SIGNAL COMPRESSION VIA NEURAL IMPLICIT REPRESENTATIONS

Francesca Pistilli, Diego Valsesia, Giulia Fracastoro, Enrico Magli

Politecnico di Torino

ABSTRACT

Existing end-to-end signal compression schemes using neural networks are largely based on an autoencoder-like structure, where a universal encoding function creates a compact latent space and the signal representation in this space is quantized and stored. Recently, advances from the field of 3D graphics have shown the possibility of building implicit representation networks, i.e., neural networks returning the value of a signal at a given query coordinate. In this paper, we propose using neural implicit representations as a novel paradigm for signal compression with neural networks, where the compact representation of the signal is defined by the very weights of the network. We discuss how this compression framework works, how to include priors in the design, and highlight interesting connections with transform coding. While the framework is general, and still lacks maturity, we already show very competitive performance on the task of compressing point cloud attributes, which is notoriously challenging due to the irregularity of the domain, but becomes trivial in the proposed framework.

Index Terms— Implicit representation, signal compression.

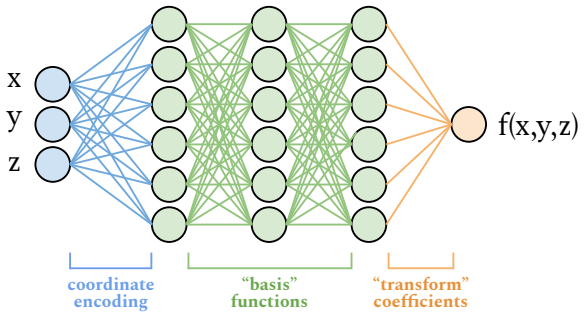


Fig. 1: A neural implicit representation network processes an input coordinate from the signal domain and returns the value of the signal at that coordinate, sharing weight values over the whole domain. The network can be loosely understood to compute a set of of basis functions optimized for the signal to be represented and mix them in the final layer, in an analogy with transform coding (see Sec. 2.2).

1. INTRODUCTION

While compression has traditionally been performed using model-based methods, several techniques based on neural networks have recently appeared and there is a growing interest in exploiting the representation capabilities of deep neural networks. The so-called end-to-end compression schemes seek to implement the entire compression pipeline using neural networks without handcrafted priors, so that optimal representations can be learned from data. The dominating archetype in the literature is the use of auto-encoder structures [1, 2, 3, 4, 5, 6], where an encoder network is trained on a representative dataset to extract a compact vector representing the input signal, and a decoder recovers an estimate of the original from the compressed information. In this paradigm, the encoder and decoder networks are universal, and the quantized and entropy-coded compact vector represents the compressed information. Some works exploit the latent space of generative models [7, 8] to obtain compact representations but they are conceptually analogous to autoencoders. Convolutional dictionary learning and its deep learning extensions [9] use a network to learn discrete atoms that can be combined to reconstruct the signal, although they have not been extensively applied to compression. However, the main idea is still to use universal encoders and store the sparse coefficients providing the signal representation.

In this paper, we examine a different *paradigm* for signal compression with neural networks. We use neural networks that take as input a single coordinate from the signal domain (e.g., the location of a pixel) and return the value of the signal at that coordinate (e.g., the RGB value of that pixel) while sharing the parameters for any input coordinate. In training, we essentially make the network overfit the signal we want to represent, so that the network itself, in its weights, biases and architecture, becomes the compact representation of our signal. The motivation for this work is given by the recent success of neural implicit representations in 3D rendering problems [10, 11, 12], where it has been shown that recent architectures of this kind built as coordinate-based multilayer perceptrons, with periodic activation functions [13] or specific embedding layers [14], can successfully fit the high-frequency components of signals, which was previous thought to be challenging. We present the general framework of this novel compression paradigm, which we refer to as NIC (Neural Implicit Compression), formalize its connection with transform coding, and highlight the importance of efficient coding of the weights of the neural network, as the rate of our representation depends entirely on that. We also show how priors can be introduced by means of meta-learning, i.e., providing a universal pretraining of the network for a given class of signals. Finetuning on the signal of interest then provides the final value of the weights, which can be differentially encoded with respect to their universal initializations, with significant rate savings with respect to a random initialization.

This is a preliminary exploration of this new compression paradigm and we highlight a number of interesting research questions that can significantly improve our basic design, if answered. Nevertheless, we show extremely promising experimental results on a sample application, namely compression of point cloud attributes. This application fully leverages the advantages of the new paradigm, because the irregularity of the domain makes it difficult to design traditional compression methods, while it can be trivially handled by our framework which reaches performance close to the latest MPEG G-PCC standard (v12.0 test model) [1].

[1] https://github.com/MPEGGroup/mpeg-pcc-tmc13

2. NEURAL IMPLICIT COMPRESSION

2.1. Neural Implicit Representations of signals

Neural implicit representations seek to represent arbitrary continuous functions by means of a neural network, typically a multilayer perceptron (MLP), where the input of the network is a coordinate $\mathbf{x}$ from the domain of the function and the output is the corresponding value of the function $f(\mathbf{x})$ (see Fig.1). A signal is a sampled version of $f(\mathbf{x})$ on a regular or irregular domain, e.g., an image, a point cloud, etc.. The weights and biases of the network are shared for different input coordinates and are all the information required to represent the signal. Recent advances have shown how designs using sinusoidal activation functions ("SIRENs" [13]) or the use of Fourier embeddings as the first layer [14] significantly improve performance by allowing the network to successfully fit the high frequency components of the signal.

While existing works have mostly focused on using this machinery to address 3D rendering problems, we are interesting in investigating the *efficiency* of such neural implicit representations. In particular, we show how they can serve as *compressed* signal representations, whose efficiency can be measured in rate-distortion terms.

We remark how NIC is a different *paradigm* with respect to existing works based on autoencoders. In autoencoders, the encoder/decoder networks are universal, and the compact vector is the signal representation, i.e., what needs to be saved. On the other hand, a neural implicit representation is a network that is trained to represent a specific signal and the network itself needs to saved as it *is* the signal itself. This paradigm has potential advantages in optimizing the representation for the specific instance that is being compressed rather than relying on a training dataset that may or may not be perfectly representative of the signals to be compressed. The use of coordinate-based MLPs can also simplify some challenging compression tasks. As an example, compression of point cloud attributes is challenging because the signal is supported on an irregular domain (the scattered points in the geometry), while training a coordinate-based MLP mapping from a point coordinate to the corresponding attribute value is trivial.

In the remainder of the paper we will focus on using the recently proposed SIREN implicit representations [13], which are an MLP with sinusoidal activation functions.

2.2. Connections with transform coding

The following result shows that there is a strong connection between neural implicit representations and transform coding by identifying a one-to-one correspondence between the global minimum of a two-layer SIREN with sufficient capacity and the well-known discrete cosine transform (DCT) [15].

Proposition 2.1. *A two-layer SIREN approximating a continuous function $f : \mathbb{R} \to \mathbb{R}$ with N hidden features is equivalent to an N-point 1D-DCT.*

Proof. Given an input scalar coordinate x, we can define the corresponding output y of the two-layer SIREN as

$$y = \sum_{k=0}^{N-1} w_k^{(2)} \sin(w_k^{(1)} x + b_k^{(1)}) + b_k^{(2)}$$

where $\mathbf{b}^{(i)}$ is the bias of the i-th layer, and $\mathbf{w}^{(1)} \in \mathbb{R}^N$ and $\mathbf{w}^{(2)} \in \mathbb{R}^N$ are the weights of the first and second layer of the network, respectively. If we suppose that the input coordinates of the SIREN are sampled from a regular grid (i.e., $x = n$ where $0 \leq n \leq N-1$) and the loss function is the MSE between the output of the network y and the corresponding value of the continuous function $f(x)$ sampled on that grid, we can observe that the global minimum of the loss function is reached when the SIREN converges to the inverse of the N-point 1D-DCT, which is defined as follows

$$y_n = \sum_{k=0}^{N-1} \hat{y}_k \cos\left(\frac{\pi}{N} k \left(n + \frac{1}{2}\right)\right) \quad \text{with } 0 \leq n \leq N-1,$$

where $\hat{y}_k$ is the k-th DCT coefficient of the signal y. Therefore, the two-layer SIREN reaches the global minimum (zero MSE), i.e., converges to the inverse of the N-point 1D-DCT, when $w_k^{(2)} = \hat{y}_k$, $w_k^{(1)} = \frac{\pi}{N} k$, $b_k^{(1)} = \frac{\pi}{2N} k + \frac{\pi}{2}$, and $b_k^{(2)} = 0$. □

It is interesting to notice how this suggests that the first layer acts a basis function generator, while the last layer learns how to combine the basis functions with suitable weights akin to transform-domain coefficients. However, notice that this is only a loose interpretation of what happens when deeper networks when fewer features are used.

2.3. Signal compression

Compressing a signal with a neural implicit representations requires to perform the following basic operations:

1. Define the architecture of the coordinate-based MLP to be used; this step requires the choice of a network design (e.g, SIRENs), as well as suitable sizing of the model in terms of number of parameters and layers. The size of the model directly affects the accuracy of signal fit, but scaling laws are still unclear (e.g., whether depth or width is more important) as this is the first work looking at the efficiency of such representations. Preliminary empirical evidence from our experiments seems to suggest a preference for width instead of depth.
2. Train the network by minimizing a suitable regression loss. For example, denoting as $\mathbf{x}$ a sampled input coordinate and with $y_{\mathbf{x}}$ the signal value at that coordinate, the network parameters θ can be learned by minimizing the MSE:
$$\hat{\theta} = \arg\min_{\theta} \sum_{\mathbf{x}} \|f_\theta(\mathbf{x}) - y_{\mathbf{x}}\|_2^2$$
3. Compactly represent the weights of the network. Since the rate of the compressed representation entirely relies on coding the values of the weights and biases, they must be represented in the most efficient way. Techniques like network sparsification [16] and quantization [17] can be used to reduce the number of parameters and/or their precision. As an example, a uniform scalar quantizer can be applied to the parameters of each layer.
4. Use an entropy encoder on the quantized weights and biases and save any required side information (e.g., sparsification pattern, or quantization step sizes)

Decoding the compressed signal simply amounts to performing a forward pass through the sparse/quantized network, for all the coordinates of interest (e.g., all the pixels in a grid of arbitrary resolution). This "random access" property, where any coordinate can be queried for the corresponding signal value, also implies that decoding at different resolutions or on irregular grids is trivial.

2.4. Priors via Meta Learning

Implicit neural networks can be used to learn any signal but they are not able to exploit any prior knowledge about data properties. To overcome this problem we propose to exploit meta-learning techniques. Meta-learning [18] is a framework to address few-shot learning problems by exploiting previously gained experience over a task to improve future learning performance. In particular, in our case, a first training of the network is performed over a collection of data in order to find a good initialization point that improves the results for previously unseen data over the same task. The goal is to learn some low-level properties of the data of interest, such as smoothness over the domain, or inherit some common characteristics of a specific class of data. In [19] the idea of using well-known meta-learning algorithms to learn the initial weight parameters of coordinate-based neural representations is presented with the purpose of speeding up the convergence.

Our idea is to exploit a meta-learned initialization to improve the rate-distortion efficiency of the representation. For instance, consider a very simple class of signal as images of faces from a public dataset such as FFHQ [20]. Ideally, the meta-learning pretraining on this dataset learns commonly recurring patterns in the class, such as eyes, nose or mouth and provides an initialization that already contains these elements. The subsequent finetuning over a novel image would benefit from the insightful initialization and the final values of the weights of the network would not deviate much from their initial values. To exploit this, we propose an innovative compression methodology where the difference between the final weights of the network and their meta-learned initialization is quantized and used to recover the signal instead of directly quantizing the final values. This achieves significant rate savings because the meta-learned initialization is a strong predictor of the final weight value. Also notice that the meta-learned initialization is universal, as it can be recreated from the public dataset or written in a standard, so that it does not require to be encoded.

There are several meta-learning algorithms [18, 21] in the literature, and we exploit a simplified variant of Model-Agnostic Meta Learning (MAML) [18]. In our algorithm, one signal at a time (for instance an image) is fed into the coordinate-based network and few steps of inner optimization are performed. After the inner training, an outer optimizer is updated and a new inner loop with a novel input signal is started.

3. EXPERIMENTS

In this section, as an example application, we evaluate the performance of the NIC paradigm in the context of point cloud attribute compression. The implicit neural network considered is a SIREN that takes as input the 3D coordinates of a point cloud and provides as output the corresponding RGB values.

3.1. Experimental setting

After choosing a suitable design of the SIREN in terms of number of layers and parameters, the network is pretrained with the meta-learning algorithm from Sec. 2.4. Stochastic Gradient Descend (SGD) with learning rate equal to 0.01 and Adam with 10^{-5} have been used as inner and outer optimizers respectively. The network is trained with 10 inner steps for 1000 iterates. The meta-learning dataset exploited for pretraining is a collection of point clouds from the Microsoft Voxelized Upper Bodies dataset [22]. After obtaining this initialization, a finetuning is performed specifically for each point cloud to be compressed from the test set. The Adam optimizer has been employed with a fixed learning rate equal to 10^{-5}. Each network is trained for 20000 iterations by minimizing a loss function that is based on the MSE between the original colors and the predicted ones in the YUV space:

$$L_{\mathrm{Tot}} = \alpha \mathrm{MSE}_{\mathrm{Y}} + \beta \mathrm{MSE}_{\mathrm{U}} + \gamma \mathrm{MSE}_{\mathrm{V}}, \tag{1}$$

where α, β and γ are coefficients aimed modulating the relative importance of luminance and chrominance. In our experiment we use $\alpha = 0.6, \beta = 0.2, \gamma = 0.2$ to slightly promote luminance, as common practice in traditional codecs. After finetuning, the network parameters are differentially encoded with respect to the meta-learned intialization. The differences are quantized with a uniform scalar quantizer, where the quantization step size is adapted layer-by-layer on the basis of the dynamic range of the parameters belonging to each layer, and entropy-coded with an arithmetic encoder.

3.2. Experimental results

The test set to be compressed is composed by four point clouds from the 8i Voxelized Full Bodies dataset [23]: `Loot_vox10_1200`, `Longdress_vox10_1300`, `Redandblack_vox10_1550`, `Soldier_vox10_0690`. Notice that they are strictly disjoint from the data used by meta-learning. In order to obtain a rate-distortion curve several networks have been trained, where different numbers of layers, features and quantization step sizes allow to reach different rate-distortion points. We have tested networks with 60, 80, 130 and 170 features per layer, with 5, 7 or 9 hidden layers, and quantization step sizes corresponding to a number of levels from 2^2 to 2^{12}.

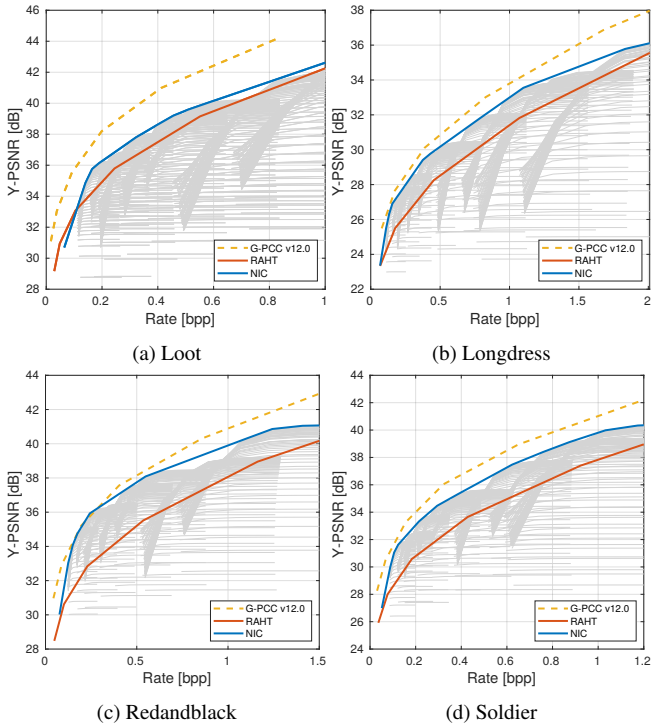
Fig.2 compares the results of the proposed method with the latest version of the MPEG G-PCC standard (v12.0 test model), and with the Region-Adaptive Hierarchical Transform (RAHT) [24], a recent algorithm that exploits a hierarchical transform based on Haar wavelets. The grey lines in the figure show all the operating points that can be obtained by different networks at different training iterations with different quantization levels. The performance of the proposed method is determined by the envelope of these curves as one wold design the most suitable network for the target rate/quality. Performance is measured in terms of Y-PSNR. From Fig.2 it can be seen that NIC significantly improves over RAHT, and reaches performance close to G-PCC v12.0, especially at low rates. This is especially significant because the proposed method is a proof-of-concept that has not been as extensively optimized, and, indeed, there are several developments that may further improve its performance (also see Sec. 4).

Also notice that our training directly promotes the luminance channel over the chrominance, as shown in Eq.(1); for completeness, we also evaluate the total PSNR (on Y, U and V). Table 1 shows such result by reporting the BD-Rate with respect to RAHT, confirming that our method still improves over RAHT even when the distortion on chrominance is taken into account.

3.3. Effectiveness of differential meta-learning

We study the effectiveness of meta-learning by assessing the rate-distortion performance achieved by various weight coding options.

First of all, in Fig.3 the first hidden layer of the network with 80 features and 9 hidden layers trained over the `Redandblack_vox10_1550` point cloud is considered. In particular, the distribution of the finetuned weights is compared with the distribution of their difference with respect to the meta-learned



(a) Loot (b) Longdress

(c) Redandblack (d) Soldier

Fig. 2: Attribute compression rate-distortion performance. Average BD-Rate: NIC vs. RAHT= -32.07%, NIC vs. G-PCC v12.0 = 47.21%.

Table 1: BD-Rate over the total PSNR of NIC versus RAHT.

Loot	Longdress	Redandblack	Soldier
-10.23 %	-38.88 %	-22.10 %	-33.39 %

initialization. The advantage of differential weight compression is clearly visible, as it results in a distribution with significantly lower variance, thus leading to increased rate-distortion efficiency.

In Fig.4 several methods are compared by means of rate-distortion curves: the proposed method with differential compression with respect to the universal meta-learned initialization; classic network compression, i.e., directly quantizing the final values of the weights; and a network with random initialization and quantization of the final values of the weights. It is clearly visible that the differential coding strategy increases the rate-distortion efficiency, showing how meta-learning effectively supplies prior information about the signal characteristics. Notice how meta-learning alone without differential compression does not significantly improve the performance over random initialization. This is due to the fact that both networks converge to similar quality levels, albeit with different speeds. Finally, notice how differential coding also has a beneficial effect on quality, rather than rate alone, because the distortion introduced on weights is nonlinearly related to final distortion on the point cloud attributes.

4. DISCUSSION AND FUTURE DEVELOPMENTS

In this paper we introduced a novel paradigm for signal compression by means of neural networks consisting in representing the signal

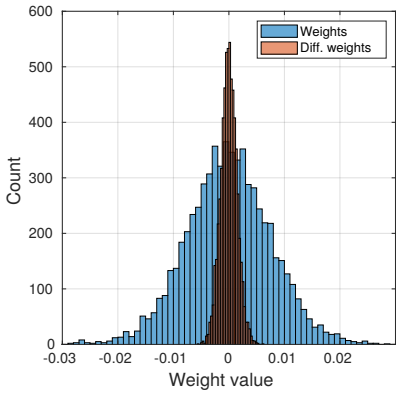
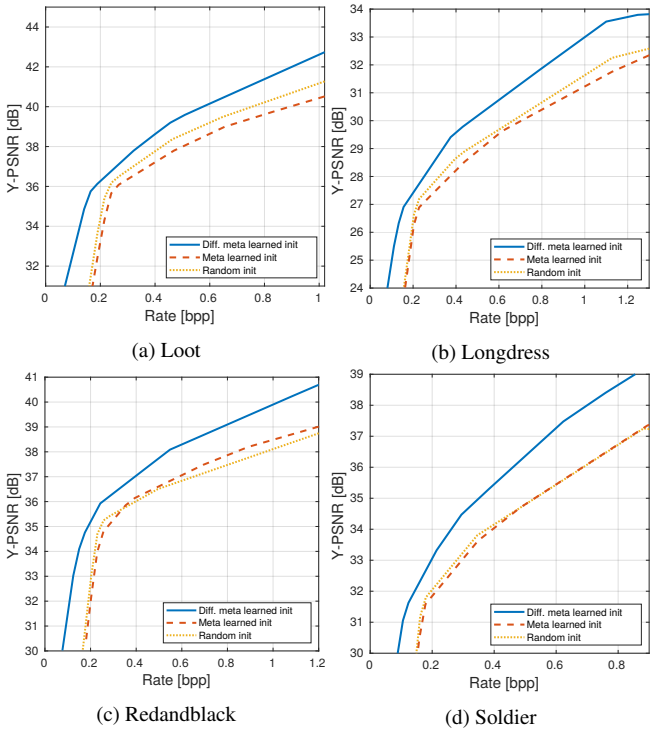


Fig. 3: Distribution of weight values against weight differences.



(a) Loot (b) Longdress

(c) Redandblack (d) Soldier

Fig. 4: Rate-distortion effectiveness of differential meta-learning.

through the weights and biases of a neural implicit representation. The technique is appealing as it can simplify and optimize the design of compression schemes for the most challenging data types. There are several open questions that can open multiple lines of research for this topic and, possibly, significantly improve performance with respect to our early results. In particular, if the early portion of the network learns basis functions for the signal, is it possible to pretrain it to be more universal, something akin to dictionary learning, and possibly saving rate? Moreover, how does one optimally prune and quantize the network weights without drastically lowering quality? Can priors be incorporated in ways other than meta-learning, e.g., via new architectures but still retaining the "random-access" property of using a single-coordinate input?

5. REFERENCES

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2016.

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

[3] David Minnen, Johannes Ballé, and George Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *arXiv preprint arXiv:1809.02736*, 2018.

[4] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Chunlei Cai, Li Chen, Xiaoyun Zhang, and Zhiyong Gao, "End-to-end optimized roi image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 3442–3457, 2020.

[6] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux, "Folding-based compression of point cloud attributes," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3309–3313.

[7] Bowen Liu, Ang Cao, and Hun-Seok Kim, "Unified signal compression using generative adversarial networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3177–3181.

[8] Bowen Liu, Yu Chen, Shiyu Liu, and Hun-Seok Kim, "Deep learning in latent space for video prediction and compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 701–710.

[9] Cristina Garcia-Cardona and Brendt Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366–381, 2018.

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[11] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth, "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla, "Deformable neural radiance fields," *arXiv preprint arXiv:2011.12948*, 2020.

[13] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[14] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[15] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[16] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485–532, 2020.

[17] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds. 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, PMLR.

[19] 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 *CVPR*, 2021.

[20] Tero Karras, Samuli Laine, and Timo Aila, "A style-based generator architecture for generative adversarial networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4396–4405.

[21] Alex Nichol, Joshua Achiam, and J. Schulman, "On first-order meta-learning algorithms," *ArXiv*, vol. abs/1803.02999, 2018.

[22] C. Loop, Q. Cai, S.O. Escolano, and Philip A. Chou, "Microsoft voxelized upper bodies – a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, May 2016.

[23] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A. Chou, "8i voxelized full bodies - a voxelized point cloud dataset," in *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, January 2017.

[24] Ricardo L. de Queiroz and Philip A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016.