Adaptive Network Combination for Single-Image Reflection Removal: A Domain Generalization Perspective

Ming Liu$^{1,2}$ Jianan Pan$^1$ Zifei Yan$^1$ Wangmeng Zuo$^1$ (✉) Lei Zhang$^2$

$^1$Harbin Institute of Technology, Harbin, China, $^2$The Hong Kong Polytechnic University, Hong Kong, China

Abstract

Recently, multiple synthetic and real-world datasets have been built to facilitate the training of deep single image reflection removal (SIRR) models. Meanwhile, diverse testing sets are also provided with different types of reflections and scenes. However, the non-negligible domain gaps between training and testing sets make it difficult to learn deep models generalizing well to testing images. The diversity of reflections and scenes further makes it a mission impossible to learn a single model being effective to all testing sets and real-world reflections. In this paper, we tackle these issues by learning SIRR models from a domain generalization perspective. Particularly, for each source set, a specific SIRR model is trained to serve as a domain expert of relevant reflection types. For a given reflection-contaminated image, we present a reflection type-aware weighting (RTAW) module to predict expert-wise weights. RTAW can then be incorporated with adaptive network combination (AdaNEC) for handling different reflection types and scenes, i.e., generalizing to unknown domains. Two representative AdaNEC methods, i.e., output fusion (OF) and network interpolation (NI), are provided by considering both adaptation levels and efficiency. For images from one source set, we train RTAW to only predict expert-wise weights of other domain experts for improving generalization ability, while the weights of all experts are predicted and employed during testing. An in-domain expert (IDE) loss is presented for training RTAW. Extensive experiments show the appealing performance gain of our AdaNEC on different state-of-the-art SIRR networks. Source code and pre-trained models will available at https://github.com/csmliu/AdaNEC.

1. Introduction

When capturing images through glass, undesired reflection by the glass surface is one of the key factors causing image quality degradation, and the existence of these reflections also hampers many other vision applications such as image classification and autonomous driving. In recent years, driven by the development of deep neural networks (DNNs), learning-based methods have achieved continuous improvements on single-image reflection removal (SIRR) tasks [6, 9, 10, 16, 21, 25, 31, 44, 49, 50, 53, 55, 56].

For training these methods, generally a large amount of pair-wise images are required. To this end, several real-world training sets [21, 49, 55] have been collected. However, due to the difficulty and cost to collect real-world reflection and clean image pairs, these datasets contain only a limited number of images. As a remedy, multiple reflection synthesis methods [9, 16, 50, 53, 55, 56] are introduced, which can be used for joint-training with real-world datasets. Albeit the tremendous efforts to synthesize or render realistic reflection-contaminated images, there is still a large gap between synthetic images and real ones. Besides, the real-world reflection sets are collected under distinct circumstances, leading to diverse reflection types with varying light conditions, glass types, and photographing parameters.

The non-negligible domain gaps between training and testing sets make the generalization to testing images difficult, and the diversity of reflections and scenes in real-world images further makes it a mission impossible to learn a single model for all testing sets. As a result, previous reflection removal methods usually have to seek a balanced yet sub-optimal solution on diverse reflection types, remaining some questions to tap the potential of SIRR methods: i) how to adjust the learning policy to train deep models generalizing well to testing images and ii) how to tune the model adaptively during inference to deal with various reflection types in different testing images or datasets.

Aiming to learn a model that can generalize to an unseen target domain from one or more different yet related source domains, domain generalization (DG) has been successfully applied in many high-level vision tasks such as classification and semantic segmentation [46, 57]. In this paper, we introduce domain generalization into SIRR to answer the above questions, where the rationality and feasibility are shown via theoretical analysis and statistical evaluation (see Sec. 3.1 for details). In particular, for each training
set (source domain), we train a specific SIRR model, which serves as the domain expert of relevant reflection types. Then, for a given testing image (target domain), these models can cooperate to perform reflection removal according to their expertise level on the target reflection type.

In order to evaluate the expertise level, we present a reflection type-aware weighting (RTAW) module, which predicts an expert-wise weight for each model. During training, we follow the leave-one-domain-out (LODO) strategy [22], i.e., for images from one source set, RTAW only predicts expert-wise weights of other domain experts, and this source set serves as a pseudo target set for improving generalization ability. RTAW can then be incorporated with the proposed adaptive network combination (AdaNEC) to handle different reflection types and scenes, i.e., generalizing to unknown domains, where the weights of all experts are predicted and employed. An in-domain expert (IDE) loss is introduced for training RTAW. Pert (IDE) loss is introduced for training RTAW. While in the NI manner, we interpolate parameters of the experts, which boosts the performance with minor computation costs. It is worth noting that images from the same testing set can be regarded as a domain due to the similar capturing circumstance. Therefore, we can further apply the domain-level AdaNEC, which uses the average RTAW weights to get rid of the interference factors in certain images and achieves a more stable prediction.

To evaluate the proposed method, we take three state-of-the-art methods (i.e., ERRNet [49], IBCLN [21], and RAGNet [25]) as the backbone reflection removal networks, and incorporate our proposed AdaNEC method into these backbone models1. Both quantitative and qualitative experimental results show that the AdaNEC-boosted models perform favorably against their original joint-training counterparts, indicating the effectiveness of our AdaNEC method.

In general, the main contribution of this paper involves:

- Based on the formation process of reflections, we propose to learn SIRR models from a domain generalization perspective instead of the joint-training scheme in existing methods.
- Given SIRR models as domain experts, a reflection type-aware weighting (RTAW) module is proposed to predict the expert-wise weights, which can be incorporated with adaptive network combination (AdaNEC) for generalizing to unseen domains. An in-domain expert (IDE) loss is introduced for training RTAW.
- Two representative AdaNEC methods, i.e., output fusion (OF) and network interpolation (NI), are provided considering both adaptation levels and efficiency.
- Extensive experimental results on different state-of-the-art SIRR networks show that the proposed AdaNEC can achieve appealing performance gains in comparison with their joint-training counterparts.

2. Related Work

Reflection removal methods generally can be divided into two categories, i.e., single-image reflection removal (SIRR) and multi-image reflection removal (MIRR). We focus on learning-based SIRR in this paper, some MIRR [1, 11, 14, 17, 19, 23, 26, 32–34, 38, 39, 51, 52, 54] and traditional SIRR [2, 20, 24, 35, 45] methods are listed for reference. In the following, we first review the learning-based SIRR methods and relevant datasets, then briefly introduce the domain generalization and network interpolation methods.

2.1. Learning-based SIRR Methods

For removing reflection accurately and producing clear results, Fan et al. [9] designed a gradient loss to constrain the transmission edge maps, Zhang et al. [55] further proposed an exclusion loss to minimize the gradient correlation of reflection/transmission layers. To alleviate the misalignment problem, Wei et al. [49] leveraged the highest-level VGG [37] features which are insensitive to misalignment. Reflection distribution is also considered to facilitate SIRR, Dong et al. [6] predicted a reflection confidence map which is integrated into a composition loss, while Li et al. [25] defined an explicit mask loss by leveraging reflection strength. In this paper, we enhance the SIRR task from a domain generalization perspective, which is orthogonal to and cooperates well with the above methods.

Various network structures have also been explored in SIRR. CEILNet presented by Fan et al. [9] is a two-stage architecture, where the edge maps are predicted before the estimation of the transmission layer. The two-stage setting is followed and developed by DMGN [10], Dong et al. [6], and RAGNet [25], and they estimate the reflection layer in the prior stage instead. Zhang et al. [55] stacked the features extracted by multiple layers of a pre-trained VGG-19 [37] model at the beginning of SIRR models, which is retained in a series of subsequent works [10, 49]. Yang et al. [53] proposed to predict the reflection layer and the transmission layer iteratively, and IBCLN [21] further introduced recurrent neural networks, which is also exploited in [6, 10]. Prasad et al. [31] proposed a multi-scale structure which starts from a low-resolution input and progressively grows into the desired resolution. It is worth noting that the backbones in this paper can cover most of these architectures.
2.2. Single-Image Reflection Removal Datasets

In the early studies on learning-based SIRR, due to the lack of real-world training images, synthesis methods were introduced to train the SIRR models [9, 53]. Subsequent works also tried to synthesize more realistic reflection-contaminated images by considering camera pose [55], generative adversarial networks [50], rendering software [16], and physical models [56].

Meanwhile, several real-world datasets have been collected. Zhang et al. [55], Li et al. [21], and Wei et al. [49] captured three real-world datasets with 109, 220, and 450 image pairs, respectively. For testing, a real-world benchmark dataset termed as SIR2 has been collected [43], which is composed of three subsets, i.e., Wild, Solid, and Postcard. Interestingly, Wan et al. [44] collected 3,250 reflection layers by putting a black piece of paper behind the glass, which provides another way for synthesis. Lei et al. [18] collected a perfectly aligned testing dataset by capturing raw images and obtained the transmission layer by subtracting the reflection layer from the reflection-contaminated image.

In summary, multiple datasets have been built with diverse reflection types and scenes, and the non-negligible domain gaps make it a critical problem to generalize and adapt to various domains for enhancing SIRR.

2.3. Domain Generalization

Domain generalization (DG), also known as out-of-distribution (OOD) generalization, aims at learning a model for generalizing to OOD samples, has attracted upsurging attention since it was introduced by Blanchard et al. [3]. Recent DG methods improve generalization from one or more aspects of data manipulation, representation learning, and learning strategies. The category most relevant to this paper in DG is ensemble learning [5, 7, 27, 47, 58]. These methods generally train domain-specific branches or networks, and fuse the features [7, 47] or predictions [27, 58] via a predicted weight during inference. Ding et al. [5] learned a domain-invariant model by distilling domain-specific models under structured low-rank supervision. Most DG methods focus on high-level vision tasks, in this paper, we explore the SIRR task from a domain generalization perspective, hoping to encourage the application of DG methods in low-level vision tasks. Please refer to [46, 57] for a comprehensive review of DG methods.

2.4. Network Interpolation

Checkpoint ensemble [4] is a commonly used network interpolation strategy in recent years, which interpolates the parameters from different iterations for more stable training and enhanced performance. Chen et al. [36] proposed to learn the residual of parameters in a pre-trained model with a different yet related learning objective, and performed model interpolation in the objective-space. Wang et al. [48] analyzed the interpolation between two separately trained networks, and achieved continuous imagery effect transition. However, the interpolation weights of these methods should be manually tuned. In this paper, we leverage the power of domain generalization and predict the interpolation weights for generalizing to unseen domains.

3. Method

In this section, we first provide an theoretical analysis and show the domain gaps statistically, which indicates the rationality and feasibility to solve the SIRR problem from a domain generalization perspective. Then, given domain experts of various reflection types, a reflection type-aware weighting (RTAW) module is deployed for predicting the expert-wise weights, and we propose an in-domain expert loss for training RTAW. An adaptive network combination (AdaNEC) method is then incorporated with RTAW for generalizing to unknown target domains.

3.1. Domain Gaps between SIRR Datasets

In this paper, we propose to improve SIRR from a DG perspective. Then, here comes a premise question, i.e., to what extent are these datasets different from each other? To answer this question, we follow Torralba et al. [40] and design a simple experiment to statistically evaluate the domain gaps between different datasets. Specifically, the training sets employed in ERRNet [49], i.e., Syn_CEIL [9], Real89 [55], and Unaligned [49], are randomly split into training and testing sets at a ratio of around 8:2. Then the pairs of reflection-contaminated image and the corresponding dataset index label are used to train a small-scale classification model for telling which dataset a testing image is from. The classification model is composed of five convolution layers and one fully-connected layer. As shown in Tab. 1, the model can achieve 86% overall accuracy. Even with only 71 training samples, it still achieves ~60% accuracy on Real89 [55] dataset. The result clearly shows the domain gaps between SIRR datasets, making it a promising way to apply domain generalization for enhancing SIRR methods.

However, as shown in [46, 57], most DG methods focus on high-level vision tasks like image classification and semantic segmentation. Therefore, we provide a theoretical analysis to discuss the feasibility of leveraging DG methods for SIRR. As revealed in [29] and [56], the acquisition...
of reflection-contaminated images can be formulated by

\[ I = \mathcal{I}(\Omega T^r + \Phi R^r), \tag{1} \]

where \( T^r \) and \( R^r \) are the raw signals of transmission and reflection layers, \( \Omega \) and \( \Phi \) represent the refractive and reflective amplitude coefficient maps relevant to the glass characteristics and photographing parameters, and \( \mathcal{I} \) denotes the camera image signal processing (ISP) pipeline to obtain sRGB images. Most synthesis methods [9, 16, 53, 55, 56] also approximate this formulation. In other words, considering the parameters in Eqn. (1), a manifold is spanned by the images with diverse reflection types, which settles the foundation of our DG-based solution.

3.2. Overall Solution

Since most previous SIRR methods are trained with a mixture of various datasets, they have to compromise between different domains, which is often not within the scope of target testing sets (see Fig. 1(a)). Instead, suppose that there are \( N \) training datasets, where the \( i \)-th dataset is denoted by \( S_i \), we train an SIRR model for each of the dataset. In this case, we can obtain \( N \) domain experts \( \{ G_i \}_{i=1}^{N} \) (see the Red, Cyan and Green regions in Fig. 1(b)), where the expert \( G_i \) is specialized on the relevant reflection types in \( S_i \). Then, for a reflection-contaminated image \( I \), we can combine these domain reflections for better reflection removal according to certain weights (e.g., \( w_1 \sim w_3 \) in Fig. 1(b)).

A reflection type-aware weighting (RTAW) module is introduced to evaluate the expertise level and predict the expert-wise weights of each model \( G_i \) on \( I \), where an adaptive network combination (AdaNEC) method is incorporated to better leverage the predicted RTAW weights. In the following, we explain the design of RTAW and AdaNEC in detail.

3.3. Reflection Type-aware Weighting Module

The reflection type-aware weighting (RTAW) module is comprised of two parts, i.e., the feature extraction networks and a cross-domain attention module. As shown in Fig. 2, a feature extractor \( \mathcal{F} \) is deployed to extract the universal feature vector \( q \) from the reflection-contaminated input \( I \), i.e., \( q = \mathcal{F}(I) \). Then, for each domain expert \( G_i \), we obtain a domain-specific feature vector \( k_i \) via the corresponding extractor \( \mathcal{F}_i \), i.e., \( k_i = \mathcal{F}_i(I) \). Inspired by the self-attention layer in Transformer [41], we present a cross-domain attention module (CDAM) to evaluate the expertise level of each domain expert by matching their feature vector \( k_i \) with the common feature \( q \), i.e.,

\[ v_i = (W_k^\top k_i) \odot (W_q^\top q), \tag{2} \]

where \( W_k \) and \( W_q \) are parameters of the fully-connected layers for \( k \) and \( q \) respectively, and \( W_k \) is shared by all of \( k_1 \) to \( k_N \), \( \odot \) denotes inner-product. Then for a testing image, all domain experts are exploited, while the RTAW weights can be calculated by using the expertise level, i.e.,

\[ w = \sigma(v), \tag{3} \]

where \( \sigma \) is the softmax operation and \( v = \{v_i\}_{i=1}^{N} \).

3.4. Learning Objective

Apart from the reflection removal losses, an in-domain expert (IDE) loss is introduced for training the proposed RTAW module, and we introduce the two parts of the learning objective in the following. Note that parameters of the domain experts are fixed when training the RTAW module.

Reflection Removal Losses. When training RTAW, there is no extra dataset to serve as the target domain, thus we follow the leave-one-domain-out strategy [22]. Specifically, given a training sample \( I_{tr} \) from the \( i \)-th training set \( S_i \),
we denote by $\mathbf{T}$ and $\mathbf{R}$ the ground-truth transmission and reflection layers, respectively. The $i$-th source domain will serve as a pseudo target domain during this iteration. Only domain experts except for $G_i$ will be employed, i.e.,

$$
\mathbf{w}_i^G = \sigma(\mathbf{v}_i^G),
$$

(4)

$$
\hat{\mathbf{T}}_i^G = \sum_{j \neq i} w_{j} \cdot G_j(\mathbf{I}_{tr}),
$$

(5)

where $\mathcal{C}$ denotes the complement set and $\mathbf{v}_i^G = \{v_j\}_{j \neq i}$, $\mathbf{w}_i^G = \{w_j\}_{j \neq i}$. Note that the reflection layer $\hat{\mathbf{R}}_i^G$ is generated in a similar way if the backbone model requires it for loss calculation. In the following, we briefly summarize the reflection removal losses.

**Fidelity Loss.** Generally $\ell_1$ or $\ell_2$ is employed as reconstruction loss on the transmission layer or reflection layer,

$$
\mathcal{L}_{fid} = \left\| \hat{\mathbf{Y}}_i - \mathbf{Y} \right\|_p, \mathbf{Y} \in \{\mathbf{T}, \mathbf{R}, \nabla \mathbf{T}\},
$$

(6)

where $\nabla$ means gradient calculation operation, $p \in \{1, 2\}$, and $\mathbf{Y}$ denotes the corresponding ground-truth of $\hat{\mathbf{Y}}_i^G$.

**Reconstruction Loss.** For synthetic pairs, the synthesis methods can be leveraged for better training, and the reconstruction loss is defined as

$$
\mathcal{L}_{rec} = \left\| \text{Syn}(\hat{\mathbf{T}}_i^G, \hat{\mathbf{R}}_i^G) - \mathbf{I}_{tr} \right\|_p,
$$

(7)

where $\text{Syn}$ denotes the differentiable synthesis methods.

**VGG-based Loss.** The VGG [37] architecture has been widely used for loss function design, i.e.,

$$
\mathcal{L}_{VGG} = \sum_i \lambda_i \left\| \phi_l(\hat{\mathbf{T}}_i^G) - \phi_l(\mathbf{T}) \right\|_p,
$$

(8)

where $\phi_l$ denotes the $l$-th layer of the VGG model, $\lambda_i$ is the corresponding weight. The VGG-based loss is generally used following perceptual loss [15] ($\mathcal{L}_{\text{percept}}$). In [49], Wei et al. proposed to use the “conv5_2” feature for alignment-invariant loss ($\mathcal{L}_{\text{inv}}$).

**Adversarial Loss.** Adversarial loss [13] is an effective approach to enhancing the output image quality. Given a discriminator $D$, the adversarial loss can be defined by

$$
\mathcal{L}_{adv} = - \log D(\hat{\mathbf{T}}_i^G).
$$

(9)

To sum up, the reflection removal loss $\mathcal{L}_{RR}$ can be defined by combining these loss functions or their variants,

$$
\mathcal{L}_{RR} = \lambda_{\text{fid}} \mathcal{L}_{fid} + \lambda_{\text{rec}} \mathcal{L}_{rec} + \lambda_{\text{percept}} \mathcal{L}_{\text{percept}}
+ \lambda_{\text{inv}} \mathcal{L}_{\text{inv}} + \lambda_{\text{adv}} \mathcal{L}_{adv},
$$

(10)

where $\lambda_{\text{fid}}, \lambda_{\text{rec}}, \lambda_{\text{percept}}, \lambda_{\text{inv}},$ and $\lambda_{\text{adv}}$ are hyper parameters following the original settings in the backbone methods. Note that they can be zero if the corresponding loss function is not applied in a backbone method.

**In-domain Expert Loss.** Suppose that when the $i$-th source domain $S_i$ serves as the pseudo target domain, what if the domain expert $G_i$ is also deployed? Since $G_i$ is trained on $S_i$, it can be regarded as an in-domain expert. Therefore, we can require that the final results mostly rely on it. To this end, we propose an in-domain expert (IDE) loss, which is defined based on the cross-entropy (CE) loss [12, 28] by

$$
\mathcal{L}_{\text{IDE}} = CE(\mathbf{w}, i) = - \log w_i.
$$

(11)

We empirically find that the IDE loss not only promotes the reflection removal performance but also makes the training more stable. Further discussions are given in Sec. 5.1.

The overall learning objective for training RTAW is

$$
\mathcal{L} = \mathcal{L}_{RR} + \lambda \mathcal{L}_{\text{IDE}},
$$

(12)

where $\lambda$ is set to 0.1 for all three backbone methods. Please refer to the supplementary material for more details about the loss function configuration in this paper.

### 3.5. Adaptive Network Combination

As shown in Fig. 1(b), the RTAW weights can be employed for adaptively generalizing to target domains via adaptive network combination (AdaNEC). In this paper, considering both adaptation level and efficiency, we provide two representative AdaNEC methods termed output fusion (OF) and network interpolation (NI), respectively.

**Output Fusion (OF).** The OF manner is similar to the training procedure of RTAW, except that all domain experts are employed during inference to generate the output, i.e.,

$$
\hat{\mathbf{T}}_{OF} = \sum_{i=1}^{N} w_i \cdot G_i(\mathbf{I}),
$$

(13)

where the results generated by the domain experts are fused as final output. However, such a method requires multiple model inference, which brings heavy extra computation burden and time consumption.

**Network Interpolation (NI).** To alleviate this problem, the NI manner interpolates the model parameters in advance, and the inference procedure is processed with the fused model parameters, which can be formulated by

$$
\hat{\mathbf{T}}_{NI} = \left( \sum_{i=1}^{N} w_i \cdot G_i \right)(\mathbf{I}),
$$

(14)

where only a few extra computations (i.e., the lightweight RTAW module and parameter interpolation) are introduced. Besides, as shown in Sec. 3.1, generally the reflection-contaminated images in a dataset form a domain. In this case, we can use the average RTAW weights of a dataset for more stable prediction and apply domain-level AdaNEC.

| Backbone | Training Datasets | Testing Datasets |
|----------|-------------------|------------------|
| ERNNet [49] | SynCHEL, Real89, Unaligned | Real20, SIR² |
| IBCLN [21] | SynCHEL, Real89, Nature200 | Real20, SIR², Nature20 |
| RAGNet [25] | SynCHEL, SynCHEL, Real89 | Real20, SIR² |

Table 2. Dataset configuration of the backbone methods.
Table 3: Quantitative comparison against the state-of-the-art backbones. For calculating the PSNR and SSIM indices, we follow the testing protocol of the backbone methods. Note that the results of ERRNet [49] and RAGNet [25] on Nature20 are given for reference (highlighted with gray shadow), and the average indices are calculated on the other four datasets (Real20, Wild, Postcard, and Solid) according to the official setting. The best results are marked with bold, and the image amount of each dataset is provided in the parentheses.

| Methods       | Real20 (20) | Wild (55) | Postcard (199) | Solid (200) | Nature20 (20) | Average (474/494) |
|---------------|-------------|-----------|----------------|-------------|---------------|------------------|
|               | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ |
| ERRNet [49]   | 22.08 | 0.781 | 25.13 | 0.899 | 22.76 | 0.864 | 24.62 | 0.898 | 20.86 | 0.757 | 23.79 | 0.877 |
| ERRNet_{OF}   | 22.80 | 0.790 | 25.26 | 0.890 | 23.08 | 0.874 | 25.26 | 0.904 | 20.99 | 0.768 | 24.24 | 0.885 |
| ERRNet_{NI}   | 22.81 | 0.791 | 25.70 | 0.895 | 23.56 | 0.884 | 25.13 | 0.902 | 21.20 | 0.771 | 24.44 | 0.889 |
| IBCLN [21]    | 21.86 | 0.762 | 24.71 | 0.886 | 23.39 | 0.875 | 24.87 | 0.893 | 24.03 | 0.787 | 24.10 | 0.877 |
| IBCLN_{OF}    | 22.52 | 0.789 | 25.77 | 0.897 | 24.27 | 0.889 | 25.24 | 0.900 | 24.74 | 0.820 | 24.78 | 0.888 |
| IBCLN_{NI}    | 22.04 | 0.782 | 25.35 | 0.894 | 23.34 | 0.887 | 24.85 | 0.897 | 24.59 | 0.818 | 24.17 | 0.885 |
| RAGNet [25]   | 22.95 | 0.793 | 25.52 | 0.880 | 23.67 | 0.879 | 26.15 | 0.903 | 21.21 | 0.765 | 24.90 | 0.886 |
| RAGNet_{OF}   | 23.34 | 0.807 | 25.85 | 0.896 | 25.20 | 0.903 | 26.17 | 0.908 | 21.48 | 0.776 | 25.60 | 0.900 |
| RAGNet_{NI}   | 23.18 | 0.802 | 26.25 | 0.899 | 24.90 | 0.906 | 25.66 | 0.903 | 21.44 | 0.777 | 25.31 | 0.900 |

4. Experiments

4.1. Implementation Details

Datasets. Since the core idea of this work is to make better use of the training sets and enhance SIRR from a domain generalization perspective, we follow the training configuration of the backbone methods and have no requirements on extra training datasets. In the following, the relevant training and testing datasets are briefly introduced.

SynCEIL - The synthetic dataset proposed in CEIL-Net [9], where 7,643 image pairs are selected from the PASCAL VOC dataset [8], and each pair serves as transmission and reflection layer, respectively. Blurriness and luminance of the reflection layer are considered for synthesis.

SynZhang - The synthetic dataset proposed by Zhang et al. [55], where 13,700 indoor-outdoor image pairs are collected from Flickr. The synthesis method is adapted from SynCEIL and considers the angle for capturing images.

Real89 & Real20, Nature200 & Nature20 - Two well aligned real-world datasets captured by Zhang et al. [55] and Li et al. [21], where Real89 and Nature200 are used for training while Real20 and Nature20 are used for testing.

Unaligned - The training set collected by Wei et al. [49], where the reflection-contaminated images and transmission layers are mildly aligned. Note that there are 250 pairs captured by DSLR camera and 200 by mobile phone, we use the 250 DSLR pairs following the official code of [49].

SIR2 - A benchmark dataset4 collected by Wan et al. [43], which is composed of three subsets, i.e., Wild (55), Solid (200), and Postcard (199), where the Wild subset is captured in the wild and the other two subsets are collected in a controlled laboratory environment.

- https://github.com/WeiYu_dataset/Real89
- https://github.com/Vandermode/ERRNet
- https://github.com/JHL-HUST/IBCLN
- https://github.com/liyucs/RAGNet

The training/testing dataset configuration of the backbone methods is summarized in Tab. 2.

Backbones and Implementation Details. In this paper, we use three state-of-the-art methods as backbone models, i.e., ERRNet5 [49], IBCLN6 [21], and RAGNet7 [25]. As introduced in Sec. 2.1, they are the best performed SIRR methods with source code and pre-trained model available, and most of recent structure design categories are covered by these methods. All the three methods are implemented in the PyTorch framework. The experiments are conducted on a PC with an Nvidia RTX 3090 GPU. The source code and pre-trained models will be publicly available.

Testing Protocols. As shown in Tab. 2, the backbone methods are trained and evaluated on various datasets. For a fair comparison, apart from the proposed IDE Loss $L_{IDE}$, we train our method with only the training sets and loss functions it used given a backbone model (please refer to the supplementary material for more details), and evaluate the model performance following its testing protocol. Since only the model trained with aligned image pairs is given, we fine-tuned the ERRNet model following official instructions, and achieved an average PSNR index of 23.79 dB, which is higher than the performance (23.59 dB) reported by Wei et al. [49]. We also note that since the image size of Real20 is too large for inference, following the backbone methods, the shorter side of Real20 images is resized to 512, 400, and 300 pixels for evaluating ERRNet [49], IBCLN [21], and RAGNet [25], respectively.

4.2. Comparison with Backbone Methods

As shown in Tab. 3, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) indices of two
AdaNEC methods are provided for comparing against each backbone method, which are denoted by the subscripts $OF$ and $NI$, respectively. We can see that, the output fusion ($OF$) manner generally can achieve an average PSNR gain of about $0.45 \sim 0.7$ dB, and the performance on all datasets is boosted to some extent, showing the effectiveness of our AdaNEC method. On the other hand, considering the computation amount and inference efficiency, the network interpolation ($NI$) manner also achieves considerable performance gain on most datasets.

It is worth noting that, both AdaNEC$_{OF}$ and AdaNEC$_{NI}$ achieve higher SSIM index than the backbone methods in all conditions, showing the promoted reflection removal ability. For ERRNet [49] and RAGNet [49], we also evaluate them on the Nature20 dataset, which is not considered by these methods. As shown by the gray shaded regions in Tab. 3, the proposed method can generalize better to the unseen extra testing set.

As shown in Figs. 3 to 5, we also evaluate the proposed method qualitatively. It can be seen that, when incorporated with our AdaNEC method from a domain generalization perspective, the backbone methods can be enhanced in terms of their reflection removal ability and output image quality. Please zoom in for better observation, and refer to the supplementary material for more qualitative results.

5. Ablation Study and Analysis

5.1. Ablation Study

In this part, we conduct some ablation studies to show the contribution and influence of different parts of the proposed AdaNEC method. Considering the training efficiency and relatively simple structure, the ablation studies are conducted on ERRNet [49].

Performance of domain experts. As shown in Tab. 4, when trained with a single dataset, the performance of the domain experts may be enhanced on some datasets (e.g., generally the performance on Real20 will be better when trained with Real89, since they are the training/testing set from the same dataset), and the final output fusion ($OF$) results can approximate these enhanced performance. For Solid, all these domain experts perform slightly worse than the backbone model. However, a decent performance gain is obtained when incorporating with AdaNEC, which shows the generalization ability of the proposed method. We also show the visual results in Fig. 6, and the AdaNEC-boosted models can generate better reflection removal results.
Table 5. Ablation studies on RTAW module. The indices better than the backbone model are highlighted with green and those worse than the backbone model are highlighted with red (the difference of PSNR < 0.05dB and SSIM < 0.004 has been ignored).

| Methods       | Real20  | Wild   | Postcard | Solid  | Average |
|---------------|---------|--------|----------|--------|---------|
|               | PSNR↑   | SSIM↑  | PSNR↑    | SSIM↑  | PSNR↑   | SSIM↑  | PSNR↑   | SSIM↑  | PSNR↑   | SSIM↑  |
| ERRNet [49]   | 22.08   | 0.781  | 25.13    | 0.889  | 22.76   | 0.864  | 24.62   | 0.898  | 23.79   | 0.877  |
| ERRNet_{avg}  | 21.50   | 0.760  | 25.63    | 0.899  | 22.95   | 0.880  | 25.26   | 0.904  | 24.17   | 0.887  |
| ERRNet_{cl}   | 21.79   | 0.772  | 25.14    | 0.888  | 23.17   | 0.870  | 25.20   | 0.899  | 24.20   | 0.880  |
| ERRNet_{pl}   | 22.11   | 0.782  | 24.91    | 0.882  | 23.50   | 0.873  | 24.89   | 0.894  | 24.19   | 0.879  |
| ERRNet_{OF}   | 22.80   | 0.790  | 25.26    | 0.890  | 23.08   | 0.874  | 25.26   | 0.904  | 24.24   | 0.885  |

**In-domain Expert (IDE) Loss.** As illustrated in Sec. 3.4, the IDE loss is introduced from a straightforward idea. On the one hand, it explicitly requires that the reflection removal should rely more on the in-domain expert, which accords with the idea of determining the weights according to the expertise level in domain generalization. On the other hand, if the RTAW module is trained only with the reflection removal losses, it will keep pursuing extreme weights for some domain experts, resulting in large and unstable values in the RTAW outputs (i.e., vi). When the IDE loss is applied, the weight of other domain experts will be constrained to a reasonable range, and we empirically find that the training of RTAW module will become more stable.

As shown in Tab. 4, when the IDE loss is discarded (denoted by ERRNet_{avg, L_{IDE}}), the training of RTAW is usually unstable and may result in different order of magnitude of the expertise level (vi). Therefore, the final result heavily relies on a specific domain expert. We can see that the performance of ERRNet_{avg, L_{IDE}} is very close to the domain expert trained on Real89 dataset, indicating the failure of training.

**RTAW Module.** We also conduct ablation studies on the structure of RTAW module, and three variants are compared in the following. 1) ERRNet_{avg}: A plain average operation, which is the simplest way to perform both output fusion and network interpolation. 2) ERRNet_{cl}: A classification model has been deployed in Sec. 3.1, which shows the ability to distinguish the source of a reflection-contaminated image. Hence, it is reasonable to take such an architecture as an RTAW module variant. 3) ERRNet_{pl}: In this case, N classification models are deployed in parallel, where each model predicts the expertise level for a specific domain expert.

As shown in Tab. 5, the generalization ability is not guaranteed when a plain average operation is taken, resulting in performance degradation of ERRNet_{avg} on the Real20 dataset. ERRNet_{cl} and ERRNet_{pl} also face the problem of performance degradation on some datasets. On the contrary, the RTAW architecture with cross-domain attention module (i.e., ERRNet_{OF}) can generalize better to all target domains, showing the superiority of the proposed RTAW design.

5.2. Limitations and Future Work

As shown in Tab. 3, when using the IBCLN [21] backbone, the network interpolation (NI) manner does not work well and brings minor performance gain. This phenomenon should be attributed to the loop structure, which usually leads to accumulated errors as observed by Li et al. [25], and the interpolated parameters may exacerbate this effect. Besides, as shown in Tab. 4, although the proposed method can boost the reflection removal performance, it still does not reach the full potential of all domain experts (e.g., the qualitative performance of ERRNet_{OF} on Postcard is worse than the domain expert trained on Unaligned).

As a remedy, in the future, we will explore the specialized architecture to better exploit the power of domain generalization methods. It is worth noting that the proposed cross-domain attention module (CDAM) can be naturally expanded to a multi-head version like the self-attention layer in Transformers [41], which can be used as a component in such an architecture for converting various amount of source domains into a fixed number of features.

5.3. Ethical Discussions

The reflection removal methods can promote the image quality and benefit subsequent tasks like autonomous driving, the datasets are either from postcards, solid objects, natural scenes or allowed for academic usage, therefore there are no ethical issues or negative societal impacts.

6. Conclusion

In this paper, we analyzed the domain gaps between reflection-contaminated images, and propose to enhance the learning-based single-image reflection removal (SIRR) methods from a domain generalization perspective. Given domain experts of various reflection types, a reflection type-aware weight (RTAW) module is introduced to predict the expert-wise weights, which are used for adaptive network combination (AdaNEC). An in-domain expert loss is proposed for training the RTAW module. Experimental results show that the proposed AdaNEC can bring appealing performance gain on different state-of-the-art SIRR networks.
References

[1] Amit Agrawal, Ramesh Raskar, Shree K. Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. In ACM SIGGRAPH, pages 828–835, 2005. 2

[2] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4498–4506, 2017. 2

[3] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. Advances in Neural Information Processing Systems, 24:2178–2186, 2011. 3

[4] Hugh Chen, Scott Lundberg, and Su-In Lee. Checkpoint ensembles: Ensemble methods from a single training process. arXiv preprint arXiv:1710.03282, 2017. 3

[5] Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. IEEE Transactions on Image Processing, 27(1):304–313, 2017. 3

[6] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson WH Lau. Location-aware single image reflection removal. In IEEE International Conference on Computer Vision, pages 5017–5026, 2021. 1, 2

[7] Antonio D’Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In German Conference on Pattern Recognition, pages 187–198, 2018. 3

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, 2010. 6

[9] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In IEEE International Conference on Computer Vision, pages 3238–3247, 2017. 1, 2, 3, 4, 6

[10] Xin Feng, Wenzie Pei, Zhihui Jia, Fanglin Chen, David Zhang, and Guangming Lu. Deep-masking generative network: A unified framework for background restoration from superimposed images. IEEE Transactions on Image Processing, 30:4867–4882, 2021. 1, 2

[11] Kun Gai, Zhenwei Shi, and Changshui Zhang. Blind separation of superimposed moving images using image statistics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(1):19–32, 2012. 2

[12] J. J. Good. Rational decisions. Journal of the Royal Statistical Society. Series B (Methodological), 14(1):107–114, 1952. 5

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014. 5

[14] Xiaojie Guo, Xiaochun Cao, and Yi Ma. Robust separation of reflection from multiple images. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2187–2194, 2014. 2

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, pages 694–711, 2016. 5

[16] Soomin Kim, Yuchi Huo, and Sung-Eui Yoon. Single image reflection removal with physically-based training images. In IEEE Conference on Computer Vision and Pattern Recognition, pages 5164–5173, 2020. 1, 3, 4

[17] Chenyang Lei and Qifeng Chen. Robust reflection removal with reflection-free flash-only cues. In IEEE Conference on Computer Vision and Pattern Recognition, pages 14811–14820, 2021. 2

[18] Chenyang Lei, Xuhua Huang, Chenyang Qi, Yankun Zhao, Wenxiu Sun, Qiong Yan, and Qifeng Chen. A categorized reflection removal dataset with diverse real-world scenes. arXiv preprint arXiv:2108.03380, 2021. 3

[19] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1750–1758, 2020. 2

[20] Anat Levin and Yair Weiss. User assisted separation of reflections from a single image using a sparsity prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(9):1647–1654, 2007. 2

[21] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E. Hopcroft. Single image reflection removal through cascaded refinement. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3565–3574, 2020. 1, 2, 3, 5, 6, 7, 8

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In IEEE International Conference on Computer Vision, pages 5542–5550, 2017. 1, 2

[23] Yu Li and Michael S. Brown. Exploiting reflection change for automatic reflection removal. In IEEE International Conference on Computer Vision, pages 2432–2439, 2013. 2

[24] Yu Li and Michael S. Brown. Single image layer separation using relative smoothness. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2752–2759, 2014. 2

[25] Yu Li, Ming Liu, Yaling Yi, Qince Li, Dongwei Ren, and Wangmeng Zuo. Two-stage single image reflection removal with reflection-aware guidance. arXiv preprint arXiv:2012.00945, 2020. 1, 2, 5, 6, 7, 8

[26] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In IEEE Conference on Computer Vision and Pattern Recognition, pages 14215–14224, 2020. 2

[27] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In IEEE International Conference on Image Processing, pages 1353–1357, 2018. 3

[28] Shie Mannor, Dori Peleg, and Reuven Rubinstein. The cross entropy method for classification. In International Conference on Machine Learning, pages 561–568, 2005. 5

[29] Robert E Parkin. A note on the extinction coefficient and absorptivity of glass. Solar Energy, 114:196–197, 2015. 3

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming
Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, pages 8026–8037, 2019.

[31] BH Prasad, Lokesh R Boregowda, Kaushik Mitra, Sanjoy Chowdhury, et al. V-DESIIR: Very fast deep embedded single image reflection removal. In IEEE International Conference on Computer Vision, pages 2390–2399, 2021.

[32] Abhijith Punnapurath and Michael S. Brown. Reflection removal using a dual-pixel sensor. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1556–1565, 2019.

[33] Bernard Sarel and Michal Irani. Separating transparent layers through layer information exchange. In European Conference on Computer Vision, pages 328–341, 2004.

[34] Bernard Sarel and Michal Irani. Separating transparent layers through layer information exchange. In European Conference on Computer Vision, pages 328–341, 2004.

[35] YiChang Shih, Dilip Krishnan, Fredo Durand, and Bernard Sarel and Michal Irani. Separating transparent layers from multiple images containing reflections. In ACM International Conference on Multimedia, pages 466–470, 2016.

[36] Richard Szeliski, Shai Avidan, and Padmanabhan Anandan. Robust optical flow estimation of double-layer images under transparency or reflection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1692–1701, 2019.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[38] Chao Sun, Shuaicheng Liu, Taotao Yang, Bing Zeng, Zhengning Wang, and Guanghui Liu. Automatic reflection removal using gradient intensity and motion cues. In ACM International Conference on Multimedia, pages 2390–2399, 2021.

[39] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection removal exploiting misaligned training data and network enhancements. In IEEE Conference on Computer Vision and Pattern Recognition, pages 8178–8187, 2019.

[40] Patrick Wieschollek, Orazio Gallo, Jinwei Gu, and Jan Kautz. Separating reflection and transmission images in the wild. In European Conference on Computer Vision, pages 89–104, 2018.

[41] Jie Yang, Dong Gong, Lingqiao Liu, and William T. Freeman. A computational approach for obstruction-free photography. ACM Transactions on Graphics, 34(4):1–11, 2015.

[42] Jiaolong Yang, Hongdong Li, Yuchao Dai, and Robby T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3771–3779, 2019.

[43] Patrick Wieschollek, Orazio Gallo, Jinwei Gu, and Jan Kautz. Separating reflection and transmission images in the wild. In European Conference on Computer Vision, pages 89–104, 2018.

[44] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. ACM Transactions on Graphics, 34(4):1–11, 2015.

[45] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In European Conference on Computer Vision, pages 654–669, 2018.

[46] Jiaolong Yang, Hongdong Li, Yuchao Dai, and Robby T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1410–1419, 2016.

[47] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4786–4794, 2018.

[48] Qian Zheng, Boxin Shi, Jinnan Chen, Xudong Jiang, Ling-Yu Duan, and Alex C Kot. Single image reflection removal with absorption effect. In IEEE Conference on Computer Vision and Pattern Recognition, pages 13395–13404, 2021.

[49] Kaiyang Zhou, Ziwei Liu, Yu Qiao, and Qifeng Chen. Single image reflection removal using a dual-pixel sensor. In European Conference on Computer Vision, pages 1410–1419, 2016.

[50] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization: A survey on domain generalization. arXiv preprint arXiv:2103.02503, 2021.

[51] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization: A survey on domain generalization. arXiv preprint arXiv:2103.02503, 2021.