SCF-FGSM: Boosting Transferable Targeted Adversarial Attacks with Feature Mixup and Space Fine-Tuning

Guodong Liu, Houwang Jiang, Wenxing Liao, Xiaolong Liu, Zhiyu Lin, Shihua Zhan*
School of Computer Science and Information Engineering, Fujian Agriculture and Forestry University, Fuzhou, China
**Corresponding Author*

Abstract: With the widespread application of deep neural networks (DNNs) in critical fields such as autonomous driving and medical diagnosis, their adversarial robustness has become a research hotspot. In black-box attack scenarios, the transferability of targeted adversarial examples is limited by differences in decision boundaries between models, and existing methods struggle to achieve efficient attacks. To address this, we propose a novel targeted adversarial attack method, SCF-FGSM, which combines Self-Universality (SU), Clean Feature Mixup (CFM), and Feature Space Fine-Tuning. This method enhances the local feature consistency of adversarial examples through SU, utilizes CFM to generate diverse perturbations to overcome inter-model differences, and incorporates feature space fine-tuning to achieve precise alignment of target features across models. Experiments on the ImageNet dataset demonstrate that SCF-FGSM significantly outperforms existing methods in transferability and attack success rate, especially under Logit loss. Ablation studies and visualization analyses further validate the contributions of each module to transferability, revealing a synergistic mechanism between feature space alignment and perturbation diversity. This provides theoretical support and a technical pathway for improving the transferability of adversarial attacks.

Keywords: Targeted Adversarial Attack; Transferability; Deep Neural Networks; Black-Box Attack

1. Introduction

Deep neural networks (DNNs) have been widely adopted in various fields, raising concerns about their robustness and security. Adversarial attacks pose a significant threat by introducing subtle perturbations that cause misclassification [1]. The transferability of adversarial examples enables black-box attacks, increasing security risks. Studying transferability not only improves defense mechanisms but also provides insights into DNN behavior.

Adversarial attacks can be classified as untargeted or targeted. Untargeted attacks push predictions toward any incorrect label, while targeted attacks force misclassification into a specific class. However, targeted attacks suffer from lower transferability due to inconsistent decision boundaries across models [2], making them particularly challenging in black-box scenarios.

To enhance transferability, researchers have explored feature alignment-based methods. Inkawhich et al. [3] proposed aligning adversarial examples with target class feature distributions, but this requires auxiliary models for each class, increasing computational cost and reliance on shared data distributions. Another study [4] optimized transferability by perturbing intermediate-layer features, yet it assumes that source and target models share the same data distribution, which is often impractical. Additionally, it applies fixed hierarchical perturbations, lacking adaptive feature optimization.

Wei et al. [18] introduced Self-Universality (SU) to remove the need for auxiliary models, improving targeted attack success rates. However, SU suffers from overfitting to source model features and limited feature space utilization, restricting its transferability to black-box models.

To address these limitations, this study proposes SCF-FGSM, integrating Self-Universality (SU), Competition Feature Mixup (CFM) [19], and Feature Space Fine-Tuning[20]. CFM enhances perturbation

diversity through feature competition, while FSFT optimizes feature alignment, improving transferability across models.

Experiments on ImageNet demonstrate that SCF-FGSM significantly improves transferability and targeted attack success rates, outperforming SU and validating its effectiveness and robustness.

2. Related Works

2.1 Transferable Untargeted Attacks

Untargeted attacks aim to mislead DNN-based classifiers into producing incorrect outputs. The iterative fast gradient sign method (I-FGSM) [5] is a baseline method for many untargeted attacks and can be expressed by the following formula:

$$x_0^{adv} = x,\ x_{N+1}^{adv} = Clip_{x,\epsilon}\{x_N^{adv} + \alpha\,\mathrm{sign}(\nabla_x J(x_N^{adv}, y_0))\} \quad (1)$$

where x is the original image; x_0^{adv} is the adversarial example; α is the step size used to update the sample with the gradient sign in each iteration; y_0 is the original label; $sign(\ )$ is the sign function; $Clip_{x,\epsilon}\{\ \}$ is the clipping function that ensures the generated adversarial example satisfies the L_∞ norm constraint; and $\nabla_x J(\ )$ is the gradient of the loss function with respect to the input x_N^{adv}.

I-FGSM's iterative updates improve attack effectiveness in white-box models but result in poor transferability. To address this, researchers have explored data augmentation techniques to mitigate overfitting and enhance transferability.

The transferability of adversarial examples has been explored in multiple ways, primarily through optimization-based and generation-based methods[26].The diverse input method (DI) [6] improves adversarial perturbations by applying random resizing and zero-padding. Lin et al. [7] proposed the scale-invariant method (SIM), which scales input values by a factor of 2 per iteration to introduce scale variations. The admix method [8] extends SIM by integrating images from other labels into the optimization process.To further increase sample diversity, Dong et al. [9] introduced the translation-invariant method (TIM), which optimizes perturbations by incorporating multiple pixel-shifted input versions.Beyond data augmentation, gradient-based optimization strategies also improve transferability. The momentum iterative method (MI) [10] accumulates gradients to escape local minima, while variance tuning (VT) [11] refines gradient updates by adjusting variance, stabilizing optimization, and increasing attack success rates. Previous studies [9] demonstrated that combining DI [6], TI [9], and MI [10] referred to as DTMI—achieves superior transferability.

In contrast to input-level augmentation and gradient optimization, feature space operations directly manipulate intermediate model layers to enhance transferability. Wang et al. [12] proposed the Feature Importance-Aware Attack (FIA), which evaluates feature contributions and prioritizes perturbations on critical features while avoiding excessive reliance on model-specific representations. This significantly improves cross-model transferability.

Zhou et al. [13] further optimized adversarial transferability by maximizing the distance between natural and adversarial images in the intermediate feature space, mitigating gradient vanishing and penalizing high-frequency perturbations. Zhang et al. [14] refined this approach by incorporating integrated gradients [15] to measure feature importance more effectively.

2.2 Transferable Targeted Attacks

Transferable targeted attacks aim to mislead models into predicting a specified target class. Compared to untargeted attacks, they require more precise perturbation optimization, making transferability a greater challenge.

The Feature Distribution Attack (FDA) [3] enhances transferability by modeling intra-class and intra-layer feature distributions, modifying intermediate-layer features instead of solely relying on classification layers. This approach improves black-box transferability more effectively than decision boundary-based methods. FDA further incorporates feature disruption and source minimization terms, and subsequent work [4] refined its performance by integrating cross-entropy loss and multi-layer optimization.

The Targeted Transferable Perturbations (TTP) method [16] improves transferability by maximizing feature space consistency between source and target models. Instead of relying on explicit decision boundary information, TTP adopts a generator-discriminator framework,

using global distribution matching and local neighborhood similarity optimization to enhance black-box attack success. However, TTP requires additional training on auxiliary networks, increasing computational cost.

Another approach to improving transferability is loss function optimization. Zhao et al. [17] demonstrated that directly optimizing the Logit output of the target class mitigates the gradient vanishing problem in cross-entropy loss, leading to stronger attack performance. Unlike resource-intensive methods, Logit loss-based attacks require no additional data or model training, achieving higher targeted transferability through simple iterative optimization.

$$\mathcal{L}_{logit}(f(x^{adv}),y_t) = -l_t(f(x^{adv}))$$

where f represents the white-box source model, $\mathcal{L}_{logit}(\ )$ denotes the Logit loss, y_t is the target class, and $l_t(f(x^{adv}))$ represents the logit value of the model for the target class y_t.

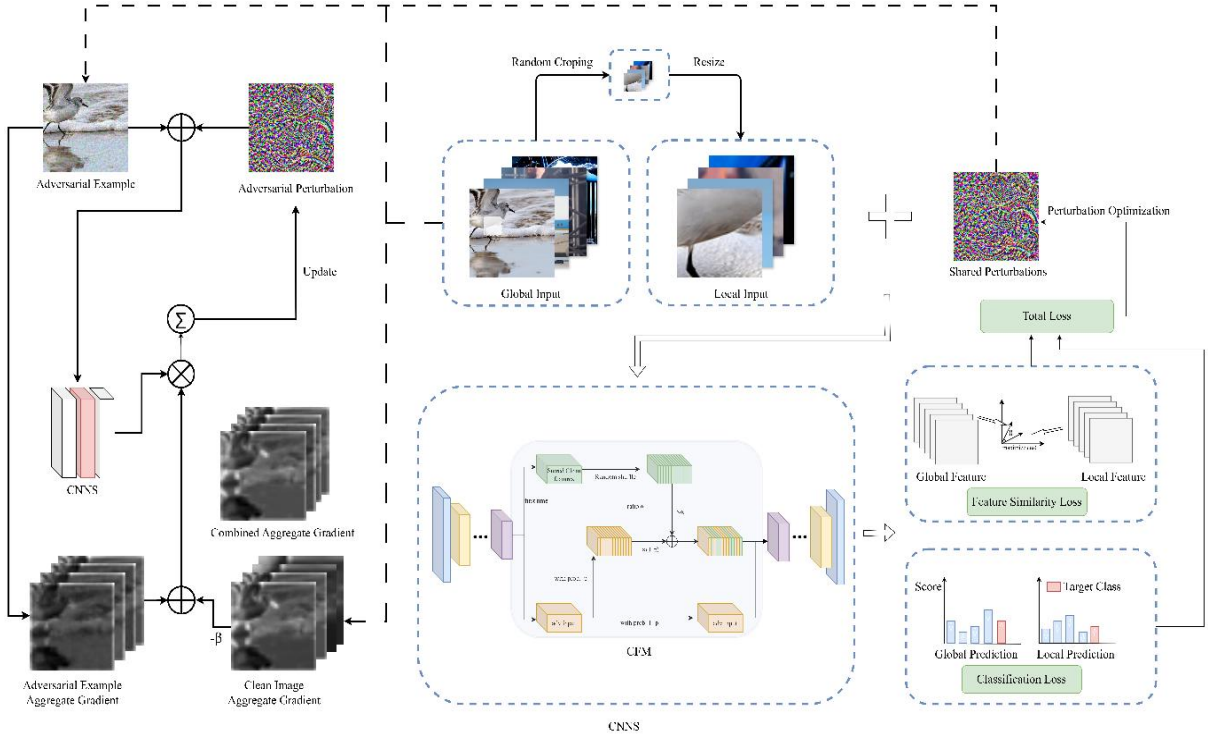

Figure 1. Overview of the SCF-FGSM Method

3. Methodology

3.1 Self-Universality (SU) Attack

Wei et al. [18] demonstrated that DTMI achieves higher targeted transferability than I-FGSM by generating more generalizable perturbations, which enhance attack effectiveness across models. Based on this, Self-Universality (SU) was proposed to further improve transferability in targeted attacks.

Unlike traditional methods that optimize perturbations across different images, SU enhances self-universality by making perturbations agnostic to different local regions within a single image. This allows perturbations to generalize without additional data. SU achieves this by randomly cropping local regions of an image and incorporating them into the iterative attack process. The cropped regions are resized to match the original image dimensions, ensuring consistency.

To strengthen local robustness, SU introduces a feature similarity loss, which maximizes cosine similarity between intermediate features of the global and local inputs. The similarity loss can be formulated as follows:

$$g_{i+1} = \nabla_\delta\big(J(f(x+\delta_i),y_t) + J(f(Loc(x,s)+\delta_i),y_t) - \lambda \cdot CS(f_l(x+\delta_i),f_l(Loc(x,s)+\delta_i))\big) \quad (3)$$

where $Loc(x,s)$ represents the operation of randomly cropping and adjusting the image shape; $f_l(\ )$ denotes the extraction of features from the l-th layer of the white-box source model f; and $CS(\ )$ calculates the cosine similarity score between the global and local input features of the adversarial example. Combining SU with DTMI improves self-

universality, leading to better transferability in targeted attacks. However, SU primarily optimizes local feature consistency, limiting its impact on global feature diversity. As a result, SU-generated adversarial examples may lack sufficient global coverage, reducing their adaptability in black-box models.

3.2 Clean Feature Mixup (CFM)

CFM [19] enhances the transferability of targeted adversarial examples by introducing feature competition during the attack process. Unlike SU, which optimizes local features, CFM optimizes the global feature distribution by randomly mixing clean and adversarial features in the feature space.

CFM applies linear interpolation [21] to blend outputs from selected convolutional and fully connected layers with stored clean features. To prevent excessive interference, it is only attached to deeper layers, where feature maps are smaller, minimizing disruptions from input transformations. Non-activated features are stored before mixing to preserve critical information.To maintain stability across network depths, CFM randomly applies feature mixup with probability p. Within a batch, clean features are shuffled at the image level, enabling adversarial perturbations to mix with both target-class and non-target-class features, enhancing competitive interference. Additionally, the mixing ratio is randomly sampled per channel, increasing diversity across different feature dimensions.

This method efficiently balances feature competition and diversity, improving targeted attack transferability while minimizing distortions to the original feature representation.Mathematically, during the inference phase, the CFM module performs linear interpolation between the input features and the stored clean features. For a batch containing B images, the CFM module stores B clean feature maps, denoted as $\{f_1^c,...,f_B^c\}$, where each feature map has a dimension of $\mathbb{R}^{C\times H\times W}$.The formula for random feature mixing is given as follows:

$$f_i' = (1-\alpha_i)\odot f_i + \alpha_i \odot f_{s_i}^c, \quad i=1,...,B, \tag{4}$$

where $\odot$ represents element-wise multiplication; s_i denotes the randomly shuffled indices; and $\alpha_i \in \mathbb{R}^{C\times 1\times 1}$ is the randomly sampled channel-wise mixing ratio, satisfying $\alpha_i \sim U(0,\alpha_{\max})$.

This study integrates the CFM module into the base perturbations provided by SU, leveraging its random activation and feature shuffling strategies to enhance the diversity of perturbations. Furthermore, by adjusting the upper bound parameter $\alpha_{\max}$ for the mixing ratio and the activation probability p, the method further optimizes the alignment of adversarial examples with target class features, improving the transferability of targeted attacks.

3.3 Feature Space Fine-Tuning(FT)

In untargeted attacks, adversarial examples are pushed away from the clean image in the feature space. However, in targeted attacks, no single feature space point perfectly represents the target class, making direct feature alignment challenging. Feature Space Fine-Tuning addresses this by adjusting feature-level representations of adversarial examples to improve black-box transferability.

Feature space fine-tuning starts with the adversarial example x^{adv} generated by the baseline attack and optimizes its feature distribution through the following steps: First, the accumulated gradient is computed by extracting features from an intermediate layer of the source model and calculating the accumulated gradient with respect to the target class y_t:

$$\Delta\overline{k}_{x^{adv},t} = \sum_{i=1}^{N} \nabla_k J(f_k(x^{adv}), y_t) \tag{5}$$

where f_k represents the feature output of the k-th layer, and J is the classification loss function. At the same time, the accumulated gradient with respect to the original class y_0 is computed from the clean image x:

$$\Delta\overline{k}_{x,0} = \sum_{i=1}^{N} \nabla_k J(f_k(x), y_0) \tag{6}$$

Afterward, the gradients of the target class and the original class are weighted and combined to guide the subsequent optimization:

$$\Delta\overline{k}_{\text{combine}} = \Delta\overline{k}_{x^{adv},t} - \beta\Delta\overline{k}_{x,0} \tag{7}$$

where β is a weighting factor that balances the influence of the target class and the original class.Finally, feature space fine-tuning is applied to the adversarial example x^{adv} using the following optimization objective:

$$\arg\max_{x_{ft}^{adv}} \sum \left(\Delta \bar{k}_{\text{combine}} \cdot f(x_{ft}^{adv}) \right), \quad \text{s.t.} \quad |x_{ft}^{adv} - x|_\infty \leq \epsilon \quad (8)$$

his objective encourages features related to the target class y_t while suppressing features associated with the original class y_o . The implementation of feature space fine-tuning is illustrated in Figure 1.

3.4 SCF-FGSM

The proposed SCF-FGSM method first generates base perturbations using the SU method, ensuring self-universality within the local feature space. Based on the perturbations generated by the SU method, the CFM module is introduced to further enhance the global feature diversity of adversarial perturbations through random activation and feature mixing strategies. Finally, feature space fine-tuning is applied to optimize the perturbations, further improving the alignment of adversarial examples with the target class features while suppressing original class features. The detailed process is shown Figure 1.

4. Experiment

4.1 Experimental Settings

Dataset: Our experiments use the dataset first introduced in the NIPS 2017 Adversarial Attacks and Defenses Competition*.(dataset: https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset). It consists of 1000 images of size 299×299, along with their corresponding ground truth labels and target classes.

Models: We evaluate the transferability of SCF adversarial examples using four pretrained models: Inceptionv 3(Inc-v3)[22], Resnet 50[25][25],DenseNet 121[23] and VGG 16[24]

Hyperparameter settings:The maximum perturbation of adversarial examples is set $\epsilon = 16$ .The step size is set to $\alpha = 2$ .The maximum number of iterations is set to $I = 300$ .In the CFM parameter settings, the channel mixing ratio $\alpha_i \sim U(0, 1)$ is randomly sampled, and the mixing probability p is set to 0.1.The number of iterations for feature space fine-tuning is set to $N_{ft} = 10$ .For the fine-tuning layer k , we select:Mixed_6b for Inc-v3,The last layer of the third block for ResNet-50 and DenseNet-121,Conv4_3 for VGG-16.

Table 1. The Table Presents the Impact of SCF-FGSM on the Attack Success Rate (TASR) Compared to the Baseline SU Attack under Different Source Models.

Attack	White-box Model: Res50		
	Dense121	VGG16	Inc-v3
CE-SU	**52.2**	**41.8**	**8.5**
CE-SCF	44.4	41.5	7.9
Logit-SU	**76.6**	67.8	11.9
Logit-SCF	74.0	**68.8**	**16.9**

Attack	White-box Model: Dense121		
	Res50	VGG16	Inc-v3
CE-SU	34.8	29	9.8
CE-SCF	**41.7**	**38.3**	**11.5**
Logit-SU	51.6	47.6	12.3
Logit-SCF	**63.5**	**58.6**	**17.2**

Attack	White-box Model: VGG16		
	Res50	Dense121	Inc-v3
CE-SU	2.2	2.1	0.1
CE-SCF	**3.3**	**3.1**	**0.2**
Logit-SU	14.5	16.8	1.3
Logit-SCF	**16.7**	**20.4**	0.8

Attack	White-box Model: Inc-v3		
	Res50	Dense121	VGG16
CE-SU	2.7	4.7	1.9
CE-SCF	**4.4**	**6.6**	**5.1**
Logit-SU	3.4	6.1	3.2
Logit-SCF	**6.1**	**11.7**	**8.8**

4.2 Single-Model Transfer Attack

To evaluate the effectiveness of the SCF method, we conducted single-model transfer attack experiments, where adversarial examples generated from a white-box model were tested on three black-box models. The results, shown in Table 1, are evaluated using the targeted attack success rate (TASR).

SCF achieves a higher TASR under Logit loss, outperforming Logit-SU across multiple models. For instance, when DenseNet-121 is the white-box model, SCF improves the attack success rate on ResNet-50 and VGG-16, indicating that feature mixing and optimization enhance perturbation transferability beyond merely minimizing classification loss. Among different white-box models, ResNet-50 and DenseNet-121 demonstrate better transferability compared to VGG-16 and Inception-v3, likely due to their richer feature representations. In contrast, shallower models rely more on surface-level features, limiting

their ability to generate effective perturbations for black-box attacks. While SU outperforms SCF in certain CE loss scenarios, Logit-SCF consistently achieves higher success rates, especially in cases where feature alignment with the target class is crucial. This suggests that Logit-SCF better optimizes perturbations for targeted attacks, making it more effective in cross-model transferability.

Despite lower overall success rates when VGG-16 and Inception-v3 are used as white-box models, SCF still shows improvements over SU, leveraging feature space fine-tuning to capture target class features more effectively. This demonstrates SCF's adaptability in constrained scenarios.

Experimental results confirm that SCF significantly outperforms SU in single-model transfer attacks, particularly under Logit loss. SCF is especially effective when applied to deeper models like DenseNet-121 and ResNet-50, highlighting its ability to generate globally robust adversarial perturbations. Furthermore, its adaptability across different architectures reinforces its potential in enhancing targeted adversarial attacks.

4.3 Ablation Studies

Effectiveness of the CFM Module: We evaluated the effectiveness of the CFM module through ablation experiments, with results presented in Table 2. The findings indicate that CFM improves attack success rates, particularly under Logit loss, by enhancing feature mixing and increasing perturbation diversity.

For example, when DenseNet-121 is the white-box model, Logit-SCF with CFM (Logit-SCFw) achieves higher transferability than Logit-SCF without CFM (Logit-SCFwo), increasing the success rate on ResNet-50 and VGG-16. Similarly, using VGG-16 as the white-box model, the attack success rate on DenseNet-121 improves, confirming CFM's role in boosting transferability for smaller models.

However, CFM does not always enhance performance. For instance, with ResNet-50 as the white-box model, attack success rates on DenseNet-121 and Inception-v3 slightly decrease. This may result from CFM intensifying global perturbations, leading to the loss of local feature information, reducing adaptability for certain target models. Under CE loss, the impact of CFM is less pronounced, with minor performance drops in some cases.

These results suggest that CFM's effectiveness depends on the attack environment, and its feature mixing mechanism varies across different target models. Future work could optimize CFM hyperparameters to refine feature mixing strategies, ensuring it maximizes transferability while minimizing negative impact on specific models.

Table 2. The Table Presents the Effect of Incorporating the CFM Module on the Attack Success Rate (TASR) across Different Source Models

Attack	White-box Model: Res50		
	Dense121	VGG16	Inc-v3
CE-SCFwo	**47.4**	40.2	7.7
CE-SCFw	44.4	**41.5**	**7.9**
Logit-SCFwo	73.2	68.1	**17.8**
Logit-SCFw	**74.0**	**68.8**	16.9

Attack	White-box Model: Dense121		
	Res50	VGG16	Inc-v3
CE-SCFwo	41.6	37.5	9.9
CE-SCFw	**41.7**	**38.3**	**11.5**
Logit-SCFwo	62.1	57.6	17.1
Logit-SCFw	**63.5**	**58.6**	**17.2**

Attack	White-box Model: VGG16		
	Res50	Dense121	Inc-v3
CE-SCFwo	2.3	2.4	0
CE-SCFw	**3.3**	**3.1**	**0.2**
Logit-SCFwo	**18.7**	19.4	**1.3**
Logit-SCFw	16.7	**20.4**	0.8

Attack	White-box Model: Inc-v3		
	Res50	Dense121	VGG16
CE-SCFwo	3.9	6.2	5.1
CE-SCFw	**4.4**	**6.6**	**5.1**
Logit-SCFwo	**6.5**	11.2	7.9
Logit-SCFw	6.1	**11.7**	**8.8**

w/ indicates that the CFM module is included, while **w/o** denotes that the CFM module is not applied.

Number of Iterations for Feature Space Fine-Tuning: We conducted an ablation study to assess the impact of feature space fine-tuning iterations on TASR, with results shown in Figure 2. Figure 2(a) corresponds to CE loss, and Figure 2(b) to Logit loss.

Regardless of the loss function, TASR converges around 10 iterations, with limited improvements beyond this point. Increasing iterations to 15 or 20 yields minimal gains and may even degrade performance in certain

models. For example, under Logit loss, the TASR for ResNet-50 and VGG-16 stabilizes at 50% and 20%, respectively, after 10 iterations. For complex models like Inception-v3, an optimal iteration count enhances adversarial feature alignment. Under Logit loss, the TASR of Inception-v3 improves from 15% initially to nearly 20% at 10 iterations, after which performance plateaus.

In summary, 10 iterations provide an effective balance between optimization and computational efficiency, enhancing transferability while avoiding performance degradation or unnecessary computational costs. This setting ensures efficient fine-tuning across various models and loss functions.

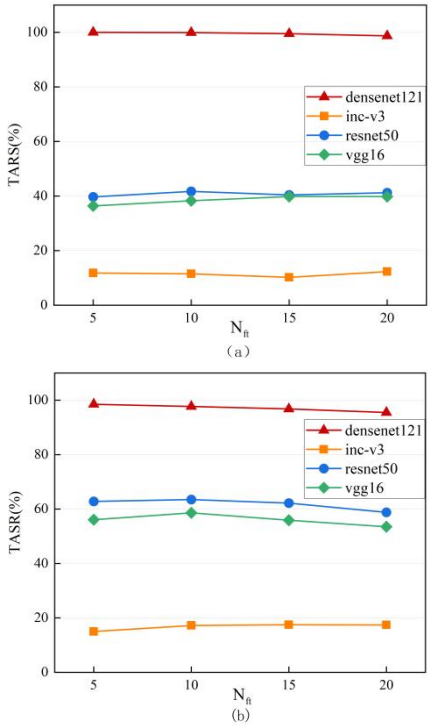

Figure 2. Number of Iterations for Feature Space Fine-Tuning, Figure 2(a) corresponds to CE loss, and Figure 2(b) corresponds to Logit loss.

5. Conclusion

This study addresses the challenge of improving the transferability of targeted adversarial examples by proposing an efficient attack method, SCF-FGSM, which integrates the SU method, CFM module, and feature space fine-tuning. Experimental results demonstrate that the proposed method significantly enhances transferability across multiple white-box and black-box models, particularly achieving superior performance in targeted attack success rate under Logit loss.Ablation studies validate the effectiveness of both the CFM module and feature space fine-tuning, while further exploring the optimization of hyperparameters such as the number of iterations.For future work, dynamic feature selection mechanisms and more efficient feature mixing strategies can be explored to further improve the generalization and transferability of adversarial examples.

Acknowledgments

This paper was supported by the Fujian Provincial Natural Science Foundation of China (No.2021JO1129), the Fujian Provincial Higher Education Technology Research Association Fund Project (No.H2000134A), and the Fujian Agriculture and Forestry University Horizontal Technology Innovation Fund (No.KHF190015).

References

[1] GoodFellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[C]//International Conference on Learning Representations. San Diego: ICLR, 2015: 1-11.

[2] Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).

[3] Liu Y, Chen X, Liu C, et al. Delving into transferable adversarial examples and black-box attacks[J]. arXiv preprint arXiv:1611.02770, 2016.

[4] Inkawhich N, Liang K J, Carin L, et al. Transferable perturbations of deep feature distributions[J]. arXiv preprint arXiv:2004.12519, 2020.

[5] Inkawhich N, Liang K, Wang B, et al. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability[J]. Advances in Neural Information Processing Systems, 2020, 33: 20791-20801.

[6] Kurakin A, Goodfellow I J, Bengio S. Adversarial examples in the physical world[M]//Artificial intelligence safety and security. Chapman and Hall/CRC, 2018: 99-112.

[7] Xie C, Zhang Z, Zhou Y, et al. Improving transferability of adversarial examples with input diversity[C]//Proceedings of

the IEEE/CVF conference on computer vision and pattern recognition. 2019: 2730-2739.

[8] Lin J, Song C, He K, et al. Nesterov accelerated gradient and scale invariance for adversarial attacks[J]. arXiv preprint arXiv:1908.06281, 2019.

[9] Wang X, He X, Wang J, et al. Admix: Enhancing the transferability of adversarial attacks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 16158-16167.

[10] Dong Y, Pang T, Su H, et al. Evading defenses to transferable adversarial examples by translation-invariant attacks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4312-4321.

[11] Dong Y, Liao F, Pang T, et al. Boosting adversarial attacks with momentum[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 9185-9193.

[12] Wang X, He K. Enhancing the transferability of adversarial attacks through variance tuning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 1924-1933.

[13] Wang Z, Guo H, Zhang Z, et al. Feature importance-aware transferable adversarial attacks[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 7639-7648.

[14] Zhou W, Hou X, Chen Y, et al. Transferable adversarial perturbations[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 452-467.

[15] Zhang J, Wu W, Huang J, et al. Improving adversarial transferability via neuron attribution-based attacks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 14993-15002.

[16] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks[C]//International conference on machine learning. PMLR, 2017: 3319-3328.

[17] Naseer M, Khan S, Hayat M, et al. On generating transferable targeted perturbations[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 7708-7717.

[18] Zhao Z, Liu Z, Larson M. On success and simplicity: A second look at transferable targeted attacks[J]. Advances in Neural Information Processing Systems, 2021, 34: 6115-6128.

[19] Wei Z, Chen J, Wu Z, et al. Enhancing the self-universality for transferable targeted attacks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 12281-12290.

[20] Byun J, Kwon M J, Cho S, et al. Introducing competition to boost the transferability of targeted adversarial examples through clean feature mixup[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 24648-24657.

[21] Zeng H, Chen B, Peng A. Enhancing targeted transferability via feature space fine-tuning[C]//ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 4475-4479.

[22] Zhang H. mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.

[23] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

[24] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[25] Simonyan K. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.、

[26] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[27] Gu J, Jia X, de Jorge P, et al. A survey on transferability of adversarial examples across deep neural networks[J]. arXiv preprint arXiv:2310.17626, 2023.

