Learning Equilibrium Transformation for Gamut Expansion and Color Restoration

Jun Xiao[1], Changjian Shui[2], Zhi-Song Liu[3], Qian Ye[4], and Kin-Man Lam[1]

[1] The Hong Kong Polytechnic University, Hong Kong SAR, China
[2] Vector Institute, Toronto, Canada
[3] Lappeenranta-Lahti University of Technology, Lahti, Finland
[4] Graduate School of Information Sciences, Tohoku University, Japan
jun.xiao@connect.polyu.hk changjian.shui@vectorinstitute.ai
zhisong.liu@lut.fi qian@vision.is.tohoku.ac.jp kin.man.lam@polyu.edu.hk

Abstract. Existing imaging systems support wide-gamut images like ProPhoto RGB, but most images are typically encoded in a narrower gamut space (e.g., sRGB). To this end, these images can be enhanced by learning to recover the original color values beyond the sRGB gamut, or out-of-gamut values. Current methods incorporate the metadata from the target wide-gamut images to expand the gamut, while preventing distortion of in-gamut values. However, this metadata is hard to obtain in real-world scenarios. In this paper, we propose a novel method that requires no metadata. We formulate gamut expansion as a "root-finding" problem and learn an equilibrium transformation via a neural network. Specifically, our method defines a dynamic system that keeps in-gamut values stable to prevent color distortion and updates out-of-gamut values recurrently. Therefore, we employ an implicit recurrent mechanism to iteratively extract features, which can effectively mitigate the vanishing gradient problem, and reduce the GPU memory consumption to $\mathcal{O}(1)$ complexity during training. Experiments demonstrate the effectiveness and efficiency of our model, in terms of gamut expansion and color restoration, outperforming state-of-the-art models by 0.40dB, in terms of PSNR, with a size of 40K parameters only. The codes are available at: `https://github.com/junxiao01/LETNet`.

Keywords: Computational photography · Color enhancement · Equilibrium model

1 Introduction

Modern imaging systems, such as digital single-reflex (DSLR) and smartphone cameras, can support the ProPhoto RGB (ProRGB) color space [21], which has a wide color gamut and can display up to 90% of visible colors. However, in practice, most images are encoded in the standard RGB (sRGB) color space [38] via a clipping and projection process. Compared with ProRGB space, sRGB space covers a narrow color gamut with 30% visible colors. To improve the visual

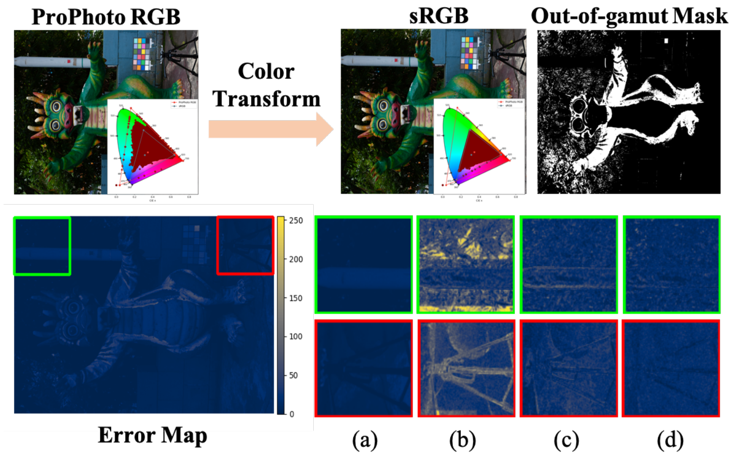

Fig. 1: Illustration of color transformation from the ProRGB space to the sRGB space, including the out-of-gamut mask and the error maps. (a) The ground-truth patches, and the patches generated by (b) Restormer [48], (c) SwinIR [27], and (d) Ours.

quality, one could apply inverse restoration to recover the colors missing from the sRGB space, which is typically an ill-posed problem.

As depicted in Fig. 1, when an image is initially encoded in a wide-gamut space such as ProRGB, converting it to sRGB involves clipping and projecting those color values that fall outside the target-gamut space. Gamut expansion as an inverse process, aims to restore the out-of-gamut color values, but this process is ill-posed due to the non-invertibility. Furthermore, conventional deep image restoration models do not work well when they are applied to gamut expansion. Specifically, they will unavoidably introduce additional errors within the gamut space, shown in Figs. 1(b) and 1(c). Besides, from a practical perspective, gamut expansion algorithms are typically integrated into resource-constrained devices such as smartphones and personal laptops, whereas advanced restoration models are built with complex network structures with limited practical applications. To this end, lightweight and effective methods are broadly considered. Through leveraging the external information such as out-of-gamut masks [24] or metadata extracted from target images [23, 25], previous gamut-expansion methods can effectively learn the inverse function for color enhancement.

In this paper, we follow this line of research to develop effective gamut-expansion models **without using external information**, which is often infeasible in practice. Concretely, we consider gamut expansion as a "root-finding" problem, where the missing color values are the solutions to implicit equations. To solve this problem, we propose a novel method that defines a dynamic system to learn an equilibrium transformation through a lightweight input-injection network that adopts shallow and simple network structures. Such a system maintains stable in-gamut pixel values to avoid distortion, and it recurrently updates out-of-gamut pixel values for restoration. However, shallow network structures typically struggle to extract discriminative features for restoration and enhancement. To address this issue, we further employ an Implicit Recurrent Mechanism (IRM) to decouple the forward and backward propagations. In the forward pass, IRM leverages an off-the-shelf optimization solver to perform iterative feature extraction, ultimately leading to a fixed point in the latent space. This mechanism boosts the ability of our model in feature representation, as the fixed-point features are equivalent to features extracted from infinite recurrent layers. In the backpropagation phase, IRM relies on the fixed-point features, without requiring storage of any intermediate results throughout the iteration process. As a result, our model can avoid the vanishing gradient problem and maintain constant GPU consumption during training. Coupled with the proposed lightweight input-injection network, our proposed model is well-suited for resource-constrained devices.

The main contributions of this paper are as follows:

1. We formulate gamut expansion as a "root-finding" problem and propose a novel method that adopts a lightweight input-injection network to learn an equilibrium transformation to solve the problem, without using any external information.
2. We exploit an implicit recurrent mechanism to decouple forward and backward propagations in training. This mechanism effectively enhances the capability of our model in feature representation, while reducing GPU memory consumption to $\mathcal{O}(1)$ complexity, which is friendly to resource-constrained devices.
3. Experiments show that our model can effectively restore corrupted out-of-gamut values, while suppressing distortion of in-gamut values for gamut expansion and color restoration, significantly outperforming state-of-the-art methods by 0.40dB, with 40K model parameters only.

2 Related Works

2.1 Color Space Conversion

Gamut reduction solely degrades out-of-gamut color values, because the gamut of different color spaces is not consistent. To address this issue, the methods in [49, 50] use the local features extracted from the images to fit the target gamut space. GamutNet [24] incorporates the out-of-gamut masks into a "U-shaped" network to generate the target images from the given sRGB images in an end-to-end manner. Inspired by the method in [31], Le *et al.* [23] proposed a method to estimate the color inverse function by leveraging the metadata. Furthermore, a lightweight MLP model was used to learn the inverse function, specifically optimized for testing samples [25]. The metadata used consists of partial testing pixel values extracted from the target wide-gamut images and their narrow-gamut counterparts. However, such metadata is usually unavailable in real-world scenarios, where only sRGB images are accessible. In this paper, we focus on gamut expansion and color restoration in this common situation and propose a lightweight and effective method to address this issue.

2.2 Implicit Deep Learning

Recently, many research works [2, 3, 7, 9, 13, 14, 18, 26, 36, 44] focus on implicit learning methods that, unlike conventional deep models, do not rely on explicit computational graphs or sequentially stacking neural layers. Instead, these implicit networks aim to model a dynamic system, where the solutions of these implicit networks correspond to the solutions of the defined dynamic systems. Neural ODE [9, 12, 29, 32, 46] is a classic work that models infinitesimal steps of a residual block. The Deep Equilibrium (DEQ) network [2, 26, 44] is another type of implicit learning that directly solves a "root-finding" problem and finds the fixed-point representation of a shallow network. Recent studies demonstrate that DEQ-based models can achieve impressive results in generative modeling [6, 22, 35], landmark detection [30], language processing [4], etc. However, to the best of our knowledge, these implicit models have not been applied to gamut expansion and color restoration. In this paper, we consider gamut expansion as a "root-finding" problem and propose a method to learn the equilibrium transformation based on implicit learning techniques to address this problem.

3 Methodology

3.1 Problem Formulation

When converting a ProRGB image I_{PP} to its sRGB counterpart I_{s}, a nonlinear imaging process is applied, i.e., $I_{\mathrm{s}} = f_e(\mathrm{clip}(M \cdot I_{\mathrm{PP}}))$, where M and $f_{\mathrm{e}}(\cdot)$ represent the color mapping matrix and the gamma-encoding function [38], respectively. If the color value $\mathbf{c}_{\mathrm{PP}}$ in the ProRGB image exceeds the range of the sRGB gamut space, it will be clipped and projected onto the boundary of the target color space, generating a color value $\mathbf{c}_s$ in the sRGB space. Notably, this process will only result in the loss of information from the out-of-gamut values, but it does not affect the in-gamut values, i.e., $\mathbf{c}_{\mathrm{PP}} = \mathbf{c}'_s = M^{-1} f_{\mathrm{d}}(\mathbf{c}_{\mathrm{s}})$, where $f_{\mathrm{d}}(\cdot)$ is the decoding function and M^{-1} is the inverse color mapping function.

To perfectly reconstruct the corresponding ProRGB image from a sRGB version, the inverse function should primarily focus on restoring the out-of-gamut values while keeping the in-gamut values unchanged, as follows:

$$\hat{\mathbf{c}}_{\mathrm{PP}} = g(\mathbf{c}'_s) = \begin{cases} f(\mathbf{c}'_s), & \mathbf{c}'_s \in \mathcal{C}_{\mathrm{out}}, \\ \mathbf{c}'_s, & \mathbf{c}'_s \in \mathcal{C}_{\mathrm{in}}, \end{cases}, \tag{1}$$

where $\hat{\mathbf{c}}_{\mathrm{PP}}$ is the estimated color value, $f(\cdot)$ represents the restoration function, and $\mathcal{C}_{\mathrm{in}}$ and $\mathcal{C}_{\mathrm{out}}$ denote the sets of in-gamut and out-of-gamut values, respectively. Previous studies learned the inverse mapping function $g(\cdot)$ from the input color values $\mathbf{c}'_s$ through neural networks [24, 25] or the optimization methods [23] from the input color values $\mathbf{c}'_{\mathrm{s}}$. These methods leverage the external information about the target images, which aids in identifying out-of-gamut and in-gamut values, thereby simplifying the problem. Unlike these methods, in this paper, we propose a novel method to directly learn the inverse function in this paper, without requiring external information.

We consider the output $\hat{\mathbf{c}}_{\mathrm{PP}}$ to be a latent variable, denoted as $\mathbf{z}$ and define the inverse mapping function $g(\cdot)$ as a joint function of the input $\mathbf{x}$ and the latent variable $\mathbf{z}$, as follows:

$$g(\mathbf{x}, \mathbf{z}) - \mathbf{z} = 0. \tag{2}$$

Here, we convert gamut expansion into a "root-finding" problem, using color values in the ProRGB space, denoted as $\mathbf{z}^*$, serving as the solutions for this problem. In this formulation, the inverse function $g(\cdot)$ is an implicit function that defines a dynamic system. Typically, this problem has a fixed-point solution, and the inverse function $g(\cdot)$ is often referred to as equilibrium transformation. In other words, the solution $\mathbf{z}^*$ remains stable when we further apply the equilibrium transformation, which is a desirable property for gamut expansion. The in-gamut values are the solution of this problem at the initial stage and will remain stable during further equilibrium transformation, i.e., $\mathbf{z}^* = g(\mathbf{x}, \mathbf{z}^*)$. In contrast, the out-of-gamut values are iteratively updated and gradually approach the solution by solving this problem, i.e., $\mathbf{z}^{(k+1)} = g(\mathbf{x}, \mathbf{z}^{(k)}) \rightarrow \mathbf{z}^*$, for $k \rightarrow \infty$. The remaining key issue is how to learn this equilibrium transformation.

3.2 Equilibrium Transformation Learning

In this paper, we propose an effective method to learn equilibrium transformation via a shallow input-injection network, which is well-suited for resource-constrained devices. The overall pipeline of our proposed method is depicted in Fig. 2. Firstly, our method projects the input sRGB image into the latent space, leading to an initial point for the problem. Then, the equilibrium learning module takes this as input and computes the fixed-point solution in the latent space with an implicit recurrent mechanism, which employs an off-the-shelf optimization solver to model an "infinite" number of recurrent layers. Finally, our method transfers the fixed-point features back to the spatial domain for reconstruction.

Initialization. Given an input sRGB image I_{s}, we first apply two convolutional layers to project it into latent space and form shallow features, denoted as $\mathbf{x}$. Previous studies [11,37,39,47] have shown that integrating positional information into the model can improve reconstruction performance. In our method, we extract positional information from the input image to form the coordinate map, denoted as $\mathbf{p}$. Then, we compute the Fourier representation of the coordinate map, i.e., $\mathcal{T}(\mathbf{p}) = [\cos(\mathbf{p}), \sin(\mathbf{p})]$, and utilize a convolutional layer to generate the initial latent feature $\mathbf{z}^{(0)}$. The features $\mathbf{x}$ and $\mathbf{z}^{(0)}$ jointly form the initial input of the equilibrium learning module.

Equilibrium Learning Module. Given the initial input of $\mathbf{x}$ and $\mathbf{z}^{(0)}$, the equilibrium learning module iteratively calculates the latent features, as follows:

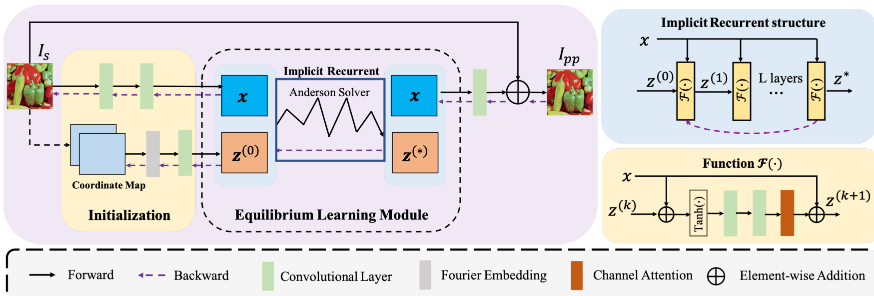

Fig. 2: Left: Illustration of the overall pipeline of our proposed method for learning equilibrium transformation for gamut expansion and color restoration. Right: Illustration of the structure of the equilibrium learning module, which is modeled by an input-injection neural network with the implicit recurrent mechanism.

$$\mathbf{z}^{(k+1)} = \underbrace{h_\theta(\tanh(\mathbf{z}^{(k)} + \mathbf{x})) + \mathbf{x}}_{\mathcal{F}=g(\mathbf{x},\mathbf{z}^{(k)})}, \tag{3}$$

for $k = 0, \cdots, K$, where K is the number of iterations. Typically, $\mathbf{z}^{(k+1)}$ will converge to the fixed-point solution $\mathbf{z}^*$, when K is sufficiently large. In our method, we propose a shallow input-injection network to model the equilibrium transformation, denoted as $\mathcal{F}$, thereby avoiding the increase in computational complexity. The overall structure of the input-injection network is illustrated on the right of Fig. 2. In this network structure, we first inject the input information $\mathbf{x}$ extracted from the input images into the latent features $\mathbf{z}^{(k)}$, and then apply two convolutional layers, denoted by h_θ with parameters θ, using the Tanh activation for feature extraction. Then, we apply the channel attention mechanism [20] to the output and use residual connections to generate the updated feature $\mathbf{z}^{(k+1)}$. To perform iterative computation, recurrent structures are commonly utilized in conventional deep learning models. However, previous studies [19,33] have shown that recurrent networks often suffer from the vanishing gradient problem, which may lead to unstable training and cannot guarantee to produce fixed-point features. In addition, an L-layer recurrent network used to store intermediate results in the recurrent process for backpropagation requires $\mathcal{O}(L)$ computational complexity, i.e., the training cost is significantly increased.

To address these issues, we utilize the implicit recurrent mechanism [2,28] for feature extraction. Unlike recurrent networks, the implicit recurrent mechanism adopts an off-the-shelf optimization solver (e.g., quasi-Newton methods [1,8])) to iteratively extract features, ultimately leading to a fixed point. This mechanism is equivalent to modeling an infinite number of recurrent layers, so it effectively enhances the capability of our lightweight and shallow model utilized in feature representation and contributes to performance improvement. In implementation, we adopt the Anderson iterative method [1] to iteratively updates the latent features $\mathbf{z}^{(k)}$ based on the previous m_k results, computed as follows:

Algorithm 1 Anderson Iteration Procedure

Input: initial point $\mathbf{z}^{(0)}$, and fixed-point function $\mathcal{F}(\cdot)$
Parameters: Max storage size m, and the parameter β
while $k <= K$ **do**
  Set $m_k = \min(m, k)$.
  Compute weights α_i^k for the past m_k steps.
  Update the latent features $\mathbf{z}^{(k+1)}$ via Eq. (4).
  $k \leftarrow k + 1$.
end while

$$\mathbf{z}^{(k+1)} = \beta \sum_{i=0}^{m_k} \alpha_i^k \mathcal{F}(\mathbf{z}^{k-m_k+i}) + (1-\beta) \sum_{i=0}^{m_k} \alpha_i^k \mathbf{z}^{(k-m_i+i)}, \tag{4}$$

where α_i^k is the weight for the i-th previous step at the k-th iteration, and these weights $\boldsymbol{\alpha}^k$ are determined by minimizing $\|G^k \boldsymbol{\alpha}^k\|_2$ subject to $\sum_{i=1}^{m_k} \alpha_i^k = 1$, where matrix G^k contains the past residuals. Further details can be found in [40]. The fixed-point function $\mathcal{F}(\cdot)$ is the input-injection network in our method, and β is a hyper-parameter. The iterative process is illustrated in Algorithm 1

After obtaining the fixed-point features, we apply a convolutional layer to transfer them to the spatial domain, and then use a global connection to fuse the output and the input sRGB image to generate the ProRGB image $\hat{I}_{\mathrm{PP}}$.

3.3 Model Training: IRM

Our method adopts an off-the-shelf optimization solver in the input-injection network, so we cannot directly backpropagate gradient information through the optimization solver to update model parameters. As a result, the forward and backward propagation are decoupled in our method. To train the model and update the parameters, we rely on the implicit function theorem [2,22,28].

Theorem 1. ***(Implicit Function Theorem (IFT))****. Assume that the loss function is $\mathcal{L}$ and the network is parameterized by θ. Given a fixed-point representation $\mathbf{z}^*$ and an input representation $\mathbf{x}$, the gradient of the implicit recurrent mechanism (IRM) in the backward pass is computed as follows:*

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^*} \left(I - J_{\mathcal{F}(\mathbf{z}^*)}\right)^{-1} \frac{\partial \mathcal{F}(\mathbf{z}^*, \mathbf{x})}{\partial \theta}, \tag{5}$$

where $J_{\mathcal{F}(\mathbf{z}^*)} = \frac{\partial \mathcal{F}}{\partial \mathbf{z}^*}$ is the Jacobian matrix computed at the fixed point $\mathbf{z}^*$.

The proof of Theorem 1 can be found in [2]. Notably, this theorem allows us to directly differentiate at the fixed-point features $\mathbf{z}^*$ and achieve backpropagation, without the need of storing the trajectory (i.e., the intermediate results) in the forward pass. Therefore, our model is free from the vanishing gradient problem and takes only $\mathcal{O}(1)$ memory complexity for backpropagation. Compared with recurrent networks, this significantly reduces memory requirements.

Nevertheless, we still need to iteratively solve for fixed-point representations in the latent space. In each iteration, we have to compute the inverse of the Jacobian matrix $\mathcal{J} = (I - J_{\mathcal{F}(\mathbf{z}^*)})^{-1}$, which requires $\mathcal{O}(N^3)$ computational complexity. When dealing with high-dimensional signals (e.g., images), the cost of computing this inversion will become intractable. To address this issue, many efforts have been made to approximating the inverse-Jacobian matrix $\mathcal{J}$ by solving a linear system: $\mathcal{J}^T = \mathcal{J}^T \frac{\partial \mathcal{F}(\mathbf{z}^*)}{\partial \mathbf{z}^*} + \frac{\partial \mathcal{L}}{\partial \mathbf{z}^*}$, where T denotes the transpose operator. However, this method still needs an off-the-shelf optimization solver to iteratively obtain the solution. Instead, we approximate $\mathcal{J}$ with its Neumann series expansion [17], to obtain the gradients as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} \approx \lim_{N \to \infty} \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^*} \sum_{n=0}^{N} (J_{\mathcal{F}(\boldsymbol{z}^*)})^n \frac{\partial \mathcal{F}(\boldsymbol{z}^*, \hat{\boldsymbol{x}})}{\partial \theta}. \tag{6}$$

Let $G = \sum_{n=0}^{N} (J_{\mathcal{F}(\boldsymbol{z}^*)})^n$, which is an approximation of the inverse Jacobian matrix $\mathcal{J}$. Specifically, we set $G = I$, the identity matrix, to simplify backward propagation of the proposed model into a single-step computation $\frac{\partial \mathcal{L}}{\partial \theta} \approx \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^*} \frac{\partial \mathcal{F}(\boldsymbol{z}^*, \hat{\boldsymbol{x}})}{\partial \theta}$, which is Jacobian-free. This means that our method uses the inexact gradients to update the parameters, without requiring the computation of the inverse Jacobian matrix. This can significantly reduce training time. The capability of using inexact gradients is a direct and unique consequence of the fixed-point formulation, with the assumption of a certain level of stability of the underlying dynamics system [5, 15–17].

4 Experiments and Analysis

4.1 Experiment Settings

Dataset Information. In our experiments, we adopt a public dataset [25], which is specifically designed for gamut expansion, denoted as the GE dataset. This dataset provides 2,000 samples for training and 200 samples for testing. Each sample contains a wide-gamut image encoded in the ProPhoto RGB space and its corresponding version in the sRGB space. The test samples have different percentages of out-of-gamut colors, and the sRGB images of each test sample have at least one million pixels of out-of-gamut color values. The resolution of the provided samples ranges from 2000×3000 to 4000×6000.

Training Details. To train the model, we randomly crop local image patches of size 128×128 at each training iteration. The batch size is set to 64. We adopt the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ to update model parameters, and the total number of epochs is set to 200. In the training procedure, the cosine annealing strategy is utilized to adaptively decay the learning rate. As our model utilizes an off-the-shelf optimization solver, we set the number of iterations to 50 to obtain the fixed-point features. ℓ_2 loss is used as the loss function to measure the reconstruction error. We use the Pytorch [34] framework and an NVIDIA 3090 GPU to implement our model, which takes approximately two days to complete the training process.

Evaluation Metrics. To compare the performance of different methods, we evaluate the reconstruction and perceptual quality of the generated images. Specifically, we measure the reconstruction qualities using peak-signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), which have been widely used in numerous low-level vision tasks [10, 41–43, 45, 51, 53]. To evaluate the perceptual quality of the generated images, we utilize the LPIPS score [52]. In addition, we additionally measure model complexity, in terms of the number of model parameters, the GPU memory consumption, and the running time. When the resolution of the test samples is larger than 2000×3000 and the GPU memory is limited, the test sample is cropped into local image patches of the size 1024×1024, resulting in a total of 3079 image patches for evaluation.

4.2 Experiments on the GE Dataset

In this experiment, we compare our proposed method with the naive inversion (NI) method, improved-sampled (I-Sampled) method [23], GamutMLP [25], Restormer [48], SwinIR [27], and GamutNet [24]. The NI method directly maps the input sRGB images back to the ProRGB space without utilizing any additional restoration methods. This method serves as the baseline model in our experiment. The I-Sampled method and GamutMLP are image-specific methods, which optimize each test sample using metadata. GamutNet is a specialized network for gamut expansion that utilizes an out-of-gamut mask to avoid introducing artifacts into in-gamut values. Restormer and SwinIR are two state-of-the-art image restoration methods, which have demonstrated remarkable performance in several low-level vision tasks. For a fair comparison, we use the source codes of GamutNet, Restormer, and SwinIR and retrain them on the GE dataset, following their default settings. For the I-Sampled method and GamutMLP, we directly employ their public models for evaluation.

We evaluate the performance (i.e., PSNR, SSIM, and LPIPS scores) of the different methods on both the entire image (All) and the out-of-gamut areas (Out-of-gamut). Table 1 tabulates the performance of the methods on the GE dataset, including the number of their model parameters. When compared to those image-specific methods, such as GamutMLP, our model archives superior performance on both entire images and out-of-gamut values, in terms of PSNR, without relying on metadata. The distinct advantage of our method is well suitable for real-world scenarios, where only sRGB images are available.

Compared with general restoration methods, our model significantly outperforms them. Specifically, our model surpasses the second-best model, i.e., SwinIR by 0.48dB and 0.33dB on entire images and out-of-gamut values, respectively. Notably, our model consists of only 0.04M parameters, indicating much lower model complexity. This makes it suitable for resource-constrained devices. Furthermore, in Fig. 3, we show the generated images, the error maps, and the color distribution in the gamut spaces of the different methods. Analysis of the error maps shows that GamutMLP produces promising results for out-of-gamut values. However, it introduces substantial errors for in-gamut values, resulting in a distorted color distribution in the gamut space. In comparison, our model

Table 1: Illustration of the average PSNR, SSIM, and LPIPS scores of different models on the GE dataset [25], and the number of their model parameters. ✓and ✗indicate whether using metadata. 'S' denotes the image-specific methods. 'G' denotes the general methods. "-" denotes the "not available". The best results are highlighted in red. The second-best results are highlighted in blue.

Methods		Meta Data	# Params	All			Out-Of-Gamut		
				PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NI		✗	-	46.61	0.9838	0.0042	47.10	0.9838	0.0027
'S'	I-Sampled [23]	✓	-	49.78	0.9841	0.0034	48.96	0.9878	0.0022
	GamutMLP [25]	✓	4.96K	52.54	0.9958	0.0018	52.19	0.9963	0.0011
'G'	Restormer [48]	✗	26.13M	52.23	0.9932	0.0023	51.51	0.9937	0.0014
	SwinIR [27]	✗	3.13M	52.30	0.9930	0.0023	51.90	0.9934	0.0014
	GamutNet [24]	✓	4.37M	52.67	0.9929	0.0028	51.71	0.9933	0.0018
	Ours	✗	0.04M	52.78	0.9931	0.0025	52.23	0.9934	0.0016

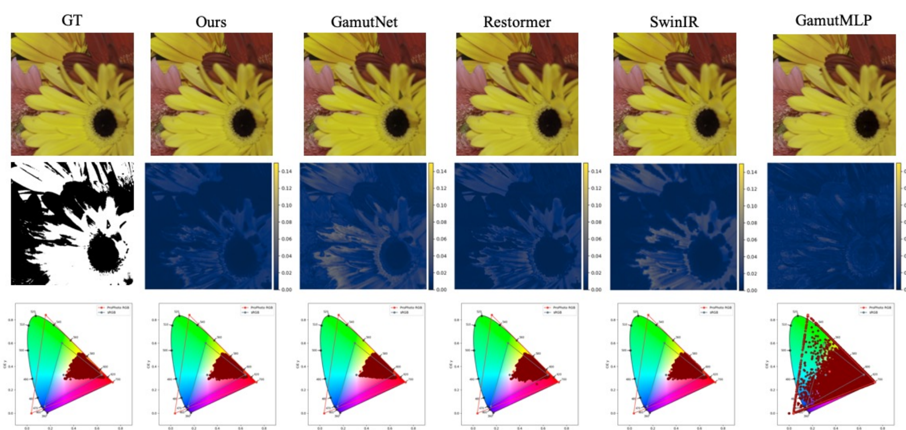

Fig. 3: Illustration of the ground-truth images, error maps, and color gamuts generated by different methods.

achieves the smallest reconstruction errors, and its color distribution is close to that of the target image. All these results show the effectiveness and efficiency of our model in gamut expansion and color restoration, particularly in situations where only sRGB images are available and the devices are resource-constrained.

4.3 Experiments on Out-of-gamut Regions

In real-world situations, sRGB images contain varying percentages of pixels with out-of-gamut values. In this experiment, we delve deeper into evaluating the performance of our model across images with different percentages of out-of-gamut pixels. To facilitate this analysis, we categorize the testing samples into four groups: $A : 0 \leq \delta_{\text{OG}} < 25$, $B : 25 \leq \delta_{\text{OG}} < 50$, $C : 50 \leq \delta_{\text{OG}} < 75$, and $D : 75 \leq \delta_{\text{OG}} \leq 100$, where δ_{OG} represents the percentage of pixels with out-of-gamut values in an image. We evaluate the performance of different models on these four groups, and the average PSNR and SSIM scores of the out-of-gamut regions are tabulated in Table 2.

As observed, our model significantly outperforms the compared models by a substantial margin for groups A, B, and D. Particularly for groups A and D, our model surpasses the second-best method by 0.32dB and 0.36dB, respectively. In group C, the performances achieved by our method and SwinIR are slightly different. These results show the superior capability of the proposed model in handling sRGB images with varying percentages of out-of-gamut areas.

Table 2: Illustration of the PSNR and SSIM of different methods on images with different ratios of out-of-gamut values. The testing samples are divided into four groups. The term "Average" refers to performance in the full range ($0 \leq \delta_{\text{OG}} \leq 100$). The best results are highlighted in bold.

Methods	Restormer [48]	SwinIR [27]	GamutNet [24]	Ours
	PSNR↑/ SSIM↑	PSNR↑/ SSIM↑	PSNR↑/ SSIM↑	PSNR↑/ SSIM↑
$A : 0 \leq \delta_{\text{OG}} < 25$	52.09/0.9976	52.51/0.9976	44.71/0.9912	**52.83**/**0.9978**
$B : 25 \leq \delta_{\text{OG}} < 50$	51.06/0.9934	51.55/0.9935	44.35/0.9858	**51.68**/**0.9936**
$C : 50 \leq \delta_{\text{OG}} < 75$	50.35/0.9862	**50.42**/**0.9858**	43.73/0.9745	50.41/0.9840
$D : 75 \leq \delta_{\text{OG}} \leq 100$	51.68/0.9777	52.06/0.9755	44.27/0.9617	**52.42**/**0.9774**
Average	51.51/0.9937	51.90/**0.9934**	51.71/0.9933	**52.23**/**0.9934**

4.4 Ablation Studies

Experiments on Implicit Recurrent Mechanism. To facilitate the learning of equilibrium transformation, we adopt a shallow input-injection network, based on the implicit recurrent mechanism. To evaluate its effectiveness in feature representation and GPU memory reduction, we compare our proposed model with other models using two different settings: a plain model, denoted as Plain, which does not adopt the recurrent structure, and the model that employs the traditional recurrent structure, denoted as Recurrent. To ensure a fair comparison, we maintain consistent configurations for these models, and set the number of iterations or the number of recurrent layers to 50.

The average PSNR, the utilized GPU memory in training, and the runtime in testing are illustrated in Fig. 4. These results show that our model, incorporating the implicit recurrent mechanism (IRM), significantly improves the

Table 3: The utilized GPU memory of different methods in the inference process. (Note: the input size is 1024×1024 implemented on a single NVIDIA V100 GPU.)

Methods	GamutNet	Restormer	SwinIR	IRM
Memory	2.1G	27.8G	26.2G	2.6G

performance by 2.52dB, with only a slight increase in runtime and GPU memory consumption. The increased runtime is an acceptable trade-off brought by the iterative optimization solver. In comparison to the model using traditional recurrent structures, our model surpasses it by a large margin while requiring much less GPU memory consumption. Furthermore, Table 3 presents the GPU memory utilization of GamutNet, Restormer, SwinIR, and IRM for a comprehensive comparison. Compared to Restormer and SwinIR, our model exhibits a superior advantage, in terms of GPU memory sizes, during inference.

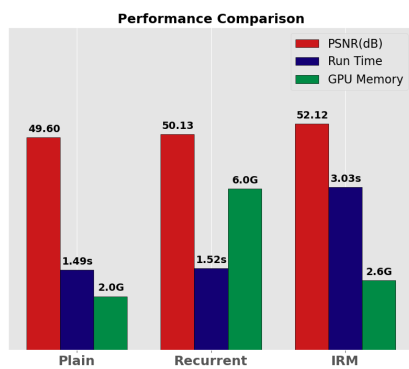

Fig. 4: Performance comparison of the plain model (Plain), the recurrent model (Recurrent), and the implicit recurrent model (IRM), in terms of PSNR (dB), running time in testing, and GPU memory in training. Note that the input size is 1024×1024 and the implementation device is a single NVIDIA 3090 GPU.

Experiments on Forward Pass. We also investigate how the number of iterations for the off-the-shelf optimization solver impacts the reconstruction performance across entire images, out-of-gamut areas, and in-gamut areas. To achieve this, we evaluate our model on a test sample, while varying the number of iterations for the optimization solver. The performance metrics, in terms of PSRN and root-mean-square error (RMSE), measured at different numbers of iterations are depicted in Fig. 5. We also visualize the reconstruction error maps generated when the iterations $K = 10, 15, 20, 30, 40,$ and 50, as shown in Fig. 6.

From these results, we observe that the performance on in-gamut values significantly improves and then gradually converges to a stable state, resulting in better performance compared to entire images and out-of-gamut areas. Similarly, the RMSE of in-gamut areas experiences rapid reduction, ultimately achieving a lower reconstruction error. These results demonstrate that the equilibrium transformation learned by our method effectively stabilizes the in-gamut values, preventing distortion, while iteratively updating and converging the out-of-gamut values to fixed-point solutions.

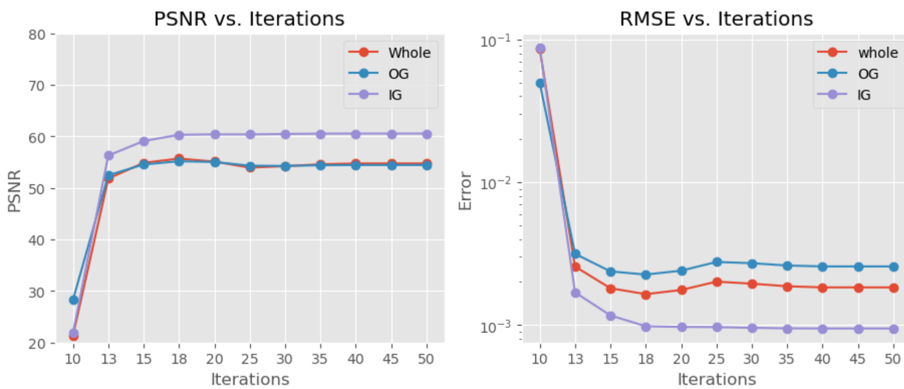

Fig. 5: PSNR (Left) and RMSE (Right) of different areas at various iterations.

Experiments on Input-injection Network. Our proposed method learns the equilibrium transformation through a shallow and lightweight network. We explore how network structures affect the performance of our method. Specifically, we evaluate models with different activation functions (e.g., Sin [37] and ReLU) and models without input injection (w/o input) or without utilizing the channel attention mechanism (w/o CA). The average PSNR, SSIM, and LPIPS scores, for models with the various settings are tabulated in Table 4.

We observe that the network structures and activation functions employed in our method have a significant impact on the overall performance. Notably, if a model does not initially inject the input into the latent features, there is a nearly 1.00dB PSNR drop. Similarly, when the model uses the ReLU function, the performance is seriously degraded. In this paper, we use a shallow structure to prevent an increase in computational complexity, resulting in a lightweight model. The exploration of other effective network structures that utilize the implicit recurrent mechanism is left for future research.

5 Conclusion

In this paper, we focus on expanding a narrow gamut space (e.g., sRGB) to a wider gamut space (e.g., ProPhoto RGB) and recovering color values that exceed the range of the narrow gamut space, when only sRGB images are available. To address this issue, we consider gamut expansion as a “root-finding" problem and

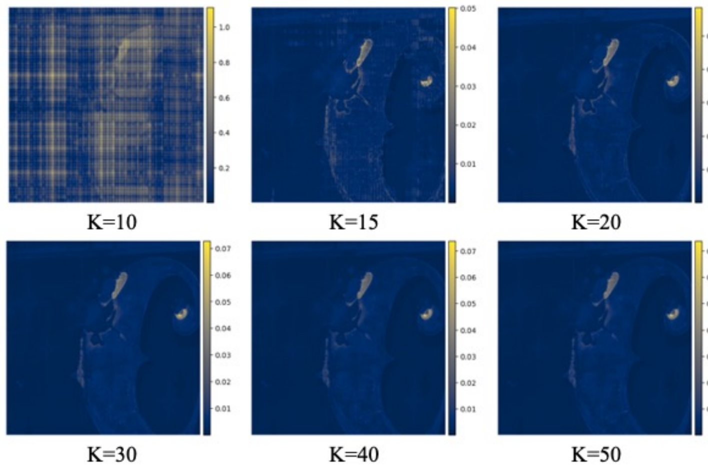

Fig. 6: The error maps generated when $K = 10, 15, 20, 30, 40,$ and 50.

Table 4: The average PSNR, SSIM, and LPIPS of the models with different settings on the out-of-gamut values. 'Act' denotes the activation function. The best results are highlighted in bold.

Settings		PSNR↑	SSIM↑	LPIPS↓
'Act'	Tanh	**52.23**	**0.9934**	**0.0016**
	ReLU	50.76	0.9911	0.0020
	Sin	52.12	0.9934	**0.0016**
w/o Input		51.27	0.9915	0.0018
w/o CA		51.68	0.9921	0.0017
Input + CA		**52.23**	**0.9934**	**0.0016**

propose a novel method to solve it. Specifically, our method learns the equilibrium transformation through a shallow and lightweight network. This defines a dynamic system in the latent space, and the equilibrium transformation learned can help stabilize the in-gamut values to prevent distortion, while recurrently updating out-of-gamut values. We further employ the implicit recurrent mechanism to iteratively extract features, resulting in fixed-point features for reconstruction. This mechanism effectively enhances the feature representation capability of our model and reduces GPU memory consumption to $\mathcal{O}(1)$ complexity. Experiments show the effectiveness and efficiency of our model in gamut expansion and color restoration. Our model significantly outperforms state-of-the-art methods, while using much less model complexity, i.e., 40K model parameters.

Acknowledgements

This work was supported by the Hong Kong Research Grants Council (RGC) Research Impact Fund (RIF) under Grant R5001-18.

References

1. Anderson, D.G.: Iterative procedures for nonlinear integral equations. Journal of the ACM (JACM) **12**(4), 547–560 (1965)
2. Bai, S., Kolter, J.Z., Koltun, V.: Deep equilibrium models. Advances in Neural Information Processing Systems **32** (2019)
3. Bai, S., Koltun, V., Kolter, J.Z.: Multiscale deep equilibrium models. Advances in Neural Information Processing Systems **33**, 5238–5250 (2020)
4. Bai, S., Koltun, V., Kolter, J.Z.: Neural deep equilibrium solvers. In: International Conference on Learning Representations (2021)
5. Bai, S., Koltun, V., Kolter, J.Z.: Stabilizing equilibrium models by jacobian regularization. arXiv preprint arXiv:2106.14342 (2021)
6. Biloš, M., Sommer, J., Rangapuram, S.S., Januschowski, T., Günnemann, S.: Neural flows: Efficient alternative to neural odes. Advances in neural information processing systems **34**, 21325–21337 (2021)
7. Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., Vert, J.P.: Efficient and modular implicit differentiation. Advances in neural information processing systems **35**, 5230–5242 (2022)
8. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Mathematics of computation **19**(92), 577–593 (1965)
9. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in neural information processing systems **31** (2018)
10. Chen, X., Li, H., Li, M., Pan, J.: Learning a sparse transformer network for effective image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5896–5905 (2023)
11. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8628–8638 (2021)
12. Dupont, E., Doucet, A., Teh, Y.W.: Augmented neural odes. Advances in neural information processing systems **32** (2019)
13. Finzi, M., Wang, K.A., Wilson, A.G.: Simplifying hamiltonian and lagrangian neural networks via explicit constraints. Advances in neural information processing systems **33**, 13880–13889 (2020)
14. Florence, P., Lynch, C., Zeng, A., Ramirez, O.A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., Tompson, J.: Implicit behavioral cloning. In: Conference on Robot Learning. pp. 158–168. PMLR (2022)
15. Fung, S.W., Heaton, H., Li, Q., McKenzie, D., Osher, S., Yin, W.: Fixed point networks: Implicit depth models with jacobian-free backprop. arXiv preprint arXiv:2103.12803 **3**(8), 9 (2021)
16. Geng, Z., Guo, M.H., Chen, H., Li, X., Wei, K., Lin, Z.: Is attention better than matrix decomposition? arXiv preprint arXiv:2109.04553 (2021)
17. Geng, Z., Zhang, X.Y., Bai, S., Wang, Y., Lin, Z.: On training implicit models. Advances in Neural Information Processing Systems **34**, 24247–24260 (2021)

18. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. Advances in neural information processing systems **32** (2019)
19. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6**(02), 107–116 (1998)
20. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
21. ISO: Iso - iso 22028-2:2013 - photography and graphic technology — extended colour encodings for digital image storage, manipulation and interchange — part 2: Reference output medium metric rgb colour image encoding (romm rgb). https://www.iso.org/standard/56591.html (2021-04-19)
22. Krantz, S.G., Parks, H.R.: The implicit function theorem: history, theory, and applications. Springer Science & Business Media (2002)
23. Le, H., Afifi, M., Brown, M.S.: Improving color space conversion for camera-captured images via wide-gamut metadata. In: Color and Imaging Conference. vol. 2020, pp. 193–198. Society for Imaging Science and Technology (2020)
24. Le, H., Jeong, T., Abdelhamed, A., Shin, H.J., Brown, M.S.: Gamutnet: Restoring wide-gamut colors for camera-captured images. In: Color and Imaging Conference. vol. 2021, pp. 7–12. Society for Imaging Science and Technology (2021)
25. Le, H.M., Price, B., Cohen, S., Brown, M.S.: Gamutmlp: A lightweight mlp for color loss recovery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18268–18277 (2023)
26. Li, M., Wang, Y., Lin, Z.: Cerdeq: Certifiable deep equilibrium model. In: International Conference on Machine Learning. pp. 12998–13013. PMLR (2022)
27. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1833–1844 (2021)
28. Lorraine, J., Vicol, P., Duvenaud, D.: Optimizing millions of hyperparameters by implicit differentiation. In: International conference on artificial intelligence and statistics. pp. 1540–1552. PMLR (2020)
29. Massaroli, S., Poli, M., Park, J., Yamashita, A., Asama, H.: Dissecting neural odes. Advances in Neural Information Processing Systems **33**, 3952–3963 (2020)
30. Micaelli, P., Vahdat, A., Yin, H., Kautz, J., Molchanov, P.: Recurrence without recurrence: Stable video landmark detection with deep equilibrium models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22814–22825 (2023)
31. Nguyen, R.M., Brown, M.S.: Raw image reconstruction using a self-contained srgb–jpeg image with small memory overhead. International journal of computer vision **126**, 637–650 (2018)
32. Norcliffe, A., Bodnar, C., Day, B., Simidjievski, N., Liò, P.: On second order behaviour in augmented neural odes. Advances in neural information processing systems **33**, 5911–5921 (2020)
33. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318. Pmlr (2013)
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
35. Pokle, A., Geng, Z., Kolter, J.Z.: Deep equilibrium approaches to diffusion models. Advances in Neural Information Processing Systems **35**, 37975–37990 (2022)

36. Poli, M., Massaroli, S., Scimeca, L., Chun, S., Oh, S.J., Yamashita, A., Asama, H., Park, J., Garg, A.: Neural hybrid automata: Learning dynamics with multiple modes and stochastic transitions. Advances in Neural Information Processing Systems **34**, 9977–9989 (2021)
37. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in neural information processing systems **33**, 7462–7473 (2020)
38. Stokes, M.: A standard default color space for the internet-srgb. http://www. w3. org/Graphics/Color/sRGB. html (1996)
39. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)
40. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. SIAM Journal on Numerical Analysis (2011)
41. Xiao, J., Jiang, X., Zheng, N., Yang, H., Yang, Y., Yang, Y., Li, D., Lam, K.M.: Online video super-resolution with convolutional kernel bypass grafts. IEEE Transactions on Multimedia **25**, 8972–8987 (2023)
42. Xiao, J., Lyu, Z., Zhang, C., Ju, Y., Shui, C., Lam, K.M.: Towards progressive multi-frequency representation for image warping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2995–3004 (2024)
43. Xiao, J., Ye, Q., Zhao, R., Lam, K.M., Wan, K.: Self-feature learning: An efficient deep lightweight network for image super-resolution. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4408–4416 (2021)
44. Yang, Z., Pang, T., Liu, Y.: A closer look at the adversarial robustness of deep equilibrium models. Advances in Neural Information Processing Systems **35**, 10448–10461 (2022)
45. Ye, Q., Xiao, J., Lam, K.m., Okatani, T.: Progressive and selective fusion network for high dynamic range imaging. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 5290–5297 (2021)
46. Yildiz, C., Heinonen, M., Lahdesmaki, H.: Ode2vae: Deep generative second order odes with bayesian neural networks. Advances in Neural Information Processing Systems **32** (2019)
47. Yoon, Y., Chung, I., Wang, L., Yoon, K.J.: Spheresr: 360deg image super-resolution with arbitrary projection via continuous spherical image representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5677–5686 (2022)
48. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5728–5739 (2022)
49. Zamir, S.W., Vazquez-Corral, J., Bertalmío, M.: Gamut extension for cinema. IEEE Transactions on Image Processing **26**(4), 1595–1606 (2017)
50. Zamir, S.W., Vazquez-Corral, J., Bertalmio, M.: Vision models for wide color gamut imaging in cinema. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(5), 1777–1790 (2019)
51. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6360–6376 (2021)

52. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
53. Zhu, Q., Zhou, M., Zheng, N., Li, C., Huang, J., Zhao, F.: Exploring temporal frequency spectrum in deep video deblurring. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12428–12437 (2023)