Computer Science and Information Systems 21(4):1499–1522 https://doi.org/10.2298/CSIS230907045J

Semantic Feature-Based Test Selection for Deep Neural Networks: A Frequency Domain Perspective

Zhouxian Jiang[1], Honghui Li[1], Xuetao Tian[2], and Rui Wang[1,⋆]

[1] School of Computing and Information Technology, Beijing Jiaotong University, Beijing, China
zhouxianjiang@bjtu.edu.cn
hhli@bjtu.edu.cn
rui.wang@bjtu.edu.cn
[2] Faculty of Psychology, Beijing Normal University, Beijing, China
xttian@bnu.edu.cn

Abstract. While deep neural networks (DNNs) have great potential for applications in security and safety-critical domains, their limited robustness to adversarial samples and out-of-distribution (OOD) samples raise significant concerns. In the software engineering community, significant efforts have been devoted to devising testing techniques that verify the robustness of DNNs. This paper investigates semantic feature-based test selection for DNNs from a frequency domain perspective and propose a novel method called SaFeTS. Specifically, we leverage saliency detection techniques, such as Fourier Phase Transform to extract semantic features from test cases. These features are then clustered to select diverse test cases to evaluate the robustness of DNNs and model retraining. Experiments on CIFAR-10 and SVHN datasets demonstrate that SaFeTS exposes more varied model errors compared to baseline methods. Further, retraining with SaFeTS-selected samples significantly improves adversarial and out-of-distribution robustness over state-of-the-art test selection methods.

Keywords: DNN testing, test selection, semantic feature, frequency domain, robustness.

1. Introduction

Deep learning (DL) systems are susceptible to adversarial samples [48] by malicious attacks, as well as out-of-distribution (OOD) samples [1] originating from natural environments. These vulnerabilities present potential security and safety threats, raising concerns about the application of DL technologies in security and safety-critical areas [5, 8, 38, 43, 44]. Researchers in the software engineering community have focused on verifying and improving the robustness of DL systems through testing to mitigate these threats. However, a pivotal element of a DL system is the deep neural network (DNN), which fundamentally differs from conventional software in several vital aspects: programming paradigms, source of faults, and repair approach. Conventional software is often developed based on specific requirements, while DNNs predominantly operate within a data-driven programming paradigm, making decisions through non-deterministic logic

⋆ Corresponding Author

[3]. Therefore, the faults in conventional software often stem from specific functions or lines of code. In contrast, the faults in DNNs typically manifest as deviations in model behavior, often linked to the data, learning algorithm, or training process. In addition, addressing faults in conventional software often involves pinpointing and correcting faulty code [9]. However, repairing faults in DNNs is more intricate, as they cannot be localized to a specific component (e.g., neuron or layer in the model) for direct fixing. A common approach to enhance the robustness of DNN is to expand the data set to retrain the model.

Early works have proposed structural adequacy criteria for DNN testing [35, 40], as well as test methods based on these criteria [17, 36, 40, 46, 56]. Recent efforts have extended test prioritization or selection methods to reduce computational and labeling costs. These methods are based primarily on specific rules, such as uncertainty [14, 55] and gradient-based rules [53]. However, existing test selection methods often focus on boosting the quantity of the model errors disclosed during the test rather than the diversity of misbehavior [15]. This leads to inadequate coverage of model erroneous behavior that potentially exists in real-world scenarios. Additionally, some work shows that existing test selection methods have a minimal impact on improving DNN's robustness [34, 53].

To mitigate these issues, we explore an alternative strategy to enhance the test selection for DNNs. Given the data-driven nature of DNNs, we aim to increase the diversity of test cases, not only due to its results in software testing [6, 13, 21, 32], but also due to the call for many work in DNN testing [15,31,61,62]. Essentially, relying on diverse test cases should expend the exploration scale of sample space and thus improve fault detection rate for a given test set [2]. Further, incorporating diverse samples in the retraining/repair process helps the model to learn diverse data representations, thereby improving the model's generalization ability and robustness on unknown data.

Current methods to assess the diversity of test cases often adopt distance measures, which overlooks semantic diversity. Here, the *Semantics* refers to the meaning of data, which is the essential attribute used to distinguish the differences among data. Instead, Chen et al. [11] demonstrate that the depth at which a model understands or learns from the data's semantics is strongly linked to its generalization capabilities. In turn, we can exploit the semantic diversity of the data for testing to reveal more model faults. Additionally, researchers [11, 18, 41, 60] found that DNNs lack robustness because they over-fitted to noisy high-frequency information and ignored semantic information during training. Suppressing high-frequency noise in the data will highlight the semantics of the data. Thus, in this paper, we utilize the frequency domain transform method to extract the semantics of the data and design the test selection method to improve the diversity of the selected data. The noise of the data is usually added in the spatial domain and can be filtered out by some frequency domain methods. For example, in the field of image processing, frequency domain based methods can help remove unimportant high frequency components that are often invisible to the human eye but contain noise [52]. Frequency domain analysis can further enhance features in certain frequency ranges, making them easier to recognize in subsequent processing. Thus, frequency domain-based methods can extract features that are semantically consistent with human intuition and avoid the impact of data noise [11]. Besides, frequency domain-based methods leverage the prior knowledge of natural images, and therefore are model-agnostic [50]. This characteristic helps circumvent the biases introduced by neural network feature extractors.

In this paper, we propose a test selection method for DNNs from a frequency domain perspective. We empirically investigated the effectiveness of four commonly used frequency domain-based saliency detection methods in extracting semantic features. The evaluation results demonstrate that these methods can effectively extract semantic features even from perturbed data. Motivated by this observation, we proposed a novel Semantic Feature-based Test Selection method (SaFeTS). Specifically, we extract the semantic features of test cases using frequency domain-based saliency detection methods. Subsequently, we leverage these semantic features to identify potentially diverse categories of samples. Then, we select samples from each category for testing and retraining purposes. Our work innovatively introduces the frequency domain analysis method into the test selection technique, focuses on the diversity of semantic information of test cases, and provides a new way of thinking for DNN testing. Extensive experimental results validate the effectiveness of the proposed SaFeTS. Compared to the baseline methods, we disclose more diverse DNN misbehavior with the proposed method. Moreover, SaFeTS can effectively improve the adversarial and OOD robustness of DNN via retraining. The main contributions of this paper are as follows.

• We explored test selection for DNNs from a frequency domain perspective. Through a comprehensive empirical study, we observed that the frequency domain-based saliency detection methods can effectively extract semantic features consistent with human intuition, even for perturbed data.

• We proposed a novel semantic feature-based test selection method SaFeTS. SaFeTS leverages saliency detection methods: Fourier Phase Transform (FPT) [39], High-pass Filter (HPF) [27], Residual Spectrum (SR) [23], and Quaternion Fourier Transform (QFT) [16], to extract semantic features from test cases and identify diverse categories of samples based on these features. Then, we select samples from each category for testing and retraining purposes.

• We conducted extensive experimental evaluations of SaFeTS. Compared to baselines, SaFeTS can detect more diverse DNN misbehavior. By selecting samples with diverse semantics for retraining, SaFeTS can effectively improve the adversarial and OOD robustness of DNNs. In particular, SaFeTS utilizes the FPT to extract semantic features, resulting in an average performance improvement of 20.79% over the best-performing baseline on adversarial samples. We make our code public at GitHub.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we outline the empirical study to validate the efficacy of frequency domain-based methods. In Section 4, we present our proposed method, SaFeTS. Finally, we demonstrate the evaluation results and summarize this work in Section 5 and 8, respectively.

2. Related Work

This section presents an overview of the related work, encompassing test selection for DNNs, alongside the frequency domain analysis applied to image processing.

2.1. Testing for DNN

Researchers in the software engineering community have designed testing methods for DNN to mitigate security and safety threats to DL-based systems. Inspired by structural

coverage criteria in conventional software testing, Pei et al. [40] first proposed the neuron coverage criteria for DNN testing and a white-box testing framework DeepXplore. Subsequently, Ma et al. [35] extended their work by offering multi-granularity testing criteria. Based on these test criteria, various testing methods have been proposed to generate test cases to discover the misbehavior of DNNs and verify their robustness, such as differential testing [40], fuzzing testing [17], and mutation testing [36], and symbolic testing [46, 47] and metamorphic testing [49, 56].

Due to the data-driven nature of DNN, comprehensive model testing requires large-scale test cases, resulting in a vast testing overhead. Therefore, some researchers investigated test prioritization or selection techniques to reduce the cost of the testing. Feng et al. [14] proposed DeepGini, a test prioritization method, to select test cases according to the confidence level calculated by the Gini index. Wang et al. [53] introduced a robustness-oriented testing framework for DNN, which used gradient-based rules as test prioritization. Inspired by active learning, Weiss et al. [55] used the uncertainty metrics as test prioritization. In addition, some test selection methods based on mutation testing [54] and differential testing [15] were proposed.

Current test methods focus on selecting the test cases that are more prone to trigger model misbehavior, therefore to boost the quantity of the identified software errors [15]. However, Zhang et al. [61] pointed out that using the error number as an evaluation metric to measure testing effectiveness is inadequate for DNNs. It is worth taking efforts to generate or select test cases with more diversity to expose various model incorrect behavior that potentially exists in real-world scenarios. Additionally, the test cases generated or selected by current test approaches have limited effect on improving the robustness of the DL system after testing [53]. These limitations call for improved test selection techniques to increase the diversity of selected test cases and further enhance the robustness of DNNs. To address the above limitations, we propose SaFeTS for test selection, aiming at selecting more semantically diverse test cases. These test cases with semantic diversity can expose diverse DNN misbehavior during testing and improve the robustness of DNNs after retraining.

2.2. Frequency Domain Analysis

Frequency domain analysis is one of the core techniques in the field of signal processing, which mainly involves the study of the frequency components of signals [7]. There are a wide range of techniques and applications of frequency domain analysis from time sequence data to images. The Fourier transform is the basis of frequency domain analysis, which converts a signal in the time domain or spatial domain to its frequency domain representation. For images, a two-dimensional FT can be used to obtain its frequency domain representation.

Frequency domain analysis has been widely used in conventional image processing, especially for saliency detection [16, 23, 27, 33, 39]. These methods often convert spatial-domain images to frequency-domain information through the Fourier transform. They are often used to extract the semantic features of images. Although DL-based feature extractors have been well developed, frequency domain-based methods still play a vital role in improving the robustness of DNNs [11]. Recently, some work investigated the robustness generalization behaviors of DNN from a data frequency domain perspective [11, 24, 45, 51, 57, 59]. In particular, Wang et al. [51] empirically found that the invis-

ible noises in high-frequency components of input images lead DNNs to make counter-intuitive predictions. Chen et al. [11] argued that a robust DNN should focus on learning the semantic information of data rather than high-frequency noise. Hence, they trained the models on the augmented data by combining the images' phase and amplitude spectrum. Frequency domain analysis methods can separate different levels of information, such as high frequency and low frequency, phase and amplitude. The frequency domain based saliency detection method utilizes the information of different levels and can effectively extract the semantic features of the data to avoid interference from data noise.

Thus, motivated by the strong ability of the frequency domain-based saliency detection methods, we explore the effective test selection for DNNs from a frequency domain perspective in this paper. We utilize these saliency detection methods to extract data semantic features. Furthermore, these semantic features provide valuable guidance in selecting diverse and representative test cases for DNN testing and retraining.

3. Empirical Study

In this section, we conducted an empirical study to validate the effectiveness of frequency domain-based saliency detection methods involving four classic saliency detection methods. We first briefly introduce the fundamentals of saliency detection methods. Then, we applied different perturbation techniques on test images of ImageNet [12], including adversarial attack and corruption techniques, to create adversarial and OOD samples, respectively. Finally, we evaluate the saliency detection methods by comparing the semantic maps.

3.1. Frequency Domain-based Semantic Feature Extraction

In image processing, the Discrete Fourier Transform (DFT) is commonly used to convert images from the spatial domain to the frequency domain. Research has found that frequency domain information can reveal semantic features of data, which are consistent with human perception (e.g., structures, edges, and textures) [27, 39]. In the empirical study, we investigated four classic frequency domain-based saliency detection methods: FPT [39], HPF [27], SR [23] and QFT [16], which have different characteristics in image semantic feature detection.

• **FPT** [39]: When an image is transformed using the Fourier transform, it is divided into frequency components consisting of amplitude and phase. The phase spectrum is a representation of the phase information in the frequency domain of an image. Reflects the relative positional relationship between different frequency components in the image, which is often adopted to detect structural and textural changes in the image.

• **HPF** [27]: A high-pass filter aims to sharpen edges or highlight fine details by attenuating low-frequency information. It typically corresponds to the smooth regions or large-scale structures in the image, which helps to highlight the edges and contours of objects.

• **SR** [23]: The spectrum residual is based on the observation that the human visual system tends to suppress the response to frequently occurring features, while at the same time keeping sensitive to features that deviate from the norm [29]. The spectrum residual method takes the logarithmic spectrum of an input image, followed by a Gaussian blur

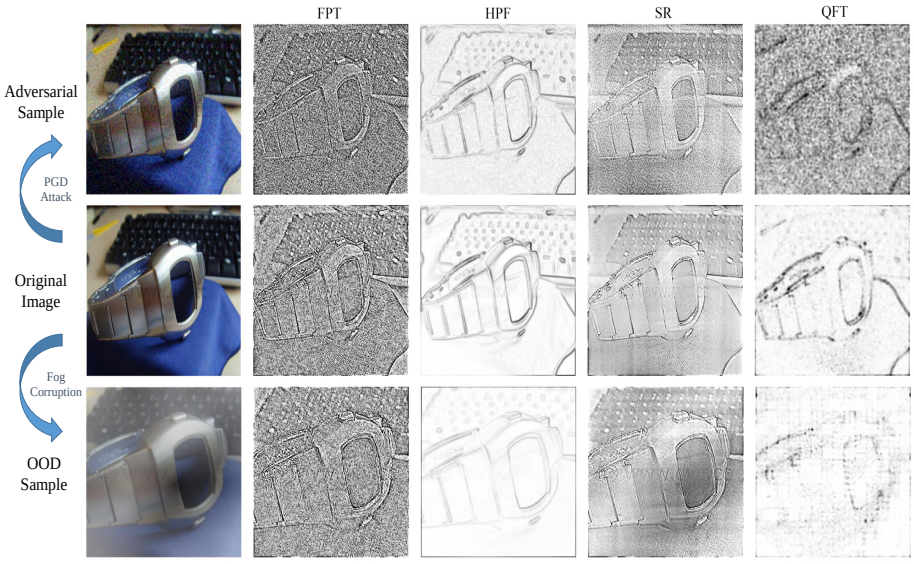


Fig. 1. Frequency domain-based semantic features of the images under different saliency detection methods. The columns from left to right are the original spatial-domain images, and the semantic maps extracted by FPT, HPF, SR, and QFT, respectively

operation. Then, the inverse Fourier transform is applied to the residual of the blurred and original log-spectrum, to obtain the saliency map.

• **QFT** [16]: QFT is an extension of the traditional Fourier transform, designed to work with quaternion-valued functions or data. QFT allows analyzing and representing color images and other multicomponent data using quaternion mathematics, simultaneously capturing an image's spatial and color information.

We applied the aforementioned methods on 1,000 images from dataset ImageNet [12]. Due to the limited space of the paper, we present one example of the images in Fig. 1. The remaining results are shown at GitHub. We observed from the second row in Fig. 1 that all four saliency detection methods preserved the semantic information of the image, such as the shape and contour of the watch. We re-verified that the extracted frequency domain-based semantic features provide a representation consistent with human perception. The experimental results confirm the feasibility of analyzing test samples from the frequency domain perspective.

3.2. Semantic Feature Extraction on Perturbed Images

Our further investigations probed whether four saliency detection methods could preserve semantic feature extraction performance on perturbed data. We utilized a pre-trained VGG16 [42] model on ImageNet [12] to generate adversarial samples by applying a gradient-based adversarial attack technique, PGD [37] and CW [10]. For crafting OOD samples, we adopted a widely used corrupted data benchmark [22], which encompasses

19 types of corruption involving weather, blur, noise, and digital changes with damage levels ranging from 1 to 5.

The first and third rows of Fig. 1 show the semantic maps on adversarial and OOD samples, respectively. Due to the space limitation, we only offer the cases of perturbations caused by PGD attacks and fog corruption. More case studies are shown at GitHub. Although both PGD attacks and fog corruption may diminish the semantic clarity of the original sample, all four methods still managed to extract semantic information from adversarial and OOD samples. In particular, FPT outperforms the other three saliency detection techniques, while QFT exhibits noticeable degradation in semantic maps.

In summary, in this section, we investigate the changes in semantic maps of adversarial and OOD samples. We found that saliency detection methods can effectively extract semantic information from perturbed samples. Inspired by these observations, we design a frequency domain-based test selection method to increase the semantic diversity of test cases for DNN model testing and robustness enhancement via retraining.

4. Approaches

In this section, we propose the SaFeTS test selection method to detect more diverse model erroneous behaviors and improve robustness via model retraining. First, we introduce the problem definition in deep learning. Then, we present the SaFeTS method in detail.

4.1. Problem Definition

The *Robustness* of deep learning model often consists of *Global Robustness* and *Empirical Robustness* [53]. Global robustness refers to the robustness of a deep learning model to disturbances or noise throughout the input space. Empirical robustness is evaluated by testing the model on specific perturbed datasets. In this paper, we study the robustness of deep learning restricted to empirical robustness, involving evaluating adversarial and OOD data. The empirical robustness is defined as follows.

Definition 1. (Empirical Robustness) Given a DNN model $\mathcal{F} : X \rightarrow Y$ and a perturbed test dataset D_t, where X and Y represent the input and output of $\mathcal{F}$, respectively. Empirical robustness is defined as $\mu : (\mathcal{F}, D_t) \rightarrow \gamma$, and $\gamma \in [0, 1]$ is the accuracy of model $\mathcal{F}$ on the test set D_t.

4.2. Semantic Feature-Based Test Selection

In this section, we introduce the proposed test selection method SaFeTS. The critical issue in test selection is how to utilize the semantics of test cases to expose the diverse misbehavior of DNN and use them to improve DNN's robustness. We design a frequency domain-based test selection method for DNNs. Fig. 2 shows the workflow of SaFeTS, and Algorithm 1 details the process of test selection based on SaFeTS. In general, SaFeTS selects test cases based on a given candidate test set through two steps: (1) semantic feature extraction and (2) clustering and sampling. Specifically, we use frequency domain-based saliency detection methods to extract the semantic features of test cases (line 4). Such methods are model-agnostic, which do not incorporate feature extractor bias. They extract semantic information while filtering out imperceptible high-frequency noise. After

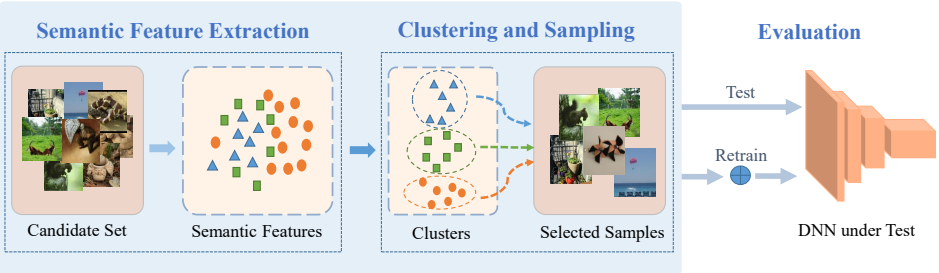


Fig. 2. The workflow of SaFeTS. SaFeTS includes two components: semantic feature extraction, clustering, and sampling. We evaluate the proposed method by testing and retraining DNN models on the selected samples

that, we adopt the clustering method to mine the semantically diverse categories of samples based on the extracted frequency-domain features (line 8). Finally, we sample from each cluster (lines 9-16), and use the sampled test cases to test and retrain DNNs.

Algorithm 1 Semantic Feature-Based Test Selection (SaFeTS)

Require: candidate set X, sampling ratio r, semantic feature set S_x, indices of sampled test cases $indicies$.
Ensure: selected set X_s.
1: $X_s \leftarrow \emptyset$, $S_x \leftarrow \emptyset$, $indices \leftarrow \emptyset$
2: **for** each $x \in X$ **do**
3: # Extract the semantic features based on frequency domain transformation methods.
4: $s \leftarrow Semantic\_extraction(x)$
5: S_x.add(s)
6: **end for**
7: # Cluster semantic feature to C categories.
8: $C \leftarrow Cluster\_semantic(S_x)$
9: **for** each $c \in C$ **do**
10: # Uniform sampling with ratio r from each cluster, get the indices of sampled test cases.
11: $indices \leftarrow indices \cup Uniform\_Sampling(c, r)$
12: **end for**
13: **for** each $i \in indicies$ **do**
14: # Select the samples from the candidate set.
15: $X_s \leftarrow X_s \cup X[i]$
16: **end for**
17: **return** X_s

Semantic feature extraction To extract the semantic features of the test cases, we transform the input image from the spatial domain to the frequency domain using a Fourier transform. Given an image g matrix with M rows and N columns, $f(m, n)$ denotes the pixel value of the m_{th} row and n_{th} column, the Fourier transform $F(u, v)$ of $f(m, n)$ is

shown in Equation (1).

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \cdot e^{-j2\pi(um/M+vn/N)} \tag{1}$$

For an image g matrix with M rows and N columns, the Fourier transform theoretically requires $(M * N)^2$ operations, which is very time-consuming. Alternatively, we use the Fast Fourier Transform (FFT) to reduce the complexity from $O((M * N)^2)$ to $O(MNlog(M * N))$ by decomposing the DFT into a series of smaller DFTs. In contrast, we can transform an input image from its frequency domain to spatial domain via Inverse Fast Fourier transform (IFFT), shown in Equation (2).

$$f(m,n) = \frac{1}{\sqrt{MN}} \sum_{v=0}^{M-1} \sum_{u=0}^{N-1} F(u,v) \cdot e^{2\pi((vm/M)+(un/N))} \tag{2}$$

Based on FFT and IFFT, we employ four classic saliency detection methods mentioned in Section 3 to extract the semantic information of the test cases. The details of the calculations are as follows:

• **FPT:** The phase spectrum of an image is the phase portion of the Fourier transform result, shown in Equation (3).

$$P(u,v) = \arctan\left(\frac{Im[F(u,v)]}{Re[F(u,v)]}\right) \tag{3}$$

Where Im$[F(u,v)]$ and Re$[F(u,v)]$ are the imaginary and real parts of $F(u,v)$, respectively. Then, the IFFT of P leads to a saliency map $S_a(m,n)$.

• **HPF:** The HPF are mainly used to enhance the high-frequency components of an image so as to highlight the edges and details. In this paper, we use Sobel filter to compute HPF which is a discrete difference filter used for edge detection to highlight the edges by computing the gradient strength of the image. Sobel filter contains two $3 * 3$ matrices which is shown in Equation (4) for detecting the edges in horizontal G_x and vertical direction G_y, respectively.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

For each pixel in the image, its gradient values in the horizontal V_x and vertical directions V_y are calculated by the convolution operation in Equation (5).

$$V_x(m,n) = f(m,n) * G_x, V_y(m,n) = f(m,n) * G_y \tag{5}$$

Then calculate the total gradient intensity, i.e., saliency map Sa, by Equation (6).

$$S_a(m,n) = \sqrt{V_x(m,n)^2 + V_y(m,n)^2} \tag{6}$$

After processing, a new image is obtained which emphasizes the edge regions in the original image and can be considered as the result of HPF process.

• **SR:** The SR method estimates significance from the residuals of the amplitude spectrum of the image. The amplitude spectrum $A(u,v) = |F(u,v)|$ and the phase spectrum $P(u,v)$ are computed after performing FFT on a given image. Then, calculate the average amplitude spectrum $\log(A(u,v))$ and its smoothed version $Sm(u,v)$. spectrum residual is computed by $R(u,v) = \log(A(u,v)) - Sm(u,v)$. The Fourier transform of the saliency map is $S_a(u,v) = e^{R(u,v)} \times e^{jP(u,v)}$. Finally, the IFFT is used to obtain the saliency map $S_a(m,n)$.

• **QFT:** QFT-based saliency detection suggests that in the human visual system, the four feature channels, motion feature M, luminance feature I, red-green neuron RG, and blue-yellow neuron BY, are almost independent. Therefore, the above four features of an image can be represented by a quaternion $q(m,n)$, defining the weighted image quaternion representation as follow:

$$q(m,n) = M(m,n) + RG(m,n) * \mu_1 + BY(m,n) * \mu_2 + I(m,n) * \mu_3 \tag{7}$$

where $\mu_i, i = 1,2,3$ satisfies $\mu_i^2 = -1$, $\mu_1 \perp \mu_2$, $\mu_2 \perp \mu_3$, $\mu_1 \perp \mu_3$, $\mu_3 = \mu_1\mu_2$. The QFT transforms an image from the spatial domain to the frequency domain as follows:

$$Q(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} q(m,n) \cdot e^{-j2\pi(um/M + vn/N)} \tag{8}$$

Here we actually extend the concept of 2D FFT to accommodate quaternionic data. Then, in the frequency domain, saliency detection is performed based on the phase information of Q (u, v), that is, $S_a(u,v) = |Q(u,v)|$. And finally, the inverse QFT of $S_a(u,v)$ leads to a saliency map $S_a(m,n)$ (extensive the concept of 2D IFFT to accommodate quaternionic data).

Clustering and sampling To group test cases into multiple categories with similar semantic features, we incorporate clustering approach in SaFeTS. In this paper, we extract the semantic feature of image samples and cluster them into various categories in the frequency domain feature space. Due to the convergence speed and cluster performance, we employ the Kmeans++ algorithm [4] as the clustering method based on semantic features to divide all samples into k clusters, where k is the number of groups. Kmeans++ algorithm is an optimized version of the Kmeans clustering algorithm [19], where the initial centroids are selected through a specific randomization process to reduce the effect of randomization. It randomly selects the first centroid and then uses weighted random sampling to choose the remaining centroids.

After that, we adopted the Uniform Sampling (US) strategy to maximize the diversity of the sampled test cases. The US strategy ensures that the samples selected from each cluster are uniformly distributed, which helps to increase the diversity and coverage of the samples. In addition, since the uniform sampling strategy does not bias the selection of samples towards any particular region, it provides a wider and diverse set of test cases, which is important for deep neural network testing. In summary, the Kmeans++ and US strategies can be easily scaled or adapted to different datasets and application scenarios, and provide an efficient, high-quality and low-bias strategy for DNN testing that ensures sample diversity and broad coverage.

Retraining objective The samples selected based on SaFeTS can be used not only to test the DNN but also to enhance the model robustness via retraining. Training samples consist of the original training set D and the sample selected from the perturbed test set D_t based on the test selection method TS. We define the optimization objective for robustness retraining as

$$\min_{\theta} \frac{1}{L} \sum_{i=1}^{L} J_{(x_i,y_i)\in D\cup TS(D_t)}(\theta, x_i, y_i), \tag{9}$$

where L is the number of training samples, x, y is the training samples and their labels, and θ are the parameters of the model $\mathcal{F}$. Retraining aims to optimize the model parameters θ that lead to minimization of the loss function J.

In summary, we introduce SaFeTS, a test selection approach for DNNs. Employing frequency domain transformation techniques, we extract test case semantics and then apply clustering to discern distinct categories within the frequency domain feature space. SaFeTS is designed to select semantically diverse samples, uncovering a broader spectrum of DNN misbehaviors and enhancing DNN robustness through retraining.

5. Experimental Setup

In this section, we describe the experimental setup for evaluating SaFeTS, including the research questions, datasets and DNN models, evaluation methods, baselines and parameter analysis.

5.1. Research Questions

Our evaluation covers various datasets and DNN models. We benchmark SaFeTS against baseline methods with the intent to address the following three research questions (RQs).

- **RQ1.** Can SaFeTS expose more diverse erroneous behavior of DNNs?
- **RQ2.** Can SaFeTS effectively improve the adversarial robustness of DNNs via retraining?
- **RQ3.** Can SaFeTS effectively enhance the OOD robustness of DNNs via retraining?

5.2. Models and Datasets

In the experiment, to evaluate the effectiveness of testing and retraining, we used two datasets and six pre-trained models: CIFAR-10 dataset [30] (ResNet18 [20], VGG16 [42], DenseNet121 [25]) and SVHN dataset [58] (ResNet50 [20], Wide-ResNet50 [20] and SqueezeNet [26]). We applied the advanced adversarial attack technique PGD [37] to generate the candidate set for test selection. Detailed configurations of datasets, models, and attack results are reported in Table 1.

CIFAR-10 [30]: The CIFAR-10 dataset is a widely used benchmark for image classification tasks. It comprises 60,000 32x32 color images across 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training and 10,000 test images.

SVHN [58]: The SVHN dataset, or Street View House Numbers dataset, is another popular dataset for digit recognition in natural images. It contains over 600,000 digit

images obtained from real-world street view images. The dataset is divided into training, validation, and test sets, and it covers a wide range of digit variations and orientations.

ResNet18, ResNet50, and Wide-ResNet50 [20]: ResNet18 is a convolutional neural network architecture known for its effectiveness in deep learning tasks. It consists of 18 layers and introduces the concept of residual blocks, which helps address the vanishing gradient problem and enables the training of deep networks. ResNet50 is an extension of the ResNet architecture with 50 layers. It offers even greater model capacity and captures intricate features from images. Wide-ResNet50 is an extended version of ResNet50 with wider layers for better feature learning and regularization.

VGG16 [42]: VGG16 is a deep convolutional neural network architecture developed by the Visual Geometry Group at the University of Oxford. The architecture consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. VGG16's key feature is using small 3x3 convolutional filters in multiple layers to learn hierarchical features from images.

DenseNet121 [25]: DenseNet is a neural network architecture that introduces dense connections between layers. DenseNet121 is a variant with 121 layers, which includes dense blocks where each layer is connected to every other layer in a feed-forward manner.

SqueezeNet [26]: SqueezeNet is a compact neural network architecture designed for deep learning tasks on resource-constrained devices or environments. It achieves high accuracy with a significantly reduced number of parameters. The architecture employs various strategies such as 1x1 convolutions and 'fire' modules (combinations of squeeze and expand operations) to maintain model efficiency while retaining strong representational power.

Table 1. Detailed experiment configurations

Dataset	Model	Accuracy%	# Adversarial	#Error	Robustness%
CIFAR-10	ResNet18	95.35	10,000	8,953	10.47
	VGG16	93.88	10,000	9,358	6.42
	DenseNet121	94.78	10,000	8,990	10.10
SVHN	ResNet50	91.56	26,032	14,297	45.08
	Wide-ResNet50	91.69	26,032	14,435	44.55
	SqueezeNet	92.55	26,032	10,569	59.40

5.3. Evaluation methods

The evaluation methods we adopted in this section for testing and retraining are shown as below.

Diversity of misbehavior: To investigate the diversity of the misbehavior triggered by test cases, we visualized the confusion matrix of the prediction results. A confusion matrix is a table often used in classification task to illustrate the model performance on test data. In a confusion matrix, the rows represent the true labels, and the columns represent the predicted labels. Each entry in the matrix indicates the number of occurrences where a true label was predicted as a specific class. We removed the diagonal elements (correct

predictions) and then focused on the distribution of the non-diagonal elements (erroneous predictions). We evaluate the diversity of misbehavior by calculating the variance of the confusion matrix. The more uniformly distributed the misclassified samples, the smaller the variance of the confusion matrix, indicating higher diversity of the misclassified samples.

Accuracy on perturbed set (Robustness): The accuracy of the model on perturbed dataset. This metric is used to evaluate the effectiveness of SaFeTS in improving the robustness generalization of the model after retraining.

Accuracy on the original test set (Clean Accuracy): The model's accuracy on the original test data. This metric is used to evaluate the standard generalization of the model after retraining.

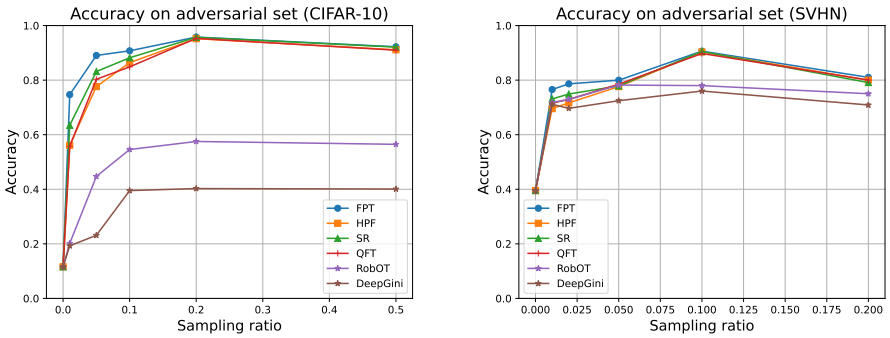
5.4. Baselines

In this paper, we use current state-of-the-art test selection techniques as baselines, including DeepGini [14] and RobOT [53]. We then evaluated these baseline methods on our proposed dataset.

DeepGini [14]: DeepGini is a test selection technique that focuses on selecting test cases based on confidence scores (computed by the Gini index) of the model predictions. It prioritizes test cases for which the model is lowly confident in its predictions. This technique explores the least confident predictions and potentially reveals the model misbehavior.

RobOT [53]: RobOT is a test selection technique based on the gradient information of the model. RobOT calculates the robustness of the test case on the basis of its gradient of the model's loss function. It prioritizes samples with large gradients, which have a higher likelihood of leading to incorrect model predictions.

5.5. Parameter Analysis



(a) CIFAR-10 **(b)** SVHN

Fig. 3. Accuracy on adversarial set with different sample ratios and test selection methods

This section analyzes the parameters of adversarial sample generation, as well as the retraining sample rate. For generating adversarial samples on CIFAR-10 and SVHN, we employ the Torchattacks library [28] to implement the advanced attack PGD [37]. For both the CIFAR-10 dataset and the SVHN dataset, we set the maximum perturbation $eps = 4/255$, and the number of steps $steps = 10$.

To choose a suitable sampling ratio for retraining, we set different sampling ratios for SVHN (ResNet50) and CIFAR-10 (ResNet18) and observed the accuracy changes on the adversarial set. Fig. 3 shows the accuracy on an adversarial set of re-trained models according to the change in the sampling ratio. We noticed that our method performs better on the CIFAR-10 and SVHN datasets than the baselines in improving robustness. However, the accuracy of the retrained model does not always increase as the sampling ratio increases. This phenomenon may be due to the excessive introduction of noisy data for training that instead destroys the performance of the model. In addition, increasing the number of training samples will increase the training overhead. Thus, we set sampling ratios of 0.2 and 0.1 for the CIFAR-10 and SVHN datasets to balance performance and time spent in the subsequent experiments.

In subsequent evaluations, to mitigate the impact of randomness in sampling, we ran the experiments five times for each configuration and then reported the mean values as the results of testing and retraining. Additionally, we carefully tuned other hyperparameters to achieve the best version of our method, such as the k-value in clustering. For other details please refer to our experimental code on GitHub.

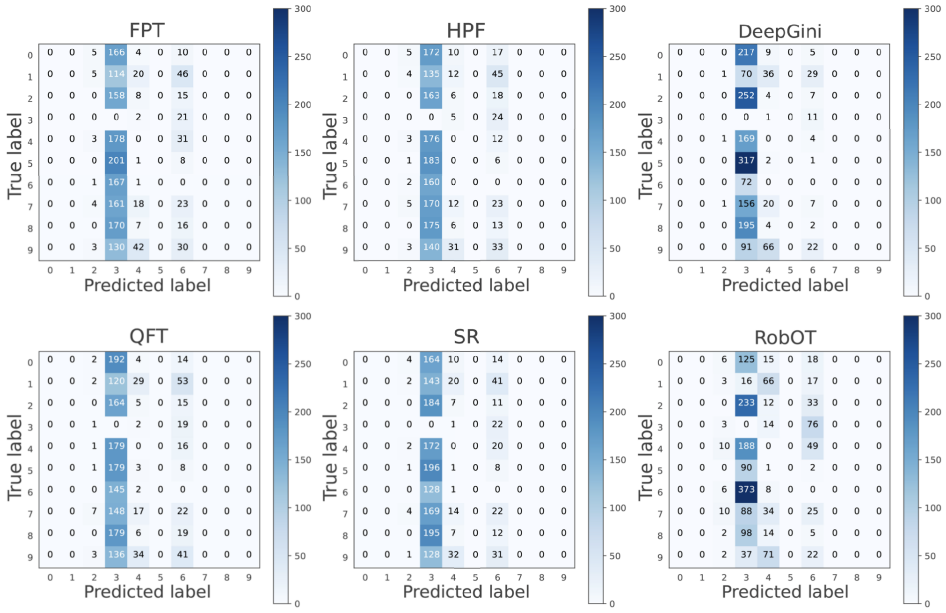


Fig. 4. The confusion matrices of test results on CIFAR-10

6. Analysis

This section reports the results of the evaluation based on three research questions mentioned in Sec. 5.1, including the evaluation on diversity of test cases, and the effect of robustness enhancement on adversarial and OOD samples.

6.1. RQ1: Evaluation of the Diversity of Selected Test Cases

We evaluated the effectiveness of SaFeTS for testing by investigating the ability of SaFeTS to expose diverse models' misbehavior. As shown in Table 1, we generated adversarial samples as the candidate set, using the PGD attack on the test set of CIFAR-10 and SVHN. Then, we adopted different test selection methods to select samples from the candidate set.

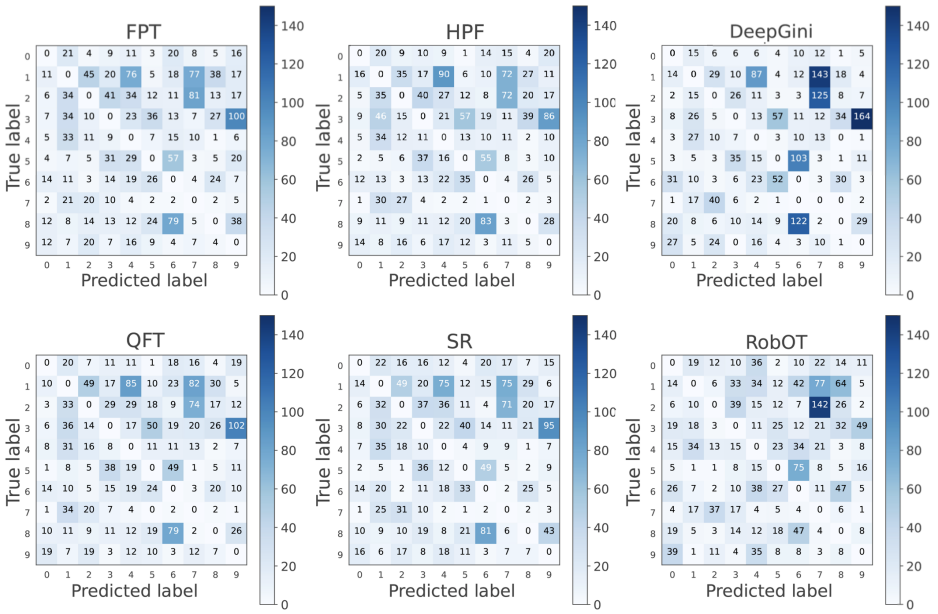
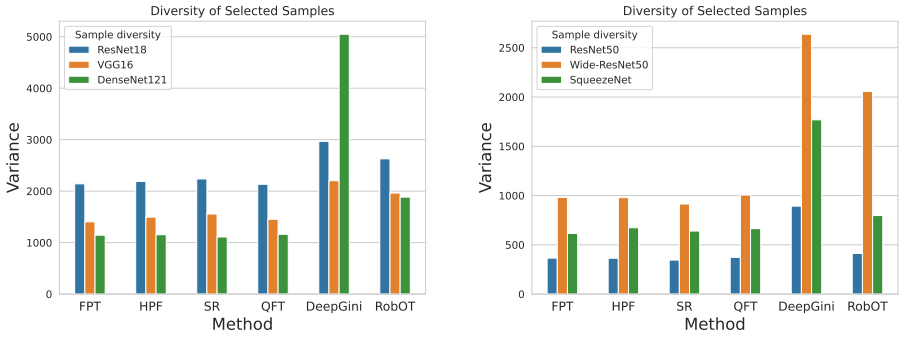


Fig. 5. The confusion matrices of test results on SVHN

Fig. 4 and Fig. 5 show the confusion matrices of the testing results on CIFAR-10 (ResNet18) and SVHN (ResNet50), respectively. According to the confusion matrices, FPT, HPF, SR, and QFT identify samples with a broader coverage of misclassifications, instead of an exclusive focus on particular errors. On the contrary, the baseline methods are heavily focused on certain predictive mistakes. In the case of CIFAR-10, DeepGini exhibits a bias towards samples where the prediction of class '5' as the class '3' occurs, while RobOT tends to emphasize instances where class '6' is misclassified as class '3'. Similarly, in SVHN, DeepGini prefers examples where class '3' is mispredicted as class



(a) CIFAR-10 **(b)** SVHN

Fig. 6. The variances of confusion matrices on different datasets

'9', while RobOT is inclined towards samples with class '2' being wrongly assigned to class '7'.

We calculated the variances associated with these confusion matrices to evaluate the model's incorrect behavior's diversity comprehensively. A lower variance in the confusion matrix indicates a more uniform distribution of misclassified samples and more diversity. Fig. 6 illustrates the variances across the confusion matrices for both CIFAR-10 (left) and SVHN (right). The variances of the samples selected by SaFeTS, namely, FPT, HPF, SR, QFT, indicate a consistently consistent trend and are smaller than those of DeepGini and RobOT. This observation underscores the enhanced diversity within SaFeTS's selection.

When we select test cases based on semantic features, we are in fact picking samples that are semantically different but could lead to classification errors. Such samples are highly diverse because they cover a wide range in sample space that could trigger error predicting. When using a frequency domain based approach, it is more easily to distinguish and capture these different semantic features. For example, HPF highlights high-frequency details, which helps to identify possible problems with the model focus in high-frequency area. Selecting samples with different semantic features through SaFeTS means that we evaluate the model's performance on multiple semantic types, rather than just focusing on a particular error pattern. Based on the experimental results, we can answer RQ1.

Answer to RQ1: Compared to baselines, SaFeTS can effectively select test cases that expose the misbehavior of more diverse models. SaFeTS mitigates the selection of samples too focused on a particular type of error and ensures the diversity and balance of the samples in subsequent retraining.

6.2. RQ2: Evaluation of the Adversarial Robustness

To evaluate SaFeTS in the adversarial robustness improvement of the model, we crafted a candidate set based on PGD attacks to generate 10 and 20 times more adversarial samples on the training sets of SVHN and CIFAR-10, respectively. Then we compared the robustness performance of baseline methods and SaFeTS by retraining on the select test cases.

We reported the robustness and accuracy of the retrained models with sampling ratios 0.2 (CIFAR-10) and 0.1 (SVHN) in Table 2. The results indicate that all four saliency detection methods outperform the baseline approach, with particular prominence observed in the performance of the FPT method. Specifically, on the CIFAR-10 dataset, the FPT, HPF, SR, and QFT methods enhance robustness by 31.21%, 30.76%, 30.96%, and 30.89%, respectively, compared to the best-performing baseline. Similarly, on the SVHN dataset, the FPT, HPF, SR, and QFT methods exhibit improvements in robustness by 10.89%, 10.66%, 10.54%, and 10.42%, respectively. According to the raw robustness reported in Table 1, after retraining, the robustness boosting on CIFAR-10 is better than that on SVHN, which is street door number data with less diversity relative to CIFAR-10. Therefore, SaFeTS performs better on more diverse datasets. In addition, the FPT works best among all the saliency detection methods, so in practice, to maximize the effectiveness of SaFeTS, more diverse data can be collected and FPT configurations can be prioritized.

In addition, SaFeTS leads to a moderate reduction in the accuracy of the retrained model, whereas retraining with DeepGini and RobOT-selected samples reduces the original accuracy of the model. For example, on CIFAR-10 and ResNet18, the FPT method reduces the model's accuracy on the original test set from 95.35% to 94.82%. The accuracy of the retrained model based on DeepGini and RobOT decreased to 70.89% and 74.49%, respectively. The experimental results illustrate that the model learns diverse semantic features by using the samples selected by SaFeTS for retraining. SaFeTS mitigates the overfitting of the perturbed data and maintains the performance of the original test set.

The experimental results demonstrate that SaFeTS can effectively distinguish and capture errors that the model may have in processing different semantic information, thus providing the model with more representative samples for retraining. Selecting samples with a high diversity is crucial for training the model, as they cover a wide range of sample space that may cause the model to make errors, which helps the model to have better generalization ability when faced with unknown adversarial samples. The frequency domain based approach selects a diverse set of samples, which reduces the risk of overfitting the model on specific types of perturbation data.

Based on the above experiments, we can answer RQ2:
Answer to RQ2: SaFeTS can effectively improve the adversarial robustness of models after retraining, especially using the FPT to extract the semantic features. FPT, HPF, SR, and QFT improved performance over the best-perform baseline by an average of 31.21%, 30.76%, 30.96%, and 30.89% on CIFAR-10, and by 10.89%, 10.66%, 10.54%, and 10.42% on SVHN, respectively.

6.3. RQ3: Evaluation of the Out-of-distribution Robustness

To verify whether SaFeTS can effectively work for improvements in OOD robustness, we investigated the performance of retraining on OOD datasets. We adopted a widely recognized corruption benchmarking [22] to generate four types of corruptions, including weather (fog), blur (Gaussian blur), noise (Gaussian noise), digital (pixelate), and five levels (level 1-5) of corrupted data on the CIFAR-10 training set for sampling. Afterward, we used the CIFAR-10-C [22] dataset to verify the robustness of generalization.

We utilized the best-perform parameters in RQ2, including using FPT to extract the semantic features of samples, setting the sampling ratio to 0.2 for each corruption type, and using K-means++ for clustering. We reported the retraining results on model ResNet18

Table 2. Accuracy on original test set (Clean Accuracy) and adversarial set (Robustness) of the retrained models

CIFAR-10	**ResNet18**		**VGG16**		**DenseNet121**	
Method	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**
FPT	95.06	**95.78 (85.31** ↑)	93.20	**93.17 (86.75** ↑)	94.06	**95.30 (85.20** ↑)
HPF	94.82	95.19 (84.72 ↑)	92.99	92.84 (86.42 ↑)	93.80	94.88 (84.78 ↑)
SR	94.73	95.69 (85.22 ↑)	92.85	92.93 (86.51 ↑)	93.72	94.88 (84.78 ↑)
QFT	94.84	95.35 (84.88 ↑)	92.68	93.00 (86.58 ↑)	93.76	94.94 (45.31 ↑)
DeepGini	70.89	40.26 (29.79 ↑)	71.45	50.25 (43.83 ↑)	77.21	55.41 (45.31 ↑)
RobOT	74.49	57.52 (47.05 ↑)	78.94	64.33 (57.91 ↑)	78.34	68.78 (58.68 ↑)
SVHN	**ResNet50**		**Wide-ResNet50**		**SqueezeNet**	
Method	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**
FPT	93.52	**90.63 (45.55** ↑)	93.21	**90.44 (45.89** ↑)	92.71	**89.23 (29.83** ↑)
HPF	92.83	90.32 (45.24 ↑)	92.98	90.25 (45.70 ↑)	93.04	89.05 (29.65 ↑)
SR	93.25	90.49 (45.41 ↑)	93.03	89.72 (45.17 ↑)	92.87	89.05 (29.65 ↑)
QFT	92.98	89.76 (44.68 ↑)	93.14	90.11 (45.56 ↑)	92.62	89.04 (29.64 ↑)
DeepGini	88.06	77.99 (32.91 ↑)	88.25	78.21 (33.66 ↑)	89.12	79.87 (20.47 ↑)
RobOT	91.89	76.01 (30.93 ↑)	91.58	79.64 (35.09 ↑)	92.15	80.01 (20.61 ↑)

Table 3. Accuracy on original test set (Clean Accuracy) and OOD set (Robustness) of the retrained models

Corruption	**Method**	**Clean Accuracy %**	**Robustness %**
Weather	SaFeTS(FPT)	97.35	**97.34**
	RobOT	98.49	94.32
	DeepGini	98.65	96.05
Blur	SaFeTS(FPT)	87.99	**95.44**
	RobOT	91.08	91.28
	DeepGini	92.09	94.19
Noise	SaFeTS(FPT)	96.51	**96.78**
	RobOT	96.51	95.34
	DeepGini	96.47	96.47
Digital	SaFeTS(FPT)	98.43	**95.01**
	RobOT	99.11	90.27
	DeepGini	99.07	91.47

in Table 3. The experimental results show that, for the weather, blur, noise, and digital corrupted data, SaFeTS has effectively improved the robust generalization compared to baselines, with a minor performance degradation on the original test set.

Unlike adversarial samples, OOD data are those that have not appeared in the training data, but are not intentionally designed to deceive the model. The model may be confused by specific noises or details in the data and miss the global characteristics. Choosing samples with diversity can help the model to better generalize to a wide variety of data, not just a specific subset of the training set. Experimental results show that SaFeTS can help models focus on more fundamental properties and avoid being distracted by unnecessary noise. By enhancing the diversity of the retraining samples, it can be ensured that the model has been trained in a variety of different contexts, thus improving its robustness in the face of unknown data. In summary, SaFeTS helps to enhance the robustness and generalization ability of the model on OOD data.

Based on the experimental results, we can answer RQ3: **Answer to RQ3:** SaFeTS can effectively improve OOD robustness of DNN models on various types of corruption data, the overall results indicate that our method can improve the performance over best-perform baseline.

7. Limitation

In this section, we discuss the limitations of SaFeTS. From the perspective of time overhead, SaFeTS depends on the convergence speed of the clustering algorithm. To explore the impact of clustering methods on the results, we also used a clustering algorithm different from Kmeans++, the GMM, which is a density-based clustering method. We reported the retraining results of CIFAR-10 (ResNet18) and SVHN (ResNet50) with two clustering methods in Table 4 (selecting samples from adversarial sets).

Table 4. The accuracy on an adversarial set of the retrained models (Robustness) and clustering time under different clustering methods (Kmeans++ and GMM)

Dataset	Method	Robustness %		Time (s)	
		Kmeans++	GMM	Kmeans++	GMM
CIFAR-10	PFT	**95.78**	95.42	**980.92**	4221.60
	HPF	95.19	**95.60**	**1417.65**	3008.59
	SR	**95.69**	95.41	**1480.64**	4558.23
	QFT	95.35	**95.88**	**2028.23**	4292.48
SVHN	PFT	**90.63**	90.07	**491.45**	2247.11
	HPF	**90.32**	90.20	**852.56**	3004.25
	SR	**90.49**	90.44	**845.57**	2457.21
	QFT	**89.76**	88.12	**1074.54**	2542.44

The experimental results show that the choice of clustering method impacts the performance of SaFeTS. Clustering with Kmeans++ performs better than GMM. The time

overhead of SaFeTS is more extensive compared to baseline methods. However, the training time for DNNs is much longer than the time overhead of test selection. Therefore, the time overhead of test selection can almost be ignored. In our retraining experiments, with the addition of 73,257 and 200,000 samples to SVHN and CIFAR-10, the time needed for each epoch is approximately 16 minutes and 20 minutes, respectively. And each training is run separately on a single 3090 GPU. Thus, it is worth trading more test selection time for better testing and retraining results. Nevertheless, to reduce the time overhead of sample selection, more efficient clustering algorithms or parallel computing can be attempted to speed up the process.

8. Conclusion

In summary, this research proposed SaFeTS, a novel semantic feature-based test selection approach for DNNs grounded in frequency domain analysis. The core innovation lies in utilizing saliency detection methods to extract semantic features aligned with human perception. SaFeTS then discerns diverse categories within this semantic feature space via clustering. By sampling test cases from each cluster, SaFeTS selects data with rich semantics for testing and retraining DNNs. Extensive experiments highlighted the ability of SaFeTS to uncover a more varied model misbehavior compared to existing selection techniques. Further, retraining on the semantically diverse samples enhanced adversarial and out-of-distribution robustness, with particularly strong gains of over 20% on adversarial accuracy. The proposed frequency domain viewpoint thus offers valuable new perspectives on testing and improving DNN robustness. Limitations like clustering overhead present opportunities for future work on optimizing SaFeTS. By releasing our method publicly, this research aims to provide an effective resource to advance rigorous engineering of trustworthy DNNs. In future work, we will aim to investigate the generalization of the SeFeTS approach to other domains besides image classification and computer vision, especially security-critical domains.

Acknowledgments. This work was supported by the Talent Fund of Beijing Jiaotong University, No. 2023XKRC041, and National Natural Science Foundation of China, No. 62273027, and Beijing Natural Science Foundation, No. L211021.

References

1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion 76, 243–297 (2021)
2. Aghababaeyan, Z., Abdellatif, M., Briand, L., S, R., Bagherzadeh, M.: Black-box testing of deep neural networks through test case diversity. IEEE Transactions on Software Engineering 49(5), 3182–3204 (2023)
3. Alves, E.E., Bhatt, D., Hall, B., Driscoll, K., Murugesan, A., Rushby, J.: Considerations in assuring safety of increasingly autonomous systems. Tech. rep. (2018)
4. Arthur, D., Vassilvitskii, S.: K-means++ the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035 (2007)
5. Bazarbash, M.: Fintech in financial inclusion: Machine learning applications in assessing credit risk. Tech. rep., International Monetary Fund (2019), technical report

6. Biagiola, M., Stocco, A., Ricca, F., Tonella, P.: Diversity-based web test generation. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 142–153 (2019)
7. Boashash, B.: Time-frequency signal analysis and processing: a comprehensive reference (2015)
8. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
9. Borandag, E., Ozcift, A., Kilinc, D., Yucalar, F.: Majority vote feature selection algorithm in software fault prediction. Computer Science and Information Systems 16(2), 515–539 (2019)
10. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (sp). pp. 39–57. IEEE (2017)
11. Chen, G., Peng, P., Ma, L., Li, J., Du, L., Tian, Y.: Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 458–467 (2021)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
13. Feldt, R., Poulding, S., Clark, D., Yoo, S.: Test set diameter: Quantifying the diversity of sets of test cases. In: 2016 IEEE 8th International Conference on Software Testing, verification and validation (ICST). pp. 223–233. IEEE (2016)
14. Feng, Y., Shi, Q., Gao, X., et al.: Deepgini: prioritizing massive tests to enhance the robustness of deep neural networks. In: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 177–188 (2020)
15. Gao, X., Feng, Y., Yin, Y., Liu, Z., Chen, Z., Xu, B.: Adaptive test selection for deep neural networks. In: Proceedings of the 44th International Conference on Software Engineering. pp. 73–85 (2022)
16. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
17. Guo, J., Jiang, Y., Zhao, Y., Chen, Q., Sun, J.: Dlfuzz: Differential fuzzing testing of deep learning systems. pp. 739–743 (2018)
18. Han, S., Lin, C., Shen, C., Wang, Q., Guan, X.: Interpreting adversarial examples in deep learning: A review. Association for Computing Machinery (2023), https://doi.org/10.1145/3594869
19. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) 28(1), 100–108 (1979)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
21. Hemmati, H., Fang, Z., Mantyla, M.V.: Prioritizing manual test cases in traditional and rapid release environments. In: 2015 IEEE 8th International Conference on Software Testing, verification and validation (ICST). pp. 1–10. IEEE (2015)
22. Hendrycks D, D.T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2020)
23. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: 2007 IEEE Conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)
24. Huang, B., Tao, C., Lin, R., Wong, N.: Frequency regularization for improving adversarial robustness. arXiv preprint arXiv:2212.12732 (2022)
25. Huang, G., Liu, Z., Van Der Maaten, L., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017)

26. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
27. Imamoglu, N., Lin, W., Fang, Y.: A saliency detection model using low-level features based on wavelet transform. IEEE transactions on multimedia 15(1), 96–105 (2012)
28. Kim, H.: Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950 (2020)
29. Koch, C., Poggio, T.: Predicting the visual world: Silence is golden. Nature Neuroscience pp. 2(1):9–10 (1999)
30. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
31. Langford, M.A., Cheng, B.H.: Enki: a diversity-driven approach to test and train robust learning-enabled systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 15(2), 1–32 (2021)
32. Lazić, L.: Software testing optimization by advanced quantitative defect management. Computer Science and Information Systems 7(3), 459–487 (2010)
33. Li, J., Levine, M.D., An, X., Xu, X., He, H.: Visual saliency based on scale-space analysis in the frequency domain. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(4), 996–1010 (2013)
34. Li, Z., Ma, X., Xu, C., Cao, C.: Structural coverage criteria for neural networks could be misleading. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). pp. 89–92. IEEE (2019)
35. Ma, L., Juefei-Xu, F., Zhang, F., et al.: Deepgauge: Multi-granularity testing criteria for deep learning systems. pp. 120–131 (2018)
36. Ma, L., Zhang, F., Sun, J., et al.: Deepmutation: Mutation testing of deep learning systems. pp. 100–111. IEEE (2018)
37. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
38. Mehrtash, A., Wells, W.M., Tempany, C.M., Abolmaesumi, P., Kapur, T.: Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. IEEE Transactions on Medical Imaging 39(12), 3868–3878 (2020)
39. Oppenheim, A., Lim, J.: The importance of phase in signals. Proceedings of the IEEE 69(5), 529–541 (1981)
40. Pei, K., Cao, Y., Yang, J., Jana, S.: Deepxplore: Automated whitebox testing of deep learning systems. pp. 1–18 (2017)
41. Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P.: Towards out-of-distribution generalization: A survey. arXiv preprint arXiv:2108.13624 (2021)
42. Simonyan, K., Zisserman, A.: Very deep convolutional net-works for large-scale image recognition. In International Conference on Learning Representations (ICLR) (2015)
43. Song, H., Gao, S., Li, Y., Liu, L., Dong, H.: Train-centric communication based autonomous train control system. IEEE Transactions on Intelligent Vehicles 8(1), 721–731 (2023)
44. Song, H., Sun, Z., Wang, H., Qu, T., Zhang, Z., Dong, H.: Enhancing train position perception through ai-driven multi-source information fusion. Control Theory and Technology pp. 1–12 (2023)
45. Sun, J., Mehra, A., Kailkhura, B., et al.: Certified adversarial defenses meet out-of-distribution corruptions: Benchmarking robustness and simple baselines. arXiv preprint arXiv:2112.00659 (2021)
46. Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., Ashmore, R.: Deepconcolic: testing and debugging deep neural networks. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 111–114. IEEE (2019)
47. Sun, Y., Wu, M., Ruan, W., Huang, X., Kwiatkowska, M., Kroening, D.: Concolic testing for deep neural networks. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 109–119 (2018)

48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
49. Tian, Y., Pei, K., Jana, S., Ray, B.: Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering. pp. 303–314 (2018)
50. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: computation in neural systems 14(3), 391 (2003)
51. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8684–8694 (2020)
52. Wang, H., Wang, Y., Ren, W.: Image denoising using anisotropic second and fourth order diffusions based on gradient vector convolution. Computer Science and Information Systems 9(4), 1493–1511 (2012)
53. Wang, J., Chen, J., Sun, Y., et al.: Robot: Robustness-oriented testing for deep learning systems. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 300–311. IEEE (2021)
54. Wang, Z., You, H., Chen, J., Zhang, Y., Dong, X., Zhang, W.: Prioritizing test inputs for deep neural networks via mutation analysis. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 397–409. IEEE (2021)
55. Weiss, M., Tonella, P.: Simple techniques work surprisingly well for neural network test prioritization and active learning (replicability study). In: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 139–150 (2022)
56. Xie, X., Ma, L., Juefei-Xu, F., et al.: Deephunter: a coverage-guided fuzz testing framework for deep neural networks. pp. 146–157 (2019)
57. Xu, Z.Q.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26. pp. 264–274. Springer (2019)
58. Y. Netzer, T. Wang, A.C.: Reading digits in natural images with unsupervised feature learning. arXiv preprint arXiv:1412.6806 (2011)
59. Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. Advances in Neural Information Processing Systems 32 (2019)
60. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM 64(3), 107–115 (2021)
61. Zhang, Z., Xie, X.: On the investigation of essential diversities for deep learning testing criteria. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS). pp. 394–405 (2019)
62. Zhao, C., Mu, Y., Chen, X., Zhao, J., Ju, X., Wang, G.: Can test input selection methods for deep neural network guarantee test diversity? a large-scale empirical study. Information and Software Technology 150, 106982 (2022)

Zhouxian Jiang received the BS degree in computer science and technology from Beijing Jiaotong University, Beijing, China. She is currently pursuing a PhD in software engineering at Beijing Jiaotong University, Beijing, China. Since 2017, she has worked in the Software Evaluation Laboratory of Beijing Jiaotong University. Her research interests include deep learning testing, software reliability testing, and deep learning uncertainty. She has been engaged in software testing for 5 years, and has participated in more than 10 software evaluation projects, research projects, as well as National Key Research and Development Projects (China). E-mail: zhouxianjiang@bjtu.edu.cn

Honghui Li received her MS degree in computer science from the Central South University, Changsha, China, in 1987. Her research interests include software testing technology and test automation. Currently, she is a Professor at the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. She is currently the Deputy Director of the Engineering Research Center of Network Management Technology for High Speed Railway of MOE, Beijing, China. She has long been engaged in software quality assurance technology, high-reliability software, data mining analysis, and railway information technology. She has undertaken national-level provincial and ministerial research such as the National 863 Program, the Nuclear High-Level Project, and the China Railway Corporation. E-mail: hhli@bjtu.edu.cn.

Xuetao Tian received the BS and PhD degree in computer science and technology from Beijing Jiaotong University, Beijing, China. He is now a postdoctoral researcher in Faculty of Psychology, Beijing Normal University, Beijing, China. His research interests include personality prediction, recommender systems, and information extraction. He has published 12 research papers at international journals and conference proceedings. E-mail: xttian@bnu.edu.cn.

Rui Wang received the BS and MS degrees in electronic and information engineering from Beijing Jiaotong University, Beijing, China, and the PhD degree in computer science from the National Institute of Applied Science (INSA) Toulouse, France. Her doctoral research work was conducted at the Laboratory for Analysis and Architecture of Systems, French National Centre for Scientific Research (LAAS-CNRS). Since 2018, she has been an Assistant Professor at the Computer and Information Technology School in Beijing Jiaotong University. Her main fields of interest include machine learning testing, adversarial robustness, and uncertainty quantification. E-mail: rui.wang@bjtu.edu.cn.

Received: September 07, 2023; Accepted: November 18, 2023.