HoloZip: High Hologram Compression via Latent-of-Latent Coding

Huaizhi Qu*, Yujie Wang*, Ruichen Zhang, Hengyu Lian, Mufan Qiu, Samarjit Chakraborty, Henry Fuchs, Tianlong Chen†, Praneeth Chakravarthula

Abstract—Holographic displays are gaining increasing popularity, particularly as a holy grail solution to augmented and virtual reality (AR/VR) wearable displays. However, the generation of holograms is computationally intensive for AR/VR edge devices, which are expected to be compact and lightweight with low power consumption and heat dissipation to last all day. To address this, we propose a distributed hologram generation framework, dubbed `HoloZip`, to jointly generate, compress, transmit, and decode computer-generated holograms across the cloud-edge devices. Specifically, we divide the compute-intensive hologram generation between the cloud and the local edge device by performing most of the hologram image generation via a vision transformer based backbone on the cloud, compress and transmit, then decode on a local edge device by a lightweight model. Our `HoloZip` framework supports both both 2D/3D holograms as well as holographic videos, achieving a reconstruction quality of over 30 dB in PSNR (lossy compression standard) with bit rates as low as 0.6 bits per pixel as validated by experiments.

Index Terms—Computational Display, Holography, Near-Eye Display, AR/VR, Compression

1 Introduction

Holographic displays are regarded as a promising solution for next-generation VR/AR systems due to their ability to provide depth cues, compensate for optical aberrations, and support compact form factors [1], [2]. Holography records and reconstructs the light field of 3D scenes by leveraging light interference and diffraction. Specifically, the light field of a 3D scene is encoded into an interference pattern, *i.e.*, hologram, which can be replayed by illuminating the hologram properly. Recent advances have demonstrated compact eyeglass-style displays [3], [4], [5], further advancing the practicality of holographic display technologies. To remain compact and lightweight, such glasses are typically constrained by limited onboard computing power and battery capacity. This motivates cloud-assisted holographic display systems, where computationally intensive hologram generation is offloaded to cloud servers, and the synthesized hologram data is transmitted to lightweight edge device clients. In this context, cloud-collaborative computation and hologram compression are essential for efficient data transmission and improved system practicality [6].

Computer-generated holography (CGH) encodes the light field of digital 3D assets into holograms via numerically simulating wave propagation. Traditional CGH methods span a wide range of approaches, including ray tracing [7], polygon- or layer-based rendering [8], [9], and iterative image-based algorithms such as the Gerchberg–Saxton (GS) method [10] or optimization-based solvers [11], [12], [13]. However, these methods generally suffer from high computational costs and slow runtimes. Recent works propose neural network-based models that achieve real-time, high-quality hologram synthesis [11], [14], [15], [16]. However, due to the high resolution and large receptive field requirements inherent in hologram generation, these methods can still pose challenges for edge devices with limited computing and battery life.

Additionally, holograms significantly differ from natural images in their statistical properties, making holographic image reconstructions highly sensitive to distortions in hologram data. As a result, conventional lossy codecs designed for natural images or videos are often less effective for hologram compression. Furthermore, unlike natural images, whose statistics are relatively consistent, hologram distributions vary depending on the hologram generation method. This suggests that designing hologram generation and compression as separate modules—as is common in natural image processing—may lead to suboptimal performance. Early work by Wang et al. [6] addressed this by jointly treating hologram generation and compression via an end-to-end neural framework, achieving phase hologram compression while ensuring image quality. The method [17] extended this idea by integrating a JPEG simulator into an iterative optimization loop. Nonetheless, the compression performance of these methods remains modest, typically above 2 bits per pixel (bpp) on average, leaving substantial room for improvement in bandwidth-constrained scenarios.

In this work, we explore pushing hologram compression toward lower bitrates in both 2D and 3D holography while ensuring high reconstruction quality. This is especially beneficial for deployment in bandwidth-constrained or mobile edge scenarios, where stable, efficient transmission is critical. Although prior work [6] has made notable progress, their resulting compression rates remain relatively high, typically at around 2-5 bits per pixel (bpp). This may be due to insufficient exploitation of redundancies in hologram and its latent space. To tackle this, we propose a latent-of-latent

- * *Equal contribution*
- † *Equal advising*
- *All the authors are with University of North Carolina at Chapel Hill*

strategy: the wavefield at the SLM plane is first encoded into an initial latent space, which is further reduced to a more compact latent. The distribution of this compact latent space is then modeled for entropy coding. This hierarchical design enables more efficient compression (≈ 0.6 bpp) while maintaining reconstruction quality $> 30\ dB$ in simulation.

We incorporate efficient Vision Transformer (ViT) modules [18], [19], [20], [21] for initial latent space extraction. By adapting ViTs for modeling long-range dependencies, we extract expressive latent maps that serve as inputs to our latent-of-latent compression scheme. To achieve the scheme, we draw on techniques from Video Compression Transformer (VCT) [21], a ViT architecture tailored for compression, and strategically incorporate it to reduce data redundancy in the initial latent maps and advance compression efficiency. This nested latent-of-latent compression enables extremely low bitrates while preserving image quality. Moreover, to realize a cloud-edge collaborative design, we constrain the use of computationally intensive ViTs to hologram generation, encoding, and compression, which can be deployed on cloud servers. The decoding process is handled by a lightweight decoder that can be efficiently run on edge devices. Experimental results demonstrate our method's applicability to both 2D and 3D holography, achieving up to an $8\times$ bitrate reduction compared to the state-of-the-art method, DPRC [6], while maintaining high reconstruction quality.

In sumamry, our contributions are as follows:

- We propose a novel *latent-of-latent* compression strategy that targets low-bitrate hologram compression while maintaining high reconstruction quality, which can seamlessly be integrated into end-to-end hologram generation frameworks.
- We design ViT-based modules to effectively capture long-range dependencies when generating compact latent representations. The proposed framework supports both 2D holograms and sequential holographic video by modeling temporal redundancy through cross-attention in data distribution modeling.
- We demonstrate with comprehensive experiments that `HoloZip` can effectively compress hologram data, outperforms existing approaches such as DPRC, and requires only $\frac{1}{8}$ bpp compared to DPRC.

2 Related Work

Computer-generated holography (CGH) performs numerical simulations to generate 2D holograms that can represent the 3D light field. Due to the high diffraction efficiency of phase-only spatial light modulators (SLMs), phase-only holograms are commonly used. In this work, we primarily focus on prior efforts related to hologram and image compression. A brief overview of CGH generation methods is provided in the Supplementary Material.

2.1 Hologram Compression

The compression of holograms is a critical enabler for practical holographic applications. as holograms for reconstructing 3D scenes require huge storage space and high transmission bandwidth [22]. However, most widely used lossy image or video codecs—such as JPEG [23], JPEG2000 [24], and HEVC—are designed to exploit the statistical properties of natural images. When directly applied to holograms, which exhibit fundamentally different signal characteristics (e.g., sharp phase discontinuities and high-frequency content), these codecs typically introduce severe distortions in reconstructed images, especially at low bitrates, impairing visual fidelity of the holographic display.

Existing methods have attempted to adapt traditional video codecs for hologram compression. Blinder et al. enhance HEVC by modeling rigid motion [25] and use JPEG 2000 for static off-axis holograms [26]; however, these approaches are limited to single-object scenes or static images, respectively. Similarly, Oh et al. [27] modify HEVC for phase-only holographic videos, but their method remains non-differentiable. Muhamad et al. [28] follow a classical, non-end-to-end trainable pipeline for plenoptic data. Other approaches have explored phase-difference encoding [22], alternative phase representations [29], and phase unwrapping strategies [30]. Jia et al. [31] utilize a foveated rendering strategy that requires gaze cues from eye tracking, while our method works with full-resolution data without gaze input.

Recent efforts have explored deep learning-based hologram compression. Jiao et al. [32] develop a hybrid approach that leverages neural networks to post-process JPEG-compressed holograms in order to restore degraded reconstructions. However, since the overall pipeline is not end-to-end optimizable, the method—while partially effective—still results in reconstruction distortions. Shi et al. [33] combine H.265 with a neural residual module, but their pipeline is not end-to-end optimizable. More recently, Wang et al. [6] propose an end-to-end learnable framework for joint phase retrieval and compression. The method [6] integrates differentiable modeling of the compression process on a compact latent representation produced by neural modules. Zhou et al. [17] introduce a JPEG-aware CGH framework that integrates a differentiable JPEG simulator into an SGD-based hologram optimization loop, maintaining compatibility with the legacy JPEG codec. While effective, the compression performance of these methods still falls significantly behind that achieved in natural image compression, leaving substantial room for further improvement. The method by Ban et al. [34] supports both images and videos by using two separate hyperprior coders. In contrast, our framework unifies image and video compression by adopting a single Transformer-based entropy model on a latent-of-latent space.

2.2 Neural Image Compression

Traditional codecs such as JPEG [23] and JPEG2000 [24] use hand-crafted modules (e.g., DCT, quantization, entropy coding), which are difficult to jointly optimize, thus limiting compression efficiency [35], [36]. In recent years, neural compression methods have enabled end-to-end optimization and have shown superior performance [37], [38], [39], [40]. Early work by Ballé et al. [37] introduces an autoencoder-based framework that achieves both high pixel-level fidelity and perceptual quality, substantially outperforming traditional codecs such as JPEG. Follow-up works improved upon this framework by incorporating hyperprior models [38], enabling more accurate entropy modeling of the latent space and further reducing bitrate at similar quality. More recent efforts have explored perceptual-oriented compression

TABLE 1: Table of variables

Symbol	Data Type	Dimension	Description
$\mathbf{A}_t$	Float	$H \times W \times 1$	Target amplitude map
$\mathbf{P}_t$	Float	$H \times W \times 1$	Output from IP
$\mathbf{H}$	Float	$H \times W \times 1$	Generated hologram H
$\mathbf{v}$	Float	$\frac{H}{4} \times \frac{W}{4} \times 8$	Latent space
$\mathbf{v}_l$	Float	$\frac{H}{64} \times \frac{W}{64} \times 192$	Latent of latent space
$\hat{\mathbf{v}}$	Float	$\frac{H}{4} \times \frac{W}{4} \times 8$	Decoded $\mathbf{v}$
$\hat{\mathbf{v}}_l$	Float	$\frac{H}{4} \times \frac{W}{4} \times 8$	Decoded $\mathbf{v}_l$
μ	Float	$\frac{H}{4} \times \frac{W}{4} \times 8$	Mean of the Gaussian model for v
σ	Float	$\frac{H}{4} \times \frac{W}{4} \times 8$	Scale of the Gaussian model for v
$\hat{\mathbf{A}}_t$	Float	$H \times W \times 1$	Simulated reconstruction of A_t
c_l	Binary bits	—	Bitstream coded for $\hat{\mathbf{v}}_l$

–: The lengths of the bitstreams are dynamically changed according to the probability distribution of the elements within the data.

using adversarial losses [40] or diffusion models [41], [42] to enhance visual realism, especially at extremely low bitrates. Although these methods are developed for natural images, their core principles—such as learned entropy estimation and perceptual-aware objectives—are highly relevant to hologram compression. Inspired by this line of research, we incorporate a Transformer-based image compression framework to effectively model a compact latent-of-latent space, which is produced in our uniquely designed framework for hologram generation and compression framework, enhancing the hologram compression efficiency notably.

3 Computer Generated Holography

Computer-generated holography (CGH) numerically simulates the optical process of hologram recording and replay. In a typical setup, a phase-only spatial light modulator (SLM) is used to modulate the wavefront of a coherent light source due to its high diffraction efficiency. This requires computing a phase-only hologram $\mathbf{H}$ to produce the desired intensity distribution at the image plane. To model wave propagation from the SLM plane to the image plane, the band-limited angular spectrum method (ASM) [43] is commonly adopted for its balance between accuracy and computational efficiency. The forward propagation is formulated as:

$$\begin{aligned} f_d^p(\mathbf{H})(x,y) &= \iint \mathcal{F}\left(e^{j\mathbf{H}}\right)(u_x,u_y)\cdot\mathcal{H}(u_x,u_y) \\ &\quad\cdot e^{j2\pi(u_x x+u_y y)}\,du_x du_y, \\ \mathcal{H}(u_x,u_y) &= \begin{cases} e^{j2\pi d\sqrt{\frac{1}{\lambda^2}-u_x^2-u_y^2}}, & \text{if } u_x^2+u_y^2<\frac{1}{\lambda^2}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

$\mathcal{F}(\cdot)$ denotes the Fourier transform, and u_x, u_y are the spatial frequencies. As only the intensity $|\hat{\mathbf{A}}_t|^2$ of the reconstructed wavefield is observable, the holographic phase retrieval problem is typically formulated as an optimization task:

$$\mathbf{H} = \arg\min_{\mathbf{H}} \mathcal{L}_{rec}\left(|\hat{\mathbf{A}}_t|^2, |\mathbf{A}_t|^2\right). \tag{2}$$

$|\mathbf{A}_t|^2$ is the target intensity and $\mathcal{L}_{rec}(\cdot)$ is the reconstruction loss function such as mean squared error (MSE), total variation difference, or a weighted combination thereof.

4 HoloZip

4.1 Overview

The pipeline of our framework `HoloZip` is illustrated in Figure 1. We integrate hologram generation and compression into a unified, end-to-end trainable neural framework. Given a target RGB image (amplitude) $\mathbf{A}_t$, `HoloZip` employs an overall encoding-decoding scheme to generate a phase-only hologram $\mathbf{H}$. The encoding process yields a latent representation $\mathbf{v}$, which can be decoded into the final hologram either directly or after being transmitted. Compression modules are designed in the latent space $\mathbf{v}$, enabling highly efficient representation and transmission.

The encoding procedure consists of a phase initialization module $\mathcal{IP}$ and a latent encoding module $\mathcal{E}_p$. We integrate efficient vision transformers (ViTs)—detailed in Section 4.2—to implement these two modules to leverage ViT's strong representation capacity and enhance the hologram synthesis performance. Specifically, $\mathcal{IP}$ takes the input RGB image (amplitude) $\mathbf{A}_t$ and estimates a phase map $\mathbf{P}_t$ at the target plane. The formed wavefield is propagated to the SLM plane using the angular spectrum method (Eq. 1). The propagated wavefield is then encoded by $\mathcal{E}_p$ into a latent space $\mathbf{v}$, which contains $16\times$ fewer pixels, resulting in a $4\times$ reduction in data volume ($2 \times H \times W \rightarrow 8 \times \frac{H}{4} \times \frac{W}{4}$). And a decoder $\mathcal{D}_p$ takes either the extracted latent map $\mathbf{v}$ or the transmitted one $\hat{\mathbf{v}}$ to generate the hologram $\mathbf{H}$, which are then be propagated to reconstruct the amplitude $\hat{\mathbf{A}}_t$ at the target plane. While direct entropy encoding of latent maps $\mathbf{v}$ is feasible [6], [44], it does not fully exploit the spatial redundancy within the latent space. To achieve ultra-efficient compression, we introduce a ***latent-of-latent*** compression strategy that treats the latent $\mathbf{v}$ as an "image" and further encodes it into a more compact representation $\mathbf{q}$ before entropy coding. This hierarchical approach employs a Video Compression Transformer (VCT) (Section 4.3) to generate the final bitstream. The compression procedure is designed to support both single-frame and multi-frame (video) compression.

4.2 ViT-based Phase Estimator and Latent Encoder

Vision transformers (ViTs) [18], [19], [20] have shown remarkable success across a wide range of vision tasks. However, their application in holography, particularly for compression-oriented frameworks, remains underexplored. In `HoloZip`, we leverage ViTs to perform target-plane phase initialization and latent encoding to improve reconstruction quality of generated holograms. A major challenge in adopting ViTs for holography lies in their prohibitive computational cost, which scales quartically with input resolution—problematic for holographic systems that typically operate on high-resolution inputs (e.g., 1920×1080). To address this, we adopt the Swin Transformer architecture [19], [45], a computationally efficient ViT variant, to construct both the phase extraction network $\mathcal{IP}$ and the latent encoding network $\mathcal{E}_p$ in our framework. This design preserves the modeling advantages of ViTs while ensuring scalability for practical, high-resolution hologram computation.

Given an input image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, standard ViTs first divide it into $N = \frac{HW}{P^2}$ non-overlapping patches of size $P \times P$, which are flattened and projected into a D-dimensional embedding space, with positional encoding

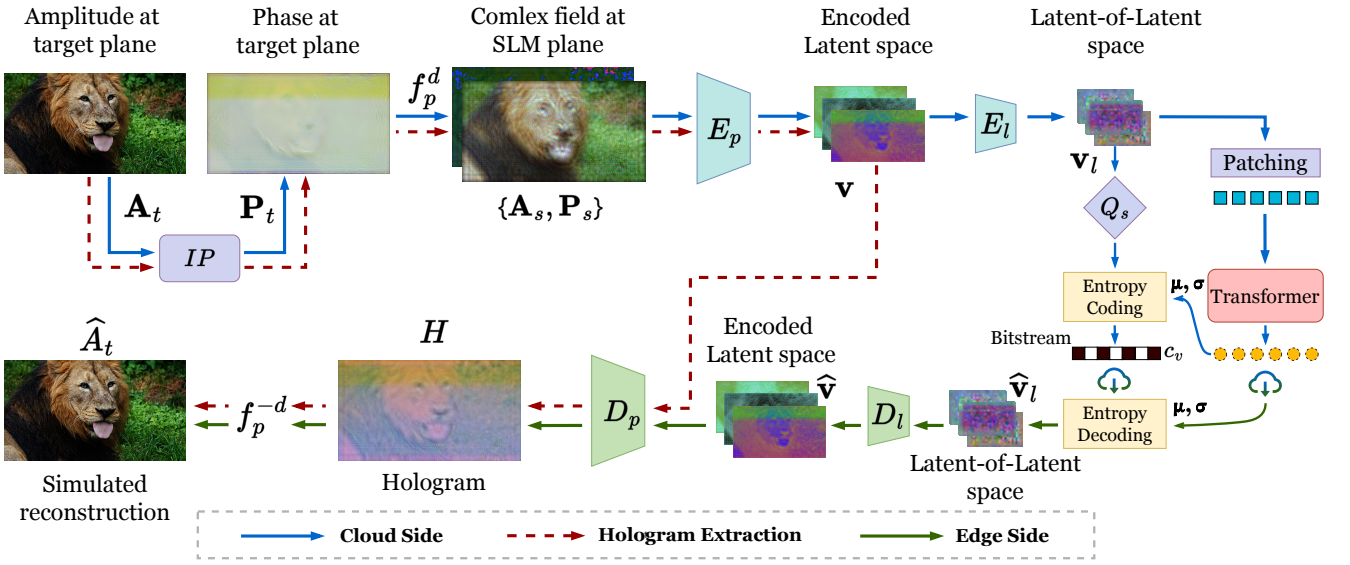

Fig. 1: Overview of the proposed ViT-based holography generation and compression framework. The components within the framework can be divided into encoding and decoding components, where the decoding components are deployed locally or to edge devices in a cloud-collaborative computation scenario. The cloud side extracts the phase map from the input natural images or video frames, and these frames are encoded into latent space by a ViT encoder to reduce the spatial size as the first stage of compression. The encoded latent maps are further compressed with the VCT to maximumly reduce the data amount for transmission as the second stage compression. After being transmitted to the edge side, a lightweight decoder first restores the extracted latent maps from the bit flow, followed by a light CNN upscaling module to restore the original spatial size. In our framework, the phase extractor $\mathcal{IP}$ and latent encoder $\mathcal{E}_p$ are implemented with ViTs, specifically Swin-UNet and Swin Transformer, respectively.

added to preserve spatial information. The resulting token sequence $\mathbf{z}_0$ is then passed through multiple transformer layers. At the l-th layer, each token first undergoes layer normalization (LN) and multi-head self-attention (MSA), followed by a residual connection, i.e., $\mathbf{z}'_l = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1}$. The result is then passed through another residual block consisting of a layer normalization (LN) and a multilayer perceptron (MLP), yielding the final output $\mathbf{z}_l = \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l$. As the patch account N grows quadratically with resolution, the computational cost of standard ViTs increases quartically and becomes prohibitive for high-resolution holograms.

To ameliorate this, we adopt the window-based multi-head self-attention W-MSA$(\cdot)$ and shifted window-based multi-head self-attention SW-MSA$(\cdot)$, which restrict the attention mechanism to local windows rather than the entire input, as illustrated in Figure 2. This window-based design greatly improves scalability while preserving modeling capacity. Following [45], we interleave the W-MSA$(\cdot)$ and SW-MSA$(\cdot)$ in consecutive layers. Specifically, for layers l and $l+1$,

$$\begin{cases} \mathbf{z}'_l = \text{W-MSA}\left(\mathrm{LN}\left(\mathbf{z}_{l-1}\right)\right) + \mathbf{z}_{l-1}, \\ \mathbf{z}_l = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'_l\right)\right) + \mathbf{z}'_l, \\ \mathbf{z}'_{l+1} = \text{SW-MSA}\left(\mathrm{LN}\left(\mathbf{z}_l\right)\right) + \mathbf{z}_l, \\ \mathbf{z}_{l+1} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'_{l+1}\right)\right) + \mathbf{z}'_{l+1}. \end{cases} \tag{3}$$

Additionally, we merge the neighboring patches after certain transformer layers to further reduce the computation cost by shrinking the resolution. This hierarchical merging strategy aligns with the architecture of the Unet [46], which has been adopted for phase extraction in the previous work [6], [11]. Motivated by this, we adopt Swin-Unet [45], which integrates the Swin Transformer into a Unet, as an efficient initial phase extractor $\mathcal{IP}$, as illustrated in Figure 3.

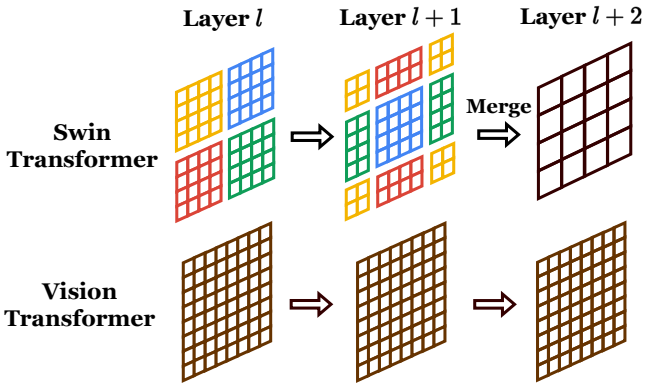

Fig. 2: Illustration of the comparison between W-MSA$(\cdot)$, SW-MSA$(\cdot)$ and standard MSA$(\cdot)$ in ViTs. Every small square denotes an image patch, and the larger colored areas represent the attention window within which patches attend to each other. In MSA$(\cdot)$, every image patch attends globally to all other patches, resulting in $O(N^2)$ computation complexity, where $N = \frac{H}{P} \times \frac{W}{P}$ is the number of patches. In contrast, both W-MSA$(\cdot)$ and SW-MSA$(\cdot)$ limit attention to local windows, reducing the computational complexity to $\frac{N^3}{d^4}$, where d is the size for each attention window. After alternating layers of W-MSA$(\cdot)$ and SW-MSA$(\cdot)$, neighboring image patches are merged to further reduce the computation cost in the following ViT layers.

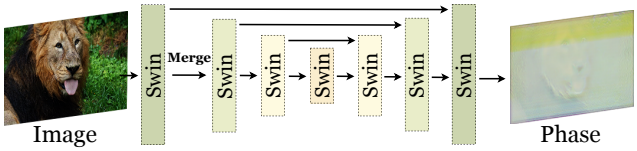

Fig. 3: Illustration of the Swin-Unet architecture. Each downsampling block consists of two consecutive Swin Transformer blocks followed by one patch merging block, which progressively reduces the spatial resolution. Each upsampling block contains one patch expanding block to recover spatial resolution, followed by two Swin Transformer blocks for feature refinement.

4.3 Latent of Latent Compression

4.3.1 Visual Compression Transformer

The overall encoder–decoder architecture reduces the volume of data to be transmitted. Building on this, DPRC [6] employs a hyperprior network [38] to model the data distribution within latent space and utilizes entropy coding [47] to compress the latent representation into a bitstream. However, DPRC's compression performance still falls short of state-of-the-art (SOTA) compression methods [21], [48], [49] for natural images and videos, primarily due to insufficient exploitation of latent space redundancy. Furthermore, when processing holographic video frames, DPRC merely leverages residuals between latent representations of consecutive frames, limiting its ability to capture temporal dependencies.

To address these limitations, we incorporate the Visual Compression Transformer (VCT) [21], an SOTA framework designed to fully exploit spatial and temporal redundancy for ultra-high compression efficiency. As shown in the right panel of Figure 4, the latent representation $\mathbf{v}$ undergoes further downsampling via an auxiliary encoder $\mathcal{E}_l$ and is patchfied, followed by entropy coding for transmission. On the receiver side, entropy decoding reconstructs the patches, which are then fed into a decoder $\mathcal{D}_p$ to recover the latent map $\hat{\mathbf{v}}$. Effective compression hinges on accurate probability estimation during entropy coding. To this end, we adopt Transformers [50] to model probability distributions of encoded latent patches. For video compression, as illustrated in Figure 4, the distribution estimation of the $(i+1)$-th frame is conditioned on the preceding i-th frame through cross-attention in the Transformer layers, achieving temporal redundancy removal.

4.3.2 Compression and Decompression

In this section, we elaborate on the coding and decoding process of the further encoded latent space (*i.e.*, latent of latent). Prior to entropy coding, the latent representation is quantized into a discrete set via a rounding operation. To enable end-to-end learning, we replace the non-differentiable rounding with a differentiable alternative, denoted as Q_s:

$$\tilde{\mathbf{x}} = Q_s(\mathbf{x}) = \mathrm{sg}([\mathbf{x}] - \mathbf{x}), \tag{4}$$

where $\mathbf{x}$ is the input to round, $\tilde{\mathbf{x}}$ is the rounded output, $\mathrm{sg}(\cdot)$ is the stop-gradient operator and $[\cdot]$ is a real rounding operation. The compression process starts by encoding the latent map $\mathbf{v}$ with the encoder $\mathcal{E}_l$ to produce the latent-of-latent $\mathbf{v}_l = \mathcal{E}_l(\mathbf{v})$, which are then divided into patches $\mathbf{p}_l$. The entropy coding takes the patches and the predicted mean $\mathbf{m}_l$ and scale $\mathbf{s}_l$ from the Transformer as input and compresses the patches $\mathbf{p}_l$ to a bitstream. Specifically, the mean $\mathbf{m}_l$ is first rounded as $\tilde{\mathbf{m}}_l = Q_s(\mathbf{m}_l)$. Then the patches $\mathbf{p}_l$ are subtracted by the rounded mean as $\tilde{\mathbf{p}}'_l = \mathbf{p}_l - \tilde{\mathbf{m}}_l$. Finally, the modified patches are coded utilizing the predicted scale $\mathbf{s}_l$

$$\mathbf{c}_l = \mathrm{Coding}(\tilde{\mathbf{p}}'_l, \mathbf{s}_l), \tag{5}$$

where $\mathbf{c}_l$ is the bitstream to transfer. The decoding process is the inverse of the coding process

$$\hat{\mathbf{p}}_l = \mathrm{Decoding}(\mathbf{c}_l, \mathbf{s}_l) + \mathbf{m}_l. \tag{6}$$

The decoded patches are then reshaped back to the reconstructed latent-of-latent $\hat{\mathbf{v}}_l$.

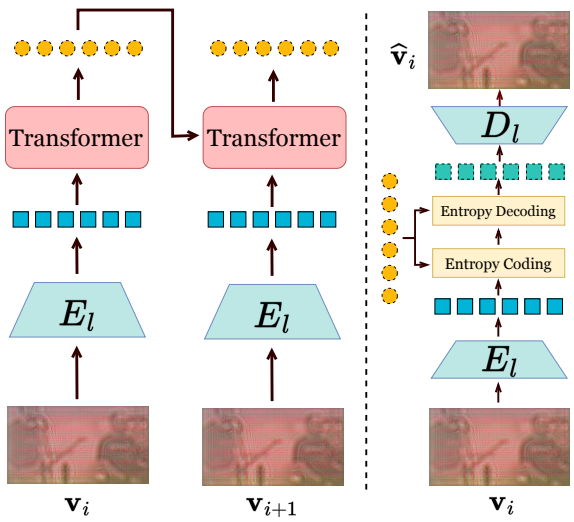

Fig. 4: Illustration of the VCT architecture. Each frame e is first processed by an encoder $\mathcal{E}_l$ to reduce the resolution and converted to patches. The patches will be used as the input for a Transformer that predicts the probability distribution required for entropy encoding. For video compression on the **left panel**, the statistical information from the preceding frame $\mathbf{v}_i$ is incorporated into the prediction of the current frame $\mathbf{v}_{i+1}$ via cross-attention in the Transformer. The **right panel** details the encoding-decoding pipeline for a single frame $\mathbf{v}_i$: The frame is initially encoded by the encoder $\mathcal{E}_l$ and transformed into patches and compressed via entropy coding guided by the Transformer-estimated probability distribution. After transmission, entropy decoding reconstructs the patches with the probability distribution, which are then restored by the decoder $\mathcal{D}_l$ to the original latent map.

4.4 Model Training

We train `HoloZip` in two stages. In the first stage, we train the hologram synthesis sub-framework—highlighted by the red arrows in Figure 1—which consists of the phase extractor $\mathcal{IP}$, the encoder $\mathcal{E}_p$, and the decoder $\mathcal{D}_p$. This sub-framework is trained by minimizing the reconstruction loss $\mathcal{L}_{\text{rec}}$ (in Eq. 3) between the reconstructed and target intensity. In the second stage, we freeze the modules trained in the first stage and activate the compression-related components, including the VCT. The overall training objective becomes:

$$\mathcal{L}_c = \alpha \cdot R + \mathcal{L}_{\text{rec}}. \tag{7}$$

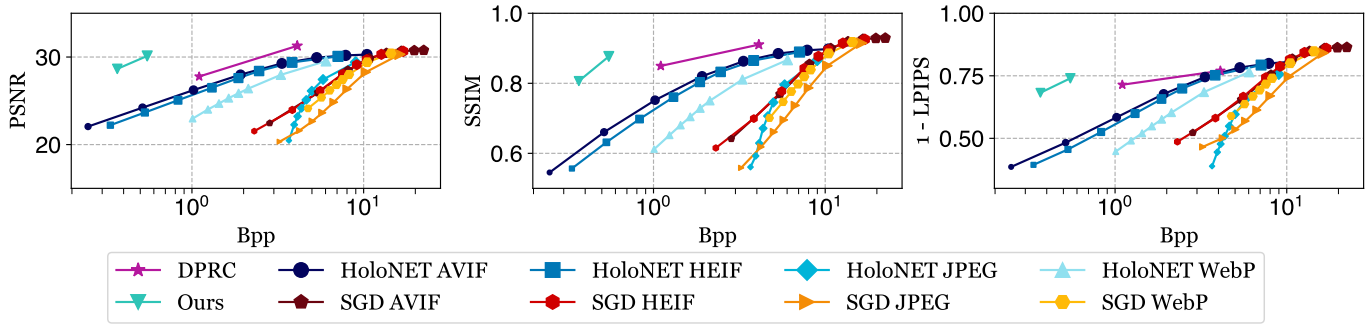

Fig. 5: Comparison of compression methods through reconstruction quality versus compression rate. Bpp represents bits per pixel consumed to encode the holograms. SGD [54], [55] and HoloNet [55] are optimization and deep learning-based methods for hologram extraction. JPEG, WebP, AVIF, and HEIF are image compression methods and are combined with SGD or HoloNet to compress the extracted holograms. A larger value in the y-axis indicates better performance.

TABLE 2: Reconstruction performance without compression. *SGD is a per-image optimization-based method. We consider it to be the upper bound for reconstruction quality.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s)
SGD*	35.0037	0.9549	0.1352	26.6641
HoloNet	29.7014	0.9114	0.2394	0.0176
DPRC	30.4155	0.9237	0.2006	0.0201
Ours (`HoloZip`)	**32.4256**	**0.9119**	**0.1883**	0.0528

TABLE 3: Results for ablation studies.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Variant 1	32.4061	0.9171	0.2149
Variant 2	32.1875	0.9158	0.2081
Ours (Full model)	**33.1414**	**0.9244**	**0.1883**

Variant 1: Replaces the ViT-based $\mathcal{IP}$ with a UNet.
Variant 2: Replaces the ViT-based $\mathcal{E}_p$ with a CNN encoder.

This loss jointly measures reconstruction fidelity and compression efficiency, where R denotes the bitrate estimated for the bitstream, and α is a weighting factor that balances the trade-off between reconstruction and compression.

5 Experiments

5.1 Settings

Our framework is evaluated on several standard datasets for holographic compression. For 2D holographic image compression, we use the DIV2K dataset [51], while for 2D holographic video compression, we use the CVQAD dataset [52]. For 3D hologram compression, the MIT-CGH dataset [14] is used for training and quantitative evaluation. For 3D hologram compression, we also perform qualitative assessments on $1080p$ natural images with depth maps generated by a depth estimation model [53].

Implementation Details. In the Supplementary Material, we provide details of the network architectures of the neural modules within the `HoloZip` framework and more details regarding the implementation and training. We also include a detailed description of the holographic display prototype we built for experimental evaluation.

TABLE 4: Effectiveness of the latent-of-latent strategy.

	PSNR ↑	SSIM ↑	LPIPS ↓	nbpp ↓
No latent-of-latent	31.6524	0.9096	0.1229	4.3737
Ours (Full model)	30.1246	0.8767	0.2599	0.5489

TABLE 5: Quality of holographic video compression. The background color denotes similar or higher bpp to `HoloZip`.

Methods	Compression	PSNR ↑	SSIM ↑	LPIPS ↓	nbpp ↓
HoloNet	H.264 crf=10	20.8714	0.6508	0.5298	0.9046
	H.264 crf=15	20.1933	0.5916	0.5784	0.4368
	H.264 crf=20	19.2868	0.5213	0.6305	0.1888
	H.264 crf=23	18.7677	0.4842	0.6542	0.1179
	vp9 crf=20	20.7154	0.6469	0.5405	0.5646
	vp9 crf=30	20.1859	0.6043	0.5761	0.2989
	vp9 crf=40	19.3638	0.5465	0.6218	0.1159
	av1 crf=61	17.5841	0.4854	0.6620	0.0035
	av1 crf=63	17.1828	0.4621	0.6727	0.0017
SGD	H.264 crf=10	19.2590	0.5025	0.5347	5.3378
	H.264 crf=15	19.1124	0.4914	0.5412	3.9938
	H.264 crf=20	18.7088	0.4618	0.5632	2.6474
	H.264 crf=23	18.2683	0.4326	0.5912	1.8662
	vp9 crf=20	18.9126	0.4806	0.5474	3.6023
	vp9 crf=30	18.4509	0.4513	0.5724	2.4231
	vp9 crf=40	17.5627	0.3988	0.6239	1.2356
	av1 crf=61	14.1708	0.3136	0.6409	0.0040
	av1 crf=63	13.9960	0.2925	0.6366	0.0016
DPRC	Single Frame	33.6677	0.9238	0.1619	3.1769
HoloZip	Low	31.6272	0.8758	0.2005	0.2591
	High	31.6328	0.8767	0.1978	0.7072

5.2 Hologram Generation Quality

We first evaluate the performance of hologram generation sub-framework (without compression-related modules involved) via evaluating reconstruction image quality and summarize the results in Table 2. We compare our approach against the per-image optimization-based method SGD [55], and deep learning-based methods HoloNet [55] and DPRC [6]. As shown in Table 2, SGD achieves the highest reconstruction quality but incurs significantly higher computation time due to the need for iterative optimization per frame. In contrast, deep learning-based methods, including `HoloZip`, perform a fast network inference per image, resulting in much faster hologram generation. Compared to HoloNet and DPRC, `HoloZip` achieves consistently better results across all three evaluation metrics, demonstrating its superior ability to recover high-fidelity holograms from input

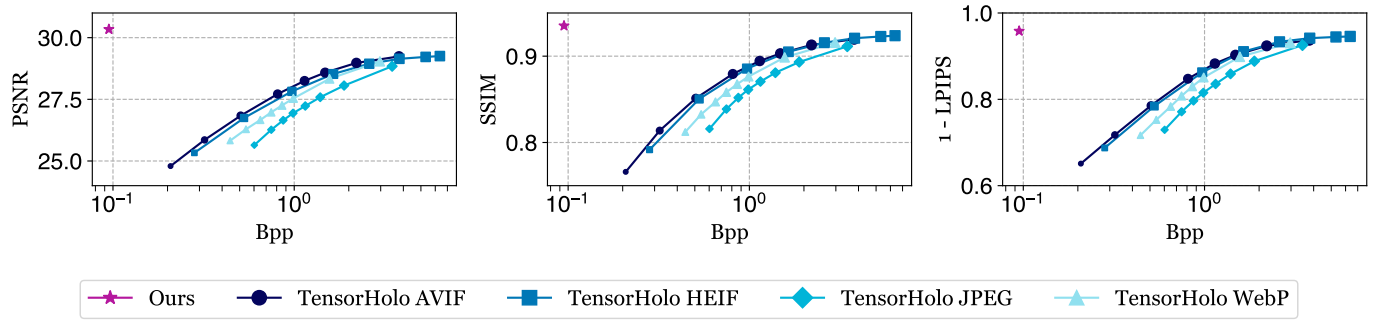

Fig. 6: Comparison of compression methods through reconstruction quality versus compression rate on 3D holography

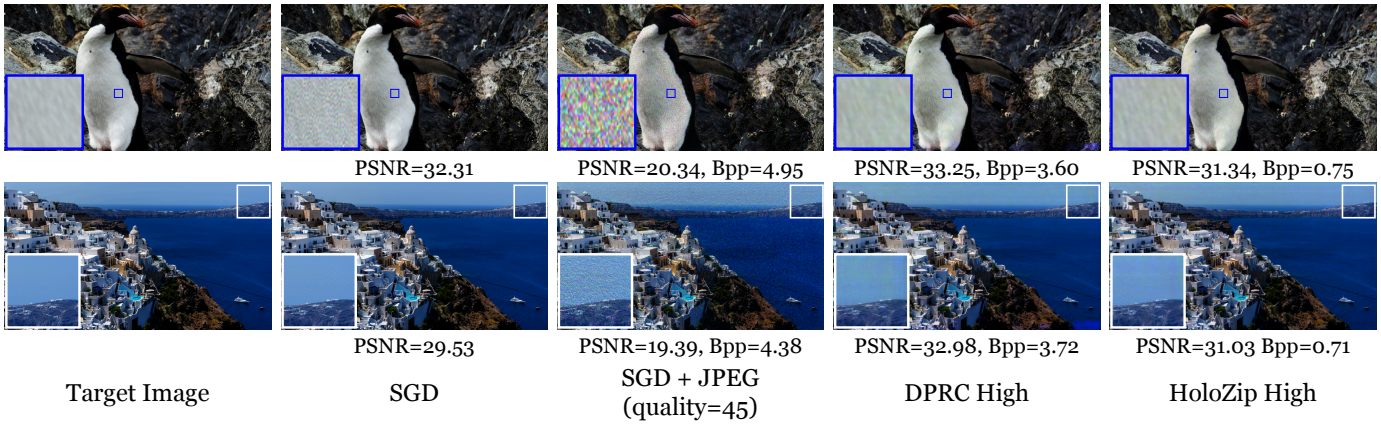

Fig. 7: Comparison of reconstructed images from compressed holograms by different 2D holography methods.

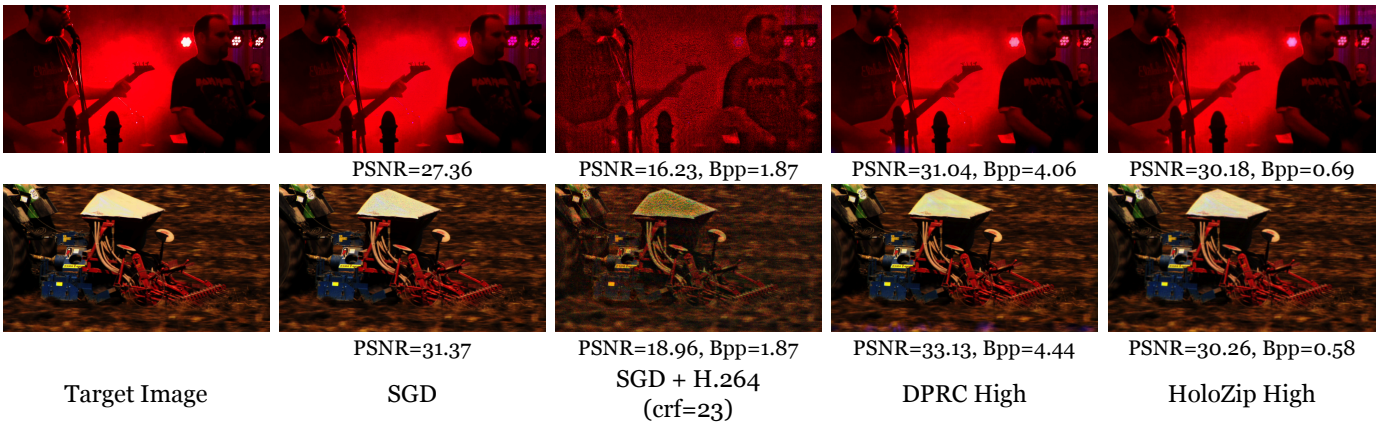

Fig. 8: Reconstructed video frames from compressed holograms by different methods.

images. Moreover, despite integrating ViTs, `HoloZip` maintains comparable inference speed to other learning-based methods, owing to the incorporated efficiency designs.

Ablation Studies. We conduct ablation studies to evaluate the effectiveness of the ViT-based phase initialization network $\mathcal{IP}$ and latent encoder $\mathcal{E}_p$. Specifically, we test two variants: Variant-1 replaces $\mathcal{IP}$ with a standard convolutional UNet, and Variant-2 replaces $\mathcal{E}_p$ with a convolutional encoder similar to that used in DPRC. The results are shown in Table 3. Compared to our full model, both variants exhibit a clear performance drop (by 0.74 dB and 0.95 dB, respectively), demonstrating the importance of our ViT-based designs. These results validate our architectural choices and support the motivation for incorporating ViTs to enhance holographic reconstruction quality.

5.3 Hologram Image Compression

We evaluate the compression performance of our proposed framework, `HoloZip`. Quantitative and qualitative results are presented in Figure 5 and Figure 7, respectively. Our comparison includes standard image codecs (JPEG [56], WebP [57], HEIF [58], and AVIF [59]) applied to holograms from SGD or HoloNet, and the learning-based method DPRC. The rate-distortion curves in Figure 5 reveal two key insights: ❶ Standard image codecs are ill-suited for hologram compression, particularly at low bitrates. These codecs, optimized for perceptual quality in natural images, often discard high-frequency information critical for holographic reconstruction. ❷ `HoloZip` achieves similar reconstruction quality to DPRC but at a significantly lower bitrate. This compression efficiency stems from our latent-of-latent strategy, which effectively exploits redundancy in the latent space

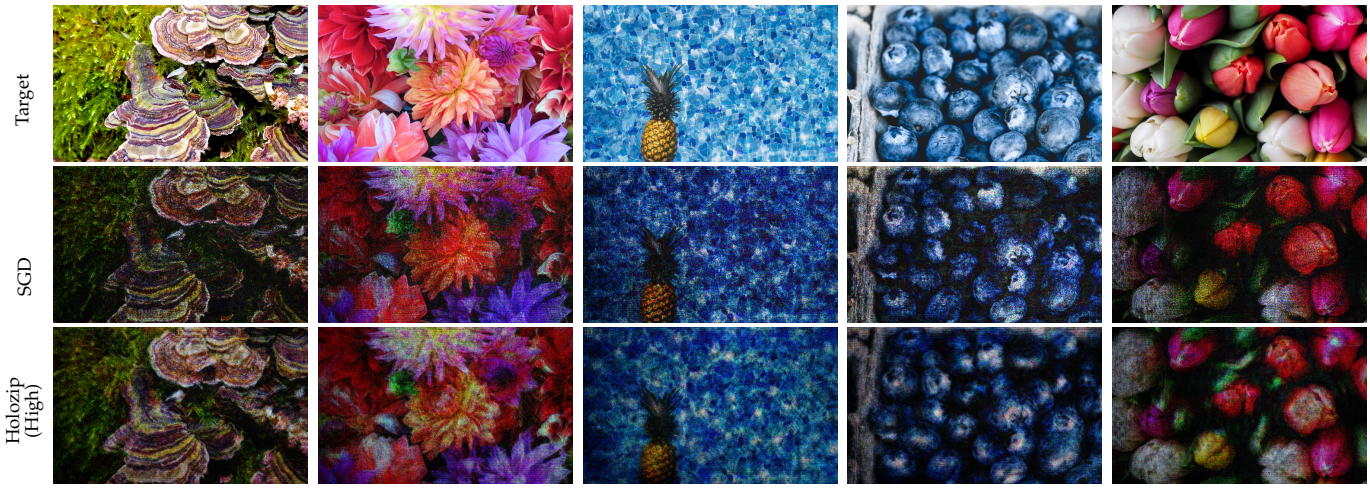

Fig. 9: Experimental results of SGD and HoloZip. For SGD, we capture images for holograms without compression.

TABLE 6: Decompression time of DPRC and HoloZip.

Decompression Time (s)	DPRC	Ours
	0.0276	0.0318

while incurring only a marginal increase in decompression time as shown in Table 6.

To verify that the notable bitrate reduction stems from the latent-of-latent compression scheme, we conduct an ablation study by disabling it. The results are summarized in Table 4. As shown, removing the latent-of-latent strategy and relying solely on a hyperprior model to compress the initial latent $\mathbf{v}$ yields similar reconstruction quality (with only a 1.5 dB increase), but leads to an $8\times$ increase in bitrate. These results demonstrate that the proposed hierarchical compression strategy substantially improves compression and transmission efficiency with minimal impact on reconstruction quality.

3D Hologram Compression. We further evaluate our method on 3D holographic image compression. As baselines, we equip TensorHolo [14] with standard image codecs including JPEG, WebP, HEIF, and AVIF. Quantitative results in Figure 6 show that the decompressed holograms from HoloZip achieve higher quality at much lower bitrates. In the Supplementary Material, we provided visual results for decompressed holograms from our method and alternative solutions, which demonstrate that our method preserves hologram quality under high compression ratios. Simulated reconstructions of decompressed 3D holograms produced by our method are also included in the Supplementary Material.

5.4 Hologram Video Compression

We further evaluate HoloZip on holographic video compression. We conduct experiments on the CVQAD dataset [52] and compare it with HoloNet and SGD equipped with H.264 [60], vp9 [61], and av1 [62] codecs. From these results, we observe the following: ❶ Although video codecs are able to high compression ratios, applying them directly to compress extracted holographic videos leads to a substantial drop in reconstruction quality. ❷ DPRC achieves higher reconstruction quality than video codecs but suffers from limited compression efficiency. ❸ In contrast, HoloZip achieves a significantly better balance, offering much higher compression efficiency while maintaining comparable or superior reconstruction quality. At similar bitrates, HoloZip clearly outperforms all tested video codecs. Sampled visual results are provided in Figure 8.

5.5 Experimental Validation of HoloZip

To further validate our method, we conduct experiments using a custom-built holographic prototype display. Details of the prototype display are provided in the Supplementary Material. Representative results are shown in Figure 9. We compare our method with holograms generated by SGD without compression, used here as a reference. The holograms produced by HoloZip undergo compression and decompression to simulate practical transmission scenarios. Despite this, HoloZip achieves comparable visual quality to SGD in the captured images, further demonstrating its effectiveness and applicability in real holographic display systems.

6 Conclusion

In this work, we explore advancing hologram compression toward lower bitrates while preserving high reconstruction quality—essential for real-world, bandwidth-limited AR/VR application scenarios. To this end, we propose a latent-of-latent compression strategy and incorporate it into developing an end-to-end learnable framework for hologram generation and compression. Moreover, we leverage Vision Transformers (ViTs), known for modeling long-range dependencies, to learn more compact and expressive latent representations. Experimental results show that our method achieves high-fidelity reconstruction at low bitrates as low as 0.6 bpp, reducing transmission overhead by up to 8×, while consistently maintaining reconstruction quality above the 30 dB threshold—significantly outperforming prior methods such as DPRC. This makes our framework highly promising for bandwidth-constrained settings, including mobile and cloud-assisted AR/VR. We further extend our approach to support holographic video, further improving compression effciency via capturing temporal redundancy for sequential

data. Looking forward, we see promising opportunities to extend this framework to more complex and dynamic 3D assets, enabling interactive applications such as holographic gaming and real-time teleportation.

ACKNOWLEDGMENTS

This work has been supported by National Science Foundation (NSF) grant #2107454.

REFERENCES

[1] D. Kim, S.-W. Nam, B. Lee, J.-M. Seo, and B. Lee, "Accommodative holography: improving accommodation response for perceptually realistic holographic displays," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[2] D. Kim, S.-W. Nam, S. Choi, J.-M. Seo, G. Wetzstein, and Y. Jeong, "Holographic parallax improves 3d perceptual realism," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–13, 2024.

[3] A. Maimone, A. Georgiou, and J. S. Kollin, "Holographic near-eye displays for virtual and augmented reality," *ACM Transactions on Graphics (Tog)*, vol. 36, no. 4, pp. 1–16, 2017.

[4] J. Kim, M. Gopakumar, S. Choi, Y. Peng, W. Lopes, and G. Wetzstein, "Holographic glasses for virtual reality," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9.

[5] M. Gopakumar, G.-Y. Lee, S. Choi, B. Chao, Y. Peng, J. Kim, and G. Wetzstein, "Full-colour 3d holographic augmented-reality displays with metasurface waveguides," *Nature*, vol. 629, no. 8013, pp. 791–797, 2024.

[6] Y. Wang, P. Chakravarthula, Q. Sun, and B. Chen, "Joint neural phase retrieval and compression for energy-and computation-efficient holography on the edge," *ACM Transactions on Graphics*, vol. 41, no. 4, 2022.

[7] K. Wakunami and M. Yamaguchi, "Calculation for computer generated hologram using ray-sampling plane," *Optics Express*, vol. 19, no. 10, pp. 9086–9101, 2011.

[8] K. Matsushima, "Computer-generated holograms for three-dimensional surface objects with shade and texture," *Applied optics*, vol. 44, no. 22, pp. 4607–4614, 2005.

[9] H. Kim, J. Hahn, and B. Lee, "Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography," *Applied optics*, vol. 47, no. 19, pp. D117–D127, 2008.

[10] R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972. [Online]. Available: https://api.semanticscholar.org/CorpusID:55691159

[11] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, 2020.

[12] P. Chakravarthula, Y. Peng, J. Kollin, H. Fuchs, and F. Heide, "Wirtinger holography for near-eye displays," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, 2019.

[13] S. Choi, M. Gopakumar, Y. Peng, J. Kim, and G. Wetzstein, "Neural 3d holography: Learning accurate wave propagation models for 3d holographic virtual and augmented reality displays," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, 2021.

[14] L. Shi, B. Li, C. Kim, P. Kellnhofer, and W. Matusik, "Towards real-time photorealistic 3d holography with deep neural networks," *Nature*, vol. 591, no. 7849, pp. 234–239, 2021.

[15] D. Yang, W. Seo, H. Yu, S. I. Kim, B. Shin, C.-K. Lee, S. Moon, J. An, J.-Y. Hong, G. Sung *et al.*, "Diffraction-engineered holography: Beyond the depth representation limit of holographic displays," *Nature Communications*, vol. 13, no. 1, pp. 1–11, 2022.

[16] M. Hossein Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, "Deepcgh: 3d computer-generated holography using deep learning," *Optics Express*, vol. 28, no. 18, pp. 26 636–26 650, 2020.

[17] M. Zhou, H. Zhang, S. Jiao, P. Chakravarthula, and Z. Geng, "End-to-end compression-aware computer-generated holography," *Optics Express*, vol. 31, no. 26, pp. 43 908–43 919, 2023.

[18] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[20] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[21] F. Mentzer, G. Toderici, D. Minnen, S.-J. Hwang, S. Caelles, M. Lucic, and E. Agustsson, "Vct: A video compression transformer," *arXiv preprint arXiv:2206.07307*, 2022.

[22] H. Gu and G. Jin, "Phase-difference-based compression of phase-only holograms for holographic three-dimensional display," *Optics Express*, vol. 26, no. 26, pp. 33 592–33 603, 2018.

[23] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[24] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer Publishing Company, Incorporated, 2013.

[25] D. Blinder, C. Schretter, and P. Schelkens, "Global motion compensation for compressing holographic videos," *Optics express*, vol. 26, no. 20, pp. 25 524–25 533, 2018.

[26] D. Blinder, T. Bruylants, H. Ottevaere, A. Munteanu, and P. Schelkens, "Jpeg 2000-based compression of fringe patterns for digital holographic microscopy," *Optical Engineering*, vol. 53, no. 12, pp. 123 102–123 102, 2014.

[27] K.-J. Oh, H. Ban, S. Choi, H. Ko, and H. Y. Kim, "Hevc extension for phase hologram compression," *Optics Express*, vol. 31, no. 6, pp. 9146–9164, 2023.

[28] R. Kizhakkumkara Muhamad, T. Birnbaum, A. Gilles, S. Mahmoudpour, K.-J. Oh, M. Pereira, C. Perra, A. Pinheiro, and P. Schelkens, "Jpeg pleno holography: scope and technology validation procedures," *Applied optics*, vol. 60, no. 3, pp. 641–651, 2021.

[29] A. V. Zea, A. L. V. Amado, M. Tebaldi, and R. Torroba, "Alternative representation for optimized phase compression in holographic data," *OSA Continuum*, vol. 2, no. 3, pp. 572–581, 2019.

[30] H.-Y. Tu, C.-H. Hsieh, and H.-G. Hoang, "Compression of phase image for three-dimensional object," in *2014 21st International Workshop on Active-Matrix Flatpanel Displays and Devices (AM-FPD)*. IEEE, 2014, pp. 97–100.

[31] J. Jia, Z. Dong, Y. Ling, Y. Li, and Y. Su, "Deep learning-based approach for efficient generation and transmission of high-definition computer-generated holography," in *Advances in Display Technologies XIV*, vol. 12908. SPIE, 2024, pp. 35–41.

[32] S. Jiao, Z. Jin, C. Chang, C. Zhou, W. Zou, and X. Li, "Compression of phase-only holograms with jpeg standard and deep learning," *Applied Sciences*, vol. 8, no. 8, p. 1258, 2018.

[33] L. Shi, R. Webb, L. Xiao, C. Kim, and C. Jang, "Neural compression for hologram images and videos," *Optics Letters*, vol. 47, no. 22, pp. 6013–6016, 2022.

[34] H. Ban, S. Choi, J. Y. Cha, Y. Kim, and H. Y. Kim, "Nhvc: Neural holographic video compression with scalable architecture," in *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2024, pp. 969–978.

[35] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[36] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1683–1698, 2020.

[37] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *Ineternational Conference on Learning Representation*, 2017.

[38] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Ineternational Conference on Learning Representation*, 2018.

[39] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," vol. 31, 2018.

[40] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," vol. 33, 2020, pp. 11 913–11 924.

[41] E. Hoogeboom, E. Agustsson, F. Mentzer, L. Versari, G. Toderici, and L. Theis, "High-fidelity image compression with score-based generative models," *arXiv preprint arXiv:2305.18231*, 2023.

[42] R. Yang and S. Mandt, "Lossy image compression with conditional diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 64 971–64 995, 2023.

[43] K. Matsushima and T. Shimobaba, "Band-limited angular spectrum method for numerical simulation of free-space propagation in far

and near fields," *Optics express*, vol. 17, no. 22, pp. 19 662–19 673, 2009.
[44] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.
[45] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *European conference on computer vision*. Springer, 2022, pp. 205–218.
[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
[47] J. Rissanen and G. Langdon, "Universal modeling and coding," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 12–23, 1981.
[48] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5708–5717, 2022. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=\&arnumber=9879846
[49] Y. Gao, W. Huang, S. Li, H. Yuan, M. Ye, and S. Ma, "Spatial-temporal transformer based video compression framework," in *arXiv.org*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusId:262084218
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[51] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.
[52] A. Antsiferova, S. Lavrushkin, M. Smirnov, A. Gushchin, D. Vatolin, and D. Kulikov, "Video compression dataset and benchmark of learning-based video-quality metrics," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 13 814–13 825. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/59ac9f01ea2f701310f3d42037546e4a-Paper-Datasets_and_Benchmarks.pdf
[53] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," 2024. [Online]. Available: https://arxiv.org/abs/2401.10891
[54] P. Chakravarthula, Y. Peng, J. Kollin, H. Fuchs, and F. Heide, "Wirtinger holography for near-eye displays," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.
[55] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–14, 2020.
[56] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
[57] "An image format for the Web | WebP," https://developers.google.com/speed/webp.
[58] "HEIF - High Efficiency Image File Format," https://nokiatech.github.io/heif/index.html.
[59] "AV1 Image File Format (AVIF)," https://aomediacodec.github.io/av1-avif/v1.1.0.html.
[60] "H.264 : Advanced video coding for generic audiovisual services," https://www.itu.int/rec/T-REC-H.264.
[61] "The WebM Project | VP9 Video Codec Summary," https://www.webmproject.org/vp9/.
[62] "AV1 Video Codec," https://aomedia.org/specifications/av1/.