EXPLORING THE GENERALIZABILITY OF CNNS VIA ACTIVATED REPRESENTATIONAL SUBSTITUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Convolutional neural networks (CNNs) have achieved remarkable success in various fields due to their excellent generalizability. To explore the relationship between CNN representations and generalization, we propose an Activation Representation Substitution (ARS) metric based on the disentangled visual representations of convolution kernels. Without additional data, we obtain the disentangled visual representation of a kernel in the convolutional layer by iterating over a random image, and feed it into the CNN. The output activations of the other kernels in that convolutional layer are then investigated. When all other output activations are small, the ARS of the convolution kernel from that representation is also small, indicating that the representation is important for CNN. Our experiments with ablation analysis confirm the importance of the low ARS convolution kernel on accuracy. With ARS, we also explain batch normalization and class selectivity. By comparing the model performances on the test set, we empirically find that when the convolutional layer contains a large number of low ARS convolution kernels, the model has good generalization. ARS is a metric that can be used to better understand model generalizability without using external data.

1 INTRODUCTION

Convolutional neural networks (CNNs) become a standard technique in image recognition He et al. (2016) due to their excellent generalizability. However, evaluating CNN generalization ability is difficult, as even CNNs trained with the same strategy and data have varied generalizability Zhang et al. (2021). Also, their generalization performance is hard to be explained Yoon et al. (2019). Over-parameterized CNNs, for example, do not overfit but have strong generalization ability Allen-Zhu et al. (2019). How CNNs generalize on different datasets remains unsolved to researchers.

In statistical learning theories, there are methods to evaluate a model's ability to generalize, including VC dimension Vapnik (1999) and Rademacher complexity Bartlett & Mendelson (2002), which analyze the generalization error of a model. These methods explore the relationship between models' robustness, complexity, and generalization. And these theories suggest that models with an excessive amount of parameters tend to overfit the training data, which is detrimental to the model's generalizability. Yet, these findings do not seem to stand well when examining the performance of CNNs, which are usually over-parameterized Zhang et al. (2021). Thus, statistical learning theories cannot explain the generalizability of CNNs. Apart from these theories, many studies have evaluated the generalizability of CNNs via their gradients in the optimization process. For instance, some researches Hochreiter & Schmidhuber (1997); Neyshabur et al. (2017) argued that the generalizability of CNNs is associated with the flatness of the minimum and PAC–Bayes bounds, where flat minima are indicative of good model generalization. Wang et al. 2018 proved that the generalizability of the model is related to the smoothness of solution landscape, number of the parameters, and training data. But Dinh et al. 2017 drew a contradictory conclusion that CNNs with sharp minima solution landscape also have good generalizability. The major problem with these methods is that they can only estimate the generalization bound yet cannot quantify the model's generalizability.

In practice, we use the cross-validation method on a pre-designed test set to evaluate the model's generalizability, with higher accuracy indicating better generalizability. But this approach has also been questioned by Recht et al. 2018, who argues that the good performance of CNNs on the designed test set may not demonstrate their generalizability. Because the number of data in the test set is

limited, the performance of the model depends on the designed test set. Models that perform well may have lower accuracies than those that perform poorly when inferring new data.

In this regard, we argue that CNNs, as a kind of representation learning, have exhibited a strong correlation between the representation of their main component of convolution kernels and the generalizability. Therefore, we can explore the model's generalizability by analyzing the representations of the convolution kernels. Based on this insight, we propose a novel metric named Activated Representation Substitution (ARS), which quantifies the importance of representations of each convolution kernel. When a convolution kernel has a low ARS, only a few other kernels are responsive to its representation. In other words, the representation of that convolution kernel is difficult to be substituted, and it is important. Based on ARS, we can explore the relationship between the features learned by the convolution kernel and the generalization. When a CNN has more convolution kernels with low ARS, it indicates that the model has many important representations and is sensitive to a wide variety of features, thus having good generalizability.

Specifically, the acquisition of the representations of CNNs is the most important step in analysing generalizability. For CNNs with entangled representations Levine et al. (2018), we first propose a disentangled visual representation algorithm named Independent Activation Maximization (IAM), which originates from the Activation Maximization (AM) algorithm Erhan et al. (2009). It suppresses the responses of other convolution kernels in the same layer. Then, we feed the generated representations into the CNN and analyze the output activations of all the convolution kernels in the corresponding layer. A high output activation indicates that the convolution kernel produces a high-response to the visual representation. Based on the responses of other convolution kernels to the generated representations, we propose Representational Substitution (RS) and Activated Response (AR), which are two metrics that can evaluate the substitutability and activations of the representations. Finally, combining RS and AR, we obtain the ARS of the convolution kernel. By feeding all visual representations, we can obtain the ARS of each convolution kernel. As mentioned above, the convolution kernel with low ARS is important. In our experiments, we first verify the effectiveness of the IAM. Under ablation analysis, we find that convolution kernels with low ARS are more important for the accuracy of the model. Then, we use ARS to explain batch normalization (BN) Ioffe & Szegedy (2015) and class selectivity (CS) Morcos et al. (2018). We also compare the performance of the models under the test set with their overall ARS. The experimental results show that the models with lower overall ARS have better generalizability. The contributions of this study are summarized as follows:

- Our proposed IAM algorithm can obtain a disentangled visual representation of the convolution kernel, demonstrating the features extracted by the convolution kernel.
- Our proposed ARS can quantify the importance of the representations of each convolution kernel. A low ARS of the convolution kernel indicates that its representation is more important for the accuracy of the model.
- ARS can explain some regularization terms and the generalization performance of CNNs. It provides insights into the explanation of the model.
- We find that models with lower overall ARS have higher generalizability. ARS is a statistic metric that may be used to analyze model generalizability without involving external data.

2 METHODOLOGY

2.1 INDEPENDENT ACTIVATION MAXIMIZATION

To obtain a visual representation, Erhan et al. 2009 propose the AM algorithm. The algorithm first inputs a randomly generated noisy image and records the output activation of the convolution kernel for which the visual representation needs to be obtained. Then, through a gradient ascent algorithm, the input image is continuously updated so that the output activation of the targeted convolution kernel becomes larger. The final updated image is the visual representation of the convolution kernel.

In detail, in image recognition, we assume that the weights of the CNN are Θ. During the training, the weights of the model are continuously updated by the backpropagation algorithm. And we use Θ^* to denote the optimal weights of the model after convergence. For a new image x, each convolution kernel produces a different response to the input data. When the image x is fed to the model for feedforward operations, these convolution kernels are located on different convolutional layers of the

CNN. Therefore, we define the output activation of each convolution kernel as $h_{l,i}(x, \Theta^*)$, which represents the output activation of the i-th convolution kernel at the l-th layer when the input is x. In the AM algorithm, the input image is random noise. In the gradual update, we expect the output activation of the target convolution kernel to be maximum, which can be defined as follows

$$x^* = \arg\max h_{l,i}(x, \Theta^*). \tag{1}$$

With optimal weights Θ^*, the initial random noise x is updated iteratively by a gradient ascent algorithm. The final x^* allows the i-th convolution kernel at the l-th layer to produce the maximum activation, which is the visual representation of the corresponding convolution kernel.

However, in CNNs, the representations of convolution kernels are entangled Levine et al. (2018). The generated x^* may be a mixture of multiple convolution kernel representations. When evaluating the model's generalizability, we need to quantify the importance of the representations of each convolution kernel. Such a mixture of representations cannot be used directly to evaluate model's generalizability. In response, we propose an improvement to the AM algorithm called the IAM algorithm. During the generation of x^*, it suppresses the responses of other convolution kernels in the same layer. And the final x^* causes only the target convolution kernel to produce a high response as much as possible. The IAM algorithm is defined as

$$x^* = \arg\max(h_{l,i}(x, \Theta^*) - \frac{1}{I}\sum\nolimits_{i=1}^{I} h_{l,-i}(x, \Theta^*)). \tag{2}$$

Under the IAM algorithm, the final x^* obtains high activations in the target convolution kernel while maintaining the small activations of the other convolution kernels in the same layer. It is a disentangled visual representation of the target convolution kernel. However, because some representations of the convolution kernels are similar or meaningless, there could still be many convolution kernels that produce a high response in the same layer. But these high-response convolution kernels are also the key to quantifying the importance of convolution kernel representation.

2.2 Representational Substitution

After obtaining the disentangled representation of each convolution kernel, we can evaluate the model generalization by measuring these representations. When the model has more representations, it has good generalization Zhang et al. (2021); Meng et al. (2020). And because the number of convolution kernels for the same model is fixed, we need to evaluate whether the representation of the convolution kernel can be substituted by others. Representations are not substitutable, indicating that the model has more representations.

We feed the disentangled representation into the model and record the output activations of all convolution kernels of the layer where the convolution kernel that generated the representation is located. As mentioned above, other convolution kernels may also have high activations. The representations of these convolution kernels are redundant or similar to the one that generated the representation, so the IAM cannot suppress the responses of these convolution kernels. We propose RS to quantify the proportion of these convolution kernels. When the RS of the target convolution kernel is small, the representations of the convolution kernels cannot be substituted by the representations of other convolution kernels. The formula for RS is defined as follows

$$RS(l, i) = \frac{N_{x_{l,j} > x_{l,i}}}{N_{x_l}} \quad s.t. \quad x_{l,i} = f(IAM(l, i)), \tag{3}$$

where $RS(l, i)$ denotes the rate of kernels that produce higher response to the disentangled representation of the i-th convolution kernel in the l-th layer. N_{x_l} denotes the number of convolution kernels of the l-th layer, and $x_{l,i}$ refers to the out activation of the target convolution kernel. $N_{x_{l,j} > x_{l,i}}$ denotes the number of other convolution kernels with output activations greater than the target convolution kernel in the l-th layer. $f(\cdot)$ denotes the feedforward operation of the model, while the input image is the disentangled visual representation of the target convolution kernel. RS can quantify the substitutability of representations on the same layer in the range $(0, 1)$. The convolution kernels with low RS indicate that their representations are hard to be substituted.

2.3 ACTIVATED REPRESENTATIONAL SUBSTITUTION

However, RS cannot be used to quantify the importance of the representations of the convolution kernels. When the visual representation causes multiple convolution kernels to produce high responses, there are two main causes, (1) the representation of the convolution kernel is redundant, which is similar to the representations of other convolution kernels, and (2) the convolution kernels do not learn the features. According to Equ. 3, when the visual representation is almost 0, the output activations of all other convolution kernels on the same layer also are close to 0. This may lead to many activations larger than the target convolution kernel but meaningless. So both cases lead to a large RS of the target convolution kernel, but for different importance of the representations. On the other hand, there is also a difference in the representation of the convolution kernel for small RS. When the output activation is passed to the next layer, if the activation is small, then it may also have a small impact on the operations in the later layers. In summary, although RS quantifies the substitutability of the representations of the convolution kernel, it does not quantify the importance of the representations.

We further propose ARS to quantify the importance of the representation of the convolution kernel. First, we use the activated representation (AR) to represent the proportion of non-zero values in the activation, as follows

$$AR(l,i) = N_{x_{l,i,k}>0}/N_{x_{l,i}}, \tag{4}$$

where $N_{x_{l,i,k}>0}$ denotes the dimension of the output activation greater than 0, and $N_{x_{l,i}}$ denotes the dimension of the output activation of the target convolution kernel. AR indicates the effect of activations that can be passed to later layers. Combining the RS and AR of the convolution kernel, ARS can be defined as

$$ARS(l,i) = \frac{2\tan^{-1}(RS(l,i)/AR(l,i))}{\pi} \tag{5}$$

The value range of ARS is in $(0, 1)$. For a convolution kernel with a small RS, the ARS of the convolution kernel becomes smaller when its AR is large. It indicates that the representation of this convolution kernel is more important. When its AR is small, the ARS of the convolution kernel becomes larger, and its representation is less important considering the impact on the later layers. Similarly, for a convolution kernel with a large RS, the ARS becomes smaller when its AR is large. Because its representation is meaningful compared to the case when no features are learned. The ARS of the convolution kernel without learned features is large. The ARS can distinguish these differences and quantify the importance of the representation. Considering the large range of RS to AR ratios and the greater importance of low ARS, we used an inverse trigonometric function to remap the ratio $(0, 1)$. It changes quickly at smaller values and slowly at larger values. For model's generalizability, convolution kernels with small ARS are important.

2.4 MODEL ARS

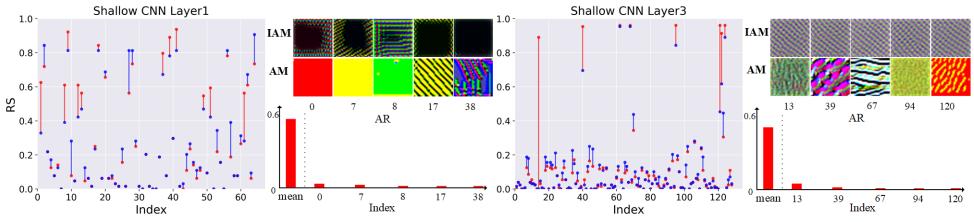
The generalization ability of the model is related to the representation of all convolution kernels. Therefore, we use the overall ARS of the CNN, which can be written as

$$MARS = \sum\nolimits_{l}^{L} \alpha_l (\frac{1}{I_l} \sum\nolimits_{i}^{I} ARS(l,i)). \tag{6}$$

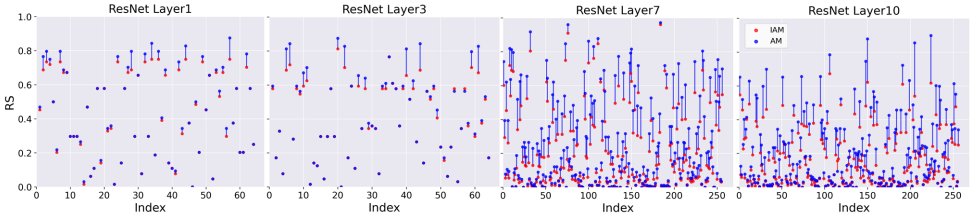
α is the hyperparameter, which is the weight of each convolutional layer. In general, α is large for the shallow layers and small for the deep layers.

3 EXPERIMENTS

In the experiment, we used two models to validate ARS, a pre-trained ResNet He et al. (2015) under ImageNet, and a shallow CNN in VGG style Simonyan & Zisserman (2015). The shallow CNN has four convolutional layers with 64, 64, 128, and 128 convolution kernels.

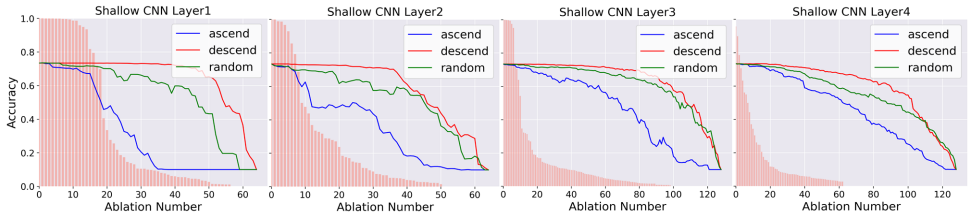


(a)

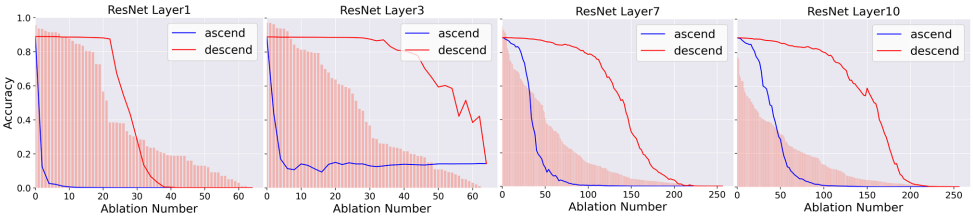


(b)

Figure 1: The blue and red dots represent the RS under AM and IAM. When the RS of IAM is smaller than the RS of AM, we use the blue line. On the contrary, we use the red line. **(a)** layer 1 and layer 3 of the shallow CNN. And some convolution kernels with red lines for AM, IAM, and their AR. **(b)** represents the convolution kernels in layer 1, layer 3, layer 7 and layer 10 of ResNet.



(a)



(b)

Figure 2: **(a)** and **(b)** are the cumulative ablations at layers. The colors represent the different ablation modes. The curves indicate the accuracy on the test set. The red histogram indicates the ARS of the convolution kernel in descending order.

3.1 RS AND ARS

The visual representation of the convolution kernel is the basis for evaluating the model's generalizability. Our proposed IAM algorithm, compared to the AM algorithm, is effective in suppressing the responses of other convolution kernels and obtaining the disentangled visual representation. It is important for the RS of the convolution kernel.

The shallow CNN is trained from the CIFAR-10 dataset Krizhevsky & Hinton (2009). Figure 1(a) shows the RS of the same convolution kernel after using the AM and the IAM algorithm for the first and third layers of the shallow CNN. For the same convolution kernel, the blue dots indicate the RS

obtained by AM, while the red dots indicate the RS obtained by IAM. In Figure 1(a), most of the differences among them are marked with blue lines, which indicates that the RS by AM is higher than the RS by IAM. This is because the disentangled visual representation under the IAM algorithm is difficult for other convolution kernels to produce a high response, and the RS is small. In Figure 1(b), this conclusion also holds for the pre-trained ResNet. We visualize the RS comparisons on layer 1, layer 3, layer 7, and layer 10. Most of the RS by IAM is lower. The experimental results show that the RS under IAM can better represent whether the representations are easily substituted.

In Figure 1(a), there are also some convolution kernels in the shallow CNN with red lines, which indicates that the RS computed by IAM is higher than that calculated by AM. Their visual representations under AM contain complex textures by visualizing the IAM and AM of these convolution kernels. These visual representations generated by the AM algorithm are influenced by other convolution kernels rather than their visual representations. When we use the IAM to obtain their disentangled visual representations, these convolution kernels do not extract meaningful features. Most of their visual representations are close to 0, which leads to other convolution kernels producing similar results that improve RS. So RS does not demonstrate the importance of representations. We also show the AR of these convolution kernels. The AR of these convolution kernels is lower than the average AR of all convolution kernels in the same layer. Therefore, the ARS combines AR and RS to represent the importance of the convolution kernel representation. For example, for convolution kernel 12 in layer 1 of the shallow CNN, its RS is 0.61 and AR is 0.0141, which gives its ARS of 0.985. It indicates that the representation of this convolution kernel is not important. In contrast, in ResNet, there are few differences with red lines, which suggests that most convolution kernels can extract meaningful features. Regarding AM and IAM, the step number setting is not sensitive to the final result and the examples can be found on Appendix A.

3.2 Ablation Analysis of ARS

To verify the effect of different ARS on model's generalizability, we perform cumulative ablation experiments on the convolution kernels and evaluate them by the change in accuracy under the test set. We let the output of the selected convolution kernels be zero according to the order of ARS and gradually ablate all the convolution kernels in a layer. Figure 2 shows the results of the ablation experiments for shallow CNN and ResNet. The red and blue curves represent the cumulative ablation in descending and ascending order. The final accuracy of some layers is not zero because the output is fixed to a certain class, and its accuracy does not change. In shallow CNN and ResNet, the red curve is always above the blue curve, which indicates that the convolution kernels with low ARS have a greater impact on model accuracy when ablating the same number of convolution kernels. When we perform cumulative ablation of ARS in descending order, the accuracy of the model changes slowly in the early stage. When the ARS becomes very low, the accuracy of the model changes rapidly. These suggest that the ablation of convolution kernels with large ARS has little effect on model accuracy. Therefore, when we perform cumulative ablation of ARS in ascending order, the accuracy of the model drops quickly. More, in shallow CNNs, we conduct random cumulative ablation on convolution kernels. The accuracy of the model gradually decreases, which fails to show the importance of the representation. These experimental results show that the representation of low ARS convolution kernels is more important for model generalization.

Table 1: Maximum accuracy gap and mean ARS of shallow CNNs.

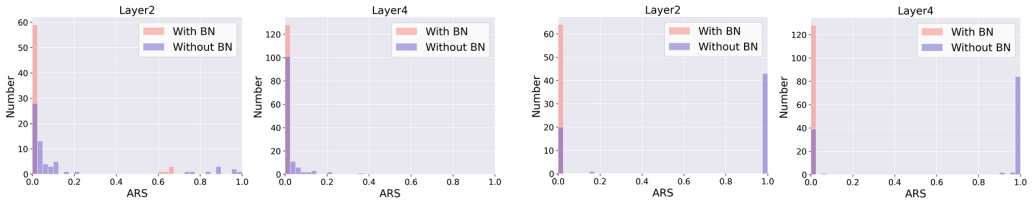
	Shallow CNN				ResNet50			
	layer1	layer2	layer3	layer4	layer1	layer3	layer7	layer10
Gap	0.627	0.438	0.425	0.294	0.882	0.856	0.827	0.808
Mean	0.338	0.227	0.132	0.088	0.329	0.254	0.203	0.158

To show the difference between the two ablation modes, we calculate the maximum gap in accuracy when ablating the same number of convolution kernels. Table 1 shows that the maximum accuracy gap for the first layer in shallow CNN and ResNet is 0.627 and 0.882. This gap decreases as the depth of the CNN increases. Because the average ARS of the layers gradually decreases, their maximum accuracy gap also becomes smaller. On the other hand, this gap changes rapidly in the shallow CNN but slowly in ResNet. We record the ARS when the accuracy is reduced by 5%. In the shallow CNN,

the ARS corresponding to each layer is 0.0455, 0.0528, 0.0175, and 0.0374, which shows that the ablation with ARS greater than 0.06 has minimal effect on the model accuracy of the shallow CNN. In the first seven layers of ResNet, the maximum ARS threshold is 0.87 when the accuracy is reduced by 5%. So for ResNet, ablating convolution kernels with ARS greater than 0.87 in the first seven layers does not affect the accuracy. The experimental results show that ARS can be used as a metric to design new model compression algorithms.

3.3 Relationship Between BN and ARS

ARS can evaluate the importance of each convolution kernel representation. So for those strategies that can affect the model's generalizability, such as BN, ARS can also measure their differences. Thus, we explored the differences in ARS between models with and without BN.



(a) (b)

Figure 3: **(a)** denotes the two models trained under the CIFAR-10 dataset. **(b)** denotes the model trained under the SVHN dataset. They both represent the ARS distribution in shallow CNNs with and without BN. The ARS of all convolution kernels is small when BN is added.

Specifically, we add BN layers to a new shallow CNN. We trained both models with the CIFAR-10 and SVHN Netzer et al. (2011) datasets. Their accuracies are about 4% higher than those of the shallow CNN without BN. The experimental results are shown in Figure 3, where we show the distribution of the ARS of the convolution kernels of both models under both datasets for layer 2 and layer 4. In these layers, the ARS of the convolution kernels with BN are mostly around 0, and their overall ARS is low. And for the model without BN has more convolution kernels with large ARS, which indicates that the representations of these convolution kernels are not important for the model's generalizability. Under the SVHN dataset, for the CNN without BN, many of the convolution kernels are close to 1, which indicates that the model generalization ability is poorer. It also has about 6% lower accuracy than the model with BN.

The addition of the BN layer supports the validity of ARS, and the reason why BN is effective is explained. BN encourages each convolution kernel to obtain low ARS and makes the representations of convolution kernels important, thus improving the model generalization. It also provides insights for designing a new regularization to make the ARS of the convolution kernel smaller.

3.4 Relationship Between CS and ARS

ARS can also be used to explain other conclusions about the model's generalizability. For example, Morcos et al. 2018 propose that CS quantifies the response of convolution kernels to classes. Their conclusion demonstrates that convolution kernels with high CS are not important for the generalizability, and it is counterintuitive. A high CS refers to a high response of a convolution kernel to a class, and it is easy to explain. For this result, we do a correlation analysis about the CS and ARS.

Figure 4 shows that ARS and CS have a strong correlation with each other both in shallow CNN and ResNet of the first few convolutional layers. Convolution kernels with high CS also have high ARS. This indicates that the representations of these convolution kernels are not important for the generalization of the model. When the convolution kernels have high responses to classes, it suggests that they have large AR. Then the large ARS is caused by a higher RS. This indicates that the representations of other convolution kernels easily replace the representation of that convolution kernel. So they are unimportant and have little impact on the model's generalizability. This is consistent with the conclusion of CS about the model's generalizability. More, based on the distribution of ARS,

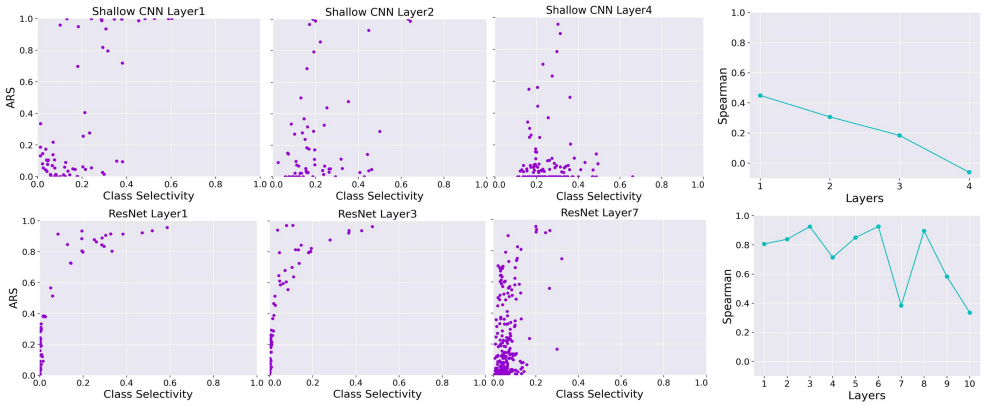


Figure 4: The correlations between the shallow CNN and ResNet at different layers of ARS and CS. The last plot represents the Spearman correlation coefficients between ARS and CS for all layers.

we can also know that low CS is also not always beneficial for the model's generalizability. ARS explains the conclusion of CS about the model's generalizability.

Furthermore, we find that the correlation between CS and ARS is decreasing. We believe that this shows the limitation of the CS method. For example, neurons before the classifier have high responses to classes and high CS. These neurons, however, are important for the model. This is inconsistent with the conclusion of CS. According to the principle of ARS, their representations are important, and they have low ARS. The ARS quantifies better the importance of the convolution kernel representations. And the computation of CS relies on a large number of images. In contrast, the computation of ARS is done by quantifying the importance of the representations of the convolution kernels and without using datasets. ARS provides a new approach to model explanation.

3.5 Relationship Between In-Domain Testing and MARS

In the previous section, we mentioned the limitations of using test sets to measure model generalizability Recht et al. (2018). However, a reasonably designed and consistent benchmark is still a widely acknowledged way for a fair comparison of model generalizability. Therefore, to explore the relationship between model's generalizability and MARS, we trained 200 shallow CNNs with the same structure on CIFAR-10. By modifying the learning rate during training, these models have different accuracies on the test set. Then, we selected the four best and four worst performing shallow CNNs to analyze their ARS.

Table 2: The average ARS for each layer on CIFAR-10, as well as their MARS and OA.

Model	Layer1	Layer2	Layer3	Layer4	MARS	OA(%)
Good1	0.0841	0.0429	0.0413	0.0361	0.3317	75.24
Good2	0.0694	0.1070	0.0479	0.0321	0.4329	75.33
Good3	0.1008	0.0766	0.0661	0.0397	0.4605	74.84
Good4	0.0843	0.0596	0.0402	0.0538	0.3818	74.68
Mean	0.0848	0.0755	0.0518	0.0360	0.4084	75.14
Bad1	0.2075	0.1343	0.1818	0.0744	0.7689	47.08
Bad2	0.2355	0.0999	0.0935	0.0854	0.6820	46.60
Bad3	0.1944	0.1477	0.1422	0.1187	0.7741	46.35
Bad4	0.2044	0.0690	0.1929	0.0982	0.7013	46.91
Mean	0.2125	0.1273	0.1392	0.0928	0.7417	46.68

Table 2 shows the average ARS of each layer, MARS, and overall accuracy (OA). For MARS, the α of each layer are 2, 2, 1, 1. We suggest larger α for the shallow layers and smaller ones for the deep layers. Because of the ablation analysis, the shallow layer has more convolutional kernels with low ARS and contributes more to the model's generalizability. A small MARS indicates the good

generalizability of the model. By comparing the poorly performing models, the good models have lower ARS on each layer. The highest accuracy among all the well-performing models is 75.33%. Its MARS is 0.4329, and the difference in performance is slight compared to the Good1 model, which has a lower MARS. The MARS is consistent with the accuracy of the models on the test set. We also validated our metric on the CIFAR-100 dataset, and the experimental results are shown in Appendix A. For the models with low OA, their MARS is relatively high, and they have poor generalization. The experimental results indicate that the MARS obtained by each layer of ARS can be used to evaluate the model's generalizability.

3.6 Relationship Between Out-of-Domain Testing and MARS

In the out-of-domain (OOD) testing, we choose the four best and worst-performing models on corrupted CIFAR-10 among the models trained on CIFAR-10. Their ARS, MARS, and OA are shown in Table 3. The experimental results show that the models with lower MARS have higher OA. Our proposed metric is still successful in evaluating the generalization performance of the models.

Table 3: The average ARS for each layer on the corrupted CIFAR-10, as well as their MARS and OA

Model	Layer1	Layer2	Layer3	Layer4	MARS	OA(%)
Good1	0.0007	0.0064	0.1421	0.0769	0.2332	67.52
Good2	0.0007	0.0067	0.1869	0.0789	0.2807	66.77
Good3	0.0007	0.0010	0.0310	0.4053	0.4399	66.52
Good4	0.0007	0.0062	0.3657	0.0289	0.4083	66.12
Mean	0.0007	0.0047	0.1200	0.1870	0.3197	66.94
Bad1	0.0007	0.0015	0.4949	0.1144	0.6138	52.12
Bad2	0.0007	0.0019	0.5270	0.0186	0.5508	52.33
Bad3	0.0008	0.0028	0.4677	0.7939	1.2687	51.91
Bad4	0.0008	0.0020	0.0508	0.4404	0.4967	53.12
Mean	0.0007	0.0021	0.4965	0.3090	0.8111	52.12

In another set of OOD testing, we used the same model selection strategy as corrupted CIFAR-10 to find the best and worst-performing on the CIFAR-5m dataset. The results of their experiments are shown in Appendix A. Our proposed metric can still select the best-performing models. The low MARS, i.e., the model with many important representations, performs better in generalization. All these experiments show that ARS and MARS can be successfully applied to measure the model's generalizability in cross-domain scenarios.

4 Conclusion

Our proposed ARS can quantify the importance of representations through the disentangled visual representations of the convolution kernel without using a test set. The representations of the convolution kernel with low ARS are more important for the model's generalizability. In our experiments, we also explain the role of BN and CS through ARS, which can be used as a method of model explanation. With the overall ARS of the model, we also quantify the model's generalizability. The models with lower overall ARS have better generalizability under the in-domain and out-of-domain testing. Our method does not require additional data to measure the model's generalizability. It not only helps us understand the model but also helps us get a model with better generalizability. It can extend to different data-starved applications, such as few-shot learning and active learning.

However, ARS also has limitations. Comparison of ARS between different models is still a problem. But we believe when the α reasonably sums up the overall ARS of each layer, we can compare their MARS and generalization ability on different models. The definitions of RS and AR do not take into account the value of the activations. The relationship between the representations of these convolutional kernels and the selected kernel is complex. It remains a challenging problem to quantify the substitutability of the selected convolutional kernel by the higher activations. And the computation of IAM is time-consuming. In our future work, we will further explore the above issues. We also expect to use convolution kernels with low ARS as loss functions to obtain good generalizability during training.

REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/62dad6e273d32235ae02b7d321578ee8-Paper.pdf.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1019–1028. JMLR. org, 2017.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

S Hochreiter and J Schmidhuber. Flat minima. *Neural Computation*, 9(1):1, 1997.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015.

A Krizhevsky and G Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 1, 01 2009.

Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. Deep learning and quantum entanglement: Fundamental connections with implications to network design. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SywXXwJAb.

F. Meng, H. Cheng, K. Li, Z. Xu, R. Ji, X. Sun, and G. Lu. Filter grafting for deep neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to cifar-10? *CoRR*, abs/1806.00451, 2018. URL http://arxiv.org/abs/1806.00451.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Huan Wang, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. Identifying generalization properties in neural networks. 2018.

Chris J. M. Yoon, G. Hamarneh, and Rafeef Garbi. Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification. In *MICCAI*, volume 11767, pp. 365–373. Springer, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021.

A Appendix

Step Number Setting

Regarding AM and IAM, the step number setting is not sensitive to the final result. We randomly selected several convolutional kernels on layer 4 of the shallow CNN. Figure 5 shows their AM and IAM for the different number of iterations. When the number of iterations is sufficient, the change of step number does not affect the final experimental results. In the experiment, we set the iterations to 1500.

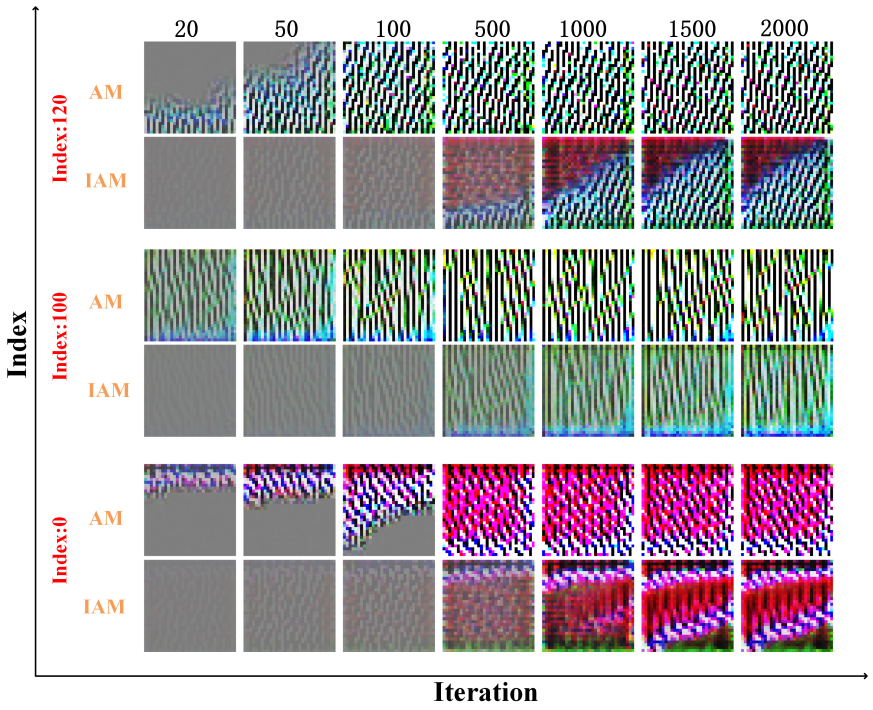


Figure 5: The visualization results of AM and IAM for the different number of iterations.

MARS on CIFAR-100

We validated our metric on the CIFAR-100 dataset and the experimental results are shown in Table 4. For the models with low OA, their MARS is relatively high, and they have poor generalization. The experimental results indicate that the MARS obtained by each layer of ARS can be used to evaluate the model's generalizability.

Table 4: The average ARS for each layer of the models on CIFAR-100, as well as their MARS and OA

Model	Layer1	Layer2	Layer3	Layer4	MARS	OA(%)
Good1	0.0250	0.0330	0.1095	0.0192	0.2448	48.54
Good2	0.0435	0.0329	0.0804	0.0268	0.2601	47.28
Good3	0.0232	0.0260	0.0938	0.0225	0.2146	49.16
Good4	0.0363	0.0256	0.0661	0.0561	0.2462	47.24
Mean	0.0306	0.0306	0.0946	0.0228	0.2398	48.33
Bad1	0.0.83	0.0489	0.0922	0.0823	0.3488	25.52
Bad2	0.0573	0.0947	0.0476	0.0507	0.4023	25.47
Bad3	0.0519	0.0306	0.0828	0.0650	0.3128	25.43
Bad4	0.0385	0.1055	0.0870	0.0568	0.4320	25.77
Mean	0.0546	0.0581	0.0742	0.0660	0.3546	25.47

MARS ON CIFAR-5M

We validated our metric on the CIFAR-5m dataset. The results of their experiments are shown in Table 5. Our proposed metric can still select the best-performing models. The low MARS, i.e., the model with many important representations, performs better in generalization.

Table 5: The average ARS for each layer of the models on the CIFAR-5m, as well as their MARS and OA

Model	Layer1	Layer2	Layer3	Layer4	MARS	OA(%)
Good1	0.0006	0.0021	0.0504	0.2126	0.2683	48.20
Good2	0.0005	0.0059	0.1077	0.2106	0.3311	45.22
Good3	0.0006	0.0010	0.2215	0.0309	0.2555	55.13
Good4	0.0006	0.0021	0.0443	0.0519	0.1016	52.48
Mean	0.0006	0.0030	0.1265	0.1514	0.2850	49.52
Bad1	0.0007	0.0024	0.4311	0.4547	0.8921	39.71
Bad2	0.0007	0.0019	0.7238	0.1340	0.8629	40.71
Bad3	0.0007	0.0420	0.4264	0.0589	0.5706	43.82
Bad4	0.0007	0.0011	0.4310	0.5452	0.9798	39.17
Mean	0.0007	0.0154	0.5271	0.2159	0.7752	41.41