Dense Pixel-to-Pixel Harmonization via Continuous Image Representation

Jianqi Chen, Yilan Zhang, Zhengxia Zou, Keyan Chen, and Zhenwei Shi*, *Member, IEEE*

Abstract—High-resolution (HR) image harmonization is of great significance in real-world applications such as image synthesis and image editing. However, due to the high memory costs, existing dense pixel-to-pixel harmonization methods are mainly focusing on processing low-resolution (LR) images. Some recent works resort to combining with color-to-color transformations but are either limited to certain resolutions or heavily depend on hand-crafted image filters. In this work, we explore leveraging the implicit neural representation (INR) and propose a novel image Harmonization method based on Implicit neural Networks (HINet), which to the best of our knowledge, is the first dense pixel-to-pixel method applicable to HR images without any hand-crafted filter design. Inspired by the Retinex theory, we decouple the MLPs into two parts to respectively capture the content and environment of composite images. A Low-Resolution Image Prior (LRIP) network is designed to alleviate the Boundary Inconsistency problem, and we also propose new designs for the training and inference process. Extensive experiments have demonstrated the effectiveness of our method compared with state-of-the-art methods. Furthermore, some interesting and practical applications of the proposed method are explored. Our code is available at https://github.com/WindVChen/INR-Harmonization.

Index Terms—Image harmonization, implicit neural representation, high resolution, pixel-to-pixel.

Fig. 1. Structural differences between the existing harmonization methods and our proposed HINet.

I. INTRODUCTION

IMAGE compositing, a fundamental technique in image processing, has been widely used in various applications such as image editing [1]–[3], data augmentation [4]–[6], *etc*. It encompasses various methods such as image matting [7] and shadow generation/removal [8], with the objective of generating realistic synthetic outputs by extracting foreground objects from one image and seamlessly integrating them into another background image. Nonetheless, inconsistencies in color spaces between the foreground and background of the composite image often result in perceptual disparities due to variations in lighting and tonal qualities. This challenge, distinct from image enhancement techniques [9], [10] which primarily address the overall appearance of an image, frequently necessitates manual adjustments of the color distribution within the foreground image layers, which demands much professional knowledge and does cost lots of time.

Aimed to harmonize the composite images, [11] first introduced deep learning into the task and many effective approaches have been proposed [12]–[17] in recent years. These data-driven approaches, compared with traditional ones [18]–[21] that rely on matching low-level statistics between foreground and background, have demonstrated better results with their strong semantic representation capability. However, as these methods mostly adopt a U-Net [22] like structure, the harmonization process is essentially a dense pixel-to-pixel transformation and costs much GPU memory [23]. As a result, it is difficult for these methods to be applied for processing high-resolution (HR) images, such as $2K$ or $4K$, and most of them only perform at a low resolution of $256{\times}256$ pixels. To achieve HR image harmonization, more recent works [23]–[25] proposed to leverage color-to-color transformations which saves much memory. Despite the applicability of harmonizing HR images, these methods are either limited to certain resolutions [23], or heavily rely on hand-crafted filters [24], [25] which are cumbersome in design and also limit the potential of deep learning networks.

Jianqi Chen, Yilan Zhang, Keyan Chen, and Zhenwei Shi are with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, and with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

Zhengxia Zou is with the Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China.

Considering the unaffordable memory cost of the current pixel-to-pixel deep learning-based harmonization frameworks, we study if it is possible to apply a scale-adaptive dense pixel-to-pixel transformation to HR image harmonization. In our method, we are inspired by the recently popular paradigm Implicit Neural Representation (INR) [26], [27], which leverages a stack of multilayer perceptrons (MLP) to represent a 3D scene [28] or a 2D image [26] parameterized by continuous coordinate input. Two features of INR appeal to us. Firstly, different from the CNN structure that takes the whole image/feature map as input, the input to INR structure is a vector containing grid coordinates, providing us more control over the memory cost. Secondly, by inputting continuous coordinates (x, y) and outputting RGB values, the MLP models in its weights a continuous image that is not limited by resolution, which may benefit HR image harmonization. Inspired by these two advantages, in this paper we explore a novel dense pixel-to-pixel image Harmonization method based on Implicit neural Networks (HINet). Fig. 1 shows the differences between our method and previous harmonization methods.

Directly applying conventional INR approaches to the image harmonization task can lead to several issues, including subpar performance, discontinuous patterns, and high memory consumption. We have provided an extensive analysis of these challenges in Sec. III-B. In light of these challenges, we have meticulously designed the proposed HINet to achieve superior harmonization results. Inspired by Retinex theory [29], [30], we decouple the MLPs in the HINet decoder into a content extraction part and an appearance rendering part. One is to preserve content structure and determine what objects are in the image, while the other is to capture the global environment of the image. The parameters of these two parts' MLPs are predicted by different encoder layers, thus alleviating the burden of the last output layer. Furthermore, to prevent the potential inconsistency problem of the local MLPs, we design a Low-Resolution Image Prior (LRIP) structure, where the MLPs of the content extraction part are more finely divided into several parts, each processing a specific resolution of the input with the former lower resolution part's output as a prior. This structure can both solve the inconsistency problem and reduce memory costs. For HR image harmonization training, we design a Random Step Crop (RSC) strategy, while also introducing the inference process for ultra-HR images. Moreover, following [24], [25], we include an optional 3D lookup table (LUT) [31], [32] rendering branch to enrich the comprehensibility of the model and enable the manual control.

Our contributions can be summarized as follows:

- As far as we know, our method is the first dense pixel-to-pixel image harmonization method that can be applied to HR images. The proposed HINet requires no hand-crafted filters and liberates the strong learning ability of the neural networks.
- We explore leveraging INR in image harmonization. We expect that the new paradigm can pave way for more future research on HR harmonization task.
- Extensive experiments have demonstrated the effectiveness of our method. Compared with previous methods, the HINet can achieve state-of-the-art performance on HR

image harmonization. Moreover, we have explored some interesting potentials of the HINet in practical usages, such as arbitrary resolution harmonization, and region-based harmonization for both images and videos.

II. RELATED WORKS

Image Harmonization. In this part, we mainly focus on reviewing deep learning-based image harmonization methods, as their superiority over the traditional methods [18]–[21], [34] has been demonstrated [11]. Since Tsai *et al.* [11] pioneered conducting image harmonization with deep neural networks, many data-driven methods [12]–[14], [16], [17], [35] have been proposed and achieve good results on LR images. Cun *et al.* [35] proposed to leverage the attention mechanism for better learning features of foreground and background. Cong *et al.* [12] considered the domain shift to harmonize composited images. Ling *et al.* [14] referred to AdaIN [36] and regarded image harmonization as a style transfer problem. Sofiiuk *et al.* [16] proposed to utilize high-level semantic features from pre-trained models. Hang *et al.* [17] leveraged contrastive learning to narrow the solution space. These methods, although performing well on LR image harmonization (256×256), are hard to be applied to harmonize HR images due to the high memory cost in their U-Net [22] structure design. To meet the practical needs of HR image harmonization, more recent works [23]–[25], [37]–[39] resorted to color-to-color operations. Cong *et al.* [23] proposed to combine features of pixel-to-pixel and color-to-color transformations, and [24], [25] manually designed several image filters and predicted their parameters. Despite their potential in processing HR images, there remain some limitations. For CDTNet [23], since the Refinement Module alone requires about 6GB memory for a single 2048×2048 image, it will consume unbearable memory at higher resolution (*e.g.* 6K) and thus only works for certain high resolutions. For [24], [25], although they are capable of being applied to flexible high resolutions, these methods rely heavily on hand-crafted filters that are cumbersome in design [24], [25]. Recent [37] proposed to approximate the filters with predicted piece-wise linear functions, yet may fail to model complex scenes due to the simplicity of the function. Another more recent study [38] also utilizes piece-wise linear curves and predicts additional shading maps to enhance local control. Additionally, [39] approximates color transformations through the application of an affine matrix and predicts a corresponding parameter map.

In contrast to previous dense pixel-to-pixel harmonization approaches relying on CNN structures, our work introduces an INR-based method meticulously tailored for processing high-resolution composite images in a dense pixel-to-pixel manner. Notably, it represents the first dense pixel-to-pixel high-resolution image harmonization method. The utilization of this dense pixel-to-pixel approach enables us to harness the full potential of deep networks, surpassing the capabilities of hand-crafted filters commonly employed in color-to-color methods. This enhancement empowers us to effectively address more complex scenarios, ultimately yielding state-of-the-art performance.
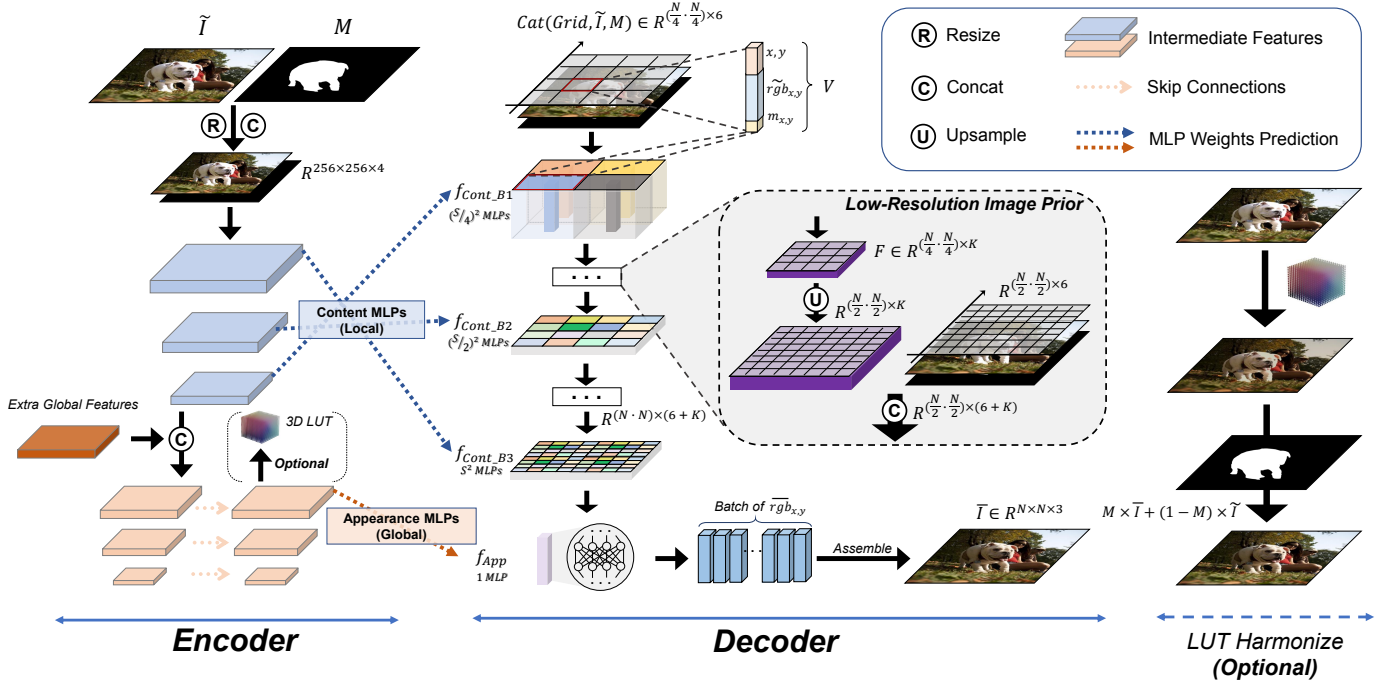
Fig. 2. The pipeline of our method. The HINet consists of an Encoder, a Decoder, and an optional LUT Harmonize module. Given a downsampled composite image $\widetilde{I}$ and its mask M, the Encoder predicts parameters of the decoder's MLPs and 3D LUT (optional). Fix the MLPs' parameters, we feed into the decoder a batch of vectors V, which is a concatenation of grid coordinate (x, y), value $m_{x,y}$ in M, and value $\widetilde{rgb}_{x,y}$ in $\widetilde{I}$. We then assemble the output vectors $\overline{rgb}_{x,y}$, and obtain harmonized images $\overline{I}$. Note that the number of layers in the figure is simplified, please refer to Sec. IV-A for more details. Details of the Encoder structure can be referred to [16], where "Extra Global Features" denotes the features from an additional HRNet [33].

Implicit Neural Representation. The INR method was originally proposed in [40], and has gained much popularity in 3D area recently [28], [41], [42], where it can represent a continuous 3D shape and is a memory-economic way compared to traditional approaches [43]–[47] such as point cloud and voxel. The key idea of INR is to convert originally sparse coordinates into continuous signals. Some recent studies [26], [27] show that INR with Fourier embedding and periodic activation like sinusoidal can be well applied to 2D area and represent photorealistic images. Since the coordinates are in a continuous real space, the generated images are then continuous. Inspired by INR, many recent works have explored leveraging it into different tasks and achieving some good results, *e.g.* image-to-image translation [48], image super-resolution [49], image generation [50]–[52], *etc*. Different from these works, we focus on building a dense pixel-to-pixel image harmonization network that can be applied to ultra-HR images. The HINet structure has been carefully designed and some interesting potentials for practical use have been explored.

III. PROPOSED METHOD

A. Overview

The HINet architecture consists of an encoder and a decoder. The encoder structure aligns with prior research [16], [25], incorporating additional global features within its intermediate layer. In the decoder segment, we adhere to the INR paradigm, employing a stack of MLPs. The overall architecture is illustrated in Fig. 2. Given a composite image $\widetilde{I}$, we input its resized version (256x256) and the corresponding mask M into the encoder. This step enables the prediction of parameters for the decoder MLPs. Once the decoder's weights are determined, we feed it with a batch of vectors V, with a batch size matching the pixel count of the original composite image. The vector V represents a concatenation of grid coordinates (x, y), mask values $m_{x,y}$, and composite image values $\widetilde{rgb}_{x,y}$. These vectors undergo processing by the decoder MLPs, yielding harmonized signals $\overline{rgb}_{x,y}$. By assembling these output RGB values, we generate the final harmonized image $\overline{I}$.

In the subsequent subsections, we will delineate the challenges associated with implementing INR for image harmonization in Section B. Following this, Sections C through E will provide an in-depth exposition of our network designs, specifically crafted to address these challenges.

B. Analysis of Existing Challenges

Leveraging INR for the image harmonization task is a challenging task. Recently, there have been many approaches [48], [50]–[52] applying INR to tasks like image generation and image translation. These methods usually take encoder's output features as the MLPs' weights and get generated images by feeding in coordinates. Some of them apply a stack of globally representative MLPs [50] where every coordinate is processed by the same MLP, while others apply locally representative MLPs [48] that coordinates are split into different parts and processed by corresponding MLPs. Although these methods achieved attractive results in some tasks, there are

mainly three challenges in transferring to image harmonization task.

The first challenge is the design of the INR structure. Since the harmonization task requires preserving content structure while aligning color space between foreground and background, we may encounter content loss [48] if we choose to utilize global MLPs. Whereas if we choose to leverage local MLPs, the memory cost will dramatically increase which is unaffordable, especially for real harmonization scenarios where images can reach high resolutions, and it may also introduce the problem of inconsistency in the boundaries of adjacent image regions (see Sec. III-D).

The second challenge is the insufficiency of the encoder output features. Existing methods mostly predict the parameters of all MLPs only from the features of the last encoder layer. Although deep layers do capture rich semantic information, content structure information is not well preserved, and it is burdensome to predict a large number of parameters by a single layer. A downsampling operation may reduce the number of parameters, yet inevitably decrease the output image fidelity due to more structural information loss.

The third challenge is how to perform HR training and inference without consuming too much memory. Real-world harmonization scenarios often encounter ultra-HR images, *e.g.*, $6K$. Even just harmonizing a single one (*e.g.*, 6048×4032 pixels) will consume lots of memory as the vectors input to INR can build a huge batch ($\approx 10^7$). Dealing with such an ultra-HR problem is still underexplored by previous INR works.

Considering the aforementioned challenges, we have meticulously crafted HINet. These challenges are individually tackled by our solutions in Sec. III-C, Sec. III-D, and Sec. III-E.

C. Decoupled Content and Appearance MLPs

Existing INR approaches either leverage a stack of globally representative MLPs [50], [51] where each layer is a single MLP, or locally representative MLPs [48] where each layer is an MLP matrix that consists several MLPs (The differences are displayed in Fig. 2). The former may have a low fidelity problem [26], while the latter may have an out-of-memory problem as there are more MLPs.

Referring to Retinex theory [29], [30] that decomposes an image into illumination and reflectance, we can also decompose the harmonization task into two pieces: determining the environment and the content objects of an image. Along with this idea, we decouple the MLPs in the decoder into a content extraction part f_{Cont} and an appearance rendering part f_{App}. Specifically, f_{Cont} leverages locally representative MLPs to both extract objects information and ensure content structure retention, while f_{App} adopts a globally representative MLP to capture the background environmental condition. Since the content structure is mostly preserved in low-level features and the objects recognition only requires local receptive fields, we predict the parameters of f_{Cont} by shallow encoder layers. For f_{App}, we predict its parameters from deep layers that can capture global and high-level features, thus conducive to environment capture.

(a) Composite image **(b) Boundary inconsistency problem**

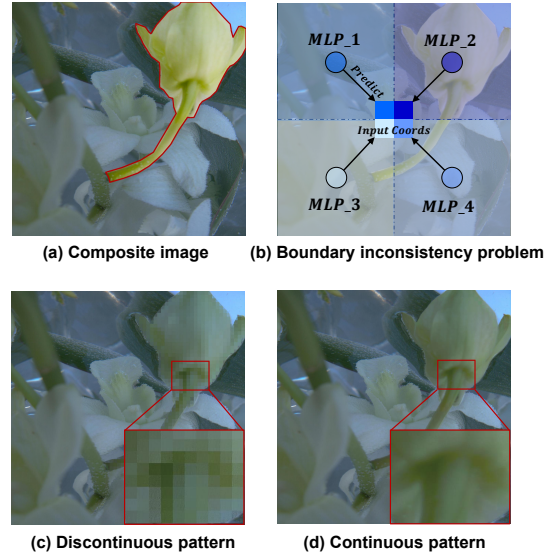**(c) Discontinuous pattern** **(d) Continuous pattern**

Fig. 3. Boundary Inconsistency Problem Illustration. When utilizing locally representative MLPs on a composite image (a), the inconsistency problem emerges. To clearly illustrate this, we consider only four local MLPs, as depicted in (b). Near the boundary of these MLPs, the MLP processing the input coordinate abruptly transitions from one to another, causing a discontinuous pattern, as seen in (c). In contrast, our designed LRIP structure ensures a continuous result, as depicted in (d).

It should be noted that recent works [13], [15] also build networks from the perspective of Retinex. However, unlike their strict adherence to Retinex theory (explicitly output illumination and reflectance images, and then multiplying the two), we implicitly embed the idea into the decoder structure design to extract content and environment information. To the best of our knowledge, our design of decoupled MLPs has not been previously explored by other INR works. Moreover, such a design not only offloads the last layer of the encoder and makes the structure of the encoder well aligned with that of the decoder, but also benefits from both local and global MLPs (as shown in Tab. VIII).

D. Low-Resolution Image Prior

In this section, we first discuss the boundary inconsistency problem caused by the design of the f_{Cont}, then we introduce our solution. To be specific, since f_{Cont} adopts an MLP matrix structure, the input is divided into several parts, each corresponding to a specific MLP. Therefore, at the boundary of two adjacent MLPs, the processing MLP will suddenly switch from one to another, leading to a discontinuous pattern. We illustrate this problem in Fig. 3.

A very straightforward solution for the above problem is to leverage bilinear interpolation instead of the nearest match. For each input, we query its nearest four corner MLPs and calculate the interpolated MLP. Take the four corner MLPs as f^p, where $p \in \{1:4\}$, from the top-left corner to the bottom-right one, and the area enclosed by the current position and each corner as s^p, then the generated MLP can be formulated as:

$$f_{gen} = \sum_{p \in \{1:4\}} \frac{s^{p'}}{s^{all}} \cdot f^p \tag{1}$$

where $s^{all} = \sum_{p \in \{1:4\}} s^p$ and p' is the opposite corner of p. In this way, each input vector is processed by a continuous MLP matrix, thus alleviating the inconsistency problem. However, although the bilinear interpolation strategy looks simple and effective, it is not feasible in practice. For example, suppose the original MLP matrix is composed of 16×16 MLPs, in this case, even if we harmonize an LR 256×256 image, then by using the interpolation strategy, we will finally generate a 256×256 MLP matrix, an unbearable $\times 256$ increase in the number of parameters.

To both reduce the memory cost and avoid the boundary inconsistency between blocks, we propose a new network structure named "Low-Resolution Image Prior (LRIP)". Specifically, we divide the MLPs in f_{Cont} into several blocks. We feed the input vectors to each block, and the batch size increases hierarchically. Except for the first one, each block is conditioned on the output features of the previous block (See Fig. 2 for more details). Given a 256×256 image and an LRIP structure with two blocks B_1, B_2 (for simplicity), the process is defined as follows:

$$F_1 = B_1(V_{128^2}) \tag{2}$$

$$F_2 = B_2(Cat(V_{256^2}, Up(F_1))) \tag{3}$$

where V_{N^2} denotes the input vectors with a batch size of N^2, F is the output feature which has the same batch size as the input, $Cat(\cdot)$ is the concatenation operation, $Up(\cdot)$ is the upsampling operation. We adopt the bilinear upsampling in LRIP. In this way, we convert the idea of continuous MLPs to continuous input. Each input is conditioned on the previous block, thus learning a more global representation and can alleviate the inconsistency problem effectively. Furthermore, since the blocks (except the last one) have a lower resolution input, the LRIP structure can save a lot of memory while maintaining high-quality results.

Note that in [51], the authors designed multi-scale INRs which seems similar to our LRIP structure. However, there are many differences. The main difference is in the decoder structure. Due to the different aims, [51] merely used a stack of global MLPs as the decoder structure, while in our design, we need to leverage local MLPs to ensure content retention. Furthermore, their decoder's parameters are all from the output features of the encoder's last layer, while ours are well aligned with the encoder structure which can make full use of all encoder layers.

E. HR Image Harmonization

Compared with LR image harmonization, it is more challenging to harmonize an HR image, especially for dense pixel-to-pixel transformation methods. We, therefore, propose designs for both the training and inference processes.

Multiple inputs. The conventional input of the INR is the coordinate (x, y) [26], [27], [50], [51]. Although with only the coordinate as input, we can achieve some good results in LR image harmonization, the quality deteriorates sharply when harmonizing higher-resolution images. Considering that the network only sees the LR image (the encoder's input is a down-sampled version of the image), when applied to HR

image harmonization, the decoder is required to not only do harmonization but also super-resolution. We show that this will be a much more challenging task and the network just fails to achieve both. Therefore, in practice, apart from the coordinate (x, y), we also feed the composite image's RGB value $\widetilde{rgb}_{x,y}$ and the mask value $m_{x,y}$ into the input vector, which is expected to provide guidance for the decoder and make it focus on the harmonization task. In this way, our input is finally a 6D vector that can be formulated as $V = (x, y, \widetilde{rgb}_{x,y}, m_{x,y})$.

HR training process. When harmonizing LR images, the training process is straightforward and we can just feed all the input vectors into the decoder. However, as mentioned in Sec. I, it is unaffordable for HR image harmonization due to the huge batch size. Benefiting from the advantage of the INR that the input is a batch of vectors, rather than a whole feature matrix as CNN, we can feed partial vectors into the decoder, and we design a Random Step Crop (RSC) strategy which is simple but effective. To be specific, we crop out the same local area from the composite image, the coordinate map, and the mask, and feed the vectors in this area into the decoder. The RSC strategy is somewhat similar to the regular RandomCrop augmentation, except that we not only need to crop the original resolution images but also the downsampled ones to meet the needs of the LR input in the LRIP structure (Please refer to Sec. III-D.). Furthermore, the motivation of the RSC strategy is for the feasible HR image training but not the data augmentation. Following [25], we also employ a progressive training strategy, first training on LR images and then finetuning on HR ones, which can bring in better results.

HR inference process. Similar to the problem in the training process, the inference can also encounter a memory problem. Here we split the input batch into several sub-batches along the image's row dimension (also can be along the column), feed these sub-batches into the decoder one by one, and then assemble them as the harmonization result. Another problem is that in the real world, the resolution of many images is not divisible by the downsampling multiple of the LRIP structure. To deal with that, each block of LRIP takes input vectors of the same batch size (identical to the composite image's size) instead of different ones.

Optional 3D LUT prediction. In the proposed HINet, we can optionally predict the 3D LUT parameters for harmonization. The 3D LUT is a lookup table that maps an RGB value to another value, with which we can harmonize the foreground of the composite image. The motivation of this optional design is for facilitating the manual control of users and enhancing the network's comprehensibility. Since 3D LUT is essentially a global transformation, we predict its parameters by the features for f_{App}. Different from the elaborately designed filters in [24], [25], we predict the 3D LUT parameters directly and can obtain competitive results. Note again that the 3D LUT is optional and the network performance is not affected by its existence (see Tab. XII).

It is also worth noting that the LUT prediction is independent of our INR decoder. That is, users have the flexibility to choose which to use in inference based on their preference for higher harmonization quality (INR decoder) or more control over the result (3D LUT). This is quite different from recent

TABLE I
COMPARISONS WITH RECENT STATE-OF-THE-ART HR HARMONIZATION METHODS [23]–[25]. SINCE CDTNET [23] IS NOT OPEN SOURCE AND ONLY WORKS ON CERTAIN HIGH RESOLUTIONS AS DISCUSSED IN SEC. II, WE DIRECTLY QUOTE THEIR CDTNET-256 RESULTS (NOT CDTNET-512, WHOSE INPUT CONFIGURATION IS NOT ALIGNED WITH OTHER HR METHODS) ON HADOBE5K SUB-DATASET. FOR HARMONIZER [25] AND DCCF [24], AS THERE MISSED SOME METRIC RESULTS IN THE ORIGINAL PAPERS, WE RE-RUN THEIR INFERENCE CODE ON THE ORIGINAL RESOLUTION VERSION OF THE IHARMONY4 DATASET [12] WITH THE SAME DEVICE. THE BEST RESULT IS SHOWN IN BOLD.

HAdobe5K	Metric	CDTNet [23]	Ours
1024 × 1024	MSE↓	**21.24**	22.68
	fMSE↓	**152.13**	187.97
	PSNR↑	**38.77**	38.38
	SSIM↑	0.9868	**0.9886**
2048 × 2048	MSE↓	29.02	**24.08**
	fMSE↓	198.85	**192.20**
	PSNR↑	37.66	**38.35**
	SSIM↑	0.9845	**0.9886**
Original resolution ($\sim 6K$)	MSE↓	-	**21.81**
	fMSE↓	-	**173.72**
	PSNR↑	-	**38.71**
	SSIM↑	-	**0.9871**

| Dataset	Metric	Harmonizer [24]	DCCF [25]	Ours
HAdobe5K	MSE↓	24.09	23.12	**21.45**
	fMSE↓	193.70	195.60	**172.79**
	PSNR↑	37.82	37.78	**38.67**
	SSIM↑	0.9339	0.9858	**0.9873**
HCOCO	MSE↓	20.39	**16.84**	17.29
	fMSE↓	364.52	317.43	**315.98**
	PSNR↑	37.80	**38.65**	38.65
	SSIM↑	0.9858	**0.9929**	0.9927
Hday2night	MSE↓	**37.72**	55.78	51.24
	fMSE↓	**636.04**	715.52	713.66
	PSNR↑	37.20	**37.52**	37.35
	SSIM↑	0.9548	0.9787	**0.9801**
Hflickr	MSE↓	67.82	**64.62**	66.56
	fMSE↓	473.30	**438.44**	449.71
	PSNR↑	33.44	**33.61**	33.56
	SSIM↑	0.9714	0.9843	**0.9844**
All	MSE↓	27.09	24.72	**24.62**
	fMSE↓	331.73	302.57	**296.31**
	PSNR↑	37.31	37.84	**38.07**
	SSIM↑	0.9685	0.9896	**0.9900**
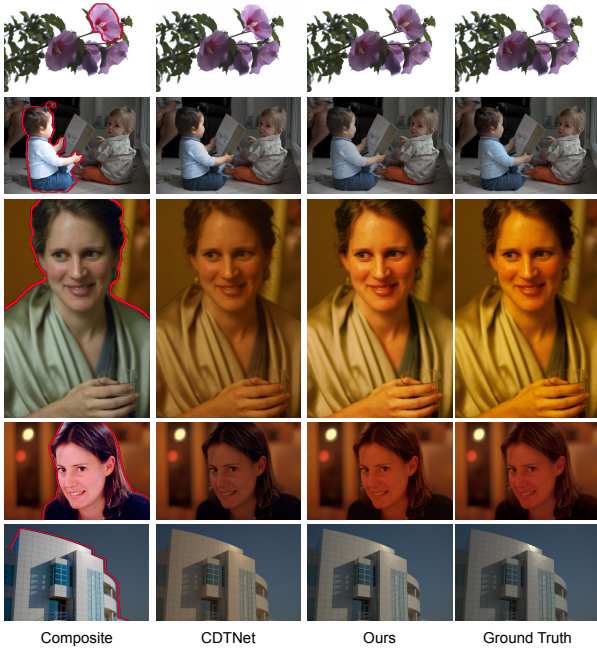
Fig. 4. Visual comparisons on 2048×2048 HR version of HAdobe5K sub-dataset. From left to right, we show the composite images, the results of [23] and ours, and the ground truth images. The foreground is stroked by a red line. We have resized the images to their original aspect ratio for a better view.

Fig. 5. Visual comparisons on the original resolution of iHarmony4 dataset (resolution can reach $6K$). We are the first dense pixel-to-pixel method that can be applied to the original resolution. From left to right, we show the composite images, the results of [24], [25] and ours, and the ground truth images. The foreground is stroked by a red line. Please zoom in for a better view.

CDTNet [23] as LUT prediction is an integral part of their final structure.

IV. EXPERIMENTS

A. Experimental Settings

Datasets. We follow previous papers to train and evaluate our method on the benchmark dataset iHarmony4 [12], which is synthesized using color transformation methods such as [53] and consists of 4 sub-datasets (HAdobe5k, HCOCO, Hday2night, and HFlickr), with 73146 images in total. For the HR image harmonization, we follow [23] to evaluate on HAdobe5k sub-dataset that consists of HR images among the four sub-datasets, and also follow [24], [25] to evaluate on the original resolution iHarmony4 dataset without any downsampling operation. To further illustrate the effectiveness, we follow the existing approaches [13], [14], [17], [23] and evaluate our method on 99 LR real composite images released by [11] and 100 HR ones released by [23].

Evaluation metrics. We follow the previous methods and evaluate the harmonization performance with Mean Squared

Fig. 6. Visual comparisons on 256×256 LR version of iHarmony4 dataset. From left to right, we show the composite images, the results of [12]–[15] and ours, and the ground truth images. The foreground is stroked by a red line. We have resized the images to their original aspect ratio for a better view.
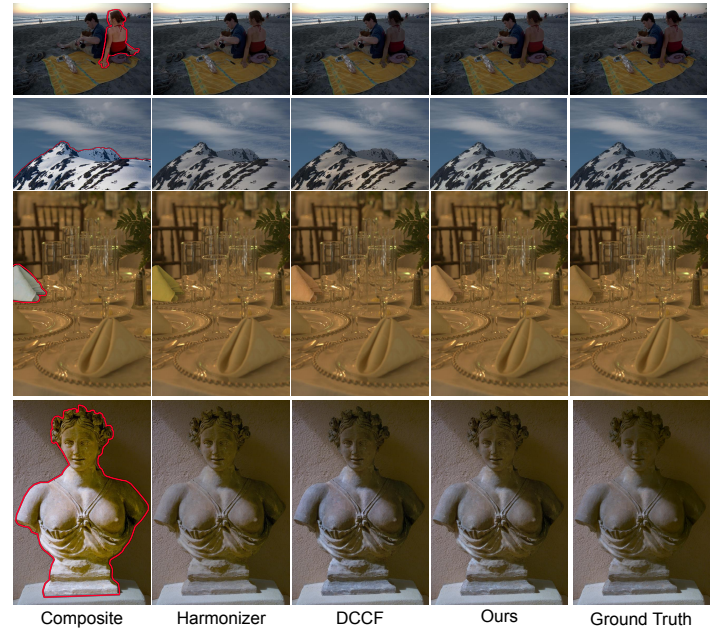

Fig. 7. Visual comparisons on 100 HR real composite images (1024×1024) released by [23]. As there is no ground truth, from left to right, we show the composite images, the masks, and the results of [23] and ours. We have resized the images to their original aspect ratio for a better view.

Fig. 8. Visual comparisons on 99 LR real composite images (256×256) released by [11]. As there is no ground truth, from left to right, we show the composite images, the masks, and the results of [12]–[15] and ours. We have resized the images to their original aspect ratio for a better view.

Error (MSE), foreground MSE (fMSE, only consider the foreground area), Peak Signal-to-Noise Ratio (PSNR), and Structure Similarity Index Measure (SSIM).

Implementation details. We adopt the iDIH-HRNet [16] as the encoder structure. The first three encoder layers are leveraged to predict parameters of f_{Cont} which is a three-block LRIP structure, while the remained layers are utilized to construct a U-Net [22] like structure and the output is for predicting f_{App} and the optional 3D LUT (See Fig. 2 for more details). If not specified, the number of the three LRIP blocks' hidden layers are 3, 2, and 1 respectively, and we set 2 hidden layers for f_{App}. All the hidden layers are of 32 dimensions. We adopt the same positional embedding as [50] and also leverage the Factorized Multiplicative Modulation [51] to reduce redundant parameters. The 3D LUT dimension is set to 7.

We only utilize L2 loss to supervise the harmonization results, in addition to an extra regularization ensuring that the 3D LUT values do not overflow. We adopt AdamW [54] optimizer with an initial learning rate $1e^{-4}$. We train our model for 60 epochs with a batch size of 16, and the learning rate decays in a Cosine Annealing strategy. The model is implemented with Pytorch and we conduct training and evaluation on a single RTX 3090 GPU.

B. Comparison with Existing Methods

HR image harmonization. We here compare our method with the recent HR image harmonization methods [23]–[25], [37], which leverage color-to-color transformations. We conduct experiments on 1024×1024 and 2048×2048 versions of HAdobe5K sub-dataset, in the same way as in [23]. Results in Tab. I show that we achieve better performance than [23] on higher resolution (2048×2048). Although [23] performs well on 1024×1024, with the resolution increasing, its performance drops sharply, while our method maintains stable performance and even achieves better results on the original resolution.

Following [24], [25], we conduct the harmonization experiment on the original resolution of iHarmony4 dataset. It is worth noting that within the four sub-datasets of iHarmony4, HAdobe5K stands out as the sole dataset predominantly comprising images with resolutions ranging from 2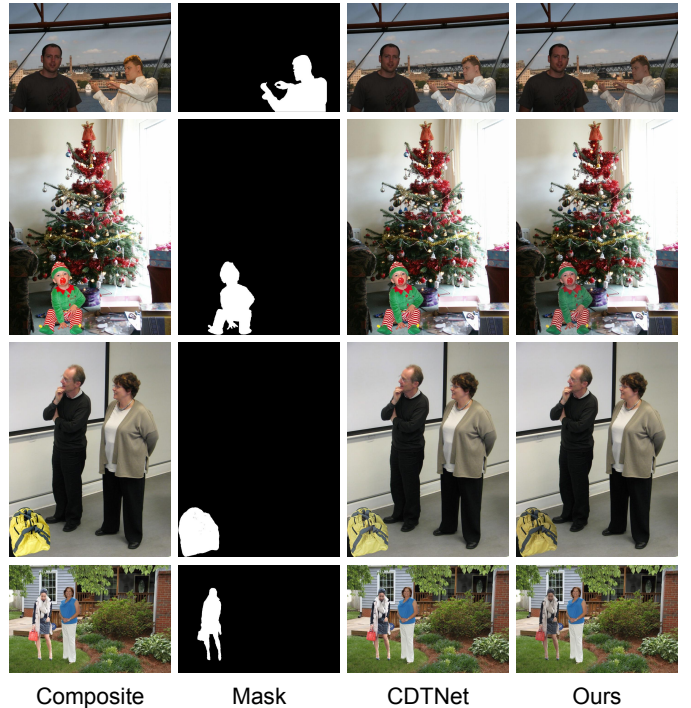K to 6K resolution (1944~6048), while all other datasets feature images with resolutions below 1024 (HCOCO/120~640, Hday2night/313~854, Hflickr/150~1024). From Tab. I, our method substantially beats the previous DCCF in all metrics (*e.g.*, PSNR 38.67 *vs.* 37.78) on HAdobe5K. This demonstrates our method's effectiveness in handling ultra-HR images and underscores our superiority in real-world harmonization scenarios that frequently involve HR imagery. Besides, we can also observe that the HINet outperforms the two state-of-the-art HR harmonization methods on the entire iHarmony4 dataset.

We also compare with the recent S^{2}CRNet [37]. Since their open-source pre-trained model requires extra semantic labels as input, we do not display the comparison results in

TABLE II

COMPARISONS ON 256×256 LR VERSION OF THE iHARMONY4 DATASET [12] WITH OTHER DENSE PIXEL-TO-PIXEL HARMONIZATION METHODS. THE BEST RESULT IS SHOWN IN BOLD, WHILE THE SECOND BEST IS UNDERLINED. THE ARROW OF THE METRIC INDICATES IN WHICH DIRECTION THE VALUE IS BETTER. THE RESULTS OF THE STATE-OF-THE-ART METHODS ARE QUOTED FROM THEIR SOURCE PAPERS, WHILE "-" INDICATES THAT NO RELEVANT RESULTS ARE PROVIDED.

Dataset	Metric	DIH [11]	S²AM [35]	DoveNet [12]	BargainNet [55]	RainNet [14]	IntrinsicIH [15]	IHT [13]	iDIH-HRNet [16]	CDTNet [23]	Ours
HAdobe5K	MSE↓	92.65	48.22	52.32	39.94	-	43.02	38.53	<u>21.80</u>	**20.62**	23.11
	fMSE↓	-	-	-	-	-	284.21	265.11	-	-	170.85
	PSNR↑	32.28	35.34	34.34	35.34	36.22	35.20	36.88	37.19	<u>38.24</u>	**38.31**
HCOCO	MSE↓	51.85	33.07	36.72	24.84	-	24.92	16.89	**13.93**	<u>16.25</u>	16.41
	fMSE↓	-	-	-	-	-	416.38	299.3	-	-	296.45
	PSNR↑	34.69	36.09	35.83	37.03	37.08	37.16	38.76	**39.63**	39.15	<u>39.16</u>
Hday2night	MSE↓	82.34	48.78	54.05	50.98	-	55.53	53.01	60.18	**36.72**	<u>51.60</u>
	fMSE↓	-	-	-	-	-	797.04	704.42	-	-	670.32
	PSNR↑	34.62	35.60	35.18	35.67	34.83	35.96	37.10	37.71	**37.95**	<u>37.81</u>
HFlickr	MSE↓	163.38	124.53	133.14	97.32	-	105.13	74.51	**59.42**	68.61	<u>68.52</u>
	fMSE↓	-	-	-	-	-	716.60	515.45	-	-	448.77
	PSNR↑	29.55	31.00	30.21	31.34	31.64	31.34	33.13	**33.88**	<u>33.55</u>	33.53
All	MSE↓	76.77	48.00	52.36	37.82	40.29	38.71	30.30	**22.15**	<u>23.75</u>	24.82
	fMSE↓	773.18	481.79	549.96	405.23	469.60	400.29	320.78	<u>256.34</u>	**252.05**	283.56
	PSNR↑	33.41	35.29	34.75	35.88	36.12	35.90	37.55	<u>38.24</u>	38.23	**38.26**

TABLE III

USER STUDY ON 99 LR REAL COMPOSITE IMAGES RELEASED BY [11]. FOR THE VOTING METRIC, EACH USER CAN SELECT MORE THAN ONE REALISTIC IMAGE. "TOTAL VOTES" REPRESENTS THE NUMBER OF TIMES ONE METHOD'S RESULTS ARE CHOSEN AS REALITY. "RATIO" DENOTES THE PERCENTAGE AMONG ALL VOTES. FOR THE B-T SCORE, EACH USER MUST CHOOSE THE PREFERRED ONE FROM A PAIR OF TWO IMAGES. THE BEST VALUE IS SHOWN IN BOLD AND THE SECOND BEST IS UNDERLINED.

Metric		Composite	DoveNet [12]	RainNet [14]	IntrinsicIH [15]	IHT [13]	Ours
Voting	Total votes	161	215	293	**326**	299	<u>306</u>
	Ratio	10.06%	13.44%	18.31%	**20.38%**	18.69%	<u>19.13%</u>
B-T Score		0.0688	0.1375	0.1848	0.1901	**0.2285**	<u>0.1903</u>

TABLE IV

USER STUDY ON 100 HR REAL COMPOSITE IMAGES RELEASED BY [23]. FOR THE VOTING METRIC, EACH USER CAN SELECT MORE THAN ONE REALISTIC IMAGE. "TOTAL VOTES" REPRESENTS THE NUMBER OF TIMES ONE METHOD'S RESULTS ARE CHOSEN AS REALITY. "RATIO" DENOTES THE PERCENTAGE AMONG ALL VOTES. FOR THE B-T SCORE, EACH USER MUST CHOOSE THE PREFERRED ONE FROM A PAIR OF TWO IMAGES.

Metric		Composite	CDTNet [23]	Ours
Voting	Total votes	274	349	**512**
	Ratio	24.14%	30.75%	**45.11%**
B-T Score		0.260	0.308	**0.432**

Tab. I. Referring to the original paper [37], both S²CRNet-S and S²CRNet-V get about 36∼37 PSNR on 2048×2048 HAdobe5K, while ours can get 38.35 PSNR (Tab. I). Therefore, our method is still better even without additional label input. We also extend our comparison to more recent works [38], [39]. We cite their results directly from their respective source papers. When evaluated on the 2048×2048 HAdobe5k dataset, our method achieves a PSNR of 38.35, slightly outperforming [38] which attains 38.29. While [39] achieved notable results with their ViT backbone on the full-resolution iHarmony4 dataset, when evaluated under a similar experimental setup (using a CNN backbone and only L2 loss), our results remained competitive, with our method reaching 38.07 PSNR compared to their 38.05 PSNR.

For visual comparisons, we display the results on 2048 × 2048 version of HAdobe5K sub-dataset in Fig. 4, aligned with [23], and display the results on the original resolution of iHarmony4 dataset (resolution can reach $6K$) in Fig. 5, aligned with [24], [25].

LR image harmonization. We also evaluate our method on LR image harmonization in Tab. II. Since the existing pixel-to-pixel methods [11]–[16], [35], [55] cannot be applied to HR images, for fair comparisons, we conduct experiments on 256×256 LR iHarmony4 dataset and achieve competitive performance. We also take [23] into consideration which is a combination of pixel-to-pixel and color-to-color transformations, while we do not consider the recent [17] as it introduces extra training data. From the results, we can observe that the HINet can achieve competitive results on LR image harmonization compared with other state-of-the-art methods. Considering the comparison results on HR images displayed in Tab. I, it can be seen that our method achieves more performance gains as the image resolution increases. We visualize the comparisons on 256×256 version of iHarmony4 dataset in Fig. 6.

Real composite images. Since the iHarmony4 is a synthetic dataset [12], to better reveal the performance of our method on real images, we follow [23] to harmonize 100 HR real composite images released by [23] in Fig. 7 and also follow [14], [15], [17] to visualize the harmonization results on 99 LR real composite images released by [11] in Fig. 8. We conduct user studies for a fair comparison. The result shows the superiority of our method.

Since there are no ground truth images for the real composite images, we cannot leverage the former metrics (PSNR, MSE, and fMSE) to evaluate our performance. Here, we

TABLE V
EFFCIENCY COMPARISON ON A SINGLE 2048×2048 IMAGE.

Metrics	Color-to-Color Methods			Dense Pixel-to-Pixel Methods		
	S^2CRNet-S	Harmonizer	DCCF	IntrinsicIH	IHT	Ours
Params(M)	1.15	4.73	18.09	33.80	21.80	38.21
MACs(G)	0.605	0.036	12.677	OOM	OOM	36.484
Mem(M) (Train)	9621	4732	4159	OOM	OOM	**2835**
Mem(M) (Inference)	346	931	1525	OOM	OOM	7365/2439/**831**
Time(s)	0.16	0.15	0.14	OOM	OOM	**0.23**/0.52/1.12

TABLE VI
THE EFFICIENCY OF THE DESIGN OF MLPS DECOUPLING AND LRIP
STRUCTURE. THE METRICS ARE EVALUATED ON 256×256 IMAGES WITH
A BATCH SIZE OF 2.

Structure	Params (M)	Mem (MB)
Pure local MLPs with LRIP	40.38	1233.20
LRIP Blocks with the same size inputs	**38.21**	1411.16
ours	**38.21**	**1137.52**

TABLE VII
THE EFFICIENCY OF THE DESIGN OF THE RSC TRAINING STRATEGY FOR
HR IMAGE HARMONIZATION. "DIRECT FINETUNE" DENOTES INPUTTING
ALL THE VECTORS FOR TRAINING, NOT USING THE RSC STRATEGY. THE
METRICS ARE EVALUATED ON IMAGES WITH A BATCH SIZE OF 2.

Resolution	Strategy	Mem (MB)
1024×1024	Direct finetune	4586.35
	RSC finetune	**4090.86**
2048×2048	Direct finetune	15641.33
	RSC finetune	**4090.86**

TABLE VIII
DEMONSTRATION OF THE EFFECTIVENESS OF MLPS DECOUPLING. THE
EXPERIMENTS ARE CONDUCTED ON LR iHarmony4 DATASET.

Structure	MSE↓	fMSE↓	PSNR↑
Pure global MLPs	25.35	290.01	38.13
Pure local MLPs	25.56	294.97	38.00
Decoupled MLPs (ours)	**24.82**	**283.56**	**38.26**
Params from the last layer	25.17	288.04	38.17
Params from multiple layers (ours)	**24.82**	**283.56**	**38.26**

follow [14] to conduct a user study. Specifically, we invite 8 volunteers to choose the most realistic one/ones from the composite image and the results of the compared methods. Each time, we present an image and its variants to the user. The order is randomly shuffled, thus the users do not know which method the image belongs to. Every volunteer will evaluate the whole 100 HR real composite images and the 99 LR ones. From the results in Tab. IV and Tab. III, we can observe that our method can achieve the best performance on HR image harmonization and achieve competitive performance with the state-of-the-art methods on LR image harmonization.

Additionally, we adopted the Bradley-Terry model (B-T model) [56] for ranking, following [12], [24]. In this metric, volunteers were presented with pairs of results, randomly sampled from all methods (including composite images). They were required to select the preferred result in each pair. Pairwise comparisons were conducted on the 100 HR real composite images and the 99 LR ones, resulting in 1485 LR pairs and 300 HR pairs. We invited another 5 volunteers to participate in this ranking study, and the corresponding B-T scores are provided in Tab. IV and Tab. III. Notably, the conclusions drawn from the aforementioned voting metric remain robust, despite some variations in the rankings of LR images.

C. Efficiency Analyses

In Tab. V, we take a single 2048×2048 image as an example and compare the efficiency with existing methods from the perspective of model parameters (Params), calculation amount (MACs), memory overhead (Mem) during training and inference, and inference runtime (Time). For the memory cost and runtime of inference, we show results when the input is split into 1/4/16 parts (see Sec. III-E). From the results, thanks to the RSC strategy designed in Sec. III-E, we have the lowest training memory cost (less than 3GB) among all methods. By varying the number of input splits, we can achieve competitive performance with color-to-color methods either on inference memory cost or runtime. Regarding model parameters, our approach offers flexibility during the inference phase. We can

prune it to exclusively employ our INR decoder (34.71M parameters) or solely utilize the predicted 3D LUT (23.98M parameters), adapting to different scenarios. Consequently, we can match the parameter count with other dense pixel-to-pixel methods. Most importantly, we hope to draw the attention that our method is the first dense pixel-to-pixel method that can handle HR images ($\sim$6K, others encounter out-of-memory (OOM)). Considering pixel-to-pixel methods can model more complex scenarios than color-to-color ones (see Sec. II), our method, with SOTA harmonization performance and competitive efficiency, is of great significance and beneficial to this field's development.

Moreover. to validate the efficiency of our design, we here adopt Model Parameters (Params) and GPU Memory Cost (Mem) metrics. From Tab. VI, we can see that splitting MLPs into f_{Cont} and f_{App} can both reduce parameters and save memory cost compared with the structure of pure local MLPs, while the design of LRIP that leverages different batch sizes of input vectors can save much memory. From Tab. VII, we can observe that, with image resolution increasing, the memory cost of the direct finetuning grows sharply, while there is no impact on the memory cost of our RSC strategy, nor is the performance (Please see Tab. XII).

D. Ablation Studies

Effectiveness of MLPs decoupling. To verify the effectiveness of the decoupled design of f_{Cont} and f_{App}, we conduct experiments in Tab. VIII where we modify the HINet with pure global MLPs or local MLPs. We also compare with a modified version that the parameters of MLPs are all from the last layer of the encoder. The results have validated the superiority of our design.

Effectiveness of LRIP structure. We compare with the HINet without LRIP structure in Tab. IX, from which we can observe that the network with the LRIP structure achieves higher accuracy. To go a step further, we also compare with the structure with each block fed inputs of the same batch size

TABLE IX
DEMONSTRATION OF THE EFFECTIVENESS OF THE LRIP DESIGN. "NO LRIP" DENOTES ONLY THE FIRST BLOCK HAS THE INPUT VECTORS. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

Structure	MSE↓	fMSE↓	PSNR↑
No LRIP	25.65	288.41	38.10
Blocks with the same batch size inputs	25.01	285.79	38.16
LRIP (ours)	**24.82**	**283.56**	**38.26**

TABLE X
THE EFFECTIVENESS OF MULTIPLE INPUTS FOR HR IMAGE HARMONIZATION. THE EXPERIMENTS ARE CONDUCTED ON HADOBE5K SUB-DATASET. THE BEST RESULT IS SHOWN IN BOLD. "BILINEAR RESIZE" DENOTES INTERPOLATING LR HARMONIZATION RESULTS FOR HR RESULTS, WHILE "DIRECT QUERY" DENOTES QUERYING THE DECODER THAT IS TRAINED ON LR IMAGES FOR HR HARMONIZATION RESULTS.

Resolution	Input type & Strategy	MSE↓	fMSE↓	PSNR↑	SSIM↑
256×256	SingleInput	22.79	183.85	37.92	0.9894
	MultipleInput	**22.52**	**182.11**	**38.00**	**0.9897**
1024×1024	SingleInput + Bilinear resize	33.84	293.11	35.69	0.9691
	MultipleInput + Bilinear resize	**32.40**	**287.84**	**35.78**	**0.9701**
	SingleInput + Direct query	69.93	631.6	32.01	0.9346
	MultipleInput + Direct query	**23.44**	**194.7**	**38.02**	**0.9883**

TABLE XI
DEMONSTRATION OF THE EFFECTIVENESS OF RSC STRATEGY ON HADOBE5K DATASET. "DIRECT QUERY" DENOTES QUERYING THE DECODER THAT TRAINED ON LR IMAGES FOR HR HARMONIZATION RESULTS. "DIRECT FINETUNE" DENOTES INPUTTING ALL THE VECTORS FOR TRAINING, NOT USING THE RSC STRATEGY. "OOM" REPRESENTS THE OUT-OF-MEMORY PROBLEM.

Resolution	Strategy	MSE↓	fMSE↓	PSNR↑
1024×1024	Direct query	23.44	194.70	38.02
	Train from scratch	36.44	263.28	36.85
	Direct finetune	23.46	193.27	**38.38**
	RSC finetune	**22.68**	**187.97**	**38.38**
2048×2048	Direct query	26.04	212.46	37.76
	Direct finetune (OOM)	-	-	-
	RSC finetune	**24.08**	**192.20**	**38.35**
Original resolution ($\sim 6K$)	Direct query	32.63	246.33	37.07
	Direct finetune (OOM)	-	-	-
	RSC finetune	**21.81**	**173.72**	**38.71**

TABLE XII
DEMONSTRATION THAT THE HINET DOES NOT RELY ON 3D LUT PREDICTION. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4.

Metric	Only LUT prediction	Extra LUT Prediction	
		w/o	w/
MSE↓	-	**24.74**	24.82
fMSE↓	-	**283.05**	283.56
PSNR↑	-	**38.29**	38.26
LUT-MSE↓	25.81	-	**25.56**
LUT-fMSE↓	297.72	-	**293.66**
LUT-PSNR↑	38.03	-	**38.08**

TABLE XIII
EFFECT OF DIFFERENT NUMBERS OF CHANNELS IN MLPS ON THE FINAL PERFORMANCE. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

MLP width	MSE↓	fMSE↓	PSNR↑
16	25.68	293.70	38.04
32	24.82	283.56	38.26
64	**24.74**	**283.05**	**38.31**

TABLE XIV
EFFECT OF DIFFERENT NUMBERS OF HIDDEN LAYERS IN f_{Cont} ON THE FINAL PERFORMANCE. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

f_{Cont} structure	Depth	MSE↓	fMSE↓	PSNR↑
Same depth all blocks	1	24.82	284.91	38.15
	2	25.30	288.94	38.13
	3	24.89	282.58	38.24
Different depth each block	1, 2, 3	**24.26**	**282.00**	38.24
	3, 2, 1	24.82	283.56	**38.26**

TABLE XV
EFFECT OF DIFFERENT NUMBERS OF HIDDEN LAYERS IN f_{App} ON THE FINAL PERFORMANCE. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

f_{App} depth	MSE↓	fMSE↓	PSNR↑
1	24.91	286.64	38.23
2	24.82	**283.56**	**38.26**
4	**24.62**	286.73	38.17

TABLE XVI
PERFORMANCE OF DIFFERENT POSITION EMBEDDINGS. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

Positional embeddings	MSE↓	fMSE↓	PSNR↑
Nerf [28]	33.64	321.59	37.95
RFF [27]	28.15	326.94	37.62
INR-GAN [51]	25.48	285.64	**38.28**
CIPS [50]	**24.82**	**283.56**	38.26

TABLE XVII
PERFORMANCE OF DIFFERENT 3D LUT DIMENSIONS. THE EXPERIMENTS ARE CONDUCTED ON LR IHARMONY4 DATASET.

3D LUT dimensions	LUT-MSE↓	LUT-fMSE↓	LUT-PSNR↑
5	**25.55**	299.47	37.95
7	25.81	**297.72**	**38.03**
13	25.59	298.48	37.93
17	25.90	305.43	37.88

ground, while the other parts keep the same as the composite image).

To evaluate the effectiveness of the designed RSC HR training strategy, we conduct experiments in Tab. XI on the HAdobe5K sub-dataset, as it has much higher resolution among the four datasets in the iHarmony4 dataset [23]. We compare with results by directly querying the decoder trained on LR images for HR harmonization, and results by directly finetuning/training from scratch the network with HR images (here we only consider 1024×1024, images with higher resolution will encounter an out-of-memory problem). From the results, we can see that our proposed RSC training strategy greatly improves the performance of HR image harmonization.

In Tab. XII, we show that using an additional 3D LUT

(LRIP is fed inputs of gradually increasing batch size), and the LRIP again achieves better results.

Evaluation of specific designs. As mentioned in Sec. III-E, we leverage multiple inputs to help harmonize HR images. To compare with only using the grid coordinate as input, experiments are conducted as shown in Tab. X. We can observe that although different types of inputs have competitive results on LR image harmonization, when applied to HR images, only the coordinate input cannot achieve satisfying accuracy, even worse than using bilinear interpolation (only resize the fore-
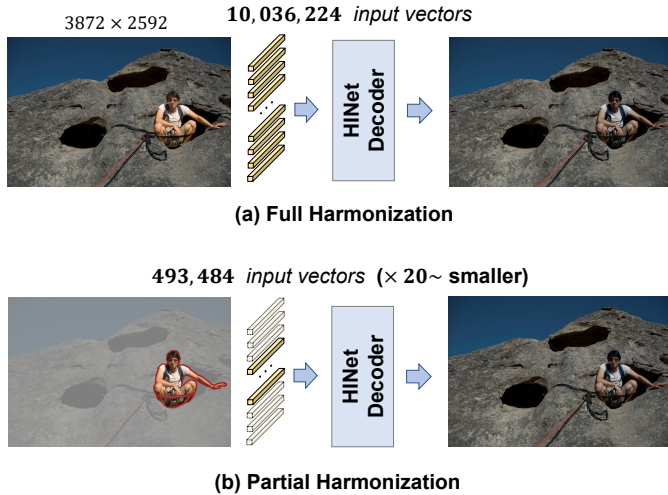
Fig. 9. Region-based harmonization of composite images. (a) displays the normal harmonization process that feeds all the vectors into the decoder, while (b) displays the region-based harmonization process.

prediction head has trivial impacts on the final harmonization results (even has a 0.03 PSNR drop). Thus we do not rely on it like [23]–[25]. Moreover, we can observe that with our method, the predicted 3D LUT can achieve better results.

Settings of MLPs width. We here test the effect of different numbers of channels in MLPs on the harmonization performance. From Tab. XIII, we can observe that with the MLPs' width increasing, the results get better. Considering the practical memory limitation, we choose 32 channels as the final design.

Settings of MLPs depth. In this part, we test different settings of the MLP hidden layers' number in f_{Cont} and f_{App} respectively. From the results in Tab. XIV, the effectiveness of adopting decreasing number of hidden layers for blocks in the LRIP structure is verified. Not only does it maintain harmonization performance, but such a structure also saves a lot of memory cost (the batch size of the input vectors increases from the first block to the last block). From the results in Tab. XV, we set two hidden layers for f_{App}.

Choices of positional embedding. As mentioned in [27], [28], a proper positional embedding is important for INR performance. Here we try several different positional embeddings [27], [28], [50], [51]. From the results in Tab. XVI, we finally adopt the positional embedding in [50].

3D LUT dimensions. To evaluate the effect of different 3D LUT dimensions, we conduct experiments in Tab. XVII. We can observe that there is a peak in the middle. Dimensions that are too large or too small can both lead to performance degradation. We consider that small dimensions may be not sufficient enough to cover the variation range, while large dimensions may cause redundant parameters' predictions, leading to much burden on the encoder.

V. APPLICATIONS

A. Region-based image harmonization

As mentioned in Sec. III-E, one property of the INR decoder is that the input is no longer a feature map but a batch of vectors. By leveraging this feature, we can achieve region-based harmonization by only feeding the vectors inside the foreground area into the decoder, while leaving the remained area untouched. In this way, the proposed HINet can harmonize a partial area of the composite image, thus saving much memory cost and achieving speedup. This is a feature that existing methods [16], [17], [23] do not have, since their CNN decoder needs to receive the output features of the entire image from the encoder as input, while we utilize the INR paradigm whose input is a batch of vectors V. Fig. 9 displays the region-based harmonization process. Given a $3K$ composite image, if we feed the vectors of the entire image into the decoder (somewhat similar to the CNN process), there will be about 10 million input vectors, which will cause much memory consumption and slow processing speed. While, if we apply region-based harmonization and only harmonize the vectors in the foreground region, the number of input vectors will be reduced to 0.5 million, just equivalent to processing half a $1K$ image!

Take advantage of this potential, we can even apply the HINet to video harmonization, while preventing much memory cost. We here just take the video as a stack of image frames, and the process is similar to Fig. 9. We display the video harmonization process in Fig. 10.

B. Arbitrary resolution image harmonization

One defect of the previous methods [15], [17] is that once the network structures are configured, these methods can only harmonize the images with a fixed resolution. If we want to harmonize images with other resolutions, the network structure must be re-configured and retrained. To demonstrate it more clearly, suppose that we have a pure convolution structure designed for 256×256 images, then when the input is 256×256 or a multiple of that, the network can handle it. However, if the input is *e.g.* 256×257 (only one value increased), then the network may fail to process it, as the resolution 257 is not divisible by the downsampling multiple of the network's encoder. It is feasible to resize the image to the nearest multiple of 256×256, but since the downsampling multiple of the encoder is usually large [16], much information may be lost if the image is directly resized to a lower resolution (See Fig. 11 as an example), while if the image is resized to a higher resolution, then there will be much redundant computational overhead. As a comparison, the proposed HINet can achieve arbitrary resolution image harmonization. Since our method is built based on the INR paradigm, we consider the task as harmonizing continuous images rather than discrete image arrays. Therefore, in our method, we only need to sample intermediate coordinates to construct an input batch with the size identical to the image, and then can produce a higher fidelity harmonization result than direct using interpolation, even if the target resolution has never been seen by the model (See Fig. 11 and the experiments in Tab. X). We illustrate this feature in Fig. 12 for better comprehensibility. Given vectors of different batch sizes, we can get harmonization results at different resolutions.
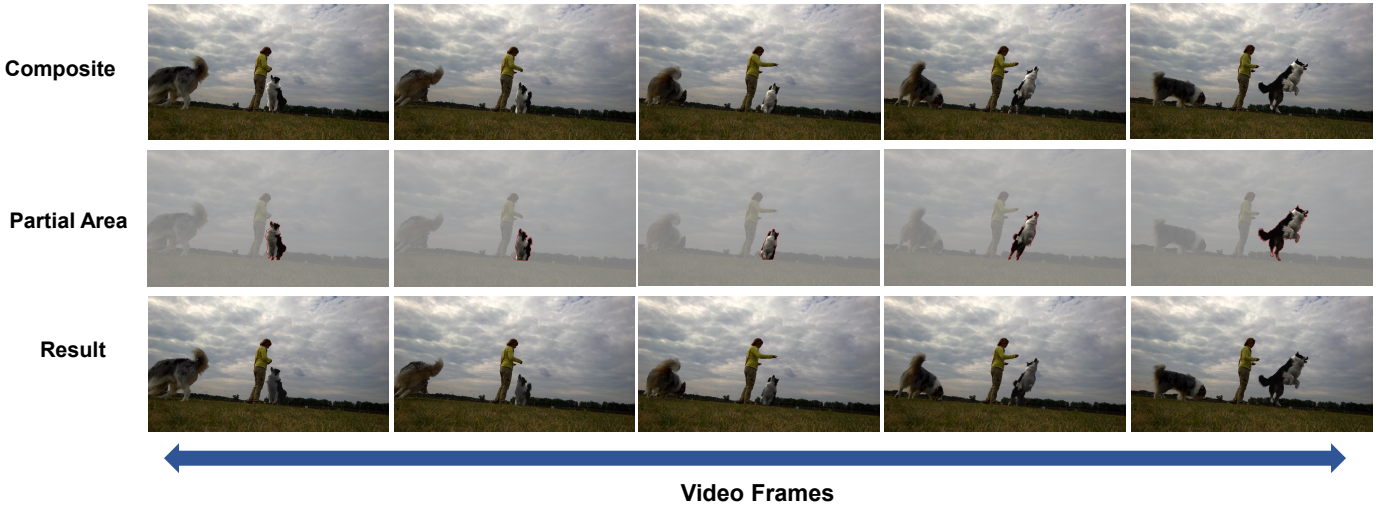
Video Frames

Fig. 10. Region-based harmonization of composite video clips. The first row denotes the composite frames. In the second row, we stroke the partial area whose vectors are fed into the decoder with a red line, while the remained untouched region is made transparent. We display our harmonization results in the third row. Please zoom in for a better review.

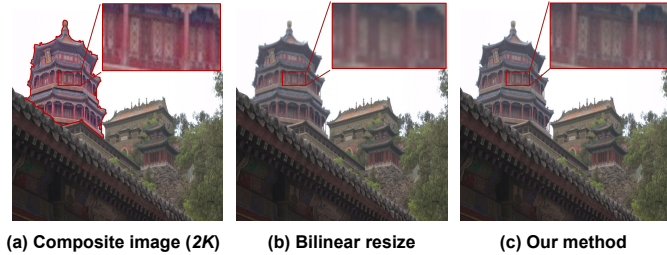**(a) Composite image (2K)**   **(b) Bilinear resize**   **(c) Our method**

Fig. 11. Arbitrary resolution harmonization potential of our method. Take a 2K image (a) as an example, the harmonization result by bilinear interpolation (b) losses much information, while ours (c) keeps high fidelity even though we only train the network on LR images (Please zoom in for a better view).
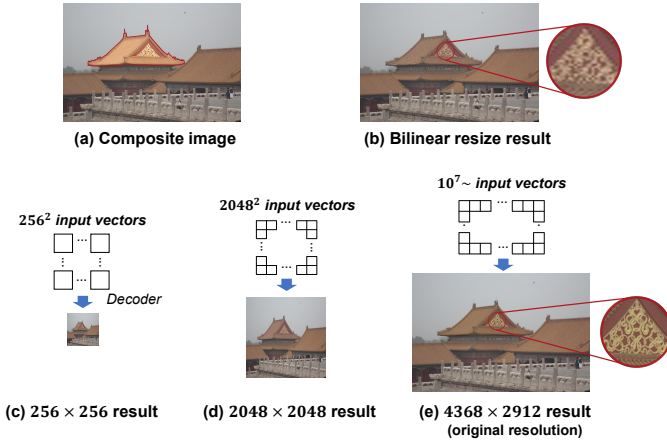

Fig. 12. Illustration of arbitrary resolution harmonization. (a) is the composite image with the foreground stroked by a red line. (c), (d), and (e) are harmonization results with different resolutions. (b) is the result of bilinearly resizing (c) (only resize the foreground, other region keeps the same as (a)). Please zoom in for a better view.

C. Optional usage of 3D LUT

In Sec. IV-D, we mention that the HINet can optionally predict a 3D LUT for controllable harmonization, and the exis-
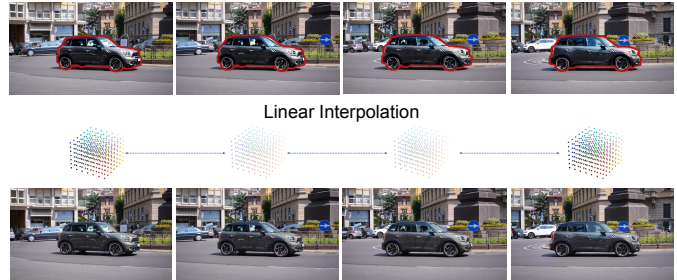

Linear Interpolation

Fig. 13. Illustration of smooth 3D LUT interpolation on video harmonization. The first row is frames of the composite video with the foreground stroked by a red line. The second row is the predicted 3D LUT. The third row is the harmonization result with the interpolated 3D LUT. Please zoom in for a better view.

tence of this optional part will not affect the final performance of the network. The motivation follows [24], [25] to make more space for manual control of the harmonization result, since the HINet is essentially a black box with little control by the user. Different from the complex design of the hand-crafted image filters in [24], [25], we adopt 3D LUT, which is a global RGB-to-RGB mapping. Although [24] claimed that directly predicting the filters' parameters cannot achieve good results, for which they utilized a hierarchical structure, in the HINet, we have no special design for 3D LUT prediction but directly predict it by the encoder's features, and the performance still looks good (Please refer to Tab. XII).

The potential usages of 3D LUT are two-fold. On the one hand, the users can better understand how the network harmonizes composite images. Along with that, they can manually change the harmonization result by modifying the parameters of the 3D LUT, which is easy with the help of PhotoShop and 3D LUT Creator tools. On the other hand, if high harmonization quality is not pursued, we can apply 3D LUTs to video harmonization for fast and continuous results. As mentioned in [24], harmonizing each video frame independently can lead to flickering phenomena, so they leverage the

exponential moving average on the hand-crafted image filters' arguments for smoothness. Inspired by them, we here linearly interpolate the parameters of 3D LUT for continuous results. Suppose the 3D LUT parameters at M_{th} frame is lut_M, that at N_{th} frame is lut_N, then the parameters at intermediate frames $(M : N)$ can be formulated as:

$$lut_K = lut_M + (lut_N - lut_M) \times \frac{K - M}{N - M}, K \in (M : N) \quad (4)$$

Then, we can extract several frames at intervals from the video, predict their 3D LUT, and interpolate the other frames'. It allows for video harmonization to be fairly quick and results to be continuous. We display the process in Fig. 13.

TABLE XVIII
PERFORMANCE ON VIDEO HARMONIZATION. THE EXPERIMENTS ARE CONDUCTED ON THE HYOUTUBE DATASET. "TL" DENOTES TEMPORAL LOSS. THE BEST RESULT IS SHOWN IN BOLD, WHILE THE SECOND BEST IS UNDERLINED.

Method	fMSE	MSE	PSNR	TL	Time(s)
Lu *et al.* [57]	186.72	26.50	37.61	**5.11**	2.37
Harmonizer	211.49	32.28	36.48	23.72	1.49
Harmonizer (EMA)	197.36	30.17	36.84	17.11	1.49
Ours(LUT)	176.99	24.45	38.56	7.09	<u>1.23</u>
Ours(LUT-Interpolation)	<u>171.80</u>	<u>23.73</u>	<u>38.71</u>	6.73	**0.86**
Ours(Decoder)	**159.42**	**22.38**	**39.12**	<u>6.06</u>	1.53

D. Video Harmonization

For a comprehensive evaluation, we extended our method to the video harmonization task and conducted experiments on the publicly available HYouTube dataset [57], consisting of 3194 video samples, each consisting of 20 frames. We resized the frames to 256×256, in alignment with [57]. We compared our approach with Harmonizer [24] and the video harmonization framework presented in [57]. Both Harmonizer and our method were trained from scratch on the training set of HYouTube. Regarding [57], we directly reference the results reported in their source paper. As Harmonizer is originally designed for image harmonization, we also compare its exponential-moving-average (EMA) variant tailored for video harmonization. We set its EMA coefficient to 1/6, corresponding to a frame interval of 167ms in HYouTube, which is also in line with Harmonizer's source code. For our method, we assess both the performance of the INR decoder and the 3D LUT. We also evaluated the 3D LUT interpolation strategy mentioned in Sec. V-C, where we sample five key frames from the 20-frame video clip. We measured performance using metrics such as fMSE, MSE, and PSNR. Additionally, we employed the Temporal Loss (TL) in [57] to assess temporal consistency and measured the inference time for processing a 20-frame video clip on a single RTX 3090 with the batch size set to 1.

The results, presented in Tab. XVIII, indicate that our method outperforms the other methods in almost every metric for video harmonization, except for temporal consistency. This demonstrates the superior generalization of our approach. Regarding temporal consistency, our 3D LUT interpolation

strategy not only enhances performance but also accelerates the harmonization process. In contrast, Harmonizer's EMA strategy, while reducing the TL metric value, does not improve processing speed since it still calculates every frame's result. Both Harmonizer and our method fall short when compared to [57]. We attribute this to the fact that [57] incorporates temporal information, considering several previous and future frames in their network's input, thereby having access to more temporal data. In contrast, Harmonizer and our approach rely on simpler averaging and interpolation strategies.

VI. LIMITATIONS

Although we have carefully designed the structure, the limitation of the INR still remains, especially on the speed performance when being applied to ultra-HR image harmonization. Since we need to split the input of the decoder into different parts to avoid being out of memory, the memory performance improves along with a certain speed sacrifice. Moreover, compared with the existing methods that leverage a U-Net like structure, there is still space to better fuse the shallow features and deep features in the HINet, which can provide richer features for the MLPs' predictions. We leave these limitations to our future work.

VII. CONCLUSION

In this paper, we explore a novel method for HR image harmonization with dense pixel-to-pixel transformations. We leverage the implicit neural representation and carefully design the decoder's structure to ensure visual harmony and reasonable memory cost. To our best knowledge, the proposed HINet is the first dense pixel-to-pixel harmonization method that can be applied to images $\sim 6K$ without any hand-crafted image filter and is also the first approach that leverages INR for the harmonization task. Experiments conducted on iHarmony4 dataset have demonstrated the effectiveness of our method for HR image harmonization. Some application potentials in practical usage are explored. We expect that our work can pave way for more research on deep learning-based HR image harmonization.

REFERENCES

[1] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 313–318. [Online]. Available: https://doi.org/10.1145/1201775.882269

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[3] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, 2003.

[4] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[5] L. Zhang, T. Wen, J. Min, J. Wang, D. Han, and J. Shi, "Learning object placement by inpainting for compositional data augmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 566–581.

[6] H. Wang, Q. Wang, H. Zhang, J. Yang, and W. Zuo, "Constrained online cut-paste for object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 4071–4083, 2020.

[7] Y. Wang, L. Tang, Y. Zhong, and B. Li, "From composited to real-world: Transformer-based natural image matting," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[8] N. Inoue and T. Yamasaki, "Learning from synthetic shadows for shadow detection and removal," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4187–4197, 2020.

[9] Y. Ren, Z. Ying, T. H. Li, and G. Li, "Lecarm: Low-light image enhancement using the camera response model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 968–981, 2018.

[10] Z. Zhao, B. Xiong, L. Wang, Q. Ou, L. Yu, and F. Kuang, "Retinexdip: A unified deep framework for low-light image enhancement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1076–1088, 2021.

[11] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3789–3797.

[12] W. Cong, J. Zhang, L. Niu, L. Liu, Z. Ling, W. Li, and L. Zhang, "Dovenet: Deep image harmonization via domain verification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8394–8403.

[13] Z. Guo, D. Guo, H. Zheng, Z. Gu, B. Zheng, and J. Dong, "Image harmonization with transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 870–14 879.

[14] J. Ling, H. Xue, L. Song, R. Xie, and X. Gu, "Region-aware adaptive instance normalization for image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9361–9370.

[15] Z. Guo, H. Zheng, Y. Jiang, Z. Gu, and B. Zheng, "Intrinsic image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 367–16 376.

[16] K. Sofiiuk, P. Popenova, and A. Konushin, "Foreground-aware semantic representations for image harmonization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1620–1629.

[17] Y. Hang, B. Xia, W. Yang, and Q. Liao, "Scs-co: Self-consistent style contrastive learning for image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 710–19 719.

[18] J.-F. Lalonde and A. A. Efros, "Using color compatibility for assessing image realism," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[19] S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier, "Understanding and improving the realism of image composites," *ACM Transactions on graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[20] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.

[21] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to color transfer," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1434–1439.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[23] W. Cong, X. Tao, L. Niu, J. Liang, X. Gao, Q. Sun, and L. Zhang, "High-resolution image harmonization via collaborative dual transformations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 470–18 479.

[24] Z. Ke, C. Sun, L. Zhu, K. Xu, and R. W. Lau, "Harmonizer: Learning to perform white-box image and video harmonization," in *European Conference on Computer Vision*. Springer, 2022, pp. 690–706.

[25] B. Xue, S. Ran, Q. Chen, R. Jia, B. Zhao, and B. Zhao, "Dccf: Deep comprehensible color filter learning framework for high-resolution image harmonization," in *European Conference on Computer Vision*, 2022.

[26] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[27] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[29] E. H. Land and J. J. McCann, "Lightness and retinex theory," *Josa*, vol. 61, no. 1, pp. 1–11, 1971.

[30] E. H. Land, "The retinex theory of color vision," *Scientific american*, vol. 237, no. 6, pp. 108–129, 1977.

[31] H. C. Karaimer and M. S. Brown, "A software platform for manipulating the camera imaging pipeline," in *European Conference on Computer Vision*. Springer, 2016, pp. 429–444.

[32] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang, "Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[34] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-scale image harmonization," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–10, 2010.

[35] X. Cun and C.-M. Pun, "Improving the harmony of the composite image by spatial-separated attention module," *IEEE Transactions on Image Processing*, vol. 29, pp. 4759–4771, 2020.

[36] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[37] J. Liang, X. Cun, C.-M. Pun, and J. Wang, "Spatial-separated curve rendering network for efficient and high-resolution image harmonization," in *European Conference on Computer Vision*. Springer, 2022, pp. 334–349.

[38] K. Wang, M. Gharbi, H. Zhang, Z. Xia, and E. Shechtman, "Semi-supervised parametric real-world image harmonization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5927–5936.

[39] J. J. A. Guerreiro, M. Nakazawa, and B. Stenger, "Pct-net: Full resolution image harmonization using pixel-wise color transformations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5917–5926.

[40] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic programming and evolvable machines*, vol. 8, no. 2, pp. 131–162, 2007.

[41] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[42] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[43] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European conference on computer vision*. Springer, 2016, pp. 628–644.

[44] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[45] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 52–67.

[46] J. Wang, H. Zhu, H. Liu, and Z. Ma, "Lossy point cloud geometry compression via end-to-end learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4909–4923, 2021.

[47] C. Benedek, B. Gálai, B. Nagy, and Z. Jankó, "Lidar-based gait analysis and activity recognition in a 4d surveillance system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 101–113, 2016.

[48] T. R. Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 882–14 891.

[49] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8628–8638.

[50] I. Anokhin, K. Demochkin, T. Khakhulin, G. Sterkin, V. Lempitsky, and D. Korzhenkov, "Image generators with conditionally-independent pixel

synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 278–14 287.

[51] I. Skorokhodov, S. Ignatyev, and M. Elhoseiny, "Adversarial generation of continuous images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 753–10 764.

[52] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 852–863, 2021.

[53] U. Fecker, M. Barkowsky, and A. Kaup, "Histogram-based prefiltering for luminance and chrominance compensation of multiview video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1258–1267, 2008.

[54] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[55] W. Cong, L. Niu, J. Zhang, J. Liang, and L. Zhang, "Bargainnet: Background-guided domain translation for image harmonization," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[56] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[57] X. Lu, S. Huang, L. Niu, W. Cong, and L. Zhang, "Deep video harmonization with color mapping consistency," *IJCAI*, 2022.

Keyan Chen received the B.S. and M.S. degrees from the School of Astronautics, Beihang University, Beijing, China, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree with the Image Processing Center, School of Astronautics, Beihang University. His research interests include image processing, machine learning, and pattern recognition.

Jianqi Chen received his B.S. degree from the Image Processing Center, School of Astronautics, Beihang University in 2021. He is currently pursuing his M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. His research interests lie in AISafety, image synthesis, and remote sensing image processing.

Yilan Zhang received her B.S. degree from the Image Processing Center, School of Astronautics, Beihang University in 2021. She is currently pursuing her M.S. degree in the Image Processing Center, School of Astronautics, Beihang University. Her research interests include deep learning, computer vision, and multimodal learning.

Zhenwei Shi (Senior Member, IEEE) is currently a Professor and Dean of the Image Processing Center, School of Astronautics, Beihang University. He has authored or co-authored over 200 scientific articles in refereed journals and proceedings, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Geoscience and Remote Sensing, the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and the IEEE International Conference on Computer Vision (ICCV). His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Prof. Shi serves as an Editor for IEEE Transactions on Geoscience and Remote Sensing, Pattern Recognition, ISPRS Journal of Photogrammetry and Remote Sensing, Infrared Physics and Technology, etc. His personal website is http://levir.buaa.edu.cn/.

Zhengxia Zou received the B.S. and Ph.D. degrees from the Image Processing Center, School of Astronautics, Beihang University, Beijing, China, in 2013 and 2018, respectively. From 2018 to 2021, he was a Post-Doctoral Research Fellow with the University of Michigan, Ann Arbor, MI, USA.

He is currently a Professor at the School of Astronautics, Beihang University. He has published more than 20 peer-reviewed papers in top-tier journals and conferences, including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Geoscience and Remote Sensing, IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), IEEE International Conference on Computer Vision (ICCV), and Association for the Advancement of Artificial Intelligence (AAAI).

His research has been featured in more than 30 global tech media outlets and adopted by multiple application platforms with over 50 million users worldwide.

His research interests include computer vision and related problems in remote sensing and autonomous driving. His personal website is https://zhengxiazou.github.io/.