WBFlow: Few-shot White Balance for sRGB Images via Reversible Neural Flows

Chunxiao Li, Xuejing Kang, Anlong Ming*

School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications
{chunxiaol, kangxuejing, mal}@bupt.edu.cn

Abstract

The sRGB white balance methods aim to correct the nonlinear color cast of sRGB images without accessing raw values. Although existing methods have achieved increasingly better results, their generalization to sRGB images from multiple cameras is still under explored. In this paper, we propose the network named WBFlow that not only performs superior white balance for sRGB images but also generalizes well to multiple cameras. Specifically, we take advantage of neural flow to ensure the reversibility of WBFlow, which enables lossless rendering of color cast sRGB images back to pseudo raw features for linear white balancing and thus achieves superior performance. Furthermore, inspired by camera transformation approaches, we have designed a camera transformation (CT) in pseudo raw feature space to generalize WBFlow for different cameras via few shot learning. By utilizing a few sRGB images from an untrained camera, our WBFlow can perform well on this camera by learning the camera specific parameters of CT. Extensive experiments show that WBFlow achieves superior camera generalization and accuracy on three public datasets as well as our rendered multiple camera sRGB dataset. Our code is available at https://github.com/ChunxiaoLe/WBFlow.

1 Introduction

The color cast in sRGB images is caused by improper color temperature settings in the white balance (WB) module and then exacerbated by non-linear color renderings in image signal processor (ISP) [Afifi et al., 2019a]. The sRGB White balance (sRGB-WB) methods aim to correct such color cast without accessing raw values. It is an emerging direction and has an essential impact on some high-level computer vision tasks, such as object recognition and image segmentation [Afifi and Brown, 2019].

To perform accurate sRGB-WB, an ideal way is to reverse color-cast sRGB images to unprocessed raw values for linear correcting and then re-render them to obtain white-balanced sRGB images [Afifi and Brown, 2020a]. However, this process is challenging for sRGB-WB methods because the essential reversibility is difficult to achieve without raw values [Chakrabarti et al., 2014]. In fact, instead of the ideal WB process, recent sRGB-WB methods [Afifi et al., 2019a; Afifi et al., 2019b; Afifi and Brown, 2020b] directly utilize irreversible polynomial kernel functions [Hong et al., 2001] to train the mappings between color-cast sRGB images and ground truths (GTs). These trained mappings are not appropriate on untrained cameras due to the inherent color differences among sRGB images from different cameras [Afifi, 2021]. This seriously limits cross-camera generalization in real-world applications. In Figure 1(b), the representative sRGB-WB method, KNN-WB [Afifi et al., 2019a], results in a noticeable greenish cast on the untrained camera.

Few endeavors have been made to address this problem. The only method, Deep WB [Afifi and Brown, 2020a], proposes modeling the ideal WB process end-to-end via U-Net [Ronneberger et al., 2015]. It further generalizes to different cameras by jointly training multi-camera sRGB images. However, due to the lossy structure of U-Net and the vanilla training strategy, Deep WB becomes irreversible and suppresses camera specificity. Thus, in Figure 1(c), although

![Figure 1: Existing sRGB-WB methods are not effective in generalizing well to multiple cameras. KNN-WB [Afifi et al., 2019a] has a negative WB performance on the untrained camera, as in obvious greenish output in the second row of (b). Although Deep WB [Afifi and Brown, 2020a] suppresses this color bias, it performs worse on the trained camera, as in the output with dark colors in the first row of (c). Comparatively, our WBFlow shows superior performance on both trained and untrained cameras. As shown in (d), it generates the outputs with similar color histograms to GTs in (e).](image-url)
Deep WB improves generalization to the untrained camera, it has a degraded performance on the trained camera. In this paper, we propose an sRGB-WB network called WBFlow that achieves superior performance on both trained and untrained cameras. Unlike existing irreversible sRGB-WB methods, we take advantage of neural flows [Kingma and Dhariwal, 2018] to enable the reversibility of WBFlow. Specifically, we design Reversible Non-linear Rendering Transformation (RNRT) that simulates ISP color renderings by additive coupling layers [Kingma and Dhariwal, 2018]. Due to the inherent reversibility of additive coupling layers, RNRT can losslessly render color-cast sRGB images back to their pseudo-raw features via forward propagation. Then, to correct the color cast of pseudo-raw features, we present Reversible Linear Correction Transformation (RLCT) to generate white-balanced sRGB images. Our contributions are as follows:

1. We propose WBFlow that generalizes well to multiple cameras via reversible neural flows and few-shot learning.
2. We propose RNRT and RLCT to simulate nonlinear color renderings and linear color transformations to enable the reversibility of WBFlow.
3. We propose CT that generalizes WBFlow to untrained cameras by updating its weights via few-shot learning.
4. Extensive experiments show that WBFlow achieves superior multi-camera generalization on three public datasets and our rendered multi-camera sRGB dataset.

2 Related Works

White Balance Methods for Raw Images. The white balance methods for raw images fall into computational color constancy (CCC) research. Most CCC methods [Gijsenij and Gevers, 2007; Foster, 2011; Hu et al., 2017; Lo et al., 2021; Song et al., 2021; Xu et al., 2021; Tang et al., 2022; Zhang et al., 2022] achieve the goal of correcting the color cast in raw images by estimating illumination colors. They make up the WB module of camera ISPs. However, none of these methods can be applied to sRGB images [Afifi et al., 2019a], since the linear color cast of raw images is broken due to non-linear ISP renderings. Different from CCC methods, our WBFlow focuses on correcting the non-linear color cast of sRGB images.

White Balance Methods for sRGB Images. Recent sRGB-WB methods corrected the color cast in sRGB images by optimizing non-linear mappings (exemplar-based method) [Afifi et al., 2019a; Afifi et al., 2019b; Afifi and Brown, 2020b; Afifi and Brown, 2019; Afifi et al., 2020] or modeling ideal WB process (DNN-based method) [Afifi and Brown, 2020a]. In detail, exemplar-based methods combined trained non-linear mappings to correct inputs. Since these mappings are irreversible, the specificities of trained cameras are mixed and applied to encode white-balanced sRGB images during testing, making them biased towards trained cameras rather than test ones. Inevitably, the performances of exemplar-based methods are inferior on untrained cameras (Figure 1(b)). To solve this problem, the DNN-based method, Deep WB, proposed rendering inputs back to pseudo-raw features by U-Net encoder, then correcting pseudo-raw features and re-rendering them to white-balanced sRGB images by U-Net decoder. It also allowed WB manipulation in sRGB images by setting two additional decoders to render inputs with 2850 Kelvin(K) and 7500K. However, due to lossy pooling layers, Deep WB is irreversible, making the content of pseudo-raw features less complete than actual raw values and further restricting WB accuracy. Further, the joint training strategy suppressed the specificities of different cameras [Huang et al., 2020], and hence together with irreversible structure, limited the improvement of multi-camera generalization (Figure 1(c)). Our WBFlow belongs to DNN-based methods and achieves superior generalization to multiple cameras via reversible neural flows and few-shot learning.

3 White Balance for sRGB Image via Reversible Flows and Few-shot Learning

In this section, we introduce WBFlow that aims to achieve superior generalization on multiple cameras. We start with formulating the ideal WB process for sRGB images. Then, we introduce details of WBFlow and how it generalizes to multiple cameras via few-shot learning.

3.1 Ideal White Balance Process for sRGB Images

An sRGB image $I_n$ that is captured by camera $n$ and rendered by color temperature $ct$ can be formed as [Afifi et al., 2019a]:

$$I_n^{ct} = f_n(T_n W_{ct} I_n^{raw}),$$

where $I_n^{raw}$ is a raw image from camera $n$, $W_{ct}$ is a WB module with color temperature $ct$, $T_n$ is a color transformation matrix for camera $n$, $f_n(\cdot)$ is a non-linear rendering function, including color enhancement, tone manipulation, and gamma encoding et al. Corresponding correct sRGB image $I_n^{wb}$, i.e. GT, is obtained by manually setting $W_{ct}$ according to actual illumination color. Here, we denote this ideal setting as $W_{ideal}$. Therefore, the relationship between $I_n^{wb}$ and $I_n^{ct}$ can be modeled as:

$$I_n^{wb} = f_n(T_n W_{ideal} W_{ct}^{-1} T_n^{-1} f_n^{-1}(I_n^{ct})).$$

From Equation 2, the ideal WB process is reversible to render color-cast sRGB image toward its white-balanced version while preserving complete content. Specifically, $I_n^{wb}$ is first reversed back to its color-cast linear values via $f_n^{-1}(\cdot)$; then, $T_n$ transforms color-cast linear values into raw values and then $W_{ideal} W_{ct}^{-1}$ white balance them; finally, $f_n(\cdot)$ and $T_n$ render white-balanced raw values to obtain $I_n^{wb}$.
As discussed in Section 1 and 2, existing sRGB-WB methods [Afifi et al., 2019a; Afifi et al., 2020; Afifi and Brown, 2020b; Afifi et al., 2021; Afifi and Brown, 2020a] are irreversible and thus fail to satisfy Equation 2, which limits their generalization to multiple cameras. In this paper, we propose an sRGB-WB network named WBFow that satisfies the reversibility of Equation 2 and generalizes well to cameras via few-shot learning. Details will be introduced as follows.

3.2 White Balance for sRGB Images via Reversible Transformations

To achieve accurate white balance on multiple cameras, our primary goal is to appropriately model the ideal WB process. According to Equation 2, this process has two types of modules: nonlinear color rendering function ($F_n(\cdot)$) and linear color transformations ($W_{ct}, W_{ideal}, T_n$). Deep WB [Afifi and Brown, 2020a] models these two kinds of modules via identical irreversible structures, which only improves multi-camera generalization to some extent. To solve this problem, different from Deep WB, we model two kinds of modules by individual reversible functions:

$$\hat{I}_{wb} = F^{-1}(M(F(I_{wb}))),$$  

where $F(\cdot)$ is a reversible non-linear function that losslessly renders color-cast sRGB image $I_{ct}$ back to pseudo-raw features, $M(\cdot)$ is a reversible linear function that corrects the color cast of pseudo-raw features and preserves the content information, $F^{-1}(\cdot)$ is the inverse of $F(\cdot)$ that re-renders white-balanced pseudo-raw features to corresponding sRGB image $I_{wb}$. Theoretically, due to $F(\cdot)$ and $M(\cdot)$, Equation 3 satisfies the reversibility of Equation 2 to generate accurate white-balanced sRGB images on different cameras. Following are the implementation details.

Reversible Non-linear Rendering Transformation

As discussed, the reversible non-linear function $F(\cdot)$ aims to obtain pseudo-raw features with lossless content. Deep WB [Afifi and Brown, 2020a] cannot satisfy it because of lossy pooling layers. Here, considering color renderings included in $F(\cdot)$ are independent [Ramanath et al., 2005], we model them by a sequence of reversible bijective sub-functions: $F = F_1 \circ F_2 \cdots \circ F_K$. This way, the relationship between color-cast sRGB image $I_{ct}$ and intermediate feature $z_{ct,k}^n$ is:

$$I_{ct} \xrightarrow{F_1} z_{ct,1}^n \xrightarrow{F_2} z_{ct,2}^n \cdots \xrightarrow{F_K} z_{ct,K}^n,$$

where $z_{ct,K}^n = z_{ct}^n$ is pseudo-raw feature.

In practice, we adopt additive coupling layers [Kingma and Dhariwal, 2018] to model $\{F_k\}$, which are effective in simulating nonlinear reversible transformations [Kingma and Dhariwal, 2018; An et al., 2021]. In our WBFow, the forward computation of the additive coupling layer is:

$$z_{ct,k}^{n,a} = \text{Split}(z_{ct,k}^n),$$
$$z_{ct,k}^{n,b} = \text{Conv}(z_{ct,k}^{n,a}) + z_{ct,k}^{n,b},$$
$$z_{ct,k+1}^{n} = \text{Concat}(z_{ct,k}^{n,a}, z_{ct,k}^{n,b}),$$

where $\text{Split}(\cdot)$ splits $z_{ct,k}^n$ into two parts along channel dimension to obtain $z_{ct,k}^{n,a}$ and $z_{ct,k}^{n,b}$. $\text{Conv}(\cdot)$ renders colors of $z_{ct,k}^{n,a}$ with $3 \times 3$ convolutions and relu functions. $\text{Concat}(\cdot)$ concatenates $z_{ct,k}^{n,a}$ and $z_{ct,k}^{n,b}$ to form rendered feature $z_{ct,k+1}^n$. The reverse computation of the additive coupling layer is easily derived from Equation 5. We can superimpose additive coupling layers to establish the reversible mapping between input color-cast sRGB image and pseudo-raw feature as Equation 4. However, as in Equation 5, the additive coupling layer leaves some channels of intermediate feature unchanged due to $\text{Split}(\cdot)$. To solve this problem, following [Kingma and Dhariwal, 2018], we use reversible $1 \times 1$ convolution that integrates all channels by fixing the channel numbers of input and output as the same. We also adopt Actnorm [Kingma and Dhariwal, 2018] to normalize features, an alternative to BatchNorm that accelerates the training times.
The proposed RNRT is shown in Figure 2, consisting of additive coupling layers, reversible $1 \times 1$ convolutions, and Actnorms. All components of RNRT are reversible, making content lossless during in forward and backward propagations. As shown in Figure 3(c), we can obtain pseudo-raw features with lossless content from color-cast input in Figure 3(a) via forward RNRT. Simultaneously, in Figure 3(b), the input can be restored accurately from pseudo-raw features in Figure 3(c) via reverse RNRT.

Reversible Linear Correction Transformation

In Equation 2 and 3, the reversible linear function $\mathcal{M}(\cdot)$ is linear and reversible due to the WB module and color transformation matrix. To properly implement it while preserving lossless content information, inspired by [Huang and Belongie, 2017], we propose RLCT that separates color and content information from pseudo-raw features and corrects the color information by reversible $1 \times 1$ convolutions [Kingma and Dhariwal, 2018]. The formation is:

$$z_{\text{wb}}^{n} = T_{\sigma}(\sigma_{z})(z_{\text{ct}}^{n}/\sigma_{z}) + T_{\mu}(\mu_{z}),$$

where channel mean and variance, $\mu_{z}$ and $\sigma_{z}$, represents the color information of pseudo-raw feature $z_{\text{ct}}^{n}$, [Stricker and Orengo, 1995]. They also help to keep content information by normalizing $z_{\text{ct}}^{n}$, i.e., $(z_{\text{ct}}^{n}/\mu_{z})/\sigma_{z}$. For individual color information, we correct it by respectively forwarding $\mu_{z}$ and $\sigma_{z}$ into linear mappings $T_{\sigma}(\cdot)$ and $T_{\mu}(\cdot)$, both of which consist of three reversible $1 \times 1$ convolutions [Kingma and Dhariwal, 2018]. This way, in Figure 3(f), the pseudo-raw features are white-balanced by re-scaling lossless content with learned channel mean $T_{\sigma}(\sigma_{z})$ and variance $T_{\mu}(\mu_{z})$. Further, in Figure 3(e), the white-balanced sRGB image is successfully rendered from white-balanced pseudo-raw features in Figure 3(f) via reverse propagation of RNRT. It has an almost identical color histogram as GT in Figure 3(d).

As introduced above, RNRT and RLCT are reversible and thus cooperate to ensure complete reversibility, which satisfies the requirement of Equation 2. Thus, our WBFlow can significantly improve the floor of WB accuracy on multiple cameras. As shown in Figure 4(a), with the same multi-camera training strategy of Deep WB, our WBFlow converges with a much lower $l_{1}$ loss on multiple cameras.

3.3 Camera Generalization via Few-shot Learning

According to Equation 2, the actual raw features should be specific to various cameras. However, existing methods suppressed such specificities due to combined trained mappings [Afifi et al., 2019a; Afifi and Brown, 2020b] or joint multi-camera training strategy [Afifi and Brown, 2020a]. Inevitably, this will result in a limited multi-camera generalization. To enhance the camera specificity of pseudo-raw feature, we introduce few-shot learning into WBFlow, which is based on the fact that inter-camera transformation can be achieved by learning the mapping between raw color values from two cameras [Gao et al., 2017]. In Figure 2, we set the camera transformation named CT after RNRT. For camera $n$, Equation 6 can be re-written as:

$$z_{\text{wb}}^{n} = w_{\text{cam}}^{n} * \left\{ T_{\sigma}(\sigma_{z})\left(z_{\text{ct}}^{n}/\sigma_{z}\right) + T_{\mu}(\mu_{z}) \right\},$$

where $*$ is the group convolution operator, $w_{\text{cam}}^{n}$ is a set of weights learned from the camera labels by two reversible neural flow combinations. This strategy ensures that the learned weights are unique to the different cameras. To adapt WBFlow to camera $n$, we fix the pre-trained parameter of RNRT, then train the new weights in Equation 7 by minimizing the $l_{1}$ loss with a few samples. This way, the specificity of pseudo-raw features to camera $n$ is enhanced, thus improving the generalization effect. In Figure 4(b), the $\Delta E^{2000}$ values of WBFlow on the new camera, PanasonicGX1, are always lower than these of Deep WB, regardless of the number of training cameras.

4 Experiments

In this section, we compare the effectiveness of WBFlow with state-of-the-art sRGB-WB methods in terms of multi-camera generalization, inference time, and few-shot learning. In addition, ablation experiments are designed to evaluate the effectiveness of each part in WBFlow.

4.1 Datasets

Public Datasets. Following [Afifi and Brown, 2020a], we randomly selected 12000 sRGB images from the first fold of Set1 [Afifi et al., 2019a; Cheng et al., 2014] to train WBFlow.
we used the remaining two folds (Set1-Test), Set2 [Afifi et al., 2019a] and rendered cube dataset [Afifi et al., 2019a; Banic et al., 2017] for evaluation.

**Rendered Multi-camera sRGB Dataset.** We collected a multi-camera sRGB dataset to evaluate the multi-camera generalization effect. Specifically, we selected and compiled 184 groups of raw images from the NUS dataset [Cheng et al., 2014]. In each group, the raw images are consistent in scenes and different in the cameras: Canon1DsMkIII, Canon600D, FujifilmXM1, NikonD5200, OlympusEPL6, PanasonicGX1, SamsungNX2000, and SonyA57. To obtain color-cast sRGB versions of these images, following [Afifi et al., 2019a], we rendered them by Adobe Camera Raw with five common color temperatures (2850 K, 3800 K, 5500 K, 6500 K, and 7500 K) and camera standard photo finishing. We obtain GTs by manually selecting the correct color temperature from the middle gray patches in the color checker of each raw image. The rest of the operations remain unchanged. Our multi-camera sRGB dataset contains 7360 sRGB images with 184 scenes, five color temperatures, and eight cameras.

### 4.2 Implementation Details

**Loss function.** Following [Afifi and Brown, 2020a], we apply $l_1$ loss to train WBFlow:

$$\arg \min_{F,M} \sum_{ct} \sum_{n} \left\| F^{-1}(M(F(I^n_{ct}))) - I^n_{w,b} \right\|_1.$$  \hspace{1cm} (8)

**Training and Testing Detail.** We implemented WBFlow on PyTorch with CUDA support and used the Adam [Kingma and Ba, 2014] with $\beta_1 = 0.9$ and learning rate $10^{-4}$ to optimize it. For the experiments with all training images, we trained WBFlow for 340000 iterations with batch size 4. While for few-shot experiments, we trained CT for 15000 iterations. We used color jittering, average blur, geometric rotation, and flipping to augment data. During testing, following [Afifi and Brown, 2020a], we resized all input images to a maximum dimension of 656 pixels and set a color mapping procedure to compute the final white-balanced sRGB images.

**Error Metric.** We used the same error metrics as existing sRGB-WB methods [Afifi and Brown, 2020a; Afifi et al., 2019a]: mean angle error (MAE) and $\Delta E_{2000}$ [Sharma et al., 2005]. We reported the mean, first quantile (Q1), second quantile (Q2), and third quantile (Q3) for evaluation. A lower error metric denotes a better sRGB-WB performance.

### 4.3 Multi-camera Generalization Evaluation

Table 1 shows the quantitative results of our WBFlow and state-of-the-art sRGB-WB methods on three public datasets. The top results are in bold.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\Delta E_{2000}$ ↓</th>
<th>Mean Angle Error (MAE) ↓</th>
<th>Inference Time (s) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Q1</td>
<td>Q2</td>
</tr>
<tr>
<td><strong>Set1–Test (trained cameras)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN-WB [Afifi et al., 2019a]</td>
<td>3.58</td>
<td>2.07</td>
<td>3.09</td>
</tr>
<tr>
<td>MixedWB [Afifi et al., 2021]</td>
<td>4.55</td>
<td>3.00</td>
<td>4.15</td>
</tr>
<tr>
<td>Deep WB [Afifi and Brown, 2020a]</td>
<td>3.77</td>
<td>2.16</td>
<td>3.30</td>
</tr>
<tr>
<td>Our WBFlow</td>
<td>3.13</td>
<td>1.92</td>
<td>2.79</td>
</tr>
<tr>
<td><strong>Set2 (untrained cameras)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN-WB [Afifi et al., 2019a]</td>
<td>5.60</td>
<td>3.43</td>
<td>4.90</td>
</tr>
<tr>
<td>MixedWB [Afifi et al., 2021]</td>
<td>6.05</td>
<td>3.45</td>
<td>4.92</td>
</tr>
<tr>
<td>Deep WB [Afifi and Brown, 2020a]</td>
<td>4.90</td>
<td>3.13</td>
<td>4.35</td>
</tr>
<tr>
<td>Our WBFlow</td>
<td>4.64</td>
<td>3.16</td>
<td>4.07</td>
</tr>
<tr>
<td><strong>Rendered Cube Dataset (untrained cameras)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNN-WB [Afifi et al., 2019a]</td>
<td>5.68</td>
<td>3.22</td>
<td>4.61</td>
</tr>
<tr>
<td>MixedWB [Afifi et al., 2021]</td>
<td>5.03</td>
<td>2.07</td>
<td>3.12</td>
</tr>
<tr>
<td>Deep WB [Afifi and Brown, 2020a]</td>
<td>4.59</td>
<td>2.68</td>
<td>3.81</td>
</tr>
<tr>
<td>Our WBFlow</td>
<td>4.28</td>
<td>2.71</td>
<td>3.77</td>
</tr>
</tbody>
</table>

Table 1: Quantitative results of our WBFlow and state-of-the-art sRGB-WB methods on three public datasets. The top results are in bold.
appear in Set2. These results verify the effectiveness of our WBFlow in multi-camera generalization.

**Inference Time Comparison.** From Table 1, our WBFlow achieves superior WB performance on all datasets with almost the same inference time as Deep WB (the most related method). Compared with KNN-WB with the fastest inference time, our WBFlow outperforms it by about 12.57%, 17.14% and 24.65% for the mean of $\Delta E^{2000}$ in three datasets.

**Qualitative Comparison.** To qualitatively compare the multi-camera generalization, we show the generated white-balanced sRGB images and their color histograms of KNN-WB, Deep WB, and WBFlow on eight cameras in Figure 5. From it, WBFlow achieves the most similar colors as GTs on all cameras, consistent with the quantitate comparison.

### 4.4 Few Shot Evaluation

To validate few-shot capacity, we compared the performances of WBFlow with the most related method, Deep WB [Afifi and Brown, 2020a], on four untrained cameras. We retrained Deep WB and WBFlow by randomly selecting 5000 sRGB images from the first fold of Set1 and used four untrained cameras in the last 84 groups of our rendered multi-camera sRGB dataset for evaluation: OlympusEPL6, PanasonicGX1, SamsungNX2000, and SonyA57. Training details are followed by [Afifi and Brown, 2020a]. For few-shot experiments, we randomly selected K images (0, 5, 10, 20) for untrained cameras from the first 100 groups of our rendered multi-camera sRGB dataset to train the parameters of CT, which are 5.82% of our WBFlow’s. To avoid randomness and disturbance, we repeated experiments 1000 times with randomly selected images and then computed the $\Delta E^{2000}$ average in Table 2.

---

Table 2: Few shot evaluation of WBFlow on rendered multi-camera sRGB dataset (Deep WB: [Afifi and Brown, 2020a]).

<table>
<thead>
<tr>
<th>Methods</th>
<th>OlympusEPL6 ($\Delta E^{2000}$)</th>
<th>PanasonicGX1 ($\Delta E^{2000}$)</th>
<th>SamsungNX2000 ($\Delta E^{2000}$)</th>
<th>SonyA57 ($\Delta E^{2000}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>Deep WB</td>
<td>7.12</td>
<td>5.43</td>
<td>6.94</td>
<td>8.46</td>
</tr>
<tr>
<td>WBFlow K=0</td>
<td>6.19</td>
<td>4.39</td>
<td>5.73</td>
<td>7.72</td>
</tr>
<tr>
<td>WBFlow K=5</td>
<td>6.13</td>
<td>4.33</td>
<td>5.68</td>
<td>7.75</td>
</tr>
<tr>
<td>WBFlow K=10</td>
<td>6.00</td>
<td>4.37</td>
<td>5.37</td>
<td>7.18</td>
</tr>
<tr>
<td>WBFlow K=20</td>
<td>5.97</td>
<td>4.25</td>
<td>5.39</td>
<td>7.36</td>
</tr>
</tbody>
</table>
Figure 6: Ablation comparisons of WBFlow for sRGB images from eight cameras (fixed scenes and fixed color temperatures): (a) different variants of WBFlow and (b) different flow numbers of RNRT.

From Table 2, WBFlow significantly outperforms Deep WB even with no shots (K=0) due to the reversible structure. For example, the average $\Delta E_{2000}$ values of WBFlow are smaller than those of Deep WB by about 17.44%, 23.09%, 21.68%, and 23.45% in Q2 on four untrained cameras. This superiority was enhanced when we updated CT with K=5, 10, 20 to enhance the camera specificity. Especially, the few-shot learning effectively improves Q3 of average $\Delta E_{2000}$ on four untrained cameras. This indicates that our WBFlow with few-shot learning is superior in handling untrained cameras.

4.5 Ablation Analysis
To verify the effectiveness of each part in WBFlow, we conducted an ablation analysis on our rendered multi-camera sRGB dataset. Since Deep WB is the most related method, we report the average $\Delta E_{2000}$ values of RNRT on eight cameras are much worse than Deep WB, which shows that only modeling color renderings hardly improves the multi-cameras generalization. This phenomenon is significantly mitigated when we integrate RNRT and RLCT. Specifically, compared with only RNRT, RNRT&RLCT work together to reduce the average $\Delta E_{2000}$ values by about 50% on eight cameras. RNRT&RLCT also considerably outperforms Deep WB by 13% on eight cameras. This is because both RNRT and RLCT are reversible to effectively model the ideal WB process. When we add CT, the multi-camera generalization is further improved, as in the smallest average $\Delta E_{2000}$ of RNRT&RLCT&CT on eight cameras.

Camera Specificity Verification. Since channel average and variance can represent the color information of pseudo-raw features [Stricker and Orengo, 1995], we compute them for white-balanced pseudo-raw features without and with CT on eight cameras to verify the camera specificity in Figure 7. With CT, the difference between mean/variance for all white balance pseudo-raw feature channels becomes significantly larger on eight cameras. This verifies that CT can significantly enhance the camera specificity of pseudo-raw features, thus improving multi-camera generalization.

5 Conclusion
In this paper, we propose an sRGB-WB network named WBFlow, which performs superior white balance for sRGB images and generalizes well to multiple cameras. Unlike existing irreversible sRGB-WB methods that fail to model the ideal WB process, WBFlow successfully models this process through reversible RNRT and RLCT. Furthermore, we generalize WBFlow to multiple cameras by enhancing the camera characteristic of pseudo-raw features via few-shot learning. Extensive experiments indicate the superiority of our WBFlow in multi-camera generalization and few-shot sRGB-WB tasks. Ablation analysis shows the effectiveness of each part of WBFlow and their optimal combinations.

Acknowledgements
This work was supported by the national key R & D program intergovernmental international science and technology innovation cooperation project (2021YFE0101600).
References


