Scalable COOL-CHIC: Dual-Resolution Images from a Single Bitstream

Martin Benjak* Yi-Hsin Chen† Wen-Hsiao Peng† Jörn Ostermann*
* Institute for Information Processing, Leibniz Universität Hannover, Germany
†Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
benjak@tnt.uni-hannover.de

***Abstract*—In this work, we propose Scalable Cool-Chic (SCC), a scalable overfitted neural image codec that encodes the same image in two different resolutions with a fixed scaling factor of 2 into one bitstream. Experiments show that our method reduces the necessary bitrate to encode two representations of the same image by up to 30.10% in terms of BD-rate compared to encoding both representations independently using Cool-Chic while increasing the decoding time of high-resolution images by only 6.59%. The bitstream is structured in a way that the low-resolution image can already be decoded, when only a part of the bitstream has been received. The code is available at `https://github.com/mbenjak/Scalable-CC`.**

***Index Terms*—Image compression, low-complexity, neural coding, overfitting, scalable coding**

I. Introduction

Most lossy image and video codecs since H.261 and JPEG [1] are based on the transform coding scheme, in which signals are encoded by quantization and entropy coding in the transform domain rather than the pixel domain. The aforementioned codecs, as well as more modern ones such as JPEG XL [2], HEVC [3], and VVC [4], use linear transformations like the DCT or DST. However, JPEG 2000 [5] takes a slightly different approach by using a wavelet transform instead.

Neural image codecs [6]–[9] still follow the transform coding scheme, but they replace a few modules such as the transform and entropy model with neural networks. These neural networks are trained using gradient descent and a rate-distortion loss function on large and varied image datasets with the goal of having a single codec that performs well on many different images. Modern neural codecs such as [10] outperform VVC in image and video compression. However, neural codecs require up to one million multiplications per decoded pixel, making them several orders of magnitude more complex than conventional codecs.

Dupont et al. [11] introduced COIN, which does not follow the transform coding scheme. Instead, a lightweight Multi-Layer Perceptron (MLP) maps the coordinates of a pixel to its RGB values. To encode an image, the MLP is overfitted to the pixels of a single image and then quantized and transmitted. Thus, the image information is encoded in the MLP itself. This approach has the benefit of requiring orders of magnitude fewer multiplications per pixel than previous neural image codecs. However, the compression performance of this scheme is not on par with that of more computationally complex neural image codecs.

This work was supported by the German Federal Ministry of the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (GreenAutoML4FAS project no. 67KI32007A) and Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project no. 508272532.

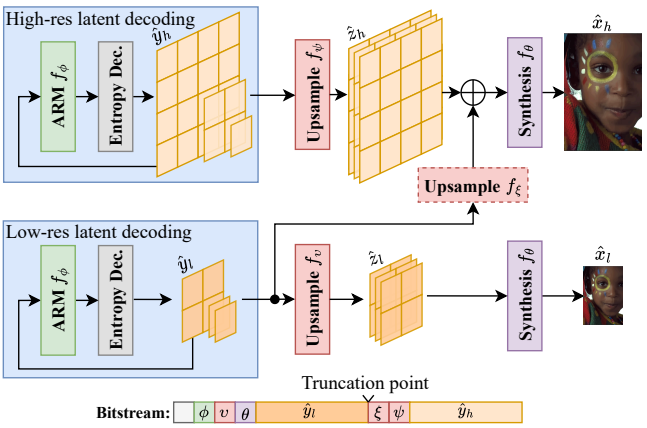

Fig. 1. Decoding a bitstream into a high-resolution or low-resolution image. For low-resolution images $\hat{x}_l$, only the bottom half of the diagram is executed. For high-resolution images $\hat{x}_h$, both the low- and high-resolution latents $\hat{y}_l$, $\hat{y}_h$ are decoded and summed up after upsampling with f_ξ and f_ψ, respectively. Finally, $\hat{x}_h$ is reconstructed using the synthesis f_θ. All weights are signaled in the bitstream and f_ϕ and f_θ share the same weights for low- and high-resolution decoding. The bitstream can be split at the truncation point, if only the low-resolution image $\hat{x}_l$ is needed.

COIN's relatively low performance is due to the non-local nature of the MLP, which means that each MLP parameter affects every pixel of the image and vice versa. To address this issue, Ladune et al. introduced Cool-Chic (CC) [12], which extends COIN with a hierarchical latent representation which contains most of the information. The latent is entropy encoded using an Auto-Regressive Model (ARM). After entropy decoding, the latent is upsampled such that each upsampled hierarchical layer of it has the same spatial resolution and is fed into a neural synthesis network which outputs the RGB values for each pixel. The entire codec is overfitted to a single image and the ARM and synthesis network, along with the entropy-encoded latent, are quantized, entropy-encoded and signaled together. Cool-Chic has been iteratively extended: [13] added adaptive upsampling and replaced the synthesis MLP with a lightweight CNN; [14] added several optimization improvements as well as small architectural changes that led to substantial gains; [15] added a CPU-only decoder, as well

as encoder speed-ups; and [16] made the adaptive upsampling filter separable and symmetric, which reduces the complexity and leaves headroom for more complex filters.

Scalable coding describes methods that provide multiple interdependent representations of the same image or video, varying for example in spatial or fidelity dimensions.

Scalable image and video codecs [4], [17]–[20] encode the lowest quality representation using a standard encoder such as VVC or HEVC. LCEVC [17] encodes higher-resolution representations differentially by subtracting the upsampled lower-resolution representation from the input image, followed by transform coding the resulting residual. Neural scalable codecs, such as [19], work similarly, but instead of differential coding, conditional coding is used. SHVC [20] and the multi-layer mode of VVC [4] exploit dependencies between different layers by processing reconstructed pictures from one layer as a reference picture for predictive coding in another layer.

Both JPEG and JPEG XL support scalable coding, either by grouping DCT coefficients into spectral bands or by successively encoding higher-precision AC coefficients [1], [2]. JPEG 2000 uses a discrete wavelet transform to map the image to a hierarchical subband representation. The bitstream is structured so that each subband can be reconstructed consecutively, providing an inherent progressive decoding mode.

Scalable coding has many applications including generating thumbnails for photo galleries on phones or file browsers, decoding satellite images at multiple resolutions for navigation systems, and broadcasting of videos at multiple resolutions. Compared to independently encoding different representations, scalable coding offers lower total bitrates.

In this work, we extend Cool-Chic with a novel scalable mode that encodes two different spatial resolutions of the same image into the same bitstream at a lower bitrate compared to independently encoding both resolutions. We refer to this extended codec as Scalable Cool-Chic (SCC). The decoder can decode the bitstream either into the full-resolution image or into a low-resolution image. The scale factor between both representations is fixed to 2. The benefit of decoding the low-resolution image is a reduced decoding time and decoder memory footprint, since the decoding time is proportional to the number of decoded pixels. The bitstream is arranged in such a way, that the low-resolution image can already be decoded when only the first part of the bitstream has been received.

II. Scalable Decoding Method

Our codec can encode an image such that a low-resolution representation as well as a high-resolution representation can be decoded from the same bitstream. The resolution of the high-resolution representation is equal to the original resolution of the encoded image, while the resolution of the low-resolution representation is reduced by a scaling factor of 2. The decoder can decide which representations to decode. The decoding pipeline for low-resolution and high-resolution images is shown in Fig. 1.

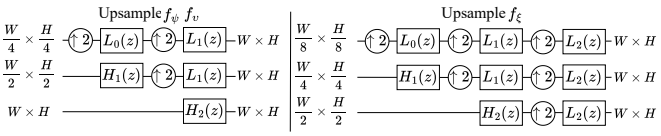

Fig. 2. Architecture of the upsamplers for $L = 3$ layer latent pyramids based on [16]. It consists of the learnable upsampling filters $L_{0..2}(z)$ and the additional learnable filters $H_{1..2}(z)$. Both $L_{0..2}(z)$ and $H_{1..2}(z)$ are separable and symmetric.

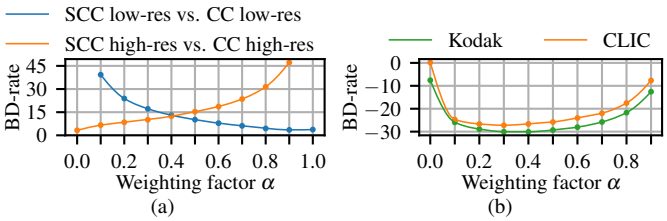

Fig. 3. Impact of α on low- and high-resolution decoding on Kodak (a). BD-rates between *PCC high-res* and *CC high-res + low-res* over α (b).

The information for the low-resolution and high-resolution images is mainly encoded in the L layer latent pyramids $\hat{y}_l = \{\hat{y}_{l,i} \in \mathbb{Z}^{H/2^{(i+1)} \times W/2^{(i+1)}}, i = 0, \cdots, L-1\}$ and $\hat{y}_h = \{\hat{y}_{h,i} \in \mathbb{Z}^{H/2^i \times W/2^i}, i = 0, \cdots, L-1\}$. The neural network parameters ϕ, υ, θ, ξ, and ψ are overfitted to the single image that they are able to decode and thus contain some information about the image as well.

The decoding process for low-resolution images is identical to [16]: First, $\hat{y}_l$ is entropy decoded using the ARM f_ϕ which iterates over all elements of each layer in $\hat{y}_l$ and considers the neighbors of the currently decoded element to model the probability distribution of the current element. The latent pyramid $\hat{y}_l$ is then upsampled into a tensor of shape $L \times H/2 \times W/2$ using f_υ followed by the synthesis f_θ resulting in the reconstructed image

$$\hat{x}_l = f_\theta(f_\upsilon(\hat{y}_l)). \tag{1}$$

To reconstruct the high-resolution image, both latent pyramids $\hat{y}_l$ and $\hat{y}_h$ are entropy decoded using the ARM f_ϕ. The latent pyramids $\hat{y}_l$ and $\hat{y}_h$ are then upsampled using f_ξ and f_ψ and summed up to a $L \times H \times W$ tensor followed by the synthesis f_θ resulting in the reconstructed image

$$\hat{x}_h = f_\theta(f_\xi(\hat{y}_l) + f_\psi(\hat{y}_h)). \tag{2}$$

Note that both the ARM f_ϕ and synthesis f_θ use the same parameters for high- and low-resolution images. The upsamplers f_υ and f_ψ share the same architecture, however they use different parameters. The upsampler f_ψ has one additional upsampling step and also different parameters than the other upsamplers (see Fig. 2).

The bitstream consists of two parts. The first part is sufficient to decode the low-resolution image and consists of picture headers, the neural network parameters ϕ, υ and θ followed by the low-resolution latent pyramid $\hat{y}_l$. The second part of the bitstream can be used together with the first

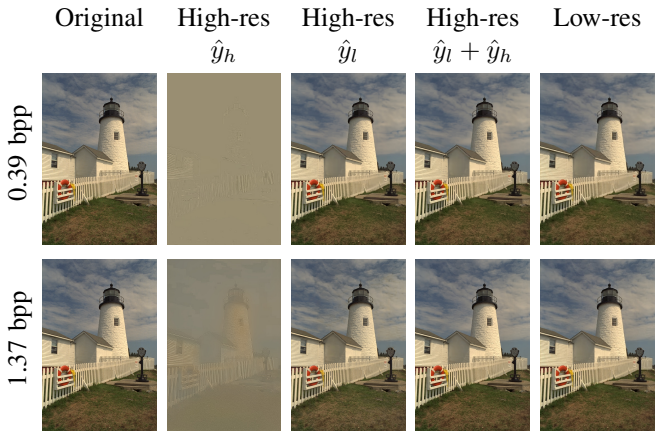

Fig. 4. Example images encoded at two different bitrates with our SCC codec for $\alpha = 0.4$ tested on *kodim19* from [21]. High-resolution images are shown reconstructed only from the high-resolution latent $\hat{y}_h$, low-resolution latent $\hat{y}_l$, and from both latents. Full-resolution images, zoom in to see the details.

part to decode the high-resolution image and consists of the upsampler parameters ξ and ψ followed by the high-resolution latent pyramid $\hat{h}_h$. All weights and biases for the neural networks used in the decoding process are signaled.

III. Encoding Method

An image x_h is encoded by learning both the latents $\hat{y}_l$, $\hat{y}_h$ and the neural network parameters $\phi, \upsilon, \theta, \xi, \psi$ by minimizing the loss function $\mathcal{L}$:

$$\hat{y}_l^*, \hat{y}_h^*, \phi^*, \upsilon^*, \theta^*, \xi^*, \psi^* = \underset{\hat{y}_l, \hat{y}_h, \phi, \upsilon, \theta, \xi, \psi}{\operatorname{argmin}} \mathcal{L}. \quad (3)$$

Note that the minimization is done for only one image x_h, thus overfitting the neural networks and latents to this image. The loss function

$$\mathcal{L} = (1-\alpha)\mathrm{D}(x_h, \hat{x}_h) + \alpha \mathrm{D}(x_l, \hat{x}_l) + \lambda \mathrm{R}(\hat{y}_l) + \lambda \mathrm{R}(\hat{y}_h) \quad (4)$$

describes a typical rate-distortion loss function where the Lagrange multiplier λ balances the rate R and the MSE distortion D. The weighting factor α controls the trade-off between low and high-resolution image quality. The low-resolution target image x_l is obtained by downsampling x_h by a factor of 2 using a Lanczos filter.

The latents are quantized by rounding to the nearest integer. However, the quantization operation is not differentiable, thus we use the method proposed in [14] to obtain a differentiable proxy for the quantization.

Once the latents and parameters are overfitted, the neural network parameters are quantized and entropy coded. The quantized latents $\hat{y}_l$, $\hat{y}_h$ are entropy coded adaptively using the ARM f_ϕ.

IV. Evaluation

In this chapter, the evaluation methodology is described and the results are discussed.

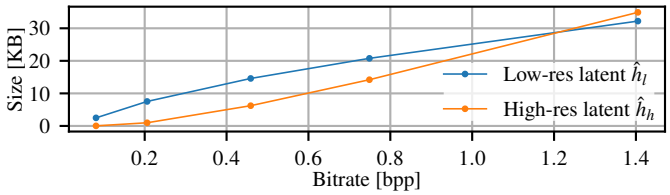

Fig. 5. Bitrate allocation between high and low-resolution latents for different total bitrates for our SCC codec at $\alpha = 0.4$ on Kodak [21].

A. Experiment Setting

Our Scalable Cool-Chic (SCC) produces a bitstream that can be decoded at high- and low-resolutions. Cool-Chic (CC) [16] encodes only one resolution, thus we compare SCC with three different variations of CC: *CC high-res*, *CC low-res* and *CC high-res + low-res*. *CC high-res* is the regular version of CC that directly encodes an image. *CC low-res* downsamples the image by a factor of two using a Lanczos filter and then encodes the downsampled image. For *CC high-res + low-res*, we encode an image using CC in both high- and low-resolutions, thus requiring the sum of both bitrates. The two decoder output resolutions of SCC are referred to as *SCC high-res* and *SCC low-res*.

CC high-res and SCC are encoded with $\lambda \in 0.0001, 0.0004, 0.001, 0.004, 0.02$ using the slowest preset and the highest decoder operating point at 1430 MAC per pixel. *CC low-res* is encoded with 77 different λ values ranging from 2×10^{-5} to 0.03. *CC high-res + low-res* is a virtual setting that uses the PSNR values from *CC high-res* and the summed up bitrates from *CC high-res* and *CC low-res*. To enable a fair comparison with SCC, *CC low-res* is PSNR-matched to *SCC low-res*.

SCC high-res is a configuration that uses the PSNR of the high-resolution image $\hat{x}_h$ together with the full bitrate. *SCC low-res* uses the PSNR of the low-resolution image $\hat{x}_l$ together with the low-resolution part of the bitrate which is measured from the beginning of the bitstream until the truncation point (see Fig. 1).

All PSNR values are computed by comparison to original full-resolution images. Low-resolution images are Lanczos upsampled prior to the PSNR calculation. All bitrates for high- and low-resolution images are measured in bit per pixel (bpp) based on the full image resolution. By this, all PSNR and bpp values are comparable to each other.

We have extended the existing C++ implementation of the Cool-Chic decoder to support SCC bitstreams. PSNR values and decoding times are measured using the C++ implementation with the decoding time being averaged over ten runs on an Intel i9-9900K CPU @ 3.60GHz. The datasets used for our experiments are Kodak [21] and CLIC 2020 evaluation [22].

B. Bitrate Allocation

The focus of the encoder and by that the bitrate allocation between high- and low-resolution can be controlled using $\alpha \in [0, 1]$. Fig. 3(a) shows that increasing α leads to decreasing BD-rates for *SCC low-res* and increasing BD-rates for *SCC high-res*. As Fig. 3(b) shows, the smallest BD-rate between

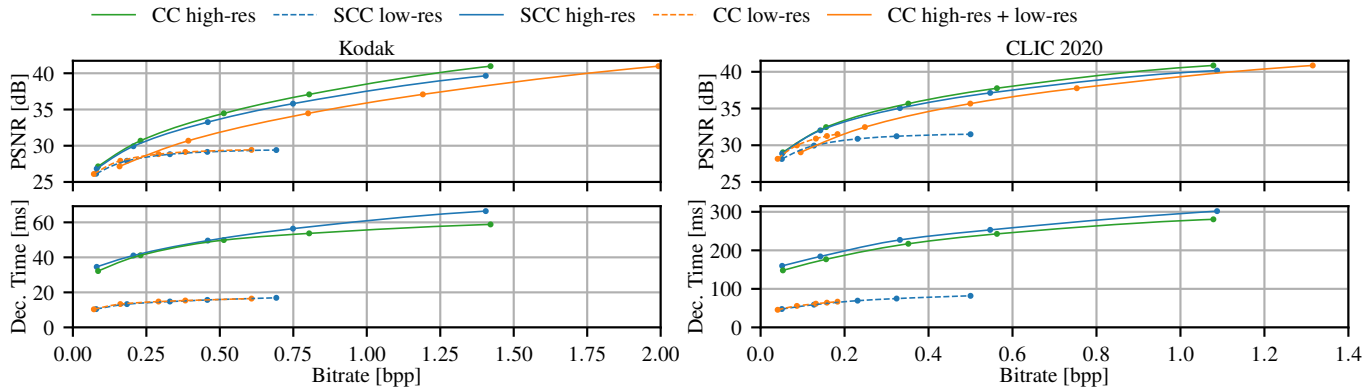

Fig. 6. PSNR and decoding time for our method Scalable Cool-Chic (SCC) and Cool-Chic (CC) in high and low-resolution. *CC high-res + low-res* uses the PSNR from *CC high-res* and the bitrate from both *CC low-res* and *CC high-res*.

SCC high-res and *CC high-res + low-res* is achieved at $\alpha = 0.4$ for Kodak and $\alpha = 0.3$ for CLIC 2020.

Fig 4 shows images encoded with SCC at two different bitrates. It is visible that $\hat{y}_h$ stores high-frequency information like edges while $\hat{y}_l$ contains low-frequency information. With increasing bitrate, more information is stored in $\hat{y}_h$. The same behavior can be observed in Fig. 5.

C. Rate-Distortion Performance and Decoding Time

Fig 6 shows rate-distortion (RD) curves evaluated on Kodak and CLIC 2020 at their corresponding optimal α settings. The RD curve of *SCC high-res* consistently falls between *CC high-res* and *CC high-res + low-res*. Thus, SCC is better than encoding both resolutions in separate Cool-Chic bitstreams, but worse than encoding a single resolution using Cool-Chic. Table I lists the corresponding BD-rates and relative decoding times.

TABLE I
BD-RATES AND RELATIVE DECODING TIMES OF OUR METHOD SCC VS. CC WITH $\alpha = 0.4$ FOR KODAK AND $\alpha = 0.3$ FOR CLIC 2020. NEGATIVE VALUES INDICATE BETTER PERFORMANCE FOR SCC.

Dataset	SCC high-res vs. CC high-res + low-res		SCC high-res vs. CC high-res		SCC low-res vs. CC low-res	
	BD-rate	Dec. time	BD-rate	Dec. time	BD-rate	Dec. time
Kodak [21]	-30.10%	6.59%	12.35%	6.59%	13.03%	-1,51%
CLIC 2020 [22]	-27.24%	5.89%	7.97%	5.89%	53.83%	-3.19%

SCC low-res needs higher bitrates than *CC low-res* at the same quality. This makes sense, since minimizing equation (4) aims to minimize the overall bitrate and does not enforce a small bitrate for $\hat{y}_l$. This indicates that either additional information from the high-resolution image is stored in $\hat{y}_l$ or that the shared ARM is suboptimal for encoding $\hat{y}_l$. However, Table II indicates that separate ARMs lead to a lower RD performance which can be explained by the additional signaling overhead to transmit the ARM parameters.

Since the decoding of the low-resolution image is done the same way as it is done in Cool-Chic, the decoding time of the low-resolution image is mostly unaffected by our extensions in SCC. For the high-resolution image, however, the decoding time is increased by up to 7.97%. This is mainly due to the added overhead of the additional upsampling and addition steps. The upsampling process takes 67.56% longer for *SCC high-res* relative to *CC high-res*.

D. Ablation

There are three locations within Fig 1 where the low- and high-resolution information can potentially be merged to reconstruct $\hat{x}_h$: at the latent-pyramid level (*Latent*), at the upsampled latent level (*Ups. Latent*) and at the output level (*Output*). *Ups. Latent* is implemented in SCC. *Latent* upsamples each layer within the latent pyramid $\hat{y}_l$ and merges by addition with each layer of $\hat{y}_h$. *Output* upsamples $\hat{x}_l$ and merges by addition with $\hat{x}_h$. Additionally, all neural networks (ARM, upsample and synthesis) can either use shared weights for high- and low-resolution or not.

Table II gives a full comparison of all these settings and shows that the configuration that we used for SCC yields the best rate-distortion performance.

TABLE II
BD-RATES FOR ALL COMBINATIONS OF SHARED MODEL WEIGHTS AND THREE DIFFERENT MERGE MODES EVALUATED ON [21] WITH $\alpha = 0.5$.

Shared Weights			Merge Mode		
ARM	Upsample	Synthesis	Ups. Latent	Latent	Output
✓	✗	✓	-29.31	-28.40	-20.55
✗	✗	✓	-28.73	-28.13	-19.10
✓	✗	✗	-28.48	-27.04	-22.89
✗	✗	✗	-27.87	-26.72	-20.76
✗	✓	✓	-27.76	-27.38	-18.04
✓	✓	✗	-27.43	-26.24	-22.63
✗	✓	✗	-27.11	-26.30	-20.48
✓	✓	✓	-26.96	-26.36	-19.17

V. CONCLUSION

In this paper, we propose a scalable extension for Cool-Chic, that encodes the same image at two different resolutions into the same bitstream. By this, the total bitrate is reduced by up to 30.10% in terms of BD-rate compared to encoding two independent bitstreams.

REFERENCES

[1] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[2] J. Alakuijala, R. Van Asseldonk, S. Boukortt, M. Bruse, I.-M. Comșa, M. Firsching, T. Fischbacher, E. Kliuchnikov, S. Gomez, R. Obryk *et al.*, "Jpeg xl next-generation image compression architecture and coding tools," in *Applications of digital image processing XLII*, vol. 11137. SPIE, 2019, pp. 112–124.

[3] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012.

[4] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc)," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463–1493, 2021.

[5] D. S. Taubman, M. W. Marcellin, and M. Rabbani, "Jpeg2000: Image compression fundamentals, standards and practice," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 286–287, 2002.

[6] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[7] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint*, 2016.

[8] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint*, 2018.

[9] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Z. Jia, B. Li, J. Li, W. Xie, L. Qi, H. Li, and Y. Lu, "Towards practical real-time neural video compression," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 12 543–12 552.

[11] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, "Coin: Compression with implicit neural representations," *arXiv preprint arXiv:2103.03123*, 2021.

[12] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, "Cool-chic: Coordinate-based low complexity hierarchical image codec," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 515–13 522.

[13] T. Leguay, T. Ladune, P. Philippe, G. Clare, F. Henry, and O. Déforges, "Low-complexity overfitted neural image codec," in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2023, pp. 1–6.

[14] H. Kim, M. Bauer, L. Theis, J. R. Schwarz, and E. Dupont, "C3: High-performance and low-complexity neural compression from a single image or video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9347–9358.

[15] T. Blard, T. Ladune, P. Philippe, G. Clare, X. Jiang, and O. Déforges, "Overfitted image coding at reduced complexity," in *2024 32nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2024, pp. 927–931.

[16] P. Philippe, T. Ladune, G. Clare, F. Henry, T. Blard, and T. Leguay, "Upsampling improvement for overfitted neural coding," *arXiv preprint arXiv:2411.19249*, 2024.

[17] G. Meardi, S. Ferrara, L. Ciccarelli, G. Cobianchi, S. Poularakis, F. Maurer, S. Battista, and A. Byagowi, "Mpeg-5 part 2: Low complexity enhancement video coding (lcevc): Overview and performance evaluation," *Applications of Digital Image Processing XLIII*, vol. 11510, pp. 238–257, 2020.

[18] M. Benjak and J. Ostermann, "Exploration of sequence-wise optimized parameters for low complexity enhancement video coding (lcevc) on 4k content," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[19] M. Benjak, Y.-H. Chen, W.-H. Peng, and J. Ostermann, "Learning-based scalable video coding with spatial and temporal prediction," in *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2023, pp. 1–5.

[20] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 20–34, 2015.

[21] "Kodak image dataset." [Online]. Available: http://r0k.us/graphics/kodak/

[22] "Challenge on learned image coding 2020," 2020. [Online]. Available: https://archive.compression.cc/2021/tasks/index.html