Improving Adversarial Robustness via Feature Pattern Consistency Constraint

Jiacong Hu\textsuperscript{1,3,4}, Jingwen Ye\textsuperscript{5}, Zunlei Feng\textsuperscript{2,3,4,\ast}, Jiazhen Yang\textsuperscript{2}, Shunyu Liu\textsuperscript{1}, Xiaotian Yu\textsuperscript{1}, Lingxiang Jia\textsuperscript{1} and Mingli Song\textsuperscript{1,3,4}

\textsuperscript{1}College of Computer Science and Technology, Zhejiang University
\textsuperscript{2}School of Software Technology, Zhejiang University
\textsuperscript{3}State Key Laboratory of Blockchain and Security, Zhejiang University
\textsuperscript{4}Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
\textsuperscript{5}National University of Singapore

jiaconghu@zju.edu.cn, jingweny@nus.edu.sg, zunleifeng@zju.edu.cn, yangjiazhen0915@outlook.com, {liushunyu,yuxiaotian,lingxiangjia,brooks\{}@zju.edu.cn

Abstract
Convolutional Neural Networks (CNNs) are well-known for their vulnerability to adversarial attacks, posing significant security concerns. In response to these threats, various defense methods have emerged to bolster the model's robustness. However, most existing methods either focus on learning from adversarial perturbations, leading to overfitting to the adversarial examples, or aim to eliminate such perturbations during inference, inevitably increasing computational burdens. Conversely, clean training, which strengthens the model's robustness by relying solely on clean examples, can address the aforementioned issues. In this paper, we align with this methodological stream and enhance its generalizability to unknown adversarial examples. This enhancement is achieved by scrutinizing the behavior of latent features within the network. Recognizing that a correct prediction relies on the correctness of the latent feature's pattern, we introduce a novel and effective Feature Pattern Consistency Constraint (FPCC) method to reinforce the latent feature's capacity to maintain the correct feature pattern. Specifically, we propose Spatial-wise Feature Modification and Channel-wise Feature Selection to enhance latent features. Subsequently, we employ the Pattern Consistency Loss to constrain the similarity between the feature pattern of the latent features and the correct feature pattern. Our experiments demonstrate that the FPCC method empowers latent features to uphold correct feature patterns even in the face of adversarial examples, resulting in inherent adversarial robustness surpassing state-of-the-art models.

1 Introduction
As deep learning continues to evolve, convolutional neural networks (CNNs) have gained widespread application across various domains, including facial recognition [Luo et al., 2022], autonomous driving [Prakash et al., 2021], and person identification [Pu et al., 2023]. However, despite their remarkable performance, CNNs exhibit a notable vulnerability to adversarial attacks. Specifically, adversarial attacks generate samples imbued with minute perturbations that are imperceptible to humans but can mislead the model into making erroneous predictions [Szegedy et al., 2013; Goodfellow et al., 2014b], posing a significant threat to security [Wang et al., 2023; Liu et al., 2023; Ma et al., 2023b].

In the realm of defending against adversarial attacks, a plethora of methods has been proposed, with the majority concentrating on the utilization or elimination of adversarial perturbations generated by adversarial attacks. Adversarial training stands out as a widely acknowledged method for leveraging adversarial perturbations [Madry et al., 2017; Zhang et al., 2019] to enhance model robustness. This approach incorporates adversarial examples during model training, allowing the model to adapt to perturbations. However, a limitation of these methods is the potential overfitting of the trained model to specific types of attacks encountered during training [Madry et al., 2017], often leading to a decline in standard accuracy [Laidlaw et al., 2020]. On the other hand, for the elimination of adversarial perturbations, adversarial purification has gained significant traction recently [Wang et al., 2022; Nie et al., 2022]. This method employs additional generative models to remove adversarial perturbations from the adversarial examples. However, it does not inherently bolster the model's robustness and may contribute to increased computational costs during inference [Croce et al., 2022]. The limitations imposed by the aforementioned methods, which concentrate on adversarial perturbations, raise a crucial question: Can we break free from the paradigm of adversarial perturbations to enhance model robustness?

Indeed, several methods have been proposed to enhance model robustness, relying solely on clean examples [Mustafa et al., 2019; Pang et al., 2019; Li et al., 2021]. The majority of these methods aim to reduce intra-class distances, thereby increasing inter-class margins. However, the augmented margin is established within the feature space of clean examples, posing a challenge in adapting to adversarial perturbations
and resulting in inferior robustness compared to adversarial training. Furthermore, the connection between the behavior of latent features and correct predictions in clean examples within these studies has not been thoroughly investigated.

In this paper, we redirect our focus to clean examples, delving into the behavior of latent features within the model. Specifically, through an analysis of feature behavior during correct predictions, we posit that correct predictions occur only when the feature pattern is correct. This pattern is defined as the relative size among dimensions in the latent feature vector. Motivated by this insight, we propose a novel and effective Feature Pattern Consistency Constraint (FPCC) method to enhance the latent feature’s capability in upholding the correct feature pattern. Within FPCC, Spatial-wise Feature Modification and Channel-wise Feature Selection are introduced, approaching the problem from the perspective of adding noise and reducing features. This is done to improve robustness in scenarios with increasing interferential features and decreasing critical features when adversarial perturbations are added to clean examples. Subsequently, a Pattern-based Robustness Optimization is presented to constrain the similarity between the feature pattern of latent features and the correct feature pattern. Experiments demonstrate that the proposed FPCC method enables latent features to uphold correct feature patterns even in the face of adversarial examples, leading to the model’s inherent adversarial robustness surpassing state-of-the-art methods.

Our contributions can be summarized as follows:

- This paper introduces a novel and effective FPCC method, which comprises Spatial-wise Feature Modification and Channel-wise Feature Selection, along with the Pattern-based Robustness Optimization technique.
- The proposed FPCC method is plug-and-play, seamlessly integratable into deep neural networks during training to augment the intrinsic robustness of the model, all without incurring any additional computational overhead during inference.
- Extensive experiments demonstrate the effectiveness of the proposed FPCC method, showcasing superior adversarial robustness and inference speed compared to state-of-the-art methods.

2 Related Work

Adversarial defense methods can be categorized into two distinct types: static defense and adaptive test-time defense.

2.1 Static Defense

Within the realm of static defense, both the inputs and the parameters of the model remain constant during the inference process. One of the most effective defenses in this category is adversarial training [Madry et al., 2017], which involves training models with adversarial examples [Zhang et al., 2019; Cheng et al., 2020; Gowal et al., 2020]. However, many adversarial training approaches can only defend against specific attacks they were trained with [Madry et al., 2017] and often experience a significant accuracy drop on clean data [Laidlaw et al., 2020].

Conversely, some methods [Li et al., 2021; Mustafa et al., 2019; Pang et al., 2019] focus on improving adversarial robustness solely by relying on clean examples to avoid overfitting to adversarial perturbations. Many of these methods aim to reduce inter-class distances, thereby increasing inter-class margins. For instance, Mustafa et al. [Mustafa et al., 2019] introduced an approach to enhance robustness by compelling the features for each class to reside within a convex polytope that is maximally separated from the polytopes of other classes. However, the increased margin is based on the feature space of clean examples, thus achieving poor robustness when countering adversarial perturbations compared to adversarial training. Therefore, some margin-based methods [Mustafa et al., 2019; Pang et al., 2019] need to be combined with adversarial training to further improve robustness.

Different from the aforementioned methods, the proposed FPCC comprises strategies to address the situation of increasing interferential features and decreasing critical features when adversarial perturbations are added to clean examples, along with techniques that constrain the features to uphold the correct feature pattern.

2.2 Adaptive Test-time Defense

Adaptive test-time defense constitutes another pivotal category within adversarial defense methods, wherein the inputs or parameters of the model undergo dynamic alterations.
during the test phase. Adversarial purification [Wang et al., 2022; Nie et al., 2022; Hill et al., 2020] is one of the most popular and effective methods in this category. It utilizes a generative model (e.g., GANs [Goodfellow et al., 2014a], Diffusion models [Ho et al., 2020; Song et al., 2020; Ma et al., 2023a; Li et al., 2023]) to remove perturbations from adversarial examples. While these methods are plug-and-play and successfully defend against most attacks, they do incur significant additional computational costs during inference and do not inherently enhance the model’s robustness [Croce et al., 2022].

Furthermore, some works [Wang et al., 2021; Chen et al., 2021; Kang et al., 2021; Dong et al., 2022; Fu et al., 2021] propose modifying parameters or activations during inference, essentially aiming to reduce the impact of perturbations during network prediction. However, these methods also incur computational overhead, and the improved robustness achieved through these approaches is limited, owing to the complexity of adversarial perturbation and the lack of diversity in training data.

In contrast to adaptive test-time defense methods, the proposed FPCC does not require integrating additional modules or dynamically adjusting defense strategies during inference. Therefore, it incurs no additional computational overhead.

3 Method
In this section, we conduct an analysis of feature behavior in the final layer of classification networks for correctly predicted samples, leading to the introduction of the concept of feature pattern. We extend this concept to other layers of the network, suggesting that a network only achieves correct predictions when its latent features align with these correct feature patterns. Furthermore, we propose the FPCC, which includes Spatial-wise Feature Modification and Channel-wise Feature Selection to address the situation of increasing interferential features and decreasing critical features when adversarial perturbations are added to clean examples. Additionally, FPCC incorporates Pattern-based Robustness Optimization to constrain the modified and selected features, ensuring the maintenance of the correct feature pattern.

3.1 The Feature Pattern
Given a classification network with $K(K \geq 2)$ categories, the softmax function is typically used to calculate the probability of an input sample $x_i$, belonging to the $y_i$-th category, where $y_i \in \{1, 2, \ldots, K\}$ is the true label of $x_i$. For the feature vector $z_i \in \mathbb{R}^K$ of $x_i$ at the final layer of the network, the softmax function is defined as follows:

$$S(z_i[y_i]) = \frac{\exp(z_i[y_i])}{\sum_{k=1}^{K} \exp(z_i[k])}, \quad (1)$$

where $z_i[k]$ is the value in the $k$-th dimension of $z_i$. Given that $y_i$ is the true label of $x_i$, the network is usually trained using cross-entropy loss to maximize the probability $S(z_i[y_i])$:

$$L_{CE,i} = -\log(S(z_i[y_i])). \quad (2)$$

Generally, the network’s ability to correctly predict the label of $x_i$ is not solely based on the specific values of $z_i$, but rather on whether the value in the $y_i$-th dimension $z_i[y_i]$ is relatively larger than the values in other dimensions. We define this relative size among dimensions in the feature vector as the feature pattern. In the network’s final layer, each category has a distinct feature pattern, similar to the true label’s one-hot vector. Correct predictions are made when the sample’s feature pattern is correct, meaning it closely resembles the true category’s feature pattern. We propose that this concept of feature pattern can be generalized to other layers of the network.

Next, we formalize the definition of the feature pattern, taking the $l$-th layer as an example and assuming it to be a fully connected layer (the case of convolutional layers will be discussed later). The feature pattern $p_i^{(l)}$ for a given sample $x_i$ at the $l$-th layer is articulated as the $z$-score normalization of the features $x_i^{(l)}$ at the same layer. This normalization procedure is employed to eliminate the feature scale, thereby upholding solely the relative sizes of the features across each dimension:

$$p_i^{(l)} = \frac{x_i^{(l)} - \bar{x}_i^{(l)}}{\sigma(x_i^{(l)})} + \epsilon, \quad (3)$$

where $\sigma(x_i^{(l)}) = \sqrt{\frac{1}{D_i} \sum_{d=1}^{D_i} (x_i^{(l)}[d] - \bar{x}_i^{(l)})}$, $\bar{x}_i^{(l)} = \frac{1}{D_i} \sum_{d=1}^{D_i} x_i^{(l)}[d]$, $D_i$ is the dimensionality of the features at the $l$-th layer, and $x_i^{(l)}[d]$ denotes the value of the $d$-th dimension within the feature $x_i^{(l)}$.

For a convolutional layer at the $l$-th layer, in contrast to fully connected layers, the latent features are derived by averaging the output feature map:

$$x_i^{(l)} = \frac{1}{H^{(l)} W^{(l)}} \sum_{h=1}^{H^{(l)}} \sum_{w=1}^{W^{(l)}} f_i^{(l)}[h][w], \quad (4)$$

where $f_i^{(l)} \in \mathbb{R}^{D^{(l)} \times H^{(l)} \times W^{(l)}}$ denotes the feature map of $x_i$ at the $l$-th layer, with $H^{(l)}$ and $W^{(l)}$ representing the height and width of the feature map at the $l$-th layer, respectively. Subsequently, the corresponding feature pattern $p_i^{(l)}$ can be determined using Eqn. (3) as delineated above.
As depicted in Fig. 2, within an intermediate layer of the network, the feature pattern of incorrectly predicted samples deviates significantly from the feature pattern of the ground-truth category. In contrast, the feature pattern of correctly predicted samples closely mirrors the feature pattern of the ground-truth category. Consequently, akin to the final layer of the network, at the l-th layer, correct predictions hinge on the correctness of the feature pattern \( p_i^{(l)} \) for \( x_i \), i.e., its similarity to the feature pattern \( d_{y_i}^{(l)} \) of the true category. The similarity between the feature pattern of a sample \( x_i \) and the ground-truth category \( y_i \) can be quantified by the L1 distance between them:

\[
\|p_i^{(l)} - d_{y_i}^{(l)}\|_1 \rightarrow 0 \Rightarrow y_i = \arg \max_k S(z_i[k]),
\]

where \( d^{(l)} \in \mathbb{R}^{K \times D^{(l)}} \) represents the feature patterns of all categories at the l-th layer, and \( d_{y_i}^{(l)} \) signifies the feature pattern of the \( y_i \)-th category. Therefore, the enhancement of network robustness is contingent upon fortifying the capacity of features to uphold the correct feature patterns.

### 3.2 Feature Pattern Consistency Constraint

The introduced FPCC, designed to enhance the ability of features in upholding correct feature patterns, encompasses two strategies: Spatial-wise Feature Modification and Channel-wise Feature Selection. These strategies are devised to address the challenge posed by the increasing interferential features and decreasing critical features. Additionally, the FPCC incorporates a technique known as Pattern-based Robustness Optimization, which serves the purpose of constraining features to uphold the correct feature pattern.

#### Spatial-wise Feature Modification (SFM)

In order to address the challenge of increasing interferential features arising from the addition of adversarial perturbations to clean examples, we introduce Spatial-wise Feature Modification (SFM) to introduce controlled noise into the features.

Specifically, within SFM, a noise term \( \delta \) is randomly generated within a budget of \( \epsilon \) and subsequently added to the sample \( x_i \):

\[
x_i^{aug} = x_i + \delta, \quad \delta \in (-\epsilon, +\epsilon).
\]

It is crucial to highlight that Eqn.(6) diverges from the generation of adversarial examples for adversarial training [Madry et al., 2017] through adversarial attacks, as expressed by the following equation:

\[
x_i^{adv} = x_i + \delta_i^{adv}, \quad \delta_i^{adv} = \arg \max_{\|\delta_i^{adv}\|_1 \leq \epsilon} \mathcal{L}_{CE}(x_i + \delta_i^{adv}, y_i).
\]

Our objective is not to adapt the model specifically to adversarial perturbations but rather to randomly generate noise, via SFM, to simulate the increase in interferential features. Subsequently, we aim to constrain these features to uphold correct feature patterns. Additional insights from comparative distillation experiments pertaining to this aspect can be found in Section 4.4.

#### Algorithm 1 Training Model with PRO

**Require:** Training data \( \{x_i\} \), ground truth labels \( \{y_i\} \), initialized parameters \( \theta \) in classification network, trainable parameters \( \{d^{(l)}| l \in N\} \), hyperparameter \( \lambda \), epochs \( T \), the number of iteration \( t \leftarrow 0 \).

**Ensure:** The parameters \( \theta \).

for \( t \to T \) do

Compute the joint loss by \( \mathcal{L} = \mathcal{L}_{CE} + \lambda \sum_{l \in N} \mathcal{L}_{PT}^{(l)} \).

Compute the gradients \( \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}_{CE}}{\partial \theta} + \lambda \sum_{l \in N} \frac{\partial \mathcal{L}_{PT}^{(l)}}{\partial \theta} \).

Update the parameters \( \theta \) by \( \theta = \arg \min_\theta \mathcal{L} \).

Update the parameters \( d^{(l)}, \forall l \in N \).

end for

#### Channel-wise Feature Selection (CFS)

To address the challenge of decreasing critical features resulting from the addition of adversarial perturbations to clean examples, we introduce Channel-wise Feature Selection (CFS). This approach involves the random selection of a subset of features, followed by the imposition of constraints to maintain the consistency of the feature pattern. In CFS, a random subset of features is selected from a layer preceding the l-th layer (e.g., the \((l - n)\)-th layer) utilizing a Bernoulli distribution:

\[
x_i^{(l-n)} = x_i^{(l-n)} \cdot b_i^{(l-n)}, \quad b_i^{(l-n)} \sim B(\gamma),
\]

where \( b^{(l-n)} \in \mathbb{R}^{D^{(l-n)}} \) is drawn from a Bernoulli distribution with probability \( \gamma \). Subsequently, the selected features from the \((l - n)\)-th layer, after undergoing \( n \) layers of linear operations, are constrained to maintain the correct feature pattern at the l-th layer.

It is important to highlight that Eqn. (8) diverges from the concept of Dropout [Srivastava et al., 2014], where a scaling factor of \( \frac{1}{1-\gamma} \) is utilized to preserve the expected value of the features:

\[
x_i^{(l-n)} = x_i^{(l-n)} \cdot \frac{b_i^{(l-n)} \times \frac{1}{1-\gamma}}{\frac{1}{1-\gamma}} \sim B(\gamma).
\]

The proposed FPCC exclusively constrains the selected original features \( x_i^{(l-n)} \), rather than the scaled features \( x_i^{(l-n)} \times \frac{1}{1-\gamma} \). Comprehensive ablation studies on this aspect are detailed in Section 4.4.

#### Pattern-based Robustness Optimization (PRO)

To ensure that features, modified and selected by SFM and CFS, uphold correct feature patterns—specifically, ensuring the feature pattern of a sample aligns with the feature pattern of its ground-truth category—we propose Pattern-based Robustness Optimization (PRO). PRO constrains the L1 distance between the feature pattern of a sample at different layers (e.g., the l-th layer) and the feature pattern of its ground-truth category:

\[
\mathcal{L}_{PT,i}^{(l)} = ||p_i^{(l)} - d_{y_i}^{(l)}||_1.
\]

It is crucial to emphasize that CFS and PRO are combined, as illustrated in Fig. 1, and are incorporated into various layers of the network. A comprehensive set of ablation studies pertaining to this aspect is presented in Section 4.4.
In PRO, the feature patterns of categories $d^{(i)}$ in the $l$-th layer are treated as learnable parameters and updated through gradient descent in each batch. Let $M$ denote the number of samples in a batch, and $N$ indicate the layers where feature patterns are constrained. The total loss in PRO is given as:

$$
\mathcal{L} = \mathcal{L}_{CE} + \lambda \sum_{l \in N} \mathcal{L}_{PT,l}^{(i)} = \sum_{i=1}^{M} \mathcal{L}_{CE,i} + \lambda \sum_{l \in N} \mathcal{L}_{PT,l}^{(i)}.
$$

$$
= - \sum_{i=1}^{M} \log \left( \frac{\exp(z_l[y_i])}{\sum_{k=1}^{K} \exp(z_l[k])} \right) + \lambda \sum_{i=1}^{M} \sum_{l \in N} \|p_i^{(l)} - d^{(l)}_y\|_1.
$$

where $\lambda$ is a balancing factor for the two losses. The entire PRO process is end-to-end, as shown in Algorithm 1.

In summary, the proposed FPCP method incorporates SFM, CFS, PRO to bolster the network’s inherent adversarial robustness. Notably, during the inference phase, the network obviates the need for the introduction of random noise, feature selection, or the constraint of sample feature patterns.

## 4 Experiments

### 4.1 Experimental Settings

**Datasets and Networks.** We conducted experiments using two widely utilized datasets: CIFAR-10 [Krizhevsky, 2009] and CIFAR-100 [Krizhevsky, 2009]. Additionally, we evaluated the proposed method across three well-established classification networks: WRN-28-10 [Zagoruyko and Komodakis, 2016], ResNet-50 [He et al., 2016], and VGG-16 [Simonyan et al., 2013]. As elaborated in Section 3.2, CFS and PRO are combined and can be integrated into various layers of the networks. The specific configuration details of CFS and PRO are provided in the Supplementary Material.

**Adversarial Attacks.** We assess the effectiveness of the proposed FPCP against a variety of formidable adaptive attacks, including the widely adopted Projected Gradient Descent (PGD) [Madry et al., 2017] and its variant, AutoPGD [Croce and Hein, 2020]. To address the impact of randomization in model training, we also incorporate the Expectation Over Transformation (EOT) technique into PGD (referred to as EOT-PGD) [Liu et al., 2018], with EOT set to 10 by default. Additionally, we compare the performance of our method against single-step non-adaptive attacks and transferability-enhanced attacks. Specifically, we employ three common attacks: FGSM [Goodfellow et al., 2014b], BIM [Kurakin et al., 2018], and RFGSM [Tramèr et al., 2017]. To demonstrate the broad applicability of our method beyond traditional full-pixel attacks, we evaluate its performance against PIM [Gao et al., 2020], an iterative black-box patch attack method, and PIM++ [Gao et al., 2020b], a targeted patch attack approach. Unless explicitly stated otherwise, the $\ell_\infty$ and $\ell_2$ attacks on models trained on the CIFAR-10 and CIFAR-100 datasets use a fixed budget of $\epsilon_\infty = 8/255$ and $\epsilon_2 = 0.5$, respectively.

<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>PGD</th>
<th>AutoPGD</th>
<th>EOT-PGD</th>
<th>FGSM</th>
<th>BIM</th>
<th>MIM</th>
<th>PIM</th>
<th>PIM++</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WRN-28-10</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGDAT</td>
<td>87.24</td>
<td>61.38 / 50.71</td>
<td>61.11 / 50.31</td>
<td>60.16 / 50.51</td>
<td>58.86</td>
<td>50.70</td>
<td>51.48</td>
<td>19.36</td>
<td>19.50</td>
</tr>
<tr>
<td>TRADES</td>
<td>84.60</td>
<td>57.82 / 50.71</td>
<td>57.50 / 50.30</td>
<td>58.62 / 50.71</td>
<td>59.57</td>
<td>50.75</td>
<td>51.75</td>
<td>25.58</td>
<td>23.20</td>
</tr>
<tr>
<td>AWP</td>
<td>84.09</td>
<td>63.59 / 57.68</td>
<td>63.44 / 57.35</td>
<td>62.45 / 57.79</td>
<td>62.92</td>
<td>57.68</td>
<td>58.08</td>
<td>31.23</td>
<td>30.75</td>
</tr>
<tr>
<td>MARGIN</td>
<td>85.80</td>
<td>65.91 / 62.34</td>
<td>65.76 / 62.10</td>
<td>65.91 / 62.34</td>
<td>67.10</td>
<td>62.34</td>
<td>62.82</td>
<td>38.46</td>
<td>37.29</td>
</tr>
<tr>
<td>SCORE</td>
<td>87.84</td>
<td>67.04 / 63.75</td>
<td>66.80 / 63.40</td>
<td>65.04 / 63.25</td>
<td>68.82</td>
<td>63.69</td>
<td>64.34</td>
<td>38.26</td>
<td>37.24</td>
</tr>
<tr>
<td><strong>FPCC (Ours)</strong></td>
<td><strong>88.64</strong></td>
<td><strong>67.28 / 65.58</strong></td>
<td><strong>67.14 / 62.84</strong></td>
<td><strong>66.90 / 65.72</strong></td>
<td><strong>69.33</strong></td>
<td><strong>65.97</strong></td>
<td><strong>66.49</strong></td>
<td><strong>57.95</strong></td>
<td><strong>59.01</strong></td>
</tr>
<tr>
<td><strong>ResNet-50</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGDAT</td>
<td>84.44</td>
<td>61.00 / 51.80</td>
<td>60.79 / 51.39</td>
<td>61.20 / 50.95</td>
<td>58.30</td>
<td>51.76</td>
<td>52.46</td>
<td>22.15</td>
<td>22.21</td>
</tr>
<tr>
<td>TRADES</td>
<td>84.05</td>
<td>57.57 / 51.23</td>
<td>57.52 / 50.73</td>
<td>57.74 / 51.23</td>
<td>59.63</td>
<td>51.25</td>
<td>52.32</td>
<td>26.19</td>
<td>23.29</td>
</tr>
<tr>
<td>AWP</td>
<td>79.32</td>
<td>59.95 / 52.84</td>
<td>59.81 / 52.75</td>
<td>59.85 / 53.84</td>
<td>56.22</td>
<td>52.86</td>
<td>53.18</td>
<td>27.97</td>
<td>28.65</td>
</tr>
<tr>
<td>MARGIN</td>
<td>83.02</td>
<td>63.86 / 59.31</td>
<td>63.63 / 59.14</td>
<td>64.86 / 59.33</td>
<td>63.15</td>
<td>59.33</td>
<td>59.67</td>
<td>35.65</td>
<td>34.99</td>
</tr>
<tr>
<td>SCORE</td>
<td>85.96</td>
<td>66.34 / 61.63</td>
<td>66.16 / 61.37</td>
<td>65.44 / 61.62</td>
<td>66.42</td>
<td>61.69</td>
<td>62.16</td>
<td>36.24</td>
<td>35.4</td>
</tr>
<tr>
<td><strong>FPCC (Ours)</strong></td>
<td><strong>87.65</strong></td>
<td><strong>66.92 / 61.82</strong></td>
<td><strong>66.53 / 61.39</strong></td>
<td><strong>65.23 / 62.72</strong></td>
<td><strong>67.21</strong></td>
<td><strong>61.92</strong></td>
<td><strong>62.22</strong></td>
<td><strong>52.29</strong></td>
<td><strong>54.52</strong></td>
</tr>
<tr>
<td><strong>RHS</strong></td>
<td>90.45</td>
<td>27.68 / 24.20</td>
<td>26.96 / 22.78</td>
<td>27.65 / 23.14</td>
<td>43.65</td>
<td>36.89</td>
<td>38.03</td>
<td>14.56</td>
<td>14.36</td>
</tr>
<tr>
<td>RHS+AT</td>
<td>91.89</td>
<td>55.13 / 46.43</td>
<td>55.00 / 46.09</td>
<td>56.14 / 46.43</td>
<td>57.86</td>
<td>49.56</td>
<td>50.89</td>
<td>22.47</td>
<td>21.38</td>
</tr>
<tr>
<td>MMC</td>
<td><strong>92.70</strong></td>
<td>29.02 / 25.12</td>
<td>28.46 / 24.97</td>
<td>29.23 / 25.32</td>
<td>44.70</td>
<td>38.43</td>
<td>39.95</td>
<td>16.68</td>
<td>16.17</td>
</tr>
<tr>
<td>MMC+AT</td>
<td>81.80</td>
<td>60.34 / 55.03</td>
<td>57.92 / 53.87</td>
<td>61.35 / 55.43</td>
<td>59.32</td>
<td>52.86</td>
<td>53.76</td>
<td>24.32</td>
<td>22.69</td>
</tr>
<tr>
<td>PROC</td>
<td>91.20</td>
<td>29.64 / 26.60</td>
<td>28.06 / 24.32</td>
<td>29.66 / 26.63</td>
<td>42.13</td>
<td>33.65</td>
<td>34.46</td>
<td>15.03</td>
<td>14.68</td>
</tr>
<tr>
<td><strong>FPCC (Ours)</strong></td>
<td><strong>88.64</strong></td>
<td><strong>67.28 / 65.58</strong></td>
<td><strong>67.14 / 62.84</strong></td>
<td><strong>66.90 / 65.72</strong></td>
<td><strong>69.33</strong></td>
<td><strong>65.97</strong></td>
<td><strong>66.49</strong></td>
<td><strong>57.95</strong></td>
<td><strong>59.01</strong></td>
</tr>
</tbody>
</table>

Table 1: Standard accuracy (evaluated on clean data) and robust accuracy (against attacks) on CIFAR-10. Specifically, 'Clean' denotes the accuracy on clean data. 'Score1 / Score2' represents the accuracy against $\ell_2$ and $\ell_\infty$, respectively. For the remaining cases, 'Score' signifies the accuracy against $\ell_\infty$ attacks.
Table 2: Standard accuracy and robust accuracy against \( \ell_2 \) and \( \ell_\infty \) attack on CIFAR-100.

<table>
<thead>
<tr>
<th>VGG-16</th>
<th>PGD</th>
<th>AutoPGD</th>
<th>EOT-PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>47.05</td>
<td>32.93 / 30.28</td>
<td>32.67 / 23.76</td>
</tr>
<tr>
<td>PGDAT</td>
<td>TRADES</td>
<td>51.14</td>
<td>32.70 / 31.22</td>
</tr>
<tr>
<td>AWP</td>
<td>49.49</td>
<td>34.07 / 30.10</td>
<td>33.89 / 30.89</td>
</tr>
<tr>
<td>MARGIN</td>
<td>53.86</td>
<td>36.88 / 34.52</td>
<td>36.63 / 33.30</td>
</tr>
<tr>
<td>SCORE</td>
<td>51.39</td>
<td>35.51 / 32.42</td>
<td>35.35 / 32.00</td>
</tr>
<tr>
<td>FPCC (Ours)</td>
<td>59.87</td>
<td>39.74 / 38.69</td>
<td>39.98 / 38.54</td>
</tr>
<tr>
<td>RHS</td>
<td>67.76</td>
<td>11.69 / 10.10</td>
<td>10.82 / 09.76</td>
</tr>
<tr>
<td>RHS+AT</td>
<td>62.10</td>
<td>34.27 / 29.88</td>
<td>33.97 / 28.67</td>
</tr>
<tr>
<td>MMC</td>
<td>68.83</td>
<td>12.34 / 10.79</td>
<td>11.36 / 10.08</td>
</tr>
<tr>
<td>MMC+AT</td>
<td>52.67</td>
<td>30.28 / 26.37</td>
<td>29.78 / 25.46</td>
</tr>
<tr>
<td>PROC</td>
<td>63.46</td>
<td>13.36 / 12.68</td>
<td>11.04 / 12.83</td>
</tr>
<tr>
<td>FPCC (Ours)</td>
<td>59.87</td>
<td>39.74 / 38.69</td>
<td>39.98 / 38.54</td>
</tr>
</tbody>
</table>

Table 3: Accuracy of models trained with and without FPCC against white-box and black-box attacks. Both attack types employed the PGD (\( \ell_\infty \)) attack. The experiments utilized the CIFAR-10 dataset.

<table>
<thead>
<tr>
<th>Clean</th>
<th>White</th>
<th>Black</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>Base</td>
<td>93.21</td>
</tr>
<tr>
<td>FPCC</td>
<td>89.42</td>
<td>74.37</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>Base</td>
<td>93.77</td>
</tr>
<tr>
<td>FPCC</td>
<td>87.65</td>
<td>61.82</td>
</tr>
<tr>
<td>WRN-28-10</td>
<td>Base</td>
<td>94.54</td>
</tr>
<tr>
<td>FPCC</td>
<td>88.64</td>
<td>65.58</td>
</tr>
</tbody>
</table>

Table 4: Average \( \ell_2 \) norm of gradients of the loss with respect to input images. The experiments were conducted on CIFAR-10.

<table>
<thead>
<tr>
<th>Gradient (( \ell_2 ) Norm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
</tr>
<tr>
<td>ResNet-50</td>
</tr>
<tr>
<td>WRN-28-10</td>
</tr>
</tbody>
</table>

4.2 Comparison with SOTA Methods

Given that our method belongs to the realm of static defense, we selected multiple SOTA methods from this category for a fair comparison. These methods include adversarial training: PGDAT [Madry et al., 2017], TRADES [Zhang et al., 2019], AWP [Wu et al., 2020], MARGIN [Gowal et al., 2020], SCORE [Pang et al., 2022], and the methods relying on clean examples: RHS [Mustafa et al., 2019], MMC [Pang et al., 2019], PROC [Li et al., 2021]. The RHS and MMC can be combined with adversarial training in the original paper, denoted as RHS+AT and MMC+AT, respectively. Additionally, we conducted comparisons between our method and adversarial purification methods. Due to space limitations, detailed results are provided in the Supplementary Material.

Table 1 presents the accuracy against \( \ell_2 \) and \( \ell_\infty \) attacks on CIFAR-10. The proposed FPCC method demonstrates higher standard accuracy compared to all adversarial training methods, and its robust accuracy surpasses methods relying solely on clean examples. For example, in WRN-28-10, the proposed FPCC achieves the highest accuracy of 65.58% and 57.95% against the PGD (\( \ell_\infty \)) attack and PIM attack, respectively. Similarly, in ResNet-50 against the PGD (\( \ell_2 \)) attack, FPCC achieves a robust accuracy 41.97% higher than RHS and surpasses the standard accuracy of PGDAT by 3.21%.

Table 2 illustrates the accuracy against \( \ell_2 \) and \( \ell_\infty \) attacks on CIFAR-100. The proposed FPCC method achieves the highest accuracy. For instance, the standard accuracy of FPCC improves by 4.47% compared to SCORE. Moreover, against AutoPGD (\( \ell_2 \)) attack, FPCC improves accuracy by 28.94% and 29.16% compared to PROC and RHS, respectively.

4.3 Gradient Obfuscation

Some studies [Athalye et al., 2018; Carlini et al., 2019; Tramer et al., 2020] have pointed out that the observed robustness improvements in many defense methods are due to gradient obfuscation, a phenomenon where the model’s gradients are assumed to be hidden and unknown [Athalye et al., 2018]. To determine whether a defense method is based on gradient obfuscation, various works have suggested and employed several sanity checks, as outlined in [Athalye et al., 2018; Carlini et al., 2019], yet many methods have failed these tests. Key checks include: 1. Robustness against single-step attacks should be superior to that against iterative attacks. 2. Robustness against black-box attacks should be better than against white-box attacks.

We have demonstrated that our method passes these sanity checks. By comparing Tables 1 and 2, it is evident that our method exhibits better adversarial robustness against FGSM (a single-step attack) compared to PGD (\( \ell_\infty \) and \( \ell_2 \), a iterative attack). Table 3 further shows that our method performs better under black-box attacks compared to white-box attacks.

Gradient vanishing is another form of gradient obfuscation. In Table 4, we assessed whether our proposed method suffers from gradient vanishing. It is observable that in our method, the \( \ell_2 \) norm of the gradients of the input image is not zero, indicating the absence of gradient vanishing.

4.4 Ablation Studies

Effectiveness of SFM, CFS, and PRO. We first conducted ablation studies on SFM, CFS, and PRO, as shown in Table 5. It is observed that the robust accuracy did not improve when the model was solely trained with SFM, CFS, or even PRO (indices ‘2’, ‘3’, and ‘4’). However, the proposed FPCC achieved a robust accuracy of 74.37% (index ‘5’). The reason is that the latent features without modification (SFM) or selection (CFS) inherently possess the ability to uphold correct feature patterns, implying that there is no enhancement of features through the PRO.

In index ‘6’, SFM in FPCC was replaced with adversarial examples, resulting in a robust accuracy of 46.17%, while the standard accuracy dropped to 82.21%. This decline is attributed to the model trained with adversarial examples becoming overfitted to adversarial perturbations. In index ‘7’, CFS in FPCC was replaced with Dropout. The robust accuracy of the model approached zero, as Dropout scales the selected features, causing PRO to constrain the scaled features rather than the original features. Center loss [Wen et al., 2016] is widely used in adversarial defense for reducing
Impact of the Positions and Quantities of CFS and PRO. The blue lines and red lines represent the cumulative insertion of CFS and PRO into the network, either from the shallow to the deep layer or from the deep to the shallow layer, respectively. Experiments were conducted on the VGG-16 using the CIFAR-10, and robust accuracy was measured against the PGD ($\ell_2$) attack.

Experiments were conducted on the VGG-16 using the CIFAR-10 dataset. Robust accuracy was measured against the PGD ($\ell_2$) attack. In index ‘8’, PRO in FPCC was replaced with center loss, leading to a robust accuracy of 21.79%. The improved robustness can be attributed to center loss promoting cohesiveness among intra-class features. However, it is less effective than PRO, as PRO constrains features to uphold correct feature patterns, aligning more closely with the observed behavior in correct predictions. Moreover, aligning features closer to a feature center does not necessarily align with the model’s prediction logic.

Table 5: Ablation studies for SFM, CFS, and PRO. ‘SA’ and ‘RA’ denote the standard accuracy evaluated on clean data and robust accuracy evaluated against the PGD ($\ell_2$) attack, respectively. ‘+Adv.’ denotes adversarial examples produced via the FGSM attack. ‘Drop.’ and ‘Cent.’ correspond to Dropout and center loss, respectively. All experiments were conducted on the VGG-16 architecture using the CIFAR-10 dataset.

Impact of the Proportion of Features Selected in CFS. Fig. 4 illustrates the impact of the proportion of features selected in CFS on accuracy. As observed, when the proportion of selected features decreases from 100% to 80%, the robust accuracy of the model initially increases to its peak, while the standard accuracy experiences a slight decline. This is because when 100% of the features are selected, all of them inherently possess the ability to uphold correct feature patterns, implying that there is no enhancement of features. Furthermore, when the proportion of selected features falls below 50%, both the robust and standard accuracies of the model decline sharply. This decrease can be attributed to the increased difficulty in upholding correct feature patterns with a smaller selection of features.

5 Conclusion

This paper diverges from most adversarial defense methods that focus on perturbations, directing attention instead to the clean examples. We introduce a novel and effective Feature Pattern Consistency Constraint (FPCC) method to strengthen the ability of features to uphold correct feature patterns. Experimental validation shows that the reinforced features can uphold correct patterns even when confronted with adversarial disturbances, enhancing the network’s intrinsic adversarial robustness beyond current SOTA methods. However, our approach has only been experimentally validated and lacks theoretical backing. Additionally, recent methods employing additional modules for adversarial purification, though not enhancing the model’s inherent robustness, effectively remove perturbations from adversarial examples and surpass our method in terms of robust accuracy. Hence, there is significant room for research in enhancing model adversarial robustness from the perspective of clean training, which will be the focus of our future work.
Acknowledgments
This work is supported by National Natural Science Foundation of China (U20B2066), Zhejiang Province High-Level Talents Special Support Program “Leading Talent of Technological Innovation of Ten-Thousands Talents Program” (No. 2022R52046), Zhejiang Province “LingYan” Research and Development Plan Project (No. 2024C01114).

References


