Appendix of SHACIRA - Scalable HAsh-grid Compression for Implicit Neural Representations

Sharath Girish
University of Maryland
sgirish@cs.umd.edu

Abhinav Shrivastava
University of Maryland
abhinav@cs.umd.edu

Kamal Gupta
University of Maryland
kampta@cs.umd.edu

1. Probability models

We define the probability models similar to [1]. The underlying probability density of the latents $\widehat{\mathbf{Q}}$ is defined by its Cumulative Density Function (CDF) $c : \mathbb{R} \rightarrow [0, 1]$, with the constraints:

$$c(-\infty) = 0, \quad c(\infty) = 1, \quad \frac{\partial c(x)}{\partial x} \geq 0 \tag{1}$$

This is represented using MLPs which take in a real valued scalar and output a CDF value between 0 and 1. Each dimension in $\widehat{\mathbf{Q}}$ is represented by a separate model. To satisfy the monotonicity constraint, [1] use a combination of tanh and softplus activations for each layer of the MLP. A sigmoid activation is used at the final layer to constrain the CDF between 0 and 1. To model the true distribution, the standard uniform distribution $n \sim \mathcal{U}[-0.5, 0.5]$ is convolved with the density model to derive the Probability Mass Function (PMF) of the latents as

$$P_d(x) = c\left(\widehat{\mathbf{Q}}_d + \frac{1}{2}\right) - c\left(\widehat{\mathbf{Q}}_d - \frac{1}{2}\right) \tag{2}$$

The entropy regularization loss is then the self information loss given by

$$\mathcal{L}_I(\widehat{\mathbf{Q}}) = -\frac{1}{T}\sum_{d=1}^{D}\sum_{i=1}^{T} \log_2\left(P_d(x)\right) \tag{3}$$

2. Experimental settings

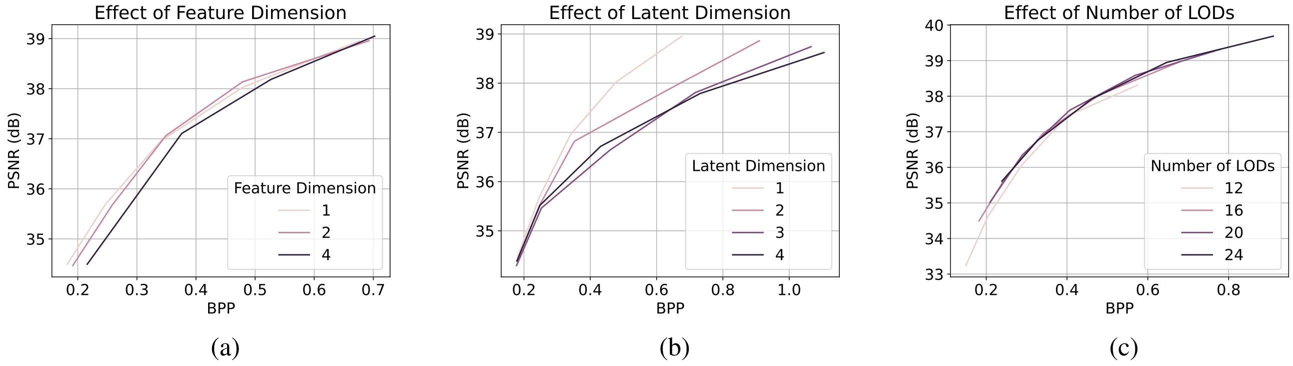
The latent dimension is set to 1 for images and video and 2 for 3D experiments. The feature dimension obtained after decoding the latents is set to 1 for Kodak and UVG-F, 2 for high resolution (giga-pixel) images and UVG for videos, and 4 for 3D experiments. Since we do not compress MLP weights but include their floating-point size in our PSNR-BPP tradeoff calculation, we vary the hidden dimension according to the signal resolution. For images, we set the dimension to be 16 for Kodak, 48 for UVG-F and SMACS, and 96 for the remaining high-resolution images. For videos and 3D, we set the layer size to 128. The number of layers is fixed to 2 for all cases. Note that higher layer size generally leads to better PSNR at the cost of a proportionally higher BPP but further gains can be obtained by compressing these weights as well. This is orthogonal to our direction of compression of the feature grid itself.

For all our experiments, we initialize the decoder parameters with a normal distribution $\mathcal{N}(0, 0.1)$, latents with $\mathcal{U}(-0.01, 0.01)$, MLP weights with the Xavier initialization [2], and probability model parameters as in [1].

We use the Adam optimizer for jointly optimizing all network parameters. We set the learning rate of MLP parameters to be $1.0e^{-3}$ for Kodak and $5.0e^{-4}$ for all other experiments. The learning rate for the probability models is fixed at $1.0e^{-4}$ following [3]. The decoder learning rate is set to 0.01 for images, video and 0.1 for 3D experiments. We set the learning rate of the latents to be 0.01 for Kodak, UVG-F, and UVG, 0.05 for 3D, and 0.1 for the other higher-resolution images. We observe that the training is not very susceptible to variations in the initialization or learning rates and the values are obtained from a coarse search for each signal domain.

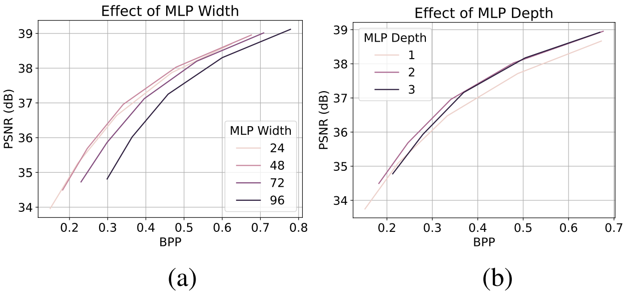
3. Sensitivity Analysis

In this section, we analyze the effect of various components of our pipeline. We pick the Jockey image (1080 × 1920) from UVG-F as the benchmark for our analysis. We set the default values of entropy regularization to $1.0e^{-4}$, latent and feature dimensions to 1 each, MLP width to 48, MLP depth to 2, and annealing period to 0.95. Each subsection below analyzes varying a single parameter while keeping others fixed to their default values. For the analysis, we compare tradeoff curves by increasing the number of entries from 2^{13} to 2^{17} in multiples of 2 which provides a natural way of increasing the number of parameters and subsequently the PSNR and BPP. Note that better tradeoff curves indicate shifting upwards (higher PSNR) and to the left (lower BPP).



(a) (b) (c)

Figure 1: Effect of latent and feature dimension and number of LODs. Increasing the latent dimension leads to a much higher size at smaller increases in PSNR. Optimal value of feature dimension (or decoded latents) is 1 or 2. Increasing the number of LODs shifts the curve upwards and to the right yielding no difference in tradeoff curves.



(a) (b)

Figure 2: Effect of MLP Width and Depth. Increasing MLP hidden layer size (width) is detrimental due to a large increase in uncompressed parameters (and BPP) with a small increase in PSNR. Increasing the number of hidden layers beyond 2 has little effect on the tradeoff curve.

3.1. Effect of Latent and Feature Dimension

We ablate the number of LODs, the number of latent dimensions, and feature dimensions after decoding the latents. Results are shown in Fig.1. We see that increasing the feature dimension from 1 to 2 does not significantly alter the tradeoff curve while increasing it to 4 leads to a small drop in performance. On the other hand, increasing the latent dimension has a strong impact on the tradeoff curve as it directly impacts the number of latents entropy encoded. BPP visibly increases with higher latent dimension but leads to only modest gains in PSNR. Increasing the number of LODs shifts the curve upwards and to the right but has no overall impact in terms of improving PSNR for a fixed BPP.

3.2. Effect of MLP Width and Depth

Finally, we analyze the effect of the hidden dimension (width) of the MLP as well as the number of layers (depth) in Fig.2. Increasing the depth from 1 to 2 shows a marginal improvement while further increases do not have a large effect. This shows that the MLP's representation capability caps at a certain value as the majority of parameters are present in the feature grid. Increasing MLP width on the hand leads to a clear drop in performance as the number of uncompressed parameters in the MLP increases approximately quadratically leading to a larger BPP (shifting to the right) but with only small gains in PSNR (small shift up).

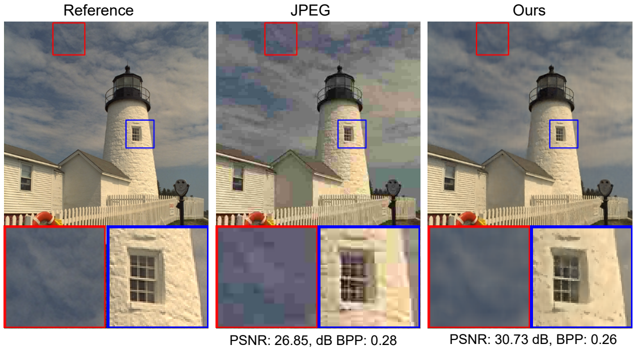


Figure 3: Qualitative results on Kodak. We significantly outperform JPEG in the low BPP regime ($+4dB$) at similar BPP ($\sim$0.26). JPEG fails to capture the intricate details in the image such as the window of the building or the clouds in the sky, exhibiting severe artifacts.

4. Feature grid visualization

We visualize the learned latents after training on the Beauty image from UVG-F in Fig.4. With increasing LOD or feature resolution (from left-to-right and top-to-bottom), we see that finer details of the image are captured. Thus, the latents represent the image features at different scales/levels. This can be particularly useful for downstream tasks which may require features at different scales. Additionally, such a hierarchy enables the application of our method in streaming scenarios with higher bitrates leading

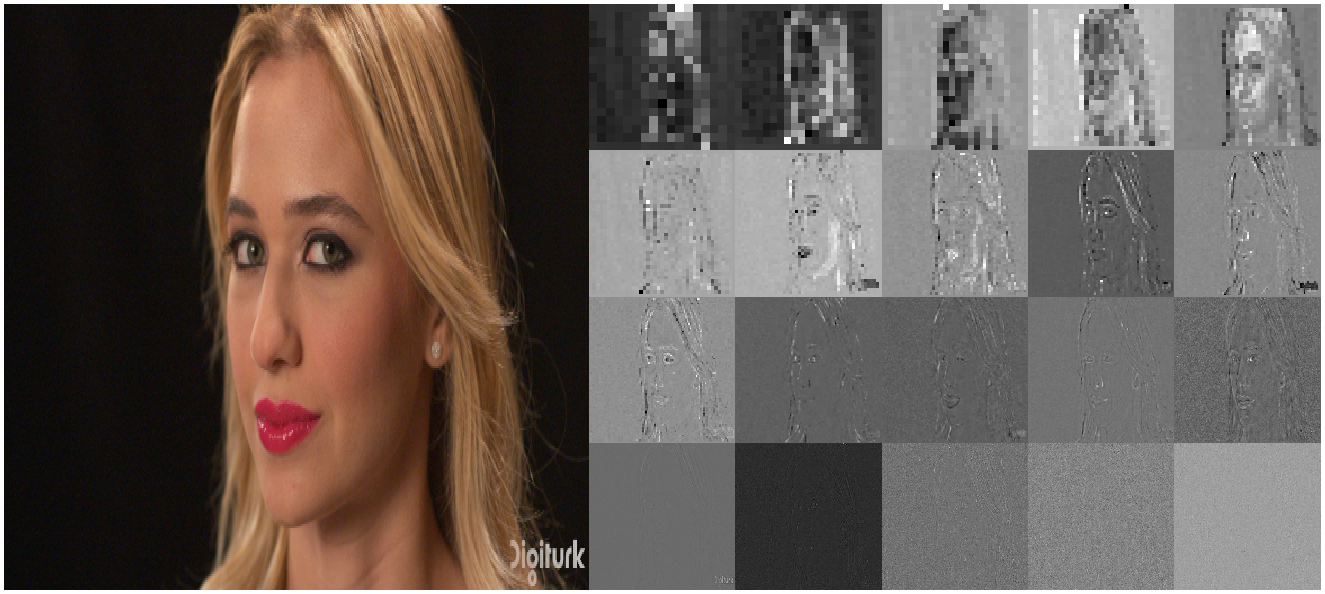
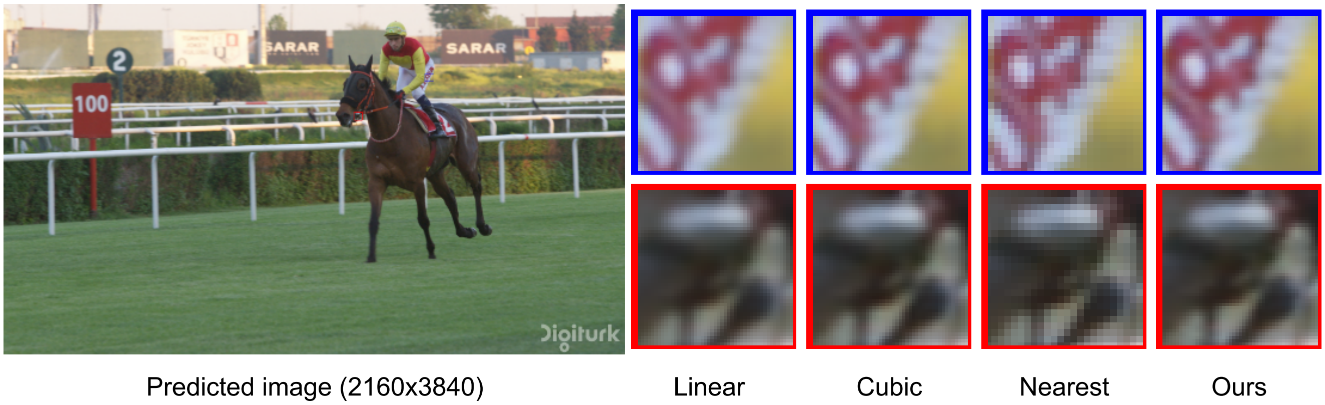


Figure 4: Visualization of learned features (right) for Beauty image (left) from UVG-F. Feature maps at increasing LODs (left-to-right then top-to-bottom) capture finer details in the image highlighting the hierarchical features learned for each signal.



Predicted image (2160x3840) Linear Cubic Nearest Ours

Figure 5: Application of our approach for super-resolution of the Jockey image in the UVG-F dataset. We obtain sharper images compared with standard upsampling methods at 2x upsampling factor (2160 $\times$ 3840).

to higher PSNR (as also discussed in Sec.4.5). Beyond a certain level, we see that the features become less informative globally. This is due to the fact that the grid resolution at finer levels is larger than the number of entries in the latents which is fixed. Multiple locations in the grid map to the same entry in the latent space.

5. Image superresolution

We show the capability of our approach to perform super-resolution of images by providing the input coordinate grids at the desired resolution. Fig.5 shows results on the super-resolution of the Jockey image from UVG-F (1080×1920 resolution) by a factor of 2. We obtain slightly sharper images as compared to standard upsampling methods such as linear, cubic or nearest neighbor interpolation.

6. Additional visualizations

We show additional visualizations of the Pearl (23466×20000) and SMACS images (4630×4537) in Fig.6 (top row and bottom row respectively). We see that we continue to obtain high quality reconstructions and achieve similar PSNR compared to Instant-NGP while being much smaller in storage size (in terms of BPP). Significant artifacts and discoloration are also visible for the traditional JPEG while still requiring more memory than our approach.

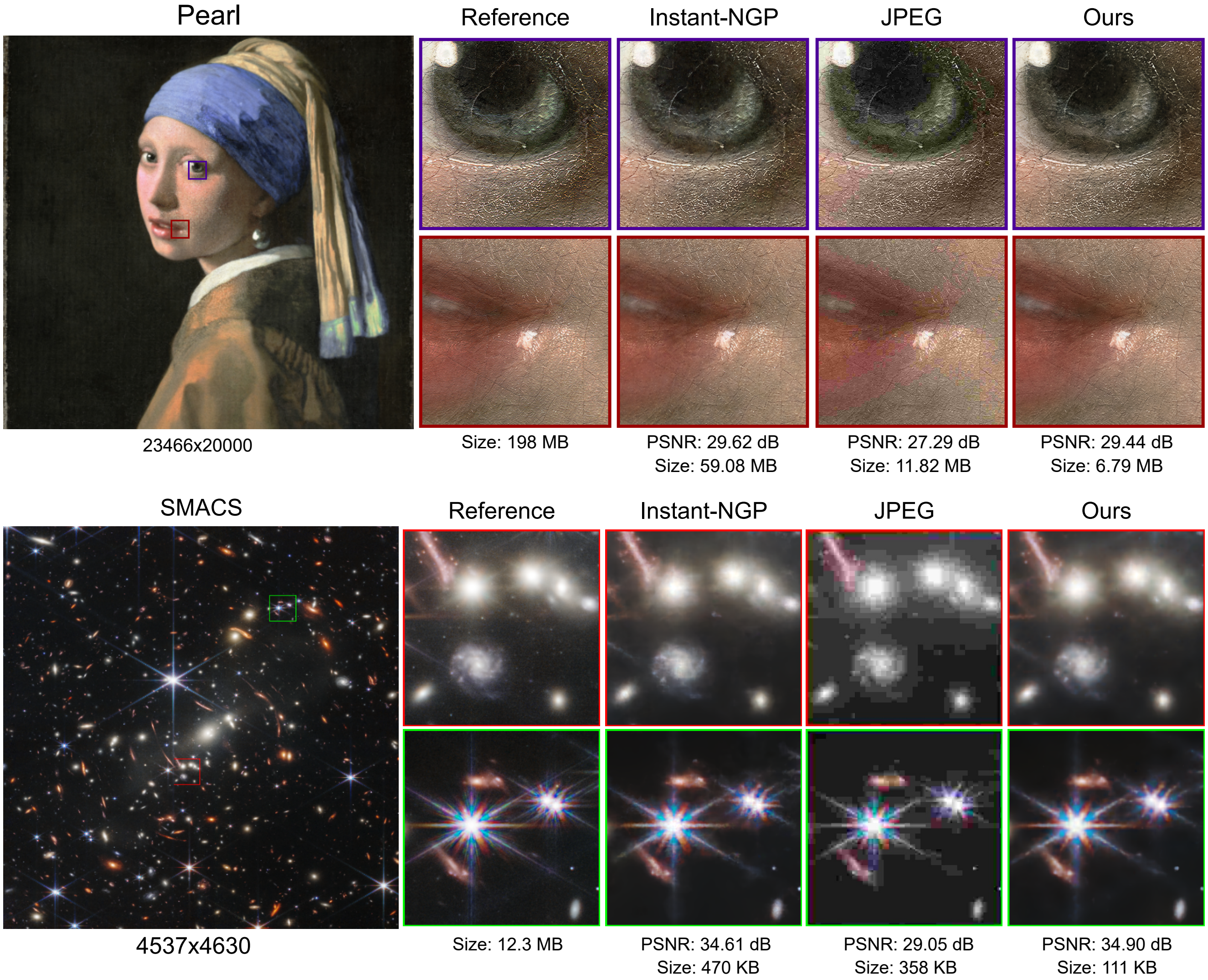


Figure 6: Visualization of Pearl and SMACS images. For Pearl, we maintain a similar PSNR as Instant-NGP [4] while being $\sim 9/\times$ smaller. For SMACS, we marginally outperform them with more than a $4\times$ compression factor. JPEG exhibits visual discoloration artifacts leading to a much lower PSNR even with lower compression factors.

References

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 1

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 1

[3] Sharath Girish, Kamal Gupta, Saurabh Singh, and Abhinav Shrivastava. Lilnetx: Lightweight networks with extreme model compression and structured sparsification. *ArXiv*, abs/2204.02965, 2022. 1

[4] 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. 4