

JPEG Processing Neural Operator for Backward-Compatible Coding

Woo Kyoung Han^{1*}
Sang Hyun Park²

Yongjun Lee^{1*}
Sunghoon Im^{2†}

Byeonghun Lee¹
Kyong Hwan Jin^{1†}

¹Korea University

²DGIST

{wookyoung0727, lyj9805, byeonghun_lee, kyong-jin}@korea.ac.kr, {shpark13135, sunghoonim}@dgist.ac.kr

Abstract

Despite significant advances in learning-based lossy compression algorithms, standardizing codecs remains a critical challenge. In this paper, we present the JPEG Processing Neural Operator (JPNeO), a next-generation JPEG algorithm that maintains full backward compatibility with the current JPEG format. Our JPNeO improves chroma component preservation and enhances reconstruction fidelity compared to existing artifact removal methods by incorporating neural operators in both the encoding and decoding stages. JPNeO achieves practical benefits in terms of reduced memory usage and parameter count. We further validate our hypothesis about the existence of a space with high mutual information through empirical evidence. In summary, the JPNeO functions as a high-performance out-of-the-box image compression pipeline without changing source coding's protocol. Our source code is available at <https://github.com/WooKyoungHan/JPNeO>.

1. Introduction

All compression algorithms depend on how a signal is transformed into a symbol, also known as source coding [14, 38]. With the advancement of Deep Neural Networks (DNNs), lossy image compression has progressed towards more effective transformation [8, 9, 33]. DNN-based approaches utilizing nonlinear transform coding show remarkable performance in compression. Consequently, research on standardization is actively progressing in line with these advancements [5, 6].

However, a significant proportion of images have already been compressed using legacy codecs, and the standardization of DNN-based compression methods is expected to require considerable time. Moreover, among legacy codecs, JPEG [45] compression is widely integrated into image signal processors (ISPs) [34], making it an inevitable part

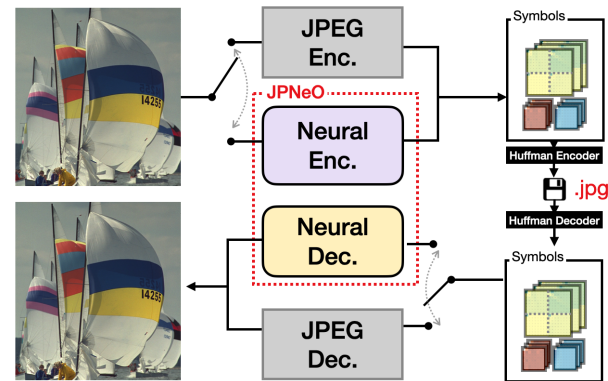


Figure 1. **Overall Concept** of the proposed JPNeO method. Our method enables flexible switching between the conventional JPEG encoder and decoder as needed. Our approach ensures interoperability, allowing conventional JPEG-encoded files to be decoded with a conventional JPEG decoder and existing JPEG files to be decoded using our method.

of the image processing pipeline. Therefore, research on DNN-based approaches for JPEG is actively ongoing with a focus on the compression artifact removal [15, 17, 23] and the exploration encoding images [41, 42]. Several notable studies have explored processing JPEG images without relying on the conventional decoder. Park and Johnson [36] bypass the JPEG decoder and apply the data to classification tasks. Han et al. [22] bypass the JPEG decoder and propose a DNN decoder generating high-quality images.

Building on these advancements, we propose a complete **JPEG Processing Neural Operator (JPNeO)**, DNN-based codec that is fully backward-compatible with the existing JPEG. Fig. 1 presents the overall concept of our JPNeO. Our JPNeO consists of a JPEG encoding and decoding neural operator (JENO and JDNO), allowing flexible replacement with conventional JPEG encoders and decoders. Conventional JPEG decoders rely solely on encoder-constrained information. To overcome this limitation, we integrate a decoder that learns image priors enabling high-quality image reconstruction. To improve the JPEG encoder, we extend existing research [41] to develop an encoder that maps images to a space optimized for JPEG. We hypothesize that a more efficient latent space exists where the latent vectors

*Equal Contribution.

†Corresponding author.

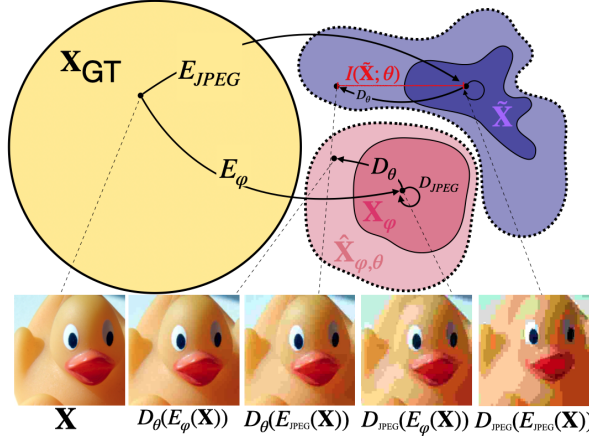


Figure 2. **Illustration of image sets under the JPEG process.** $E_{\varphi/\text{JPEG}}$ and $D_{\theta/\text{JPEG}}$ indicate the neural/JPEG encoder and decoder, respectively. E_{φ} maps images to a range with lower distortion than the E_{JPEG} range. D_{JPEG} is limited to its original range. Our JPNeO’s D_{θ} directs images to a range with lower distortion.

are relatively closer to the ground truth. As demonstrated in Fig. 2, the embedded mutual information of the JPNeO is leveraged for both encoding and decoding high-quality images. We provide experimental validation to support this interpretation. By incorporating a neural operator in the encoding and decoding stages, we propose a novel backward-compatible JPEG processing model that is efficient in parameters and memory usage.

In summary, our main contributions are as follows:

- We propose a learning-based codec, JPNeO, that is fully compatible with the existing JPEG storage format in all cases, ultimately achieving state-of-the-art performance within the JPEG compression framework.
- We provide experimental validation to demonstrate that our encoder and decoder increase the mutual information with the original image at high and low bit rates respectively.
- Our approach is efficient in terms of memory usage and parameter count, and this efficiency is further enhanced by the flexibility to decouple the encoder and decoder.

2. Related Work

Lossy Image Compression Lossy compression is a core technology that reduces file size by discarding information while preserving perceptual quality and methods like JPEG [45] and JPEG2000 [43] have been industry standards based on transform coding. More recently, re-compression techniques—including JPEG XL [3] and the learned lossless JPEG re-compression of Guo et al. [20]—have been proposed to achieve higher efficiency while remaining practical for real-world deployment. Although DNN-based methods with non-linear transform [8, 9, 33] show remarkable performance in lossy compression, the widespread adoption of legacy standards makes it challenging to replace existing

standards and image processing pipelines. Therefore, it is crucial to develop methods that maintain backward compatibility with existing standards. In this paper, we focus on JPEG[45], one of the most widely used compression standards. The proposed JPNeO is designed to achieve full backward compatibility by enabling selective use of either neural or legacy components in both the encoder and decoder, thereby allowing seamless operation with existing JPEG bitstreams.

Deep Learning for JPEG Following the advancement of DNNs, image restoration methods [7, 10, 11, 15–17, 22, 23, 31] have been proposed to mitigate the distortion introduced during the encoding process. JPEG artifact removal DNNs work as auxiliary decoders integrated into the JPEG decoder. Ehrlich et al. [17] utilize a JPEG-aware approach that takes the quantization matrix as a prior and restores luma and chroma components. Jiang et al. [23] proposed a method to restore JPEG images with a blind quality factor, aiming to address the common problem of multiple JPEG compression in real-world scenarios. Recently, Han et al. [22] proposed an approach that decodes images from spectra, bypassing the conventional JPEG decoding process. In addition to decoding strategies, encoding-oriented approaches have also been explored. Talebi et al. [42] proposed a method that pre-edits images for better compression. Strümler et al. [41] suggested a trainable quantization matrix and merge with the pre-editing method. Previous studies have primarily focused on enhancing the encoder or decoder respectively and each module operates independently without sharing information. Building on these advancements, we have developed a fully JPEG-compatible codec that takes advantage of handling mutual information.

Neural Operator Several studies interpret an image as a 2D function, i.e., implicit neural representations (INR) [12, 26, 27, 37, 40, 47]. Additionally, [12, 26, 37] applied INRs to inverse problems. For JPEG, Han et al. [22] interpreted decoding spectra using an INR method. Recently, a novel neural operator method that addresses inverse problems by solving differential equations [21, 25, 28, 32, 46] has shown remarkable performance. The neural operator was designed to solve the differential equation below [25]:

$$(\mathbf{L}_a u)(x) = f(x), \quad x \in D \quad (1)$$

$$u(x) = 0, \quad x \in \partial D \quad (2)$$

where $\mathbf{L} : \mathcal{A} \mapsto \mathbf{L}(U; U^*)$ indicates differential operator and $a \in \mathcal{A}$ is coefficient functions of the PDE. The neural operator $\mathcal{G}_{\theta} : \mathcal{A} \mapsto \mathcal{U} \ni u$ is an empirical solver for Eq. (1). Previous studies [25, 28] have demonstrated that the Galerkin-type attention method is equivalent to the Galerkin method [18] and acts as the solver. We leverage this approach for both JPEG encoding and decoding to enhance our JPEG codec.

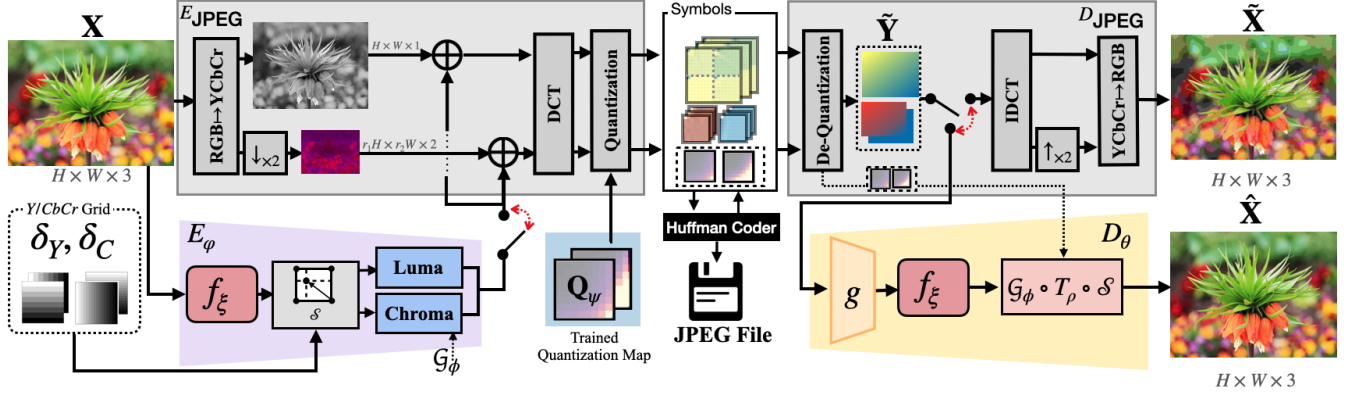


Figure 3. **Schematic overview of our JPNeO.** JPNeO consists of a JPEG encoding neural operator (JENO (E_φ)), a pre-trained quantization matrix (\mathbf{Q}_ψ), and a JPEG decoding neural operator (JDNO (D_θ)). Our JENO, consisting of a feature extractor (f_ξ), sampling (\mathcal{S}), and neural operators (\mathcal{G}_ϕ) functions as an auxiliary encoder to the existing JPEG encoder (E_{JPEG}). JDNO, composed of a group embedding (g), feature extractor f_ξ , and Cosine Neural Operator ($\mathcal{G}_\phi \circ \mathcal{T}_\rho \circ \mathcal{S}$) can replace the existing JPEG decoder (D_{JPEG}). Ultimately, our JPNeO is designed to function properly even if each module is selectively applied.

3. Problem Formulation

Preliminary Let $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ be a ground-truth image. The JPEG [45] contains an encoder ($E_{\text{JPEG}} : \mathbf{X} \mapsto \mathbf{Y}'$) and a decoder ($D_{\text{JPEG}} : \mathbf{Y}' \mapsto \tilde{\mathbf{X}}$). A source coding of the JPEG is demonstrated by the equation below:

$$\mathbf{X} \xrightarrow{\downarrow \times 2} \mathbf{X}' \xrightarrow{\text{DCT}} \mathbf{Y} \xrightarrow{[\cdot/\mathbf{Q}]} \mathbf{Y}' \xrightarrow{\mathbf{Q}} \tilde{\mathbf{Y}} \xrightarrow{\text{DCT}^{-1}} \tilde{\mathbf{X}}, \quad (3)$$

E_{JPEG} above $\mathbf{X} \rightarrow \mathbf{Y}'$ and D_{JPEG} below $\tilde{\mathbf{Y}} \rightarrow \tilde{\mathbf{X}}$

where \mathbf{Y} and \mathbf{Y}' indicate discrete cosine transform (DCT [2]) spectra and symbols, respectively. We notate (\cdot) as a distorted signals and $(\hat{\cdot})$ as predictions. \mathbf{X} is divided into luminance (\mathbf{X}_Y) and chroma (\mathbf{X}_C) components from RGB, and $\downarrow \times 2$ indicates chroma subsampling. The range resolution of the $\downarrow \times 2$ operation is optional between 4:4:4 ($H \times W$), 4:2:2 ($H \times W/2$), and 4:2:0 ($H/2 \times W/2$). The E_{JPEG} quantizes the 8×8 DCT spectrum (\mathbf{Y}) with the predefined quantization matrix ($\mathbf{Q} \in [1, 255]^{8 \times 8}$). The D_{JPEG} decodes \mathbf{Y}' in reverse order of E_{JPEG} . In summary, all JPEG losses originate from quantization and chroma subsampling, highlighted in red in Eq. (3). However, due to the principle of the data processing inequality [14] and the nature of the JPEG, the mutual information $I(\mathbf{X}; \tilde{\mathbf{X}})$ is limited to the E_{JPEG} .

Neural Decoder To address this, the Neural JPEG decoder ($:= D_\theta$) learns an image prior from datasets, thereby embedding the image prior into a trainable parameter θ . From the perspective of mutual information, it follows:

$$I(\mathbf{X}; D_\theta(E_{\text{JPEG}}(\mathbf{X}))) \sim I(\mathbf{X}; \tilde{\mathbf{X}}) + I(\tilde{\mathbf{X}}; \hat{\theta}), \quad (4)$$

$$\hat{\theta} := \arg \max_{\theta} \left[\sum_{x \in \mathbf{X}, \tilde{y} \in \tilde{\mathbf{Y}}} \log p(x | D_\theta(\tilde{y})) + \log p(\theta) \right]. \quad (5)$$

Eq. (4) indicate that it is equivalent to obtaining additional mutual information from the trained parameters by Eq. (5).

Neural Encoder In JPEG encoder, an entropy ($:= \mathcal{H}(\cdot)$) of the spectrum satisfies $\mathcal{H}(\mathbf{Y}) \geq \mathcal{H}(\mathbf{Y}')$, enabling compression. However, the mutual information between an original image and symbols satisfies $I(\mathbf{X}; \mathbf{X}') \geq I(\mathbf{X}; \tilde{\mathbf{X}})$, resulting in a loss for the image. We optimize two objectives: 1) increasing $I(\mathbf{X}; E_\varphi(\mathbf{X}))$ and 2) minimizing $\mathcal{H}(\mathbf{Y}')$ with the Neural JPEG encoder ($:= E_\varphi$).

$$I(\mathbf{X}; D_{\text{JPEG}}(E_\varphi(\mathbf{X}))) \geq I(\mathbf{X}; \tilde{\mathbf{X}}), \quad (6)$$

$$\text{s.t. } \mathcal{H}(\mathbf{X}) \geq \mathcal{H}(E_{\text{JPEG}}(\mathbf{X})) \geq \mathcal{H}(E_\varphi(\mathbf{X})). \quad (7)$$

To satisfy the Eq. (6), we add mutual information in a manner of Eq. (4) such as:

$$I(\mathbf{X}; E_\varphi(\mathbf{X})) \sim I(\mathbf{X}; \mathbf{X}') + I(\mathbf{X}'; \varphi), \quad (8)$$

$$\hat{\varphi} := \arg \max_{\varphi} \left[\sum_{x \in \mathbf{X}} \log p(x | x') + \log p(\varphi) \right]. \quad (9)$$

In Eq. (8), we add mutual information from the trained parameters by Eq. (9). In other words, our method focuses on specifically addressing the errors introduced during chroma subsampling (i.e. $\downarrow \times 2$). The continuous function property of INRs, which benefits image upsampling and subsampling [35], is extended to the neural operator while preserving these advantages.

4. Method

Training Quantization Matrix (\mathbf{Q}_ψ) Inspired from Strümpler et al. [41], our JPNeO uses the pre-trained quantization matrix instead of using the existing quantization matrix. Fig. 4 illustrates training procedure of \mathbf{Q}_ψ . The round operation ($\lfloor \cdot \rfloor$) banishes gradients of the network. Therefore, it is replaced to 3rd order approximation:

$$\lfloor x \rfloor \simeq \lfloor x \rfloor + (\lfloor x \rfloor - x)^3. \quad (10)$$

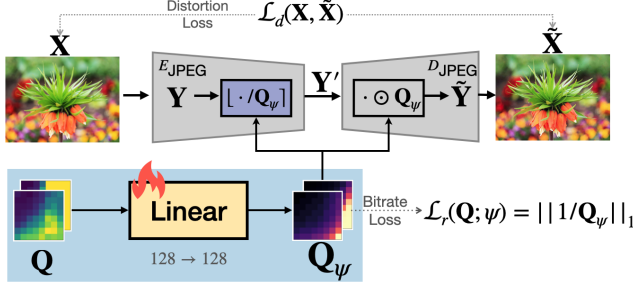


Figure 4. **Overview of learning the quantization matrix.** Only one linear layer is trained. Since quantization is not differentiable, it is replaced with an approximation operation (blue area).

Unlike existing work [41] that uses trainable parameters, we use a standard quantization map (quality factor = 50) as an input of the linear layer which is equivalent to initializing the parameters (i.e. $\mathbf{Q}_\psi = \text{Linear}(\mathbf{Q})$; $\text{Linear} : \mathbb{R}^{128} \mapsto \mathbb{R}^{128}$). The loss function is calculated using the following formula:

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}|\mathbf{Q}_\psi) = \lambda \cdot \mathcal{L}_d(\mathbf{X}, \tilde{\mathbf{X}}) + \mathcal{L}_r(\mathbf{Q}; \psi), \quad (11)$$

$$\mathcal{L}_d(\mathbf{X}, \tilde{\mathbf{X}}) := \|\mathbf{X} - \tilde{\mathbf{X}}\|_2, \quad \mathcal{L}_r(\mathbf{Q}; \psi) := \|\mathbf{1}/\mathbf{Q}_\psi\|_1. \quad (12)$$

The hyperparameter λ controls the ratio of the distortion loss function ($\mathcal{L}_d(\cdot, \cdot)$), and the bitrate loss function ($\mathcal{L}_r(\cdot)$) for the determination of the compression ratio. One model is trained per λ . Given that \mathbf{Q}_ψ is not large, we found that it is effective to store the result \mathbf{Q}_ψ after training and use them as a lookup table.

JPEG Encoding Neural Operator (JENO) We propose a novel JPEG encoding neural operator that addresses chroma subsampling. The training procedure of our JENO is illustrated in Fig. 5. Our JENO ($E_\varphi : (\mathbf{X}, (\delta_Y, \delta_C)) \mapsto (\mathbf{X}_Y, \mathbf{X}_C)$) takes RGB image (\mathbf{X}) and coordinate of Y/CbCr components ($:= \delta_{Y,C} \in \mathbb{R}^{r_1 H \times r_2 W \times 2}$) as inputs. Note that $r_{1,2} \in \{0.5, 1\}$ are determined by the subsampling mode of JPEG. JENO consists of the feature extractor ($f_\xi : \mathbf{X} \mapsto \mathbf{z} \in \mathbb{R}^{H \times W \times K}$), sampling [12, 26] ($\mathcal{S} : (\mathbf{z}, \delta) \mapsto \mathbf{z}_0 \in \mathbb{R}^{H' \times W' \times K'}$) and Galerkin-Attention module (\mathcal{G}_ϕ) [25, 46]. Our JENO utilizes a neural operator as an auxiliary encoder for JPEG such that:

$$\tilde{\mathbf{X}} = \mathcal{G}_\phi(\mathcal{S}(f_\xi(\mathbf{X}), \delta)) + \mathbf{X}', \quad (13)$$

$$\mathcal{S}(\mathbf{z}, \delta) := \{[s_i \cdot \mathbf{z}(\delta_i), \Delta\delta_i]_{i=1}^j, \mathbf{c}\}, \quad (14)$$

where δ_i are the coordinates of the nearest for neighbors of δ , $\Delta\delta = \delta - \delta_i$, $\mathbf{c} = (2/r_1, 2/r_2)$, and $s_i = \Delta\delta_{i,h} \times \Delta\delta_{i,w}$ local area of an image. We optimize a set of trainable parameters $\varphi = \{\xi; \phi\}$ as below:

$$\hat{\varphi} := \arg \min_{\varphi} \|\mathbf{X} - U(\tilde{\mathbf{X}}_\varphi)\|_1, \quad (15)$$

where $U(\cdot)$ indicates a bilinear upsampling. As a result, our JENO learns high-frequency components of \mathbf{X} by rewriting

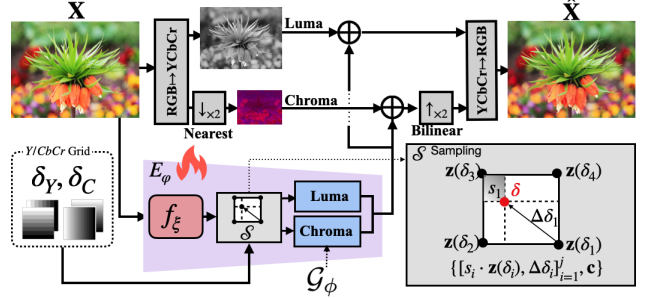


Figure 5. **Schematic flow of training our JENO.** JENO takes \mathbf{X} and coordinates of the \mathbf{X} and downsampled chroma image as input. The predicted image of JENO is produced by a JPEG decoder without the quantization process.

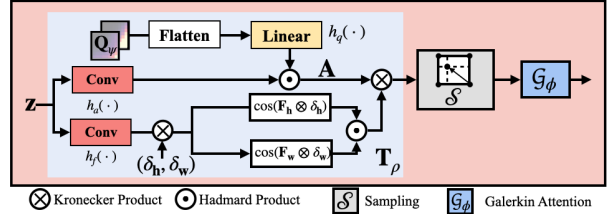


Figure 6. **Schematic flow of Cosine Neural Operator (CNO).**

Eq. (13) as below:

$$\tilde{\mathbf{X}} = U(E_\varphi(\mathbf{X})) + U(\mathbf{X}') \quad (16)$$

$$\simeq HPF(\mathbf{X}) + LPF(\mathbf{X}) \quad (17)$$

$$\therefore U(E_\varphi(\mathbf{X})) \simeq HPF(\mathbf{X}) \quad (\because U(\mathbf{X}') = LPF(\mathbf{X})), \quad (18)$$

where HPF, LPF are a high-pass filter and a low-pass filter, respectively.

JPEG Decoding Neural Operator (JDNO) For decoding symbols, we introduce JDNO ($D_\theta : (\mathbf{Y}_Y, \mathbf{Y}'_C; \mathbf{Q}) \mapsto \tilde{\mathbf{X}}$). Our JDNO is inspired by the JPEG decoding INR proposed by Han et al. [22]; however, JDEC [22] does not support 4:2:2 or 4:4:4 chroma subsampling. JDNO consists of three parts: 1) the group embedding ($g : (\tilde{\mathbf{Y}}_Y, \tilde{\mathbf{Y}}_C) \mapsto \mathbf{z} \in \mathbb{R}^{H/B \times W/B \times 3B^2}$), 2) the feature extractor (f_ξ), and 3) the neural operator with continuous cosine function ($\mathcal{G}_\psi \circ \mathcal{T}_\rho$). g takes chroma and luma spectra as input and embeds them as below:

$$g(\tilde{\mathbf{Y}}_Y, \tilde{\mathbf{Y}}_C) := [\mathcal{D}_B(\mathcal{D}_8^{-1}(\tilde{\mathbf{Y}}_Y)), \mathcal{D}_B(\mathcal{D}_8^{-1}(U' \cdot \tilde{\mathbf{Y}}_C))] \quad (19)$$

where \mathcal{D}_N indicates 2D-DCT with size of $N \times N$ and U' is an impulse response of U . In $f_\xi : \mathbf{z} \mapsto \mathbf{z}' \in \mathbb{R}^{H/B \times W/B \times 3B^2}$, feature is extracted as in JENO. We utilize continuous cosine formulation [22] into Cosine Neural Operator (CNO) which comprises three modules: $h_f : \mathbb{R}^K \mapsto \mathbb{R}^{2M}$, coefficient estimator $h_a : \mathbb{R}^K \mapsto \mathbb{R}^M$, and quantization matrix encoder $h_q : \mathbb{R}^{128} \mapsto \mathbb{R}^M$. Then, CNO formulate the latent vector \mathbf{z}' as below:

$$\mathcal{T}_\rho(\mathbf{z}', \delta; \mathbf{Q}) = \mathbf{A} \otimes (\cos(\pi \mathbf{F}_h \otimes \delta_h) \odot \cos(\pi \mathbf{F}_w \otimes \delta_w)) \quad (20)$$

$$\text{where } \mathbf{A} = h_q(\mathbf{Q}) \odot h_a(\mathbf{z}'), \quad \mathbf{F} = h_f(\mathbf{z}'). \quad (21)$$

\odot, \otimes denote Hadamard and Kronecker product, respectively. Fig. 6 shows overall flow of our CNO. $\tilde{\mathbf{X}}$ is com-

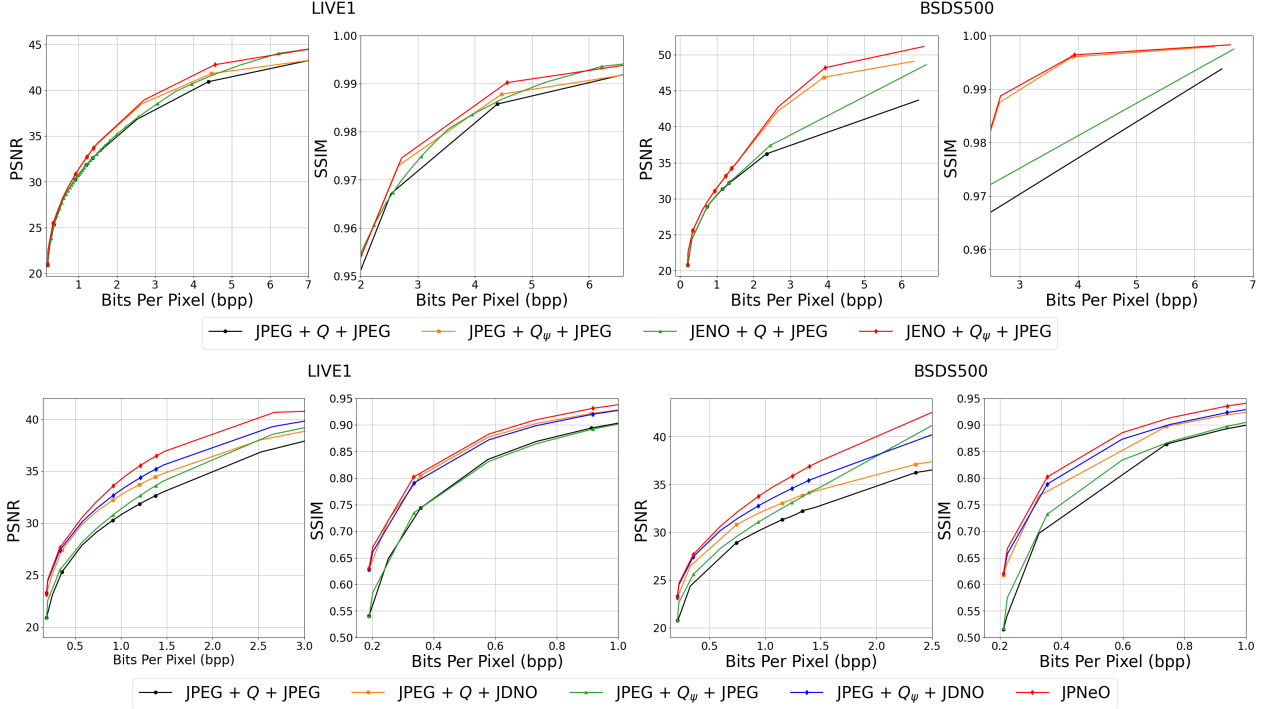


Figure 7. Quantitative ablation study on our JENO (top), JDNO (bottom), and \mathbf{Q}_ψ (PSNR(↑)/SSIM(↑)). The label indicates ‘Encoder ($E_\varphi/E_{\text{JPEG}}$)’+‘ $\mathbf{Q}/\mathbf{Q}_\psi$ ’+Decoder (D_θ/D_{JPEG}). The RD curves were measured on validation set of LIVE1 [39] and BSDS500 [4] dataset. The 4:2:0 chroma downsampling was used.

pleted by decoding $\mathbf{z}_0 = \mathbf{T}\rho(\mathbf{z}', \delta; \mathbf{Q})$ JENO’s neural operator process (Eq. (13)) i.e:

$$\hat{\mathbf{X}} = \mathcal{G}_\phi(\mathcal{S} \circ \mathbf{T}\rho(f_\xi(\mathbf{X}), \delta)). \quad (22)$$

The trainable parameters of JDNO $\Theta := \{\xi; \rho; \phi\}$ is optimized with the equation below:

$$\hat{\Theta} = \arg \min_{\Theta} \|\mathbf{X} - \hat{\mathbf{X}}_\Theta\|_1. \quad (23)$$

5. Experiments

5.1. Network Implementation Details

Quantization matrix (\mathbf{Q}_ψ) Only one linear layer is trained for \mathbf{Q}_ψ , and it is not used after trained. We apply a scaled sigmoid layer ($:= 254 \cdot \text{sig}(\cdot) + 1$) to limit the output value. After training, \mathbf{Q}_ψ is stored as an integer [1,255]. We set λ by sampling the values from $[1e^{-4}, 1e^6]$. As a result, we save 17 different \mathbf{Q}_ψ , excluding duplicate values.

JENO (E_φ) We use the EDSR-baseline [30] which has 16 residual blocks with 64 ($= K$) channels for feature extractor (f_ξ) of our JENO. Therefore, the latent vector for the neural operator contains 266 ($= K'$) channels. Following the configuration of [46], \mathcal{G}_ϕ comprises 256 channels with 16 heads, and two iterations for the kernel integral.

JDNO (D_θ) Following the previous decoding method [22], we adopt the SwinV2 attention module [29] with 256 channels ($= K$) as feature extractor (f_ξ). For group-embedding

($g(\cdot, \cdot)$), we set the block size as 4 ($= B$). In JDNO, the cosine features ($T_\rho(\mathbf{z})$) contain 128 channels ($= M$), resulting in a total of 512 input channels for the neural operator (\mathcal{G}_ϕ). h_f and h_a consist of a single 3×3 convolutional layer, while h_q is a single linear layer. \mathcal{G}_ϕ performs two iterations with 16 heads, following the same configuration as in JENO.

5.2. Training

Dataset For training JPNeO, we use DIV2K and Flickr2K [1] following the previous work [17, 22, 23] which contains 800 and 2,650 images, respectively. We crop images with size of 112×112 which is the least common multiple of JPEG ($= 16$ by considering subsampling) and the window size of Swin architecture ($= 7$) [13, 22, 36]. For JENO, we randomly choose a chroma subsampling mode between 4:2:0 and 4:2:2. For JDNO, we combine standard quantization matrix \mathbf{Q} with steps of 10 in the range [10,90] and pre-trained $[\mathbf{Q}_\psi]$ and randomly sample one.

Implementation Detail We use 4 GPUs (NVIDIA RTX 3090 24GB) for training. Our JPNeO is optimized by Adam [24] for 1,000 epochs with batch size 64. The learning rate is initialized as $1e-4$ and decayed by factor 0.5 at [200, 400, 600, 800].

5.3. Evaluation

Module-Wise Evaluation We conduct an evaluation on each part of JPNeO: JENO and JDNO. We use LIVE1[39] and the test set of BSDS500 [4]. In Fig. 7, we fix the en-

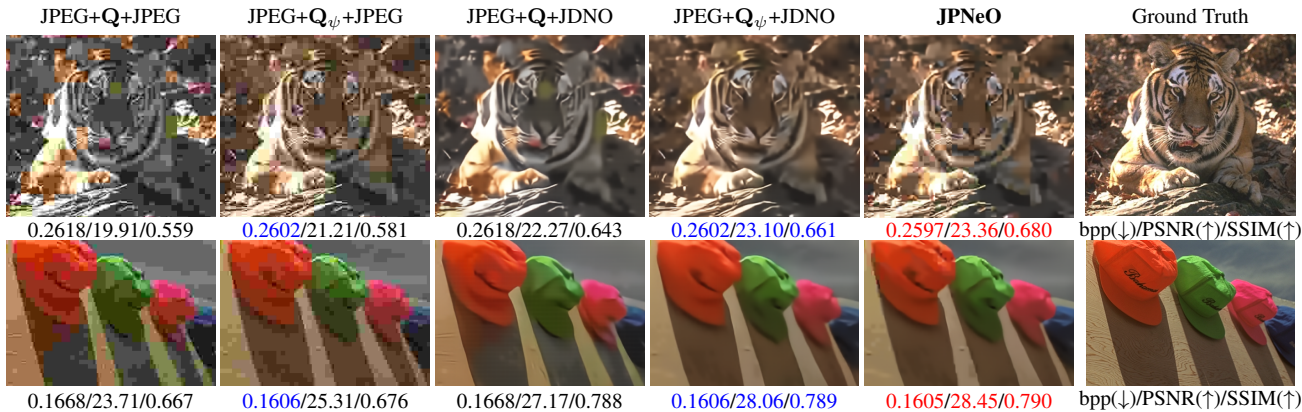


Figure 8. Qualitative ablation study in highly compressed images based on the quantization map (Q/Q_ψ) and decoder (JPEG/JDNO). Red and blue colors indicate the best and the second-best performance, respectively.

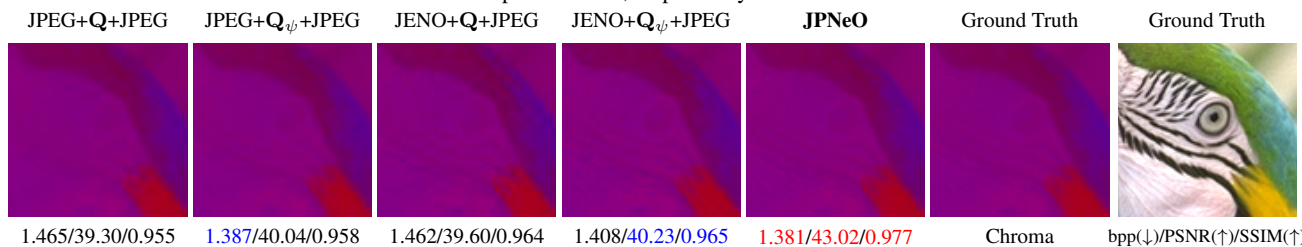


Figure 9. Qualitative comparison of chroma components in high-bpp images based on the encoder (JENO) and the quantization matrix (Q/Q_ψ). Red and blue colors indicate the best and the second-best performance, respectively.

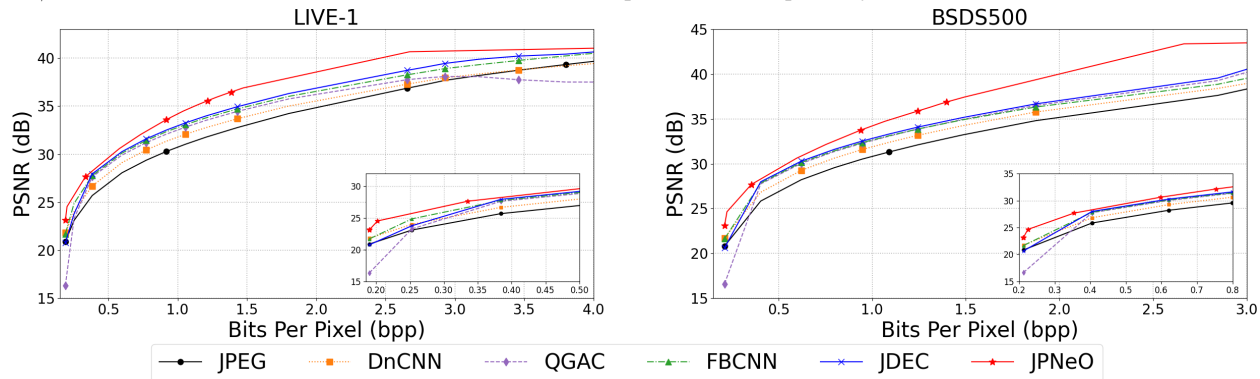


Figure 10. Rate-Distortion curve comparisons with the *JPEG artifact removal networks* on LIVE-1 [39] (left) and BSDS500 [4] (right). We highlight the highly compressed parts in the bottom right part of each graph. We show PSNR as a measure of *distortion* (higher is better).

coder or decoder as $(E_{\text{JPEG}}, D_{\text{JPEG}})$ and measure the performance of our JENO and JDNO, respectively. The performance gain is more pronounced in JDNO at low bpp, whereas JENO shows greater improvement at high bpp. In other words, JENO enhances the upper bound of image quality, while JDNO raises the lower bound, and Q, Q_ψ determines the path between them. To demonstrate this, we highlight the high bpp region for JENO and the low bpp region for JDNO, and present the corresponding SSIM results in Fig. 7. In Fig. 8 we demonstrate the qualitative analysis for our JPNeO. The performance of JDNO depends on the quality of the encoded image. Using both JENO and JDNO reduces both file size and distortion. In Fig. 9, a

qualitative comparison of chroma components at high bpp demonstrates that JENO is robust to chroma subsampling. Regardless of Q , JENO preserves chroma components better than the conventional JPEG encoder, achieving higher performance at lower bpp.

Quantitative Result To validate our JPNeO, we compare JPNeO with existing compression artifact removal models: DnCNN [48], QGAC [17], FBCNN [23], and JDEC [22]. JDEC [22] is designed to directly decode spectra into JPEG images, whereas the others act as auxiliary decoders for D_{JPEG} covering a wide range of quality factors with a single model. For fair comparison, we evaluate JPNeO using the standard quantization matrix Q instead of Q_ψ and 4:2:0

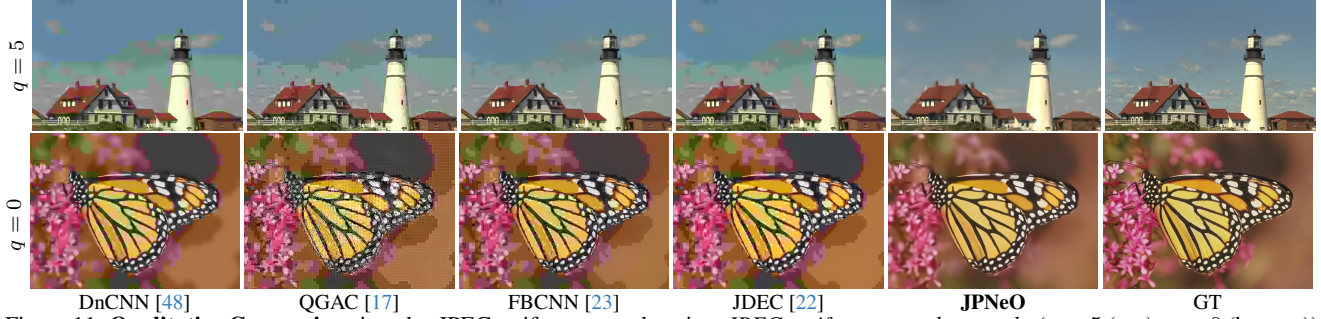


Figure 11. **Qualitative Comparison** in color JPEG artifact removal against *JPEG artifact removal networks* ($q = 5$ (top), $q = 0$ (bottom)).

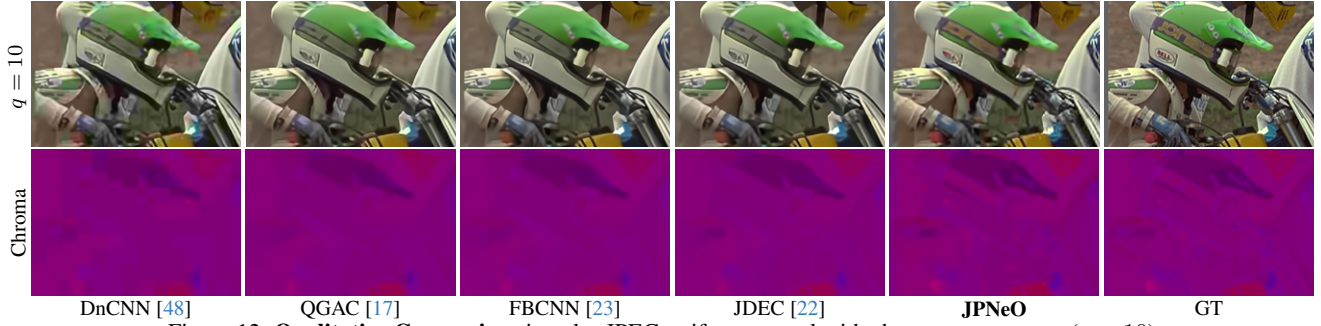


Figure 12. **Qualitative Comparison** in color JPEG artifact removal with chroma components ($q = 10$).

Methods	JPEG[45]	QGAC[17]	FBCNN[23]	JDEC[22]	JPNeO	
Dataset	q	#Params.	259.4M	70.1M	38.9M	29.7M
LIVE-1 [39]	0	20.89 19.73	16.33 15.99	21.70 21.19	20.76 20.07	23.15 22.64
		0.540	0.318	0.577	0.542	0.631
	10	25.69 24.20	27.65 27.43	27.77 27.51	27.95 27.71	28.15 27.55
		0.759	0.819	0.816	0.821	0.829
	20	28.06 26.49	29.88 29.56	30.11 29.70	30.26 29.87	30.49 29.69
		0.841	0.882	0.881	0.885	0.890
	30	29.37 27.84	31.17 30.77	31.43 30.92	31.59 31.12	31.83 30.94
		0.875	0.908	0.908	0.911	0.915
	40	30.28 28.84	32.08 31.64	32.34 31.80	32.50 31.98	32.83 31.91
		0.894	0.922	0.923	0.925	0.929
BSDS500 [4]	0	20.82 19.57	16.61 16.17	21.63 21.08	20.63 19.89	23.10 22.58
		0.517	0.314	0.558	0.514	0.618
	10	25.84 24.13	27.75 27.48	27.85 27.53	28.00 27.67	28.06 27.29
		0.759	0.819	0.814	0.819	0.824
	20	28.21 26.37	30.04 29.55	30.14 29.58	30.31 29.71	30.36 29.31
		0.844	0.884	0.881	0.885	0.887
	30	29.57 27.72	31.36 30.73	31.45 30.74	31.65 30.88	31.70 30.49
		0.880	0.911	0.909	0.912	0.914
	40	30.52 28.69	32.29 31.53	32.36 31.54	32.53 31.68	32.70 31.40
		0.900	0.926	0.924	0.927	0.929

Table 1. Quantitative comparisons (PSNR (dB) | PSNR-B (dB) (top), SSIM (bottom)) with the *JPEG artifact removal networks*. Note that the JPEG artifact removal networks are auxiliary decoder of the JPEG. We highlight the best and the second-best performance with red and blue colors, respectively.

chroma subsampling. Consistent with previous works, we use LIVE-1 [39] and the test set from BSDS500 [4]. As shown in Fig. 10, JPNeO achieves superior performance compared to all other methods, even at extremely low bpp (≤ 0.5).

For a numerical comparison, we report PSNR, structural similarity index (SSIM), and PSNR-B against baselines with the number of parameters in Tab. 1. We compare the existing artifact removal networks with quality factors $\{0, 10, 20, 30, 40\}$ for both datasets. As demonstrated in Tab. 1, JPNeO demonstrates superior performance in terms

Methods	DnCNN [48]	QGAC [17]	FBCNN [23]	JDEC [22]	JPNeO	
Dataset	q					
LIVE-1 [39]	0	23.80 29.27	16.76 29.61	23.23 29.86	22.89 28.75	24.20 32.30
		0.304 0.044	0.135 0.039	0.286 0.068	0.270 0.054	0.342 0.125
	10	28.39 34.47	28.70 37.02	28.76 37.35	28.83 37.95	28.77 38.56
		0.582 0.101	0.602 0.206	0.600 0.255	0.604 0.274	0.616 0.291
	20	30.77 37.15	31.01 39.01	31.13 39.64	31.16 40.25	30.88 41.52
		0.697 0.175	0.708 0.281	0.707 0.330	0.711 0.355	0.713 0.391
	30	32.14 38.27	32.40 40.20	32.53 40.63	32.55 41.31	32.18 42.72
		0.748 0.221	0.756 0.324	0.756 0.372	0.758 0.402	0.759 0.452
	40	33.14 38.98	33.37 40.79	33.53 41.23	33.53 41.92	33.18 43.47
		0.776 0.256	0.782 0.351	0.783 0.399	0.785 0.431	0.785 0.494
BSDS500 [4]	0	23.22 28.42	16.72 28.78	22.70 29.18	21.96 27.84	23.72 31.79
		0.306 0.058	0.141 0.045	0.294 0.091	0.271 0.035	0.360 0.164
	10	27.50 34.91	27.72 37.70	27.75 37.81	27.78 38.42	27.89 39.01
		0.582 0.101	0.636 0.264	0.630 0.302	0.634 0.321	0.649 0.138
	20	29.72 37.99	29.92 40.36	29.98 40.41	30.00 41.02	29.95 42.64
		0.740 0.218	0.753 0.346	0.751 0.376	0.755 0.407	0.758 0.448
	30	31.12 39.30	31.31 41.36	31.39 41.39	31.41 42.15	31.27 44.11
		0.797 0.270	0.806 0.389	0.805 0.409	0.808 0.444	0.809 0.508
	40	32.14 40.23	32.32 42.19	32.41 42.06	32.43 42.81	32.32 45.20
		0.829 0.315	0.836 0.422	0.836 0.436	0.837 0.472	0.839 0.567

Table 2. Quantitative comparisons of luminance and chroma components. (X_Y -PSNR| X_C -PSNR (top), X_Y -SSIM| X_C -SSIM (bottom))

of PSNR and SSIM when compared to existing networks. Notably, the performance of the proposed model is particularly pronounced in quality factor 0. We report PSNR and SSIM of each luminance and chroma component in Tab. 2 for a detailed comparison. The results demonstrate that the enhanced performance of JPNeO is a consequence of its ability to restore chroma components more effectively than other methods.

Qualitative Result We show qualitative results against JPEG artifact removal networks with low quality factors $\{0, 5\}$ in Fig. 11. While other networks do not fully remove JPEG artifacts with low-quality factors, our JPNeO significantly removes the artifacts such as the monarch of the sec-

PSNR (dB) Method	#Params. (M)	Mem. (GB)	Time (ms)	MACs (K/Pixel)	LIVE1 [39]		BSDS500 [4]	
					$q = 0$	$q = 40$	$q = 0$	$q = 40$
FBCNN [23]	70.1	0.61	71.95	1131	21.70	32.34	21.63	32.36
JDEC [22]	38.9	1.76	224.79	1605	20.76	32.50	20.63	32.53
JPNeO ⁻	8.0	0.09	222.95	1045	22.98	32.72	22.73	32.63
JPNeO	29.7	0.26	562.42	3489	23.15	32.83	23.10	32.70

Table 3. Quantitative comparison of computational costs & decoding performance.

	JENO (E_φ)			JDNO (D_θ)				JPNeO
	f_ξ	\mathcal{G}_ϕ	Total	f_ξ	T_ρ	\mathcal{G}_ϕ	Total	
#Params. (M)	1.220	1.663	2.883	22.72	0.902	3.16	26.78	29.663
MACs (K/Pixel)	609	541	1150	732	10	1664	2339	3489
Time (ms) ¹	177.85			398.8				562.42
Mem. (MB)	26.85			189.81				246.62

Table 4. Module-wise resource consumption (parameters, MACs, and time& memory consumption) within the encoder and decoder.

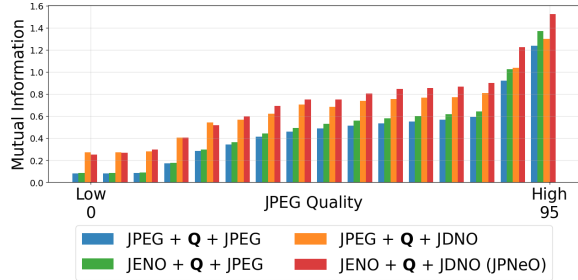


Figure 13. **Mutual information** with the ground-truth image ($I(\mathbf{X}; \tilde{\mathbf{X}})$) across quality levels in Kodak [19] (4:2:0 subsampling). The label indicates ‘Encoder ($E_\varphi/E_{\text{JPEG}}$) + ‘Q/Q_ψ’ + Decoder (D_θ/D_{JPEG})’

ond row of Fig. 11 maintaining the color tones. In Fig. 12, the comparison shows qualitative results in the RGB and chroma domains with quality factor 10. While the results in the RGB domain show a slight perceptual difference, the results in the chroma components show that JPNeO restores more fine details than the other networks.

6. Discussion

Computational Costs In Tabs. 3 and 4, we report computational cost and performance, including network parameters, time and memory consumption, and Multiply–Accumulates (MACs) per pixel. We further includes a lightweight variant (JPNeO⁻) evaluated under similar MAC/pixel constraints for a fair comparison. The size of the input for the comparison is 560×560 . Notably, our JPNeO and JPNeO⁻ surpass the performance of FBCNN[23] and JDEC[22] while requiring only a minimal amount of resources which is a significant outcome considering the adoption of JPEG across ISP pipelines. In Tab. 4, we decompose each model to compare parameter counts and MACs at the module level.

Mutual Information In Sec. 3 and Eqs. (4) and (6), the mutual information between the prior embedded in the MAP-trained parameters θ and $\tilde{\mathbf{X}}$ provides advantages for image restoration. Similarly, the mutual information between $I(\mathbf{X}'; \varphi)$ enhances encoding performance. To validate this,

¹Per-module measurement are excluded due to GPU kernel overlap.

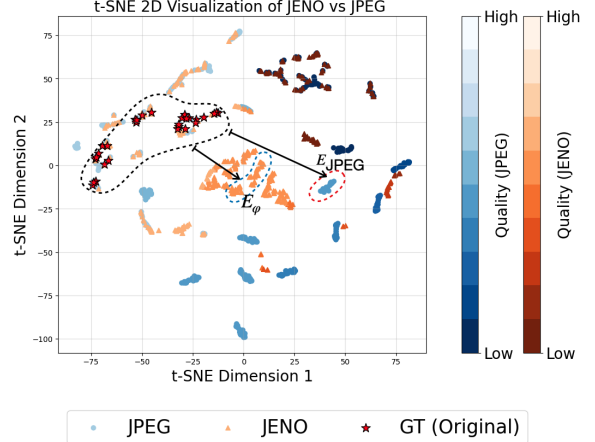


Figure 14. **t-SNE [44] clustering analysis** by quality factor on the LIVE-1 [39]. Encoded images with our JENO (E_φ) (reds) and JPEG encoder (E_{JPEG}) (blues).

we calculate the mutual information using the symbol statistics from the Kodak dataset [19] while switching the encoder and decoder between ($E_\varphi/E_{\text{JPEG}} + D_\theta/D_{\text{JPEG}}$). In Fig. 13, key observation is that as the quality increases, JENO’s mutual information rises, eventually exceeding the mutual information gain achieved by JDNO. However, at low quality, JDNO achieves a larger increase in mutual information. In conclusion, using only JENO at high bpp and only JDNO at low bpp is resource-efficient, while using both at intermediate quality levels is preferable.

t-SNE Clustering We compare the latent distances when using JENO for compression with those obtained from standard JPEG to support Fig. 2 and our hypothesis in Sec. 1. Fig. 14 presents the clustering results after extracting latent vectors using a quality predictor of FBCNN [23]. The distance between GT region and the range of E_φ is less than the distance between GT region and the range of E_{JPEG} . Fig. 14 show that JPEG compression produces significantly larger distances between latent vectors at the same quality level.

7. Conclusion

We propose JPNeO, a learning-based codec that remains backward-compatibility with the existing JPEG format while enhancing compression efficiency and reconstruction quality. By applying neural operators in both encoding and decoding and training a new quantization matrix, we significantly reduce quantization loss. Experimental results show that JPNeO successfully restores color components and achieves better PSNR and SSIM compared to existing approaches. The memory-friendly design and flexible architecture allow seamless integration with legacy JPEG systems. Moreover, JPNeO leverages learned image priors to maintain high-fidelity details at both low and high bit rates. Thus, it shows strong potential for next-generation JPEG pipelines.

Acknowledgments

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00335741, RS-2024-00413303) and Smart Health-Care for Police Officers Program through the Korea Institutes of Police Technology(KIPoT) funded by the Korean National Police Agency (RS-2022-PT000186).

References

- [1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. 5
- [2] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1): 90–93, 1974. 3
- [3] Jyrki Alakuijala, Ruud Van Asseldonk, Sami Boukourt, Martin Bruse, Iulia-Maria Comşa, Moritz Firsching, Thomas Fischbacher, Evgenii Kliuchnikov, Sebastian Gomez, Robert Obryk, et al. Jpeg xl next-generation image compression architecture and coding tools. In *Applications of digital image processing XLII*, pages 112–124. SPIE, 2019. 2
- [4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 5, 6, 7, 8, 3
- [5] João Ascenso and Evgeniy Upenik. White paper on jpeg ai scope and framework. *ISO/IEC JTC*, 1, 2021. 1
- [6] Joao Ascenso, Pinar Akyazi, Fernando Pereira, and Touradj Ebrahimi. Learning-based image coding: early solutions reviewing and subjective quality evaluation. In *Optics, Photonics and Digital Technologies for Imaging Applications VI*, pages 164–176. SPIE, 2020. 1
- [7] Yuval Bahat and Tomer Michaeli. What’s in the image? exploratory decoding of compressed images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2908–2917, 2021. 2
- [8] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1, 2
- [9] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 1, 2
- [10] Lukas Cavigelli, Pascal Hager, and Luca Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 752–759. IEEE, 2017. 2
- [11] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017. 2
- [12] 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, 2021. 2, 4
- [13] Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In *European Conference on Computer Vision*, pages 669–687. Springer, 2022. 5
- [14] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 1, 3
- [15] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE international conference on computer vision*, pages 576–584, 2015. 1, 2
- [16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [17] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 293–309. Springer, 2020. 1, 2, 5, 6, 7
- [18] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*. Springer, 2004. 2
- [19] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k.us/graphics/kodak*, 4(2), 1999. 8
- [20] Lina Guo, Xinjie Shi, Dailan He, Yuanyuan Wang, Rui Ma, Hongwei Qin, and Yan Wang. Practical learned lossless jpeg recompression with multi-level cross-channel entropy model in the dct domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5862–5871, 2022. 2
- [21] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021. 2
- [22] Woo Kyoung Han, S. Im, J. Kim, and Kyong Hwan Jin. JDEC: Jpeg decoding via enhanced continuous cosine coefficients. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 4, 5, 6, 7, 8
- [23] Jiayi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind jpeg artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4997–5006, 2021. 1, 2, 5, 6, 7, 8, 3
- [24] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5
- [25] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Aziz-zadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. 2, 4
- [26] 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. 2, 4

- [27] 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. 2
- [28] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020. 2
- [29] Jingyun Liang, Jiezhong 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, 2021. 5
- [30] 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*, 2017. 5, 3
- [31] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 2
- [32] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021. 2
- [33] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 1, 2
- [34] Junichi Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2017. 1
- [35] Zhihong Pan, Baopu Li, Dongliang He, Mingde Yao, Wenhao Wu, Tianwei Lin, Xin Li, and Errui Ding. Towards bidirectional arbitrary image rescaling: Joint optimization and cycle idempotence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17389–17398, 2022. 3
- [36] Jeongsoo Park and Justin Johnson. Rgb no more: Minimally-decoded jpeg vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22334–22346, 2023. 1, 5
- [37] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 2
- [38] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. 1
- [39] Hamid R Sheikh, Muhammad F Sabir, and Alan C Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on image processing*, 15(11):3440–3451, 2006. 5, 6, 7, 8, 2, 3
- [40] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, pages 7462–7473. Curran Associates, Inc., 2020. 2
- [41] Yannick Strümpfer, Ren Yang, and Radu Timofte. Learning to improve image compression without changing the standard decoder. In *European Conference on Computer Vision*, pages 200–216. Springer, 2020. 1, 2, 3, 4
- [42] Hossein Talebi, Damien Kelly, Xiyang Luo, Ignacio Garcia Dorado, Feng Yang, Peyman Milanfar, and Michael Elad. Better compression with deep pre-editing. *IEEE Transactions on Image Processing*, 30:6673–6685, 2021. 1, 2
- [43] D Taubman. Jpeg 2000: Image compression fundamentals, standards and practice, 2002. 2
- [44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8
- [45] G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xxviii–xxxiv, 1992. 1, 2, 3, 7
- [46] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18247–18256, 2023. 2, 4, 5
- [47] Jingyu Yang, Sheng Shen, Huanjing Yue, and Kun Li. Implicit Transformer Network for Screen Content Image Continuous Super-Resolution. In *Advances in Neural Information Processing Systems*, 2021. 2
- [48] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 6, 7