This ICCV Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Extreme Compression of Adaptive Neural Images

Leo Hoshikawa[1*], Marcos V. Conde[1*], Takeshi Ohashi[2], Atsushi Irie[2]
[1] Sony Interactive Entertainment
[2] Sony Group Corporation

Abstract

Implicit Neural Representations (INRs) and Neural Fields are a novel paradigm for signal representation, from images and audio to 3D scenes and videos. The fundamental idea is to represent a signal as a continuous and differentiable neural network. This new approach poses new theoretical questions and challenges. Considering a neural image as a 2D image represented as a neural network, we aim to explore novel neural image compression. In this work, we present a novel analysis on compressing neural fields, with focus on images and introduce Adaptive Neural Images (ANI), an efficient neural representation that enables adaptation to different inference or transmission requirements. Our proposed method allows us to reduce the bits-per-pixel (bpp) of the neural image by 8 times, without losing sensitive details or harming fidelity. Our work offers a new framework for developing compressed neural fields. We achieve a new state-of-the-art in terms of PSNR/bpp trade-off thanks to our successful implementation of 4-bit neural representations.

1. Introduction

Neural Fields, also known as Implicit Neural Representations (INRs), allow the representation of signals (or data) of all kinds and have emerged as a new paradigm in the field of signal processing, neural compression, and neural rendering [18, 34, 46, 49, 52]. Unlike traditional discrete representations (*e.g.*, image as a discrete grid of pixels, audio signals are discrete samples of amplitudes), neural fields are continuous functions that describe the signal. Such a function maps the source domain $\mathcal{X}$ of the signal to its characteristic values $\mathcal{Y}$. It maps 2D pixel coordinates to their corresponding RGB values in the image $\mathcal{I}[x, y]$. This function ϕ is approximated using neural networks (NNs), thus it is continuous and differentiable. We can formulate Neural Fields as:

$$\phi : \mathbb{R}^2 \mapsto \mathbb{R}^3 \quad \mathbf{x} \rightarrow \phi(\mathbf{x}) = \mathbf{y}, \tag{1}$$

where ϕ is the learned INR function, the domains $\mathcal{X} \in \mathbb{R}^2$ and $\mathcal{Y} \in \mathbb{R}^3$, the input coordinates $\mathbf{x} = (x, y)$, and the output RGB value $\mathbf{y} = [r, g, b]$. In summary, neural representations are essentially simple neural networks (NNs), once these networks ϕ (over)fit the signal, the model become implicitly the signal itself.

This approach has become foundational research in many areas including image compression [18, 19, 49], audio compression [50, 51], video compression [12, 13] and 3D representations (*e.g.*, NeRF, DeepSDF) [34, 36, 38, 39].

In the context of *image compression*, this method offers unique mathematical properties due to its continuous and differentiable nature [18, 19, 49]. One of the major advantages of using INRs is that there are no ties with spatial resolution; unlike conventional methods where the image resolution is tied to the discrete number of pixels, the memory needed for these representations only scales with the complexity of the underlying signal [46, 52]. In essence, INRs offer "infinite resolution", it can be sampled at any spatial resolution [46] by upsampling the input domain $\mathcal{X}$ (*e.g.* $[H, W]$ grid of coordinates), being particularly useful for high-dimensional signal parametrization whereas traditional methods struggle due to memory limitations.

Considering this, we define a *neural image* as a neural network (INR) that represents a particular image of an arbitrary resolution — see Figure 2.

Recent works [18, 19, 36] demonstrate that we can fit large images (even giga-pixel images) using "small" neural networks as INRs, which implies promising compression capabilities [18, 19]. However, neural fields represent a *lossy compression* technique, especially limited by Shannon's Theorem [44]; *i.e.* even utilizing complex deep neural networks, to parameterize the high-frequencies of certain images remains a challenging or impossible task.

In this work, we focus on the particular case of 2D images, since it is well-known that this serves as a good proxy for 3D research [34, 45, 46, 52].

Contributions (i) We provide an extensive benchmark on extreme image compression using neural fields. (ii) We propose Adaptive Neural Images (ANI), a novel neural repre-

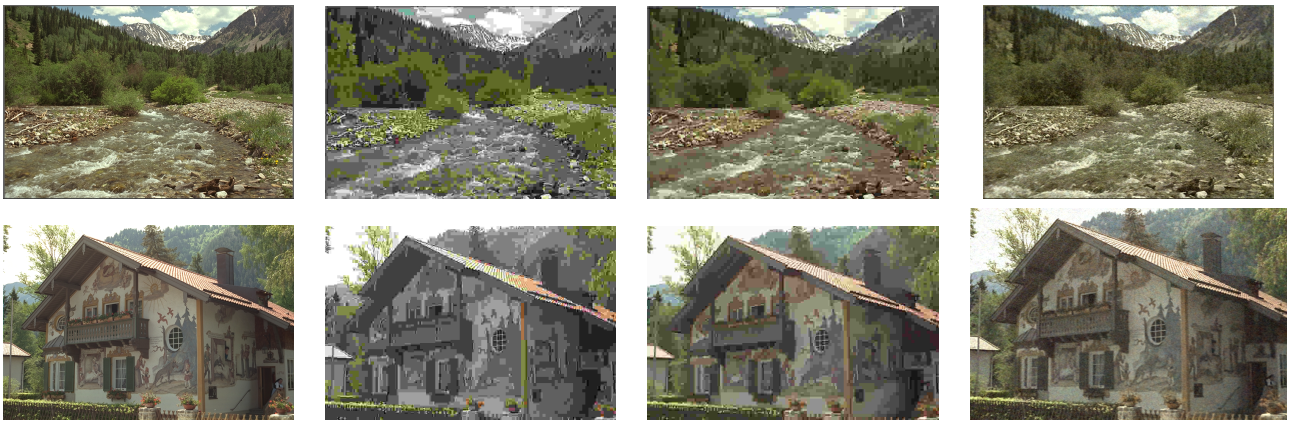
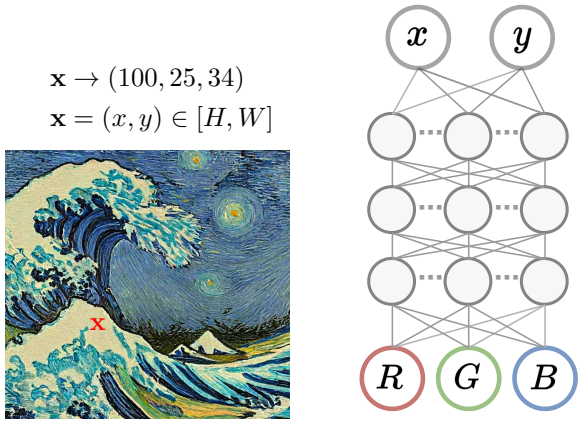


Figure 1. **Comparison with traditional codecs.** Our proposed neural image ANI (at 4-bits) state-of-the-art, high-fidelity results without clearly unpleasant artifacts. Note that all the images are around 0.3 bpp. Images taken from the Kodak dataset id 13 and 24.



(a) Conventional RGB image (b) Coordinate-based MLP

Figure 2. We illustrate the general concepts around neural image representations [46, 52]. INRs can be generalized to other sorts of signals such as audio or 3D representations.

sentation that allows adaptation to different memory and inference requirements. We achieve this by using *state-of-the-art* neural architecture search (NAS) to find the optimal neural network. Our approach allows us to reduce $8\times$ the required bits-per-pixel without losing much fidelity while establishing a new state-of-the-art in PSNR/bpp ratio. (iii) We provide useful insights related to the quantization of neural fields, that can be applied to other related tasks (*i.e.* 3D NeRF).

2. Related Work

Learned Image Compression. The concept of learned image compression was pioneered by [4], through the introduction of an end-to-end framework combining an auto-encoder with an entropy model to jointly optimize both rate and distortion metrics. Many approaches [5, 30, 32, 35] enhanced this model by incorporating a scaling hyper-prior to the architecture, and the use of autoregressive entropy models. The current trend on generative image compression represents the state-of-the-art in terms of perceptual quality [2, 3, 26, 33].

Model Compression. Due to the industry requirements in terms of inference speed, memory, and energy consumption, in recent years there has been plenty of research on model compression [15, 31, 56]. For instance,[25] proposes a simple framework: applying pruning, quantization, and entropy coding –in sequence– combined with retraining in between the steps. To optimize performance under quantization, several works use mixed-precision quantization and post-quantization optimization techniques [7, 11, 14, 16, 20, 37, 54, 55].

In particular, we adopted LSQ [20] as the base of our quantization method. It is a strong method that improves quantization using learnable scaling factors enabling extreme low-precision settings. In the context of neural fields, the neural network represents the data itself, thus, a model compression implies (additional) data compression. Despite this being a promising approach, very few works tackle this problem [18, 19, 24, 46].

Neural Architecture Search (NAS) In recent years, NAS has emerged as a powerful approach for automating the design of optimal neural network architectures for a given task, significantly reducing the need for manual experimentation [59, 60]. The field has since seen rapid progress, with methods like Efficient Neural Architecture Search (ENAS) by Pham et al. [41], which significantly reduces search time by sharing weights among different architectures. Once-for-All [10] allows us to train a single neural network and specialize it for efficient deployment.

2.1. Neural Representations

In recent years, implicit neural representations (INRs) [17, 22, 36, 46] have become increasingly popular in image processing as a novel way to parameterize an image. Also known as coordinate-based networks, these approaches use multilayer perceptrons (MLPs) to overfit one image and represent it. Multiple works have demonstrated the potential of MLPs as continuous, memory-efficient implicit representations for images [46, 49].

We denote the INRs as a function ϕ with parameters θ, defined as:

$$\begin{aligned} \phi(\mathbf{x}) = \mathbf{W}_n(\varsigma_{n-1} \circ \varsigma_{n-2} \circ \ldots \circ \varsigma_0)(\mathbf{x}) + \mathbf{b}_n \\ \varsigma_i(\mathbf{x_i}) = \alpha\left(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i\right), \end{aligned} \tag{2}$$

where ς_i are the layers of the network (considering their corresponding weight matrices $\mathbf{W}$ and bias $\mathbf{b}$), and α is a nonlinear activation *e.g*. ReLU, Tanh, Sine [46], complex Gabor wavelet [43]. Considering this formulation, the parameters of the neural network θ is a set of weights and biases of each layer (*i.e*. $\mathbf{W}$ and $\mathbf{b}$). Since the input of the MLP are the coordinates $\mathbf{x}$ in the domain $[H, W] \in \mathbb{R}^2$, these are also known as coordinate-based MLPs — see Figure 2b.

Sitzmann *et al.* [46] presented SIREN, a periodic activation function for neural networks based on the Sine function, specifically designed to better model complex natural signals and high-frequencies in the images. Tancik *et al.* [52] introduced Fourier features as input positional encodings for the network, enhancing their capability to model high-frequencies. COIN [18, 19] explored the early use of INRs for image compression. Strumpler *et al.* [49] proposed a framework for image compression and transmission using INRs. We also find other works that tackle new activation functions such as multiplicative filter networks (MFN) [21] and Wire [43], and multi-scale representations [36, 42]. Other such as Instant-NGP [36] and SHACIRA [23] approaches focus on multi-resolution representations using hierarchical representations and hash-tables to improve performance and speed.

Following previous work [18, 49], we will use SIREN [46] as the baseline model. We will explore extreme compression of the neural network, and new training techniques to derive our proposed adaptive neural images (ANI). We will also analyze the most popular and recent approaches: FourierNets [52] (MLP with Positional Encoding), SIREN [46], MFN [21], Wire [43] and DINER [57].

3. Transmission of Neural Images

Transmitting signals as INRs is a novel research problem [19, 49]. In this context, it is fundamental to understand that the image is no longer characterized as a discrete set of RGB pixels, but as a set of weights and biases (θ *i.e*. the neural network itself). We illustrate in Figure 3a the most popular approach for compressing images using INRs. First, we train the neural network ϕ to fit the image, next we can apply post-training quantization (PTQ) and encode the parameters θ using lossless entropy coding. We could also apply post-quantization retraining to improve the performance of the neural network. Finally, we can transmit the parameters θ, the client can recover the network, and thus reconstruct the natural RGB image.

Our approach considers quantization-aware training (QAT) directly, which offers better performance and a higher compression ratio. We show our method in Figure 3b.

Besides QAT, the key concept of our approach is the active neural architecture search (NAS) to produce a "once-for-all" neural network [10] *i.e*. a single network is trained to support versatile architectural configurations including depth (number of layers) and width (number of neurons). Therefore considering our neural image with parameters θ we can derive –during inference– different sub-networks with varying number of layers and neurons. We illustrate the sub-networks θ_1 and θ_2 in Figure 3b.

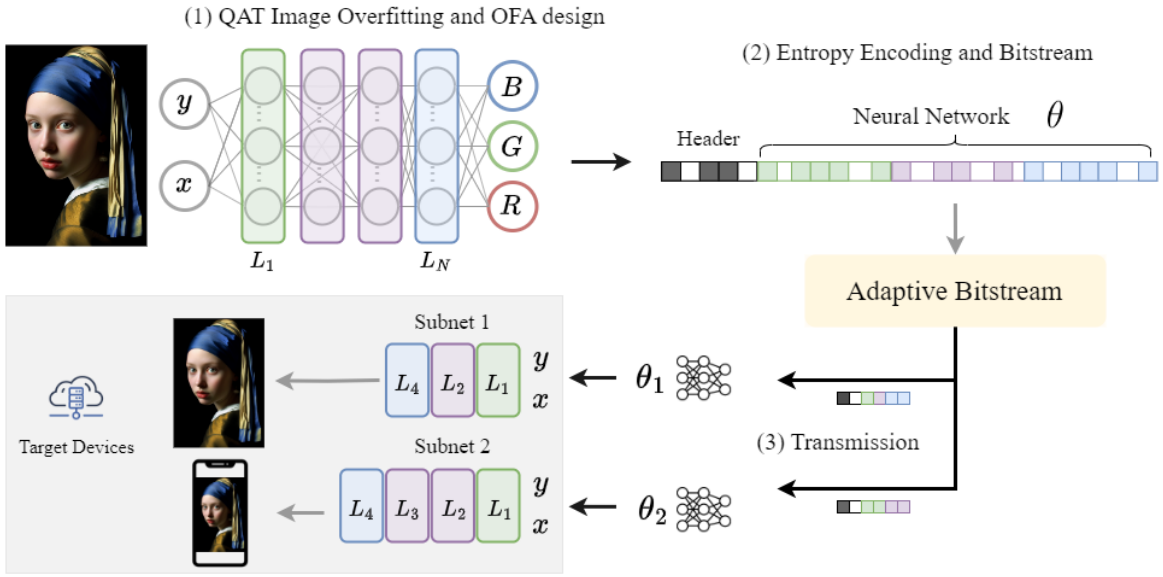
We define **Adaptive Neural Images (ANIs)** as once-for-all neural representations of images. Note that ANIs are also trained to support quantization. Note that the neural representation training is done *offline* only once for a particular image, thus, the training time is not constraint. Moreover, training to convergence is possible in a few minutes.

General Limitations Before presenting our approach, we must discuss the fundamental limitations of neural images to better understand the experimental results. First, INRs are lossy compression methods. Second, most INR approaches are signal-specific *i.e*. the neural network fits a particular image. This implies training *ad hoc* the neural network using a GPU — although this can take less than 1 minute, and meta-learning [53] helps to accelerate training. Third, the performance of the INR methods highly varies depending on hyper-parameters (*e.g*. learning rate, number of neurons and layers), and the target signal. However, there is no theoretical or practical way of predicting *a priori* which INR model fits best the signal.

4. Our Approach for Extreme Compression

Given a neural representation of an image –a neural image–, our goal is to reduce as much as possible the number of bits while preserving the original signal. Considering that the neural network represents the signal itself, we must focus on compressing the neural network (*i.e*. weights and biases).

(a) Overview of INR-based compression pipeline proposed by Y. Strümpler et al. [49]. The basic compression pipeline comprises image overfitting, quantization of the neural network, AdaRound, retraining, and lossless entropy coding (*e.g.* binarized arithmetic coding).



(b) Our proposed approach uses adaptive neural images (ANI). We perform directly quantization aware training (QAT) [7] and once-for-all (OFA) optimization [10]. Depending on the bandwidth and transmission requirements, our bitstream can be adapted (*e.g.* trimmed) allowing us to send more/less information, this is only possible thanks to the proposed ANI architecture. Moreover, depending on the target device speed and memory requirements, we can utilize smaller versions of our neural network ANI without any re-training or adaptation. We highlight the client side and target devices.

Figure 3. We illustrate the general concepts around image compression and transmission using INRs [49]. Our approach enables to adapt to diverse scenarios depending on the bandwidth, memory, and target device requirements.

4.1. Post-Training Quantization

Post-Training Quantization (PTQ) calculates quantization parameters without re-training. In our experiments, we adopted the standard PTQ algorithm proposed by [27] and wide used on several studies [49]. This algorithm allows quantization to 7-bits and 8-bits with minimal losses.

4.2. Quantization-Aware Training (QAT)

Quantization-aware training (QAT) methods have a considerable advantage over PTQ methods in terms of compression ratio [8, 20, 25, 28], allowing extremely small bit-width (2, 4-bit) at the expense of additional training time.

In general neural networks, weights follow zero-mean normal distributions, while the distribution for the activations varies greatly depending on the architecture and nonlinearities. For INRs, the behavior of the MLP and activations is well-known. In SIREN [46] the sine activation conveniently restricts the distribution to a normalized range with zero-mean. For MFNs [21], the filter passes through the sine activation and is multiplied by the output of the linear layers, conveniently restricting the range. These properties allow us to experiment with extreme compression low-bits settings (2,4-bits).

Unlike previous methods [24, 49], we use the *state-of-the-art* LSQ [20] quantization algorithm.

Following the notation from [20], we define $\bar{x}$ and $\hat{x}$ as the coded bits and quantized values, respectively. The weights and activations are quantized as follows:

$$\bar{x} = quantize(clamp(\frac{x}{s}, x_{min}, x_{max})) \quad , \quad \hat{x} = \bar{x} \times s \tag{3}$$

The *s* parameters are clipping factors learned using back-propagation [20]. For activations, x_{min} is *0* if the *x* is strictly positive, or *-1* otherwise, and x_{max} is always 1. For weights, $\mathbf{x}_{min}$ and x_{max} are always *-1* an *1* respectively.

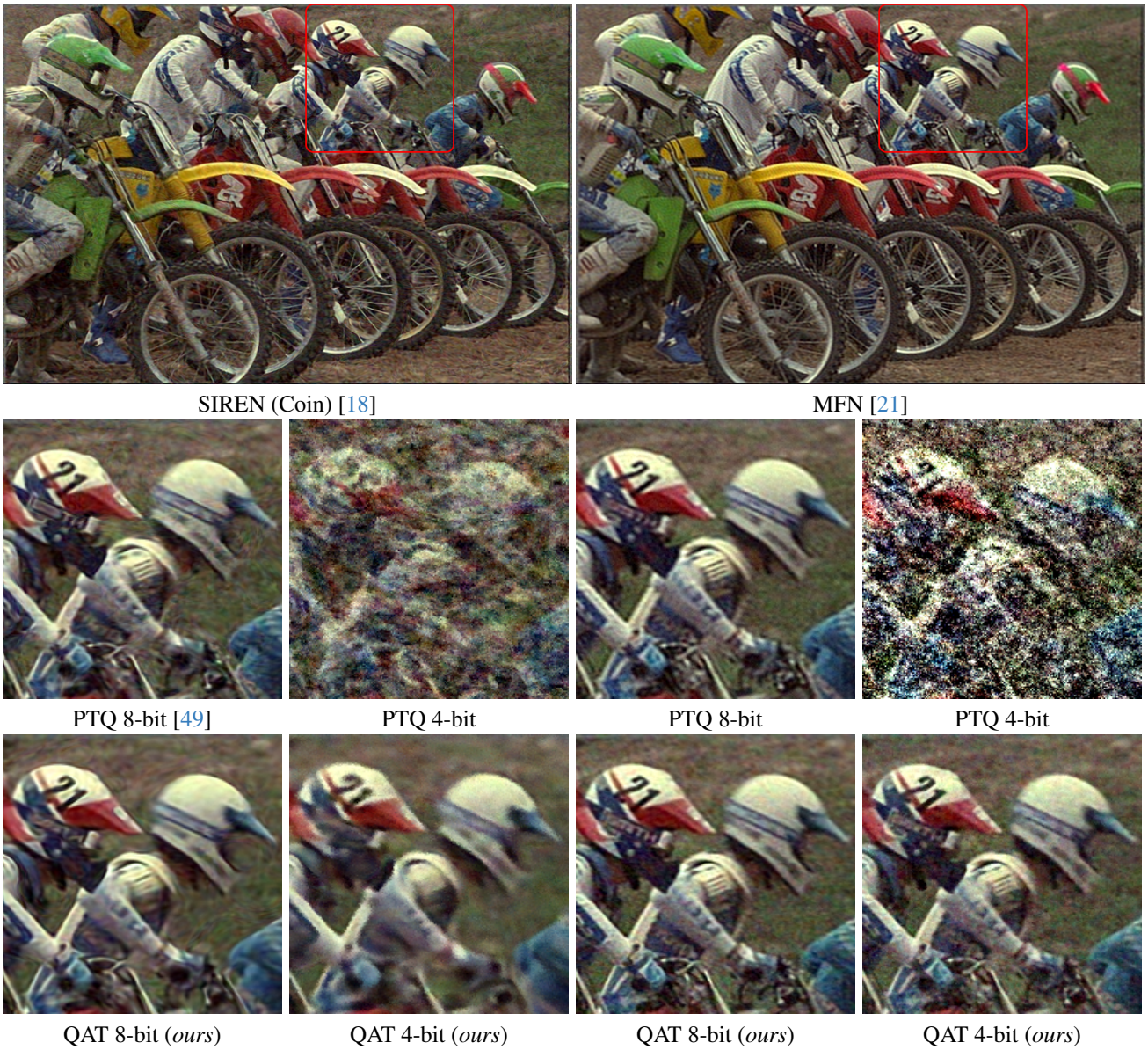


Figure 4. **Comparison between PTQ and QAT.** Visual results on Kodak [1] at different bit-widths. We can appreciate how at 4-bits PTQ loses the signal, while QAT maintains high fidelity. Our method improves previous approaches [18, 19, 49].

On n-bits, quantization is given by:

$$\bar{x} = \frac{round((\bar{x}+1)\times 2^{n-1})}{2^n} \quad , n\text{=number of bits.} \quad (4)$$

The same equation is applied to both the weights and activations. We use straight-through estimator (STE) [6] and update the quantization parameters using back-propagation. In Figure 4 we show the benefits of using LSQ [20] quantization-aware training (QAT) over post-training quantization (PTQ).

4.3. Neural Architecture Search (NAS)

We experimented with NAS to find optimal architectures automatically. Since we expect multiple target devices and different specifications, we use Once-for-All (OFA) [9], a supernet approach that allows training once and extracting multiple sub-architectures of different sizes with minimal retraining, and adapted to INRs. Since INRs are essentially MLPs, the only moving parts that can be made "elastic" are the depth (number of layers) and the width (number of channels or neurons). Additionally, to simplify the search space, we adopted a uniform number of channels for all interme-

Method	Quantization	Size(KB)	PSNR ↑	SSIM ↑	BPP ↓
SIREN [46]	Coin [18] (None)	270.28	27.98±2.73	0.782	1.812
	PTQ 8-bit [49]	71.46	27.78±2.71	0.760	0.479
	PTQ 4-bit (*ours*)	38.33	18.24±1.72	0.216	0.236
	QAT 8-bit (*ours*)	71.46	27.80±2.34	0.742	0.479
	QAT 4-bit (*ours*)	38.33	27.59±3.30	0.638	0.236
MFN [21]	Coin [18] (None)	284.29	29.16±2.80	0.822	1.906
	PTQ 8-bit [49]	85.43	28.60±2.82	0.785	0.572
	PTQ 4-bit (*ours*)	52.29	14.22±2.20	0.136	0.254
	QAT 8-bit (*ours*)	85.43	29.86±2.99	0.780	0.572
	QAT 4-bit (*ours*)	52.30	28.50±2.72	0.683	0.254

Table 1. **Quantization INR Analysis** on Kodak [1]. We report the average PSNR –over 5 runs– for the whole Kodak image dataset using different quantization settings. All the neural networks have 4 layers and 128 neurons. We are the first approach to achieve successful 4-bit quantization of neural images.

diate layers. While this restricts the search space, it allows us to train and evaluate several possible layouts.

During training, the subnets are initialized using progressive shrinking from [9]. we alternate between large and small networks to remove architecture-related bias, inspired by the sandwich rule proposed by [58]. Next, we fine-tune the sub-networks for a small amount of epochs to improve the fidelity *w.r.t* of the target image.

Algorithm 1 Once-for-all training strategy

Require: Search space of channels W=$\{W_0$,W_1,...,$W_n\}$, layers D=$\{D_0$,D_1,...,$D_n\}$
$S = W \times D$
for each $s \in S$ **do**
  params = get_model_size(s)
end for
//Argsort S using params
$idx = argsort(params)$
//Reorder S by alternating large and small architectures
$S_{sorted} = sort\_and\_shuffle(S, idx)$
$supernet = build\_model(W_n, D_n)$
$train(supernet)$
for each $s \in S_{sorted}$ **do**
  $subnet = supernet.get\_subnet(s)$
  $train(subnet)$
end for
return subnet

This OFA approach allows us to realize the adaptive bitstream with minimum effort.

5. INR Compression Benchmark

5.1. Quantization experiments

We run an exhaustive benchmark using the Kodak dataset [1], we provide the results in Table 1. Each experiment was repeated five times with different random seeds. We report the average performance of the five experiments. In all the experiments we used models with 4 layers and 128 channels.

BPP range	Bitwidth	1×ch	PSNR ↑
0.1	8-bit	3×64	25.36±2.64
	4-bit	**2×128**	**26.39±2.62**
0.5	8-bit	4×128	29.86±2.98
	4-bit	**2×256**	**30.10±3.06**
1.0	8-bit	4×128	30.24±2.98
	4-bit	**4×256**	**33.10±3.06**

Table 2. **Cost-efficiency of 4-bits method** on Kodak [1] using MFN backbone. For the same bpp budget, the 4-bit model achieves superior PSNR for all bpp ratios.

For the PTQ experiments, we follow [49]. Since the results for this technique are deterministic, we select the best-performing model per image (considering the five different runs). Following other quantization experiments, we kept the first and last layers at full precision. The impact of quantizing these layers is described as an ablation study.

5.2. Quantized NAS experiments

To develop our adaptive neural images (ANIs) we use OFA [9]. Using this NAS technique, we defined our search space of $[64, 128, 192, 256]$ channels and $[2, 3, 4, 5]$ intermediate layers. We train all possible 16 architecture combinations for 50000 epochs each.

Benchmark Conclusions Considering the results from Table 1, we are the first approach to achieve successful 4-bit quantization of neural representations. At 8-bits, both PTQ and QAT deliver similar quality without notable degradations. However, at 4-bits, the model quantized with PTQ loses the signal information, yet the model quantized with QAT maintains the signal and provides good fidelity. Our approaches improve Coin [18] and previous INR compression [49] by **+14dB** when using 4-bits. Figure 4 shows the visual results. Both SIREN [46] and MFN [21] presented similar behavior during quantization, with SIREN [46] being slightly more cost-efficient due to having fewer full precision parameters but suffers more degradation than MFN counterpart. In Figure 7, we show the results of our ANI *i.e.* a single super-network that allows inferring using subnetworks depending on the memory requirements. We provide more qualitative samples in the appendix.

We compare our approach with other compression methods in Figure 5. Our approach using 4-bits presents the best PSNR/ bpp trade-off along the whole spectrum, establishing a new state-of-the-art for INR compression.

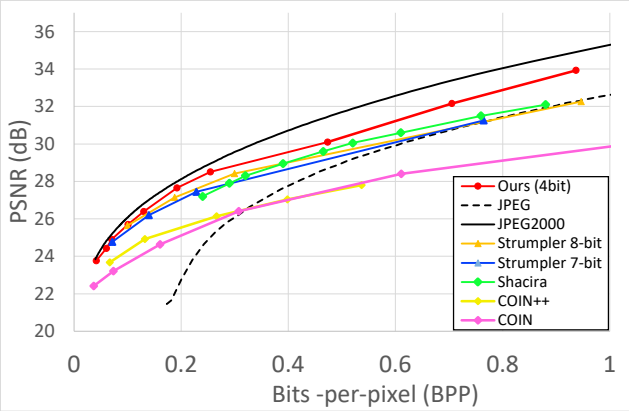


Figure 5. Comparison of our approach on the Kodak dataset with other methods. We achieved state-of-the-art performance, surpassing even newer methods such as SHACIRA [23]. Note that ANIMFN is a single neural network that can be adapted to different bpp requirements, unlike Coin [18, 19] or SIREN [46, 49].

5.3. NeRF Extension

Our approach would allow to effectively compress any MLP-based INR. We tried our compression approach on SHACIRA [23], which improves InstantNGP [36]. We aim to prove that our quantization approach can be extended to other modalities. We applied 4-bits QAT to the MLP model in SHACIRA, keeping the latent space optimization as the original. We obtained $8\times$ model size reduction with zero degradation (*32.61dB*). However, the actual model size reduction is small (from 1.96 MB to 1.82 MB) as the latent space size accounts for 90% of the total size (around 1.81MB). Figure 6 compares the qualitative results of the full precision and the quantized SHACIRA model.

We also provide samples in the supplementary material where we show 4-bit NeRFs [34] without apparent loss.

6. Technical and Implementation Details

We implement all the methods in PyTorch, using the author's implementations when available. We train all the models using the same environment with the Adam [29] optimizer, and we adapt the learning rate for each method's requirements. For instance MFN [21] uses 0.01 while SIREN [46] uses 0.001.

We use NVIDIA RTX 2080Ti and A100 (40GB and 80GB) cards. The models are optimized using the $\mathcal{L}_2$ reconstruction loss [46, 52] to minimize the RGB image reconstruction error $\sum_{x,y} \|\mathcal{I}[x,y] - \phi(x,y)\|_2^2, \quad \forall (x,y) \in [H,W]$. Note that due to the memory requirements of FHD images, the optimization is only possible on GPU cards with $>$ 40Gbs of VRAM.

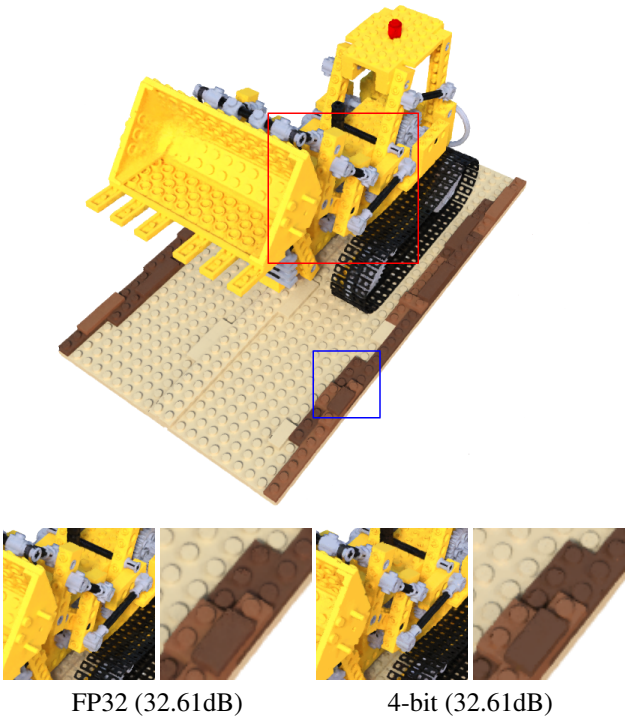


Figure 6. Experiment using our 4-bit quantization model on SHACIRA [23] 3D NERF. Our model is visually indistinguishable from the full precision model.

Ablation Studies In the supplementary, we provide ablation studies on the impact of layer quantization *i.e.* which layers suffer the most.

Applications Our method not only represent a theoretical contribution, ANIs allow to rethink content storage and transmission. Since we do not need to stream whole neural network to decode partial information, yet just a few layers, ANIs could have beneficial impact in remote sensing *i.e.* satellite imagery transmission.

Limitations. A clear limitation of using INRs for neural image compression is their stochastic nature and unstable training. Moreover, there is no practical way of predicting *a priori* which INR model fits best the signal. On the other hand, Having the once-for-all alleviates this process, as a diverse array of PSNR/bpp ratios is readily available for serving.

7. Conclusion

In this work, we present a novel analysis on compressing neural representations. We also introduce Adaptive Neu-

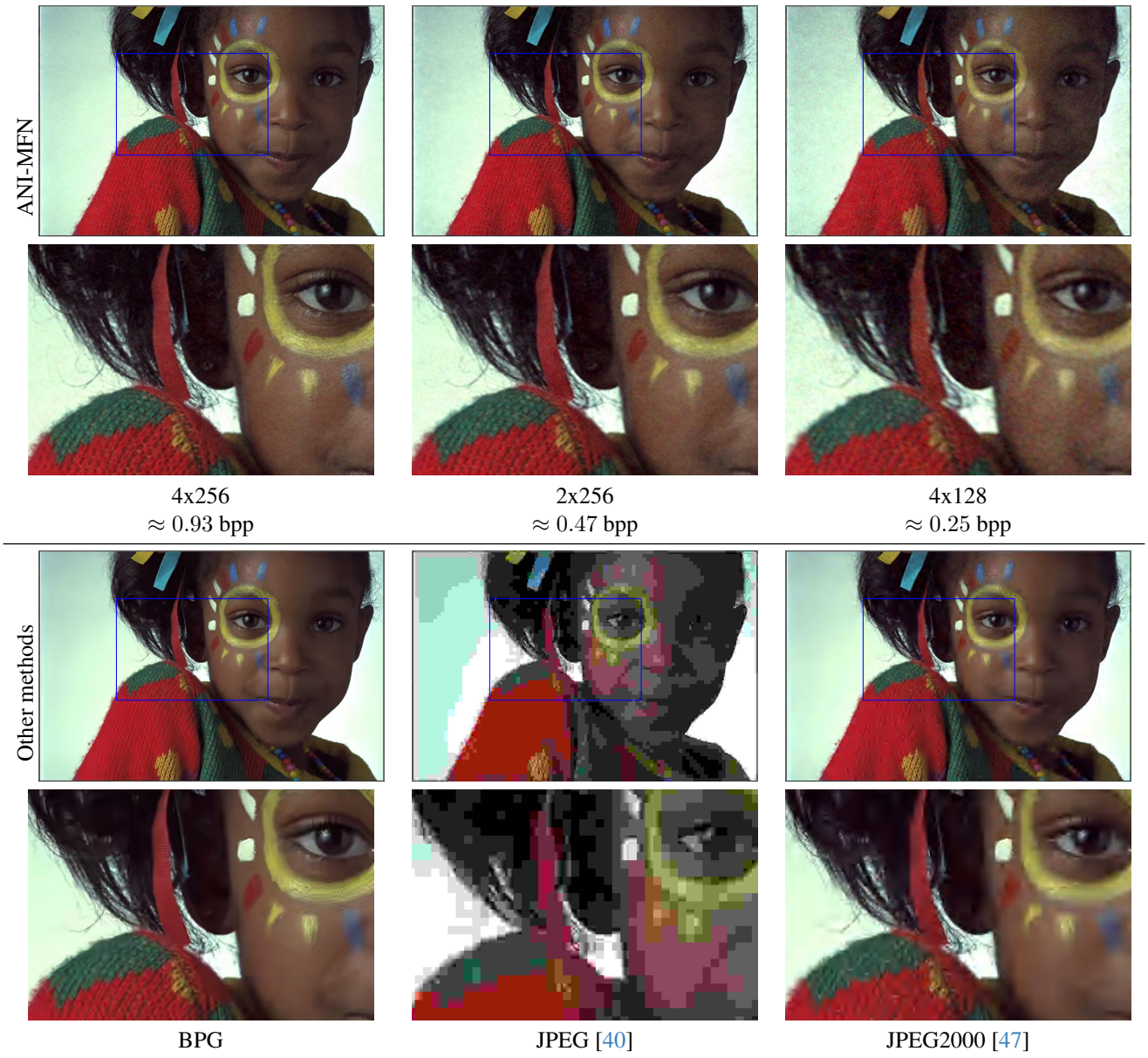


Figure 7. We present results of ANI using MFN [21] as backbone. Our single neural image can be adapted to different memory-fidelity requirements. The images correspond to a single neural network with three different subnetworks defined as layers×neurons. Our methods achieve better performance than BPG, JPEG [40], JPEG2000 [47] at ≈ 0.23 bpp.

ral Images (ANI), an efficient neural representation that enables adaptation to different inference or transmission requirements. We derive our ANI super-network using advanced once-for-all architecture search. To the best of our knowledge, we are the first approach to achieve successful 4-bit quantization of neural representations, establishing a new state-of-the-art. Moreover, this work provides the most complete benchmark for this task. Our work offers a transversal framework for developing compressed neural fields.

References

[1] Kodak lossless true color image suite. http://r0k.us/graphics/kodak/. 5, 6

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 221–231, 2019. 2

[3] Eirikur Agustsson, David Minnen, George Toderici, and Fabian Mentzer. Multi-realism image compression with a

conditional generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22324–22333, 2023. 2

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *International Conference on Learning Representations (ICLR)*, 2017. 2

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *International Conference on Learning Representations (ICLR)*, 2018. 2

[6] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 5

[7] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 696–697, 2020. 2, 4

[8] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. LSQ+: improving low-bit quantization through learnable offsets and better initialization. *CoRR*, abs/2004.09576, 2020. 4

[9] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *CoRR*, abs/1908.09791, 2019. 5, 6

[10] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 2, 3, 4

[11] Sek M. Chai. Quantization-guided training for compact TinyML models. *Research Symposium on Tiny Machine Learning*, 2021. 2

[12] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 1

[13] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2047–2057, 2022. 1

[14] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019. 2

[15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[16] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. *International Conference on Computer Vision (ICCV)*, 2019. 2

[17] Yishun Dou, Zhong Zheng, Qiaoqiao Jin, and Bingbing Ni. Multiplicative fourier level of detail. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2023. 3

[18] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 1, 2, 3, 5, 6, 7

[19] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022. 1, 2, 3, 5, 7

[20] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 2, 4, 5

[21] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2020. 3, 4, 5, 6, 7, 8

[22] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, pages 7154–7164, 2019. 3

[23] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17513–17524, 2023. 3, 7

[24] Cameron Gordon, Shin-Fang Chng, Lachlan MacDonald, and Simon Lucey. On quantizing implicit neural representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 341–350, 2023. 2, 4

[25] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *International Conference on Learning Representations, (ICLR)*, 2016. 2, 4

[26] Emiel Hoogeboom, Eirikur Agustsson, Fabian Mentzer, Luca Versari, George Toderici, and Lucas Theis. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023. 2

[27] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, abs/1712.05877, 2017. 4

[28] Sambhav Jain, Albert Gural, Michael Wu, and Chris Dick. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. *Proceedings of Machine Learning and Systems*, 2:112–128, 2020. 4

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[30] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2

[31] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better, 2021. 2

[32] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4394–4402, 2018. 2

[33] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems (NeuIPS)*, 33, 2020. 2

[34] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 7

[35] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[36] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 3, 7

[37] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization, 2020. 2

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1

[39] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, pages 9054–9063, 2021. 1

[40] William B Pennebaker and Joan L Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992. 8

[41] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018. 2

[42] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *European Conference on Computer Vision*, pages 318–333. Springer, 2022. 3

[43] 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. 3

[44] Claude E. Shannon. *Coding Theorems for a Discrete Source With a Fidelity Criterion*. 1959. 1

[45] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural. In *NeurIPS*, 2019. 1

[46] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 1, 2, 3, 4, 6, 7

[47] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58, 2001. 8

[48] Yannick Strümpler, 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. 2

[49] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022. 1, 3, 4, 5, 6, 7

[50] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022. 1

[51] Filip Szatkowski, Karol J Piczak, Przemysław Spurek, Jacek Tabor, and Tomasz Trzciński. Hypersound: Generating implicit neural representations of audio signals with hypernetworks. *arXiv preprint arXiv:2211.01839*, 2022. 1

[52] 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. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 1, 2, 3, 7

[53] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021. 3

[54] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision DNNs: All you need is a good parametrization. *International Conference on Learning Representations (ICLR)*, 2019. 2

[55] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[56] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023. 2

[57] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. Diner: Disorder-invariant implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6143–6152, 2023. 3

[58] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas S. Huang, Xiaodan Song, Ruoming Pang, and Quoc V. Le. Bignas: Scaling up neural architecture search with big single-stage models. *CoRR*, abs/2003.11142, 2020. 6

[59] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 2

[60] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 2