ABCD : Arbitrary Bitwise Coefficient for De-quantization

Woo Kyoung Han Byeonghun Lee Sang Hyun Park Kyong Hwan Jin*
Daegu Gyeongbuk Institute of Science and Technology (DGIST), Korea
{cjss7894, qudgns1113, shpark13135, kyong.jin}@dgist.ac.kr

Abstract

Modern displays and contents support more than 8bits image and video. However, bit-starving situations such as compression codecs make low bit-depth (LBD) images (<8bits), occurring banding and blurry artifacts. Previous bit depth expansion (BDE) methods still produce unsatisfactory high bit-depth (HBD) images. To this end, we propose an implicit neural function with a bit query to recover de-quantized images from arbitrarily quantized inputs. We develop a phasor estimator to exploit the information of the nearest pixels. Our method shows superior performance against prior BDE methods on natural and animation images. We also demonstrate our model on YouTube UGC datasets for de-banding. Our source code is available at https://github.com/WooKyoungHan/ABCD

1. Introduction

The bit depth in digital contents means a number of binary digits representing pixel values. As humans recognize a wide range of color and luminance, modern display devices and cameras support more than the 8-bit depth of image and video [21, 28]. Regardless of these efforts, the image and video codecs enforce HBD images to be quantized into LBD images due to bit starvation. Thus, most contents are under 8 bits leading to false contours and blurry artifacts. Bit depth expansion, a.k.a. de-quantization, aims to recover missing bits caused by such quantizations.

Conventional methods such as [6, 10, 19, 24, 36–38] have been proposed for the de-quantization problem. However, these methods suffer from blurry artifacts resulting in distortions of details or false contours in extreme BDE. Recently, learning-based approaches, a.k.a. deep neural network, have shown remarkable performances in BDE [4, 9, 18, 26, 32, 40, 43]. Most learning-based approaches [4, 9, 32, 40, 43] reconstruct HBD images in an end-to-end manner. Recent methods [18, 26] recover residual components corresponding to missing bits from LBD images. In particular, the method called D16 [26], with the best performance so far, conducts a binary classification per each bit plane. However, D16 requires multiple deep neural networks models for every bit-planes.

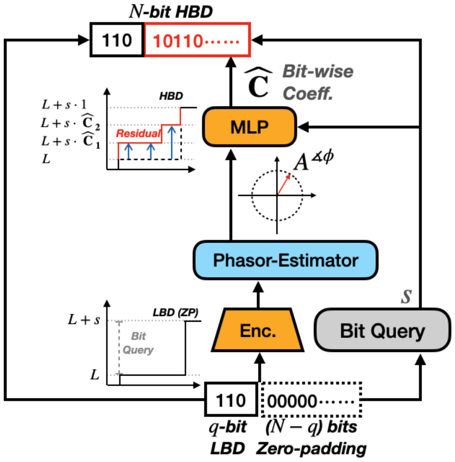


Figure 1. **Overview of arbitrary bit-depth expansion (dequantization) using ABCD.** Our ABCD estimates dominant phasors of images and calculates the bit-query of LBD. Then, an MLP takes the estimated phasor information and bit-wise query (s) to predict the bit-wise coefficient ($\widehat{\mathbf{C}}$) of HBD images.

Recently, an implicit neural representation (INR) which maps coordinates into signal values [25, 29], shows promising performances in various tasks [5, 11, 15, 22, 25, 29]. The implicit neural networks have a spectral bias problem toward low frequencies, which makes INR hard to represent high-frequency components [27]. Fortunately, several solutions are developed to relax the spectral bias [15, 23, 30, 33, 41]. However, there is no INR approach for bit depth expansion problems.

In this paper, we propose a novel model, the Arbitrary Bit-wise Coefficient model for De-quantization (ABCD), to recover missing bits from the randomly quantized LBD image to any HBD image. The proposed model addresses the spectral bias of INR and improves de-quantization quality through the use of an encoder to estimate the dominant phasors in the ground truth. As shown in Fig. 1, our encoder estimates the dominant phasors to mitigate the spectral bias of INR. Then, the model utilizes an INR to achieve arbitrary-bit reconstructions in the amplitude domain. Finally, a bit

[*] Corresponding author.

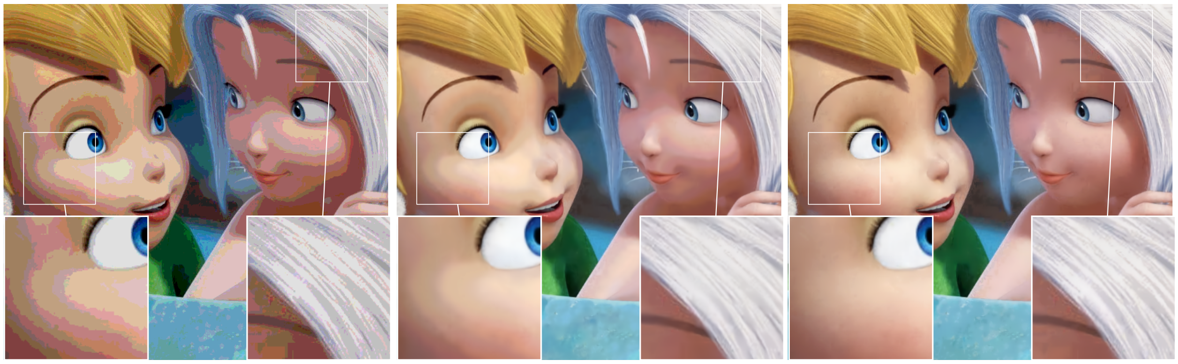


Figure 2. **Visual demonstration** 3-bit to 8-bit de-quantization: input, deep network approach (D16 [26]), and our method. The neural network method [26] reduces severe false contours of input, but false contours still remain. Our ABCD removes such artifacts, clearly.

decoding step converts bit coefficients into HBD images by multiplying the bit-basis. The proposed model represents a significant advancement over previous de-quantization techniques with providing high flexibility and accuracy as it effectively recovers missing bits from randomly quantized inputs.

In summary, our main contributions are as follows:

- We propose a bit depth expansion algorithm using an implicit neural representation with a bit query in arbitrarily quantized bit levels and demonstrate our method achieves state-of-the-art performance.
- We show that the proposed phasor estimator predicts the dominant phasors of the ground truth coefficient' in the Fourier domain.
- We validate our pre-trained model not only on five image datasets as de-quantization but also on the YouTube-UGC dataset as de-banding.

2. Related Work

Bit depth expansion There are straightforward ways for BDE, such as zero padding (ZP) method and the bit replication (BR) [36], which sets '0' or most significant bits (MSBs) for missing bits. Even though these algorithms are hardware friendly, each reconstructed signal depends only on its value without considering surrounding pixels. In contrast, interpolation methods [6, 37], content-adaptive (CA) BDE, and contour region reconstruction (CRR) effectively remove false contour artifacts. However, they blur the details. Intensity potential for adaptive de-quantization (IPAD) method [19] proposed an iterative algorithm which uses the intensity potential field calculated with the connected component label. IPAD achieved higher PSNR than the algorithms above. However, IPAD still suffers from false contour artifacts in large BDE.

Meanwhile, the BDE is highly related to de-banding. The banding artifacts are staircase-like color phenomena of images, especially in video contents. Previous works [31, 34] resolved de-banding with adaptive filter-based methods. Because the artifact is mainly derived from compression with quantization, we apply our de-quantization method to remove the banding artifact.

Learning-based Bit Recovery The proposed neural network algorithms [4, 18, 32, 40] predict favorable HBD images with higher performance than the aforementioned expert rule systems [19, 37]. The BE-CALF [18] performs BDE by recovering residuals of LBD images. However, it only supports dedicated bit LBD images as inputs. Even though the BitNet [4] supports 3 to 6-bit inputs with a single model, the performance is not as good as BE-CALF. Unlike previous works, the D4 and D16 networks [26] employ the bit-wise classification per each missing bit-plane. However, multiple models are required for predicting each bit-plane (e.g., 4 to 8-bit BDE requires four independent models). Furthermore, Fig. 2 shows that the learning-based approach [26] is not able to remove false contours completely. Unlikely the prior works, we designed the network receiving randomly quantized LBD images with single training.

Implicit Neural Representation (INR) Recently, various tasks [11, 22, 29] apply a multi-layer perceptron (MLP) as implicit neural representation. Although the INR parameterizes coordinates to continuous signals with memory-efficient frameworks, INR has two issues: per-scene optimization and spectral bias. To overcome a problem of per-scene optimization, prior works [5, 15] concatenate latent features from inputs with coordinates. The spectral bias [27] induces the network to learn mainly low frequencies. Recently, in the super-resolution task, the local texture estimator (LTE) [15] overcomes such spectral bias by learning high-resolution images' dominant frequency and phases. Nevertheless, there is no INR for de-quantization to our best knowledge. Therefore, we suggest INR as a function of given LBD data and amplitude coordinates calculated with input and output bit-depth. Moreover, we develop

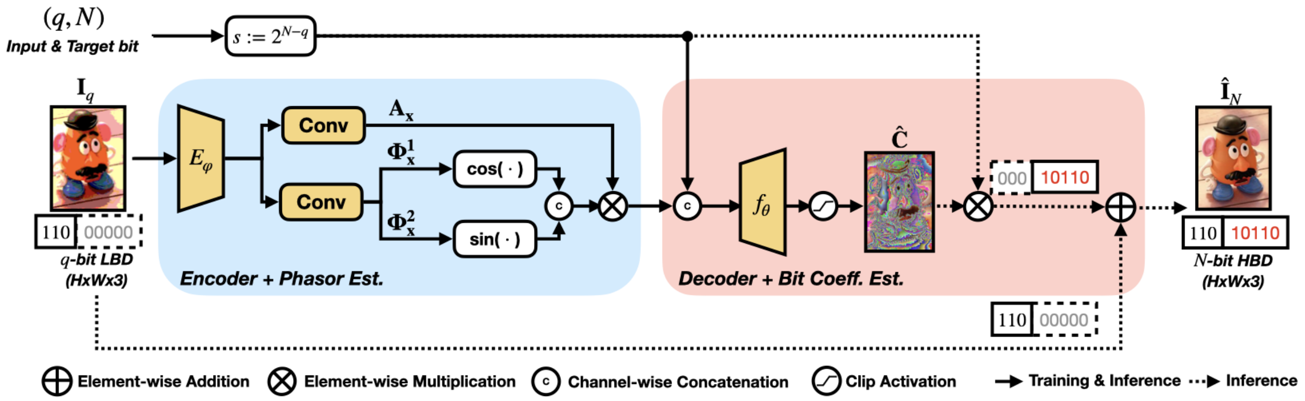


Figure 3. **Bit-depth expansion with our proposed Arbitrary Bit-wise Coefficient De-quantizator (ABCD).** ABCD-based arbitrary-bit depth expansion consists of an encoder (E_φ), a decoder (f_θ), a phasor detector (a blue-shaded region), and a coefficient estimator (a pink-shaded region). Inputs of ABCD are as follows: LBD images ($\mathbf{I}_q$), input bit (q), and target bit (N). ABCD transforms the features into phasor representations and concatenates with bit-query (s) to predict coefficient $\hat{\mathbf{C}}$. At test time, ABCD multiplies s with $\hat{\mathbf{C}}$ and adds LBD image $\mathbf{I}_q$ to retrieve HBD image $\mathbf{I}_N$.

a phasor estimator so that the network avoids spectral bias.

3. Problem Formulation

In this section, we analyze images with modular arithmetic properties and formulate implicit neural representation as a function of amplitude. Let $\mathbf{I}_N$ and $\mathbf{I}_q$ be an HBD image and an LBD image, respectively, with bit-plane zero padding when $q < N$. Then, the quantization equation from $\mathbf{I}_N$ to $\mathbf{I}_q$ is defined as below:

$$\mathbf{I}_q = \left\lfloor \frac{\mathbf{I}_N}{2^{N-q}} \right\rfloor 2^{N-q}, \tag{1}$$

where $\lfloor \cdot \rfloor$ is a floor function mapped to the greatest integer. $\mathbf{I}_q$ contains q MSBs of $\mathbf{I_N}$. Then, The HBD image is a sum of the LBD image $\mathbf{I}_q$ and the residual image $\mathbf{R}$ as below:

$$\mathbf{I}_N = \mathbf{I}_q + \mathbf{R}. \tag{2}$$

Decomposition of Quantized Signal Our approach estimates the residual image $\mathbf{R}$ with a function of both encoded latent variables and a bit-wise query. Here, we define bit-wise basis and its coefficients which normalize quantization residuals. The binary set $\{0, 1\}$ together with exclusive-or (XOR) operation $\oplus$ and multiplication $\cdot$ compose a binary field denoted by $\mathbb{F}_2$. It is known that binary vector space $\{0, 1\}^N$ over $\mathbb{F}_2$ have orthonormal basis as one-hot encoding denoted as e_n where $n = 0, \ldots, N-1$ and n is integer [1]. We interpret an arbitrary N-bit number as an element of the N-dimensional binary vector space $(\{0, 1\}^N, \oplus, \cdot)$. To realize such a binary vector in a real number, we present an arbitrary positive number a as a power series of 2 with elements (b_i) from a binary vector:

$$a = \sum_{i=-\infty}^{\infty} b_i 2^i, \tag{3}$$

where $b_i \in \{0, 1\}$ and $i \in \mathbb{Z}$. When we split above series in Eq. (3) into two terms with respect to any integer L,

$$a = \sum_{i=L+1}^{\infty} b_i 2^i + \sum_{j=-\infty}^{L} b_j 2^j = \sum_{i=L+1}^{\infty} b_i 2^i + C \cdot 2^{L+1}, \tag{4}$$

where $C \in [0, 1]$. Note that digital images consist of positive integers such as "uint8". We present Eq.(2) with Eq.(4) into a digital image form as below:

$$\mathbf{I}_N = \underbrace{\sum_{i=N-q}^{N-1} 2^i \cdot \mathbf{B}_i}_{\mathbf{I}_q} + \underbrace{2^{N-q} \cdot \mathbf{C}}_{\mathbf{R}}, \tag{5}$$

where $\mathbf{B}_i \in \{0, 1\}^{H \times W \times 3}$ denotes a binary image at i-th least significant bit-plane. Thus, the residual image $\mathbf{R}$ is a multiplication between 2^{N-q} and the real number image $\mathbf{C} \in [0, 1)^{H \times W \times 3}$. The orthonormal basis e_q of the binary vector space is equivalent to 2^{N-q} in the arithmetic form. From now on, we call $\mathbf{C}$ bit-wise coefficient of the bit-wise query ($s := 2^{N-q}$) which is equivalent to e_q.

Arbitrary Bitwise Coefficients Our method, ABCD, aims to predict coefficient $\mathbf{C}$ to reconstruct residual images by using the implicit neural representation (INR) as a function of latent vector and its quantizing bit-query. The INR parameterizes a continuous signal with an MLP fed by coordinates [5, 23, 29]. We design INR (f_θ) using amplitudes as an input coordinate so that it represents bit-wise coefficients $\mathbf{C}$ as continuous signals along the amplitude axis as follows:

$$\mathbf{C}(\mathbf{x}, s) \simeq f_\theta(\mathbf{I}_q[\mathcal{N}(\mathbf{x})], s), \tag{6}$$

where $\mathbf{x}$ is a 2D coordinate in the image domain and $\mathcal{N}(\mathbf{x})$ is a set of nearest pixels around the center pixel, $\mathbf{x}$. A decoder f_θ maps coefficient image $\widehat{\mathbf{C}}$ from domain composed with latent vector and bit-wise query; $f_\theta(z, s) : (\mathcal{Z}, \mathcal{S}) \mapsto \mathcal{C}$, where $z \in \mathcal{Z}$ is a latent tensor from an encoder E_φ, $\mathcal{S}$

is set of basis e_i and $\mathcal{C}$ is a space of predicted coefficient values from f_θ.

Phasor Estimator Inspired from [15, 33], we insert phasor estimator by modifying local texture estimator [15] to relax spectral bias. We demonstrate the difference between our ABCD and local texture estimator in Sec. 6. The overall system is constructed as follows:

$$\widehat{\mathbf{C}}(\mathbf{x}, \mathbf{I}_q, s; \boldsymbol{\Theta}) = f_\theta(h_\psi(\mathbf{z}_\mathbf{x}), s), \tag{7}$$

where $\mathbf{z} = E_\varphi(\mathbf{I}_q)$, $h_\psi(\cdot)$ denotes the phasor estimator of ABCD and $\boldsymbol{\Theta} = \{\theta, \varphi, \psi\}$ is a set of trainable parameters. Our phasor estimator ($h_\psi(\cdot)$) consists of two elements: (1) an amplitude estimator ($h_a(\cdot) : \mathbb{R}^C \mapsto \mathbb{R}^{2K}$) and (2) a phase estimator ($h_p(\cdot) : \mathbb{R}^C \mapsto \mathbb{R}^{2K}$). Thus, given a coordinate $\mathbf{x} \in \mathbb{R}^2$, the estimating function $h_\psi(\cdot) : (\mathbb{R}^C) \mapsto \mathbb{R}^{2K}$ is defined as

$$h_\psi(\mathbf{z}_\mathbf{x}) = \begin{bmatrix} \mathbf{A}^1_\mathbf{x} \\ \mathbf{A}^2_\mathbf{x} \end{bmatrix} \odot \begin{bmatrix} \cos(\pi \boldsymbol{\Phi}^1_\mathbf{x}) \\ \sin(\pi \boldsymbol{\Phi}^2_\mathbf{x}) \end{bmatrix}, \tag{8}$$

$$\text{where } \begin{bmatrix} \mathbf{A}^1_\mathbf{x} \\ \mathbf{A}^2_\mathbf{x} \end{bmatrix} = h_a(\mathbf{z}_\mathbf{x}), \quad \begin{bmatrix} \boldsymbol{\Phi}^1_\mathbf{x} \\ \boldsymbol{\Phi}^2_\mathbf{x} \end{bmatrix} = h_p(\mathbf{z}_\mathbf{x}), \tag{9}$$

$\mathbf{A}^{1,2}_\mathbf{x} \in \mathbb{R}^K$ is an amplitude vector at $\mathbf{x}$, $\boldsymbol{\Phi}^{1,2}_\mathbf{x} \in \mathbb{R}^K$ denotes a phase vector at $\mathbf{x}$, and $\odot$ represents element-wise multiplication. We interpret that by observing pixels inside a receptive field (RF), ABCD with the encoder ($h_\psi \circ E_\varphi$) estimates dominant phasors accurately. Here, the size of RF is decided by the encoder (E_φ). We visually demonstrate estimated phasors in Fig. 6.

Given a series of $\mathcal{J}$ images with the different levels of quantizations between 3 and 8, the learning problem is defined as follows:

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_{j \in \mathcal{J}} \sum_{q \in [3,8]} \|\mathbf{C}(\mathbf{I}^j_N) - \widehat{\mathbf{C}}(\mathbf{I}^j_q, s; \boldsymbol{\Theta})\|_2. \tag{10}$$

Bit decoding To retrieve de-quantized image values as in Eq.(5), bitwise coefficients ($\mathbf{C}$) is multiplied with basis value s and added with corresponding input value $\mathbf{I}_q$ from bitplane zero-padding. The whole process of decoding is written as follows:

$$\hat{\mathbf{I}}_N(\mathbf{x}) = \underbrace{\widehat{\mathbf{C}}(\mathbf{x}, \mathbf{I}_q, s; \widehat{\boldsymbol{\Theta}}) \times s}_{\mathbf{R}} + \mathbf{I}_q(\mathbf{x}) \tag{11}$$

4. Method

4.1. Network Detail

Our ABCD-based arbitrary BDE network includes an encoder (E_φ), a phasor detector (blue shaded area in Fig. 3), a decoder (f_θ), and a bit-wise coefficient estimator (pink shaded area in Fig. 3). This section describes a backbone structure (including encoder and decoder) and architectural details of ABCD.

Encoder (E_φ) and Decoder (f_θ) We use EDSR [17], RDN [42], and SwinIR [16] as an encoder (E_φ). The EDSR [17] is composed of 38 ResBlocks with 128 channels. The RDN [42] is composed of 8 RDB blocks. Since the BDE task requires equivalent sizes between input and output, we apply encoders without upsampling layers. The decoder (f_θ) is composed with 5-layer MLP and ReLU activation function [5, 15]. The dimension of the first layer is 257, and the hidden dimensions are 256.

ABCD Our ABCD contains a phasor estimator and a bit-wise coefficient estimator. Inspired by the LTE [14, 15], we hypothesize that our ABCD learns the phasor distribution from Fourier representations. The phasor estimator contains amplitude estimator (h_a), phase estimator (h_p) and sinusoidal activations. These estimators consist of 3×3 convolution layers having 256 output channels. Bit-wise coefficient estimator calculates bit-query s following Eq.(5), and concatenates it with the output of the phasor estimator (Eq.(8)). The concatenated vector contains 257 channels which are identical to an input dimension of MLP. We clip the output of network with normalized $\tanh$ activation ($0 \leq \cdot \leq 1$) to prevent overshoot and undershoot effect. Our network reconstructs HBD images by multiplying coefficients $\hat{\mathbf{C}}$ with bit-query s as in Eq. (11).

4.2. Training Strategy

We construct a minibatch with uniformly sampled quantization levels from 3-bit to 8-bit. Note that our network shows robustness for unseen levels (2-bit or 10-bit). Let q be a quantization level randomly sampled from 3 to 8 integer. With Eq.(1), we quantize the HBD image to q-bit and add bit-wise zero padding before feeding into a network. We calculate the bit-wise coordinate $s = 2^{N-q}|_{N=16}$ and divide $\mathbf{R}$ by s. We randomly sample pixels from ground truth(GT) ($\mathbf{C}$).

5. Experiment

5.1. Training

Dataset As in [26], we use 2000 16-bit images, each 1000 from the Sintel dataset and the MIT-Adobe FiveK dataset [3] for training. For evaluation, we report peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) on the MIT-Adobe FiveK [3], Sintel datasets [7], benchmarks for TESTIMAGES 1200 [2], Kodak [8], and ESPL v2 [13].

Implementation detail We use 64 × 64 patches for inputs of our network and optimize it by Adam [12]. We use 1000 epochs with a batch size 16. When we train ABCD with CNN-based encoders, such as EDSR [17] or RDN [42], the learning rate is initialized to 1e-4 and decayed by factor of 0.5 at [200, 400, 600, 800]. For a transformer encoder (SwinIR) [16], the learning rate is initialized as 1e-5 and decayed by factor of 0.5 at [500, 800, 900, 950].

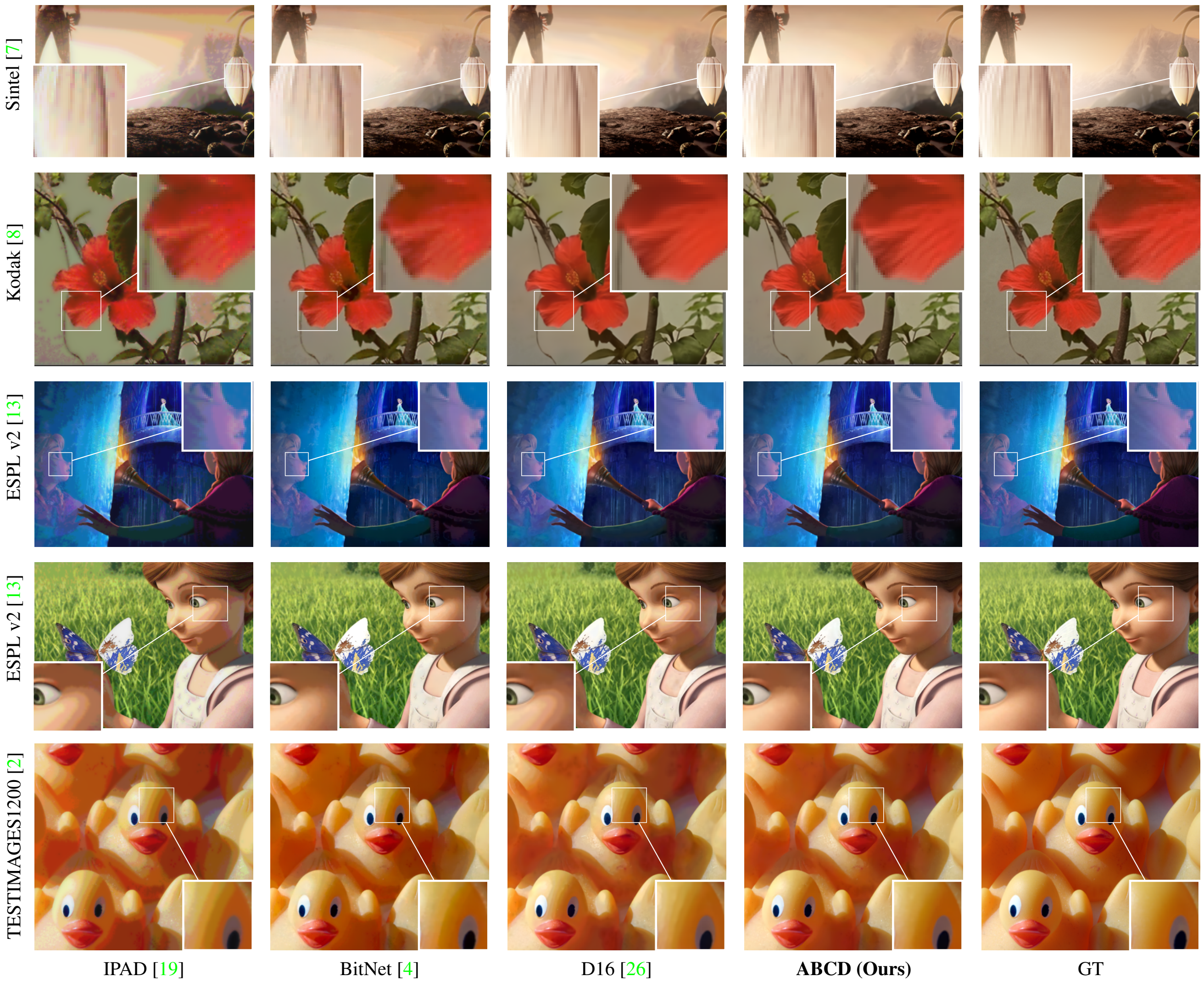


Figure 4. Qualitative comparison in **3-bit → 8-bit BDE**. SwinIR [20] is used as an encoder for ABCD.

5.2. Evaluation

Qualitative result We use 8-bit ground truth images since the standard displays support up to 8-bit. Fig. 4 shows a qualitative comparison to other BDE methods, IPAD [19], BitNet [4], and D16 [26]. We found that D16 [26] suffers from false contour and BitNet [4] blurs details. The first and second rows demonstrate that our ABCD has the advantage of reconstructing details blurred by quantization. Also, overall comparisons show that our ABCD is effective in removing false contour artifacts. Furthermore, in Fig. 5, we demonstrate that our ABCD restores the extreme bitplanes. Note that 2-bit inputs in Fig. 5 are unused sample for training.

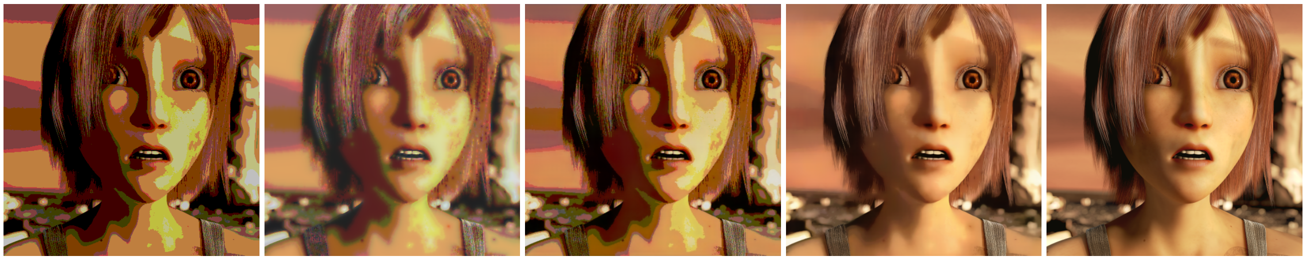
Quantitative result We compare the performance of our method against existing methods; IPAD [19], which is a non-learnable method, BitNet [4], BE-CALF [18], and D16 [26]. The input in Tab. 1 refers to q-quantized images with zero-padding at missing bitplanes. The number of trainable parameters is written below. The pre-trained BitNet supports 3-bit or 4-bit to 8-bit and 3 to 6-bit to 16-bit expansion. The BE-CALF [18] provides a pre-trained model for 4-bit or 8-bit to 16-bit expansion; however, the training code is not available. We directly copy the numeric result from the original papers. In Tab. 1 the test dataset is composed of randomly selected 50 images in the Sintel dataset and the last 1000 (filenames a4001 to 5000) images of the MIT-Adobe dataset enhanced by expert E. Our ABCD outperforms all methods with any encoder. The maximum gain of PSNR is 1.52dB on Sintel for 4→16 BDE.

We report results on benchmarks at the bottom of Tab. 1. The TESTIMAGES dataset [2] contains 40 images with 16-bit depth. The Kodak [8] dataset contains 24 natural images with 8-bit depth. The ESPL v2 [13] contains 25 animated images with 8-bit depth. In both natural and animation images, our ABCD shows better performances than

Test	Sintel						MIT-Adobe FiveK			
Method	$4 \rightarrow 8$	$4 \rightarrow 12$	$4 \rightarrow 16$	$6 \rightarrow 12$	$6 \rightarrow 16$	$8 \rightarrow 16$	$3 \rightarrow 16$	$4 \rightarrow 16$	$5 \rightarrow 16$	$6 \rightarrow 16$
Input	29.16	28.79	28.77	40.90	40.81	52.86	22.90	28.86	34.86	40.88
	0.8864	0.8844	0.8843	0.9858	0.9857	0.9990	0.7381	0.8769	0.9556	0.9871
IPAD [19]	35.86	35.78	35.76	47.66	47.62	58.62	29.86	35.74	41.18	46.43
	0.9457	0.9452	0.9451	0.9903	0.9902	0.9989	0.8624	0.9378	0.9743	0.9903
BitNet [4]	39.34	39.49	39.49	49.72	49.68	57.55	33.46	39.21	44.02	48.46
(0.94M)	0.9701	0.9719	0.9719	0.9954	0.9954	0.9989	0.9128	0.9632	0.9853	0.9943
BE-CALF [18]	39.91	39.98	39.98	51.14	51.14	59.51	-	-	-	-
(5.18M)	0.9737	0.9752	0.9752	0.9940	0.9940	0.9993	-	-	-	-
D16 [26]	41.19	41.51	41.51	53.47	53.48	63.51	34.11	39.95	44.94	49.72
($\leq$15.46M)	0.9794	0.9810	0.9810	0.9980	0.9979	0.9998	0.9279	0.9693	0.9876	0.9953
RDN-ABCD	42.31	42.84	42.84	54.07	54.10	63.75	35.14	40.94	45.68	50.08
(11.52M)	0.9831	0.9847	0.9847	0.9984	0.9984	0.9998	0.9392	0.9746	0.9893	0.9957
EDSR-ABCD	42.47	43.02	43.02	54.15	54.18	63.78	35.25	41.04	45.74	50.11
(12.22M)	0.9837	0.9852	0.9852	0.9984	0.9984	0.9998	0.9401	0.9748	0.9893	0.9957
SwinIR-ABCD	42.51	43.03	43.03	54.08	54.12	63.74	35.44	41.18	45.80	50.13
(12.10M)	0.9844	0.9855	0.9855	0.9984	0.9984	0.9998	0.9412	0.9751	0.9895	0.9957

Benchmark	TESTIMAGES 1200						KODAK		ESPL v2	
Method	$4 \rightarrow 8$	$4 \rightarrow 12$	$4 \rightarrow 16$	$6 \rightarrow 12$	$6 \rightarrow 16$	$8 \rightarrow 16$	$3 \rightarrow 8$	$4 \rightarrow 8$	$3 \rightarrow 8$	$4 \rightarrow 8$
Input	29.21	28.85	28.83	40.95	40.86	52.92	22.77	29.06	23.20	29.28
	0.8764	0.8741	0.8739	0.9856	0.9855	0.9990	0.7671	0.8998	0.6616	0.8261
IPAD [19]	36.29	36.20	36.18	47.20	47.15	57.84	29.20	34.90	29.86	35.75
	0.9450	0.9444	0.9443	0.9901	0.9899	0.9988	0.8515	0.9345	0.8379	0.9207
BitNet [4]	38.75	38.81	38.80	49.52	49.48	53.60	32.68	38.48	32.58	38.23
(0.94M)	0.9571	0.9589	0.9589	0.9944	0.9944	0.9970	0.9172	0.9659	0.9001	0.9399
BE-CALF [18]	38.45	38.50	38.50	49.85	49.84	58.11	-	38.92	-	38.43
(5.18M)	0.9632	0.9648	0.9649	0.9945	0.9945	0.9992	-	0.9681	-	0.9479
D16 [26]	40.39	40.42	40.41	52.12	52.12	61.68	33.67	39.52	33.47	39.53
($\leq$14.27M)	0.9725	0.9735	0.9735	0.9967	0.9967	0.9996	0.9337	0.9723	0.9001	0.9528
RDN-ABCD	40.81	41.36	41.38	52.56	52.59	61.72	34.38	40.11	34.21	40.20
(11.52M)	0.9745	0.9760	0.9761	0.9971	0.9971	0.9996	0.9415	0.9748	0.9093	0.9583
EDSR-ABCD	41.12	41.65	41.65	52.76	52.78	61.78	34.50	40.23	34.36	40.24
(12.22M)	0.9755	0.9770	0.9771	0.9972	0.9972	0.9996	0.9426	0.9753	0.9106	0.9580
SwinIR-ABCD	41.29	41.76	41.77	52.82	52.83	61.78	34.62	40.31	34.55	40.35
(12.10M)	0.9769	0.9779	0.9779	0.9974	0.9974	0.9997	0.9443	0.9762	0.9125	0.9584

Table 1. Quantitative comparisons (PSNR (dB)& SSIM) for **arbitrary bit-depth expansion** on the test set of Sintel [7] & MIT-Adobe FiveK [3](top) and benchmark set including TESTIMAGES 1200 [2], Kodak [8], ESPL v2 [13] (bottom). Red and blue colors indicate the best and the second-best performance, respectively. (-) indicates not reported. (q → N) refers q-bit input to N-bit output BDE.



2-bit Input   IPAD [19]   BitNet [4]   **ABCD (Ours)**   GT

Figure 5. Qualitative comparison for **2-bit → 8-bit BDE**. D16 [26] does not provide 2-bit recovery pre-trained model. SwinIR [20] is used as an encoder for ABCD.

other methods. Our ground truth of training is 16-bit depth images, so 8 and 12-bit target expansion are out-of-range for ABCD. However, thanks to the INR that maps bit-wise coordinates to its coefficient, ABCD achieves the highest PSNR.

5.3. Ablation Study

Network components In Tab. 2, we conducted ablation studies for individual components of ABCD. ABCD contains a phasor estimator in the encoder and bit-wise coordinate concatenation in the decoder. Furthermore, our output is the bit-wise coefficient ($\hat{\mathbf{C}}$) instead of pixel values ($\hat{\mathbf{I}}_{HBD}$ or $\hat{\mathbf{R}}$). To support this, we train EDSR [17]-based ABCD without each component and mechanism. (-P): Encoder without phasor estimator (replaced by ResBlocks), (-S): Decoder without bit-wise query (s), (+B): Mechanism that estimates residual image ($\hat{\mathbf{R}}$), and (+L): Addition of long skip connection of LBD images ($\mathbf{I}_{LBD}$) so that the network predicts the natural image ($\hat{\mathbf{I}}_{HBD}$).

The middle part (-P& -S) of Tab. 2 shows that the phasor estimator and bit-wise query enhance the performance of ABCD, especially in relatively low bit-depth input. The phasor estimator improved performance in a high range. Without the bit-wise query, it makes the output bit-depth to be dedicated (16-bit) and causes performance drops in the network. The ablations of mechanisms ABCD(+B) and ABCD(+L) degrade the performances of the network. For extracting the residual image ($\hat{\mathbf{R}}$), arbitrary residual images

	Input Bit → Output Bit					
	3 → 16	4 → 16	5 → 16	6 → 16	7 → 16	8 → 16
ABCD	**34.57**	**41.65**	**47.67**	**52.78**	**57.24**	**61.78**
ABCD(-P)	34.27	41.25	47.36	52.60	57.16	61.70
ABCD(-S)	34.55	41.58	47.66	52.77	**57.24**	61.76
ABCD(+B)	25.27	31.54	37.70	43.81	49.87	55.92
ABCD(+L)	34.26	41.11	46.87	51.75	56.08	60.52

Table 2. **Quantitative ablation study** of ABCD on TESTIMAGES 1200 [2] (PSNR(dB)). Definitions of -P, -S, +B, +L are shown in Sec. 5.3. EDSR [17] is used as an encoder.

	Input Bit → Output Bit				
	2 → 16	4 → 16	6 → 16	8 → 16	10 → 16
3-8 (Proposed)	**26.55**	**41.65**	**52.78**	**61.78**	**70.98**
Only-2	26.31	34.15	45.56	57.50	69.50
Only-4	20.55	41.16	46.44	57.53	69.40
Only-6	20.34	34.44	52.76	58.80	69.74
Only-8&10			diverged		

Table 3. Ablation study of ABCD for **a dedicated bit-depth expansion.** Evaluated on TESTIMAGES 1200 (PSNR(dB)). Only-k refers to training the model with k-bit quantized images.

have a different range of amplitude which make the network hard to learn.

Fixed-bit training We evaluate the effect of training ABCD for fixed bit depth expansion. The result is shown in Tab. 3. Note that the training range of our ABCD is 3 to 8 bit. So, 2-bit and 10-bit inputs are out-of-distribution. We observed that bit depth expansion for fixed-bit training has poor performance in out-of-distribution bits as well as in-distribution bit-level. Furthermore, high-bit inputs (8 or 10) diverge.

5.4. Phasor Estimation

We demonstrate that our phasor estimator extracts dominant phasors from quantized inputs. For sanity check, we sort the phasor of ground truth ($\mathbf{C}$) in descending order in the absolute value of amplitude and select K phasors. With the formula below, we calculate estimated phases, from Eq. (8):

$$\angle\hat{\mathbf{\Phi}}_{\mathbf{x}} = \tan^{-1}\left(\frac{\boldsymbol{A}^1_{\mathbf{x}}\cos\left(\pi\mathbf{\Phi}^1_{\mathbf{x}}\right) + \boldsymbol{A}^2_{\mathbf{x}}\sin\left(\pi\mathbf{\Phi}^2_{\mathbf{x}}\right)}{\boldsymbol{A}^1_{\mathbf{x}}\sin\left(\pi\mathbf{\Phi}^1_{\mathbf{x}}\right) + \boldsymbol{A}^2_{\mathbf{x}}\cos\left(\pi\mathbf{\Phi}^2_{\mathbf{x}}\right)}\right) \tag{12}$$

We compared the distribution of phasors between GT and the predicted in low-frequency and high-frequency textures of quantized inputs. In Fig. 6, red boxes in the first column are the receptive field (RF) of ABCD, and the second column refers the ground truth of $\hat{\mathbf{C}}$. The diagram in Fig. 6 represents an accumulated number for each angle. Note that the Fourier transform of images is conjugate symmetric. We find that our phasor estimator learns the distribution of the dominant phasors of both high-frequency and low-frequency textures. For example, in the third row of Fig. 6, the density of dominant phasors is accumulated near π (rad), and our ABCD follows those phasors.

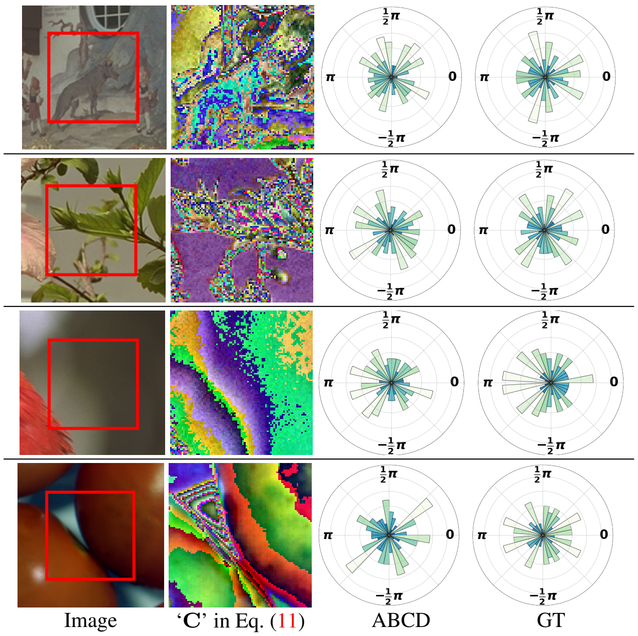


Figure 6. Comparison of the dominant phasor distribution of the ABCD and GT. SwinIR [16] is used as an encoder.

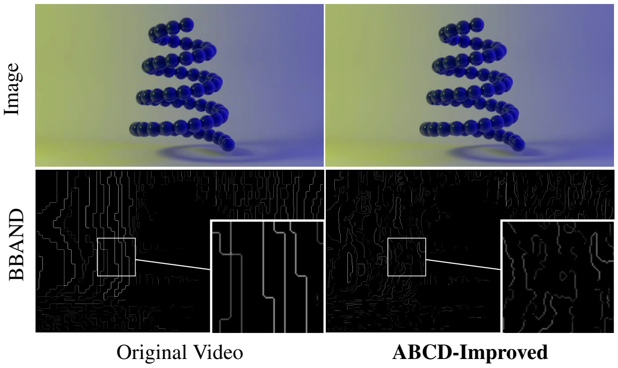


Figure 7. **Blind banding artifact detector (BBAND)** reconstruction of our ABCD. The natural images (top) and BBAND images (bottom).

method \ resolution	360p	480p	720p	1080p	Total
Original Video	0.5525	0.4500	0.4018	0.3606	0.4316
EDSR-ABCD	0.4760	0.4180	0.3476	0.3144	0.3809
Improvement	+13.85%	+7.11%	+13.49%	+12.81%	+11.75%

Table 4. Quantitative enhancement of **BBAND**(↓) metric with ABCD for **YouTube-UGC** [39] dataset.

5.5. Debanding

We validate the effectiveness of our ABCD for debanding in unseen dataset. Inspired by the relations between false contour artifacts and quantization, we hypothesize that our pre-trained ABCD can resolve banding effects. We calculate the blind banding artifact detector (BBAND) score [35] between original and enhanced frames by ABCD. As we have no information about how many bits are required for ABCD to resolve banding effects, we assume that severe artifacts demand large bit depth expansion. We enhance original images with 3→8 BDE results ($\hat{\mathbf{C}}_{3\rightarrow 8}$). The quan-

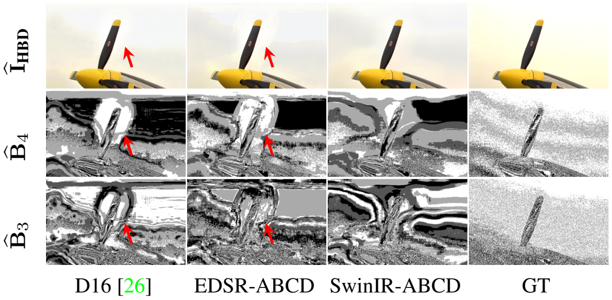


Figure 8. False contour artifacts and analysis in bitplanes for 3-bit→8-bit bit depth expansion. Comparison with D16 [26], EDSR-ABCD and SwinIR-ABCD.

titative result in Tab. 4 shows our ABCD improves BBAND score in YouTube-UGC dataset [39]. The qualitative results are in Fig. 7.

6. Discussion

Phasor Estimator The dependency on the input image sets the phases of LTE [15] and ABCD apart from each other. While LTE infers phases from their scale factor ($\hat{\mathbf{c}} := (2/r_x, 2/r_y)$), ABCD estimates phases from images. Since LTE is a super-resolution network, it requires local coordinates ($\delta := \mathbf{x}_{\text{query}} - \mathbf{x}_{\text{nearest}}$) to learn frequencies. Unlike LTE, outputs of ABCD always have the same resolution with the input images. This implies that local coordinates δ are $\vec{0}$. Thus, the formulation is given as follows:

$$\underbrace{\mathbf{A}_\mathbf{j} \odot \begin{bmatrix} \cos(\pi(\cancel{\mathbf{F}_\mathbf{j} \cdot \delta} + h_p(\hat{\mathbf{c}}))) \\ \sin(\pi(\cancel{\mathbf{F}_\mathbf{j} \cdot \delta} + h_p(\hat{\mathbf{c}}))) \end{bmatrix}_{\delta=\vec{0}}}_{\text{LTE [15]}} \rightarrow \underbrace{\begin{bmatrix} \mathbf{A}^1_\mathbf{x} \\ \mathbf{A}^2_\mathbf{x} \end{bmatrix} \odot \begin{bmatrix} \cos(\pi\mathbf{\Phi}^1_\mathbf{x}) \\ \sin(\pi\mathbf{\Phi}^2_\mathbf{x}) \end{bmatrix}}_{\text{ABCD}}.$$

Artifacts from Encoder In Fig. 8, CNN-based BDE methods have false contour artifacts in restoring a high range of bit-depth. This artifact appears when networks restore high pixel values nearby low-valued pixels. Red arrows indicate such artifacts. Since contour artifacts do not appear on the GT as well as the 3-bit input, it implies that artifacts are caused by the network. We decompose each of the predictions in bit-planes to see behaviors in bit-planes. We overcome these artifacts by using the attention model, SwinIR [16].

FLOPs and Memory In Tab. 5 and Fig. 9, we report the number or training parameters, floating point operations (FLOPs), the memory consumption and the average computation time. We test 4 to 8 bit-depth expansion in TESTIMAGES1200 [2] datasets. D16 [26] requires such parameters because of per-bit processing. For a fair comparison, all methods run on Google Colab. Although the BitNet [4] and BE-CALF [18] have smaller sizes and FLOPs than ours, their results are about 3dB lower than ours. We apply EDSR-baseline [17] and SwinIR-lightweight [16] from their official code and confirm that our framework overcomes the trade-off between computational complexities and performances.

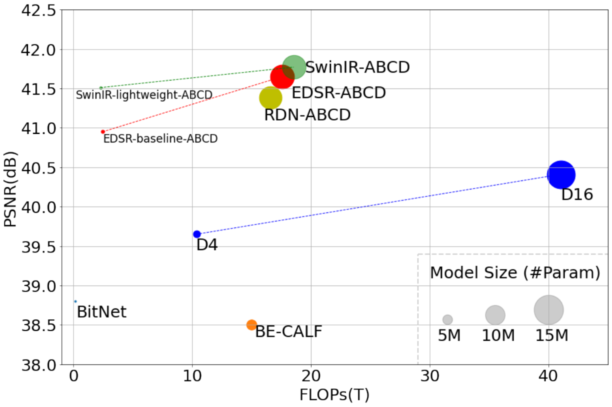


Figure 9. FLOPs and PSNR Comparison with other methods in TESTIMAGES1200 [2] for 4→16 bit-depth expansion. FLOPs are calculated with image size 1200 × 1200.

#Eval /Query	Method	# Params.	Mem. (GB)	Time (s)
1.44M (1200^2px)	BitNet [4]	0.77M	-	6.93
	D16 [26]	14.31M	4.88	14.778
	SwinIR-ABCD (ours)	12.10M	15.32	11.238
	EDSR-ABCD (ours)	12.22M	11.21	7.147
	RDN-ABCD (ours)	11.50M	10.84	8.099

Table 5. Memory consumption (MB) & computation time (s) comparison for an 4→16 BDE task.

7. Conclusion

We proposed an implicit neural network approach as a function of a bit-wise query for BDE. The residual image calculated from bit-wise coefficients recovers the arbitrary depth of missing bit planes with single training. Furthermore, we show that the proposed method removes severe artifacts such as false contour, and blurry artifacts, effectively. Our phasor estimator shows similar phasor diagrams with that of the original image leading to accurate predictions of bit-wise coefficient. The results of test and benchmark datasets demonstrate that our network outperforms state-of-the-art models up to 1.52dB.

Acknowledgement This work was partly supported by the grants of the DGIST R&D program of the Ministry of Science and ICT of KOREA (22-KUJoint-02, 21-IJRP-01), Smart HealthCare Program(www.kipot.or.kr) funded by the Korean National Police Agency(KNPA) (No. 230222M01), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Kenichi Arai and Hiroyuki Okazaki. N-dimensional binary vector spaces. *Formalized Mathematics*, 21(2):75–81, 2013. 3

[2] Nicola Asuni and Andrea Giachetti. Testimages: A large data archive for display and algorithm testing. *Journal of Graphics Tools*, 17(4):113–125, 2013. 4, 5, 6, 7, 8

[3] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 4, 6

[4] Junyoung Byun, Kyujin Shim, and Changick Kim. BitNet: Learning-based bit-depth expansion. In *Asian Conference on Computer Vision*, pages 67–82. Springer, 2018. 1, 2, 5, 6, 8

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning Continuous Image Representation With Local Implicit Image Function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8628–8638, June 2021. 1, 2, 3, 4

[6] Cheuk-Hong Cheng, Oscar C Au, Chun-Hung Liu, and Ka-Yue Yip. Bit-depth expansion by contour region reconstruction. In *2009 IEEE International Symposium on Circuits and Systems*, pages 944–947. IEEE, 2009. 1, 2

[7] Xiph Foundation. "xiph.org". *[Online]. Available: http://www.xiph.org/*. 4, 5, 6

[8] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2), 1999. 4, 5, 6

[9] Xianxu Hou and Guoping Qiu. Image companding and inverse halftoning using deep convolutional neural networks. *arXiv preprint arXiv:1707.00116*, 2017. 1

[10] LIU Chun Hung, CA Oscar, PHW WONG, and MC Kung. Bit-depth expansion by adaptive filter. In *Proc. of IEEE Int. Sym. on Circuits and Systems*, 2008. 1

[11] Chiyu "Max" Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Niessner, and Thomas Funkhouser. Local Implicit Grid Representations for 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[12] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[13] Debarati Kundu, Lark Kwon Choi, Alan C Bovik, and Brian L Evans. Perceptual quality evaluation of synthetic pictures distorted by compression and transmission. *Signal Processing: Image Communication*, 61:54–72, 2018. 4, 5, 6

[14] Jaewon Lee, Kwang Pyo Choi, and Kyong Hwan Jin. Learning local implicit fourier representation for image warping. In *European Conference on Computer Vision (ECCV)*, pages 182–200. Springer, 2022. 4

[15] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1929–1938, 2022. 1, 2, 4, 8

[16] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image Restoration Using Swin Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1833–1844, October 2021. 4, 7, 8

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 4, 6, 7, 8

[18] Jing Liu, Wanning Sun, Yuting Su, Peiguang Jing, and Xiaokang Yang. BE-CALF: bit-depth enhancement by concatenating all level features of dnn. *IEEE Transactions on Image Processing*, 28(10):4926–4940, 2019. 1, 2, 5, 6, 8

[19] Jing Liu, Guangtao Zhai, Anan Liu, Xiaokang Yang, Xibin Zhao, and Chang Wen Chen. IPAD: Intensity potential for adaptive de-quantization. *IEEE Transactions on Image Processing*, 27(10):4860–4872, 2018. 1, 2, 5, 6

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021. 5, 6

[21] Rafal Mantiuk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Perception-motivated high dynamic range video encoding. *ACM Transactions on Graphics (TOG)*, 23(3):733–741, 2004. 1

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[23] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 1, 3

[24] Gaurav Mittal, Vinit Jakhetiya, Sunil Prasad Jaiswal, Oscar C Au, Anil Kumar Tiwari, and Dai Wei. Bit-depth expansion using minimum risk based classification. In *VCIP*, pages 1–5. Citeseer, 2012. 1

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[26] Abhijith Punnappurath and Michael S Brown. A little bit more: Bitplane-wise bit-depth recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2, 4, 5, 6, 8

[27] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine*

Learning Research, pages 5301–5310. PMLR, 09–15 Jun 2019. 1, 2

[28] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. In *ACM SIGGRAPH 2004 Papers*, pages 760–768. ACM New York, NY, USA, 2004. 1

[29] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. 1, 2, 3

[30] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[31] Qing Song, Guan-Ming Su, and Pamela C. Cosman. Efficient debanding filtering for inverse tone mapped high dynamic range videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2575–2589, 2020. 2

[32] Yuting Su, Wanning Sun, Jing Liu, Guangtao Zhai, and Peiguang Jing. Photo-realistic image bit-depth enhancement via residual transposed convolutional neural network. *Neurocomputing*, 347:200–211, 2019. 1, 2

[33] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020. 1, 4

[34] Zhengzhong Tu, Jessie Lin, Yilin Wang, Balu Adsumilli, and Alan C Bovik. Adaptive debanding filter. *IEEE Signal Processing Letters*, 27:1715–1719, 2020. 2

[35] Zhengzhong Tu, Jessie Lin, Yilin Wang, Balu Adsumilli, and Alan C Bovik. BBAND index: a no-reference banding artifact predictor. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2712–2716. IEEE, 2020. 7

[36] Robert A Ulichney and Shiufun Cheung. Pixel bit-depth increase by bit replication. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts III*, volume 3300, pages 232–241. SPIE, 1998. 1, 2

[37] Pengfei Wan, Oscar C Au, Ketan Tang, Yuanfang Guo, and Lu Fang. From 2d extrapolation to 1d interpolation: Content adaptive image bit-depth expansion. In *2012 IEEE International Conference on Multimedia and Expo*, pages 170–175. IEEE, 2012. 1, 2

[38] Pengfei Wan, Gene Cheung, Dinei Florencio, Cha Zhang, and Oscar C Au. Image bit-depth enhancement via maximum a posteriori estimation of ac signal. *IEEE Transactions on Image Processing*, 25(6):2896–2909, 2016. 1

[39] Yilin Wang, Sasi Inguva, and Balu Adsumilli. YouTube UGC dataset for video compression research. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5. IEEE, 2019. 7, 8

[40] Yi Xiao, Chao Pan, Yan Zheng, Xianyi Zhu, Zheng Qin, and Jin Yuan. Gradient-guided dcnn for inverse halftoning and image expanding. In *Asian Conference on Computer Vision*, pages 207–222. Springer, 2018. 2

[41] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022. 1

[42] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual Dense Network for Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[43] Yang Zhao, Ronggang Wang, Wei Jia, Wangmeng Zuo, Xiaoping Liu, and Wen Gao. Deep reconstruction of least significant bits for bit-depth expansion. *IEEE Transactions on Image Processing*, 28(6):2847–2859, 2019. 1