Implicit Neural Representations for Image Compression

Yannick Strümpler* [1], Janis Postels* [1], Ren Yang [1], Luc Van Gool[1], and Federico Tombari [2,3]

[1] ETH Zurich
[2] Technical University of Munich
[3] Google

Abstract. Implicit Neural Representations (INRs) gained attention as a novel and effective representation for various data types. Recently, prior work applied INRs to image compressing. Such compression algorithms are promising candidates as a general purpose approach for any coordinate-based data modality. However, in order to live up to this promise current INR-based compression algorithms need to improve their rate-distortion performance by a large margin. This work progresses on this problem. First, we propose meta-learned initializations for INR-based compression which improves rate-distortion performance. As a side effect it also leads to faster convergence speed. Secondly, we introduce a simple yet highly effective change to the network architecture compared to prior work on INR-based compression. Namely, we combine SIREN networks with positional encodings which improves rate distortion performance. Our contributions to source compression with INRs vastly outperform prior work. We show that our INR-based compression algorithm, meta-learning combined with SIREN and positional encodings, outperforms JPEG2000 and Rate-Distortion Autoencoders on Kodak with 2x reduced dimensionality for the first time and closes the gap on full resolution images. To underline the generality of INR-based source compression, we further perform experiments on 3D shape compression where our method greatly outperforms Draco - a traditional compression algorithm.

1 Introduction

Living in a world where digitalization is ubiquitous and important decisions are based on big data analytics, the problem of how to store information effectively is more important than ever. Source compression is the generalized term for representing data in a compact form, that either preserves all the information (lossless compression) or sacrifices some information for even smaller file sizes (lossy compression). It is a key component to tackle the flood of image and video data that is uploaded, transmitted and downloaded from the internet every day. While lossless compression is arguably more desirable, it has a fundamental

* Equal contribution

theoretical limit, namely Shannon's entropy [47]. Therefore, lossy compression aims at trading off a file's quality with its size - called rate-distortion trade-off.

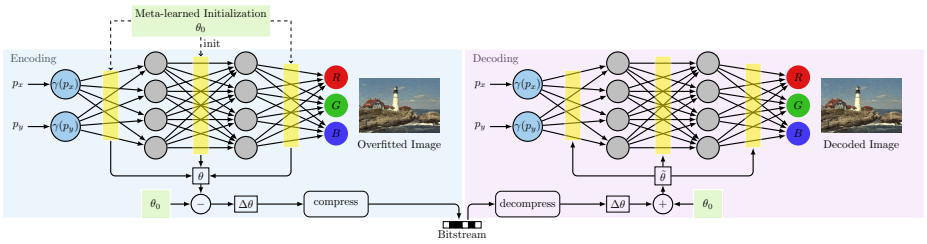


Fig. 1: Method Overview: We summarize our approach to use Implicit Neural Representations (INRs) for compression by using the model weights θ as the representation for an image. We also visualize how a meta-learned initialization θ_0 is used in the encoding and decoding process in order to compress only the weight update $\Delta\theta$ into the bitstream.

Apart from traditional hand-designed algorithms tuned for particular data modalities, e.g. audio, images or video, machine learning research has recently developed promising learned approaches to source compression by leveraging the power of neural networks. Such methods typically build on the well-known autoencoder [28] by implementing a constrained version of it. These so-called Rate-Distortion Autoencoders (RDAEs) [5,6,37,25] jointly optimize the quality of the decoded data sample and its encoded file size.

This work sidesteps the prevalent approach of RDAEs and investigates a novel paradigm for source compression - particularly focusing on image compression. Recently, Implicit Neural Representations (INRs) gained popularity as a flexible, multi-purpose data representation that is able to produce high-fidelity samples on images [49], 3D shapes [44,49] and scenes [40]. In general, INRs represent data that lives on an underlying regular grid by learning a mapping between the grid's coordinates and the corresponding data values (e.g. RGB values) and have even been hypothesized to yield well compressed representations [49]. Due to their generality and concurrent early attempts to leverage them for compression [19,11,20], INRs denote a promising candidate as a general purpose compression algorithm.

Currently there are two main challenges for INR-based compression algorithms: (1) Straightforward approaches struggle to compete even with the simplest traditional algorithms [19]. (2) Since INRs encode data by overfitting to particular instances, the encoding time is perceived impractical. To this end, we make two contributions. Firstly, we propose meta-learned for INR-based compression. We exploit recent advances in meta-learning for INRs [48,52] based on Model-Agnostic Meta-Learning (MAML) [22] to find weight initializations that can compress data with fewer gradient updates as well as yield better rate-distortion performance. Secondly, we combine SIREN with positional encodings

for INR-based compression which greatly improves rate-distrortion performance. While we focus on images, we emphasize that our proposed method can easily be adapted to any coordinate-based data modality. Overall, we introduce a compression pipeline that vastly outperforms the recently proposed COIN [19] and is competitive with traditional compression algorithms for images. Moreover, we demonstrate that meta-learned INRs already outperform JPEG2000 and a few RDAEs on downsampled images. Lastly, we emphasize the generality of INR-based image compression by directly applying our approach to 3D data compression where we outperform the traditional algorithm Draco.

2 Related Work

Learned Image Compression. Learned image compression was introduced in [6] by proposing an end-to-end autoencoder and entropy model that jointly optimizes rate and distortion. In the following, [7] extends this approach by adding a scale hyperprior, and then [41,37,33] propose employing autoregressive entropy models to further improve the compression performance. Later, Hu *et al.* [29] propose a coarse-to-fine hierarchical hyperprior, and Cheng *et al.* [14] achieve further improvements by adding attention modules and using a Gaussian Mixture Model (GMM) to estimate the distribution of latent representations. The current state-of-the art is achieved by [58]: They propose an invertible convolutional network, and apply residual feature enhancement as pre-processing and post-processing. Moreover, there are also plenty of methods aiming at variable rate compression, mainly including RNN-based autoencoders [54,55,30] and conditional autoencoders [15]. Besides, [4,38] propose image compression with Generative Adversarial Networks (GAN) to optimize perceptual quality.

Implicit Neural Representations. One of the early works on INRs is DeepSDF [45] which is a neural network representation for 3D shapes. In particular, they use a Signed Distance Function (SDF) to represent the shape by a field where every point in space holds the distance to the shape's surface. Concurrently to DeepSDF, multiple works propose similar approaches to represent 3D shapes with INRs, *e.g.* , the occupancy network [39] and the implicit field decoder [13]. Besides, INRs have also been used for scene representation [40], image representation [12,50] and compact representation [17].

Model Compression. In the past decades, there has been a plethora of works on model compression [36]. For instance,[26] proposes sequentially applying pruning, quantization and entropy coding combined with retraining in between the steps. Later, [2] suggests an end-to-end learning approach using a rate-distortion objective. To optimize performance under quantization, several works [24,57,18,56] use mixed-precision quantization, while others [31,21,10,35,43,42] propose post-quantization optimization techniques.

Model Weights for Instance-Adaptive Compression. Recently, [46] suggests finetuning the decoder weights of an RDAE on a per-instance basis and appending the weight update to the latent vector, thereby improving RDAEs. It is related to our work in that model weights are included in the represen-

tation, however the RDAE architecture fundamentally differs from ours. Most recently, Dupont *et al.* [19] propose the first INR-based image compression approach COIN, which overfits an INR's model weights to represent single images and compresses the INR using quantization. Importantly, COIN does not use meta-learning for initializing INRs, positional encodings for SIREN, post-quantization retraining and entropy coding. Furthermore, [9] recently proposed a compression algorithm for entire scenes based on compressing the weights of NeRF [40]. Moreover, concurrently NeRV [11] proposed to compress videos using INRs. While they use another data modality and neither use post-quantization retraining nor meta-learned initializations, their work shows the potential of INR-based compression of coordinate-based data. In another concurrent work, [20] also proposes to apply meta-learning to INR-based compression in an effort to extend COIN. However, unlike this work they do not outperform JPEG on the full resolution images on KODAK. Their performance is similar to our method absent of meta-learning and positional encodings (see Fig. 7).

3 Method

3.1 Background

INRs store coordinate-based data such as images, videos and 3D shapes by representing data as a continuous function from coordinates to values. For example, an image is a function of a horizontal and vertical coordinate (p_x, p_y) and maps to a color vector within a color space such as RGB:

$$I : (p_x, p_y) \to (R, G, B) \tag{1}$$

This mapping can be approximated by a neural network f_θ, typically a Multi Layer Perceptron (MLP) with parameters θ, such that $I(p_x, p_y) \approx f_\theta(p_x, p_y)$. Since these functions are continuous, INRs are resolution agnostic, *i.e.*, they can be evaluated on arbitrary coordinates within the normalized range $[-1, 1]$. To express a pixel based image tensor $\mathbf{x}$, we evaluate the image function on a uniformly spaced coordinate grid $\mathbf{p}$ such that $\mathbf{x} = I(\mathbf{p}) \in \mathbb{R}^{W \times H \times 3}$ with

$$\begin{gathered} \mathbf{p}_{ij} = \left(\frac{2i}{W-1} - 1, \frac{2j}{H-1} - 1 \right) \in [-1, 1]^2 \\ \forall\, i \in \{0, \dots, W-1\}, j \in \{0, \dots, H-1\}. \end{gathered} \tag{2}$$

Note that each coordinate vector is mapped independently:

$$f_\theta(\mathbf{p}) = \begin{bmatrix} f_\theta(\mathbf{p}_{11}) & \dots & f_\theta(\mathbf{p}_{1H}) \\ \vdots & \ddots & \vdots \\ f_\theta(\mathbf{p}_{W1}) & \dots & f_\theta(\mathbf{p}_{WH}) \end{bmatrix}. \tag{3}$$

Rate-distortion Autoencoders. The predominant approach in learned source compression are RDAEs: An encoder network produces a compressed representation, typically called a latent vector $\mathbf{z} \in \mathbb{R}^d$, which a jointly trained decoder network uses to reconstruct the original input. Early approaches enforce compactness of $\mathbf{z}$ by limiting its dimension d [27]. Newer methods constrain the representation by adding an entropy estimate, the so-called rate loss, of $\mathbf{z}$ to the loss. This rate term, reflecting the storage requirement of $\mathbf{z}$, is minimized jointly with a distortion term, that quantifies the compression error.

3.2 Image Compression using INRs

In contrast to RDAEs, INRs store all information implicitly in the network weights θ. The input to the INR itself, *i.e.*, the coordinate, does not contain any information. The encoding process is equivalent to training the INR. The decoding process is equivalent to loading a set of weights into the network and evaluating on a coordinate grid. We can summarize this as:

$$\arg\min_{\theta} \mathcal{L}(\mathbf{x}, f_\theta(\mathbf{p})) = \theta^\star \xrightarrow[\text{transmit } \theta^\star]{} \widehat{\mathbf{x}} = f_{\theta^\star}(\mathbf{p}). \tag{4}$$

Thus, we only need to store $\theta^\star$ to reconstruct a distorted version of the original image $\mathbf{x}$. With our approach, we describe a method to find $\theta^\star$ to achieve compact storage and good reconstruction at the same time.

Architecture. We use SIREN, namely a MLP using sine activations with a frequency $\omega = 30$ as proposed originally in [49], which has recently shown good performance on image data. We adopt the initialization scheme suggested by the authors. Since we aim to evaluate our method at multiple bitrates, we vary the model size to obtain a rate-distortion curve. We also provide an ablation on how to vary the model size to achieve optimal rate-distortion performance (see supplementary material) and on the architecture of the INR (see Section 4.4).

Input Encoding. An input encoding transforms the input coordinate to a higher dimension, which has been shown to improve perceptual quality [40,53]. Notably, to the best of our knowledge we are the first to combine SIREN with an input encoding - previously input encodings have only been used for INRs based on the Rectified Linear Unit (ReLU) activation functions. We apply an adapted version of the positional encoding presented in [40], where we introduce the scale parameter σ to adjust the frequency spacing (similarly to [53]) and concatenate the frequency terms with the original coordinate p (as in the SIREN codebase[1]):

$$\begin{aligned}\gamma(p) =& (p, \sin(\sigma^0 \pi p), \cos(\sigma^0 \pi p), \dots, \\ & \sin(\sigma^{L-1} \pi p), \cos(\sigma^{L-1} \pi p)).\end{aligned} \tag{5}$$

where L is the number of frequencies used. We investigate the impact of the input encoding in Section 4.4.

[1] https://github.com/vsitzmann/siren

3.3 Compression Pipeline for INRs

This section introduces our INR-based compression pipeline. First, we describe our basic approach based on randomly initialized INRs (Section 3.3). Then, we propose meta-learned initializations to improve the rate-distortion performance and encoding time of INR-based compression (Section 3.3). The entire pipeline is depicted in Fig. 2 and a higher level overview is shown in Fig. 1.

Basic Approach using Random Initialization. Stage 1: Overfitting. First, we overfit the INR f_θ to a data sample at test time. This is equivalent to calling the encoder of other learned methods. We call this step overfitting to emphasize that the INR is trained to only represent a single image. Given an image $\mathbf{x}$ and a coordinate grid $\mathbf{p}$, we minimize the objective:

$$\arg\min_\theta \mathcal{L}_{\text{MSE}}(\mathbf{x}, f_\theta(\mathbf{p})). \tag{6}$$

We use the Mean Squared Error (MSE) as the loss function to measure similarity of the ground-truth target and the INRs output:

$$\mathcal{L}_{\text{MSE}}(\mathbf{x}, \widehat{\mathbf{x}}) = \sum_i^W \sum_j^H \frac{\|\mathbf{x}_{ij} - \widehat{\mathbf{x}}_{ij}\|_2^2}{WH}. \tag{7}$$

Note that $\mathbf{x}_{ij} \in \mathbb{R}^3$ is the color vector of a single pixel.

Regularization. In image compression, we aim at minimizing distortion (*e.g.* , MSE) as well as bitrate simultaneously. Since the model entropy is not differentiable, we can not directly use it in gradient-based optimization. One option that has been used in literature is to use a differentiable entropy estimator during training [2]. We however choose to use a regularization term that approximately induces lower entropy. In particular, we apply L_1 regularization to the model weights. Overall, this yields the following optimization objective:

$$\mathcal{L}(\mathbf{x}, f_\theta(\mathbf{p})) = \mathcal{L}_{\text{MSE}}(\mathbf{x}, f_\theta(\mathbf{p})) + \lambda \|\theta\|_1 \tag{8}$$

where λ determines the importance of the L_1 regularization which induces sparsity. Our regularization term is related to the sparsity loss employed in [46]: we have the same goal of limiting the entropy of the weights, however we apply this to an INR, whereas they apply it to a traditional explicit decoder.

Stage 2: Quantization. Typically, the model weights resulting from overfitting are single precision floating point numbers requiring 32 bits per weight. To reduce the memory requirement, we quantize the weights using the AI Model Efficiency Toolkit (AIMET)[1]. We employ quantization specific to each weight tensor such that the uniformly-spaced quantization grid is adjusted to the value range of the tensor. The bitwidth determines the number of discrete levels, *i.e.*, quantization bins. We find empirically that bitwidths in the range of 7-8 lead to optimal rate-distortion performance for our models as shown in the supplement.

Stage 3: Post-Quantization Optimization. Quantization reduces the models

[1] https://quic.github.io/

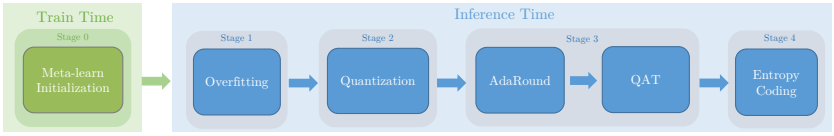


Fig. 2: Overview of INR-based compression pipeline. Blue: The basic compression pipeline comprising overfitting, quantization, AdaRound, QAT and entropy coding. Green: Additional meta-learning of initializations at training time.

performance by rounding the weights to their nearest quantization bin. We leverage two methods to mitigate this effect. First, we employ *AdaRound* [42], which is a second-order optimization method to decide whether to round a weight up or down. The core idea is that the traditional nearest rounding is not always the best choice, as shown in [42]. Subsequently, we fine-tune the quantized weights using Quantization Aware Training (QAT). This step aims to reverse part of the quantization error. Quantization is non-differentiable and we thus rely on the Straight Through Estimator (STE) [8] for the gradient computation, essentially bypassing the quantization operation during backpropagation.

Stage 4: Entropy Coding. Finally, we perform entropy coding to further losslessly compress weights. In particular, we use a binarized arithmetic coding algorithm to losslessly compress the quantized weights.

Meta-learned Initializations for Compressing INRs. Directly applying INRs to compression has two severe limitations: firstly, it requires overfitting a model from scratch to a data sample during the encoding step. Secondly, it does not allow embedding inductive biases into the compression algorithm (*e.g.* , knowledge of a particular image distribution). To this end, we apply meta-learning, i.e. Model Agnostic Meta-Learning (MAML) [23], for learning a weight initialization that is close to the weight values and entails information of the distribution of images. Previous work on meta-learning for INRs has aimed at improving mainly convergence speed [52]. The learned initialization θ_0 is claimed to be closer in weight space to the final INR. We want to exploit this fact for compression under the hypothesis that the update $\Delta\theta = \theta - \theta_0$ requires less storage than the full weight tensor θ. We thus fix θ_0 and include it in the decoder such that it is sufficient to transmit $\Delta\theta$, or, to be precise, the quantized update $\Delta\tilde{\theta}$. The decoder can then reconstruct the image by computing:

$$\tilde{\theta} = \theta_0 + \Delta\tilde{\theta}, \quad \widehat{\mathbf{x}} = f_{\tilde{\theta}}(\mathbf{p}). \tag{9}$$

We expect the value range occupied by the weight updates $\Delta\theta$ to be significantly smaller than that of the full weights θ. The range between the lowest and highest quantization bin can thus be smaller when quantizing the weight updates. At a fixed bitwidth, the stepsize in-between quantization bins will be smaller in the case of weight updates and, thus, the average rounding error is also smaller.

Note that the initialization is only learned once per distribution $\mathcal{D}$ prior to overfitting a single image. Thus, we introduce it as Stage 0. Stage 0 happens at

training time, is performed on many images and is not part of inference. Stages 1-4 happen at inference time and aim at compressing a single image. Consequently, using meta-learned initializations does not increase inference time.

Integration into a Compression Pipeline. When we want to encode only the update $\Delta\theta$, we need to adjust our compression pipeline accordingly. During overfitting we change the objective to:

$$\mathcal{L}(\mathbf{x}, f_\theta(\mathbf{p})) = \mathcal{L}_{\text{MSE}}(\mathbf{x}, f_\theta(\mathbf{p})) + \lambda \left\|\Delta\theta\right\|_1 \tag{10}$$

thus, the regularization term now induces the model weights to stay close to the initialization. Also, we directly apply quantization to the update $\Delta\theta$. In order to perform AdaRound and QAT, we apply a decomposition to all linear layers in the MLP to separate initial values from the update:

$$\begin{aligned} \mathbf{W}\mathbf{x} + \mathbf{b} &= (\mathbf{W}_0 + \Delta\mathbf{W})\mathbf{x} + (\mathbf{b}_0 + \Delta\mathbf{b}) \\ &= \underbrace{(\mathbf{W}_0\mathbf{x} + \mathbf{b}_0)}_{\text{fix}} + \underbrace{(\Delta\mathbf{W}\mathbf{x} + \Delta\mathbf{b})}_{\text{quantize \& retrain}} . \end{aligned} \tag{11}$$

This is necessary, because optimizing the rounding and QAT require the original input-output function of each linear layer. Splitting it up into two parallel linear layers, we can fix the linear layer containing $\mathbf{W}_0$ and $\mathbf{b}_0$ and apply quantization, AdaRound and QAT to the update parameters $\Delta\mathbf{W}$ and $\Delta\mathbf{b}$.

INRs for 3D Shape Compression The proposed INR-based compression pipeline is applicable to any coordinate based data modality with minimal modification. We demonstrate this for 3D shapes. A 3D shape can be represented as a signed distance function:

$$SDF : (p_x, p_y, p_z) \to d \tag{12}$$

i.e., we assign a signed distance d between each point (p_x, p_y, p_z) in 3D space and the shape surface. Here, the sign of the distance indicates whether we are inside (negative) or outside of the shape (positive). We can now simply train our INR to approximate the SDF:

$$f_\theta(\mathbf{p}) \approx SDF(\mathbf{p}). \tag{13}$$

When training INRs to estimate SDFs accurate predictions close to the surface are most important. Therefore, we adopt the sampling strategy proposed in [51].

4 Experiments

Datasets. The **Kodak** [1] dataset is a collection of 24 images containing various objects, people or landscapes. This dataset has a resolution of 768×512 pixels (vertical $\times$ horizontal). The **DIV2K** dataset introduced in [3] contains 1000 high resolution images with a width of ≈ 2000 pixels. The dataset is split

into 800 training, 100 validation and 100 test images. For our purpose of meta-learning the initialization, we resize the DIV2K images to the same resolution as Kodak (768×512). **CelebA** [34] is a dataset containing over 200'000 images of celebrities with a resolution of 178×218. We evaluate our method on 100 images that are randomly sampled from the test set. For our 3D shape compression experiment we use 5 high resolution meshes from the **Stanford 3D Scanning Repository** [16], which we normalize such that they fit into a unit cube prior to training. More details are in the supplement.

Metrics. We evaluate two metrics to analyze performance in terms of rate and distortion. We measure the rate as the total number of bits required to store the representation divided by the number of pixels $W \cdot H$ of the image:

$$\text{bitrate} = \frac{\text{total number of bits}}{WH} \quad \text{[bpp]}. \tag{14}$$

We measure distortion in terms of MSE and convert it to the Peak Signal to Noise Ratio (PSNR) using the formula:

$$\text{PSNR} = 10 \log_{10} \left(\frac{1}{MSE} \right) \quad \text{[dB]}. \tag{15}$$

Baselines. We compare our method against traditional codecs, INR based compression and learned approaches based on RDAEs.

- Traditional image compression codecs: JPEG, JPEG2000, BPG
- INR-based image compression: Dupont *et al.* [19] (COIN)
- RDAE-based image compression: Ballé *et al.* [6], Xie *et al.* [58]
- 3D mesh compression: Draco [4]

Optimization and Hyperparameters. We use a default set of hyperparameters throughout the experiment section unless mentioned otherwise. In particular, we use INRs with 3 hidden layers and sine activations combined with the positional encoding using $\sigma = 1.4$. On the higher resolution Kodak dataset, we set the number of frequencies to $L = 16$, whereas on CelebA we set $L = 12$. We vary the number of hidden units per layer M, *i.e.*, the width of the MLP, to evaluate performance at different rate-distortion operating points. We refer to our method with random initialization as the *basic* approach whereas the method including meta-learned initialization is called *meta-learned*. We found the optimal bitwidth to be $b = 7$ for the *meta-learned* approach and $b = 8$ for the *basic* approach. For additional details on the training and hyperparameters we refer to the supplementary material.

4.1 Comparison with State-of-the-Art

Full resolution. Fig. 4 depicts our results on CelebA/Kodak respectively. The proposed *basic* approach can already outperform COIN clearly over the whole

[4] https://github.com/google/draco

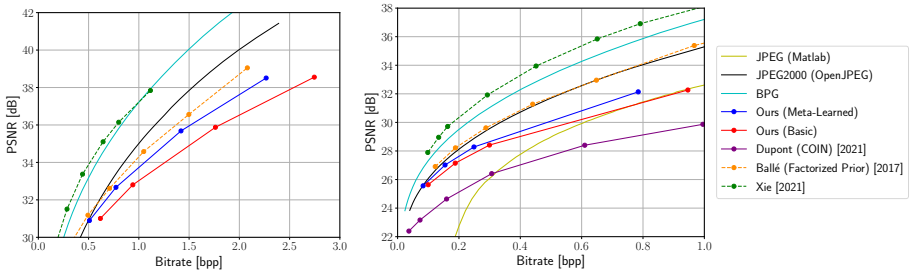


Fig. 3: Performance overview over image compression approaches including conventional (solid line), learned autoencoder (dashed line) and learned INR methods (solid line with dots) evaluated on the **CelebA** (left) **Kodak** (right) dataset.

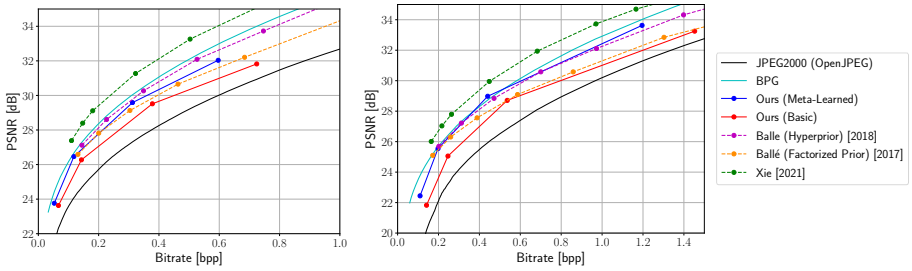


Fig. 4: Image compression approaches including conventional (solid line), RDAEs (dashed line) and learned INR-based methods (solid line with dots) evaluated on the **Kodak** dataset with image resolution reduced by a factor of two (left) and four (right). Meta-learned INRs show competitive performance in this regime.

range of bitrates. It is also better than JPEG for most bitrates, except the highest setting on CelebA. With our proposed *meta-learned* approach we improve over the *basic* approach at all bitrates. Between the two datasets, the difference is noticeably greater on the CelebA dataset. At the lowest bitrate examined the meta-learned approach reaches the performance of JPEG2000, however our approach cannot keep up with JPEG2000 at higher bitrates. On the CelebA dataset, the meta-learned approach also almost reaches the performance of an autoencoder with a factorized prior [6] at lower bitrates. Towards higher bitrates, the advantage of the autoencoder becomes clearer. BPG as well as the state-of-the-art RDAE [58] clearly outperform our method on both datasets.

Reduced image resolution. We further compare our *basic* and *meta-learned* approach with other methods on Kodak with reduced resolution (2x/4x). These image are comprised of 384×256, resp. 192×128, pixels. We observe that the *meta-learned* approach again performs strictly better than the *basic* approach. Moreover, our *meta-learned* approach demonstrates competitive performance for this image resolution outperforming all other methods, except BPG and Xie *et al.* , over the entire range of bitrates.

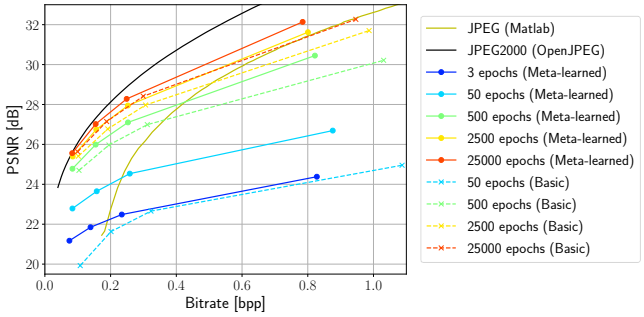


Fig. 5: Comparing the convergence speed of the meta-learned and basic approach evaluated on the Kodak dataset. The meta-learned approach converges faster, which is especially apparent in the beginning of the overfitting. After only 2500 epochs it reaches the same performance as the basic approach after 25000 epochs.

4.2 Visual Comparison to JPEG and JPEG2000

We compare compressed images of our meta-learned approach with the codecs JPEG and JPEG2000 in Fig. 6 (Kodak) and Fig. **??** (CelebA). We visually confirm that our model significantly improves over JPEG: Our model produces an overall more pleasing image with better detail and less artifacts although we operate at a lower bitrate on both images. For the Kodak image in Fig. 6 we achieve a slightly lower bitrate at the same distortion compared to JPEG2000. Visually, the JPEG2000 image shows more artifacts around edges and in regions with high frequency details. The sky is however rendered better on the JPEG2000 image because our model introduces periodic artifacts. For the CelebA image in Fig. **??** our method achieves a lower bitrate and higher PSNR than the JPEG2000 image. JPEG2000 again shows artifacts around edges (for example around the letters in the background) and smoothes out transitions from lighter to darker areas on the face. Our method produces a more natural tonal transition.

4.3 Convergence Speed

In Fig. 5 we show how the basic and meta-learned approach compare over different numbers of epochs. Especially in the beginning of the overfitting, the meta-learned approach shows significantly faster convergence. Already after the first 3 epochs, we obtain better performance than what the basic approach achieves after 50 epochs. Convergence slows down as we approach the final performance of the respective model, while the meta-learned approach maintains the advantage: It achieves the same performance after 2500 epochs as the basic approach after 25000 epochs. This amounts to a reduction in training time of 90%.

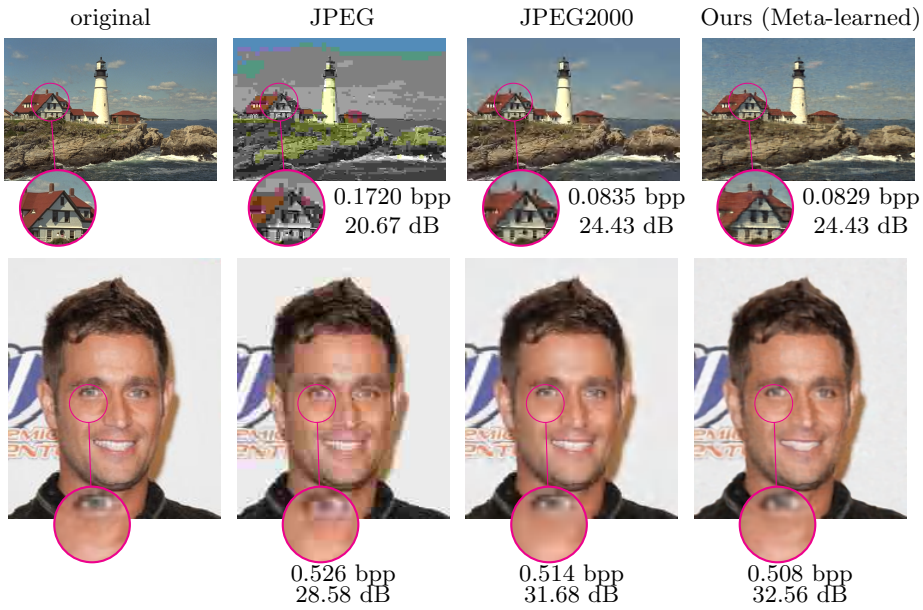


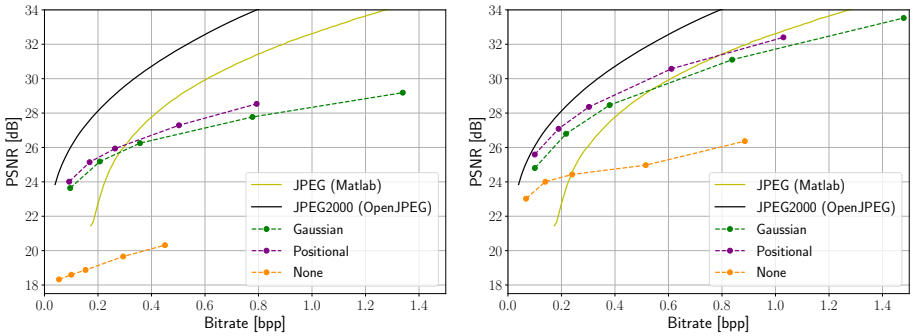
Fig. 6: Visual comparison of images compressed with JPEG (quality factor 1/13), JPEG2000 (compression factor 287/47) and our meta-learned approach on Kodak/CelebA (top/bottom). We use a model with a hidden dimension of $M = 32/24$. JPEG introduces heavy block artifacts and loss of color information resulting in the worst image in comparison. JPEG2000 shows blurring and blocking around edges. Our method maintains better local contrast but shows periodic artifacts visible in the sky as well as smearing at some edges.

4.4 Choosing Input Encoding and Activation

An important architecture choice is the combination of input encoding and the activation function used. We compare against the Gaussian encoding proposed in [53]. For this encoding we use the same number of frequencies as hidden dimensions ($L = M$) as in [53] and a standard deviation of $\sigma = 4$. We train models with different hidden dimensions ($M \in \{32, 48, 64, 96, 128\}$) and different input encodings on the Kodak dataset starting from random initializations using the regularization parameter $\lambda = 10^{-6}$.

Looking at Fig. 7a, compared to Fig. 7b we can see that the sine activation outperforms the ReLU activation in every configuration, especially at higher bitrates. The best overall input encoding is *positional* encoding beating *Gaussian* for both activations. The MLP without input encoding and sine activations, the SIREN architecture, performs significantly better than its ReLU counterpart but still cannot reach the performance of the models with input encoding.

Importantly, we investigate whether positional encoding improves SIREN in general or rather renders it more robust to quantization. Therefore, we measure the quantization error of our basic approach for different bitwidths. The result is



(a) ReLU (b) Sine

Fig. 7: Rate-distortion performance of different combinations of input encoding and activation function on the Kodak dataset.

depicted in Fig. 8. ReLU and sine activations both show a reduced quantization error when trained with positional encoding. However, the effect is most obvious in the case of SIREN. Comparing the PSNR-delta of SIREN with and without positional encoding in Fig. 8 with Fig. 7 (b) reveals that applying positional encoding makes SIREN predominantly more robust to quantization.

4.5 3D Shape Compression

To demonstrate that our algorithm is applicable to coordinate-based data beyond images, we provide an additional experiment showing its performance on the task of 3D shape compression. Since the main goal of this experiment is to show the transferability of INRs-based compression, we only train our basic approach without meta-learning on 3D shapes. We plot the chamfer distance averaged over all shapes against the storage required in Fig. 9 and compare to the algorithm Draco which is based on mesh quantization. We focus on the comparison with mesh-based compression algorithms because they also preserve a continuous surface unlike the alternative approach of point cloud compression. We require much fewer bits to encode a shape of similar quality than Draco. Further details regarding this experiment are in the supplement.

5 Conclusion

Overall, INRs demonstrated great potential as a compressed representation for images. Our main contributions, the use of meta-learned initializations and SIREN combined with positional encodings, largely improve rate-distortion performance compared to previous methods [19] performing image compression based on INRs. Moreover, our approach is the first INRs-based method that is competitive with traditional codecs over a large portion of bitrates.

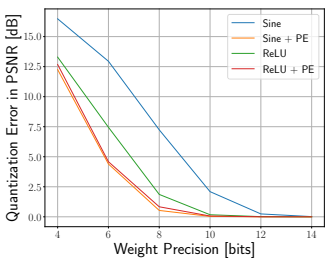


Fig. 8: Quantization error of our basic model using ReLU/-sine activations with/without positional encoding (PE).

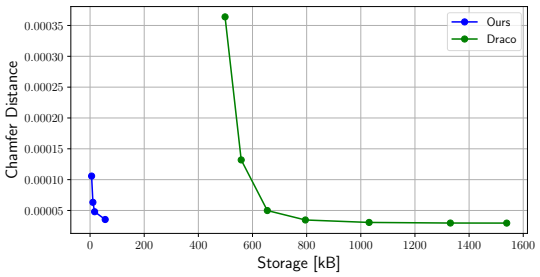


Fig. 9: Rate-Distortion performance for 3D shape compression of our method (basic) and the traditional algorithm Draco. We clearly outperform Draco.

Meta-learned initializations are superior to random initializations. Specifically, they reduce the bitrate at the same reconstruction quality. This supports the hypothesis that weight updates are more compressible. In particular, the performance gain is larger on the CelebA dataset, where the initializations are trained on an image distribution that is more similar to the test set. Moreover, the distribution of faces has less variation than the distribution of natural scenes which eases learning a single strong initialization. Consequently, we make our compression algorithm adaptive to a certain distribution by including *a priori* knowledge into the initialization.

Moreover, meta-learned initializations are a potential solution for long encoding times of INR-based compression: Our meta-learned approach can reduce training time by up to 90% at a fixed performance.

We also highlight the importance of applying input encodings in INR-based compression (see Fig. 7). This demonstrates significance of choosing the correct inductive biases for compression and is another promising future research avenue. Furthermore, the observation that input encodings render INRs more robust to quantization (see Fig. 8) has potential applications beyond compression.

Interestingly, the here proposed INR-based compression technique is competitive on lower resolution images (see Section 4.1). However, the performance falls short of RDAEs and BPG on higher resolution images. We hypothesize that processing pixels independently has inefficient scaling properties. Therefore, it is crucial for future research to develop novel architectures for INRs beyond the MLP that mitigate the current deficits at at high resolution images.

Lastly, our basic approach outperforms the traditional algorithm Draco on 3D mesh compression (see Section 4.5). Thus conducting further research into 3D shape compression based on INRs denotes a promising direction.

6 Acknowledgements

This work was partially supported by Google.

References

1. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/ 8
2. Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., Van Gool, L.: Soft-to-hard vector quantization for end-to-end learning compressible representations. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS). pp. 1141–1151 (2017) 3, 6
3. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017) 8
4. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Gool, L.V.: Generative adversarial networks for extreme learned image compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 221–231 (2019) 3
5. Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R.A., Murphy, K.: Fixing a broken ELBO. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 159–168. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/alemi18a.html 2
6. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. International Conference on Learning Representations (ICLR) (2017) 2, 3, 9, 10
7. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. International Conference on Learning Representations (ICLR) (2018) 3
8. Bengio, Y.: Estimating or propagating gradients through stochastic neurons (2013) 7
9. Bird, T., Ballé, J., Singh, S., Chou, P.A.: 3d scene compression through entropy penalized neural representation functions. In: 2021 Picture Coding Symposium (PCS). pp. 1–5. IEEE (2021) 4
10. Chai, S.M.: Quantization-guided training for compact TinyML models. Research Symposium on Tiny Machine Learning (2021) 3
11. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: Nerv: Neural representations for videos. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021) 2, 4
12. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
13. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
14. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3
15. Choi, Y., El-Khamy, M., Lee, J.: Variable rate deep image compression with a conditional autoencoder. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3146–3154 (2019) 3
16. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (1996) 9
17. Davies, T., Nowrouzezahrai, D., Jacobson, A.: On the effectiveness of weight-encoded neural implicit 3D shapes (2021) 3

18. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq: Hessian aware quantization of neural networks with mixed-precision. International Conference on Computer Vision (ICCV) (2019) 3
19. Dupont, E., Golinski, A., Alizadeh, M., Teh, Y.W., Doucet, A.: COIN: COmpression with implicit neural representations. Neural Compression: From Information Theory to Applications – Workshop (ICLR) (2021) 2, 3, 4, 9, 13
20. Dupont, E., Loya, H., Alizadeh, M., Goliński, A., Teh, Y.W., Doucet, A.: Coin++: Data agnostic neural compression. arXiv preprint arXiv:2201.12904 (2022) 2, 4
21. Fan*, A., Stock*, P., Graham, B., Grave, E., Gribonval, R., Jegou, H., Joulin, A.: Training with quantization noise for extreme model compression (2020) 3
22. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. International Conference on Machine Learning (ICLR) (2017) 2
23. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. pp. 1126–1135. PMLR (2017) 7
24. Habi, H.V., Jennings, R.H., Netzer, A.: HMQ: hardware friendly mixed precision quantization block for CNNs. European Conference on Computer Vision (ECCV) (2020) 3
25. Habibian, A., Rozendaal, T.v., Tomczak, J.M., Cohen, T.S.: Video compression with rate-distortion autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7033–7042 (2019) 2
26. Han, S., Mao, H., Dally, W.J.: Deep Compression: Compressing deep neural network with pruning, trained quantization and huffman coding. International Conference on Learning Representations, (ICLR) (2016) 3
27. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science pp. 504–507 (2006) 5
28. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and helmholtz free energy. Advances in neural information processing systems **6**, 3–10 (1994) 2
29. Hu, Y., Yang, W., Liu, J.: Coarse-to-fine hyper-prior modeling for learned image compression. Conference on Artificial Intelligence (AAAI) (2020) 3
30. Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., Jin Hwang, S., Shor, J., Toderici, G.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4385–4393 (2018) 3
31. Kim, T., Yoo, Y., Yang, J.: FrostNet: Towards quantization-aware network architecture search (2020) 3
32. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic gradient descent. In: Proceedings of the International Conference on Learning Representations (ICLR). pp. 1–15 (2015) 26, 27
33. Lee, J., Cho, S., Beack, S.K.: Context-adaptive entropy model for end-to-end optimized image compression. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 3
34. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. International Conference on Computer Vision (ICCV) (December 2015) 9
35. Louizos, C., Reisser, M., Blankevoort, T., Gavves, E., Welling, M.: Relaxed quantization for discretized neural networks. International Conference on Learning Representations (ICLR) (2019) 3
36. Menghani, G.: Efficient deep learning: A survey on making deep learning models smaller, faster, and better (2021) 3

37. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4394–4402 (2018) 2, 3
38. Mentzer, F., Toderici, G.D., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. Advances in Neural Information Processing Systems (NeuIPS) **33** (2020) 3
39. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4460–4470 (2019) 3
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. European Conference on Computer Vision (ECCV) (2020) 2, 3, 4, 5
41. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. Advances in Neural Information Processing Systems (NeurIPS) (2018) 3
42. Nagel, M., Amjad, R.A., van Baalen, M., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization (2020) 3, 7
43. Nagel, M., van Baalen, M., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. International Conference on Computer Vision (ICCV) (2019) 3
44. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019) 2
45. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
46. van Rozendaal, T., Huijben, I.A., Cohen, T.: Overfitting for fun and profit: Instance-adaptive data compression. International Conference on Learning Representations (ICLR) (2021) 3, 6
47. Shannon: A mathematical theory of communication. The Bell System Technical Journal pp. 379–423 (1948) 2
48. Sitzmann, V., Chan, E.R., Tucker, R., Snavely, N., Wetzstein, G.: Metasdf: Meta-learning signed distance functions. Advances in Neural Information Processing Systems (NeurIPS) (2020) 2, 19, 20
49. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems (NeurIPS) (2020) 2, 5
50. Skorokhodov, I., Ignatyev, S., Elhoseiny, M.: Adversarial generation of continuous images. Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
51. Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., Fidler, S.: Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 8
52. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. CVPR (2021) 2, 7

53. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems (NeurIPS) (2020) 5, 12
54. Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2016) 3
55. Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., Covell, M.: Full resolution image compression with recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5306–5314 (2017) 3
56. Uhlich, S., Mauch, L., Cardinaux, F., Yoshiyama, K., Garcia, J.A., Tiedemann, S., Kemp, T., Nakamura, A.: Mixed precision DNNs: All you need is a good parametrization. International Conference on Learning Representations (ICLR) (2019) 3
57. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3
58. Xie, Y., Cheng, K.L., Chen, Q.: Enhanced invertible encoding for learned image compression. ACM International Conference on Multimedia (2021) 3, 9, 10, 24, 26