





# Generative Perturbation Network for Universal Adversarial Attacks on Brain-Computer Interfaces

Jiyoung Jung , Member, IEEE, HeeJoon Moon , Student Member, IEEE, Geunhyeok Yu , Student Member, IEEE, and Hyoseok Hwang , Member, IEEE

**Abstract**—Deep neural networks (DNNs) have successfully classified EEG-based brain-computer interface (BCI) systems. However, recent studies have found that well-designed input samples, known as adversarial examples, can easily fool well-performed deep neural networks model with minor perturbations undetectable by a human. This paper proposes an efficient generative model named generative perturbation network (GPN), which can generate universal adversarial examples with the same architecture for non-targeted and targeted attacks. Furthermore, the proposed model can be efficiently extended to conditionally or simultaneously generate perturbations for various targets and victim models. Our experimental evaluation demonstrates that perturbations generated by the proposed model outperform previous approaches for crafting signal-agnostic perturbations. We demonstrate that the extended network for signal-specific methods also significantly reduces generation time while performing similarly. The transferability across classification networks of the proposed method is superior to the other methods, which shows our perturbations' high level of generality.

**Index Terms**—Adversarial attack, brain computer interfaces, EEG classification, universal adversarial perturbation.

## I. INTRODUCTION

**B**RAIN-COMPUTER interface (BCI) is a computer-based system that provides a direct communication pathway between the brain and an output device to carry out the desired action [1], [2]. The goal of human BCI systems is to translate

the activated brain signal into computer communication to operate external devices in a way that is consistent with human objectives [3]. The idea of controlling prosthetic arms with brain impulses was developed in the 1970s [4]. Since that time, BCIs have been advanced to explore users' conscious intention, as well as perception, awareness, and cognition, resulting a human-computer interface (HCI) that is enhanced by implicit information [5]. BCIs have been widely applied in various fields, including medicine [6], education [3], robotics [5], and augmented reality (AR) [7].

EEG signals have been widely used in many BCI studies because they are collected non-invasively with a high temporal resolution, using portable and inexpensive headset devices. Most EEG-based BCI studies choose to classify EEG signals for various research purposes, including emotion classification [8], motor imagery classification [9], seizure detection [10], and Alzheimer recognition [11]. Traditionally, EEG-based studies have extracted features from EEG signals and trained these features for classification using conventional machine learning methods, such as the support vector machine (SVM) and the k-nearest neighbors (KNN).

Deep neural networks (DNNs) have become popular due to their excellent generalization capacity. Classification methods that used DNNs significantly improved performance across many traditionally challenging domains. Recent surveys on the latest classification algorithms in EEG-based BCIs also reviewed that deep learning methods had exceptional results in classification accuracy [1]. Traditionally, these approaches were usually associated with particular hand-engineered features. However, DNNs promised to learn complicated features automatically from large amounts of data through end-to-end learning.

Despite remarkable successes in numerous applications, recent research has shown that deep neural networks are vulnerable to thoughtfully crafted input samples. These samples can easily deceive a well-performed deep learning network with subtle signal modification imperceptible to humans [12]. For example, Szegedy et al. [13] proposed a method that adds small perturbations on the original images for fooling a state-of-the-art classification model based on deep neural networks with high probability. These misclassified samples were named adversarial examples. Adversarial examples have emerged as a severe risk

Manuscript received 18 February 2023; revised 15 May 2023 and 11 July 2023; accepted 6 August 2023. Date of publication 9 August 2023; date of current version 7 November 2023. The work of Jiyoung Jung was supported by the 2022 Research Fund of the University of Seoul. The work of Hyoseok Hwang was supported by the National Research Foundation of Korea Grant funded by the Korea government (MSIT) under Grant NRF-2022R1C1C1008074. (Corresponding author: Hyoseok Hwang.)

Jiyoung Jung is with the Department of Artificial Intelligence, University of Seoul, Dongdaemun-gu 02504, South Korea (e-mail: jyjung@uos.ac.kr).

HeeJoon Moon, Geunhyeok Yu, and Hyoseok Hwang are with the Department of Software Convergence, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: wilko97@khu.ac.kr; geunhyeok@khu.ac.kr; hyoseok@khu.ac.kr).

Our code is available for download on <https://github.com/AIRLABkhu/Generative-Perturbation-Networks>.

Digital Object Identifier 10.1109/JBHI.2023.3303494

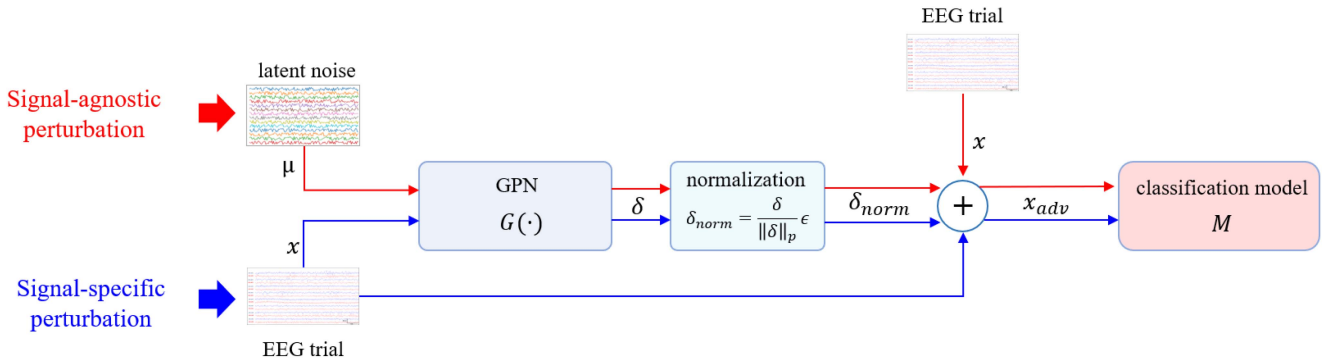


Fig. 1. Overview of the proposed method. The proposed model can be utilized for signal-agnostic and signal-specific perturbations. A fixed noise and training images are supplied to the same architecture to generate signal-agnostic perturbations (indicated by the red arrow) and signal-specific perturbations (indicated by the blue arrow), respectively.

of deep learning systems and cannot be ignored in medical applications based on artificial intelligence.

Numerous approaches have been suggested for generating adversarial examples [14]. Specifically, for each image within a dataset, signal-specific adversarial perturbations are created to target its unique signal properties [13], resulting in variations among samples. Conversely, there are signal-agnostic perturbations referred to as universal adversarial perturbations (UAPs) [15]. These UAPs possess the remarkable ability to deceive state-of-the-art recognition models with high probability, while remaining imperceptible to human observers.

Generative models play a crucial role in the field of adversarial attacks by enabling the creation of realistic perturbations or examples that can deceive machine learning models [16]. Generative models, such as generative adversarial networks (GANs) and variational autoencoders (VAEs), can be used to generate adversarial examples which are designed to be perceptually similar to legitimate inputs, but contain subtle perturbations that lead to incorrect model predictions.

The vulnerability of EEG-based BCIs using deep learning models has been investigated [17]. Adversarial attacks could cause the wheelchair or exoskeleton to malfunction when EEG-based BCIs are used to control them for the disabled. Adversarial attacks could result in misdiagnosis in clinical applications of BCIs for awareness evaluation/detection for patients with disorders of consciousness. Both white-box and black-box attacks against EEG-based BCI systems have been investigated [18]. Also, optimization-based methods to craft universal adversarial perturbation for EEG classification models have been introduced [19].

In this paper, we propose an efficient generative approach to craft adversarial perturbations to attack EEG classification models. The proposed generative perturbation network (GPN) model can generate signal-agnostic and signal-specific perturbations depending on the training methods. While an image is fed to the GPN model to generate the signal-specific perturbation distinct for each image, a fixed latent noise is fed to the same model to generate the universal perturbation as shown in 1.

We also introduce two extended versions of the proposed method: The conditional GPN (cGPN) generates signal-specific

perturbations for a specific model and attack type. In contrast, multiple GPN (mGPN) generates multiple sets of signal-specific perturbations for several models and attack types with a single set of parameters. The main contributions of this study are summarized below:

- We propose a generative model which can produce both signal-agnostic and signal-specific perturbations for EEG-based datasets. To the best of our knowledge, this is the first approach that applies perturbations by generative model to EEG-based signals.
- We also propose modified generative models that require training on a single dataset yet can generate perturbations for multiple cases. By employing the extended version of the proposed generative model, we can manipulate adversarial examples more efficiently.
- We evaluate the performance of attacks with universal adversarial perturbations in both non-targeted and targeted attacks on within- and cross-subject experiments where our approaches outperform previous methods.

## II. RELATED WORKS

*EEG classification using DNNs:* In recent studies, multiple approaches based on DNNs have been proposed for EEG classification. Lawhern et al. [20] proposed EEGNet, a compact convolutional neural network for EEG-based BCIs. It employs depthwise and separable convolutions to extract prominent EEG features across various BCI paradigms. Schirmermeister et al. [21] explored DeepConvNet and ShallowConvNet for end-to-end EEG decoding. They visualized the informative EEG features that ConvNets had learned. Kostas and Rudzicz [22] designed TIDNet using transfer learning to overcome the difficulty of generally applying DNN-BCI classifiers to multiple subjects. Lun et al. [23] proposed a deep convolutional neural networks (CNNs) structure that uses separate temporal and spatial filters to choose the raw EEG signals from the electrode pairs over the motor cortex region. Cho and Hwang [24] adopted three-dimensional CNNs for emotion recognition. They showed that the 3D reconstructions of the raw EEG signals could effectively be combined with 3D CNNs to represent features from spatiotemporal data.

*Adversarial attacks:* Adversarial attacks, specifically generating a perturbation for input data to fool neural networks, have been studied since [13]. This work introduced the term ‘adversarial examples’, showing the vulnerable properties of deep neural networks when small perturbations to the inputs are given. Goodfellow et al. [25] proposed a simple and fast method of generating adversarial examples named the fast gradient sign method (FGSM), an efficient one-step attack to craft adversarial perturbations. Moosavi et al. [26] introduced an iterative attack method Deepfool. The paper proposed an efficient method to compute perturbations relatively smaller than FGSM that could still fool deep networks. In a follow-up study of Deepfool, the authors showed the existence of very small but universal perturbation vectors, called UAPs [15]. Poursaeed et al. [27] presented deep neural networks which is trainable for transforming images to adversarial examples. It considerably improved crafting time than other iterative methods and achieved high fooling rates with small perturbation norms. Mopuri et al. [16] introduced a generative approach inspired by the GANs to model the distribution of adversarial perturbations for a given CNNs classifier. This method has the advantages of speed and diversity of perturbations without complex manifolds of adversarial perturbations.

*Adversarial attacks on BCI systems:* Feng et al. [28] proposed SAGA, a sparse adversarial EEG attack, to identify the weakness of EEG analytics. The authors designed an adaptive mask to represent diverse sparsity in adversarial attacks uniformly. Meng et al. [29] performed white-box target attacks for regression problems of EEG signal to create small perturbations to change the regression output, where we are fully aware of the regression model. Recent studies have also suggested black-box attack strategies for EEG classifiers. Liu et al. [19] proposed a total loss minimization (TLM) approach, whose goal is to generate UAPs for EEG-based BCIs. The method overcame the limitations of prior knowledge about EEG trials needed to compute perturbations, which can be obtained through each input EEG trial.

### III. PROPOSED METHOD

In this section, we define the problems that we would like to tackle in this manuscript and introduce the proposed method. A classifier  $M$  estimates a label  $M(x) \in \{1, \dots, C\}$  for input signal  $x$ , where  $C$  is the number of classes. An adversarial example with a perturbation  $\delta$  added to the original signal  $x$  can be used for two purposes, which are non-targeted attack and targeted attack. The non-targeted attack aims to make a model  $M$  classify  $M(x)$  and  $M(x + \delta)$  differently. In a targeted attack, the adversarial example pursues to be classified as the target label  $t$  as  $M(x + \delta) = t$ .

To achieve this, we investigate a generative model shown in Fig. 1. Using this model, perturbation  $\delta$  for an adversarial example  $x_{adv}$  can be generated by signal-agnostic and signal-specific methods. Both methods support non-targeted and targeted attacks by simply applying different loss functions. We also describe the flexibility of the proposed method, which allows generating perturbations in various ways, e.g., perturbation

generation for a specific model and attack type or generation of multiple perturbations at once.

#### A. Generative Perturbation Network

The proposed generative model is based on ResNet [30]; however, we modified it according to the dimensions of the EEG-based signal. The raw data of the EEG signal is represented as a two-dimensional matrix, of which rows are electrodes and columns are time sequences. In this study, we set the input size of the generative model to be  $1 \times h \times w$ , where  $h$  is the number of EEG electrodes, and  $w$  is the temporal length of the EEG segment.

The proposed generative model, named generative perturbation network (GPN), consists of the following architecture: two consecutive convolution blocks, residual blocks, a deconvolution block, and the final convolution block (see Fig. 2). In the first convolution block, a 2D convolution layer uses  $64 \ 7 \times 7$  kernel with a stride of 2 to decrease the spatial dimensions of input data. Then, we apply a 2D batch normalization layer followed by activation functions, a rectified linear unit (ReLU). The second convolution block has the same architecture as the first one but some differences. In the convolution layer of the second block,  $128 \ 3 \times 3$  kernels are used, and the stride is 1. The first and second convolution blocks are defined as:

$$\mathcal{B}_{c1}(x) = \mathcal{A}(\mathcal{N}(\mathcal{F}_{7 \times 7, 64}(x))), \quad (1)$$

$$\mathcal{B}_{c2}(x) = \mathcal{A}(\mathcal{N}(\mathcal{F}_{3 \times 3, 128}(x))), \quad (2)$$

where  $\mathcal{N}$ ,  $\mathcal{A}$  are batch normalization [31], and activation layers, respectively.  $\mathcal{F}_{m \times m, n}(\cdot)$  means 2D convolution layer with  $n$   $m \times m$  kernels.

The following two residual blocks are identical and consist of two 2D convolution layers followed by a 2D batch normalization layer and a ReLU activation function. Note that the second batch normalization layer’s output is added with the skip connection’s input before the second activation function. All convolution layers use  $128 \ 3 \times 3$  kernels with stride one, and no max-pooling is applied to maintain the dimensions. The first residual block can be represented by:

$$\mathcal{B}_{r1}(x) = \mathcal{A}(x + \mathcal{N}(\mathcal{F}_{3 \times 3, 128}(\mathcal{A}(\mathcal{N}(\mathcal{F}_{3 \times 3, 128}(x)))))). \quad (3)$$

The second residual block  $\mathcal{B}_{r2}(x)$  has the same structure as  $\mathcal{B}_{r1}(x)$ . The deconvolution block consists of a transpose convolution layer followed by a 2D batch normalization layer and a ReLU activation function as:

$$\mathcal{B}_{dc}(x) = \mathcal{A}(\mathcal{N}(\mathcal{T}_{3 \times 3, 64}(x))), \quad (4)$$

where  $\mathcal{T}_{m \times m, n}(\cdot)$  is transpose convolution layer with  $n$   $m \times m$  kernels.

In the final convolution layer, we employ a  $1 \times 1$  kernel instead of average pooling to reduce the depth to 1. Then, we apply the tanh function to ensure that the output ranges from -1 to 1. The block is defined as:

$$\mathcal{B}_{c3}(x) = \tanh(\mathcal{F}_{1 \times 1, 1}(x)). \quad (5)$$

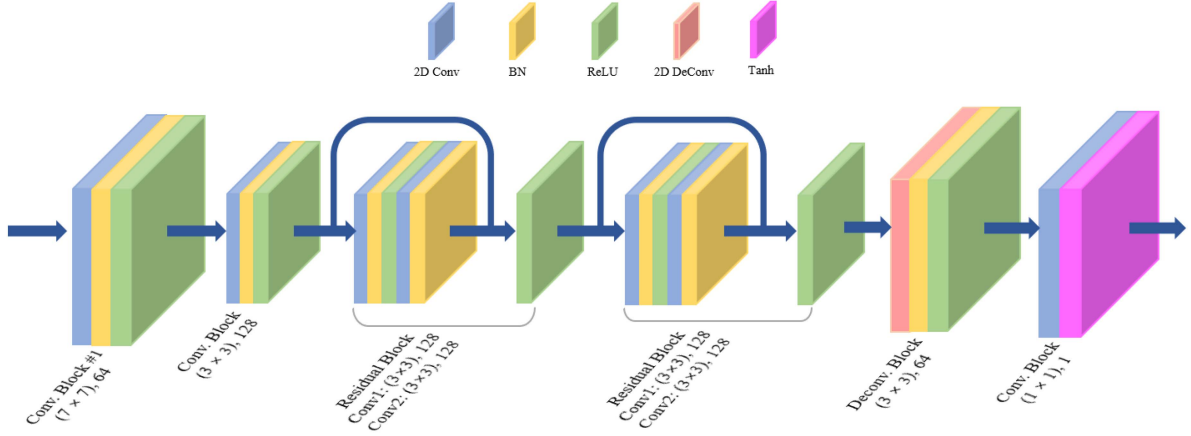


Fig. 2. Architecture for Generative Perturbation Network. The GPN uses ResNet as backbone networks which is modified according to the EEG-based dataset. BN means batch normalization.

To summarize, the proposed generative perturbation network  $G_v$  is given by:

$$G_v(x) = \mathcal{B}_{c3}(\mathcal{B}_{dc}(\mathcal{B}_{r2}(\mathcal{B}_{r1}(\mathcal{B}_{c2}(\mathcal{B}_{c1}(x)))))). \quad (6)$$

### B. Signal-Agnostic Perturbations Using GPN

Adversarial examples are made by adding perturbations to the original signal. To succeed in an attack by adding the same perturbation regardless of the signal, signal-agnostic universal perturbation has been used. In this section, we introduce the method to generate universal perturbations using GPN. Using the proposed model, the signal-agnostic perturbations are generated by

$$\delta = G_v(\mu), \quad (7)$$

where  $\mu$  is fixed random noise, of which size is same with the input EEG segments. Here, we denote the GPN as  $G_v$ , which means vanilla GPN (vGPN) to distinguish it from its extended versions. The initially generated perturbation  $\delta$  is then normalized to the range  $[-\epsilon, \epsilon]$  as

$$\delta_{norm} = \frac{\delta}{\|\delta\|_p} \epsilon, \quad (8)$$

where  $\|\delta\|_p$  is  $L_p$  norm of  $\delta$ . In our case, we set  $p = \text{inf}$ .  $\epsilon$  is the maximum permissible magnitude of the perturbation. The adversarial examples  $x_{adv}$  can be derived by adding the normalized perturbation  $\delta_{norm}$  to the original signal  $x$ ,

$$x_{adv} = x + \delta_{norm}. \quad (9)$$

Then, the adversarial examples are feed-forwarded to the classification model  $M$ . The scores or probabilities of classes from the model  $M$  is represented as  $k(x_{adv})$ . Using these scores, parameters of generator  $G_v$  could be trained iteratively.

Here, we apply different loss functions to the attack methods of non-targeted and targeted attacks. The goal of the non-targeted attack is to make the model  $M$  classify  $x_{adv}$  different from  $x$ . Therefore, we define the loss function  $\mathcal{L}_{nt}$  for non-targeted attacks as:

$$\mathcal{L}_{nt} = \log(\mathcal{H}(1 - k(x_{adv}), M(x))), \quad (10)$$

where  $\mathcal{H}(\cdot, \cdot)$  is the cross-entropy function and  $M(x)$  is label of  $x$  by the classification model  $M$ . The more  $k(x_{adv})$  differs from the original classification result  $M(x)$ , the smaller  $\mathcal{L}_{nt}$  becomes. Analogously, we define the loss function  $\mathcal{L}_t$  for targeted attacks as:

$$\mathcal{L}_t = \mathcal{H}(k(x_{adv}), C_{target}), \quad (11)$$

where  $C_{target} \in \{1, \dots, C\}$  is the target class to fool the model.  $\mathcal{L}_t$  gets smaller as  $k(x_{adv})$  becomes similar to the target label  $C_{target}$ . Note that in crafting signal-agnostic perturbations, the generated UAP is only valid for a certain set of a dataset, a classification model, and a target class (including non-targeted attack), as other previous UAP methods [19], [26].

### C. Signal-Specific Perturbations Using GPN

We can utilize the same model of crafting signal-agnostic perturbations to craft signal-specific perturbations. The only difference for generating signal-specific perturbations is an EEG-segment  $x$  is fed to the GPN instead of fixed random noise  $\mu$  as:

$$\delta = G_v(x). \quad (12)$$

The generated perturbation then follows the same procedures from (8) to (11) for normalization and calculation loss functions for a non-targeted and targeted attack. While generating one universal perturbation after learning in the signal-agnostic method, we got the weights of the parameters of  $G_v$  for each dataset, classification model, and target class. Each time in a signal-specific attack scenario, a perturbation had to be obtained by feeding input signals into the generative model, which can be cumbersome or inefficient. However, generating signal-specific perturbation using the generative model can be extended in various ways, e.g., generating perturbation with conditions or all perturbations at once. We described expanded methods of GPN in the following subsections.

### D. Extensions of Generative Perturbation Networks

Instead of optimizing perturbation directly, generating perturbation using deep neural networks has several advantages.

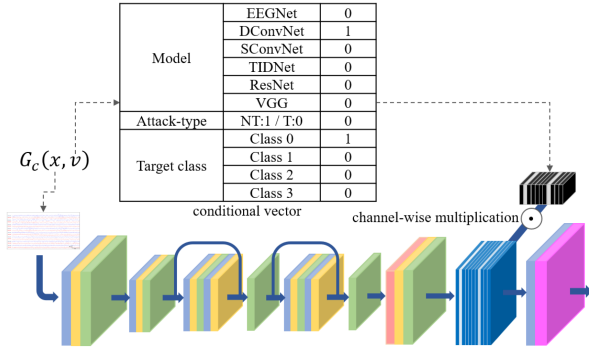


Fig. 3. Architecture for conditional generative perturbation network (cGPN).  $B_{c3}(x)$  performs an elementwise multiplication with  $v$ , the conditional vector.

One of them is the flexibility of structural modification. In this section, we propose the extended frameworks of GPN, which only need to be trained for each dataset. The extended generative models that generate signal-specific perturbations follow two strategies: generate perturbation conditionally, or generate multiple perturbations in all cases.

**Conditional Generative Perturbation Networks:** Fig. 3 shows the modified architecture of the vanilla generative model, which we named conditional generative perturbation network (cGPN). One distinctive aspect of cGPN is its utilization of a condition vector  $v$ , in addition to the input signal, to determine the generated perturbation as:

$$\delta_{GAN} = G_c(x, v). \quad (13)$$

This condition vector allows for the selection of the classification model, attack type, and target classes. The cGPN method can generate perturbations for various classification models, attack types, and target classes within a single dataset. Consequently, the length of the conditional vector is determined based on the number of classification models and target classes involved. In this model, the third convolution blocks consist of three layers - two convolution layers and an activation function:

$$B_{c3}(x) = \tanh(\mathcal{F}_{1 \times 1, 1}(\mathcal{F}_{7 \times 7, n}(x) \odot v)), \quad (14)$$

where  $v$  is the conditional vector and  $n$  is the length of the vector. The operator  $\odot$  means depth-wise multiplication. An EEG segment was iteratively trained for all condition vectors while training the model for each dataset. Different loss functions were applied depending on the attack type, non-targeted or targeted attack.

**Multiple Generative Perturbation Networks:** The second extension is the multiple generative perturbation network (mGPN). The mGPN is developed to simultaneously generate all perturbations for an input EEG segment at once. This involved modifying the original vanilla GPN model, as depicted in Fig. 4. Specifically, the third convolution block was modified as:

$$B_{c3}(x) = \tanh(\mathcal{F}_{1 \times 1, z}(x)), \quad (15)$$

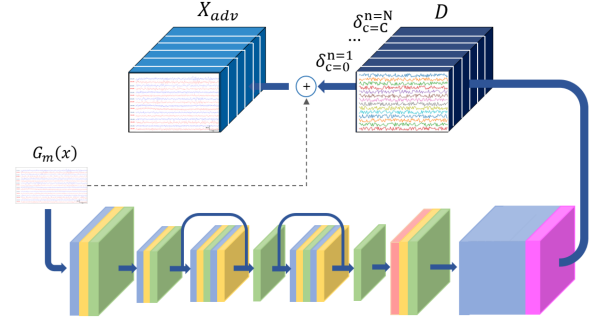


Fig. 4. Architecture for multiple generative perturbation network (mGPN). The number of depths in  $B_{c3}(x)$ ,  $z$ , is equal to the number of all perturbations to generate.

where  $z$  is the number of perturbation for all cases. The output of mGPN is

$$D = G_m(x), \quad (16)$$

where,  $D$  is the output tensor which consists of  $z$  perturbations.  $D$  is represented as a set of perturbation  $\delta_c^n$  subjected to  $n \in \{1, \dots, N\}$ ,  $c \in \{0, \dots, C\}$ , where  $n$  and  $c$  represent model index and target classes respectively. Note, we set  $\delta_0^n$  to the perturbation of non-targeted attack and  $\delta_1^n$  to  $\delta_C^n$  are perturbations of targeted attack of the classification model  $M_n$ .

We employed a combined loss function to train this model because  $D$  includes perturbations for both non-targeted and targeted attacks. The combined loss function can be defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (\alpha \mathcal{L}_{nt}^n + \beta \mathcal{L}_t^n), \quad (17)$$

where  $N$  is the number of classification model,  $\mathcal{L}_{nt}^n$  is the loss function for non-targeted attack given as:

$$\mathcal{L}_{nt}^n = \log(\mathcal{H}(1 - k(x + \delta_0^n), M_n(x))), \quad (18)$$

and  $\mathcal{L}_t^n$  is the loss function for targeted attack as:

$$\mathcal{L}_t^n = \frac{1}{C} \sum_{c=1}^C \mathcal{H}(k(x + \delta_c^n), C_{target}). \quad (19)$$

In 17,  $\alpha$  and  $\beta$  are hyper parameters that control the weight of two loss functions. In our study, we set  $\alpha$  and  $\beta$  according to the ratio of non-targeted and targeted attack cases. This means that for a dataset with  $C$  labels,  $\alpha$  and  $\beta$  are  $\frac{1}{(C+1)}$  and  $\frac{C}{(C+1)}$ , respectively.

#### IV. DATASET AND VICTIM MODELS

In this section, we briefly introduce the dataset and classification models which were used as victim models in our experiments.

##### A. Dataset for EEG Classification

In our study, we used four public BCI datasets performing different tasks to verify the generality. We pre-processed datasets normalized by setting the range of all data from 0 to 1. We divided

pre-processed EEG data into multiple segments, whose shape is  $1 \times \text{channels} \times \text{length}$ . The length of the segment depends on the dataset, and we set it to be between 1 and 4 s. The datasets used in the experiments are as follows:

*Amigos*: Amigos [32] is a multi-modal dataset for research on affect, personality traits, and mood by means of neurophysiological signals, which aims to understand the affective responses with respect to social context. The EEG signals were recorded with 14 channels sampled at 128 Hz using an EMOTIV Epc sensor. Forty participants involved in the experiment watched emotional short or long videos and rated their levels of affective levels by themselves. We only used the data for short videos, corresponding to 16 trials per subject. A band-pass filter between 4.0 Hz and 45.0 Hz was applied. Then we splitted the dataset into segments per second and labeled them as four classes according to their affective level of valence and arousal.

*DEAP*: DEAP [33] is a multi-modal dataset for the analysis of human affective states. This dataset consists of 32 participants watching 40 one-minute music videos, and their EEG signals were recorded with 32 channels sampled at 128 Hz. Participants rated their affective levels of valence, arousal, control, and like/dislike. We categorized the dataset into four classes using valence and arousal estimations in self-assessments and used 40 trials of one-minute-long EEG signals per subject. Each trial was also split into segments, as done in Amigos.

*NER2015*: This dataset was firstly introduced in the BCI challenge, IEEE Neural Engineering Conference [34], whose goal is to detect errors during the well-known BCI paradigm P300-Speller task, given the subject’s brain waves of visual stimuli. The EEG signals were recorded with 56 channels. The dataset was downsampled to 200 Hz, band-pass filter between 1 Hz and 40 Hz was applied. Among a dataset of 26 participants, we used 16 subjects for the training set and ten subjects for the test set, and there are 340 segments per subject. The dataset consists of 2 classes that we considered label 0 as a bad-feedback class and label 1 as a good-feedback class.

*Physionet*: This dataset was from movement and motor imagery (MMI) database [35], [36]. The EEG signals were recorded with 64 channels sampled at 160 Hz using the BCI2000 system. One hundred nine participants were involved in 14 experimental trials, consisting of 2 baseline and repetitions of 4 tasks. We only used eight trials of baseline and MI-tasks of task 2 and task 4 per participant, which consisted of imagining opening and closing the left or right fist, both feet. We split 64-channel EEG trials into one-second-long segments, then labeled each segment with four classes.

## B. Classification Model

We conducted our experiments with four CNN models, which are designed for EEG classification, and some well-known models frequently used to solve classification problems in computer vision. The classification methods that dedicated to EEG-based datasets are EEGNet [20], DeepConvNet, ShallowConvNet [21], and TIDNet [22]. The general classification models we used are ResNet [30] and VGG [37]. In order to convert 2D time series of EEG data into spatial pattern maps, we reshaped

EEG data into  $\text{channel} \times \text{segment length}$ . Consequently, we could handle EEG data similar to images and adapt computer-vision frameworks.

## V. EXPERIMENTAL RESULTS

In this section, we present the overall performance of our proposed models and the comparison results. To validate the performance for signal-agnostic (SA) perturbations, we evaluated the proposed method and compared the results to the existing methods [19], [26]. Experiments with signal-specific (SS) perturbations were also conducted using vanilla GPN and its extended approaches such as cGPN and mGPN. The proposed methods’ transferability across classification networks was measured and compared with previous works. We also evaluated the performance of GPNs with different sizes of epochs and number of channels to validate the effectiveness of our methods.

### A. Experimental Environment

All experiments used a 5-fold cross-validation approach of within- and cross-subject experiments to decrease the possibility of biased testing sets and provide robustness to the results. In the case of the within-subject experiment, we divided the total EEG segments of subjects in each dataset into five folds that do not overlap each other. Four folds were used for training, and the other one was used for evaluation, and in this way, the average performance was obtained by averaging the five results. We also divided all subjects into five folds to separate users in training and test data for the cross-subject experiment. In our experiments, we set the maximum permitted perturbation size  $\epsilon$  in infinity-norm to 0.0392 for all experiments, which value corresponds to a size of 10 in an image with values from 0 to 255 [27]. We also compared perturbation by adding Gaussian noise that has the same maximum amplitude  $\epsilon$  for non-targeted attacks.

For classification models, the training was conducted 200 epochs with Adam optimizer, and the initial learning rate was  $1e-3$ , decreased by half for every 50 epochs with a minibatch size of 64 EEG segments except 16 sizes of Physionet dataset and the 5-fold protocol. Sometimes, changing the momentum of the optimizer and Weight-Decay methods were used to prevent over-fitting problems. We experimented with all the methods of crafting UAPs in the same condition. Adam Optimizer, with a learning rate of  $1e-4$  and 20 epochs, was used for training GPN. Note that training and test data were completely disjoint and fed to classification models and GPN identically for fair evaluations. We experimentally found that the performance variation with the random seed was not statistically different. However, throughout our experiments with signal-agnostic GPNs, we used the fixed random seed for the initial noise due to the consistency.

The training and testing environments for all of the experiments and pre-processing step were done by a workstation computer with twenty Intel Core i9-10900X CPUs at a clock speed of 3.70 GHz and an NVIDIA RTX 2080TI graphic card. Pytorch with the Torchvision library was used to design experimental models.

TABLE I

AVERAGE ACCURACIES AND FOOLING RATE FOR SIGNAL-AGNOSTIC (SA) PERTURBATION OF THE PROPOSED METHOD (VGPN-SA), NOISY BASELINE (GAUSSIAN NOISE), AUTOENCODER MODEL(AE), AND CURRENT STATE-OF-THE-ART METHODS(DF-UAP [26], TLM-UAP [19]) ON UNIVERSAL ADVERSARIAL ATTACK ON EEG-BASED DATASET

dataset	model	Within subject									Cross subject								
		clean	non-targeted attack					targeted attack			clean	non-targeted attack					targeted attack		
		Within Acc	Noisy FR	AE FR	DF FR	TLM FR	GPN FR	AE Acc	TLM Acc	GPN Acc	Cross Acc	Noisy FR	AE FR	DF FR	TLM FR	GPN FR	AE Acc	TLM Acc	GPN Acc
Amigos	EEGNet	67.46	19.63	45.81	34.97	58.58	<b>70.76</b>	47.74	83.85	<b>89.23</b>	29.96	19.46	46.23	36.05	61.00	<b>77.04</b>	68.71	83.73	<b>87.94</b>
	D.ConvNet	50.85	17.28	48.27	31.95	71.11	<b>82.36</b>	1.10	98.77	<b>99.02</b>	28.17	23.62	33.42	31.51	<b>71.32</b>	68.43	47.12	98.62	<b>98.93</b>
	S.ConvNet	59.59	20.48	68.49	35.36	75.89	<b>77.74</b>	87.63	99.59	<b>99.77</b>	30.39	16.84	43.74	40.11	71.55	<b>74.64</b>	86.65	99.47	<b>99.68</b>
	TIDNet	80.51	28.13	50.00	29.74	49.23	<b>67.21</b>	37.44	69.93	<b>85.17</b>	28.78	34.03	43.64	33.38	56.23	<b>69.01</b>	43.04	66.19	<b>82.54</b>
	ResNet	81.18	23.59	59.29	30.40	61.31	<b>71.25</b>	61.64	73.95	<b>86.04</b>	28.78	31.66	65.22	28.90	61.20	<b>72.35</b>	81.87	72.35	<b>83.77</b>
	VGG	59.83	22.95	39.71	26.43	53.74	<b>63.03</b>	37.09	76.61	<b>82.08</b>	30.20	26.17	37.09	30.08	52.57	<b>59.43</b>	39.41	69.04	<b>78.06</b>
DEAP	EEGNet	55.51	27.93	71.96	56.94	75.50	<b>83.04</b>	78.67	98.70	<b>99.64</b>	25.03	12.77	27.09	20.52	66.56	<b>75.75</b>	70.38	97.97	<b>99.18</b>
	D.ConvNet	48.65	16.64	78.29	43.23	60.04	<b>82.29</b>	94.00	99.89	<b>99.92</b>	26.15	23.14	21.28	29.72	73.20	<b>73.33</b>	36.39	99.87	<b>99.92</b>
	S.ConvNet	51.04	17.69	73.91	49.54	60.20	<b>84.80</b>	83.82	99.91	<b>99.93</b>	23.67	11.01	23.71	19.77	<b>79.86</b>	75.13	77.75	99.92	<b>99.93</b>
	TIDNet	45.71	30.57	50.63	16.52	68.30	<b>71.92</b>	42.21	87.94	<b>94.30</b>	26.06	26.71	20.86	15.30	65.22	<b>67.50</b>	46.78	85.15	<b>93.12</b>
	ResNet	54.64	34.12	68.17	20.92	64.95	<b>74.29</b>	69.30	80.86	<b>93.36</b>	25.76	19.36	25.18	10.00	67.55	<b>75.16</b>	80.87	74.50	<b>89.52</b>
	VGG	50.51	23.15	24.87	20.09	65.10	<b>70.09</b>	40.61	90.25	<b>95.24</b>	27.03	23.10	24.72	20.70	57.53	<b>62.91</b>	29.18	84.52	<b>93.07</b>
NER2015	EEGNet	70.95	2.03	88.78	84.55	90.85	<b>90.87</b>	96.58	99.95	<b>100.00</b>	71.17	0.30	19.03	89.71	<b>99.05</b>	75.75	56.79	99.61	<b>100.00</b>
	D.ConvNet	71.17	6.13	19.21	44.96	42.99	<b>75.50</b>	60.57	99.96	<b>100.00</b>	62.82	5.41	17.69	33.11	54.24	<b>73.33</b>	62.08	99.51	<b>99.97</b>
	S.ConvNet	73.09	2.92	7.23	61.17	27.45	<b>83.65</b>	59.17	99.97	<b>100.00</b>	69.97	4.33	10.83	<b>78.93</b>	9.01	75.13	57.39	99.12	<b>100.00</b>
	TIDNet	70.63	12.63	17.46	59.72	39.94	<b>76.04</b>	63.50	99.92	<b>100.00</b>	64.08	9.68	13.44	40.91	54.12	<b>67.50</b>	59.96	99.44	<b>99.99</b>
	ResNet	68.83	4.49	27.49	14.72	30.50	<b>50.53</b>	74.38	81.54	<b>85.86</b>	64.83	3.15	75.00	38.44	49.88	<b>75.16</b>	97.06	<b>93.85</b>	81.22
	VGG	69.35	7.01	15.72	18.21	57.56	<b>62.37</b>	58.69	99.39	<b>99.95</b>	66.48	6.87	12.90	15.21	29.37	<b>62.91</b>	54.5	93.45	<b>97.82</b>
Physionet	EEGNet	60.68	3.19	19.91	49.89	<b>74.77</b>	73.37	39.85	<b>99.89</b>	99.67	63.16	1.92	16.46	54.87	72.72	<b>99.05</b>	42.63	99.84	<b>99.98</b>
	D.ConvNet	60.82	2.70	10.41	36.77	<b>74.28</b>	73.99	31.68	99.81	<b>99.94</b>	59.43	2.93	10.87	38.97	74.01	<b>72.19</b>	37.11	99.77	<b>99.92</b>
	S.ConvNet	61.22	5.40	25.82	50.41	<b>74.99</b>	72.62	57.88	99.86	<b>99.99</b>	58.59	4.14	8.57	58.50	75.44	<b>92.57</b>	47.25	99.77	<b>99.99</b>
	TIDNet	59.43	10.81	17.76	51.86	71.80	<b>71.96</b>	31.24	99.60	<b>99.97</b>	57.87	7.16	13.76	55.23	72.42	<b>76.54</b>	38.05	99.38	<b>99.92</b>
	ResNet	49.67	12.72	69.95	35.19	70.84	<b>70.94</b>	89.19	94.90	<b>98.93</b>	45.42	6.64	51.29	42.67	70.38	<b>78.58</b>	67.69	94.36	<b>98.71</b>
	VGG	60.58	7.21	24.33	44.40	71.66	<b>80.39</b>	45.08	99.35	<b>99.84</b>	56.42	6.77	22.02	46.87	72.09	<b>77.28</b>	58.69	99.02	<b>99.73</b>
average		61.75	14.98	44.05	39.50	62.15	<b>74.21</b>	60.79	93.10	<b>96.17</b>	44.52	13.63	34.38	37.89	63.19	<b>72.94</b>	58.51	92.02	<b>95.12</b>

Bold font numbers indicate the best result in each attack method.

## B. Signal-Agnostic Attack (SA)

We conducted experiments to classify signal-agnostic generated by the proposed method. To verify the effectiveness of the proposed method, we compared the result with previous approaches such as DF-UAP [26] and TLM-UAP [19], which generate signal-agnostic universal adversarial examples. We also conducted experiments with AutoEncoder (AE) to compare the performance of GPNs with a general generative model.

The experiments were conducted in two methods, i.e., non-targeted and targeted attacks. The same method was applied to both within- and cross-subject experiments. For the signal-agnostic attacks, all universal perturbations were made in advance using the training data only, then evaluated with test data.

To evaluate the performance, we employed two evaluation metrics. The first metric is the fooling rate (FR), which is applied to non-targeted attacks only. In this study, we defined FR to make a prediction different from the original prediction and defined it as:

$$FR = \frac{\sum_{i=1}^N \mathbb{1}(M(x_i + \delta) \neq M(x_i))}{N}, \quad (20)$$

where  $\mathbb{1}(\cdot)$  is the indicator function that returns 1 if the argument is true, else returns 0, and  $N$  is the total number of test

segments on which the attack is evaluated. The Second metric is classification accuracy (Acc), the percentage that predicts the ground truth or target class for a targeted attack and defined as:

$$Acc = \frac{\sum_{i=1}^N \mathbb{1}(M(x_i + \delta) = t)}{N}, \quad (21)$$

where  $t$  is a target class. In the case of a targeted attack, the higher the accuracy, the better the performance. The experimental result of classification FR and Acc achieved by the perturbations from the proposed methods and comparing approaches are presented in Table I.

Results of the within-subject experiment suggest that the proposed method is superior to other approaches in non-targeted and targeted attacks. In a non-targeted attack, four universal perturbations increase the fooling rate when compared to the noisy baseline. However, we verified that the proposed method achieved superior performance in FR over other approaches. The FR of our methods is 74.21%, which is a least 12% better performance than other algorithms. In most datasets except Physionet, universal examples by our method achieved the best performance on FR. In a targeted attack, the average accuracy of universal perturbation generated by our method is slightly higher than TLM-UAP. Note DF-UAP is not designed for a targeted

TABLE II  
AVERAGE ACCURACIES AND FOOLING RATE OF SIGNAL-SPECIFIC (SS) PERTURBATION (VANILA GPN-SS (vGPN), CONDITIONAL GPN-SS (cGPN), MULTIPLE GPN-SS (mGPN)) OF THE PROPOSED METHOD AND EXTENDED VERSIONS

dataset	Within subject							Cross subject						
	clean Acc	non-targeted attack			targeted attack			clean Acc	non-targeted attack			targeted attack		
		vGPN FR	cGPN FR	mGPN FR	vGPN Acc	cGPN Acc	mGPN Acc		vGPN FR	cGPN FR	mGPN FR	vGPN Acc	cGPN Acc	mGPN Acc
Amigos	67.46	79.45	<b>82.96</b>	77.95	98.12	<b>98.86</b>	97.92	29.96	<b>75.34</b>	68.25	73.78	96.15	89.79	<b>96.18</b>
DEAP	50.85	<b>87.00</b>	84.78	84.71	<b>99.48</b>	98.78	99.26	28.17	60.83	63.75	<b>66.41</b>	<b>95.41</b>	83.77	90.67
NER2015	59.59	<b>81.31</b>	71.36	77.62	99.74	<b>99.80</b>	94.80	30.39	70.85	<b>80.47</b>	64.87	87.39	<b>96.73</b>	94.99
Physionet	80.51	65.99	<b>71.98</b>	70.43	84.88	82.82	<b>87.76</b>	27.03	64.58	<b>68.12</b>	64.75	83.34	86.68	<b>96.18</b>
average	61.75	78.44	<b>79.82</b>	76.85	<b>95.57</b>	94.56	94.93	44.52	67.90	<b>70.15</b>	67.45	90.44	87.78	<b>92.12</b>

Bold font numbers indicate the best result in each attack method.

attack; therefore, we did not consider comparing. The average Acc of the vGPN-SA method is 3% higher than the second-best method.

We have similar results for the cross-subject experiment. In a non-targeted attack, FR of the proposed GPN-SA shows more than 9% and 28% than the second best result (TLM-UAP) and noisy baseline, respectively. In a targeted attack, the average Acc of our method is 3% higher than TLM-UAP as with the within-subject experiment. We note that the overall performance of all evaluation approaches for the cross-subject is 1–2% lower than that of the within-subject experiment. We analyzed that the correlation of EEG signals among subjects is lower than the correlation of segments of the same subject as reported in other studies [19], [38].

We also reasoned our higher performance against the same victim classification models. DF-UAP employs Deepfool, updating perturbation iteratively. The proposed method optimizes the parameters of the generative model, whereas TLM-UAP optimizes universal perturbation directly. This means that final perturbations generated from features work better for untrained data. Nonetheless, using a generative model is not the main factor of this, as the results from the performances of AE are slightly better than those that are noisy examples, and they underperform when compared to conventional methods.

### C. Signal-Specific Attack (SS)

We also experimented with signal-specific attacks using three different models. First, we used the same generative model (vGPN) yet employed a different training strategy. When vGPN was used for signal-dependent attacks, EEG segments were used as input vectors instead of the fixed noise, which is the input of the signal-agnostic method. The other two methods generate signal-specific perturbations using conditional GPN (cGPN) and multiple GPN (mGPN). As previously described, we crafted UAPs for all combinations of datasets, classification models, attack methods, and target classes for vGPN. In contrast, we trained generative networks only per dataset when using cGPN or mGPN. This means that any perturbation can be generated for all classification models, attack types, and target classes with single weights trained by a dataset once. Therefore, the training time could be efficiently reduced. This is one of the crucial advantages of using extended versions of vGPN. The reason that a separate training model is needed for each dataset is that

the size of the datasets is different, so it is hard to share the same classification model.

To demonstrate the validity of the signal-specific attacks, we also employed the same procedures being used in signal-agnostic attacks, i.e., non-targeted attacks and targeted attacks. The experimental results for the within- and cross-subject experiments were described in Table II. An interesting finding that can be observed from the experiments, the performance of cGPN-SS in non-targeted attacks shows superior performance compared to other methods both in within- and cross-subject experiments. However, the average FR of mGPN-SS was the lowest among the three in the non-targeted attack. In the within-subject experiment, the average Acc of vGPN-SS is 0.5–1% higher than cGPN-SS and mGPN-SS in the targeted attack. However, we verified that the performance of mGPN in targeted attack is 2–4% higher than other methods for the cross-subject experiment.

The performance comparison results of signal-agnostic and signal-specific perturbation suggest several findings. First, the overall performance of the within-subject experiment is higher than those of the cross-subject method. We analyzed that the similarity between the training and test segments of within-subject is higher than cross-subject. Second, contrary to our expectations, the performance of the signal-agnostic learning method was higher despite the high constraint. Finally, we validated that the performance of our extended version, i.e., cGPN and mGPN, are slightly lower despite the small amount of training, which shows the efficiency of the proposed methods.

### D. Statistical Test

We conducted a one-way repeated-measures analysis of variance (RM-ANOVA) to statistically evaluate the efficacy of GPNs and to ascertain the significance of certain contributing factors. The experiment results were tested from four different perspectives. First, we conducted RM-ANOVA, four datasets (Amigos, DEAP, NER2015, PhysioNet) as response variables and three attack methods (DF, TLM, vGPN-SA) as factors as illustrated in Fig. 5(a). This experiment sheds light on the significance of attack methods in signal-agnostic attacks. We obtained a  $p$ -value of  $8.70e-7$ , indicating the impact of attack methods on the fooling rate. The results indicated that regardless of the domain dataset, the fooling rate was significantly affected by the attack methods ( $p < 0.001$ ).



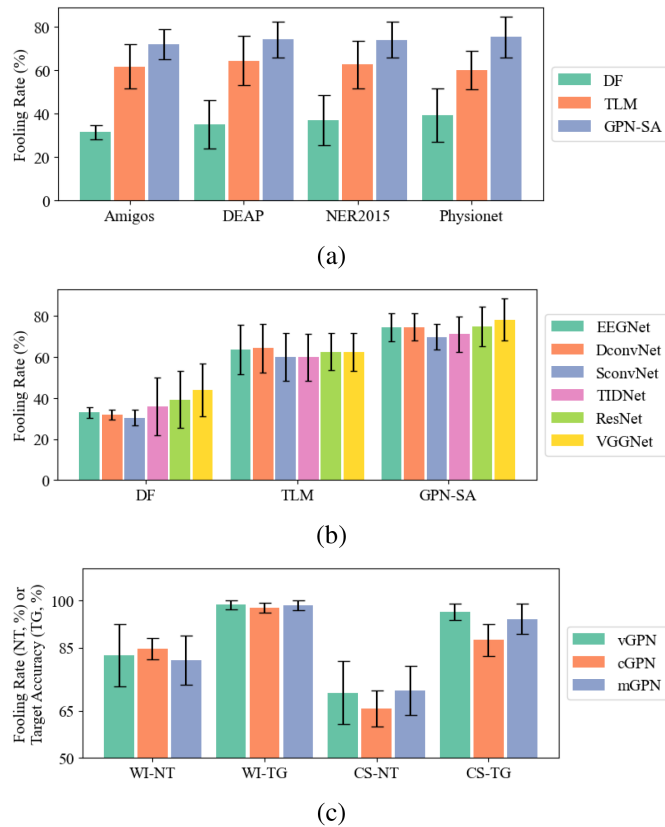


Fig. 5. Average and variance of fooling rates. Error bars denote 2 standard errors of the mean. (a) Performance of UAP-generated models for datasets in within-subject. (b) Performance of victim models for UAP-generated models. (c) Performance of vGPN and extended versions (cGPN, vGPN) for various experimental setup. WL, CS refers to within-subject, cross-subject, and NT, TG refers to non-targeted attack and targeted attack, respectively.

Second, we performed RM-ANOVA with fooling rates grouped based on datasets to determine if the fooling rate varies depending on the victim models. In this test, we set attack methods as response variables and victim models as factors. With a  $p$ -value of  $1.33e-9$ , the results showed that the fooling rate significantly depended on the victim model ( $p < 0.001$ ). Our findings revealed that VGGNet is highly vulnerable to adversarial attacks, while SconvNet is relatively robust across all three attack methods, as demonstrated in Fig. 5(b). However, GPN-SA remained effective against all six victim models compared to other attack methods.

Finally, we conducted RM-ANOVA on signal-specific tasks and GPN-variants, as shown in Fig. 5(c), and found that the type of GPN did not significantly affect the fooling rate and target accuracy across four tasks, even with  $p < 0.005$ . This means that there is no performance difference between the extended versions of GPN and vGPN, which allows us to use them in various ways. This will be explained in the discussion section.

### E. Ablation Study on Training Method

We conducted ablation study to further investigate the effectiveness of the proposed approaches. We applied various

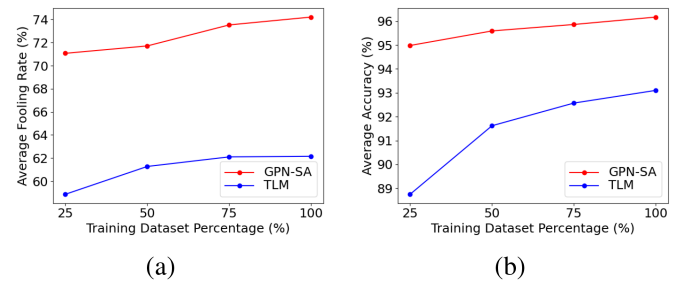


Fig. 6. Performed with different sizes of training dataset used for generating perturbations. (a) Average fooling rates of non-target attacks (b) average accuracy of target-attacks with TLM-UAP [19] and GPN-SA in 5-fold within-subject experiment. Average fooling rates and accuracy are the mean FR of 6 classification models for all EEG dataset.

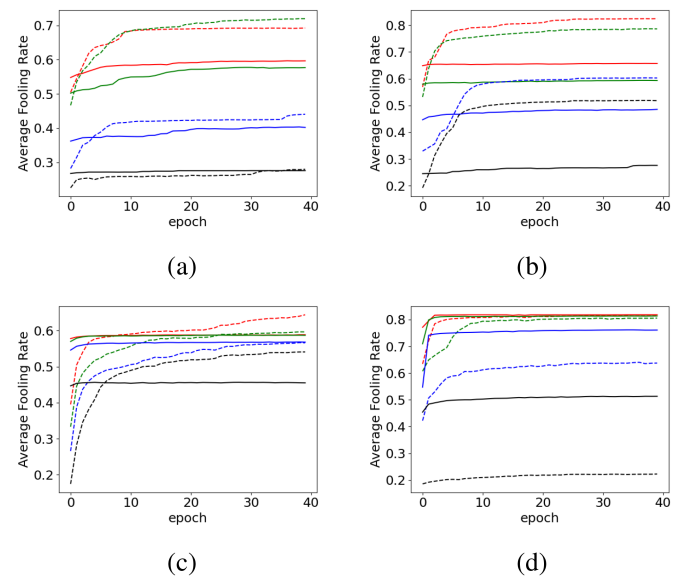


Fig. 7. Average FR of non-target attacks with vGPN-SA and vGPN-SS in 5-fold within-subject experiment, performed with different sizes of epochs and numbers of channels. The solid lines and dotted lines represent average FR of vGPN-SA and vGPN-SS, respectively. Average Fooling rate is mean FR of 6 classification models for each dataset: (a) Amigos (b) DEAP (c) NER2015 (d) Physionet.

datasets, channels, and epochs for our training methods. Fig. 6 shows the performance of the non-targeted and the targeted attacks of GPN-SA and TLM using a 25-100% training dataset. Our method can sufficiently fool the classification model with a small amount of data (even 25%). Furthermore, compared to previous methods, our model suffers less performance degradation as the number of datasets decreases. To investigate the effectiveness of the number of EEG channels, we evaluated the performance with different numbers of channels by reducing the channels to 100%, 75%, 50%, and 25%. We also investigated the performance at different lengths of epochs from 1 to 40. Results of vGPN-SA and vGPN-SS in non-targeted attacks are shown in Fig. 7. The average FR is the mean FR value of 6 classification models for each dataset. In both vGPN-SA and vGPN-SS, we verified that the FR increases as the number of channels perturbations applied increases. Also, it can be observed that the reduction in performance for channels of 100%

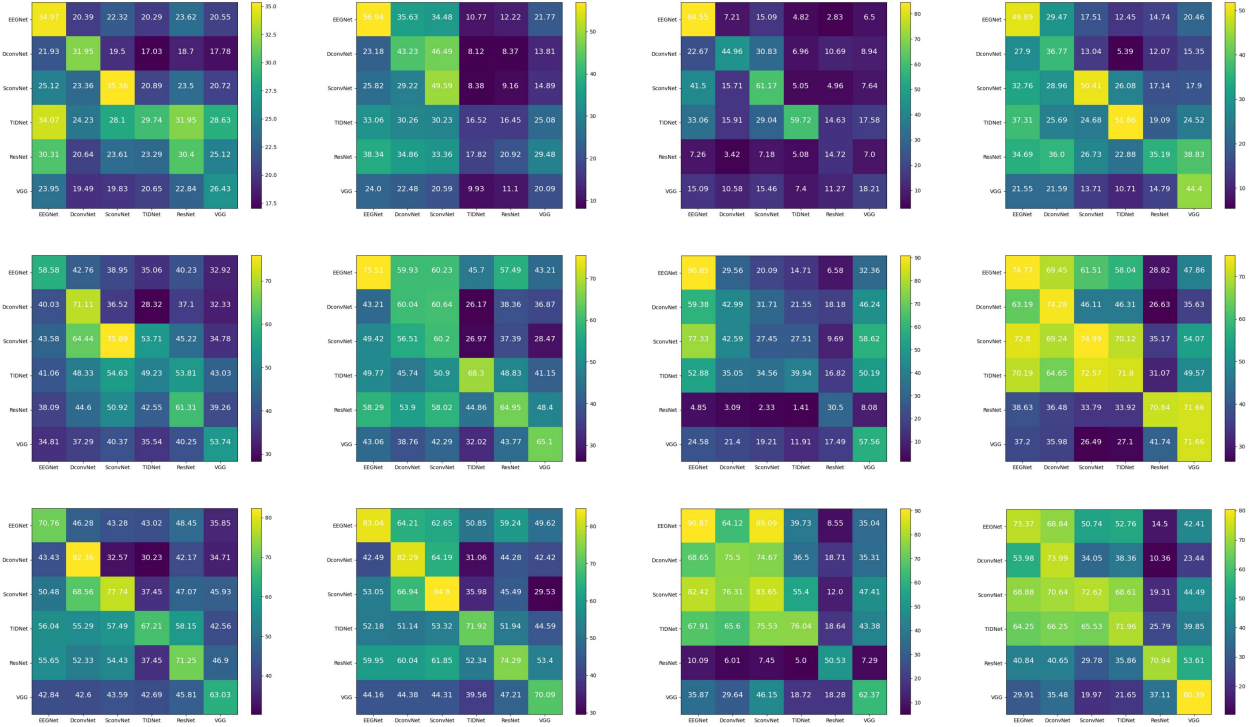


Fig. 8. Confusion matrices of the transferability on non-targeted attacks in the within-subject experiment. Matrix in each row, from top to bottom, used perturbations generated by DF-UAP, TLM-UAP, and GPN-SA. Matrix in each column, from left to right, is trained using Amigos, DEAP, NER2015, and PhysioNet.

TABLE III  
COMPARISONS OF TRANSFERABILITY

	metric	DF-UAP	TLM-UAP	GPN-SA
non-targeted	FR ( $\uparrow$ )	20.59	40.44	<b>44.61</b>
targeted	Acc ( $\uparrow$ )	-	48.57	<b>51.12</b>

Bold font numbers indicate the best value in each attack method.

to 75% is relatively small compared to the other performance reductions, i.e., 75% to 50%, 50% to 25%. The signal-agnostic method (vGPN-SA) converges earlier within ten epochs for all datasets. The results suggest that both methods converge within 20 epochs.

### F. Transferability Across Classification Networks

We investigated the transferability across classification models of the proposed methods and previous approaches in the 5-fold within-subject experiment. We generated universal adversarial perturbations for each dataset, classification model, and target class using DF-UAP, TLM-UAP, and GPN-SA. Then, we applied universal perturbations or pre-trained weights to other classification models and validated the performance.

The overall performance of transferability is shown in Table III. We verify that transferability with the proposed methods is superior to previous approaches. In the non-targeted attack for the within-subject experiment, the average FR of GPN-SA is more than 4% higher than TLM-UAP. In the targeted attack, GPN-SA shows the best performance in average Acc than other

methods. As demonstrated by our experiments, our approaches can apply to the black-box attacks that utilize substitute models.

Fig. 8 shows confusion matrices that represent FR of non-targeted attacks across classification models. In the confusion matrix, universal adversarial perturbations were made by training with a model corresponding to each row, and with this, an evaluation was conducted with a model corresponding to each column. Therefore, diagonal values of the matrix mean the fooling rates when training and testing using the same classification model. Observe that the perturbations crafted for some of the architectures generalize very well across multiple networks. In several cases, we observed that transferability is relatively high between models dedicated to EEG classification (EEGNet, DeepConvNet, ShallowConvNet, and TIDNet), and so is the case between image classification models (ResNet and VGG). We estimate that this is due to structural similarity, e.g., whether or not convolution in the temporal direction is used between the two model groups. Model-based methods are known to perform better on unseen data and are more robust to overfitting than optimization-based methods. Our experiments show that our model based method has better generality than an optimization method.

### G. Discussion

The performance of the proposed GPN surpasses that of all the experimental configurations, which could be attributed to the utilization of a non-optimization-based generation model. Notably, our approach outperforms the general generative model

known as Autoencoder. These results indicate the suitability of the proposed model for EEG datasets.

The within-subject and cross-subject experiments are investigated to evaluate the performance of the proposed model on both targeted and non-targeted attack situation. In non-targeted attack, the performance is evaluated by the FR. The proposed model shows 79.82% of FR for the signal-specific perturbation and 74.21% of FR for the signal-agnostic perturbation.

For targeted attack, the performance is evaluated by the Acc. The Acc of the proposed model is as high as 95.57% for the signal-specific perturbation generation, and 96.17% for the signal-agnostic perturbation generation. The overall performance of the within-subject experiment is slightly higher than those of the cross-subject experiments.

Certain limitations should be acknowledged. Given that our proposed method involves training a generative model, the generation process is slower compared to a TML-UAP that directly optimizes the perturbations. However, this issue can be mitigated through the use of extended GPNs. It's important to note that the perturbations generated by existing methods are contingent upon the dataset, the sacrificial model, and the target class, which includes non-targeted methods. In a scenario where we aim to deceive a dataset with  $C$  labels and  $N$  victim models using a method with a generation time of  $T$ , the total time required would be  $C \times N \times T$ . In contrast, our proposed cGPN and mGPN methods require a constant time  $T$ , irrespective of the number of labels in the dataset or the number of victim models. This allows for the efficient generation of universal examples.

## VI. CONCLUSION

This paper proposed an efficient generative model named Generative Perturbation Network (GPN), which can craft signal-agnostic and signal-specific perturbations for non-targeted and targeted attacks. Our key findings revealed that the GPN could effectively deceive EEG-based BCI classifiers, achieving high fooling rates across a variety of datasets and victim models. Notably, the GPN demonstrated the capability to generate perturbations that are transferable across different models, a significant contribution to the field. Additionally, we expanded the GPN to generate class-wise and multi-class perturbations, offering increased flexibility in adversarial attack scenarios. These advancements underscore the potential of GPN to be a versatile tool in the realm of EEG-based BCIs. The implications of this research are profound, highlighting the vulnerability of EEG-based BCIs to adversarial attacks. This understanding is crucial for the future development of more robust BCI systems. As for future work, we aim to explore further the potential of GPN in other neural network models and its applicability in real-world scenarios. We also plan to investigate countermeasures to these adversarial attacks, contributing to the development of more secure and reliable BCI systems.

## REFERENCES

[1] X. Gu et al., "EEG-based brain-computer interfaces (BCIs): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 5, pp. 1645–1666, Sep./Oct. 2021.

[2] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based brain-computer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, 2019, Art. no. 011001.

[3] T. Xu, Y. Zhou, Z. Wang, and Y. Peng, "Learning emotions EEG-based recognition and brain activity: A survey study on BCI for intelligent tutoring system," *Procedia Comput. Sci.*, vol. 130, pp. 376–382, 2018.

[4] L. M. Nirenberg, J. Hanley, and E. B. Stear, "A new approach to prosthetic control: EEG motor signal tracking with an adaptively designed phase-locked loop," *IEEE Trans. Biomed. Eng.*, vol. BME-18, no. 6, pp. 389–398, Nov. 1971.

[5] X. Chen, B. Zhao, Y. Wang, and X. Gao, "Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm," *J. Neural Eng.*, vol. 16, no. 2, 2019, Art. no. 026012.

[6] C. Guger, V. Prabhakaran, R. Spataro, D. J. Krusienski, and A. O. Hebb, "Breakthrough BCI applications in medicine," *Front. Neurosci.*, vol. 14, 2020, Art. no. 598247.

[7] Y. Ke, P. Liu, X. An, X. Song, and D. Ming, "An online SSVEP-BCI system in an optical see-through augmented reality environment," *J. Neural Eng.*, vol. 17, no. 1, 2020, Art. no. 016066.

[8] X.-W. Wang, D. Nie, and B.-L. Lu, "Emotional state classification from EEG data using machine learning approach," *Neurocomputing*, vol. 129, pp. 94–106, 2014.

[9] L. Qin, L. Ding, and B. He, "Motor imagery classification by means of source analysis for brain-computer interface applications," *J. Neural Eng.*, vol. 1, no. 3, 2004, Art. no. 135.

[10] A. H. Shoeb and J. V. Gutttag, "Application of machine learning to epileptic seizure detection," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 975–982.

[11] C. Lehmann et al., "Application and comparison of classification algorithms for recognition of Alzheimer's disease in electrical brain activity (EEG)," *J. Neurosci. Methods*, vol. 161, no. 2, pp. 342–350, 2007.

[12] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[13] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[14] A. Chaubey, N. Agrawal, K. Barnwal, K. K. Guliani, and P. Mehta, "Universal adversarial perturbations: A survey," 2020, *arXiv:2005.08087*.

[15] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773.

[16] K. R. Mopuri, U. Ojha, U. Garg, and R. V. Babu, "NAG: Network for adversary generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 742–751.

[17] X. Zhang and D. Wu, "On the vulnerability of CNN classifiers in EEG-based BCIs," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 814–825, May 2019.

[18] X. Jiang, X. Zhang, and D. Wu, "Active learning for black-box adversarial attacks in EEG-based brain-computer interfaces," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2019, pp. 361–368.

[19] Z. Liu, L. Meng, X. Zhang, W. Fang, and D. Wu, "Universal adversarial perturbations for CNN classifiers in EEG-based BCIs," *J. Neural Eng.*, vol. 18, no. 4, 2021, Art. no. 0460a4.

[20] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for eeg-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, 2018, Art. no. 056013.

[21] R. T. Schirrmeyer et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[22] D. Kostas and F. Rudzicz, "Thinker invariance: Enabling deep neural networks for BCI across more people," *J. Neural Eng.*, vol. 17, no. 5, 2020, Art. no. 056008.

[23] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, "A simplified CNN classification method for MI-EEG via the electrode pairs signals," *Front. Hum. Neurosci.*, vol. 14, 2020, Art. no. 338.

[24] J. Cho and H. Hwang, "Spatio-temporal representation of an electroencephalogram for emotion recognition using a three-dimensional convolutional neural network," *Sensors*, vol. 20, no. 12, 2020, Art. no. 3491.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[26] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

- [27] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4422–4431.
- [28] B. Feng, Y. Wang, and Y. Ding, "Saga: Sparse adversarial attack on EEG-based brain computer interface," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 975–979.
- [29] L. Meng, C.-T. Lin, T.-P. Jung, and D. Wu, "White-box target attack for EEG-based BCI regression problems," in *Proc. Int. Conf. Neural Inf. Process.*, 2019, pp. 476–488.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [32] J. A. M. Correa, M. K. Abadi, N. Sebe, and I. Patras, "AMIGOS: A dataset for affect, personality and mood research on individuals and groups," *IEEE Trans. Affect. Comput.*, vol. 12, no. 2, pp. 479–493, Apr.–Jun. 2021.
- [33] S. Koelstra et al., "DEAP: A database for emotion analysis; using physiological signals," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 18–31, Jan.–Mar. 2012.
- [34] P. Margaux, M. Emmanuel, D. Sébastien, B. Olivier, and M. Jérémie, "Objective and subjective evaluation of online error correction during p300-based spelling," *Adv. Hum.-Comput. Interact.*, vol. 2012, 2012, Art. no. 578295.
- [35] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.
- [36] A. L. Goldberger et al., "Physiobank, physiokit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [38] M. Hajinoroozi, Z. Mao, T.-P. Jung, C.-T. Lin, and Y. Huang, "EEG-based prediction of driver's cognitive performance by deep convolutional neural network," *Signal Process.: Image Commun.*, vol. 47, pp. 549–555, 2016.