ROBUST REPRESENTATION LEARNING WITH SELF-DISTILLATION FOR DOMAIN GENERALIZATION

Ankur Singh[†]*, Senthilnath Jayavelu*[†]

[†]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore

ABSTRACT

Despite the recent success of deep neural networks, there remains a need for effective methods to enhance domain generalization using vision transformers. In this paper, we propose a novel domain generalization technique called **R**obust **R**epresentation **L**earning with Self-**D**istillation (RRLD) comprising i) intermediate-block self-distillation and ii) augmentation guided self-distillation to improve the generalization capabilities of transformer-based models on unseen domains. This approach enables the network to learn robust and general features that are invariant to different augmentations and domain shifts while effectively mitigating overfitting to source domains. To evaluate the effectiveness of our proposed method, we perform extensive experiments on PACS [1] and OfficeHome [2] benchmark datasets, as well as an industrial wafer semiconductor defect dataset [3]. The results demonstrate that RRLD achieves robust and accurate generalization performance. We observe an average accuracy improvement in the range of **1.2%** to **2.3%** over the state-of-the-art on the three datasets.

Index Terms— Domain Generalization, Vision Transformers, Self-Distillation, Data Augmentation

1. INTRODUCTION

When training a neural network (NN), the goal is to make the model learn general patterns from the provided examples so that it can make accurate predictions on new examples it has never seen before. However, the network can sometimes learn patterns specific to the samples it was trained on and fails to generalize well to new examples. This has led to the concept of domain generalization, which aims to improve the performance of neural networks on unseen domains, but it has been a challenging problem to solve. In particular, deep neural networks (DNNs) have achieved impressive results on various benchmarks, but their performance often deteriorates when applied to new domains that are different from the training data distribution. This is a major limitation of these models, as it is often infeasible or impractical to gather large amounts of annotated data for every possible data distribution in a given application area. Retraining the model with such extensive datasets is also computationally expensive, demanding significant computational resources and time. Domain generalization, thus, becomes necessary to address these challenges.

Past research on domain generalization has mainly concentrated on enhancing the generalization performance of convolutional neural networks (CNNs). One approach is to use domain adaptation techniques, which aims to align the feature distributions of the source and target domains. A popular method for domain adaptation is the use of adversarial training [4], [5]. Another approach is to use domain-invariant feature representations, which are learned by minimizing the domain shift between different domains. A popular method for learning domain-invariant features is the use of domain-adversarial neural networks (DANN) [6], [7]. Data augmentation has also been used for domain generalization by training a model on augmented versions of the existing data, thereby making the model more robust to variations in the input data [8], [9]. Finally, some works propose the use of a meta-learning approach to learning a model that can quickly adapt to new domains with only a few examples [10], [11].

Recently, vision transformers (ViTs) have shown great potential in various computer vision tasks such as image classification [12], [13], object detection [14], [15], semantic segmentation [16], [17]; however, ViTs remain underexplored in the field of domain generalization. Without the ability to effectively generalize to unseen domains and data with different distributions, they may not perform optimally in real-world scenarios, thus limiting their practical deployment.

As a result, more research is needed to develop domain generalization techniques for ViTs, addressing challenges such as domain shift, and domain-invariant feature learning. To this extent, we propose a novel approach, Robust Representation Learning with Self-Distillation (RRLD), which addresses the aforementioned problem to improve the generalization capabilities of ViTs. The key components of our proposed method include intermediate-block self-distillation (IBSD) and augmentation-guided self-distillation (AGSD), allowing RRLD to leverage the power of data augmentation and self-knowledge distillation to enhance the generalization

This study is supported by the Accelerated Materials Development for Manufacturing Program at A*STAR via the AME Programmatic Fund by the Agency for Science, Technology and Research under Grant No. A1898b0043.

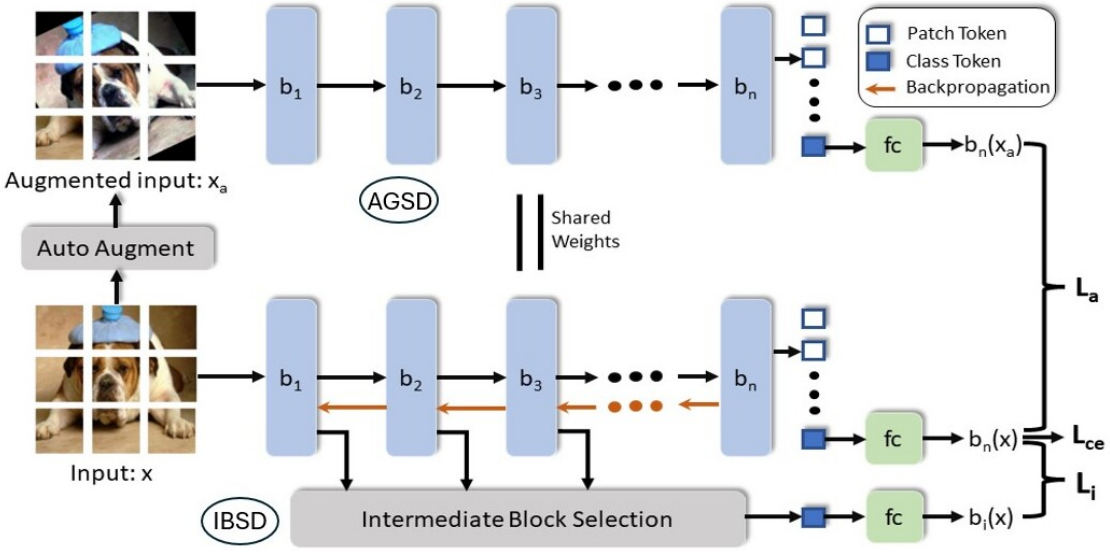


Fig. 1. RRLD: The model processes an input image x, which is first transformed by AutoAugment to produce x_a. The image x is then passed through the network to produce the output $b_n(x)$ where n is the number of blocks. A random intermediate block is selected from the network to obtain $b_i(x)$. Simultaneously, image x_a is passed through the network generating $b_n(x_a)$, with gradient computation halted during this process (Refer Algorithm 1). The losses are then computed between the three outputs.

performance of ViTs. This technique has been proven to be effective in improving domain generalization performance, as demonstrated by our experiments on two benchmark datasets: PACS [1], Office-Home [2], and an industrial wafer semiconductor defect dataset [3]. Our proposed method is an essential step towards improving the generalization capabilities of ViTs by addressing the need for effective domain generalization methods.

2. METHODOLOGY

In this section, we discuss our proposed method called 'Robust Representation Learning with Self-Distillation' (RRLD) for domain generalization. RRLD comprises: i) Intermediate-Block Self-Distillation (IBSD) and ii) Augmentation-Guided Self-Distillation (AGSD) (see Fig. 1). The utilization of these components allows for improved generalization performance on unseen domains.

2.1. Intermediate-Block Self-Distillation

In Intermediate-Block Self-Distillation (IBSD), an intermediate block is randomly sampled from the Transformer architecture and is used to make predictions [18]. The image x is passed through the sampled intermediate block b_i where $i \in \{1, 2, \ldots, n\}$, and the classification token from the block b_i is then fed into the final classifier fc to produce logits $l_i = b_i(x)$ as shown in Fig. 1. Concurrently, we also obtain logits $l_n = b_n(x)$ from the final block b_n of the vision transformer network (Fig. 1). Finally, we utilize KL Divergence to calculate the loss L_i between the predictive distributions $p(l_n)$ and $p(l_i)$ as follows:

$$L_i(p(l_n)||p(l_i)) = \sum_{j=1}^{c} \sigma(l_n/T_1)_j \log \frac{\sigma(l_n/T_1)_j}{\sigma(l_i/T_1)_j} \quad (1)$$

where $p(l_n) = \sigma(l_n/T_1)$, $p(l_i) = \sigma(l_i/T_1)$, c denotes the output dimension of fc, σ indicates the softmax function, and T_1 denotes the temperature scalar used to rescale the logits. IBSD allows us to exploit the information encoded in different parts of the network rather than relying solely on the final block. It effectively mitigates overfitting to the source domains by providing non-trivial supervisory signals for the intermediate blocks of the network, thus improving the robustness of the model.

While IBSD brings notable benefits, it only focuses on providing supervisory signals to intermediate blocks, and further improvements are sought to ensure strong and stable learning across diverse domains. To address this, we introduce Augmentation Guided Self-Distillation, as discussed in the following subsection, to further enhance the generalization capabilities of ViTs.

2.2. Augmentation-Guided Self-Distillation

The proposed self-distillation technique, Augmentation-Guided Self-Distillation (AGSD), takes advantage of the

power of data augmentation and self-knowledge distillation to improve the generalization capabilities of the model. Specifically, we utilize the Imagenet policy for AutoAugment [19] to introduce variations to the training data and then match the predictive distributions of Vision Transformers (ViTs) between the augmented and original version of the same example. This improves the generalization performance of ViTs by leveraging the knowledge gained from augmented samples of the same input.

Notably, the input image x is passed through an AutoAugment transformation, resulting in an augmented sample x_a. The original image x processed by the network produces logits l_n from the final block of the vision transformer network. The augmented sample x_a is subsequently fed into the network, generating logits $l_{a_n} = b_n(x_a)$ (see Fig. 1). During this process, gradient computation for the augmented samples x_a is halted (refer line 7 in Algorithm 1) to avoid model collapse [20], [21]. We then use the KL divergence to compute the loss L_a between the two predictive distributions $p(l_n)$ and $p(l_{a_n})$.

$$L_a(p(l_n)||p(l_{a_n})) = \sum_{j=1}^{c} \sigma(l_n/T_2)_j \log \frac{\sigma(l_n/T_2)_j}{\sigma(l_{a_n}/T_2)_j} \quad (2)$$

where $p(l_n) = \sigma(l_n/T_2)$, $p(l_{a_n}) = \sigma(l_{a_n}/T_2)$, and T_2 denotes the temperature used to rescale the logits. Augmentation-guided distillation provides a way for the network to learn to generalize to a wider range of variations in the input data, which is essential for achieving good performance on different domains.

Matching the distribution of the network between augmented versions of the same input aims to improve generalization by making the network's predictions more robust to different variations of the input. The network is trained on various augmented versions of the same example, allowing it to learn invariant representations. Furthermore, by matching the distribution of the network between augmented versions, we ensure that the network is not overfitting to specific augmentations during training, leading to better generalization on new data.

2.3. Objective function

In our proposed method, the objective function is a combination of three loss functions: the cross-entropy loss L_{ce}, the intermediate-block distillation loss L_i, and the augmentation-guided distillation loss L_a. For a given sample, the cross-entropy loss is computed between the one-hot encoded ground truth and the output obtained from the final block of the network.

The overall objective function is the sum of all three loss functions:

$$L_{total} = L_{ce} + \lambda L_i + \gamma L_a \quad (3)$$

where λ and γ are scalars multiplied to L_i and L_a to balance the contribution of the losses. Our proposed method allows the network to learn from multiple sources of information, such as the intermediate block and the augmented image, which improves the generalization and robustness capabilities of the network.

Algorithm 1 RRLD Pytorch Pseudo Code

```
# N: ViT network
for x in dataloader:
    x_a = AutoAugment(x)
    l_n, l_i = b_n(x), b_i(x)
    ŷ = softmax(l_n)

    with torch.no_grad():
        l_a_n = b_n(x_a)

    loss = L_ce(y, ŷ) + γL_a(p(l_n), p(l_a_n))
           + λL_i(p(l_n), p(l_i))

    loss.backward()

    update(N)
```

3. EXPERIMENTS AND RESULTS

3.1. Datasets

In the conducted experiments, we aimed to evaluate the performance of our proposed method for domain generalization on two benchmark datasets and an industrial dataset. Specifically, we conducted experiments on the PACS [1], and Office Home [2] datasets, popularly used for evaluating domain generalization methods. Further, we demonstrate the effectiveness of our approach in handling out-of-distribution data by performing experiments on the wafer dataset [3].

The PACS dataset includes four domains of images: Art, Cartoons, Photos, and Sketches. It has a total of 7 classes and 9,991 images. The OfficeHome dataset also includes four domains: Art, Clipart, Product, and Real. It has a total of 65 classes and 15,588 images. To demonstrate the usefulness of our approach on an industrial dataset, we conducted experiments on a wafer semiconductor defect detection dataset. This dataset contains 8,015 images belonging to 9 classes (see Fig. 2). To generate an out-of-distribution set for wafer images, we added four types of noise commonly found in semiconductor manufacturing processes (gaussian, impulse, speckle, and shot) to the images, creating a second domain (see Fig. 3). As a result, this dataset includes two domains - the original images and the noisy images - a total of 16,030 (8015×2) images.

3.2. Implementation

We segment the data from each training domain into training and validation sets, where 80% is used for training and 20% is used for validation. The validation data from each training domain are combined to create a unified validation set. The model that attains the highest accuracy on this validation set is subsequently evaluated on the target domain (test set) to report its classification accuracy. We compare the performance of two variants of Vision Transformer (ViT) models, DeiT [22], and CvT [23]. DeiT is a data-efficient transformer that employs knowledge distillation with a dedicated token achieving ViT-level accuracy on ImageNet using only ImageNet data. CvT integrates convolutions into the ViT architecture. We use DeiT-Small and CvT-21 in our experiments to provide a fair comparison with the work presented in [18]. We implemented our code in PyTorch and ran our experiments on NVIDIA RTX A6000 GPUs. We utilized the AdamW [24] optimizer and adopted the default hyperparameters proposed by [25] that include a batch size of 32 and a learning rate of 5e-05. We run our experiments three times for each test domain, using different random seeds. This allows us to obtain an average performance and also to observe any variations in the results.

In contrast to the approach used in [18], where the values of λ and T_1 vary for different domains and datasets, we maintain fixed values of $\lambda = 0.2$ and $T_1 = 5$. We also keep $\gamma = 1$ and $T_2 = 1$ constant throughout our experiments. By adopting fixed hyperparameters for our approach, we ensure a fair comparison of the model's performance across different datasets and domains. This uniformity eliminates any bias that might arise from varying hyperparameter settings and provides a consistent basis for evaluating the models. Furthermore, the use of fixed values for λ, T_1, γ, and T_2 facilitates a more straightforward and direct comparison, as the results are not influenced by hyperparameter tuning.

3.3. Evaluation

We compare our method with the state-of-the-art techniques for domain generalization utilizing vision transformer architectures as outlined in [18]. Specifically, we compare against the state-of-the-art method of ERM-ViT [18] and ERM-SDViT [18].

PACS: Table 1 compares the performance of the proposed method against ERM-ViT and ERM-SDViT on the PACS dataset for domain generalization. The results show that our method, RRLD, outperforms both ERM-ViT and ERM-SDViT in terms of accuracy on all domains, including Art, Cartoons, Photos, and Sketch. When using DeiT-Small as the backbone model, RRLD achieves an average accuracy of 88.8%, outperforming ERM-SDViT by 2.5% and ERM-ViT by a margin of 3.9%. The results are similar when using CvT-21 as the backbone model, with RRLD achieving an average accuracy of 90.5%, an increase of 2.2% over ERM-SDViT and 2.7% over ERM-ViT. These results indicate that our method, RRLD, is effective in improving the domain generalization performance on PACS.

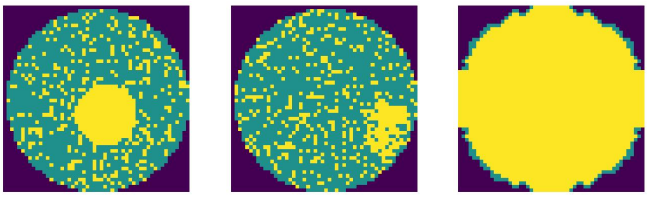


Fig. 2. Wafer Dataset Images

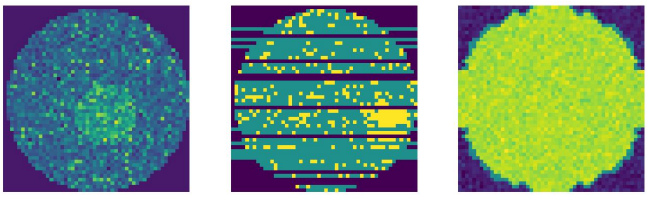


Fig. 3. Noisy Wafer Dataset Images

OfficeHome: The OfficeHome dataset presents a bigger challenge as it consists of 65 classes. Table 2 compares the performance of RRLD against the other two methods. The results indicate that RRLD achieves better performance than both ERM-ViT and ERM-SDViT in terms of average accuracy across all four domains. For instance, when using DeiT-Small as the backbone model, RRLD attains an average accuracy of 73.9%, surpassing ERM-ViT (71.4%) and ERM-SDViT (71.5%). Similarly, with CvT-21 as the backbone, RRLD obtains an average accuracy of 77.9%, outperforming both ERM-ViT (75.5%) and ERM-SDViT (75.6%). These results affirm the efficacy of our method, RRLD, in realizing enhanced domain generalization performance on the OfficeHome dataset.

Wafer Dataset: Table 3 presents the results of the proposed method, RRLD, in comparison to ERM-ViT and ERM-SDViT using the CvT-21 backbone on the Wafer dataset. It can be observed that the RRLD method outperforms both ERM-ViT and ERM-SDViT, achieving the highest accuracy of 80.6% across three runs. ERM-ViT performed better than ERM-SDViT, attaining an accuracy of 79.4%. These experiments demonstrate the superior performance of the proposed method across all three evaluated datasets.

3.4. t-SNE Visualization

In Fig. 4, we present t-SNE visualizations of class-wise feature representations obtained from ERM-SDViT (left) and RRLD (right). These visualizations correspond to the CvT-21 model trained on the photo, art, and cartoon domains, with the sketch domain serving as the target domain. Notably, the

Method	Backbone	Art	Cartoons	Photos	Sketch	Average
ERM-ViT	DeiT-Small	87.4 ± 1.2	81.5 ± 0.8	98.1 ± 0.1	72.6 ± 3.3	84.9 ± 0.9
ERM-SDViT	DeiT-Small	87.6 ± 0.3	82.4 ± 0.4	98.0 ± 0.3	77.2 ± 1.0	86.3 ± 0.2
RRLD (Ours)	DeiT-Small	**90.0** ± 0.2	**85.0** ± 0.6	**98.6** ± 0.1	**81.7** ± 0.9	**88.8** ± 0.1
ERM-ViT	CvT-21	89.0 ± 0.1	84.8 ± 0.6	98.8 ± 0.2	78.6 ± 0.3	87.8 ± 0.1
ERM-SDViT	CvT-21	90.8 ± 0.1	84.1 ± 0.5	98.3 ± 0.2	80.0 ± 1.3	88.3 ± 0.2
RRLD (Ours)	CvT-21	**91.8** ± 0.3	**86.9** ± 0.3	**98.9** ± 0.2	**84.4** ± 2.0	**90.5** ± 0.5

Table 1. Performance comparison on PACS.

Method	Backbone	Art	Clipart	Product	Real World	Average
ERM-ViT	DeiT-Small	67.6 ± 0.3	57.0 ± 0.6	79.4 ± 0.1	81.6 ± 0.4	71.4 ± 0.1
ERM-SDViT	DeiT-Small	68.3 ± 0.8	56.3 ± 0.2	79.5 ± 0.3	81.8 ± 0.1	71.5 ± 0.2
RRLD (Ours)	DeiT-Small	**71.3** ± 0.5	**59.9** ± 0.7	**81.8** ± 0.9	**82.6** ± 0.2	**73.9** ± 0.1
ERM-ViT	CvT-21	74.4 ± 0.2	59.8 ± 0.5	83.5 ± 0.4	84.1 ± 0.2	75.5 ± 0.0
ERM-SDViT	CvT-21	73.8 ± 0.6	60.7 ± 0.9	83.0 ± 0.3	85.0 ± 0.3	75.6 ± 0.2
RRLD (Ours)	CvT-21	**77.3** ± 0.5	**63.2** ± 0.3	**84.6** ± 0.7	**86.4** ± 0.4	**77.9** ± 0.4

Table 2. Performance comparison on Office Home.

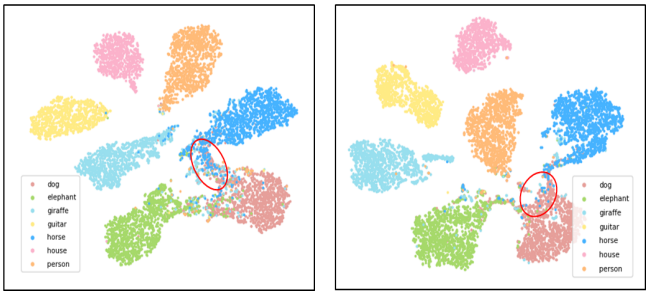


Fig. 4. Class-wise t-SNE plots obtained from ERM-SDViT (left) and RRLD (right) on the PACS dataset. ERM-SDViT exhibits some overlap in the dog and horse classes (red regions), whereas RRLD achieves well separation between these classes (red regions).

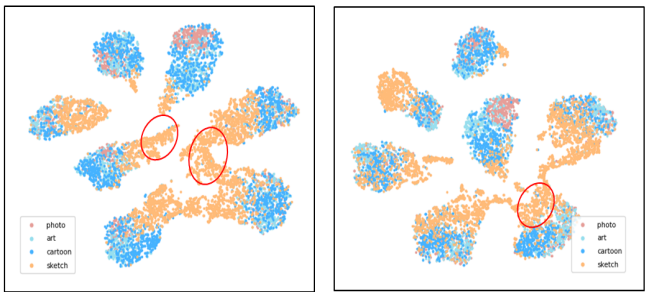


Fig. 5. Domain-wise t-SNE plots for ERM-SDViT (left) and RRLD (right) on the PACS dataset. In the t-SNE plot of ERM-SDViT, the sketch domain is separated from the other domains (highlighted in red), indicating challenges in aligning feature representations. RRLD shows better overlap among the domains, achieving a domain-invariant feature space.

Method	Backbone	Wafer Dataset
ERM-ViT	CvT-21	79.4 ± 0.6
ERM-SDViT	CvT-21	78.3 ± 0.6
RRLD (Ours)	CvT-21	**80.6** ± 0.8

Table 3. Performance comparison on Wafer dataset.

Method	Backbone	PACS Dataset
ERM-SDViT	CvT-21	88.3 ± 0.2
ERM-SDViT + AA	CvT-21	89.5 ± 0.6
RRLD (Ours)	CvT-21	**90.5** ± 0.5

Table 4. Ablation Experiment to prove the effectiveness of our approach. Autoaugment improves the performance of ERM-SDViT but is unable to outperform RRLD.

t-SNE plot for ERM-SDViT exhibits some overlap in the dog and horse classes, as indicated by the regions marked in red. In contrast, our proposed method, RRLD, showcases clear distinction and well separation between these classes, which is evident from the non-overlapping regions highlighted in red.

In Fig. 5, we visualize the domain-wise t-SNE plots for both ERM-SDViT and our proposed approach utilizing the same model as in Fig. 4. The objective of the domain-wise t-SNE plot is to observe the alignment of feature representations between different domains. Ideally, the sketch domain should overlap with other domains, signifying effective domain alignment. In the t-SNE plot of ERM-SDViT, the sketch domain is notably separated from the other domains, as indicated by the distinct cluster marked in red. This suggests that ERM-SDViT struggles to effectively align the feature representations of the sketch domain with those of other domains.

T_2 \ γ	**0.1**	**0.2**	**0.5**	**1.0**	**1.5**
1	88.9 ± 0.6	89.3 ± 0.7	89.9 ± 0.3	**90.5** ± 0.5	90.1 ± 0.9
2	88.5 ± 0.7	89.0 ± 0.5	89.8 ± 0.3	90.3 ± 0.4	89.7 ± 0.6
5	88.6 ± 0.4	88.8 ± 0.5	89.1 ± 0.7	89.8 ± 0.5	89.4 ± 0.4

Table 5. Effect of T_2 and γ on PACS Average Accuracy. RRLD maintains competitive accuracy levels with varying T_2 and γ values.

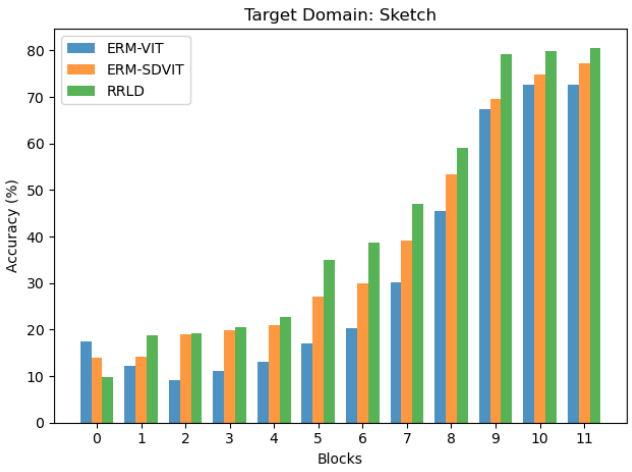


Fig. 6. Block-wise accuracy of ERM-ViT [25], ERM-SDViT [18] and our proposed method RRLD. Results are reported on the Sketch domain of the PACS dataset.

Conversely, in the t-SNE plot of our proposed method, we observe a positive outcome with significant overlap among the domains. This overlap indicates that our approach successfully aligns the feature representations of different domains resulting in a domain-invariant feature space.

3.5. Ablation Study

Here, we evaluate the effectiveness of augmentation-guided distillation (AGSD) in our proposed method, RRLD. We compare the performance of RRLD with ERM-SDViT and ERM-SDViT + AA, which is ERM-SDViT with the addition of AutoAugment (AA). The results are reported on the PACS dataset using the CvT-21 backbone in Table 4. The results of this experiment show that RRLD (uses L_{ce}, L_a and L_i) outperforms both methods, achieving 90.5% accuracy, which is an improvement of 1.0% over ERM-SDViT + AA (uses L_{ce} and L_i) and 2.2% over ERM-SDViT (uses L_{ce} and L_i). This demonstrates the effectiveness of augmentation-guided distillation in improving the generalization performance of our method. The results of this study also show that the use of data augmentation through the AutoAugment method can lead to improved performance of ERM-SDViT. However, it is unable to surpass the performance achieved by RRLD, which demonstrates its effectiveness in this domain generalization task.

In Table 5, we explore the impact of different temperature values T_2 and loss weightages γ on the average accuracy of the model on the PACS dataset. For evaluation, we select the model that achieved the best validation accuracy using the specific hyperparameter setting. Notably, for T_2 = 1, γ = 1.0, the model delivers the best average accuracy of 90.5%. As we explore higher temperature settings, the model continues to maintain competitive accuracy levels. Similar observations can be made for different values of γ, where the model demonstrates resilience to changes in the loss weightage. For instance, at γ = 0.1, T_2 = 1, the model achieves an accuracy of 88.9%, even with a higher γ value of 1.5, the model's accuracy remains notable at 90.1%.

3.6. Intermediate Block Performance

In Fig. 6, we assess ERM-ViT, ERM-SDViT, and RRLD for their intermediate block accuracies. For a fair comparison, the experiments were performed using the DeiT-Small model on the sketch domain of the PACS dataset, with the settings being the same as those used in [18]. ERM-SDViT notably enhances accuracy across intermediate blocks over ERM-ViT. However, RRLD, incorporating intermediate block self-distillation and augmentation-guided self-distillation, outperforms ERM-SDViT, demonstrating superior accuracy enhancement.

4. CONCLUSION

In this paper, we proposed a novel domain generalization approach called RRLD. Our proposed method consists of augmentation-guided self-distillation and intermediate block self-distillation, which uses a transformer network to learn domain-invariant features. We evaluated RRLD on three domain generalization datasets and compared its performance to state-of-the-art methods. Our results show that RRLD outperforms existing methods, yielding notable improvements of 2.1% on PACS, 2.3% on Office Home, and 1.2% on the Wafer dataset compared to the state-of-the-art methods. Our extensive experiments demonstrate the effectiveness of RRLD in achieving robust and accurate generalization performance. The proposed method is simple yet effective, and it could be helpful for a wide range of domain generalization tasks.

5. REFERENCES

[1] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[2] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[3] Junliang Wang, Chuqiao Xu, Zhengliang Yang, Jie Zhang, and Xiaoou Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 4, pp. 587–596, 2020.

[4] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. 2018, vol. 31, Curran Associates, Inc.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[7] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2551–2559.

[8] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese, "Generalizing to unseen domains via adversarial data augmentation," *Advances in neural information processing systems*, vol. 31, 2018.

[9] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang, "Learning to generate novel domains for domain generalization," in *European conference on computer vision*. Springer, 2020, pp. 561–578.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[11] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 558–567.

[14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[15] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[16] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[17] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12077–12090, 2021.

[18] Maryam Sultana, Muzammal Naseer, Muhammad Haris Khan, Salman Khan, and Fahad Shahbaz Khan, "Self-distilled vision transformer for domain generalization," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, December 2022, pp. 3068–3085.

[19] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[20] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[21] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13876–13885.

[22] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10347–10357.

[23] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.

[24] Ilya Loshchilov and Frank Hutter, "Fixing weight decay regularization in adam," 2018.

[25] Ishaan Gulrajani and David Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.