Learning Kernel-Modulated Neural Representation for Efficient Light Field Compression

Jinglei Shi, Yihong Xu, Christine Guillemot Fellow

Learning Kernel-Modulated Neural Representation for Efficient Light Field Compression

Jinglei Shi, Yihong Xu, Christine Guillemot *Fellow, IEEE*

***Abstract*—Light fields capture 3D scene information by recording light rays emitted from a scene at various orientations. They offer a more immersive perception, compared with classic 2D images, but at the cost of huge data volumes. In this paper, we design a compact neural network representation for the light field compression task. In the same vein as the deep image prior, the neural network takes randomly initialized noise as input and is trained in a supervised manner in order to best reconstruct the target light field Sub-Aperture Images (SAIs). The network is composed of two types of complementary kernels: descriptive kernels (*descriptors*) that store scene description information learned during training, and modulatory kernels (*modulators*) that control the rendering of different SAIs from the queried perspectives. To further enhance compactness of the network meanwhile retain high quality of the decoded light field, we propose modulator allocation and apply kernel tensor decomposition techniques, followed by non-uniform quantization and lossless entropy coding. Extensive experiments demonstrate that our method outperforms other state-of-the-art (SOTA) methods by a significant margin in the light field compression task. Moreover, after adapting descriptors, the modulators learned from one light field can be transferred to new light fields for rendering dense views, showing the potential of the solution for view synthesis.**

***Index Terms*—light field compression, compact neural representation, modulation, kernel decomposition.**

I. Introduction

LIGHT fields [1], [2] record both the intensity and direction of light rays emitted by a scene in the 3D space. The angular information allows view synthesis and 3D scene reconstruction, which can provide users with a more immersive experience when navigating within the captured scene than classical 2D images. It also enables a number of computer vision tasks such as depth estimation [3], [4], super-resolution [5], [6], instance segmentation [7], salient object detection [8]. Although the spatio-angular information of light fields offers numerous benefits for various applications, it also introduces a significant challenge in terms of data volume. The inherent redundancy in light fields results in large storage requirements, increased transmission bandwidth, and demand on display hardware. Therefore, a crucial aspect in advancing light field imaging towards practical usage scenarios is the development of effective compression solutions.

Early compression methods [9]–[11] primarily concentrated on directly compressing the lenslet images obtained from plenoptic cameras, e.g., using HEVC-intra coding. However, these intra-coding-based approaches have shown limited capability in terms of exploiting inter-view correlation. The use of video compression standards (in particular HEVC) to compress the set of light field views (or SAIs) as a pesudo video sequence [12]–[16] thus followed naturally to better exploit the correlation between the different SAIs.

Methods specifically dedicated to light field compression have then been proposed, based on either 4D transforms or on light field view synthesis techniques used as inter-view predictors. To cite only a few, a 4D-Transform mode named Multidimensional Light field Encoder (MuLE) [17] has been adopted in the JPEG Pleno coding standard, where the 4D redundancy of light fields is exploited by applying a 4D-DCT transform to 4D spatio-angular blocks. The authors in [18] proposed a graph-transform-based light field compression method tailored by the scene geometry. The authors in [19] proposed a compression framework that relies on graph learning and dictionary learning for structural redundancies removal between SAIs. Other transforms such as mixture of expert [20], homography-based low rank approximation [21] or the shearlet transform [22] have shown their superiority for narrow-baseline data, but suffer from performance degradation when the light field baseline increases.

The methods based on view synthesis techniques [23]–[27] compress a subset of SAIs as reference views, and the decoder reconstructs the full light field from the received subset by applying view synthesis methods. A synthesis-based prediction residue is then transmitted for quality enhancement. This is the case in the WaSP codec incorporated (as the 4D-prediction mode) in the JPEG Pleno light field coding standard [27]. The view prediction or synthesis is performed using disparity-compensated prediction or warping, an overview of earlier solutions can be found in [28]. While early work was considering traditional view prediction or synthesis methods, significant advances in learning-based view synthesis [29]–[32] have also triggered the development of more effective view synthesis-based light field compression (see e.g. [25], [33]). Learning-based video compression solutions [34]–[37] have also been shown to be good candidates for light field compression.

The emergence of Implicit Neural Representations (INR) such as Neural Radiance Field (NeRF) [38] has ushered in a new era of employing neural networks to represent scenes for diverse applications [39]–[42]. They rely on the network to

This work was supported by the National Natural Science Foundation of China, Grant No.62302240, the Natural Science Foundation of Tianjin Municipality, Grant No.22JCQNJC01560, the Fundamental Research Funds for the Central Universities, Nankai Univ. Grant No.63241443 and the DeepCIM project in the context of the French ANR program on Artificial Intelligence.

Jinglei Shi (corresponding author) is with the VCIP & TMCC & DISSec, College of Computer Science, Nankai University, Tianjin, China (jinglei.shi@nankai.edu.cn). Yihong Xu is with the valeo.ai, Paris, France (yihong.xu@valeo.com). Christine Guillemot is with the INRIA (Institut National de Recherche en Informatique et en Automatique) Rennes Bretagne Atlantique, Rennes, France (christine.guillemot@inria.fr).

establish mappings between pixel coordinates and their color values in a 2D image, and between 5D coordinates (position and orientation) of light rays, and color as well as density for the volumetric rendering process of a light field. This concept has also been investigated in [43] that maps timestamps to the corresponding frames for video representation. Such INR have brought about a fresh perspective of using network weights to represent images [44], video [43] and light fields [45] for compression tasks. The authors in [45] propose to train a NeRF with low-rank constraint in an Alternating Direction Method of Multipliers (ADMM) optimization framework, followed by distillation and quantization operations, to finally obtain a compact representation of light fields. Implicit neural networks can also serve as a prior in the compression context, authors in [46] propose a two-stage workflow that uses a Gated Recurrent Unit (GRU) to encode transient information between SAIs into latent vectors, which are then processed by a generator to retrieve blocks of light field views. Let us note that INR-based methods have created a link between the problem of light field compression and the one of network compression. Methods like pruning [47]–[49], tensor rank optimization [50], [51], quantization [52], [53] that address the compactness of deep models are therefore applicable for light field compression.

In this paper, we propose a novel implicit representation of light fields taking into account specific light field data characteristics which can be summarized as follows: on one hand, all SAIs exhibit similar visual content of the scene, but on the other hand, each SAI has its unique visual content that is only observed from the corresponding perspective due to the parallax and specularity. Our proposed novel network design draws inspiration from the above visual characteristics of light fields to address the problem of compression. The network is therefore composed of shared descriptive kernels (*descriptors*) and of individual modulatory kernels (*modulators*): the descriptors are repeatedly employed when rendering different SAIs, which mimic the fact that SAIs own similar visual content. To ensure that each SAI has visual content specifically observed from the corresponding perspective, the rendering process is guided by so-called "modulators", and each SAI is reconstructed using an individual set of modulators. Though belonging to the category of INR-based approach, our method differs from existing methods [43]–[45] that learn mappings between pixel or ray coordinates or frame index and color values. Here the network can rather be seen as a learned prior, in a similar vein as the deep image prior [54], which is learned by fitting the network to the target set of SAIs, taking uniformly initialized noise as input. At the end of each training iteration, modulators of the current SAI are switched when the ground-truth SAI changes. The light field to be compressed is implicitly represented by both the descriptors and the modulators, where the descriptors account for the majority of the network parameters for storing the scene information, and the modulators control the rendering of the desired SAI.

The question which thus naturally arises is the delicate balance between model compactness (i.e. number of network parameters) and the decoding quality. To address this challenge, we propose a *modulator allocation* technique and further use *kernel tensor decomposition*. The modulator allocation mechanism effectively mitigates parameter explosion, especially in scenarios where the target light field has a high angular resolution. Additionally, the kernel tensor decomposition, as a widely-used network compression technique, decomposes both high-dimensional descriptors and modulators into the product of low-dimensional components. This decomposition strategy aims to reduce the overall parameter count while preserving the reconstruction accuracy. In our efforts towards network compactness, we also adopt a quantization-aware training strategy [52] which reduces the number of bits required for each weight and detains quantization errors.

We quantitatively and qualitatively evaluate our method and compare it with several representative state-of-the-art (SOTA) methods tailored for light field compression, including video compression-based methods such as HEVC-Lozange [55], [56], HLVC [35], RLVC [36] and VVC [57], 4D-Prediction mode of the coding standard JPEG-Pleno [58], the combination of light field view synthesis methods [29] and rate-distortion optimization framework [25], named BLLFC, as well as the most recent INR-based schemes DDLF [46] and QDLR-NeRF [45]. Experimental results show that our method outperforms others by a large margin and yields better visual reconstruction quality. Moreover, we carried out a comprehensive comparison with other two INR-based methods (DDLF and QDLR-NeRF) in terms of encoding and decoding complexity, memory consumption and generalization to different types of light fields, proving the superiority of our method in the context of compression. Besides performance gains for the task of compression, another advantage of our proposed method is that the modulators learned from one light field can be applied to new light fields for synthesizing dense views after adapting descriptors. It not only verifies the functionality of the two types of kernels, but it also shows the potential for view synthesis through kernel transfer.

To summarize, the contributions of our work are as follows:

- We propose a novel implicit representation format for light fields, which is composed of the complementary *descriptors* & *modulators* to respectively store scene information and control the rendering of different SAIs.
- Thanks to *modulator allocation* and *kernel tensor decomposition* mechanisms, the network can effectively avoid parameter explosion when light fields have high angular resolution, and reach a better balance between the model compactness and decoding quality.
- We carried out extensive experiments to show that our method outperforms other SOTA methods both quantitatively and qualitatively for the task of compression. It shows better generalization on different types of data and superior abilities in terms of complexity and resource consumption than other INR-based methods.
- We further demonstrate that the learned modulators can be transferred to new light fields, helping to generate dense views of new light fields, hence showing the potential of the method for view synthesis.

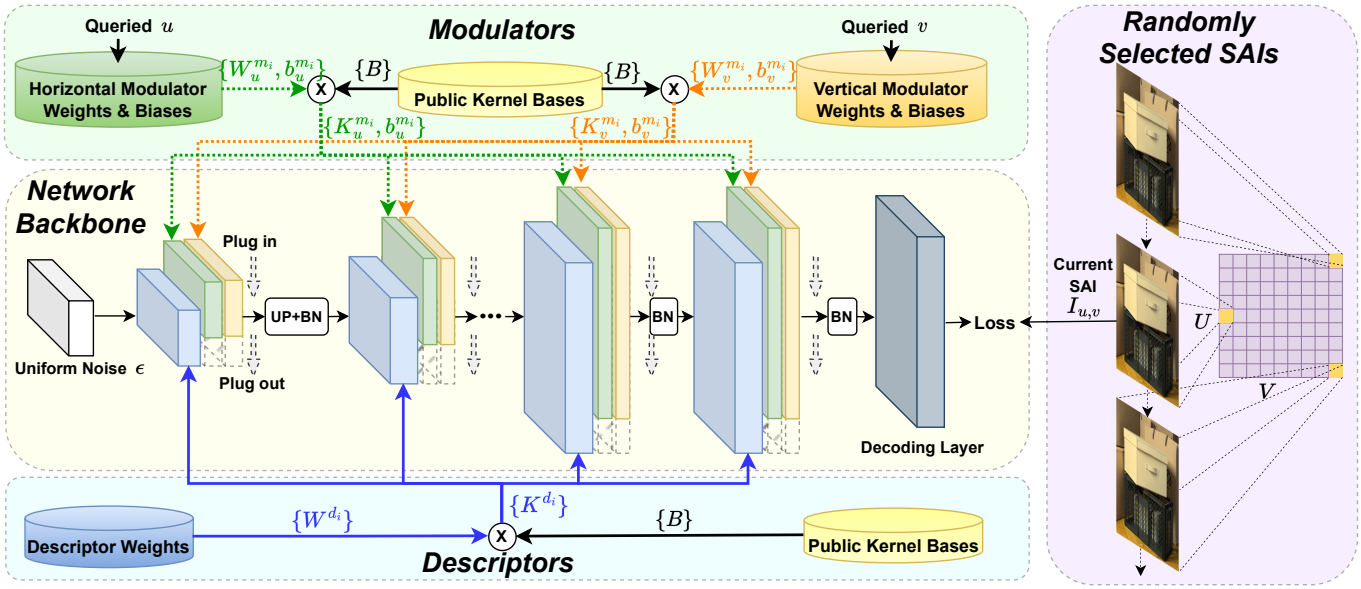


Fig. 1. Network overview: the proposed ***network*** takes uniform noise ϵ as input and its training is supervised by ***randomly selected SAIs*** of the target light field. It is composed of shared ***descriptors*** which store scene information, and switchable ***modulators*** which guide the rendering of views. To further compact the network, the modulators are allocated along the horizontal (green) and vertical (orange) directions, and both descriptors and modulators are decomposed into bases (yellow cylinder) and the corresponding weights (blue, green and orange cylinders). At the end of the training procedure, the light field is implicitly represented by the shared bases $\{B\}$, descriptor weights $\{W^{d_i}\}$ and modulator weights and biases $\{K_u^{m_i}, K_v^{m_i}, b_u^{m_i}, b_v^{m_i}\}$. A detailed illustration on how the descriptors & modulators work is shown in Fig. 2 and the network architecture is detailed in Tab. I.

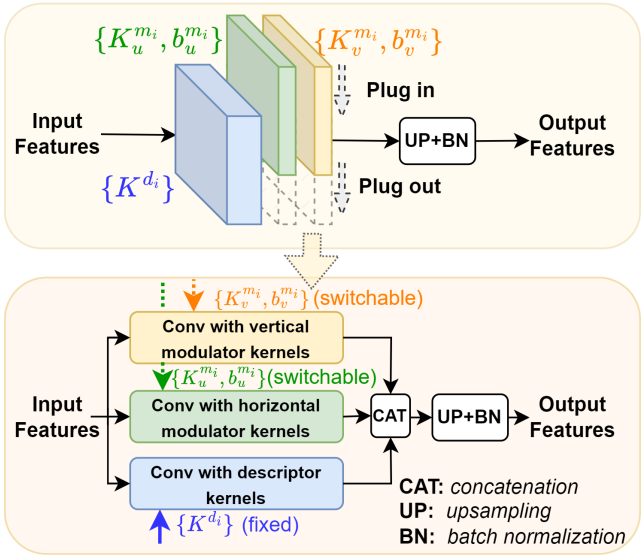


Fig. 2. Illustration on how the descriptors & modulators collaborate. The input features are respectively convolved with switchable kernels $K_u^{m_i}, K_v^{m_i}$ of vertical and horizontal modulators (with biases $b_u^{m_i}, b_v^{m_i}$) and K^{d_i} of descriptors (without biases), then concatenated (CAT) and processed by the upsampling (UP) and batch normalization (BN) operations to finally obtain the output features. The 'plug in' and 'plug out' operations, represented by gray arrows in the above figure, correspond to the switching of modulator kernels in the bottom figure.

II. Methodology

A. Notations and network overview

We represent a light field with a 4D function $L(x, y, u, v)$ following the two-parallel-plane parameterization introduced in [1], [2], where $(x, y) \in [\![1; X]\!] \times [\![1; Y]\!]$ and $(u, v) \in [\![1; U]\!] \times [\![1; V]\!]$ are respectively the spatial and angular coordinates. The SAI located at angular position (u, v) is denoted as $I_{u,v}$ throughout the rest of the paper for simplicity.

The goal of our work is to design a compact neural representation for light fields that contains a limited number of parameters while still being able to retrieve high quality views. In a previous study [59], a deep convolutional decoder was proposed to fit images for tasks such as compression, inpainting, or denoising. However, this approach was specifically designed for single RGB images, and it is not directly applicable to light fields. Although a straightforward solution would be to train the same number of deep decoders as there are SAIs in the target light field, this would be time-consuming and the resulting neural representation would require a large number of parameters, hence would not be suitable for light field compression. In [46], an advancement was made by cascading a Gated Recurrent Unit (GRU) [60] architecture and a deep decoder [59]. The GRU architecture captures transient information between SAIs, while the deep decoder captures static information within SAIs. This two-stage compression pipeline exhibited competitive performance compared to JPEG-Pleno for light fields captured using Lytro cameras. However, the introduction of the GRU module leads to issues such as an unstable training procedure, memory overflow, and degraded performance when baseline increases.

Taking into account the limitations of network designs in [46], [59], we propose an implicit convolutional network for light field representations, as depicted in Fig. 1. In this figure, the cuboids colored in blue, green, and orange represent convolutional kernels that serve distinct functionalities in each layer. Similar to [59], the network consists of sequentially connected convolutional layers. During training, the network

takes a stack of uniform noise maps ϵ as input, and the training is supervised by a randomly selected SAI denoted as $I_{u,v}$. In this approach, the image is implicitly represented by the network's parameters, as follows:

$$\Theta^* = \underset{\Theta}{arg\,min}\, E(H_\Theta(\epsilon), I_{u,v}),\ \forall I_{u,v} \in L, \tag{1}$$

where $H_\Theta(\cdot)$ represents the rendering and $\Theta = \{K^i, b^i\}$ are kernels weights and biases in each layer, i is the layer index. We use Mean Square Error (MSE) as loss function $E(\cdot)$ to supervise the training of the network.

Although both fully connected layers and convolutional layers can realize implicit neural representation, we have opted for convolutional layers as they are more flexible for different input sizes. In the same vein as the approach proposed in [54], we take uniform noise as network input, the network weights are updated by backpropagation considering a loss term which represents the data approximation or fidelity. In this manner, the trained network can be seen as a parameterization of the target light field. As the uniform noise can be generated using a pseudo random seed in practice, both encoder and decoder can pre-set the same seed value to avoid the transmission of the input noise. Based on the choices of layer type and network input, the proposed network consists of six cascaded convolutional layers with kernel size 3×3, except for the last decoding layer, which converts the channel number into 3 and has kernel size 1×1. To gradually increase the resolution of the feature maps, the first four layers are followed by bicubic upsampling (UP) with a scale factor of 2. Batch Normalization (BN) is added at the end of each layer to accelerate network convergence, except for the last layer. To expand the receptive field without increasing the number of network parameters, we set the kernel dilation factor to 2 for all intermediate layers. More details on the architecture of the network can be found in Tab. I. And a summary of main notations and corresponding definitions can be found in Tab. II.

TABLE I
PROPOSED NETWORK ARCHITECTURE. k, s, d AND *in/out* REPRESENT THE KERNEL SIZE, THE STRIDE, THE KERNEL DILATION SIZE AND THE NUMBER OF INPUT/OUTPUT CHANNELS, WHEREAS '*up2*', '*BN*', '*GELU*' AND '*Sigmoid*' REPRESENT BICUBIC UPSAMPLING BY SCALE 2, BATCH NORMALIZATION, ACTIVATION FUNCTIONS GELU AND SIGMOID. $c = c_d + c_m$ IS THE CHANNEL NUMBER OF EACH LAYER, AND c_d, c_m ARE RESPECTIVELY CHANNEL NUMBERS FOR DESCRIPTORS AND MODULATORS.

Layers	***k***	***s***	***d***	***in/out***	***input***
L_1	3	1	1	*c/c*	noise ϵ
{*up2,BN,GELU*}	-	-	-	*c/c*	L_1
L_2	3	1	2	*c/c*	{*up2,BN,GELU*}
{*up2,BN,GELU*}	-	-	-	*c/c*	L_2
L_3	3	1	2	*c/c*	{*up2,BN,GELU*}
{*up2,BN,GELU*}	-	-	-	*c/c*	L_3
L_4	3	1	2	*c/c*	{*up2,BN,GELU*}
{*up2,BN,GELU*}	-	-	-	*c/c*	L_4
L_5	3	1	1	*c/c*	{*up2,BN,GELU*}
{*BN,GELU*}	-	-	-	*c/c*	L_5
Decoder	3	1	1	*c/3*	{*up2,BN,GELU*}
Sigmoid	-	-	-	3/3	*Decoder*

B. Complementary descriptor & modulator design

As previously introduced, the combination of GRU and deep decoder in [46] enables a compact light field representation, but the utilization of GRU for modelling angular priors also suffer from limitations such as memory overflow and degraded performance for wide-baseline data. We thus follow a different design philosophy for light field representations.

SAIs of a given light field share similar scene content, while possessing distinct visual elements at the same time, due to occlusions or reflections which depend on the view direction. Therefore, the network should store the scene information that is common to all SAIs, while being controllable to render the specific visual information of each SAI. To fulfill this requirement, we define two types of kernels in the network: ***Descriptors***, which store the scene description information and constitute the majority of the network's parameters, are repeatedly used when rendering every SAI. ***Modulators***, the auxiliary view-dependent kernels indexed by angular coordinates (u, v), which modulate the rendering process and are switched from one set to another one when rendering different SAIs. As illustrated in Fig. 1, from the first to the second last layer of the network, each layer $\{K^i, b^i\}$ is composed of descriptors (colored in blue) K^{d_i} and modulators (colored in green and orange) $\{K^{m_i}_{u,v}, b^{m_i}_{u,v}\}$:

$$K^i = K^{d_i} \oplus K^{m_i}_{u,v}, \quad b^i = b^{m_i}_{u,v}, \tag{2}$$

with $\oplus$ being the concatenation operation in the last dimension. K^{d_i} and $K^{m_i}_{u,v}$ denote tensors of sizes $k \times k \times C^i_{in} \times C^{d_i}_{out}$ and $k \times k \times C^i_{in} \times C^{m_i,uv}_{out}$, respectively, where k is the kernel size, and C_{in} and C_{out} are respectively the numbers of input and output channels. Thanks to this complementary kernel design, the network does not require extra modules for explicit angular prior modelling, making the overall architecture concise and effective. Another advantage of using complementary kernel design is the computational resource reduction. The switchable kernel design makes the network generate one SAI in each forward pass, hence the memory consumption remains at a low level.

The training of such a network involves the construction of a random SAI sampling stream. Specifically, as depicted on the right side of Fig. 1, at each iteration, a random SAI is selected from $U \times V$ light field views. The selected SAI along with its corresponding angular coordinates, forms a triplet $(u, v, I_{u,v})$. Modulators $(K^{m_i}_{u,v}, b^{m_i}_{u,v})$ indexed by (u, v) are then integrated into the network to work in tandem with descriptors for rendering $\hat{I}_{u,v}$. And $I_{u,v}$ serves as the ground truth for minimizing the reconstruction error. In the subsequent iteration, a new SAI is fed into the network, and the current modulators are replaced by the next set of modulators. It is noteworthy that the scene description and rendering modulation capabilities of the descriptors and modulators are automatically acquired during the training procedure.

C. Allocation of modulator along angular directions

For INR-based methods, the compression efficiency is largely decided by the number of parameters of the network. Although our descriptors and modulators allow reducing the

TABLE II
SUMMARY OF THE MAIN NOTATIONS AND THE CORRESPONDING DEFINITIONS.

Notations	Definitions
L	the target light field
(x, y)	the spatial coordinates
(u, v)	the angular coordinates
(X, Y)	the spatial resolution
(U, V)	the angular resolution
$I_{u,v}$	the SAI at angular position (u, v)
Θ	the weights and biases of network
H	the rendering process
E	the MSE loss
K	the weight of kernel
b	the bias
i	the layer index
up2	the bicubic $2\times$ upsampling
BN	the Batch Normalization operation
GELU	the GELU activation function
K^{m_i}	the modulator kernel in i-th layer
b^{m_i}	the modulator bias in i-th layer
K^{d_i}	the descriptor kernel in i-th layer
b^{d_i}	the descriptor bias in i-th layer
C_{in}	the number of input channel
C_{out}	the number of output channel
k	the kernel size
$\hat{I}$	the rendered SAI
l	the number of layer in the network
N	the total number of parameter
B	the Fourier-Bessel bases
W	the coefficient volume
r	the number of Fourier-Bessel bases
n	the number of centroid for quantization
γ_i	the cenroids for i-th layer

number of parameters, the network may still suffer from parameter explosion. Assuming we employ an l-layer network to represent a light field, its total number of parameters N can be approximately estimated as follows:

$$N \approx lk^2 C_{in}(UVC^m_{out} + C^d_{out}), \tag{3}$$

where the number of parameters for the modulators is proportional to UV if we allocate modulators to each SAI. In the case where the target light field has high angular resolution, the number of parameters for the modulators will significantly increase and make the compression fail.

To avoid parameter explosion for high angular resolution light fields while preserving a good representation capability, instead of allocating modulators $\{K^{m_i}_{u,v}, b^{m_i}_{u,v}\}$ to each angular position of coordinates (u, v), we propose to allocate modulators along two angular directions u and v by splitting them into two subsets $\{K^{m_i}_u, b^{m_i}_u\}$ and $\{K^{m_i}_v, b^{m_i}_v\}$ as follows:

$$K^{m_i}_{u,v} = K^{m_i}_u \oplus K^{m_i}_v \tag{4}$$

$$b^{m_i}_{u,v} = b^{m_i}_u + b^{m_i}_v, \tag{5}$$

where the number of channels of $K^{m_i}_u$ and $K^{m_i}_v$ is half of that of $K^{m_i}_{u,v}$. Two subsets $\{K^{m_i}_u, b^{m_i}_u\}$ and $\{K^{m_i}_v, b^{m_i}_v\}$ are respectively represented by cuboids colored in green and orange in Fig. 1. Such allocation along orthogonal directions is based on the observation that views in the same row exhibit similar variations in the horizontal direction, while those in the same column exhibit similar variations in the vertical direction. Based on this allocation, the total number of parameters will be:

$$N \approx lk^2 C_{in}[\frac{1}{2}(U + V)C^m_{out} + C^d_{out}], \tag{6}$$

which means that the number of parameters for the modulators will be proportional to $\frac{1}{2}(U + V)$ instead of UV, implying a significant reduction of the number of parameters, particularly when dealing with high angular resolution light fields. Further discussion on the effectiveness of this allocation is given in Sec. V-B.

D. Decomposition of network kernel tensor

As mentioned earlier, the INR-based method establishes a connection between light field compression and network compression. We can also leverage network compression techniques to further enhance the network's compactness. Recall that the kernel weights $\{K^{d_i}, K^{m_i}_u, K^{m_i}_v\}$ are all four-dimension tensors, and employing suitable network compression techniques can help reducing the total number of parameters. In a related work [45], the authors applied model compression techniques to light field compression by introducing a rank-constrained NeRF [38] followed by network distillation. However, these techniques result in a complex training schedule and are primarily designed for fully-connected layers, hence not directly applicable to the proposed architecture. Inspired by the method in [51], where the authors decompose convolutional kernel tensors into a product of Fourier-Bessel (FB) bases [61], with the corresponding weights. We decompose the descriptors and modulators into the product of shared bases, denoted B (yellow cylinder) in Fig. 1, with the corresponding weights, i.e. the coefficient volumes $\{W^{d_i}, W^{m_i}_u, W^{m_i}_v\}$ (blue, green and orange cylinders in Fig. 1). Let us take descriptors K^{d_i} as an example:

$$K^{d_i} = B \otimes W^{d_i}, \tag{7}$$

where K^{d_i} is of size $k \times k \times C^i_{in} \times C^{d_i}_{out}$, B is of size $k \times k \times r$, and W^{d_i} is the coefficient volume of size $r \times C^i_{in} \times C^{d_i}_{out}$. The symbol $\otimes$ denotes the matrix multiplication and r is the number of bases in B. The authors of [51] have shown that the Fourier-Bessel (FB) bases [61] are effective bases for compressing a network for image classification and denoising. We therefore initialize B with FB bases for faster convergence, with the number of bases $r = 6$. we then update the bases during training to make them more specific to the scene being learned. Please note that a higher compression ratio can be achieved by using a smaller value of r. Using the proposed network design with complementary kernels, and by employing the techniques described above for modulator allocation and kernel tensor decomposition, a light field can be compactly represented with a set of network parameters:

$$\Theta^* = \{B, W^{d_i}, W^{m_i}_u, W^{m_i}_v, b^{m_i}_u, b^{m_i}_v\}. \tag{8}$$

E. Quantization-aware training

Besides the number of parameters required for representing a light field, the number of bits assigned to each parameter is also an important factor which impacts compression efficiency.

Although half precision (16 bits) has commonly been used in training deep learning frameworks, such a fixed-point scalar quantization with uniformly distributed centroids is still sub-optimal for the compression task. Here, we apply non-uniform quantization to each layer of the network for further network size reduction. More precisely, given a pre-defined number of centroids n for each layer (except for the last decoding layer), when working on a certain layer l_i, we perform k-means clustering of the parameters $\{W^{d_i}, W_u^{m_i}, W_v^{m_i}, b_u^{m_i}, b_v^{m_i}\}$ to obtain n centroids γ_i, and these centroids are then updated to minimize the reconstruction error as follows:

$$\gamma_i^* = \underset{\gamma_i}{arg\,min}\, E(H_\Theta(\epsilon), I_{u,v}),\ \Theta_i \in \gamma_i, \forall I_{u,v} \in L \quad (9)$$

As the quantization error accumulates throughout the network if all layers are simultaneously quantized, we adopted a strategy similar to the one in [45], [46], [53], which quantizes the network parameters layer by layer. More precisely, after quantizing the current layer, we fix the parameters of this layer with the learned codewords γ_i^*, and continue to finetune all consecutive layers. We perform 16-bit uniform quantization of the last decoding layer, as we found that non-uniformly quantizing the last layer with a small value of n brings significant quality degradation. The bases B are likewise quantized using uniform 16-bit quantization for better precision. In addition to uniform quantization with learned centroids, we also perform lossless entropy coding (Huffman coding) for further model compression. The quantized parameters of the network are transmitted from the encoder to the decoder, along with the corresponding codewords at a cost of $n \times 32$ bits, with each codeword being encoded using 32 bits.

III. Experimental Settings

A. Training details

The global learning schedule consists of two phases: the training phase and the quantization phase. Both phases utilize a learning rate of 0.01. The training phase involves 12 epochs, with each epoch defined as all SAIs being used 500 times. At each iteration, 5 SAIs are fed into the network to calculate the averaged loss. Let us note that even if a smaller learning rate and number of epochs for training might be enough for certain scenes, here we adopt a relatively large learning rate and number of epoch to ensure the convergence of the network for all scenes. In the quantization stage, we define 1 epoch as all SAIs being involved 200 times. As the quantization errors can quickly be compensated by the finetuning process, we thus perform finetuning of all consecutive layers for just 1 epoch after quantizing each layer. The whole framework is implemented in Pytorch deep learning framework and trained on a single GPU of type Nvidia Titan RTX having 24GB memory. Both encoding and decoding time will be analyzed in Sec. IV-A3.

B. Test datasets

We take four synthetic scenes *'boxes', 'sideboard', 'cotton', 'dino'* from the HCI dataset [62] and four real-world scenes *'Bikes', 'Danger', 'FountainVincent2', 'StonePillarsOutside'* from the EPFL light field dataset [63] as our basic test data. And we also tested our method with three additional challenging scenes *'Vinyl'* from [62], and *'Dinosaur'*, *'Origami'* from INRIA synthetic LF dataset [3]. These scenes are widely used by the light field research community and have distinct but representative characteristics.

The four real-world light fields are captured with a plenoptic Lytro Illum [64] camera with a narrow baseline, they have spatial resolution 432×624 and angular resolution 13×13. Due to the vignetting effect, we take the central 9×9 SAIs in our test. The introduction of micro-lens array reduces the luminance arriving at the sensor, light fields captured by Lytro Illum are generally noisy, which can be used to validate the robustness against noise for the compared methods. Both the four basic synthetic scenes and three challenging scenes are rendered using the 3D graphics software blender [65], they have spatial resolution 512×512 and angular resolution 9×9. The synthetic data mainly simulates the light fields captured by a camera array, hence they have a lower noise level and a wider baseline than those captured using a Lytro camera. Moreover, the additional scene *'Vinyl'* contains many reflective metal surfaces, and the scene *'Origami'* has $2\times$ disparity range compared with the four basic synthetic scenes, and the *'Dinosaur'* has $3\times$ the disparity range. These various scenes can comprehensively demonstrate the effectiveness of our proposed method.

C. Method configurations

We evaluate the performance of our proposed method for compression, and compare it with SOTA methods that represent recent trends in this domain, including classic video coding standard HEVC-Lozenge [55], [56] and the latest video compression standard VVC [57], learning-based video compression schemes HLVC [35] and RLVC [36], the combination of view synthesis method [29] and rate-distortion optimization framework [25], dubbed BLLFC, solutions dedicated to light field compression task such as JPEG-Pleno [58], and the most recent INR-based methods DDLF [46] and QDLR-NeRF [45]. We use official codes for all compared methods in our experiments, and each one is configured as follows:

- We use HEVC in version HM-16.10 in our test. Concerning the configuration of GOP and base QPs, we adopt a GOP of 4 as in [45], [46]. We use the software of VVC in the latest version 23.1 with its default random access slower configuration, which gives the best rate-distortion performance. We adapt QP values to make the bitrate be within the range 0-0.45 for both HEVC and VVC.
- The method BLLFC is composed of the view synthesis method FPFR [29] and rate-distortion optimization framework [25], FPFR is further finetuned on the datasets [3], [62] and [63] for better reconstruction quality.
- For two learning-based video compression schemes HLVC and RLVC, both of them have an optional hyper-parameter $\lambda = \{256, 512, 1024, 2048\}$ to control the trade-off between bitrate and distortion. The method HLVC adopts a default GOP of 10 to realize frame

TABLE III
BD-PSNR GAINS WITH RESPECT TO HEVC BASELINE. THE BEST RESULTS ARE IN BOLD, AND THE SECOND BEST RESULTS ARE UNDERLINED. *ALL INVOLVED METHODS COVER A BPP RANGE 0-0.45, EXCEPT FOR THE METHOD QDLR-NERF, WHICH COVERS A BPP RANGE 0-0.2.

Methods	boxes	sideboard	cotton	dino	Bikes	Danger	SPO	FV2	Vinyl	Origami	Dinosaur	Average
JPEG-Pleno [58]	0.46	1.77	0.99	2.74	0.20	1.38	0.58	0.30	-0.36	-2.70	-3.54	0.17
HLVC [35]	-0.78	-1.37	-3.29	-1.06	0.01	0.01	-0.15	-0.07	-5.50	-2.12	-2.57	-1.54
RLVC [36]	-0.47	-0.02	-2.71	-0.15	0.35	0.29	0.20	0.32	-4.68	-1.59	-2.01	-0.95
VVC [57]	1.85	2.24	**1.36**	1.61	1.18	1.41	1.27	1.06	2.42	0.68	**2.07**	1.56
BLLFC [25], [29]	0.42	0.72	0.64	0.73	0.62	1.10	0.80	0.64	0.65	-0.32	1.05	0.64
DDLF [46]	-1.00	-1.75	-3.14	-0.31	0.69	0.93	-0.01	0.90	-3.10	0.95	-2.20	-0.73
Q-NeRF* [45]	2.00	2.72	-0.11	2.58	0.36	1.53	0.11	0.38	0.01	2.77	1.80	1.29
Ours	**2.74**	**4.84**	0.21	**4.34**	**1.88**	**3.20**	**1.93**	**2.50**	**2.50**	**4.56**	1.99	**2.79**

prediction via three hierarchical quality layers, while for RLVC, 6 P-frames are bidirectionally encoded with a GOP of 13.

- The software version of JPEG Pleno we use is the Verification Model 2.0 in the WaSP mode, and we use disparity maps predicted by [3] in the compression process.
- For the method QDLR-NeRF, as both the tensor rank r and the number of centroids n for quantization can control the model size, we use four different ranks $r = \{40, 70, 90, 150\}$ with a fixed number of centroids $n = 256$ to have medium bitrate, then we reduce the number of centroids to $n = \{128, 64, 32\}$ with a fixed rank $r = 40$ for low bitrates.
- The method DDLF is used with parameters $(z_a, z_s) = \{(15, 30), (20, 40), (25, 50), (30, 60)\}$ and 256 centroids, where (z_a, z_s) denote respectively the number of channels of the input spatial and angular code vectors. Both handcrafted and neural-based upsamplings (i.e. pixel shuffle) are involved for having a wide range of bitrates.
- Finally, for our method, we vary the number of channels c_m and c_d in each layer for the modulators and the descriptors to achieve different bitrates. More precisely, we use $(c_m, c_d) = \{(2, 48), (2, 63), (2, 78), (2, 93), (2, 123), (2, 153), (2, 183)\}$ in our network. Although a small rank r and number n of centroids can decrease the bitrate, small values of these parameters can severely degrade the compression quality, hence we used $r = 6$ and $n = 256$ in our test.

IV. EXPERIMENTAL RESULTS

A. Compression performance analysis

1) Rate-distortion: We illustrate in Fig. 3 the rate-distortion curves, in terms of decoding quality (PSNR) and bitrate (bpp), for all compared methods. We also calculate the BD-PSNR gains using the Bjontegaard metric [66] in Tab. III to assess each method, taking the results of HEVC-Lozenge as its baseline. We can observe that the proposed method outperforms other methods on most of the scenes by a large margin, including these challenging ones.

Although DDLF [46], QDLR-NeRF [45], and our proposed method all belong to INR-based methods, they exhibit different performances due to their distinct design philosophies. Specifically: DDLF adopts a design where the transient information is modeled using a GRU module, while the static information is modeled using a deep decoder. However, the GRU architecture performs well only when the light fields have small disparity, as large parallax makes the variations between SAIs hard to capture by GRU. This explains the performance degradation of DDLF on wide-baseline synthetic light fields. QDLR-NeRF initially uses an MLP to store the scene information and then employs low-rank optimization, distillation, and quantization to reduce the model size, ultimately achieving the goal of compression. The quality of the learned scene information directly affects the compression performance. Noise and artifacts that interfere with the learning of scene information can lead to lower compression performance. This is verified by QDLR-NeRF's relatively worse performance on light fields captured using a Lytro camera. In comparison, our method stands out due to the cooperation of descriptors and switchable modulators. This feature enables the learning of each SAI to be conducted individually, and limits the impact of factors such as baseline, noise and artifacts. As a result, our method exhibits more stable and higher performance on various light fields.

2) Visual comparison: Fig. 4 shows averaged error maps across all SAIs for each method at a similar bitrate. In the visualization, red indicates a large error value, while blue represents a small error. It is evident that our proposed method outperforms other methods in terms of decoding error, particularly when dealing with highly textured scenes. Furthermore, Fig. 5 shows decoded SAIs of the scene *'sideboard'*, generated by three INR-based methods: DDLF, QDLR-NeRF, and our method. We can notice that our method successfully reconstructs clear floor texture (as seen in the zoomed regions) even at low bitrate (approximately 0.06 bpp). These error maps and decoded SAIs provide compelling evidence for the effectiveness of our method.

3) Memory consumption and encoding-decoding time: When evaluating a compression algorithm, both memory consumption and complexity play crucial roles. Lower memory consumption ensures broader hardware support, while decoding time directly impacts the delay in displaying light fields. In Fig. 6, we present the memory usage and decoding time for each learning-based method working on the GPU platform. Among these methods, DDLF [46] employs a GRU to recurrently process SAIs and decodes a block of views at once, resulting in shorter decoding time but higher memory consumption than ours. The QDLR-NeRF method [45] adopts a pixel-wise rendering mechanism, leading to a slower decoding procedure. Additionally, due to the complexity of their

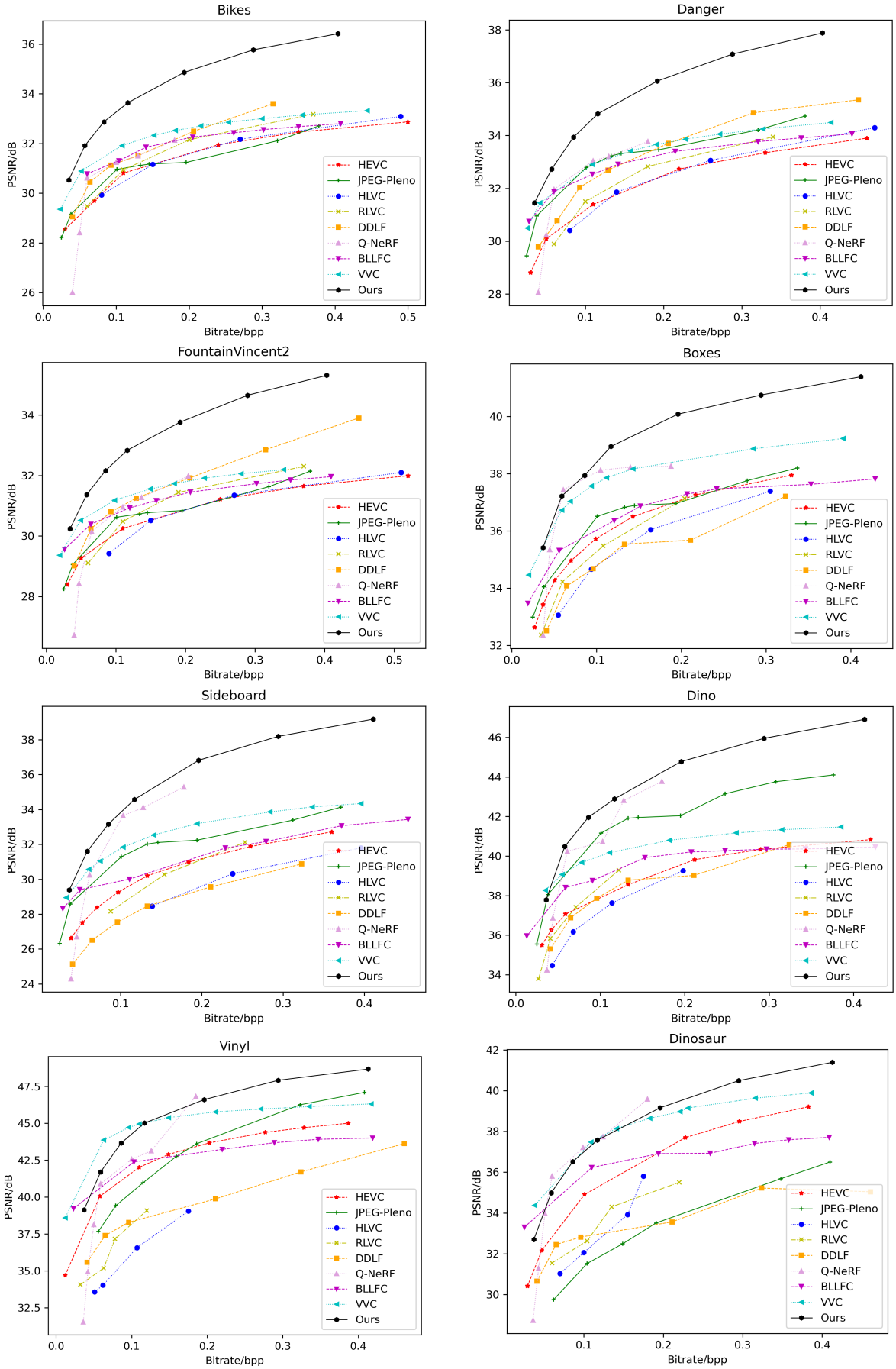


Fig. 3. Rate-distortion curves of *'Bikes'*, *'Danger'*, *'FountainVincent2'*, *'boxes'*, *'sideboard'*, *'dino'*, *'Vinyl'* and *'Dinosaur'* for HEVC-Lozenge [55], [56], JPEG-Pleno [58], HLVC [35], RLVC [36], DDLF [46], QDLR-NeRF [45], BLLFC [25], [29], VVC [57] and ours.

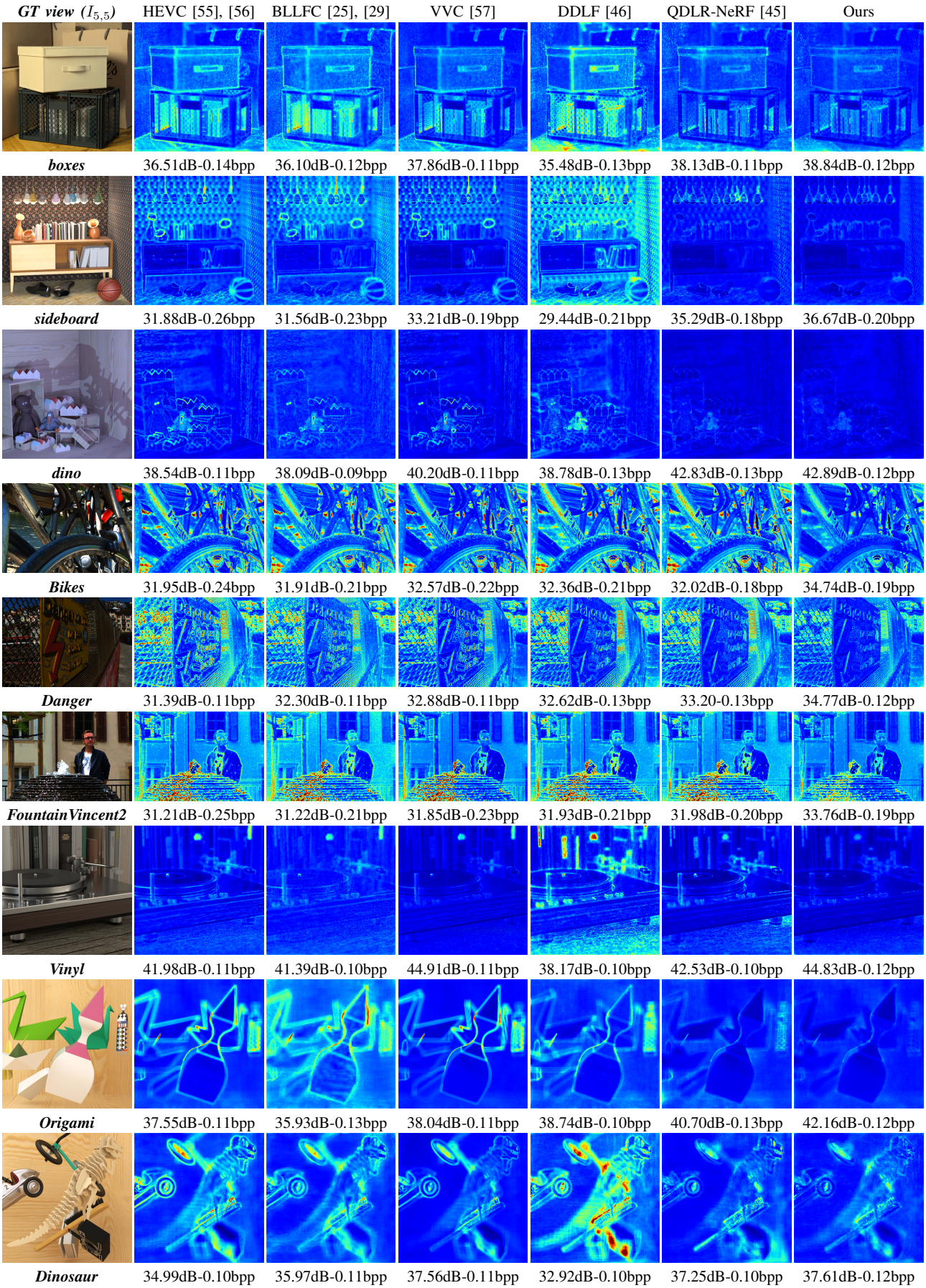
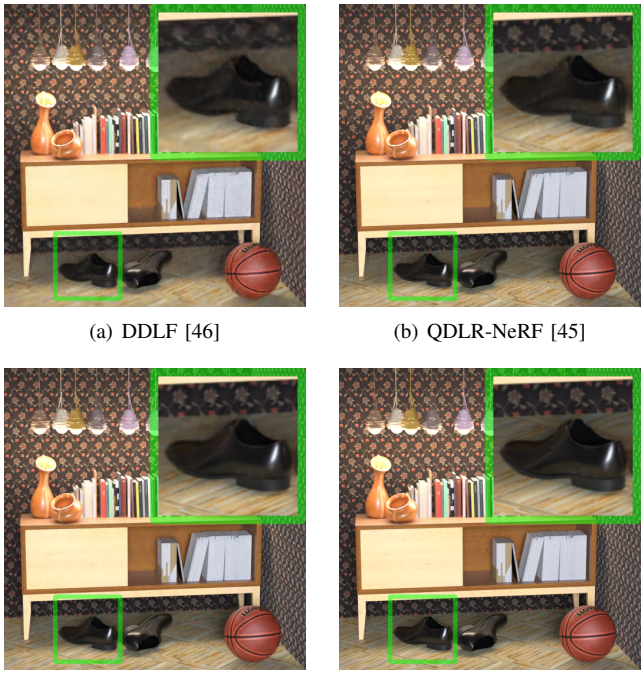


Fig. 4. Averaged error maps of decompressed light fields using different methods, along with the PSNR and bitrate values.



(a) DDLF [46] (b) QDLR-NeRF [45]

(c) Ours (d) GT View

Fig. 5. A visualization example of a decoded view using (a). DDLF [46] (26.51dB, bpp=0.065), (b). QDLR-NeRF [45] (30.25dB, bpp=0.062), (c). Our method (31.60dB, bpp=0.059), (d). Ground-truth view. A local region is zoomed for comparison.

pipelines, both the HLVC [35] and RLVC [36] methods require more memory and time for decoding each SAI. In contrary, thanks to the switchable modulator design, our network can decode SAI one by one with lower memory consumption, and the fully convolutional network also ensures a quick forward pass with less inference time. Though slightly slower than DDLF, our method presents the best trade-off between memory consumption and decoding time.

When considering encoding time, learning-based compression methods have inherent limitations compared to classical compression standards like HEVC and JPEG-Pleno: though methods HLVC [35] and RLVC [36] exhibit competitive encoding times to HEVC and JPEG-Pleno, they require a large training set, and their compression capacity heavily depends on the scale and quality of the training set used. While for INR-based methods, the encoding time mainly consists of fitting the network to the target light field. As a result, their encoding time is generally longer than that of other methods. However, in certain applications, encoding time is not as critical as decoding time, as the encoding process can be performed in parallel and offline.

To gain a better understanding of the training efficiency of our method, we provide Fig. 7 that illustrates the quality of the decoded light field in relation to the training time, and compare the encoding efficiency with other two INR-based methods. We only account for the time spent on the network initialization and exclude the time for low-rank optimization and distillation for the method QDLR-NeRF. We can see in Fig. 7 that, even without taking low-rank optimization and distillation into account, the method QDLR-NeRF needs a long training schedule. When QDLR-NeRF is optimized using the SSIM [67] or LPIP [68] metrics that require to calculate all pixels in a local image region (several forward-passes), there is a longer training schedule and more memory consumption. While both DDLF and our method can quickly reach a high performance after an encoding of 1-2 hours, our method has much higher performance than DDLF.

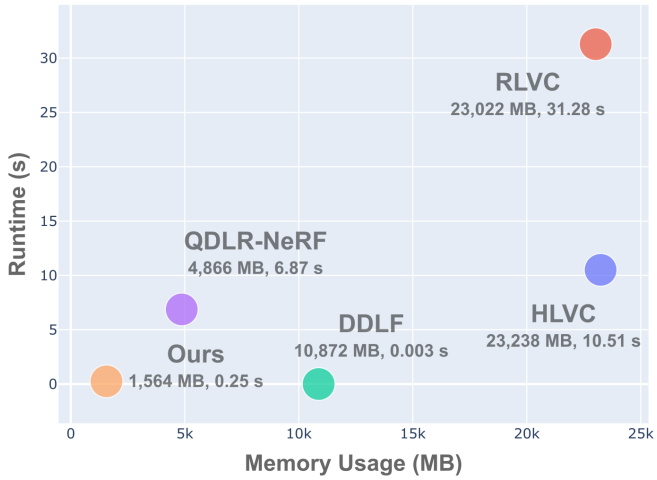


Fig. 6. GPU memory usage and time for decoding per SAI of each learning-based method, measured on Nvidia Titan RTX and the scene '*sideboard*' ($512 \times 512 \times 9 \times 9$).

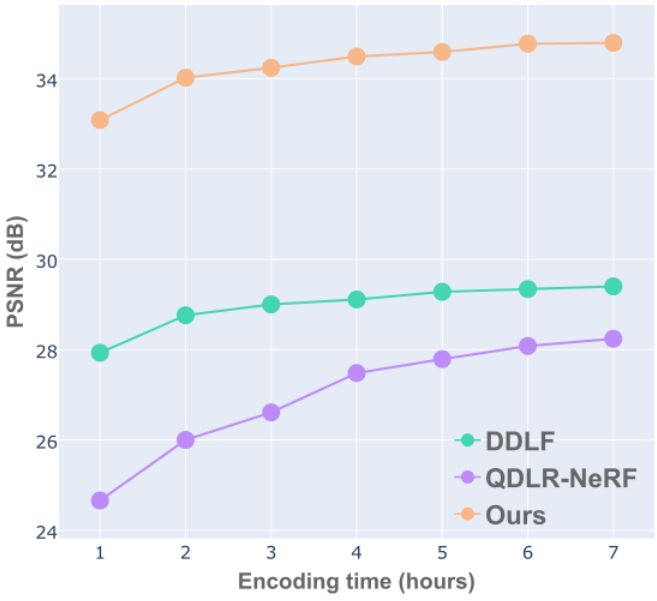


Fig. 7. Performance in terms of encoding time for each network-based method. The experiment is carried out on the scene '*sideboard*' ($512 \times 512 \times 9 \times 9$).

B. *Transfer of modulators*

We defined two types of kernels in our network design: descriptors, which store scene information, and modulators, which control the rendering of SAIs with respect to the desired perspectives. The experiment in this section demonstrates that the modulators can be non-scene-specific if the descriptors are

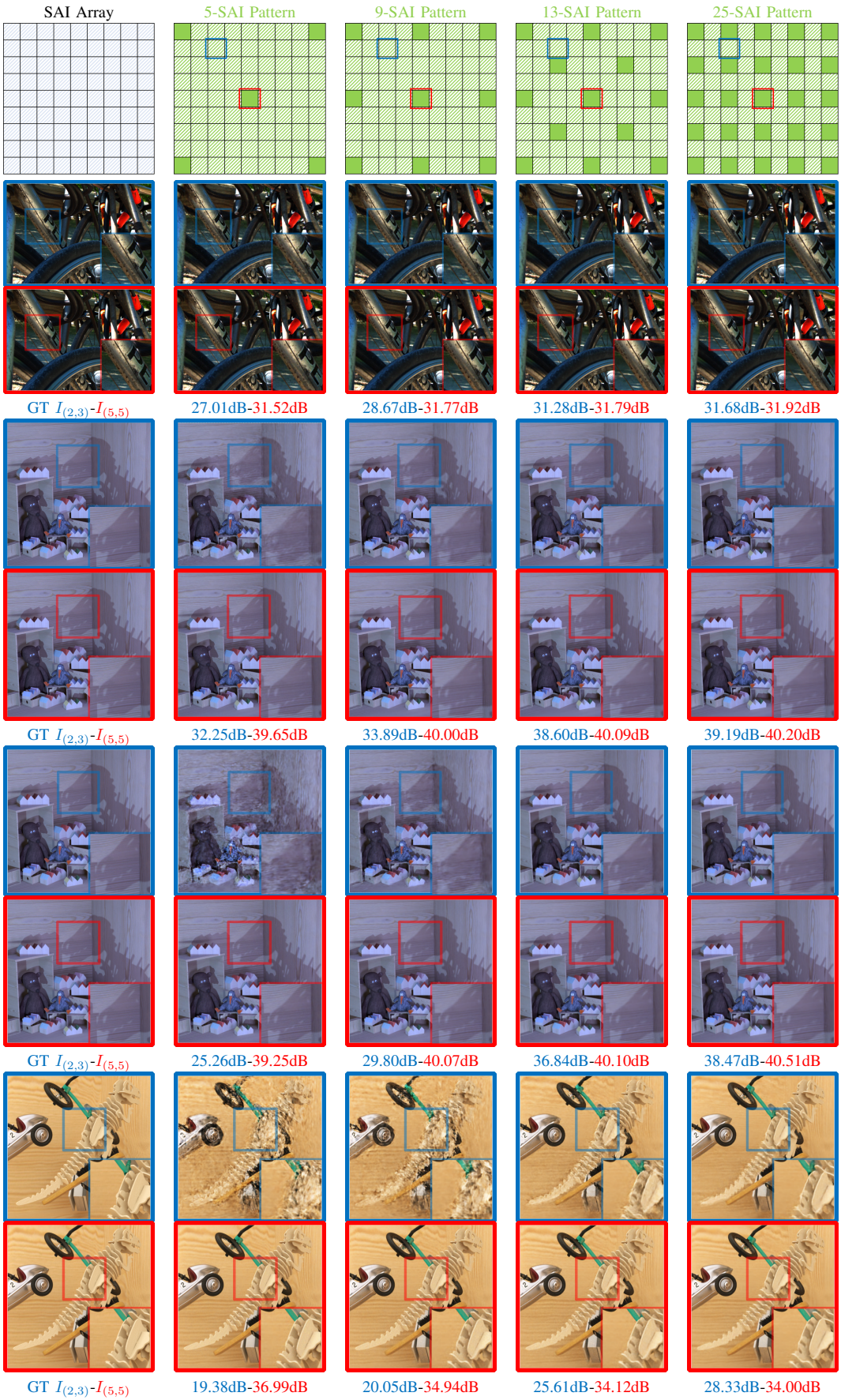


Fig. 8. Several patterns for retraining and corresponding rendered views. The 1st row shows 4 patterns with an increasing number of SAIs for retraining. We also illustrate the rendered views $\hat{I}_{(5,5)}$ and $\hat{I}_{(2,3)}$ that use the involved/uninvolved modulators in the (3rd, 5th, 7th and 9th)/(2nd, 4th, 6th and 8th) rows. The rows 2-3 show rendered views for *'bikes'*, the rows 4-5 show results for *'dino'* using modulators from *'boxes'*, the rows 6-7 show images for *'dino'* using modulators from *'Dinosaur'*, and the rows 8-9 illustrate synthesized views for *'Dinosaur'* using modulators of *'dino'*.

appropriately adapted, i.e. the modulators learned on one light field can be transferred to the new light fields. More precisely, we take two light fields L_1 and L_2 each with 9×9 SAIs as an example, and carry out the following steps:

1. ***Pretraining on*** L_1: we first train the network using all SAIs of L_1, during which both descriptors $\{K^d_{L_1}\}$ and $9+9$ sets of modulators $\{K^m_{u,L_1}, K^m_{v,L_1}, b^m_{u,L_1}, b^m_{v,L_1}\}$ are learned.
2. ***Retraining descriptors on*** L_2: we then fix the learned modulators $\{K^m_{u,L_1}, K^m_{v,L_1}, b^m_{u,L_1}, b^m_{v,L_1}\}$ and retrain descriptors using a subset of SAIs (e.g. a sparse 3×3 views) of L_2 for one epoch to obtain $\{K^d_{L_2}\}$.
3. ***Rendering all SAIs of*** L_2**:** we render all SAIs of L_2 with the updated descriptors $\{K^d_{L_2}\}$ and modulators $\{K^m_{u,L_1}, K^m_{v,L_1}, b^m_{u,L_1}, b^m_{v,L_1}\}$.

In our experiment, we specifically utilize a subset of SAIs from L_2 to retrain the descriptors for step 2, indicating that only a part of modulators is involved in this procedure. There are two main reasons for adopting sparse sampling instead of all views: (a). The SAIs inside the subset provide information about the new scene. Retraining on these views helps adapt the descriptors for storing new scene information. (b). It is important to note that the modulators for the views outside the subset are entirely excluded from the retraining process. If these modulators can successfully work with the descriptors to synthesize SAIs, it suggests that the modulators are non-scene-specific and their modulation function is transferrable. Conversely, if the excluded modulators fail to generate SAIs while those involved in the retraining procedure perform well for rendering, it would imply that the functions of the modulators and descriptors are scene-specific and are endowed only by training on the current scene. We test four cases with several patterns for selecting subsets of views:

(a). Pretraining the network on the scene *'danger'* and retraining descriptors on *'bikes'*.
(b). Pretraining the network on the scene *'boxes'* then retraining descriptors on the scene *'dino'*.
(c). Pretraining the network on the scene *'Dinosaur'* then retraining descriptors on the scene *'dino'*.
(d). Pretraining the network on the scene *'dino'* then retraining descriptors on the scene *'Dinosaur'*.

As both *'danger'* and *'bikes'* are captured using the same Lytro camera, while *'boxes'* and *'dino'* are synthesized using different camera array configurations, the first two cases respectively represent the transfer of modulators between cameras with the same and distinct configurations. The scene *'Dinosaur'* has about $3\times$ the disparity range than the scene *'dino'* and the same image size 512×512, the cases (c) and (d) respectively represent the transfer from a sparse light field to a dense one, and the inverse. Fig. 8 showcases the patterns of the subsets of views and the rendered SAIs. The first row depicts the patterns with an increasing number of SAIs used for retraining (**retraining patterns**), where the views inside the subset are colored in green, and other excluded views are noted with green slashes. The squares framed with red and blue boxes indicate the positions of the SAIs shown from the second to the ninth rows, they respectively represent SAIs rendered using modulators involved and not involved (noted as '**involved modulator**' and '**uninvolved modulator**') in the retraining procedure. Rows 2-3 show generated SAIs of *'bikes'* transferred from *'danger'*, rows 4-5 show rendered SAIs of the scene *'dino'* transferred from *'boxes'*, rows 6-7 illustrate views of the scene *'dino'* transferred from *'Dinosaur'* and row 8-9 demonstrate views of the scene *'Dinosaur'* using modulators of the scene *'dino'*.

We observe that both the involved and uninvolved modulators can work with the descriptors to generate SAIs of the new light fields, even if this transfer occurs between cameras with different configurations like shown in case (b). The transfers between light fields having different disparity ranges in both cases (c) and (d) are also achievable, but they exhibit varying levels of difficulty: as sparse light fields have more complex occlusions and missing information, if we compare the results in the 4th (case (b)), the 6th (case (c)) and the 8th (case (d)) rows in Fig. 8, 5-SAI pattern can already generate high quality views for transfer between light fields having similar disparity range in case (b), but it necessitates 9-SAI pattern to produce good views for transfers from sparse light fields to dense ones in case (c), while at least 13-SAI pattern is required for synthesizing views of decent quality when transferring kernels from dense light fields to sparse ones in case (d). If we compare the results in the 8th and the 9th rows in case (d), adding SAIs in the retraining patterns leads to increasing quality of rendered views $\hat{I}_{2,3}$ as more scene information is provided to descriptors, but decreasing quality of views $\hat{I}_{5,5}$, this is because modulators from dense light fields are naturally less suitable for sparse ones, more modulators in the retraining process will degrade the adaptation of descriptors. Globally, more SAIs in the subset improves the quality of views rendered with uninvolved modulators. And SAIs generated using involved modulators show better quality than those generated using uninvolved modulators, as modulators involved in the retraining step always better match descriptors than those uninvolved ones. Let us note that such a modulator transfer operation also implies a new solution for view synthesis task, one can generate novel dense views by transferring the learned modulators to the target light field.

V. Ablation Study

A. Proportion of modulator parameters

When adopting our network architecture for light field compression, the proportion of modulator parameters plays a key role in the compression performance. To explore the optimal proportion of modulators for network design, we conducted experiments by varying the proportions of modulator parameters under a fixed total parameter constraint. Tab. IV gives the average PSNR and quality variance across 9×9 views for 8 different scenes, where c_m and c_d respectively denote the numbers of channels for the modulators and descriptors. Fig. 9 shows the averaged PSNR on 8 scenes for different viewpoints.

From both Fig. 9 and Tab. IV, we can observe that the proportion of modulator parameters directly affects the network's performance. For a similar total number of parameters,

a higher proportion of modulator parameters implies a lower proportion for the descriptors. This results in a relatively lower averaged PSNR and smaller quality variance among views. This occurs because the network has a limited number of parameters for storing scene information, but enough parameters to modulate the rendering of SAIs. Instead, reducing the proportion of modulator parameters will spare more parameters for descriptors, which improves the quality of the decoded views, but weakens the network's modulation capability and leads to a larger quality variance. The above observation can serve as a guideline for network design for different compression demands: when aiming for a high-quality representation of the entire light field, it is preferable to use a lower proportion of modulator parameters. However, if maintaining consistency between SAIs is a priority, a higher proportion is recommended.

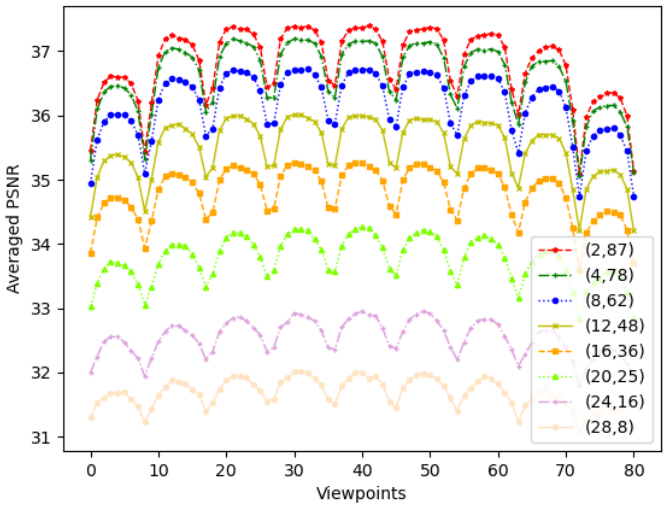


Fig. 9. Averaged PSNR for the different viewpoints, with the index of the top left corner view being labeled as '0' and the index of the bottom right corner view as '80'.

TABLE IV
AVERAGED PSNR AND VARIANCE IN TERMS OF DIFFERENT PROPORTIONS OF MODULATOR PARAMETERS, CALCULATED OVER 81 VIEWS OF 8 SCENES UNDER A FIXED TOTAL PARAMETER NUMBER CONSTRAINT.

(c_m, c_d)	Proportion	PSNR(dB)	Variance
(28,8)	97%	31.67	0.085
(24,16)	93%	32.54	0.102
(20,25)	88%	33.77	0.147
(16,36)	80%	34.80	0.186
(12,48)	70%	35.51	0.228
(8,62)	55%	36.18	0.275
(4,78)	33%	36.62	0.312
(2,87)	19%	36.81	0.345

B. Effectiveness of kernel design

To validate the modulator allocation and kernel tensor decomposition, under the constraint of similar total number of network parameters, we tested three network variants:

(a). The network without both modulator allocation and kernel tensor decomposition designs, which means that the network is composed of normal convolutional kernels, and we allocate angular kernels to each SAI per network layer, which is denoted as *Net*†.

(b). The network without modulator allocation but with kernel tensor decomposition, which is denoted as *Net**.

(c). The network that adopts both modulator allocation and kernel tensor decomposition designs, this variant is denoted as *Net*.

TABLE V
PERFORMANCE COMPARISON BETWEEN DIFFERENT NETWORK VARIANTS FOR LOW, MODERATE AND HIGH NUMBERS OF PARAMETERS. THE BEST PERFORMANCES ARE IN BOLD.

	#Param	103K	325K	809K
Net†	(c_m, c_d)	(2,11)	(2,34)	(2,72)
	PSNR (dB)	27.43	33.21	37.16
*Net**	(c_m, c_d)	(2,16)	(2,47)	(2,97)
	PSNR (dB)	28.78	34.16	37.92
Net	(c_m, c_d)	(2,48)	(2,93)	(2,153)
	PSNR (dB)	**33.95**	**37.47**	**39.98**

We measure the averaged PSNR of the networks having small, moderate, and large parameter numbers, corresponding to low, intermediate, and high bitrates in the context of compression. Tab. V summarizes the performance of each network variant for different numbers of parameters. The application of modulator allocation results in significant parameter savings that can be allocated to the descriptors for performance enhancement. And the adoption of tensor decomposition enables the reduction of the number of parameters in kernels, thereby accommodating more kernels in the network. The combination of both modulator allocation and kernel tensor decomposition results in a notable improvement of the network's performance.

C. Contributions of each network design option

To highlight the contribution of each network design step, we evaluated the performance evolution after implementing each design step (including modulator allocation, kernel tensor decomposition, and quantization). Tab. VI, gives the average PSNR results for eight tested scenes and the corresponding network size when applying each design step. For comparison, we consider the original network configuration without modulator allocation, tensor decomposition, and quantization as the baseline, It has the numbers of channels $(c_m, c_d) = (2, 48)$. Due to the high angular resolution $(U, V) = (9, 9)$, without adopting modulator allocation, even if $c_m = 2$ is much smaller than $c_d = 48$, the network still has a large proportion of parameters allocated to modulators. Therefore we can observe about 20% size reduction when applying the modulator allocation technique with only 0.15dB performance degradation. Around 0.6dB loss is caused by the tensor decomposition technique, please note that tensor decomposition is a typical network compression method, other advanced decomposition method is likewise applicable to our method. Finally, the quantization operation brings 0.8dB degradation after compacting the network size from 20.93% to 9.86%. These techniques globally realize more than $10\times$ compression with about 1.6dB quality degradation.

TABLE VI
PSNR AVERAGED ON THE 8 TEST LIGHT FIELDS AT THE DIFFERENT STEPS OF THE PROPOSED WORKFLOW: ORIGINAL NETWORK (NET-ORG), NETWORK AFTER USING MODULATOR ALLOCATION (NET-MA), AFTER USING TENSOR DECOMPOSITION (NET-TD), AFTER 8-BIT QUANTIZATION (NET-8BIT)), AND AFTER APPLYING HUFFMAN CODING (NET-HUFF)

Metrics	Net-org	Net-ma	Net-td	Net-8bit	Net-Huff
PSNR	34.77dB	34.62dB	33.95dB	33.13dB	33.13dB
Parameters	246K	153K	103K	103K	103K
Bits/parameter	32	16	16	8	7.4
Size	100%	31.10%	20.93%	10.67%	9.86%

VI. LIMITATIONS AND DISCUSSION

While the proposed light field compression method shows its superiority compared to representative methods in terms of rate-distortion performance, there are a few limitations that could be addressed in the future:

1). Our method demonstrates promising performance on various light fields, including those captured with Lytro cameras and camera arrays, and these light fields contain complex textures, even in the case of reflective surfaces and different disparity ranges. However, the performance of the proposed method is below the one of state of the art video compression methods for very sparse light field, due to the fact that our method is based on the assumption that all SAIs share similar visual content, which is less verified for sparse light fields.

2). The transfer of modulators between light fields having similar disparity works well. When the source and target light fields have different disparity ranges, the method will require more SAIs in the retraining subset for good synthesis quality, especially the transfer from dense source light fields to sparse target ones, as the latter have large occlusions.

VII. CONCLUSION

In this paper, we address the challenge of light field compression by proposing a novel compact neural representation. Our method utilizes two types of complementary kernels: descriptors and modulators. Descriptors capture scene information, while modulators are used to modulate the rendering of different SAIs. To enhance the network's compactness, we propose allocating modulators across two angular dimensions and we further decompose the kernel tensor into low-dimensional components. Through extensive experiments, we demonstrate that our network-based representation outperforms other compression methods while consuming less computational resources. Furthermore, we highlight that the modulators exhibit a non-scene-specific nature and can be transferred to new light field data for rendering dense views. This finding suggests a new approach to view synthesis methods, introducing a distinct philosophy in this field.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Kai Gu and Dr. Yiqun Liu for technical support. The first author also sincerely thanks Ms. Yutong Gao for her encouragement and companionship.

REFERENCES

[1] M. Levoy and P. Hanrahan. Light field rendering. In *Annu. Conf. Comput. Graph. Interact. Techn.*, pages 31–42, 1996.

[2] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *ACM Trans. Graph. (TOG)*, pages 43–54, 1996.

[3] J. Shi, X. Jiang, and C. Guillemot. A framework for learning depth from a flexible subset of dense and sparse light field views. *IEEE Trans. Image Process. (TIP)*, 28(12):5867–5880, Dec 2019.

[4] Y. Wang, L. Wang, Z. Liang, J. Yang, W. An, and Y. Guo. Occlusion-aware cost constructor for light field depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 19809–19818, 2022.

[5] S. Wang, T. Zhou, Y. Lu, and H. Di. Detail-preserving transformer for light field image super-resolution. In *AAAI Conf. Artif. Intell. (AAAI)*, volume 36, pages 2522–2530, 2022.

[6] Y. Wang, L. Wang, G. Wu, J. Yang, W. An, J. Yu, and Y. Guo. Disentangling light fields for super-resolution and disparity estimation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2022.

[7] Y. Xu, H. Nagahara, A. Shimada, and R. Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 3442–3450, 2015.

[8] D. Jing, S. Zhang, R. Cong, and Y. Lin. Occlusion-aware bi-directional guided network for light field salient object detection. In *ACM Int. Conf. Multimedia (MM)*, pages 1692–1701, 2021.

[9] L. Lucas, C. Conti, P. Nunes, L. D. Soares, N. Rodrigues, C. L. Pagliari, E. Da Silva, and S. De Faria. Locally linear embedding-based prediction for 3d holoscopic image coding using HEVC. In *Eur. Sign. Process. Conf. (EUSIPCO)*, pages 11–15, 2014.

[10] C. Conti, P. Nunes, and L. Soares. HEVC-based light field image coding with bi-predicted self-similarity compensation. In *IEEE Int. Conf. Multimedia and Expo. Worksh. (ICMEW)*, pages 1–4, 2016.

[11] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. Da Silva, and L. Soares. Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction. In *IEEE Int. Conf. Multimedia and Expo. Worksh. (ICMEW)*, pages 1–4, 2016.

[12] F. Dai, J. Zhang, Y. Ma, and Y. Zhang. Lenselet image compression scheme based on subaperture images streaming. In *IEEE Int. Conf. Image Process. (ICIP)*, pages 4733–4737, 2015.

[13] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng. Pseudo-sequence-based light field image compression. In *IEEE Int. Conf. Multimedia and Expo. Worksh. (ICMEW)*, pages 1–4, 2016.

[14] W. Ahmad, R. Olsson, and M. Sjostrom. Interpreting plenoptic images as multiview sequences for improved compression. In *IEEE Int. Conf. Image Process. (ICIP)*, 2017.

[15] I. Viola, M. Řeřábek, and T. Ebrahimi. Comparison and evaluation of light field image coding approaches. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 11(7):1092–1106, 2017.

[16] L. Li, Z. Li, B. Li, D. Liu, and H. Li. Pseudo-sequence-based 2-D hierarchical coding structure for light-field image compression. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 11(7):1107–1119, 2017.

[17] MB. de Carvalho, MP. Pereira, G. Alves, E. da Silva, C. Pagliari, F. Pereira, and V. Testoni. A 4D DCT-based lenslet light field codec. In *ICIP*, pages 435–439, 2018.

[18] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot. Geometry-aware graph transforms for light field compact representation. *IEEE Trans. Image Process. (TIP)*, 29:602–616, 2020.

[19] Y. Zhang, W. Dai, Y. Li, C. Li, J. Hou, J. Zou, , and H. Xiong. Light field compression with graph learning and dictionary-guided sparse coding. *IEEE Trans. Multimedia (TMM)*, 25, 2023.

[20] R. Verhack, T. Sikora, G. Van Wallendael, and P. Lambert. Steered mixture-of-experts for light field images and video: Representation and coding. *IEEE Trans. Multimedia (TMM)*, 22(3):579–593, 2020.

[21] X. Jiang, M. Le Pendu, R. Farrugia, and C. Guillemot. Light field compression with homography-based low-rank approximation. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 11(7):1132–1145, Oct. 2017.

[22] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson. Shearlet transform-based light field compression under low bitrates. *IEEE Trans. Image Process. (TIP)*, 29:4269–4280, 2020.

[23] J. Chen, J. Hou, and L. P. Chau. Light field compression with disparity-guided sparse coding based on structural key views. *IEEE Trans. Image Process. (TIP)*, 27:314–324, Dec. 2016.

[24] X. Huang, P. An, Y. Chen, D. Liu, and L. Shen. Low bitrate light field compression with geometry and content consistency. *IEEE Trans. Multimedia (TMM)*, 24:152–165, 2020.

[25] J. Hou, J. Chen, and L.P. Chau. Light field image compression based on bi-level view compensation with rate-distortion optimization. *IEEE Trans. Circuit Syst. Video Technol. (TCSVT)*, 29:517–530, 2019.

[26] S. Zhao and Z. Chen. Light field image coding via linear approximation prior. In *IEEE Int. Conf. Image Process. (ICIP)*, 2017.

[27] A. Pekka and T. Ioan. WaSP: Hierarchical warping, merging, and sparse prediction for light field image compression. *Eur. Worksh. on Visual Inform. Process. (EUVIP)*, pages 1–6, 2018.

[28] C. Conti, L. D. Soares, and P. Nunes. Dense light field coding: A survey. *IEEE Access*, 8:49244–49284, 2020.

[29] J. Shi, X. Jiang, and C. Guillemot. Learning fused pixel and feature-based view reconstructions for light fields. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2555–2564, 2020.

[30] J. Jin, J. Hou, J. Chen, H. Zeng, S. Kwong, and J. Yu. Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 44(04):1819–1836, 2022.

[31] J. Shi, X. Jiang, and C. Guillemot. Deep residual architecture using pixel and feature cues for view synthesis and temporal interpolation. *IEEE Trans. on Comput. Imaging (TCI)*, 8:246–259, 2022.

[32] M. Guo, J. Hou, J. Jin, H. Liu, H. Zeng, and J. Lu. Content-aware warping for view synthesis. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 45(08):9486–9503, Aug. 2023.

[33] H. Amirpour, C. Guillemot, and C. Timmerer. FuRA: Fully random access light field image compression. In *Eur. Workshop on Vis. Info. Process. (EUVIP)*, pages 1–6, Sept. 2022.

[34] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. Dvc: An end-to-end deep video compression framework. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 11006–11015, 2019.

[35] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[36] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE J. Sel. Topics Signal Process. (JSTSP)*, 15(2):388–401, 2021.

[37] R. Yang, L. Van Gool, and R. Timofte. Perceptual learned video compression with recurrent conditional GAN. *Int. Joint Conf. Artif. Intell. (IJCAI)*, 2022.

[38] 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. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[39] M. Bemana, K. Myszkowski, H. Seidel, and T. Ritschel. X-fields: Implicit neural view-, light-and time-image interpolation. *ACM Trans. Graph. (TOG)*, 39(6):1–15, 2020.

[40] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. Srinivasan P, and J. Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 16190–16199, 2022.

[41] X. Wang, R. Courant, J. Shi, E. Marchand, and M. Christie. JAWS: Just a wild shot for cinematic transfer in neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 16933–16942, 2023.

[42] J. Qiu, P. Jiang, Y. Zhu, Z. Yin, M. Cheng, and B. Ren. Looking through the glass: Neural surface reconstruction against high specular reflections. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 20823–20833, 2023.

[43] H. Chen, B. He, H. Wang, Y. Ren, S.N. Lim, and A. Shrivastava. NeRV: Neural representations for videos. In *Adv. Neural Inform. Process. Syst. (NIPS)*, volume 34, pages 21557–21568, 2021.

[44] Y. Strumpler, J. Postels, R. Yang, L. Van Gool, and F. Tombari. Implicit neural representations for image compression. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 74–91, Sept. 2022.

[45] J. Shi and C. Guillemot. Light field compression via compact neural scene representation. In *IEEE Int. Conf. Acoust., Speech, and Sign. Process. (ICASSP)*, pages 1–5. IEEE, 2023.

[46] X. Jiang, J. Shi, and C. Guillemot. An untrained neural network prior for light field compression. *IEEE Trans. Image Process. (TIP)*, 31:6922–6936, 2022.

[47] T. Zhang, S. Ye, K. Zhang, J. Tang, W. Wen, M. Fardad, and Y. Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 184–199, 2018.

[48] S. Gao, F. Huang, W. Cai, and H. Huang. Network pruning via performance maximization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9270–9280, 2021.

[49] Z. Wang, C. Li, and X. Wang. Convolutional neural network pruning with structural redundancy reduction. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 14913–14922, 2021.

[50] M. Yin, Y. Sui, S. Liao, and B. Yuan. Towards efficient tensor decomposition-based DNN model compression with optimization framework. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10674–10683, 2021.

[51] Q. Qiu, X. Cheng, G. Sapiro, et al. DCFNet: Deep neural network with decomposed convolutional filters. In *Int. Conf. on Mach. Learn. (ICML)*, pages 4198–4207, 2018.

[52] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2704–2713, 2018.

[53] A. Fan, P. Stock, B. Graham, E. Grave, R. Gribonval, H. Jegou, and A. Joulin. Training with quantization noise for extreme model compression. *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[54] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9446–9454, 2018.

[55] ISO/IEC JTC 1/SC29. High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 2: High Efficiency Video Coding. ISO/IEC 23008-2:2017. Technical report, 2017.

[56] M. Rizkallah, T. Maugey, C. Yaacoub, and C. Guillemot. Impact of light field compression on focus stack and extended focus images. In *IEEE Eur. Signal Process. Conf. (EUSIPCO)*, pages 898–902, 2016.

[57] W. Adam, B. Jens, H. Tobias, B. Christian, and al. VVenC: An open and optimized vvc encoder implementation. In *IEEE Int. Conf. Multimedia and Expo. Worksh. (ICMEW)*, pages 1–2.

[58] Jpeg pleno. https://jpeg.org/jpegpleno/.

[59] H. Reinhard and H. Paul. Deep decoder: Concise image representations from untrained non-convolutional networks. *Int. Conf. Learn. Represent. (ICLR)*, 2019.

[60] K. Cho, Van B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[61] M. Abramowitz and I. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. 1964.

[62] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conf. Comput. Vis. (ACCV)*, pages 19–34, 2016.

[63] M. Rerabek and T. Ebrahimi. New light field image dataset. In *Int. Conf. on Quality of Multimedia Experience (QoMEX)*, number EPFL-CONF-218363, 2016.

[64] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan, et al. Light field photography with a hand-held plenoptic camera. *Comput. Science Tech. Report (CSTR)*, 2(11):1–11, 2005.

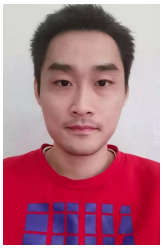
[65] Blender website. https://www.blender.org/.

[66] B. Gisle. Calculation of average psnr differences between rd-curves. In *ITU-T SG16 Q.6 Document, VCEG-M33*, 2001.

[67] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process. (TIP)*, 13(4):600–612, 2004.

[68] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 586–595, 2018.

Jinglei Shi is currently a lecturer in Nankai University, Tianjin, China. He received Bachelor's degree in Electronic Information Engineering from UESTC (University of Electronic Science and Technology of China) in 2015. Then he received Engineer's degree and Master's degree in Image Processing from IMT (Institut Mines-Telecom) Atlantique, France, in 2017, and PhD degree from University Rennes 1, France, in 2021. Then he worked as a post-doctoral researcher at INRIA (Institut National de Recherche en Informatique et en Automatique) in France. His research interests concern learning-based light field depth estimation, view synthesis, video interpolation and compression.



Yihong Xu is currently a research scientist at Valeo.ai in Paris. He obtained his Ph.D. from the RobotLearn team at Inria Grenoble Rhône-Alpes in June 2022. Previously, he received his master's degree from IMT Atlantique (previously Télécom Bretagne). His research interests lie in multiple-object tracking, domain adaptation as well as autonomous driving applications based on machine learning.

Christine Guillemot IEEE Fellow, is Director of Research at INRIA. She holds a Ph.D. degree from ENST (Ecole Nationale Superieure des Telecommunications) Paris, and a Habilitation for Research Direction from the University of Rennes. From 1985 to Oct. 1997, she has been with FRANCE TELECOM, where she has been involved in various projects in the areas of image and video coding and processing for TV, HDTV and multimedia. From Jan. 1990 to mid 1991, she has worked at Bellcore, NJ, USA, as a visiting scientist. Her research interests include: signal and image processing, and computer vision. She has served as Associate Editor for IEEE Trans. on Image Processing (from 2000 to 2003, as well as 2014-2016), for IEEE Trans. on Circuits and Systems for Video Technology (from 2004 to 2006), and for IEEE Trans. on Signal Processing (2007-2009). She has served as senior member of the editorial board of the IEEE journal on selected topics in signal processing (2013-2015) and has been senior area editor of IEEE Trans. on Image Processing (2016-2020).