the pretrained models from gluon.

Source Models	Attack	Target Models						
		SF-50 (Kinetics)	SF-101 (Kinetics)	TPN-50 (Kinetics)	TPN-101 (Kinetics)	NL-50 (Kinetics)	NL-101 (Kinetics)	AVG.
Res-101 (ImageNet)	MI-UAP	61.42	57.46	58.02	54.43	49.36	47.37	54.36
	DI-UAP	60.55	56.63	59.26	58.66	51.64	48.91	55.94
	TI-UAP	60.56	56.63	59.26	58.64	56.57	54.41	57.68
	SI-UAP	65.98	64.61	71.26	70.13	55.72	55.30	63.83
	I2V-UAP	69.31	64.37	83.85	73.64	53.26	50.97	65.90
	BTC-UAP	**83.26**	**77.63**	**93.49**	**86.23**	**68.27**	**66.97**	**79.31**
VGG16 (ImageNet)	MI-UAP	54.42	52.56	49.53	45.43	45.64	43.44	49.11
	DI-UAP	57.96	55.47	51.97	48.22	49.25	45.96	51.47
	TI-UAP	57.95	55.45	51.95	48.21	55.51	52.63	53.61
	SI-UAP	59.63	58.29	56.22	51.25	50.19	48.67	54.04
	I2V-UAP	57.82	52.73	60.40	54.90	44.62	41.92	52.06
	BTC-UAP	**74.25**	**75.05**	**82.97**	**75.68**	**68.59**	**63.93**	**73.41**
Squeeze (ImageNet)	MI-UAP	53.91	52.92	52.22	49.55	45.64	43.44	49.61
	DI-UAP	54.61	53.42	52.22	49.24	45.24	43.61	49.72
	TI-UAP	66.25	66.05	59.55	54.79	57.52	**55.06**	59.87
	SI-UAP	58.36	58.40	54.51	49.86	47.70	45.70	52.42
	I2V-UAP	66.20	63.40	66.50	58.94	54.30	50.60	59.99
	BTC-UAP	**70.71**	**68.14**	**71.29**	**65.25**	**58.88**	54.72	**64.83**

Table 4: **Attack success rates (%) of UAPs generated on image models using image data.** UAPs are optimized on ImageNet and adversarial videos are generated by adding UAPs to Kinetics-400 videos. The generated UAPs are *repeated* and added to Kinetics-400 videos until they cover the entire video. The bold numbers indicate the highest attack success rate among attack methods.

other baselines are in the supplementary material.

4.2. Experimental Results

4.2.1 Comparison with Video-based attack method

We evaluated the transferability of UAPs optimized on UCF-101 in Tables 2 and 3. Table 2 shows the performance of UAPs on each target model trained on Kinetics-400, evaluated by adding the UAPs to Kinetics-400 videos. In Table 3, we divided the UCF-101 dataset into two groups and evaluated methods on the unseen group. The gray color in the tables represents the *white-box setting*, where the target model is used as the source model during UAP generation. Please note that our method aims to transfer the UAPs generated from image models to video models for cross-modal attacks, which cannot be conducted under the white-box setting. Excluding the white-box evaluation, BTC-UAP achieves the highest transferability in most cases. For example, in Table 2, BTC-UAP (Res-101, ImageNet) achieved the highest average ASR of 79.31%, compare to All-UAP(SF-101, UCF-101) with 61.26% and TT-UAP(SF-101, UCF-101) with 70.88%.

When compared to UAPs optimized using videos as the source data, the performance of BTC-UAP generated on image data is comparable or even better. This demonstrates that our Breaking Temporal Consistancy method can effectively consider temporal information, even without video models or data, and achieve superior performance compared to I2V-UAP. Furthermore, in Table 2, the generated UAP was optimized for 32 frames, while the evaluation on Kinetics-400 was conducted on 64 frames. Therefore, we *repeated* the UAP without optimizing it for 64 frames to generate universal adversarial perturbations for Kinetics-400, following Eq.1. Despite the challenging condition of evaluating on 64 frames while the generated UAP was optimized for 32 frames, BTC-UAP still achieves high transferability in attacking video classification models. This demonstrates that our method is effective even in complete black-box situations, such as evaluating on an unseen video model with a different number of video frames.

4.2.2 Comparison with Image-based attack method

We conducted experiments to evaluate the transferability of UAPs generated using image data and models. Table 4 shows the ASR of adversarial videos, where UAP is optimized on ImageNet using each rightmost image model. In this experiment, the 32-frame UAPs are repeatedly added to Kinetics-400 videos to create adversarial videos, following Eq.1. Compared to other methods, BTC-UAP achieved the highest average ASR and demonstrated good transferability. For instance, in Table 4, the I2V-UAP has a total average ASR 60.31 % on all cases, while BTC-UAP shows superior performance with 70.79 %. This result demonstrates that our proposed method effectively considers temporal information, resulting in the highest performance among image-based methods.

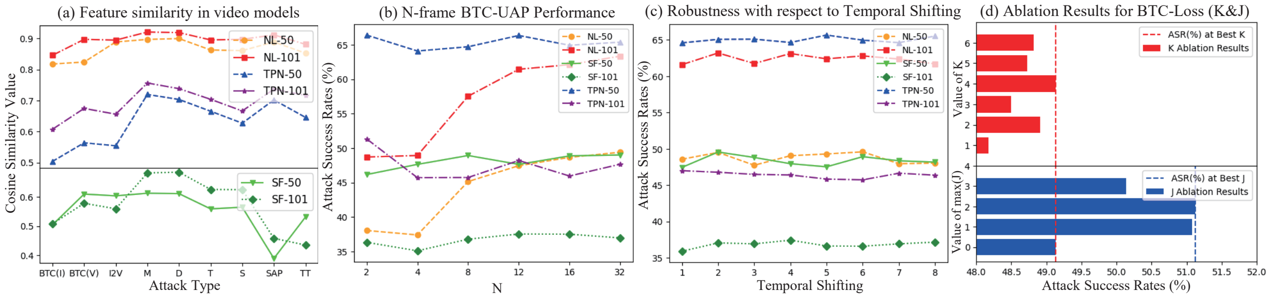

Figure 4: **Analysis and Ablation Results.** Results demonstrate that the proposed Breaking Temporal Consistency method leads to superior robustness against perturbations for videos.

4.3. Discussion

In this section, we conducted experiments to demonstrate the effectiveness of our proposed Breaking Temporal Consistency Loss for creating adversarial examples in video-based attacks. We applied the BTC-UAP generated using Res-101 and ImageNet data, to 32-frame UCF-101 videos and analyzed its effectiveness on six different models.

To analyze the performance of our proposed method, we compared the cosine similarity between the original and adversarial videos in the video model. Figure4-(a) shows cosine similarity scores between the original and adversarial videos for each attack method under black-box settings. The bottom graph shows the same comparison under white-box settings. when all attack settings are black-box, our proposed method achieved the lowest similarity score among all the attack methods. In the context of confusing video models, we found that BTC(I), which is generated using image data, is more effective than BTC(V), which is generated using video data. The results show that BTC(I) had a greater impact the UAPs generated with video models despite being generated using image models, highlighting its superior robustness.

To demonstrate the effectiveness of the proposed method with a small number of BTC-UAP frames, we applied the UAP iteratively with a small value of N, repeating a subset of N frames within the total of $T = 32$ frames in the adversarial video. We compared the results for N=2,4,8,12,16 and 32. Figure4-(b) compares the performance of BTC-UAP with different numbers of N. Even when N=2, our proposed method exhibits comparable performance, demonstrating its effectiveness even with a small number of UAP frames. We further demonstrated the shifting invariance of our proposed BTC-UAP by conducting experiments in which we shifted the UAP along the temporal axis from 1 to 8 frames. Figure4-(c) demonstrates the shifting invariance of BTC-UAP by displaying attack success rates for different temporal shifts of the UAP frames. It showed that the attack success rate was consistent regardless of the temporal shifting. These results demonstrated the BTC-UAP is robustness against temporal shifts and the effectiveness even with a small number of optimized frames.

4.4. Ablation study

In this section, we explore the effects of the most critical parameters, K and J, in our BTC-method. Specifically, we investigate the impact of the number of random noise K employed in the adversarial loss and the temporal distance of neighbors set J utilized in the temporal similarity loss. Figure 4-(d) shows that $K = 4$ provided the best performance in terms of the adversarial loss. We then conducted experiments with different symmetric sets of J while keeping K fixed at 4. In the graph, please note that we represented the highest value among the set of J on the y-axis for convenience. Our results showed that when the $max(J) = 2$, the use of a set $J = \{-2, -1, 1, 2\}$ achieved the highest performance.

Importantly, we observed that although the computations required for $K = 6$ and $K = 4$ with $J = \{-1, 1\}$ were the same, the latter yielded significantly better performance. This demonstrates that reducing the similarity between frames was a more effective approach to improving performance than simply increasing computational resources.

5. Conclusion

In this paper, we proposed the Breaking Temporal Consistency Method, which was the first to attack videos using only image models while considering temporal information. Our method was designed to minimize the similarity between neighboring frames, by jointly optimizing adversarial and temporal similarity losses. Specifically, by using adversarial loss, we reduced the similarity between original and adversarial examples, and by using temporal similarity loss, we reduced the similarity between UAPs. BTC-UAP was both temporal shift invariant and length-agnostic. Our extensive experiments on various datasets demonstrated the effectiveness of our proposed BTC-UAP .

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] Junyoung Byun, Seungju Cho, Myung-Joon Kwon, Hee-Seon Kim, and Changick Kim. Improving the transferability of targeted adversarial examples through object-based diverse input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15244–15253, 2022.

[4] 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.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019.

[6] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[7] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[8] Zhenyu Du, Fangzheng Liu, and Xuehu Yan. Sparse adversarial video attacks via superpixel-based jacobian computation. *Sensors*, 22(10):3686, 2022.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015.

[11] Jamie Hayes and George Danezis. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE, 2018.

[12] 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.

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[14] Jaehui Hwang, Jun-Hyuk Kim, Jun-Ho Choi, and Jong-Seok Lee. Just one moment: Structural vulnerability of deep action recognition against one frame attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7676, 2021.

[15] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. 2017.

[16] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019.

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

[18] Valentin Khrulkov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8562–8570, 2018.

[19] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14254–14263, 2020.

[20] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. 2016.

[21] Maosen Li, Yanhua Yang, Kun Wei, Xu Yang, and Heng Huang. Learning universal adversarial perturbation by adversarial example. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1350–1358, 2022.

[22] Shasha Li, Abhishek Aich, Shitong Zhu, Salman Asif, Chengyu Song, Amit Roy-Chowdhury, and Srikanth Krishnamurthy. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. *Advances in Neural Information Processing Systems*, 34:2085–2096, 2021.

[23] Shasha Li, Wenjie Li, Diane Cook, and Shuang Zhu. Stealthy adversarial perturbations against real-time video classification systems. In *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS)*, pages 1–16, 2018.

[24] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020.

[25] Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal adversarial perturbation via prior driven uncertainty approximation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2941–2949, 2019.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[27] KR Mopuri, U Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. In *British Machine Vision Conference 2017, BMVC 2017*. BMVA Press, 2017.

[28] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018.

[29] Roi Pony, Itay Naeh, and Shie Mannor. Over-the-air adversarial flickering attacks against video recognition networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 515–524, 2021.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[31] Minji Son, Myung-Joon Kwon, Hee-Seon Kim, Junyoung Byun, Seungju Cho, and Changick Kim. Adaptive warping network for transferable adversarial attacks. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3056–3060. IEEE, 2022.

[32] 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.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[35] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[37] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019.

[38] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12338–12345, 2020.

[39] Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Boosting the transferability of video adversarial examples via temporal translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2659–2667, 2022.

[40] Zhipeng Wei, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Cross-modal transferable adversarial attacks from images to videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15064–15073, 2022.

[41] Zhipeng Wei, Jingjing Chen, Hao Zhang, Linxi Jiang, and Yu-Gang Jiang. Adaptive temporal grouping for black-box adversarial attacks on videos. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 587–593, 2022.

[42] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[43] Shangyu Xie, Han Wang, Yu Kong, and Yuan Hong. Universal 3-dimensional perturbations for black-box attacks on video recognition systems. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1390–1407. IEEE, 2022.

[44] Yixiao Xu, Xiaolei Liu, Mingyong Yin, Teng Hu, and Kangyi Ding. Sparse adversarial attack for video via gradient-based keyframe selection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2874–2878. IEEE, 2022.

[45] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 591–600, 2020.

[46] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Cd-uap: Class discriminative universal adversarial perturbation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6754–6761, 2020.

[47] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7868–7877, 2021.

[48] Hu Zhang, Linchao Zhu, Yi Zhu, and Yi Yang. Motion-excited sampler: Video adversarial attack with sparked prior. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 240–256. Springer, 2020.