SNeRV: Scalable Neural Representations for Video Coding

Yiying Wei, Hadi Amirpour, Christian Timmerer
Christian Doppler Laboratory ATHENA, University of Klagenfurt, AT
{yiying.wei, hadi.amirpour, christian.timmerer}@aau.at

Abstract

Scalable or layered video coding encodes a video stream into multiple layers in such a way that it can be decoded at different levels of quality or resolution, depending on the capabilities of the device or the available network bandwidth. Traditional approaches are built as an extension of existing video codec standards, but lack industry deployments. In this paper, we propose a Scalable Neural Representation (SNeRV) for video coding that encodes multi-resolution/-quality videos into a single neural network comprising multiple layers. The base layer (BL) of the neural network encodes the lowest resolution/quality of the video stream. Enhancement layers (ELs) encode additional information that, using the BL as a starting point, can be used to reconstruct a higher-resolution/-quality video during the decoding process. This multi-layered structure allows the scalable bitstream to be truncated to adapt to the client's bandwidth conditions or computational decoding requirements. Unlike conventional video codecs constrained by complex and highly designed modules, SNeRV represents a video as a neural network and employs any model weight compression method for video compression. Experimental results demonstrate that SNeRV outperforms H.264/AVC's Scalable Video Coding (SVC) extension and achieves comparable decoding speed at high resolutions.

1 Introduction

In recent years, video and multimedia applications have developed rapidly on platforms, such as smartphones, tablets, laptops, and TVs. Streaming a video to a wide variety of devices needs to match the receiver's context characteristics, such as network bandwidth and device resolution. In some cases, video streaming requires bitrate adaptation by delivering different versions of the same video content to quickly adjust to changes in network conditions or device characteristics. However, traditional video compression approaches such as H.264/AVC [21] or H.265/HEVC [19] generally perform single-layer encoding, producing a single bitstream that hinders real-time adaptation of video streaming.

Layered video coding has been proposed and standardized as extensions of existing video codecs such as Scalable Video Coding (SVC) [16] for H.264/AVC. The concept of this method is to generate multi-layered bistreams through a single encoding process, accommodating the video at different versions. The layered structure of scalable video content comprises a base layer (BL) and one or more enhancement layers (ELs). The BL can be independently decoded using a legacy video decoder (e.g., AVC) and comprises the lowest quality version, and the ELs can be added to the BL to enhance the bitstream's resolution, framerate, or quality. Since this approach is based on traditional video coding standards, the improvement in coding performance is constrained by hand-designed techniques and highly engineered modules of the specific codec.

With the development of deep learning, several learning-based video compression methods [13, 9, 10] have been proposed to replace hand-crafted modules with neural networks, presenting the potential

Workshop on Machine Learning and Compression, NeurIPS 2024.

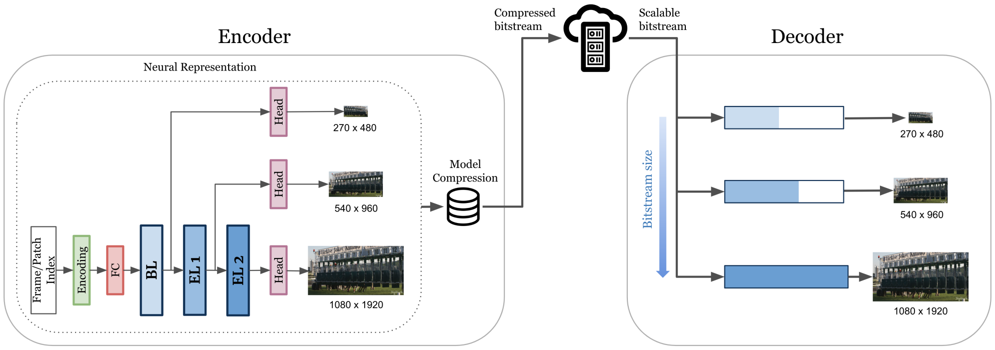


Figure 1: The overview of SNeRV. The encoder performs neural network training on multi-resolution videos to fit an implicit neural network, which includes a BL for reconstructing a lowest resolution video and subsequent ELs for higher resolution videos. The fitted model is then compressed into a scalable bitstream, allowing for adaptive streaming based on the bandwidth and the computational resources of the decoder.

to compete with traditional, standard video codecs. In this context, Implicit Neural Representations (INRs) have previously been investigated to represent and encode images [17, 18] and videos [4, 3], showing relatively faster decoding speed compared to other learning-based methods. Neural Representation for Videos (NeRV) methods [4, 12, 2, 3, 8, 7] represent a video as a neural network by encoding the video in the weights of the network. In NeRV methods, video coding involves training a neural network to represent the video content, with the trained model acting as the video bitstream. Decoding is done by simply passing data through the network in a feedforward operation.

In this paper, we propose Scalable Neural Representations (SNeRV) for video coding. Based on the single-stream NeRV structure in previous work [7], we explore a multi-stream NeRV network, which consists of executable sub-networks for constructing videos at different representations. Figure 1 shows an example of SNeRV that encoding a video at three resolutions (270p, 540p and 1080p) through one network. Specifically, a BL block corresponds to the lowest resolution of 270p, while the subsequent EL1 and EL2 blocks represent higher resolutions of 540p and 1080p, respectively. Therefore, these three resolutions can be represented by a scalable bitstream with three layers, and the video at each resolution can be decoded independently.

The remainder of this paper is structured as follows. Section 2 details the proposed SNeRV method. Experimental results are presented in Section 3 and Section 4 concludes the paper.

2 Scalable Neural Representations for Video Coding

Figure 1 presents the overview of SNeRV, which compresses a video into a multi-layered bitstream through an end-to-end training process. The multi-layer bitstream can represent the video in I resolutions, with the scale factor and output resolution adjustable according to the specified network architecture. In this paper, we encode the video at three different resolutions (270p, 540p and 1080p) as an example, i.e., $I = 3$. As shown in Figure 1, the encoder trains a SNeRV model to represent a video at three resolutions. This model consists of a BL and two ELs. The BL corresponds to the lowest supported video resolution, whereas the ELs allow for the higher resolutions by the refinement of the aforementioned BL. After a single training process, the video is represented as a multi-layered model, which is then compressed into a scalable bitstream and transmitted to the decoder. Therefore, receivers can select and decode bitstreams of different sizes corresponding to different video resolutions, depending on the limitations of the network and the device itself.

2.1 Multi-stream Network Structure

The SNeRV is based on the image-wise NeRV network from the literature [4, 7], which takes frame, patch, or pixel coordinates as inputs and leverages neural representations for reconstructing video frames. These NeRV models usually represent a video as a single stream. In this paper, we transform the single-stream video model into a multi-stream model, composed of distinct yet complementary

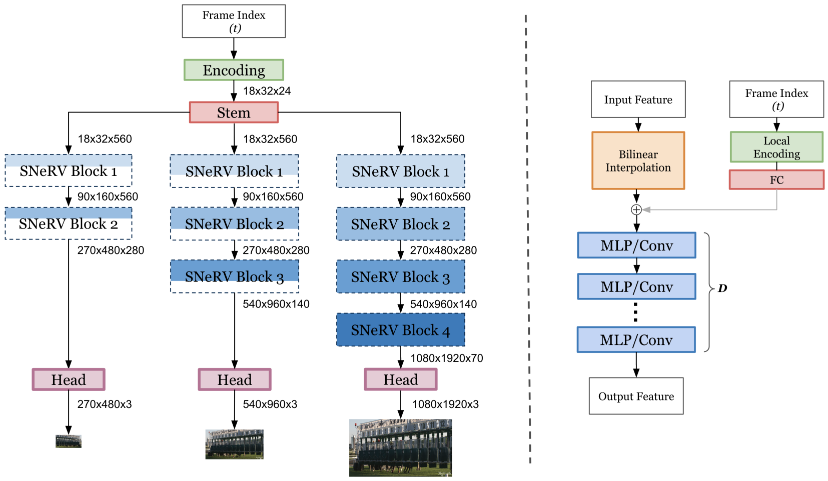


Figure 2: **Left**: The network structure of SNeRV model. Certain sub-layers of the SNeRV Blocks are shared among the three branches of sub-networks. The sub-network used for generating higher-resolution frames integrates EL along with the BL that generate lower-resolution frames. The size of the feature maps is denoted as $H_x \times W_x \times C_x$. **Right**: The SNeRV Block. The frame index t is incorporated to enhance the network's capacity to learn high-frequency information; however, it is not depicted in the overall network structure for simplicity.

network layers. Given that a video $V^{T \times H \times W \times C}$, where T, H, W and C represent the number of frames, height, width and the number of channels in the video frames, respectively. The multi-stream model can be trained on either the entire frame or divided patches of the video. Here we take a frame $v_t \in V$ for better explication ($0 < t \leq T$).

Figure 2 illustrates the multi-stream network structure, which consists of a base encoding layer, a stem layer, M SNeRV blocks and N head layers. In this paper, we utilize three video resolutions and, therefore, our method utilizes four SNeRV blocks and three heads, i.e., $M = 4$, $N = 3$. Similarly to other NeRV blocks [4, 12, 7], the SNeRV block is used to upscale the spatial dimensions of feature maps and enhance their expressiveness through multiple multilayer perceptrons (MLPs) or convolutional layers. In this work, we adopt the HiNeRV block [7] as the backbone of the SNeRV block due to its reasonable performance compared to traditional video codecs. For each SNeRV block, the feature maps are first upscaled using simple linear interpolation, enriched with embedding information, and then transformed by a set of MLPs or convolutional layers.

In the multi-stream network, the input frame index (t) is first encoded and interpolated into base feature maps with size $H_0 \times W_0 \times C_0$. The following four blocks are named SNeRV blocks 1 to 4, arranged from low to high levels. These SNeRV blocks progressively upsample and process the feature maps. Based on the desired output resolutions, we set the upscale factors to $\{5, 3, 2, 2\}$ for SNeRV blocks 1 to 4, respectively. For generating the three video representations, we use three branches of sub-networks that share certain sub-layers of the SNeRV Blocks. Since the network needs to gradually reduce the number of channels to achieve higher spatial resolution frames, the SNeRV Block at lower level would have a larger number of parameters. In order to adjust the distribution of different sub-bitstream sizes, each sub-network can select the number of sub-layers D_x according to the required size ($0 < D_x \leq D$). In practice, the sub-network responsible for generating higher-resolution frames adds ELs to the BL that generates the lower-resolution frames.

2.2 Training Strategy

The training for the SNeRV model can be considered as multi-task training. Each task represents a video in a specific presentation. During the end-to-end training, each task minimizes its individual loss to optimize the corresponding sub-network. The total loss is the weighted sum of losses for each sub-network as follows

$$\sum_{i=1}^{I} \lambda_i L_i \tag{1}$$

where I is the total number of sub-networks, L_i represents the loss of the i^{th} sub-network, and λ_i denotes the weight for the loss of each sub-network.

The loss for each sub-network is the combination of $\ell 1$ and Structural Similarity Index Measure (SSIM) [20] loss, which is computed over all pixels of the reconstructed frame and the ground-truth frame at a specific resolution, and can be expressed as

$$L_i = \alpha \left\| f_\theta^i(t) - v_t \right\|_1 + (1-\alpha)(1 - \text{SSIM}(f_\theta^i(t), v_t)) \tag{2}$$

where f_θ^i is the i^{th} sub-network parameterized by θ, $v_t \in \mathbb{R}^{H \times W \times 3}$ is the ground-truth t^{th} frame of the video, and α is hyper-parameter to balance the weight for each loss component.

2.3 Model Compression

After fitting the SNeRV network to a video, we apply the model compression pipeline in NeRV [4, 12, 7], to further reduce the model size, which in turn decreases the bitstream size. The model compression includes three sequential steps: model pruning, weight quantization, and weight encoding.

Model Pruning. Given a SNeRV network that fits on a video content, we use model pruning with fine-tuning to reduce the model size first. We adopt adaptive weighting [7] to the parameters of each layer based on its size for pruning, making wider layers with more redundant parameters more likely to be pruned. After pruning, we fine-tune the model to regain the representations.

Weight Quantization. After model pruning, we use 6-bit quantization to compress the bitstream while maintaining the quality of the representations as much as possible. With fine-tuning training using Quantization Noise [5] at 80% noise ratio, the quantization error can be effectively reduced.

Weight Encoding. Finally, we perform the weight encoding to further enhance the compression performance. The weights will be compressed using arithmetic coding [14], a type of entropy coding that leverages the character frequency. To obtain the actual bitrate, we additionally encode all essential compression information into bitstreams, including the weight pruning and quantization parameters.

3 Experimental Results

3.1 Datasets and Implementation Details

We conduct experiments using the UVG [15] dataset, which consists of seven videos at 1920×1080 resolution and a total number of 3900 frames. We downscale the original videos using *ffmpeg* with bicubic scaling to obtain low-resolution versions of 960×540 and 480×270, respectively. In order to compare with H.264/AVC [21], we reduce the frame rate of raw YUV video from 120 frames per second (fps) to 60 fps by dropping every other frame, while maintaining the original raw format.

In our experiments, we train all models for each video separately. All models are trained for 300 epochs, followed by fine-tuning for 60 epochs after weight pruning. Finally, they are further optimized for 30 epochs using fine-tuning with Quantization Noise [5]. We train the SNeRV network using Adam optimizer [6] with a learning rate of $5e-4$. For the loss of each sub-network in Equation 2, the parameter α is set to 0.5. The weight λ_i in Equation 1 is uniformly applied to the loss of each sub-network. Our experiments indicate that increasing the weight assigned to high-resolution tasks generally leads to sub-optimal results in most sub-networks. One possible explanation is that the sub-network for the high-resolution task rely on shared blocks at lower levels, which also have a significant impact on the final video quality.

For the network structure, we adjust the layer number D_x of each SNeRV block for the sub-networks to determine the size of each sub-bitstream for different resolutions. Specifically, for the 270p sub-network, the D_x for SNeRV blocks 1 and 2 are set to 4. In the case of the 540p sub-network, the D_x of SNeRV block 1, 2, and 3 are each set to 6. For the 1080p sub-network, the D_x for all SNeRV blocks are uniformly set to 8.

We vary the number of channels of the SNeRV network to achieve different bitstream rates. For conventional SVC [16], we perform experiments with multiple QP values {32,37,42,47} to obtain the results at different rates. Specifically, we use the latest SVC reference software, JSVM 9.19.15 [1]. Bitrate is calculated by dividing the bitstream size by the duration of the corresponding video, and is expressed in kilobits per second (kbps). We evaluate the video quality using two metrics: PSNR and

Table 1: Model scalability of SNeRV. The results are reported in terms of BD-PSNR/VMAF for the upscaling factor 2× (from 270p to 540p) and 2× (from 270p to 1080p).

Upscale factor	BD-Metric	Beauty	Bosph.	Honey.	Jockey	Ready.	Shake.	Yacht.	Average
2×	BD-PSNR (dB)	+0.43	+2.02	+1.60	+1.97	+0.79	+1.85	+1.04	+1.38
	BD-VMAF	+5.65	+16.04	+15.27	+15.93	+7.02	+11.72	+11.70	+11.90
4×	BD-PSNR (dB)	+1.66	+3.03	+1.47	+4.80	+1.73	+2.90	+1.05	+2.38
	BD-VMAF	+2.92	+13.78	+12.31	+16.29	+3.44	+0.19	+9.80	+9.68

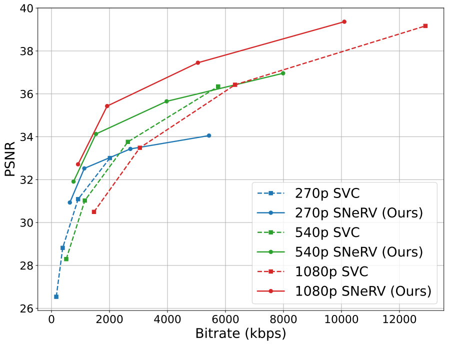


Figure 3: SNeRV *vs.* SVC based on PSNR.

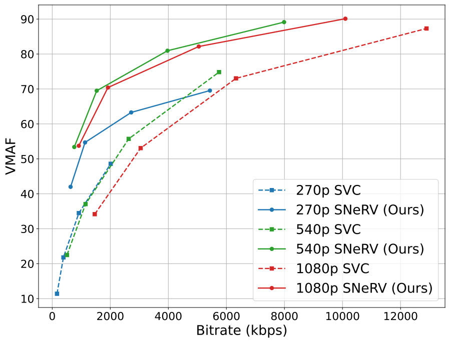


Figure 4: SNeRV *vs.* SVC based on VMAF.

VMAF [11]. To ensure a fair and straight comparison of video quality across different resolutions, we use *ffmpeg* to bicubic upscale the generated 270p and 540p videos to 1080p. In the paper, we compare these upscaled videos with the original 1080p raw video to obtain the video quality metrics.

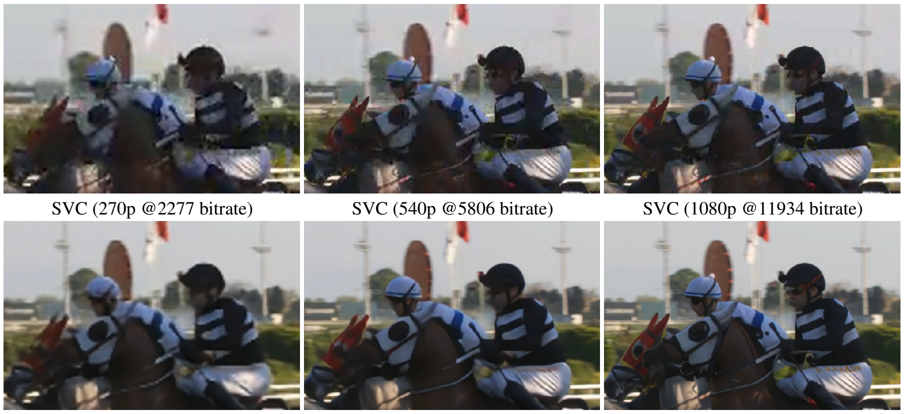
3.2 Model Scalability of SNeRV

SNeRV comprises a sub-network of BL to generate the supported lowest resolution video, and then uses the sub-networks with ELs to provide higher resolution representations. Table 1 shows the scalability of SNeRV using BD-PSNR/VAMF, specifically for the upscaling from 270p to 540p and from 270p to 1080p. Note that for the upscaling performance of 2× and 4×, we only consider the size of the ELs, rather than the entire sub-network which includes both the BL and EL. Compared to videos generated from the BL at the same bitrates, videos upscaled to 2× resolution provide +1.38/+11.90 better quality in PSNR/VMAF, while 4× upscaled videos achieve +2.38/+9.68 better quality in PSNR/VMAF. It shows that the SNeRV model can generate videos at different resolution by sharing BL and adopt certain ELs within the network. By leveraging the scalability of the SNeRV model, we can transform the traditional video compression problem into a model compression problem, further improving the compression performance of scalable video coding.

3.3 Compare with Conventional Scalable Video Coding (SVC)

Video Compression Performance. Figure 3 and Figure 4 show the overall encoding performance on the UVG dataset. Video resolutions of 270p, 540p and 1080p are represented by blue, green, and red lines, respectively. The curves of our proposed SNeRV are shown with solid lines, while the curves of SVC are represented with dashed lines. The results show that SNeRV achieves better scalable coding performance compared to SVC. In particular, SNeRV outperforms SVC based on VMAF metric, which accounts for perceived visual quality. It is also worth noting that at lower bitartes, SNeRV provides a greater quality gap compared to SVC at the same resolution, demonstrating its potential in video compression. Figure 5 shows visual comparison between SVC and SNeRV. At each resolution, SNeRV reconstructs videos with better quality using similar bitrate.

Decoding Speed. For a multi-layered bitstream, the video at each resolution can be decoded independently. We compare the decoding speed between SVC and SNeRV in terms of frame per second (fps), as shown in Table 2. Note that SVC runs on CPU, while the learning-based method SNeRV runs on GPU (a single RTX A6000). Although the decoding efficiency of SNeRV highly depends on the available GPUs, the results demonstrate that SNeRV has an advantage in reconstructing high-resolution videos. The speed can be further optimized by loading and running



SVC (270p @2277 bitrate) SVC (540p @5806 bitrate) SVC (1080p @11934 bitrate)

SNeRV (270p @2275 bitrate) SNeRV (540p @5440 bitrate) SNeRV (1080p @10106 bitrate)

Figure 5: Visual comparison between the reconstructed frames (cropped) of SVC and SNeRV (Ours).

Table 2: Decoding speed of SVC and SNeRV at the same bitstream size.

Resolution	Bitstream size	SVC	SNeRV
270p	0.32MB	**193.18** fps	69.47 fps
540p	0.82MB	26.58 fps	**27.73** fps
1080p	2.63MB	4.65 fps	**8.29** fps

the model on specialized hardware. Furthermore, SNeRV is based on an implicit neural network, allowing potentially for parallel decoding. In traditional scalable video codecs, the decoding process is sequential, starting with intra-frames and then decoding inter-frames based on those intra-frames. In contrast, our method allows direct access to any frame using its time index, enabling independent frame decoding and parallel processing.

4 Discussion

Conclusion. In this work, we propose SNeRV, a scalable neural representation for video coding that encodes videos using a multi-stream neural network. Without the need for multiple training processes to encode a single video, we train a neural network to simultaneously represent various forms of video content, such as different resolutions. Each video representation can be independently decoded by its respective sub-stream. Since the video is encoded into model weights, the SNeRV model can take advantage of general model compression techniques to compress videos efficiently. We demonstrate that SNeRV exhibits scalability for increasing video resolution and refining video representation by adding ELs to the BL of networks. Furthermore, we show that the proposed SNeRV method outperforms the traditional scalable video coding method (SVC) and achieves comparable decoding speed at high resolutions.

Limitations and Future Work. The proposed SNeRV method does have some limitations. First, while it allows for multiple video representations within a single network, its training performance is lower than that of a single-stream network with the same training settings. Similarly, in traditional video codecs, scalable coding typically underperforms compared to non-scalable coding under equivalent conditions. Second, the current architecture of SNeRV is not yet fully optimized. Although the scalability of the INR-based network has been demonstrated in the paper, we believe that further investigations in architectural design could greatly enhance the scalable coding performance. Finally, like all existing INR-based video compression methods, SNeRV suffers from slow encoding speeds, and the decoding speed still needs to be improved compared to conventional codecs.

Acknowledgment

The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association, is gratefully acknowledged. Christian Doppler Laboratory ATHENA: `https://athena.itec.aau.at/`.

References

[1] H.264/AVC scalable video coding (SVC) extension JSVM reference software. `https://vcgit.hhi.fraunhofer.de/jvet/jsvm`.

[2] Y. Bai, C. Dong, C. Wang, and C. Yuan. PS-NeRV: Patch-wise stylized neural representations for videos. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 41–45. IEEE, 2023.

[3] H. Chen, M. Gwilliam, S.-N. Lim, and A. Shrivastava. HNeRV: A hybrid neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10270–10279, 2023.

[4] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim, and A. Shrivastava. NeRV: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.

[5] A. Fan, P. Stock, B. Graham, E. Grave, R. Gribonval, H. Jegou, and A. Joulin. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*, 2020.

[6] D. P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] H. M. Kwan, G. Gao, F. Zhang, A. Gower, and D. Bull. HiNeRV: Video compression with hierarchical encoding-based neural representation. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] J. C. Lee, D. Rho, J. H. Ko, and E. Park. FFNeRV: Flow-guided frame-wise neural representations for videos. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7859–7870, 2023.

[9] J. Li, B. Li, and Y. Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021.

[10] J. Li, B. Li, and Y. Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023.

[11] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock. VMAF: The journey continues. *Netflix Technology Blog*, 25(1), 2018.

[12] Z. Li, M. Wang, H. Pi, K. Xu, J. Mei, and Y. Liu. E-NeRV: Expedite neural video representation with disentangled spatial-temporal context. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022.

[13] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11006–11015, 2019.

[14] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool. Practical full resolution learned lossless image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10629–10638, 2019.

[15] A. Mercat, M. Viitanen, and J. Vanne. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020.

[16] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H. 264/AVC standard. *IEEE Transactions on circuits and systems for video technology*, 17(9):1103–1120, 2007.

[17] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[18] Y. Strümpler, J. Postels, R. Yang, L. V. Gool, and F. Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022.

[19] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[20] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[21] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.