JPEG Processing Neural Operator for Backward-Compatible Coding

Supplementary Material

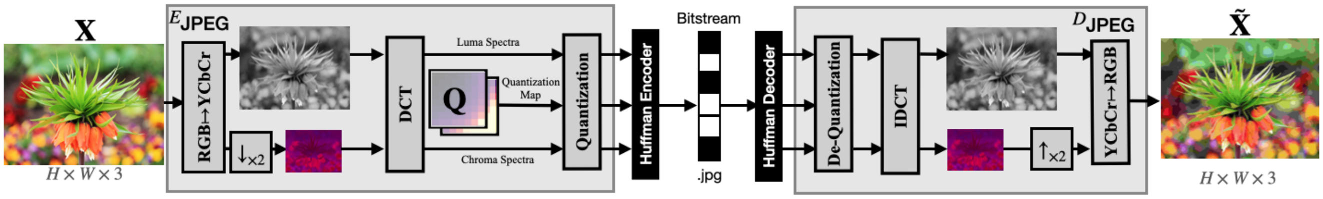


Figure S1. Schematic flow of the standard JPEG compression.

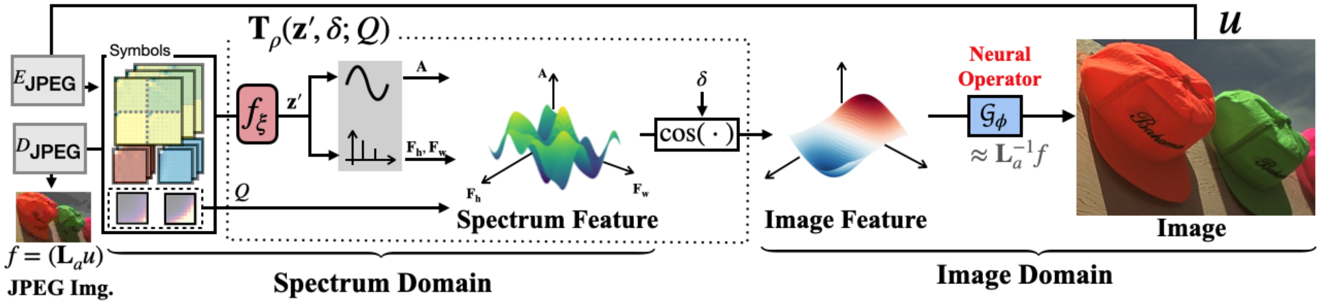


Figure S2. Visual summary of the proposed JPEG decoding. The loss incurred during the JPEG pipeline is modeled as Eq. (1) of the main paper.

1. Introduction

This supplementary material is intended to support the main paper. We provide a notation table in Tab. S1 to clarify the methodology and problem formulation. Additional background of JPEG compression codec is provided to define the problem and illustrate how the decoder works with a schematic overflow in Fig. S2. Further experimental results are included.

2. JPEG Preliminary

In this section, we build upon Sec. 3 of the main paper and elaborate on the fundamentals of JPEG compression. We then characterize the compression artifacts of JPEG and provide further details of our proposed method.

Standard JPEG compression Fig. S1 hows the process of encoding and decoding an RGB image through the standard JPEG pipeline. During encoding, the JPEG algorithm decomposes an RGB image into its luminance and chroma components. The chroma components undergo a subsampling process using the nearest neighbor method. The chroma subsampling procedure is optional, with three distinct methods: reducing both dimensions by half (4:2:0), halving only the width (4:2:2), or maintaining both dimensions unchanged (4:4:4). Subsequently, 8×8 2D-DCT transformed spectra are quantized using a predefined quantization matrix. During decoding, the stored quantization map ($\mathbf{Q}$) is multiplied with the quantized coefficients ($\mathbf{Y}'$) to recover the spectrum ($\tilde{\mathbf{Y}}$). A 2D-IDCT is then applied, followed by resizing the chroma channels to the luminance resolution. The result is transformed back to the RGB domain. Each symbol is encoded into a bitstream via Huffman coding.

A backward-compatible neural codec with JPEG, including our JPNeO, is subject to several constraints within this process. Most notably, JENO is applied as a pre-processing step before quantization. This constraint ensures that images compressed by JENO remain decodable into valid outputs using the standard JPEG decoder. As noted by Han et al. [22], JDNO operates directly on spectral representations and be capable of reconstructing high-frequency components. To support arbitrary resizing of chroma components, a coordinate-based representation—such as implicit neural representations or neural operators—is applied.

JPEG Artifact Removal JPEG compression loss mainly occurs during the encoding process and exhibits complex characteristics. In particular, the independent processing of blocks introduces noticeable visual discontinuities at block boundaries, commonly known as blocking artifacts. High-frequency components undergo more heavier quantization than low-frequencies, resulting in greater loss in high-frequency regions. In addition, the chroma channels

Symbol	Definition	Description	Meaning / Note
$\mathbf{X}$	$\in \mathbb{R}^{H\times W\times 3}$	Original RGB image	Ground-truth
$\mathbf{Y}$	$= \mathcal{D}_8(\mathbf{X})$	Real-valued discrete cosine transform's coefficients	
$\mathbf{Q}$	$\in [1, 255]^{8\times 8}$	Standard JPEG quantization matrix	Integer
$\mathbf{Q}_\psi$	$\in [1, 255]^{8\times 8}$	Learned quant. matrix	Learned as float number and stored as integer.
H, W, K, M	$\in \mathbb{N}$	Height (H), Width (W), and Channel (K, M) of a Tensor	
B	$\in \mathbb{N}$	Block size of the JDNO	4 as implementation
M	$\in \mathbb{N}$	Cosine-feature channels of T_ρ	128 as implementation
$r_{1,2}$	$\in \{0.5, 1\}$	Sub-sampling ratio	$r_{1,2}$ for the height and the width, respectively
δ	$\in \mathbb{R}^{H\times W\times 2}$	Coord. grid	Input to JENO and used in $\mathcal{S}$
$\delta_{Y,C}$	$\in \mathbb{R}^{r_1 H\times r_2 W\times 2}$	Coordinates of $\mathbf{z}$	Used in $\mathcal{S}$
$\Delta\delta$	$\in [-1, 1]^2$	Local coordinates	Used in $\mathcal{S}$
s_i	$\in \mathbb{R}$	Local area weight	Used in $\mathcal{S}$
$\mathbf{c}$	$:= (2/r_1,\ 2/r_2)$	Anti-aliasing factor	Used in $\mathcal{S}$
$\mathbf{z}$	$\in \mathbb{R}^{H\times W\times K}$	Feature map of the JPNeO	
λ	$\in (0, \infty)$	Loss trade-off	Used in training $\mathbf{Q}_\psi$
$\varphi, \psi, \theta, \xi, \rho, \phi$	–	Trainable parameters of the JPNeO	
$\mathcal{D}_B$	$\mathbf{X} \rightarrow \mathbf{Y}$	Discrete Cosine Transform (DCT) with block-size B	$\mathcal{D}^{-1}$ as an inverse DCT (IDCT)
E_{JPEG}	$\mathbf{X} \rightarrow \mathbf{Y}'$	JPEG standard encoder	
D_{JPEG}	$\mathbf{Y}' \rightarrow \tilde{\mathbf{X}}$	JPEG standard decoder	
E_φ	$(\mathbf{X}, (\delta_Y, \delta_C)) \rightarrow (\mathbf{X}_Y, \mathbf{X}_C)$	JPEG Encoding Neural Operator (JENO)	$\mathcal{G}_\phi \circ \mathcal{S} \circ f_\xi$
D_θ	$(\tilde{\mathbf{Y}}_Y, \tilde{\mathbf{Y}}_C; \mathbf{Q}) \rightarrow \hat{\mathbf{X}}$	JPEG Decoding Neural Operator (JDNO)	$\mathcal{G}_\phi \circ T_\rho \circ \mathcal{S} \circ f_\xi$
f_ξ	$\mathbf{X} \rightarrow \mathbf{z}$ or $(\tilde{\mathbf{Y}}_{\mathbf{Y}}, \tilde{\mathbf{Y}}_{\mathbf{C}}) \rightarrow \mathbf{z}$	Feature extractor of JPNeO	JENO and JDNO, respectively
$\mathcal{G}_\varphi$	$\mathbf{z} \rightarrow \hat{\mathbf{X}}$	Neural Operator	Utilizing Galerkin attention
T_ρ	$(\mathbf{z}', \delta, \mathbf{Q}) \rightarrow \mathbf{z}$	Cosine Neural Operator (CNO)	$\mathbf{A} \otimes (\cos(\pi\mathbf{F_h} \otimes \delta_\mathbf{h}) \odot \cos(\pi\mathbf{F_w} \otimes \delta_\mathbf{w}))$
h_a, h_f	$\mathbb{R}^K \rightarrow \mathbb{R}^M$	Coefficient and Frequency for T_ρ	
h_q	$\mathbb{R}^{128} \rightarrow \mathbb{R}^M$	Quantization matrix encoder	Used in $\mathbf{A} = h_q(\mathbf{Q}) \cdot h_a(\mathbf{z}')$
$\mathcal{S}$	$(\mathbf{z}, \delta) \rightarrow \{[s_i \cdot \mathbf{z}(\delta_i), \Delta\delta_i]_{i=1}^j, \mathbf{c}\}$	Sampling operator	
$U(\cdot)$	$\mathbb{R}^{r_1 H\times r_2 W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$	Upsample operator	
$(\cdot)'$	–	Variant of $(\cdot)$ that is structurally equivalent	
$\tilde{(\cdot)}$	–	$(\cdot)$ with distortions	
$\hat{(\cdot)}$	–	Prediction to $(\cdot)$ with a neural network	
$HPF,\ LPF$	–	High/Low-pass filters	
$\mathcal{L}_d, \mathcal{L}_r$	–	Distortion / bitrate loss	For $\mathbf{Q}_\psi$
$\odot,\ \otimes$	–	Hadamard / Kronecker product	Element-/tensor-wise

Table S1. Notation table for the main paper (Elements and Functions (calculations), respectively.

are resized during compression, color distortions are more severe than those in the luminance channel.

We hypothesize that JPEG distortions can be modeled as a differential equation, and our objective is to formulate and solve this equation accordingly. To realize this mechanism, we model the decoder as a neural operator. In Fig. S2, a JPEG image $f = \mathbf{L}_a u$ is stored as a DCT spectrum. We propose a cosine operator that predicts a continuous spectral representation and maps it to image features—serving two key purposes in bridging frequency and spatial domains. The first is to infer high-frequency details, thereby improving the reconstruction of missing information. The second is to bridge the spectral and image domains by translating frequency-domain information into spatial representations. Then, a neural operator then approximates the inverse transform $\mathbf{L}_a^{-1}$.

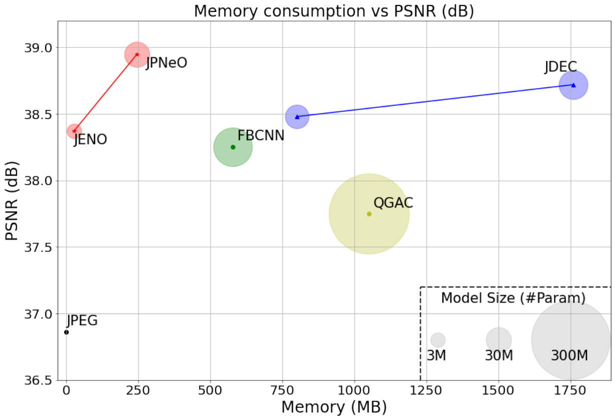
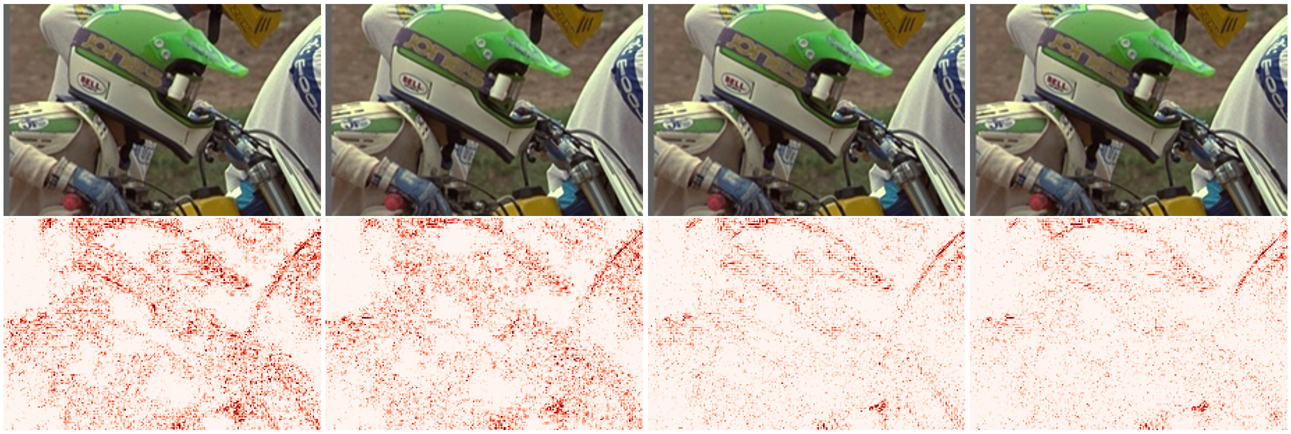


Figure S3. **PSNR and memory consumption comparison** with other methods in LIVE-1 [39]($q = 90$, 4:2:0 subsampling).

3. Additional Experiments

Memory Consumption Following the Sec. 6 of the main paper, we report memory consumption and the number of parameters. We compare our JPNeO with QGAC [17], FBCNN [23], and JDEC [22]. The size of the input for the comparison is 560×560. Fig. S3 shows the result of the comparison. We also report the memory usage of JENO. Notably, JENO surpasses the performance of FBCNN and QGAC while requiring only a minimal amount of memory.

Roles of JDNO and JENO Fig. S5 shows the results when each encoder and decoder is replaced with the JPNeO component, compared to the original JPEG. At



JPEG+$\mathbf{Q}$ JENO+$\mathbf{Q}$ JPEG+$\mathbf{Q}_\psi$ JENO+$\mathbf{Q}_\psi$

Figure S4. Qualitative comparison images (top) and corresponding error maps (bottom) compressed by varying an encoder ($E_{\text{JPEG}}/E_\varphi$) and a quantization matrix ($\mathbf{Q}/\mathbf{Q}_\psi$).

bpp|PSNR	JPEG + JPEG		JENO + JPEG		JPNeO	
	low	high	low	high	low	high
Q	0.358|25.29	1.488|33.07	0.358|25.31	1.499|33.19	0.358|**27.77**	1.499|35.95
Q_ψ	**0.335|25.50**	**1.464|34.03**	**0.335|25.51**	**1.486|34.19**	**0.335|27.77**	**1.486|36.86**

Table S2. Quantitative comparison by varying a quantization matrix ($\mathbf{Q}/\mathbf{Q}_\psi$)

low rates the dominant artifacts are from quantization and blocking; the decoder-side JDNO removes these artifacts over plain JPEG—see the shaded "JDNO lifting" band. At higher rates the quantization artifact is minor, so quality is bounded by the encoder's performance ("JENO lifting" band). As a result, JDNO is crucial in the low-bit-rate region, while JENO dominates at high rates. Fig. S5 relates to Sec. 6 and Fig. 13 of the main paper, where the increase in mutual information corresponds to improvements in PSNR. Notably, JENO is trained in a distortion-blind manner; while its impact is limited in highly degraded regions, it enhances color fidelity in high-quality areas as noted in Fig. 9 of the main paper. Fig. S4 supports the observation by supplementing Fig. 9 in the main paper by highlighting the advantage of jointly using JENO and the learned quantization map.

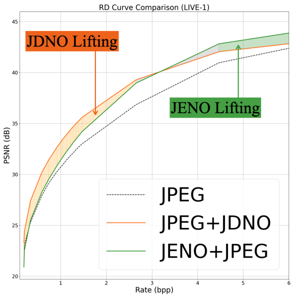


Figure S5. Improvements obtained by replacing $E_{\text{JPEG}}/D_{\text{JPEG}}$ to E_φ/D_θ compared to JPEG. The decoder contributes more significantly at low bpp levels, while the encoder becomes more influential as bpp increases.

Further, in Tab. S2 we provide a comparison between Q/Q_ψ. Since it is difficult to find a Q_ψ that matches the bpp of each Q, we show image quality at the most similar bpp values.

Dec.	LIVE1 [39]		B500 [4]		Enc.	LIVE1 [39]		B500 [4]	
	0	40	0	40		90	100	90	100
JDNO	23.25	32.17	23.20	31.89	JENO	37.20	45.47	37.84	48.61
CNN [30]	23.14	32.06	23.11	31.83	CNN [30]	37.17	45.21	37.71	47.15
UNet [23]	22.77	31.05	22.76	30.95	UNet [23]	37.13	44.87	37.67	46.68

Table S3. Quantitative ablation study by replacing components of our JPNeO with CNN [30] and U-Net [23] architectures.

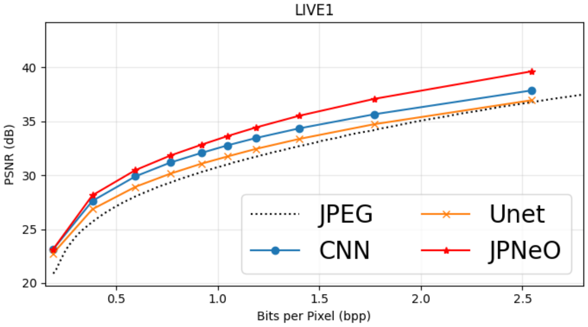


Figure S6. RD curve comparison with CNN [30] and U-Net[23] architectures.

Ablation Study In Tab. S3 and Fig. S6, we conducted ablation experiments using CNN and U-Net for the encoder and decoder. Specifically, we adopt the architecture introduced by Lim et al. [30] for CNN and Jiang et al. [23] for U-net. Our method consistently achieves better results than existing approaches. For a fair comparison, all networks were configured to have the same number of parameters. The results demonstrate that our JPNeO achieves greater efficiency under equal capacity.