APPENDIX

A. Structure of ResNet12

The detailed structure of ReNet12 is shown in Fig. A1.

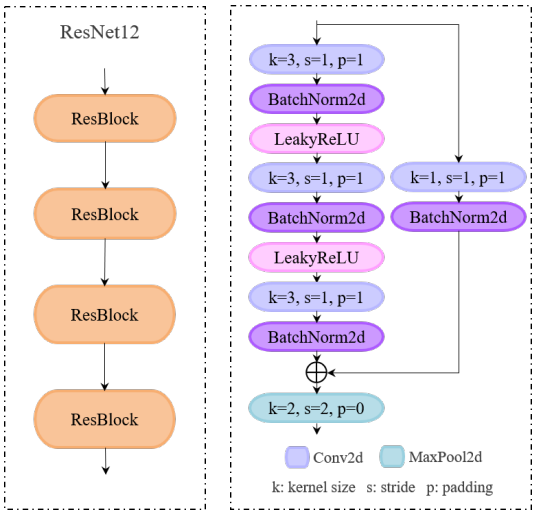


Fig. A1. Illustration of ResNet12. (a) Structure of ResNet12. (b) Structure of ResBlock.

B. Pseudocode for FSDef

The pseudocode for training and fine-tuning of FSDef is shown in Algorithm.

Algorithm 1 The training and fine-tune of FSDef.

Input: $\mathcal{D}_{Train}$, $\mathcal{D}_{Ft}$, λ_1=1, λ_2=10, λ_3=10, α=0.9
Initialize k, $r \leftarrow 1$, $\mathcal{L}_{Tr} \leftarrow []$, $\mathcal{L}_{Ft} \leftarrow []$
while $k \leq K$ **do**
  randomly sample x in $\mathcal{D}_{Train}$
    $\hat{h}$, $\hat{g} \leftarrow \mathcal{F}_\theta(x)$
    $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{Tr}$
  end
end
while $r \leq R$ **do**
  randomly sample n_s images per class in C_t from $\mathcal{D}_{Ft}$ to form a support set S
  randomly sample n_q images per class in C_t from $\mathcal{D}_{Ft}$ to form a support set Q
    $V_s \leftarrow \mathcal{F}_\theta(S)$, $V_q \leftarrow \mathcal{F}_\theta(Q)$
    caculate the centroids of classes by Eq.(9)
    $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{Ft}$
  end
end
return $\mathcal{F}_\theta$

C. The Sensibility of α

As shown in Eq. (4), the purpose of α is to prevent the attention unit from straying from its target. To explore the effects of different values of α in Eq. (4) on recognition accuracy, we change the values of α in a set of {0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8}. As shown in Fig. A2, *Pr* and *Re* increase gradually as α increases and reach a peak at α = 0.9; subsequently, *Pr* and *Re* decrease gradually as α increases. The reason for this phenomenon is that when α is too small, the regions being focused on by attention units are too similar. When α is too large, large differences in attention maps lead to decreased robustness. In the experiments, we choose the optimal value of α by validation. We can see that when α = 0.9, it can achieve the best performance.

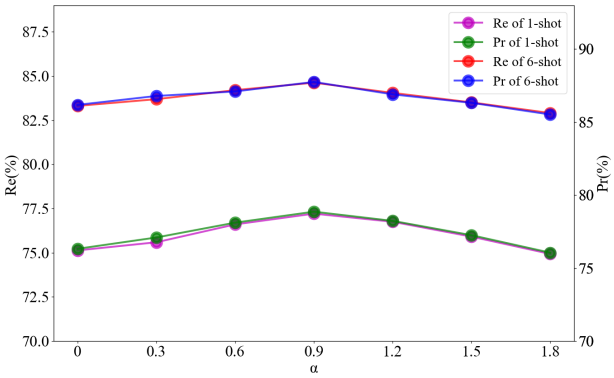
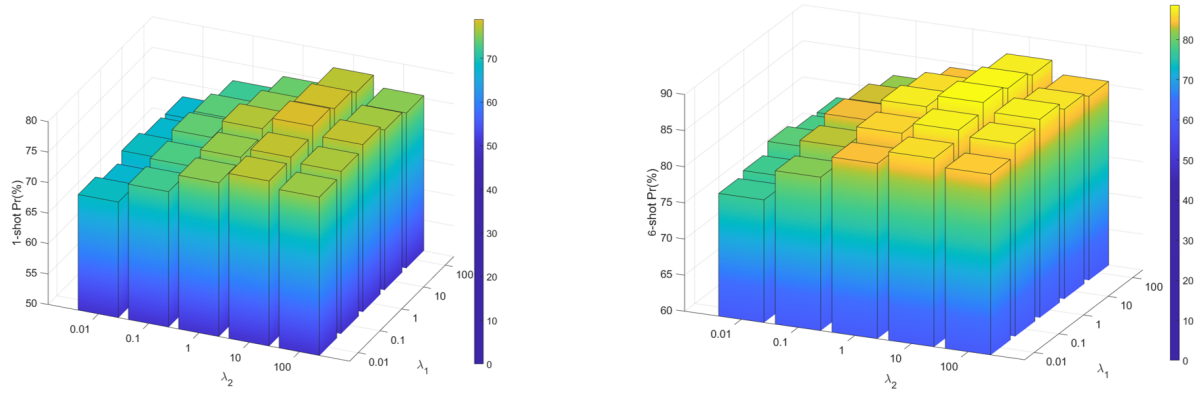


Fig. A2. *Re* and *Pr* performance with different α on insulator dateset. Among them, the line charts are the curves of *Re* of 1-shot (%), *Pr* of 1-shot (%), *Re* of 6-shot (%) and *Pr* of 6-shot (%).

D. The Sensibility of λ_1 and λ_2

In Eq. (9), the purpose of hyperparameters λ_1 and λ_2 is to balance the loss functions $\mathcal{L}_{DGA}$, $\mathcal{L}_{SLC}$, and $\mathcal{L}_{GLC}$. Specifically, λ_1 balances the objective-oriented loss function $(\mathcal{L}_{SLC} + \mathcal{L}_{GLC})$ and the attention local feature-oriented loss function LDGA, while λ_2 balances $\mathcal{L}_{SLC}$ and $\mathcal{L}_{GLC}$. To explore their impact on recognition performance, we conducted parameter sensitivity experiments using Model (d) and set α=0.9. The results are presented in Fig. A3, where we varied λ_1 between 0.01, 0.1, 0.001, 1, 10, and 100, and λ_2 between 0.01, 0.1, 0.001, 1, 10, and 100. When λ_1 was varied while λ_2 was in the range of 0.01 to 100, the *Pr* curve of 1-shot/6-shot initially increased slightly and then decreased as λ_1 increased. The best performance of *Pr* for 1-shot/6-shot is achieved when λ_1 is set to 1. We attribute this result to the fact that as λ_1 increases or decreases, the network tends to rely too heavily on single image information or group image information, limiting the improvement in accuracy. As for λ_2, when λ_1 is in the range of 0.01 to 100, the *Pr* of 1-shot/6-shot initially increased and then showes a downward trend as λ_2 increased. This is because when λ_2 exceeds 10, the model parameters update towards the direction of the local attention feature and ignore the global feature of the object, causing the attention unit to deviate from the target. When λ_2 is increased to 100, the model ignores the learning of local features, and thus, the accuracy cannot be further improved. Therefore, we set λ_1 and λ_2 to 1 and 10, respectively, for subsequent experiments.



(a) (b)

Fig. A3. Insulator defect recognition performances with different λ_1 and λ_2. (a) and (b) describe the performance of *Pr* of 1-shot and *Pr* of 6-shot respectively.

E. The Sensibility of λ_3

In the fine-tuning phase, λ_3 is used to balance $\mathcal{L}_{DGA}$ and $\mathcal{L}_{CSDM}$. To obtain the proper λ_3, λ_3 is selected from the set $\{0.01, 0.1, 1.0, 10, 100\}$ to test the effect of different λ_3 values on recognition performance. As shown in Fig. A4, with the increase in λ_3, *Pr* and *Re* also show a trend of rising and then falling. When λ_3 equals 10, *Pr* reaches the best result. This is because the object-level features are due to the region-level features, which is consistent with the choice of λ_2. λ_3 is set to 10 for subsequent experiments.

F. Comparison of FSDef with STATE-OF-THE-ART FSL Methods

To objectively verify the performance of FSDef in the defect recogniton phase, the ground trues in the insulator dataset is used as the recognition target to exclude the influence of the insulator extraction phase for the defect recognition phase. Moreover, we conducted experiments on the public data miniImageNet to validate the generalization performance

TABLE AI
COMPARISON OF FSDEF WITH STATE-OF-THE-ART FSL METHODS

Method	Dataset	1-shot (*Acc%*)	5-shot (*Acc%*)	model params (mb)	run time (ms)
Matching Networks [22]	Insulator Dataset	70.42	81.02	14.78	2.55
Prototypical Networks [21]	Insulator Dataset	71.56	81.32	8.02	1.81
Relation Networks [23]	Insulator Dataset	72.09	81.56	17.56	2.84
Baseline++ [28]	Insulator Dataset	74.10	84.08	8.02	0.96
CAN [24]	Insulator Dataset	74.01	83.35	8.04	1.60
Meta-Baseline [13]	Insulator Dataset	74.03	84.11	8.02	0.98
IEPT [14]	Insulator Dataset	75.12	85.05	8.02	2.32
FSDef (ours)	Insulator Dataset	77.03	87.12	8.62	1.03
Matching Networks [22]	miniImageNet	43.56	55.31	14.78	2.55
Prototypical Networks [21]	miniImageNet	48.70	63.11	8.02	1.81
Relation Networks [23]	miniImageNet	47.70	62.11	17.56	2.84
Activation to Parameter [33]	miniImageNet	59.60	73.74	-	-
IEPT [14]	miniImageNet	67.05	82.90	8.02	2.32
TADAM [32]	miniImageNet	58.50	76.70	-	-
MetaOptNet [31]	miniImageNet	62.64	78.63	-	-
AdaResNet [30]	miniImageNet	56.88	71.94	-	-
Meta-Baseline [13]	miniImageNet	63.17	79.26	8.02	0.98
ProtoNets + TRAML [29]	miniImageNet	60.31	77.94	-	-
Baseline++ [28]	miniImageNet	51.87	75.68	8.02	0.96
FSDef (ours)	miniImageNet	62.11	79.11	8.62	1.03

The insulator dataset is a 2-way 1/5 shot task and the insulator dataset is a 5-way 1/5 shot task. To enable consistent comparisons, we used the accuracy (*Acc*) evaluation metrics used in miniImageNet.

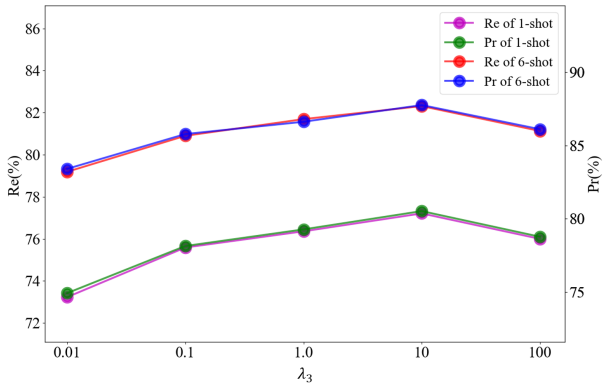


Fig. A4. *Re* and *Pr* performance with different λ_3 on insulator dateset. Among them, the line charts are the curves of *Re* of 1-shot (%), *Pr* of 1-shot (%), *Re* of 6-shot (%) and *Pr* of 6-shot (%).

of our proposed FSDef. Since the source code of some of the compared methods [29]-[33] was not available, we could not obtain the model parameters and runtimes of these methods. Therefore, the model parameters and running times of some of the methods do not appear in Table AI. It is worth noting that we use the evaluation metric accuracy used in miniImageNet as the Table AI evaluation metric for the consistency of the comparison. From Table AI, we have the following observations.

(1) From Table AI, it can be observed that our proposed FSDef achieves 62.11%/78.96% in 1-shot/5-shot, which is better than most methods on miniImageNet. Although our method does not outperform Meta-Baseline and IEPT on miniImageNet, FSDef has a clear advantage on the insulator dataset, indicating that FSDef is more effective for insulators with small inter-class variation characteristics. Additionally, it is worth noting that the source and target domains in miniImageNet are both from miniImageNet, which weakens the domain shift problem. In contrast, the source and target domains in the insulator dataset come from miniImageNet and the insulator dataset, respectively, and are very dissimilar. This puts a higher requirement on the generalization of the model, which further demonstrates that FSDef with MLC has better cross-domain generalization.