Search-and-Attack: Temporally Sparse Adversarial Perturbations on Videos

HWAN HEO\textsuperscript{1}, DOHWAN KO\textsuperscript{1}, JAEWON LEE\textsuperscript{1}, YOUNGJOON HONG\textsuperscript{2}, AND HYUNWOO J. KIM\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, Korea University, Seoul 02841, Republic of Korea
\textsuperscript{2}Department of Mathematics, Sungkyunkwan University, Suwon 16419, Republic of Korea

Corresponding authors: Youngjoon Hong (hongyj@skku.edu) and Hyunwoo J. Kim (hyunwoojkim@korea.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grants by the Korean Government through the Ministry of Science and ICT (MSIT) (Development of Integrated Cognitive Drone AI for Disaster/Emergency Situations) under Grant 2021-0-00025, in part by the ICT Creative Consilience Program under Grant IITP-2021-2020-0-01819, in part by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET), in part by the Korea Smart Farm Research and Development Foundation (KosFarm) through the Smart Farm Innovation Technology Development Program by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) through the Rural Development Administration (RDA) under Grant 421025-04, and in part by Samsung Electronics Company Ltd.

\textbf{ABSTRACT} Modern neural networks are known to be vulnerable to adversarial attacks in various domains. Although most attack methods usually densely change the input values, recent works have shown that deep neural networks (DNNs) are also vulnerable to sparse perturbations. Spatially sparse attacks on images or frames of a video have been proven effective but the temporally sparse perturbations on videos have been less explored. In this paper, we present a novel framework to generate a temporally sparse adversarial attack, called Search-and-Attack scheme, on videos. The Search-and-Attack scheme first retrieves the most vulnerable frames and then attacks only those frames. Since identifying the most vulnerable set of frames involves an expensive combinatorial optimization problem, we introduce alternative definitions or surrogate objective functions: Magnitude of the Gradients (MoG) and Frame-wise Robustness Intensity (FRI). Combining them with iterative search schemes, extensive experiments on three public benchmark datasets (UCF, HMDB, and Kinetics) show that the proposed method achieves comparable performance to state-of-the-art dense attack methods.

\textbf{INDEX TERMS} Action recognition, video classification, adversarial attack, sparse adversarial attack, safe AI.

\section{I. INTRODUCTION}

In recent years, deep neural networks (DNNs) have shown great performance in various tasks such as image classification \cite{1, 2, 3, 4} and object detection \cite{5, 6, 7, 8}. Despite the success of the modern deep neural networks, DNNs are known to be vulnerable to adversarial attacks, which are carefully crafted perturbations to fool machine learning models. Even though generating optimal adversarial perturbations is an NP-hard problem \cite{9, 10}, simple gradient-ascent based attack algorithms \cite{11, 12, 13, 14} have proven effective to deceive deep neural networks. To generate small and possibly imperceptible adversarial perturbations, various constraints (e.g., $l_{\infty}$, or $l_2$-norm) are often imposed.

While much effort has been devoted to adversarial attacks on the image, adversarial attacks have been less studied in the video domain. Recently, several studies \cite{15, 16, 17} have been conducted to investigate adversarial attacks on videos, but they do not fully leverage the relation of frames. More precisely, those studies have either attacked all frames or regularly sampled frames without considering temporal dependency between frames. This line of approaches is sub-optimal to attack video classification models that utilize the temporal information of videos. Even though most state-of-the-art video classification models use 3D convolution \cite{18, 19, 20} to capture the temporal information, the models differently cope with the temporal dynamics by using different strides along the time direction for their efficiency and flexibility. SlowFast Networks \cite{21} use even two separate 3D convolutional neural networks with two different frame rates. Exploiting both the common trait of video classification models such as 3D convolution and architecture-specific vulnerability is a key to develop less perceptible (or temporally sparse) and stronger adversarial attacks.
In this paper, we propose a simple yet effective method for temporally sparse adversarial attacks against video classification models. We propose a two-stage pipeline called the Search-and-Attack, which finds the most vulnerable frames of a video at the Search stage, and then perturbs only the frames at the Attack stage. We formally define the temporally sparse attacks and the most vulnerable frame set. Identifying the most vulnerable frames involves a nonlinear mixed integer programming. For efficient optimization, instead of the conventional loss function, we propose surrogate objective functions and find vulnerable frames with respect to the surrogate losses. With the surrogate losses, we explore a single-step method and two iterative methods for the frame selection. In the attack stage, we generate a frame-wise perturbation with a modified version of FGSM [12] and PGD [13] which only perturb the frames selected in the search stage. To test the performance of the proposed method, we carry out experiments on HMDB, UCF, and Kinetics datasets with widely used action-recognition models such as I3D, R3D, R(2+1)D, SlowFast, and IRCSNv2. The experiments show that attacking only a few vulnerable frames is as strong as attacking the entire frame. To sum up, our main contributions are fourfold.

- We formally introduce temporally sparse adversarial attacks for video action recognition models.
- We propose frame search algorithms with surrogate objective functions to identify vulnerable frames.
- We study model-specific temporal vulnerability of video classification models.
- Our experiments on HMDB, UCF, and Kinetics datasets demonstrate that the proposed methods successfully search vulnerable frames and generate strong temporally sparse adversarial perturbation on videos.

II. RELATED WORK

A. ACTION RECOGNITION

Many studies have investigated various models for action recognition. One approach is CNN+RNN based models, which train RNN models on a sequence of the frame features over time with CNNs [22]. Another common approach is 3D convolutional neural networks that take a set of frames as an input tensor. Some models learn spatial and temporal features concurrently [18], [19], while some models consist of separable 3D convolutions in which temporal and spatial convolutions are implemented separately [20]. In addition, there are two-stream CNNs with RGB images and optical flows. They use the information of short-term motions to supplement with the information of object appearance [18].

B. ADVERSARIAL ATTACK

Adversarial attack has been studied by a variety of approaches. The attack methods have been mostly demonstrated in image classification. Gradient-based methods are one of the most common methods for adversarial attack and they modify the input images with respect to computing the gradients of the cost function [12], [13]. Although these attacks usually densely change the input values, recent works have shown that DNNs are also vulnerable to sparse perturbations. For example, Jacobian-based Saliency Map Attack (JSMA) constructs a saliency map, which shows the impact of each pixel on the resulting classification, through forward gradients [23]. Based on the saliency map, the JSMA
iteratively attacks the most influential pixel. Moreover, [14] suggests optimization-based adversarial attacks that use distance metric $l_0$-norm. Recent works have studied sparse and imperceptible perturbations to introduce more effective attack methods; for more details, see e.g., [24]–[26].

C. ADVERSARIAL ATTACK TO VIDEO

Regarding the video domain, [15] has explored adversarial attacks on videos. They proposed a method that optimizes spatio-temporal sparse adversarial perturbation with a CNN+RNN [22]. However, they just sampled frames without considering temporal dependency between frames. On the other hand, [17] proposed a dense adversarial attack on a two-stream action-recognition model, which consists of optical flow and RGB. They extended FGSM and iterative FGSM attacks to the video domain [12], [27].

Unlike those optimization-based methods, attacks using a generative model have also been suggested. For example, [16] generated black-box adversarial perturbations to the real-time action-recognition model using Generative Adversarial Networks (GANs).

II. METHODS

In this section, we first formally introduce the temporally sparse adversarial perturbations. Then, we propose our Search-and-Attack methods that are efficient two-stage algorithms. At the first stage (frame selection), our algorithms search the most vulnerable frames with respect to surrogate objective functions, and at the second stage (perturbation generation), our algorithms attack the selected frames by FGSM or PGD.

A. TEMPORALLY SPARSE ADVERSARIAL ATTACKS

Let $\theta$ denote the parameters of neural networks and $x \in \mathbb{R}^{T \times C \times H \times W}$ denote an input video with its ground-truth target label $y$, where $T$, $C$, $H$, and $W$ are the numbers of frames, the number of channels, the height, and the width of a frame respectively. The $i$-th frame of an input video $x$ is denoted by $x_i \in \mathbb{R}^{C \times H \times W}$, i.e., $x = [x_1; x_2; \cdots; x_T]$. Our goal is to attack neural networks for video classification by temporally sparse perturbations. The goal can be achieved by finding the most vulnerable $N$ frames $I \in \mathbb{B}^T$ and perturbing only the selected frames. This can be written as follows:

$$\max_{\delta I} \mathcal{L}(x + \delta^I, y; \theta) \quad \text{subject to } |I| = N,$$

$$\|\delta^{(0)}\|_{\infty} \leq \epsilon, \quad \forall i \in I,$$

$$\delta^I = \sum_{i \in I} \delta^{(0)},$$

where $\mathcal{L}$ is the loss function and $\delta^{(0)} \in \mathbb{R}^{T \times C \times H \times W}$ denotes the perturbation on the $i$-th frame of the video, i.e., all elements of $\delta^{(0)}$ are zeros except those corresponding to the $i^{th}$ frame. In other words, the perturbation on the most vulnerable $N$ frames $\delta^I$ is defined as the sum of the frame-wise perturbation $\delta^{(0)}$ of the selected frames where $i \in I$.

Algorithm 1 Search-and-Attack Method

**Input:** Video $x$ and its label $y$, model parameters $\theta$, the number of attacked frames $N$, surrogate loss $J$, attack method $A$

**Output:** perturbed video $x'$

1. $I \leftarrow \text{Search}(x, y, \theta, N, J, A)$
2. $\delta^I \leftarrow A(x, y, \theta, I)$
3. $x' \leftarrow x + \delta^I$
4. return $x'$

It is worth noting that Eq. (1) is a mixed-integer non-linear program (MINLP). Since MINLP is an NP-hard combinatorial problem [28], no polynomial-time algorithm has been found yet to seek the global optimal solution. The feasible set of (1) has $(\binom{T}{N})$ solutions for $I$ and even for small problems; e.g., $T = 32, N = 4$, it is huge. Besides, the continuous variables $\delta^{(0)}$ need to be simultaneously optimized. Hence, in this paper, we instead decompose the problem in (1) into two subproblems: (i) search (frame selection by optimizing $I$), and (ii) attack (perturbation generation $\delta^I$ for the selected frames). Figure 1 illustrates the overall pipeline of our approach.

B. SURROGATE OBJECTIVE FUNCTIONS

At the search stage, we propose an alternative formulation with a surrogate objective function $J$ to perform the frame selection. The optimization problem in (1) is reduced to an integer programming problem as follows:

$$I^* = \arg \max_{I \in \mathbb{B}^T} J(x, y, I; \theta) \quad \text{subject to } |I| = N. \quad (2)$$

Alg. 1 shows how the generalized search method works.

1) MAGNITUDE OF THE GRADIENTS (MoG)

We derive a first surrogate objective function from the first-order Taylor expansion of the loss function $\mathcal{L}$. The Taylor series for the original objective function is given as follows:

$$\mathcal{L}(x + \delta^I, y; \theta) \approx \mathcal{L}(x, y; \theta) + \langle \nabla_x \mathcal{L}(x, y; \theta), \delta^I \rangle. \quad (3)$$

From the maximum perturbation constraint, i.e., $\|\delta^{(0)}\|_{\infty} \leq \epsilon$ in (1), the upper bound of the first order approximation can be derived as follows:

$$\mathcal{L}(x, y; \theta) + \langle \nabla_x \mathcal{L}(x, y; \theta), \delta^I \rangle \leq \mathcal{L}(x, y; \theta) + \epsilon \sum_{i \in I} \|\nabla_{x_i} \mathcal{L}(x, y; \theta)\|_1. \quad (4)$$

The upper bound is obtained by the sum of the $L_1$-norm of the gradient. So, we name our surrogate loss function as the sum of Magnitudes of Gradients (MoG). For a fixed data point $x$, since $\epsilon$ and $\mathcal{L}(x, y; \theta)$ are constant, we define $J_{\text{MoG}}$ as follows:

$$J_{\text{MoG}}(x, I, y; \theta) = \sum_{i \in I} \|\nabla_{x_i} \mathcal{L}(x, y; \theta)\|_1. \quad (5)$$
Algorithm 2 Greedy Search for Frame Selection

Input: Video $x$, label $y$, model parameters $\theta$, the number of frames to attack $N$, surrogate loss $J$, attack method $A$

Output: Frame Index Set $I$

1: function Search$_G(x, y, \theta, N, J, A)$
2: $I \leftarrow \phi$
3: $x_p \leftarrow x$
4: for $n = 1$ to $N$ do
5:   $\delta \leftarrow A(x_p, y, \theta, I)$
6:   $i^* \leftarrow \arg\max_{i \in I} J(x_p + \delta^{(i)}, y; \theta)$
7:   $I \leftarrow I + \{i^*\}$
8:   $x_p \leftarrow x_p + \delta^{(r)}$
9: end for
10: return $I$
11: end function

This has the same optimal solution as in Eq. (4) since $\arg\max f(x) = \arg\max af(x) + b$.

2) FRAME-WISE VULNERABILITY (FRI)

To propose the second surrogate loss function, we assume that the loss function is locally linear around data point $x$. More precisely, we assume that the increase of the loss value by $\delta^t$ is approximately similar to the sum of the increases induced by $\delta^{(i)}$. The second surrogate loss can be derived as follows:

$$L(x + \delta^t) - L(x) = L(x + \sum_{i \in I} \delta^{(i)}) - L(x) \approx \sum_{i \in I} \left(L(x + \delta^{(i)}) - L(x)\right).$$ (6)

To avoid clutter, $y$ and $\theta$ are omitted in the equation. Similar to MoG, the surrogate loss function can be further simplified by removing the constant $L(x)$. The second surrogate objective function $J$ is given as:

$$J_{\text{FRI}}(x, I, y; \theta) = \sum_{i \in I} L(x + \delta^{(i)}, y; \theta)$$ (7)

We name this surrogate loss as the sum of FRame-wise vulnerability (FRI). The frame-wise perturbation $\delta^{(i)}$ is generated by a simple method. More details will be discussed in Section III-D.

Note that the optimal solution $I^*$ with respect to MoG and FRI (given $\delta$) is simply top $N$ frames with the largest norm of gradients or frame-wise vulnerability. So, a single-step frame selection algorithm can be efficiently implemented by sorting.

C. FRAME SELECTION

As aforementioned, the proposed surrogate loss functions (e.g., MoG and FRI) yield single-step frame selection algorithms. Those surrogate losses are derived based on linearity, e.g., Eq. (3) and (6). However, if the number of frames to attack increases, the surrogate loss becomes inaccurate due to the non-linearity of the original loss function $L$, i.e., the discrepancy between the original loss and the surrogate loss gets larger so the frame selection becomes suboptimal. To address this problem, we propose iterative frame selection algorithms. Since $\delta^t$ is defined as the sum of $\delta^{(i)}$, the iterative frame selection estimates only one frame perturbation per step.

1) GREEDY SEARCH

We propose a Greedy Search algorithm for frame selection, which iteratively selects frames while updating input video $x$. The greedy search algorithm in Alg. 2 selects one frame at each iteration based on $J$. Here, $J$ can be either $J_{\text{MoG}}$ or $J_{\text{FRI}}$. Then, the selected frame $i$ is perturbed by $\delta^{(i)}$, i.e., $x_p = x_p + \delta^{(i)}$ as line 8 in Alg. 2. The updated video is utilized at the next iteration to find the subsequent vulnerable frame. Although the greedy algorithm is computationally more expensive than single-step frame selection, it generally provides better frame selection. The experimental results for the greedy algorithm are available in Section IV-D.

2) SECTION SEARCH

In the experiment of the Greedy Search, we have observed that the selected frames tend to disperse compared to the non-iterative method. More details will be discussed in Section IV-D2 and Section IV-E.

Motivated by the observation, we propose a more efficient iterative method called Section Search, which selects each frame from equally divided sections. The section search algorithm divides input video into sections $\{S_n\}_{n \in \mathbb{N}}$ with an equal length $r = \frac{\tau}{N}$, i.e., $S_n \leftarrow [(n - 1)r + j]_{j=1}^{r}$ as line 4 in Alg. 3. Then, for each section $S_n$, only one frame is selected based on $J$. Since the section search does not update the input, it is computationally more efficient than the greedy search algorithm. Also, they generally provide better frame selection than a single-step frame selection. Figure 2 illustrates the idea of the section search algorithm.
In this section, we validate the effectiveness of our attack methods on three video classification datasets: UCF101, HMDB51, Kinetics400. We first briefly introduce the datasets and provide implementation details. Also, we present the experimental results of single-step attacks, greedy search, and section search. Lastly, we discuss the motivating observation that the base networks in video classification have the architecture-specific vulnerability and in which circumstance the proposed methods are recommended.

### A. DATASET

UCF101 dataset [29] is a widely used dataset in video action-recognition. The dataset consists of 13,320 video clips from 101 human action categories with a total length of 27 hours, average duration of 7.2 seconds. All the videos have been collected from YouTube and have a fixed frame rate of 25 FPS with the resolution of 320 × 240. Specifically, we split UCF101 according to the official split-1, which splits the dataset into 8K training set and 3K testing set.

HMDB51 dataset [30] is a large collection of realistic videos from various sources, including movies and web videos. The dataset is composed of 6,849 video clips from 51 action categories (such as “jump”, “kiss” and “laugh”), with each category containing at least 101 clips. The original evaluation scheme uses three different training/testing splits. In each split, each action class has 70 clips for training and 30 clips for testing. We use split-1 for our experiment.

Kinetics400 dataset [31] is one version of the kinetics dataset, a collection of large-scale, high-quality Videos’ URLs that include about 650,000 video clips with 400/600/700 human action classes. The videos include human-object interactions such as playing instruments, as well as human-human interactions such as handshaking and hugging. Each clip is annotated with a single action class and lasts around 10 seconds. We split Kinetics400 according to the official split in [31]; 250-1000 videos per class for training, 50 videos per class for validation and 100 videos per class for testing.

### D. PERTURBATION GENERATION

We use the FGSM or PGD attack to generate adversarial perturbations. Unlike the previous adversarial attacks on video [15], [17], we applied FGSM and PGD attacks to only the selected frames \( I \) from the frame selection stage. The FGSM attack in our framework can be written as follows:

\[
A_{\text{FGSM}}(x, y, \theta, I) = \varepsilon \cdot \text{sign}(\nabla_x L(x, y; \theta)) \odot M^1,
\]

where \( \odot \) is Hadamard product (or element-wise product) and \( M^1 \in \mathbb{R}^{T \times C \times H \times W} \) is a mask with ones on selected frames and zeros on others, i.e., \( M[i, :, :, :] = 1 \), if \( i \in I \), o.w. \( M[i, :, :, :] = 0 \).

Similarly, the update step of the PGD attack in our framework for iteration \( k \in \{1, \ldots, N\} \) is defined as below:

\[
\delta_k(x, I) = \Pi_{\Delta} \left( \alpha \cdot \text{sign}(\nabla_x L(x + \delta_{k-1}, y; \theta) \odot M^1) \right),
\]

where \( \Pi_{\Delta} \) denotes the projection to the set of small perturbations \( \Delta \). Starting from \( \delta_0 = 0 \), the perturbation by PGD attack is obtained after \( N \) updates, i.e., \( A_{\text{PGD}}(x, y, \theta, I) = \delta_N(x, I) \).

### IV. EXPERIMENTS

In this section, we validate the effectiveness of our attack methods on three video classification datasets: UCF101, HMDB51, Kinetics400. We compare the surrogate loss functions, MoG and FRI, introduced in Section III-B. Table 2 shows the performance of various video classification models against the attacks based on MoG or FRI on UCF101. For perturbation generation, FGSM was used. Overall, the FRI chooses more vulnerable frame sets than the MoG achieving lower classification accuracy. FRI in (7) is the sum of loss changes induced by

![FIGURE 2. Visualization of Section Search Algorithm. For each section, only one frame is chosen based on the surrogate loss \( J \). This visualization is an example of \( J_8 = 8 \).](image)
TABLE 2. Classification accuracy on UCF 101. Combining with FGSM attacks, we compare the effectiveness of single-step frame selection methods: MoG and FRI with $N = 8$. ‘Random’ and ‘Fixed’ on the first column denotes randomly selected frames and pre-defined fixed frames (evenly-spaced) respectively.

|       | I3D | R3D | R(2+1)D | SlowFast | IRCSNv2 |
|-------|-----|-----|---------|----------|---------|
| OriginAcc | 89.2 | 92.3 | 94.6 | 90.2 | 96.2 |
| Random   | 59.5 | 38.4 | 57.2 | 63.9 | 76.3 |
| Fixed    | 75.6 | 32.1 | 51.1 | 39.9 | 74.8 |
| MoG      | 28.7 | 35.3 | 56.4 | 40.3 | 77.3 |
| FRI      | 28.8 | 31.8 | 51.8 | 39.9 | 74.8 |

FIGURE 3. The attack success rate of MoG attack on UCF101 with varying $N$. Since the degradation rate after $N = 8$ tends to be decreased on the most threat models, we fix $N$ as 8 in other experiments.

TABLE 3. Classification accuracy on UCF 101. We compare a single-step frame search and greedy search (denoted with suffix ‘-G’) for frame selection. The adversarial attacks are generated by FGSM after the frame selection. Overall, greedy search-based frame selection achieves a 2.37% higher success rate on average than single-step frame selection.

|       | I3D | R3D | R(2+1)D | SlowFast | IRCSNv2 |
|-------|-----|-----|---------|----------|---------|
| MoG-G  | 28.7 | 35.3 | 56.4 | 40.3 | 77.3 |
| MoG-G  | 28.6 | 32.4 | 49.4 | 39.6 | 73.2 |
| FRI-G  | 28.8 | 31.8 | 51.8 | 39.9 | 74.8 |
| FRI-G  | 28.5 | 28.5 | 46.9 | 39.5 | 72.8 |

D. ITERATIVE FRAME SELECTION

We evaluate the effectiveness of iterative frame selection methods: greedy search and section search. Also, we provide the experimental results to show model-specific vulnerability and why the section search can achieve comparable adversarial attacks with much less computational cost than greedy search.

1) GREEDY SEARCH

We compare greedy search-based frame selection with single-step frame selection methods. With the same surrogate objective functions as MoG and FRI in single-step frame selection, we applied the greedy search that chooses one frame at a time and update the surrogate objective functions. The greedy search algorithms with MoG and FRI are denoted by MoG-G and FRI-G, respectively. Table 3 shows classification accuracy on UCF101 against adversarial examples. Adversarial attacks (FGSM) with the greedy search-based frame selection method achieve 2.37% higher attack success rates than ones with single-step frame selection methods. For R3D, R(2+1)D, and IRCSNv2, the improvement by greedy search is significant for both surrogate objective functions. In particular, the largest improvement by greedy search is 7% with MoG-G and R(2+1)D.

2) MODEL-SPECIFIC VULNERABILITY

We observe that the performance gain by ‘Greedy Search’ varies depending on base networks as shown in Table 3. Interestingly, Greedy Search overall improved the power of adversarial attacks but the improvement of I3D and SlowFast by Greedy Search is relatively small. We investigated the different behaviors and found that models have different vulnerabilities at the frame level. For some architectures, single-step methods are sufficient to find a set of vulnerable frames. To analyze this hypothesis, we visualize the $L_1$-norm of each frame’s gradient for baseline models (Fig. 4). In the case of I3D and SlowFast, frames can be categorized into two groups: highly vulnerable frames and others. And the vulnerable frame group has significantly larger gradients than non-vulnerable frames. For instance, in Fig. 4, I3D and SlowFast have 8 highly vulnerable frames and the norms of their gradients are larger than 0.4 and 0.6 respectively. We conjecture that the significant difference makes the frame selection problem easy and leads to the small objective error.
performance gap between single-step methods and greedy search. Another interesting observation is that the vulnerable frames are evenly spaced. For example, in the SlowFast case, every fourth frame is a highly vulnerable frame (e.g., \{1, 5, 9, \cdots, 29\}). Besides, in Fig. 5 the histogram shows that both FRI and MoG with FGSM mostly chooses every fourth frame \{2, 7, 11, 15, \cdots, 31\}.

3) SECTION SEARCH
Motivated by the observation, we proposed Section Search in Section III-C2. Our experiments in Table 4 demonstrate that Section Search (denoted by MoG-S and FRI-S) also significantly improves the quality of frame selection resulting in stronger adversarial attacks than ones with single-step frame selection (denoted by MoG and FRI). Section Search improves success rate by 3.63% on average in the UCF101 dataset. The largest improvement is 10.2% with R(2+1)D, which is even larger than the improvement by Greedy Search.

E. MAIN RESULTS
For more comprehensive comparison, we provide experimental results in UCF101, HMDB51, and Kinetics400 in Table 5, Table 6, and Table 7, respectively. Before discussing the performance, we define a metric called Frame Interval Variance (FI-VAR) to summarize the patterns of selected frames. Let I denote the set of selected indices of frames, and \(i\) be the \(i\)-th smallest element in \(I\). Given the indices of selected frames \(\{I_i\}_{1 \leq i \leq N-1}\), Frame Interval Variance (FI-VAR) is defined as the variance of intervals between the indices, which is given as:

\[
\text{FI-VAR} = \frac{1}{N} \sum_{i=1}^{N} (d_i - \mu)^2,
\]

where \(d_i = I_{i+1} - I_i\) where \(i = 1, \cdots, N - 1\),

\[
d_N = I_1 + T - I_N,
\]

\[
\mu = \frac{1}{N} \sum_{i=1}^{N} d_i.
\]

We added \(d_N\) to construct a cyclic interval. FI-VAR quantifies how evenly the selected frames are spread out. When the frames are evenly spread, e.g., \{1, 5, \cdots, 29\} as discussed with SlowFast in Section IV-D2, the FI-VAR \(\approx 0\). If selected frames are congregated, then thanks to \(d_N\), the FI-VAR increases. Also, randomly scattered indices may have a large FI-VAR.

FIGURE 6. Attack Success Rate with a particular FI-VAR on UCF101. We randomly selected frames with a particular FI-VAR and attack the frames to the R3D model. The horizontal line shows the baseline success rate without particular FI-VAR. In the process of changing the FI-VAR, it can be seen that the smaller the FI-VAR, the stronger the randomly selected attack.
TABLE 5. Classification accuracy on UCF101 dataset against baseline and various Search-and-Attack methods. ‘-G’ and ‘-S’ denote the greedy search and section search respectively. For PGD attack, we use $\alpha = 0.5$, $\epsilon = 2$, $n = 4$. * and † mean the best performance and the second best performance respectively. We select 8 frames for each video. In FGSM, the FRI-S shows the best performance or the gap with the best performance is very small. Unlike in FGSM, in PGD, the breakthrough of iterative methods is significant. Interestingly, the FI-Var of the FRI-G method decreases much more in PGD than FGSM. In the process of updating the frame for each iteration, it can be seen that if the attack intensity is strong, the frame is naturally selected to increase the distance between frames.

| Perturbation | Frame Selection | N | 1D3 | R3D | (2+1)D | SlowFast | IRCNNv2 |
|--------------|-----------------|---|-----|-----|--------|---------|--------|
| -            | None            | - | 89.2| 92.3| 94.6   | 90.2    | 96.2   |
| FGS M        | All             | 32| 24.8| 4.2 | 20.3   | 32.9    | 58.1   |
| PGD          | All             | 32| 6.8 | 0   | 0      | 10.0    | 23.0   |

PGSM

Random       | 8   | 59.5| 9.3 | 38.4| 9.29| 57.2| 9.3 | 63.9| 9.3 | 79.3| 9.3 |
MoG          | 8   | 28.7| 0.2 | 35.3| 21.6| 56.4| 26.8| 40.3| 1.2 | 77.3| 26.1|
MoG-G        | 8   | 28.6*| 0.1| 32.4| 17.6| 49.4| 14.9| 39.6| 0.4 | 73.2| 15.8|
MoG-S        | 8   | 28.7 | 0   | 28.0| 2.3 | 46.2| 4.0 | 39.5*| 0   | 73.6| 2.2 |
FRI          | 8   | 28.8 | 0.6| 31.8| 21.6| 51.8| 23.8| 39.9| 0.8 | 74.8| 29.7|
FRI-G        | 8   | 28.7 | 0.6| 28.5*| 16.7| 46.9†| 18.5| 39.5*| 0.4 | 72.8| 26.2|
FRI-S        | 8   | 28.7 | 0   | 27.1*| 2.6 | 44.5*| 3.9 | 39.5*| 0   | 73.0*| 2.3 |

FGSM

Random       | 8   | 52.2| 9.2 | 13.4| 9.2 | 18.4| 9.3 | 48.7| 9.4 | 62.8| 9.3 |
MoG          | 8   | 11.6*| 0.2| 9.0 | 21.6| 17.5| 26.8| 19.3| 1.2 | 59.6| 26.1|
MoG-G        | 8   | 11.7 | 0   | 8.0 | 20.7| 14.1| 26.7| 18.7*| 0.4 | 50.6*| 15.3|
MoG-S        | 8   | 11.7 | 0   | 8.0 | 2.3 | 13.2| 4.0 | 18.8| 0   | 53.8| 2.2 |
FRI          | 8   | 11.7 | 0   | 8.4 | 21.6| 17.1| 27.0| 19.0| 0.9 | 53.8| 27.3|
FRI-G        | 8   | 11.6*| 0.1| 6.5*| 12.5| 11.6*| 11.9| 18.7*| 0.1 | 48.7*| 16.1|
FRI-S        | 8   | 11.7 | 0   | 7.9*| 2.3 | 12.7†| 3.7 | 18.7*| 0   | 53.7| 2.1 |

TABLE 6. Classification accuracy on HMDB51 dataset against baseline and various Search-and-Attack methods. ‘-G’ and ‘-S’ denote the greedy search and section search respectively. For PGD attack, we use $\alpha = 0.5$, $\epsilon = 2$, $n = 4$. * and † mean the best performance and the second best performance respectively. We select 8 frames for each video. Like UCF101, the FRI-S generally shows the best performance.

| Perturbation | Frame Selection | N | 1D3 | R3D | (2+1)D | SlowFast | IRCNNv2 |
|--------------|-----------------|---|-----|-----|--------|---------|--------|
| -            | None            | - | 65.6| 68.0| 73.4   | 70.8    | 80.3   |
| FGS M        | All             | 32| 10.6| 0   | 6.8    | 16.0    | 29.8   |
| PGD          | All             | 32| 1.3 | 0   | 0      | 1.6     | 11.9   |

PGSM

Random       | 8   | 31.5| 9.3 | 13.6| 9.6 | 27.1| 9.4 | 35.4| 9.2 | 45.2| 9.3 |
MoG          | 8   | 28.5| 2.9 | 11.7| 26.7| 27.0| 31.5| 19.7| 6.2 | 45.9| 37.2|
MoG-G        | 8   | 12.3| 2.1 | 10.9| 22.3| 21.4| 16.8| 18.5| 2.4 | 42.1| 21.6|
MoG-S        | 8   | 12.3 | 0   | 9.2 | 2.5 | 19.5| 4.2 | 18.4*| 0   | 40.9| 2.6 |
FRI          | 8   | 12.5 | 2.7| 10.4| 24.8| 22.1| 25.9| 18.7| 4.1 | 41.9| 34.3|
FRI-G        | 8   | 12.2*| 2.3| 8.9†| 23.0| 18.9†| 22.9| 18.5| 2.4 | 39.8*| 28.8|
FRI-S        | 8   | 12.3 | 0   | 8.8*| 2.9 | 17.7*| 4.0 | 18.4*| 0   | 39.8*| 2.8 |

FGD

Random       | 8   | 23.8| 9.3 | 4.4 | 9.5 | 3.2 | 9.3 | 31.4| 9.3 | 31.0| 9.3 |
MoG          | 8   | 4.3 | 2.6 | 2.7 | 26.6| 4.5 | 31.5| 8.1 | 6.2 | 29.3| 37.2|
MoG-G        | 8   | 4.2 | 1.7 | 0.8 | 23.8| 3.8 | 23.3| 7.5†| 2.3 | 26.4*| 21.5|
MoG-S        | 8   | 4.2 | 0   | 0.5 | 2.5 | 1.4 | 4.2 | 7.5†| 0   | 27.4| 2.6 |
FRI          | 8   | 4.1 | 2.4 | 2.8 | 26.9| 4.3 | 32.6| 8.0 | 4.5 | 27.4| 34.9|
FRI-G        | 8   | 3.9*| 1.6| 0.4*| 15.2| 1.2*| 15.8| 7.6 | 0.1 | 25.8*| 17.7|
FRI-S        | 8   | 4.0†| 0   | 0.5†| 2.5| 1.3†| 3.9 | 7.7 | 0   | 27.3| 2.5 |
TABLE 7. Classification accuracy on Kinetics400 dataset against baseline and various Search-and-Attack methods. '-G' and '-S' denote the greedy search and section search respectively. For PGD attack, we use \( \epsilon = 0.5 \), \( \epsilon = 2 \), \( n = 4 \), and \( \epsilon = 3 \) faster.

| Perturbation | Frame Selection | \( N \) | 13D | R3D | R(2+1)D | SlowFast | IRCSNv2 |
|--------------|-----------------|-------|-----|-----|--------|---------|--------|
| -            | None            | -     | 62.3 | -   | 60.5   | -       | 65.7   | 66.3   | 73.8   |
| FGSM         | All             | 32    | 4.6  | 0.5 | 0      | 2.2     | 6.6    | 0      | 18.2   |
| PGD          | All             | 32    | 1.4  | 0   | 0      | 2.2     | 6.6    | 2.2    | 5.7    |
| FGSM         | Random          | 8     | 20.9 | 9.3 | 5.6    | 9.3     | 13.4   | 9.4    | 26.5   | 9.3    |
| FGSM         | MoG             | 8     | 5.4  | 0.6 | 3.3    | 20.0    | 12.8   | 25.7   | 8.2    | 14.6   |
| FGSM         | MoG-G           | 8     | 5.4  | 0.6 | 2.9    | 16.3    | 9.5    | 13.8   | 8.3    | 15.4   |
| FGSM         | MoG-S           | 8     | 5.4  | 0   | 2.6    | 2.4     | 8.7    | 3.6    | 8.1*   | 0      |
| FGSM         | FRI             | 8     | 5.4  | 3.4 | 2.9    | 22.3    | 10.6   | 25.8   | 8.3    | 4.5    |
| FGSM         | FRI-G           | 8     | 5.4  | 3.5 | 2.3*   | 18.3    | 8.5*   | 21.0   | 8.2    | 4.0    |
| FGSM         | FRI-S           | 8     | 5.4  | 0   | 2.3*   | 2.6     | 8.0*   | 3.8    | 8.1*   | 0      |
| PGD          | Random          | 8     | 14.7 | 9.4 | 0.9    | 0.3     | 1.6    | 9.4    | 14.7   | 0.3    |
| PGD          | MoG             | 8     | 2.0  | 0.6 | 0.5    | 20.0    | 1.1    | 25.8   | 2.8    | 1.5    |
| PGD          | MoG-G           | 8     | 1.9* | 0.4 | 0.4*   | 18.2    | 0.9    | 19.1   | 2.9*   | 1.1    |
| PGD          | MoG-S           | 8     | 1.9* | 0   | 0.5    | 2.4     | 0.8*   | 3.6    | 2.9*   | 1.2    |
| PGD          | FRI             | 8     | 2.5  | 2.2 | 0.5    | 19.8    | 1.2    | 25.8   | 3.9    | 3.3    |
| PGD          | FRI-G           | 8     | 2.0  | 1.8 | 0.5    | 12.9    | 0.8*   | 13.8   | 3.1    | 2.2    |
| PGD          | FRI-S           | 8     | 2.5  | 0   | 0.5    | 2.4     | 0.9    | 3.6    | 2.9*   | 0      |

in UCF101, HMDB51, and Kinetics 400. It is worth noting that Section Search (FRI-S) is \( \times 1.9 \sim \times 2.3 \) faster than Greedy Search (FRI-G). We conjecture that if the perturbation method is strong (e.g., PGD), then the loss may change drastically after the perturbation. So the iterative method, especially Greedy Search is more effective. On the other hand, when the perturbation generation is relatively weak (e.g., FGSM), then Section Search is more efficient and achieves a comparable attack success rate.

V. CONCLUSION
In this work, we propose a Search-and-Attack framework for temporally sparse adversarial perturbations on videos. The Search-and-Attack framework has two stages: frame selection (Search) and perturbation generation (Attack). To identify the most vulnerable frames in the Search stage, we explore Single-step Search methods with surrogate objective functions: MoG and FRI. In addition, for more accurate frame selection with more computational power, we propose Greedy Search and Section Search. Our extensive experiments on three benchmark datasets, e.g., UCF101, HMDB51, and Kinetics400, show that Greedy Search with FRI and PGD (FRI-G + PGD) achieves the best attack success rate on average. We observe that neural networks in video classification have model-specific vulnerability in the time domain and evenly-spaced selected frames are often effective. Motivated by the observation, we propose Section Search and interestingly, with FGSM, the Section Search with FRI (FRI-S + FGSM) shows the best attack success rate although it is computationally more efficient than Greedy Search. In our experiment, we assume that the number of frames to attack \( N \) is given, not dynamically estimates \( N \). However, since a strategy on how to determine such parameters in an unsupervised manner is necessary, clustering techniques for complex networks [33]–[35] could be used to determine those parameters.

The future directions of this work include studying more imperceptible and efficient adversarial perturbation for videos and robust neural network architectures in video classification. Specifically, since the proposed attack targets the neural networks on videos, it can be used to verify the robustness of holistic video understanding systems using neural networks. For example, it can verify robustness or help to build a more robust system for a video tagging system [36] or autonomous car [37] which uses video neural networks to make proper decisions.

ACKNOWLEDGMENT
(Hwan Heo, Dohwan Ko, and Jaewon Lee contributed equally to this work.)

REFERENCES
[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. NeurIPS, 2012, pp. 1097–1105.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, 2014, pp. 1–14.
[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[6] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.
[7] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in Proc. ECCV, Sep. 2015, pp. 346–361.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.
[9] L. Wang, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, “Towards fast computation of certified robustness for ReLU networks,” in Proc. ICML, 2018, pp. 5276–5285.
[10] H. Heo, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An efficient SMT solver for verifying deep neural networks,” in Proc. CAV, 2017, pp. 97–117.
[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in Proc. ICLR, 2013, pp. 1–10.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in Proc. ICLR, 2014, pp. 1–11.

[13] A. Madaule, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep-learning models resistant to adversarial attacks,” in Proc. ICLR, 2018, pp. 1–24.

[14] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2017, pp. 39–57.

[15] X. Wei, J. Zhu, and H. Su, “Sparse adversarial perturbations for videos,” 2018, arXiv:1803.02536.

[16] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. S. Chowdhury, and A. Swami, “Stealthy adversarial perturbations against real-time video classification systems,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2019, pp. 1–15.

[17] N. Inkawhich, M. Inkawhich, Y. Chen, and H. Li, “Adversarial attacks for optical flow-based action recognition classifiers,” 2018, arXiv:1811.11875.

[18] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6299–6308.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4489–4497.

[20] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6450–6459.

[21] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast networks for video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6202–6211.

[22] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, “Long-term recurrent convolutional networks for visual recognition and description,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 2625–2634.

[23] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&amp;P), Mar. 2016, pp. 372–387.

[24] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, “SparseFool: A few pixels make a big difference,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9087–9096.

[25] F. Croce and M. Hein, “Sparse and imperceivable adversarial attacks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4724–4732.

[26] B. Luo, Y. Liu, L. Wei, and Q. Xu, “Towards imperceptible and robust adversarial example attacks against neural networks,” in Proc. AAAI, 2018, pp. 1652–1659.

[27] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” 2016, arXiv:1611.01236.

[28] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luweldke, and A. Mahajan, “Mixed-integer nonlinear optimization,” Acta Numer., vol. 22, pp. 1–131, May 2013.

[29] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, arXiv:1212.0402.

[30] H. Kuehne, H. Duang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 2556–2563.

[31] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The kinetics human action video dataset,” 2017, arXiv:1705.06950.

[32] D. Tran, H. Wang, M. Feiszli, and L. Torresani, “Video classification with channel-separated convolutional networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5552–5561.

[33] L. Hu, J. Zhang, X. Pan, H. Yan, and Z.-H. You, “HiSCF: Leveraging higher-order structures for clustering analysis in biological networks,” Bioinformatics, vol. 37, no. 4, pp. 542–550, May 2021.

[34] L. Hu, K. C. C. Chan, X. Yuan, and S. Xiong, “A variational Bayesian framework for cluster analysis in a complex network,” IEEE Trans. Knowl. Data Eng., vol. 32, no. 11, pp. 2115–2128, Nov. 2020.

[35] L. Hu and K. C. Chan, “Fuzzy clustering in a complex network based on content relevance and link structures,” IEEE Trans. Fuzzy Syst., vol. 24, no. 2, pp. 456–470, Apr. 2016.

[36] S. Ilyas and H. Ur Rehman, “A deep learning based approach for precise video tagging,” in Proc. 15th Int. Conf. Emerg. Technol. (ICET), Dec. 2019, pp. 1–6.

[37] Y. Xu, X. Yang, L. Gong, H.-C. Lin, T.-Y. Wu, Y. Li, and N. Vasconcelos, “ Explainable object-induced action decision for autonomous vehicles,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9523–9532.

**DOHWAN KO** received the B.S. degree from the Department of Computer Science and Engineering, Korea University, in 2021, where he is currently pursuing the M.S. degree, under the supervision of Prof. Hyunwoo J. Kim. His research interests include safe AI, video understandings, and visual interaction detection.

**JAEOON LEE** received the B.S. degree from the Department of Mathematics, Korea University, in 2021, where he is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, under the supervision of Prof. Hyunwoo J. Kim. His research interests include safe AI and point cloud processing.

**YOUNGJOON HONG** received the Ph.D. degree in mathematics from Indiana University, in 2015. From 2015 to 2018, he was a Research Assistant Professor at the University of Illinois, Chicago. Since 2018, he has been an Assistant Professor with the Department of Mathematics and Statistics, San Diego State University, before he joined Sungkyunkwan University. His research interests include applied analysis and computational mathematics. Much of his works are motivated by connections between practical applications, such as computational fluid dynamics, geophysics, electromagnetism, nonlinear waves, data science, and deep learning.

**HYUNWOO J. KIM** received the Ph.D. degree in computer science from the University of Wisconsin-Madison, in 2017, and the Ph.D. degree in statistics. He worked at Amazon Lab126, Sunnyvale, CA, USA. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Korea University. His research interests include geometric deep learning for graphs and manifolds, manifold statistics, machine learning, computer vision, and medical imaging.