KAN COIN: Komolgorov-Arnold Networks for Compression with Implicit Neural Representations

Jeremy Vonderfecht *vonder2@pdx.edu*
Department of Computer Science
Portland State University

Abstract

We apply the new Komolgorov-Arnold Network (KAN) architecture to the task of Compression with Implicit Neural Representations, or COIN. We train KANs as implicit neural representations of images, quantize their weights, and use those quantized weights as lossy representations of the original images. Using the Kodak dataset as a benchmark, we compare against the original, MLP-based COIN implementation by Dupont et al. (2021). We find that KANs are more robust to quantization than MLPs, but less parameter efficient, and can achieve comparable rate-distortion tradeoffs.

1 Introduction

in 2021, Dupont et al. (2021) introduced COIN, a lossy compression method based on encoding a signal using implicit neural representations. Their method outperformed JPEG2000 compression in the low-bitrate regime, and sparked interest in the topic of implicit neural representations for compression. Almost all implicit neural representations are fundamentally based upon multilayer perceptrons (MLPs). In this paper, we offer a preliminary exploration of Komolgorov-Arnold Networks (KANs) as an alternative fundamental architecture for implicit neural representations. To offer fair comparison of KANs' potential for this application, we attempt to make a roughly equivalent effort to optimize KANs as Dupont et al. (2021) made to optimize MLPs on this task.

While our results should be treated as preliminary, we do not find any significant advantage for KANs over MLPs on this task. Our results suggest that KANs are less parameter-efficient than SIRENs as implicit neural representations, but are more robust to quantization. Overall, KAN COIN achieves roughly-equivalent-to-slightly-inferior rate-distortion curves compared to the original COIN method.

2 Related Work

Compression with implicit neural representations, or COIN (Dupont et al., 2021), trains small SIREN networks to represent the images in the KODAK dataset (Kodak, 1991). They quantize the weights of those SIRENs to half precision, and use the half-precision weights as a compressed representation of the original image, achieving PSNRs which outperformed JPEG2000 in the low-bitrate regime, around 0.3 bits per pixel. Their method is very simple, and is commonly used as a baseline when evaluating more sophisticated methods. Subsequent works build on this idea in a variety of ways, with meta-learning, pruning, and more sophisticated quantization schemes (Dupont et al., 2022; Schwarz & Teh, 2022; Lee et al., 2021; Strümpler et al., 2022; Lee et al., 2021; Ramirez & Gallego-Posada, 2022; Guo et al., 2023; Gao et al., 2023). Gordon et al. (2023) is of particular interest for our project, since we also employ a more sophisticated post-training quantization scheme in our KAN-based compressed representations.

Komolgorov-Arnold Networks, or KANs, (Liu et al., 2024), are a new fundamental architecture for neural networks, which present an alternative to multilayer perceptrons (MLPs). Liu et al. (2024) argue for several theoretical advantages to KANs over MLPs, ... SUCH AS? In our experiments, we use an implementation of KANs from `https://github.com/Blealtan/efficient-kan`. To the best of our knowledge, we are the first to apply KANs to the COIN task, but there have already been several interesting attempts to use KANs as implicit neural representations for NeRFs (Li, 2024; Delin Qu, 2024). Note that, for these applications, KANs also appear not to confer any significant advantage over MLPs.

3 Methods

Vonderfecht & Liu (2024) observed that the particular image contents had little effect on the optimal COIN hyperparameters. In other words, it is difficult to "overfit" COIN hyperparameters to a single target image. Therefore, we tune our KAN method on a single image from the KODAK dataset. This is both simpler, and ensures that our final compression technique is *not* overfit to the rest of the data.

The orginal COIN method used full-batch training, but given the memory requirements of KANs, we train with a batch size of 10,000 pixels per iteration.

3.1 KANs are Robust to Quantization

Dupont et al. (2021) find that they can truncate the SIREN's learned weights from 32-bit to 16-bit precision with negligible loss of PSNR. We find that KANs are even more robust to post-training quantization, and can be quantized down to 9 bits per weight while losing less than 0.1 PSNR. SIRENs lose >1 PSNR at this level of quantization. Therefore, to achieve the same compression ratio, we train our KANs with twice as many trainable parameters as the SIREN, an then quantize those parameters to half as many bits. Since each successive doubling of the number of trainable parameters results in an improvement of roughly 2 PSNR (Vonderfecht & Liu, 2024), this strategy confers a 1.9 PSNR advantage over the default strategy of mirroring SIREN quantization.

3.2 Hyperparameter Search

For our first pass of hyperparameter optimization, we train 10,000 KANs for 200 training iterations, randomly sampling from the following hyperparameters:

- **Optimizer**: LBFGS or Adam
- **Learning Rate**: between 0.01 and 1 (log uniform sampling)
- **Layer Size**: 5 to 50 units per layer
- **Number of Hidden Layers**: 1 to 8 layers
- **Grid Size**: Number of spline control points; chosen such that the total number of trainable parameters is as close to 15,000 as possible, without going over.

We looked at the 100 highest PSNR networks from this sweep and made following observations:

- Adam achieved comparable PSNR to LBGFS, in dramatically less training time.
- With Adam, the best learning rates were concentrated near our lower bound of 0.01, indicating that we should try lower learning rates.
- The best-performing networks all had 1 to 3 hidden layers.
- The best-performing networks all had 10 to 20 units per layer.

For our second sweep, we trained another 10,000 KANs for 10,000 training iterations each, sampling from the following hyperparameters:

- **Optimizer**: only Adam

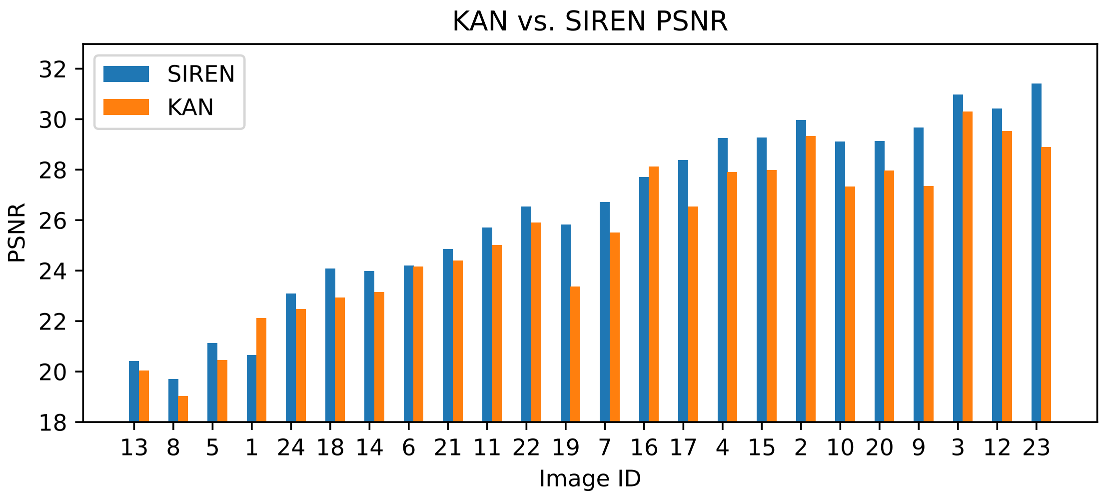

Figure 1: KAN vs. SIREN PSNR at 0.3 BPP on images from the kodak dataset

Figure 2: SIREN- vs. KAN-based COIN implementations.

- **Learning Rate**: between 0.0001 and 0.01 (log uniform sampling)
- **Layer Size**: 10 to 20 units per layer
- **Number of Hidden Layers**: 1 to 3 layers

From this hyperparameter sweep, our best resulting PSNR was 30.3. The top 100 networks from the sweep all shared the following properties:

- learning rates of approximately 0.001
- layer sizes between 12 and 16
- 3 hidden layers

4 Results

Figure 1 shows the PSNRs obtained by SIREN- and KAN-based COIN networks which represent images at 0.3 BPP, trained for the same amount of wall-clock time. SIREN based coins are quantized to 16 bits, KAN-based ones are quantized to 9 bits. On average, the KANs do about 0.9 PSNR worse than the SIRENs. Figure 2 compares visual results between KAN and SIREN based COINs.

5 Discussion

We consider this work as a preliminary investigation into the potential of KANs as parameter-efficient implicit neural representations. We do not find any advantage over traditional MLPs.

It remains very possible that some simple modification to the KAN architecture will give KANs the advantage. For example, it's possible that an alternative parameterization of KAN's scalar functions will improve performance. In Appendix A.1, we experiment with Fourier coefficients and find that these underperform the default spline parameterization. However, there are many more parameterizations to try, such as radial basis functions, wavelets, various polynomial bases, etc. However Li (2024) suggests that none of these approaches beat the parameter efficiency of traditional MLP-based NERF approaches. Therefore, we chose to conclude and publish this preliminary exploration of KAN COINs without further experiments.

References

Qizhi Chen Delin Qu. *Hands-On NeRF with KAN*, 2024. URL `https://github.com/Tavish9/KANeRF`.

Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with implicit neural representations. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*, 2021.

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Yee Whye Teh, and Arnaud Doucet. COIN++: Neural compression across modalities. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Harry Gao, Weijie Gan, Zhixin Sun, and Ulugbek S Kamilov. Sinco: A novel structural regularizer for image compression using implicit neural representations. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

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*, pp. 341–350, 2023.

Zongyu Guo, Gergely Flamich, Jiajun He, Zhibo Chen, and José Miguel Hernández-Lobato. Compression with bayesian implicit neural representations. *arXiv preprint arXiv:2305.19185*, 2023.

Kodak. Kodak dataset, 1991. URL `http://r0k.us/graphics/kodak/`.

Jaeho Lee, Jihoon Tack, Namhoon Lee, and Jinwoo Shin. Meta-learning sparse implicit neural representations. *Advances in Neural Information Processing Systems*, 34:11769–11780, 2021.

Linfei Li. X-kanerf: Kan-based nerf with various basis functions. `https://github.com/lif314/X-KANeRF`, 2024.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Juan Ramirez and Jose Gallego-Posada. L _0 onie: Compressing coins with l _0-constraints. *arXiv preprint arXiv:2207.04144*, 2022.

Jonathan Schwarz and Yee Whye Teh. Meta-learning sparse compression networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=Cct7kqbHK6`.

Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pp. 74–91. Springer, 2022.

Jeremy Vonderfecht and Feng Liu. Predicting the encoding error of sirens. *Transactions on Machine Learning Research*, 2024.