C3: High-performance and low-complexity neural compression from a single image or video

Hyunjik Kim*, Matthias Bauer*, Lucas Theis, Jonathan Richard Schwarz, Emilien Dupont*
Google DeepMind

*Equal contribution

Corresponding authors: {hyunjikk, msbauer, edupont}@google.com

Abstract

Most neural compression models are trained on large datasets of images or videos in order to generalize to unseen data. Such generalization typically requires large and expressive architectures with a high decoding complexity. Here we introduce C3, a neural compression method with strong rate-distortion (RD) performance that instead overfits a small model to each image or video separately. The resulting decoding complexity of C3 can be an order of magnitude lower than neural baselines with similar RD performance. C3 builds on Cool-chic [43] and makes several simple and effective improvements for images. We further develop new methodology to apply C3 to videos. On the CLIC2020 image benchmark, we match the RD performance of VTM, the reference implementation of the H.266 codec, with less than 3k MACs/pixel for decoding. On the UVG video benchmark, we match the RD performance of the Video Compression Transformer [60], a well-established neural video codec, with less than 5k MACs/pixel for decoding.

1. Introduction

Most neural compression models are based on autoencoders [5, 79], with an encoder mapping an image to a quantized latent vector and a decoder mapping the latent vector back to an approximate reconstruction of the image. To be practically useful as codecs, these models must *generalize*, *i.e.*, the decoder should be able to approximately reconstruct any natural image. Such a decoding function is likely to be complex and expensive to compute. Indeed, while most neural codecs enjoy very strong rate-distortion (RD) performance [16, 31, 38], their decoding complexity can make them impractical for many use cases, particularly when hardware is constrained, *e.g.*, on mobile devices [45, 83]. As a result, designing low complexity codecs that offer strong RD performance is one of the major open problems in neural compression [89].

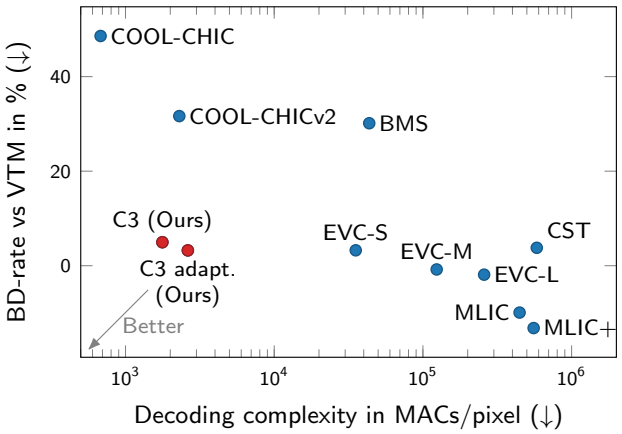

Figure 1. Rate-distortion performance (BD-rate) vs. decoding complexity on the Kodak image benchmark. Our method, C3, achieves a better trade-off than existing neural codecs.

Recently, an alternative approach to neural compression called COIN was proposed [20]. Instead of generalizing across images, COIN *overfits* a neural network to a *single* image. The quantized weights of this neural network (often referred to as a neural field [86]) are then transmitted as a compressed code for the image. As the decoder only needs to reconstruct a single image, the resulting network is significantly smaller than traditional neural decoders [6, 16, 64], reducing the decoding complexity by orders of magnitude. However, while the decoding complexity of COIN is low, its RD performance is poor, and it is therefore not a viable alternative to other codecs.

JRS is now at Harvard University.

Code: https://github.com/google-deepmind/c3_neural_compression

Author contributions: ED conceived the project and wrote the initial codebase with the help of HK and MB. HK, MB, ED, LT developed and refined the project vision with the help of JRS. MB, LT, HK, ED implemented and refined the general methodology. HK designed and implemented the video-specific methodology and evaluation with help from ED and MB. MB, HK, LT worked on scaling, evaluating and improving the efficiency of experiments. MB, ED, HK, LT wrote the paper.

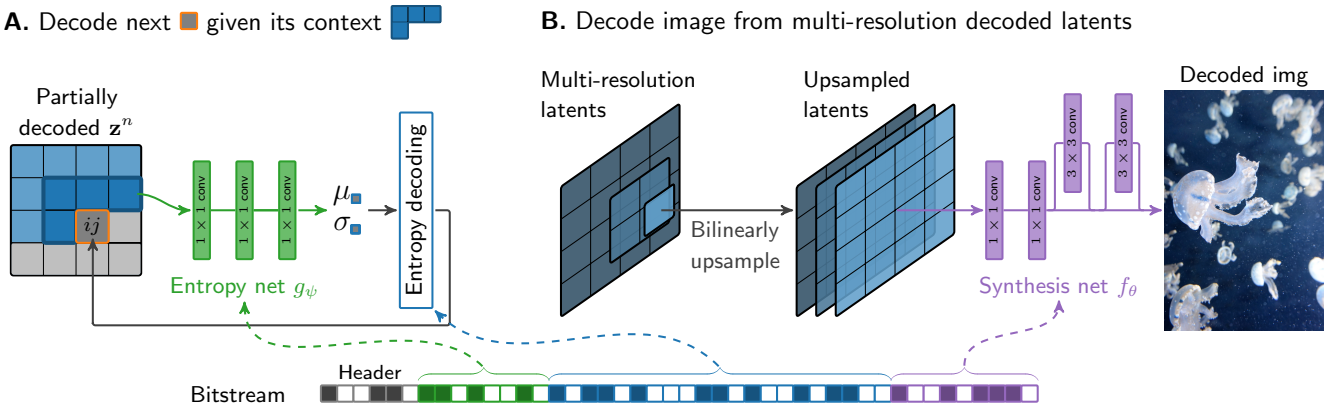

Figure 2. Decoding the bitstream into an image with Cool-chic and C3. **A.** A latent entry $\widehat{\mathbf{z}}^n_{ij}$ ( ) is autoregressively decoded by applying the entropy network g_ψ to the context $\text{context}(\mathbf{z}^n; (i, j))$ ( ). **B.** The decoded latent grids at multiple resolutions are first upsampled and then decoded into image space using the synthesis network f_θ. Figure adapted from Leguay et al. [48].

More recently, Ladune et al. [43] introduced Cool-chic which, in addition to learning a decoder per image like COIN, also learns an *entropy* model per image. This led to significantly improved RD performance while maintaining low decoding complexity. A recent extension of Cool-chic that we refer to as Cool-chic v2 [48] exceeds the RD performance of the widely used BPG/HEVC codec [8, 77] while only requiring 2.3k MACs/pixel at decoding time, an order of magnitude less than the most efficient neural codecs [29] (decoding complexity is measured in number of multiply-accumulate (MAC) operations, cf. App. B for details). Despite these impressive results, the performance of Cool-chic still falls short of the latest classical codecs such as VTM [11]. Further, Cool-chic has not been applied to video, where low decoding cost is of greater importance as fast decoding is required to maintain a satisfactory frame rate for streaming.

In this paper, we introduce C3, a neural compression method that builds on Cool-chic but substantially improves its RD performance while maintaining a low decoding complexity (see Fig. 1). More specifically, we propose a series of simple and effective improvements to the optimization, quantization, and architecture of Cool-chic. These changes reduce the BD-rate [9] compared to Cool-chic v2 by 22.2% while matching VTM on the CLIC2020 benchmark [80]. *To the best of our knowledge, C3 is the first neural compression method to achieve RD performance matching VTM on images while maintaining very low decoding complexity (less than 3k MACs/pixel).* Further, C3 is the state of the art among neural codecs obtained from a single image.

Going beyond COOL-CHIC, which is only applied to images, we also extend C3 to videos, making several crucial methodological changes enabling the application of our method to this modality. On the UVG benchmark [61], we demonstrate strong RD performance that matches VCT [60] while requiring 4.4k MACs/pixel, less than 0.1% of VCT's decoding complexity. We believe this is a promising step towards efficient neural codecs trained on a single video.

2. Background: Cool-chic

Autoencoder based neural compression methods train an encoder network (also known as analysis transform) to compress an image $\mathbf{x}$ into a quantized latent $\widehat{\mathbf{z}}$, and a corresponding decoder network (also known as synthesis transform) to reconstruct $\mathbf{x}$ from $\widehat{\mathbf{z}}$. Typically, the latent $\widehat{\mathbf{z}}$ is the only image-dependent component and is encoded into a bitstream using a shared entropy model P [89].

In contrast, Cool-chic [43] and Cool-chic v2 [48] are methods for single image compression, in which all components are fit to each image separately. In the following we provide further details on Cool-chic.

Overview. At a high level, the Cool-chic model consists of three components (cf. Fig. 2): (i) a set of latent grids at different spatial resolutions $\mathbf{z} = (\mathbf{z}^1, \dots, \mathbf{z}^N)$, (ii) a synthesis transform f_θ to decode these latents into an image, and (iii) an autoregressive entropy-coding network g_ψ that is used to convert the latents into a bitstream. Because the networks do not need to be general, they can be very small, which allows for low decoding complexity. Instead of an analysis transform, Cool-chic uses optimization to jointly fit the latents, the synthesis transform and the entropy network per image. The gradient-based optimization acts on continuous values but is quantization-aware as we describe below; for the final encoding and decoding, the latent and the network parameters are both quantized.

Latent grids. Cool-chic structures the latent $\mathbf{z}$ as a hierarchy of latent grids $(\mathbf{z}^1, \dots, \mathbf{z}^N)$ at multiple spatial resolutions to efficiently capture structure at different spatial frequencies. By default they are of shape $(h, w), (\frac{h}{2}, \frac{w}{2}), \dots, (\frac{h}{2^{N-1}}, \frac{w}{2^{N-1}})$, where h and w are the height and width of the image, respectively.

Synthesis. The synthesis transform f_θ approximately reconstructs the image $\mathbf{x}$ from these latent grids. First, each latent grid $\mathbf{z}^n$ is deterministically upsampled to the resolution of the image. Then, the synthesis network f_θ uses the resulting concatenated tensor $\mathrm{Up}(\mathbf{z})$ of shape (h, w, N) to predict the RGB values of the image, $\mathbf{x}_{\text{rec}} = f_\theta\left(\mathrm{Up}(\mathbf{z})\right)$ (see Fig. 2B). Cool-chic v2 uses learned upsampling and a small convolutional network to parameterize f_θ.

Entropy coding. For transmission, the latent grids and network parameters are quantized via rounding before being entropy-encoded into a bitstream. As this coding cost is dominated by the latent grids, an image-specific entropy model g_ψ is learned to losslessly compress them. Cool-chic uses an integrated Laplace distribution for entropy coding, where the location and scale parameters $(\mu^n_{ij}, \sigma^n_{ij})$ of the distribution for each latent grid element $\mathbf{z}^n_{ij}$ are autoregressively predicted by the entropy network from the local neighborhood of that grid element,

$$P_\psi(\mathbf{z}^n) = \textstyle\prod_{i,j} P\left(\mathbf{z}^n_{ij}; \mu^n_{ij}, \sigma^n_{ij}\right) \quad (1)$$

$$\mu^n_{ij}, \sigma^n_{ij} = g_\psi\left(\text{context}\left(\mathbf{z}^n, (i,j)\right)\right). \quad (2)$$

Here, $\text{context}(\mathbf{z}^n, (i,j))$ extracts a small causally masked neighborhood ($5-7$ latent pixels wide) around a location (i,j) from latent grid $\mathbf{z}^n$ (c.f. Fig. 2A). Individual grids are modelled independently with the same network g_ψ, $P_\psi(\mathbf{z}) = \prod_n P_\psi(\mathbf{z}^n)$.

The entropy and synthesis model are both small networks of depth ≤ 4 and width ≤ 40, and their parameters are quantized after training using different bin widths. The bin width with the best RD trade-off is chosen and added to the bitstream. The quantized network parameters $\widehat{\theta}$ and $\widehat{\psi}$ are also entropy-coded using an integrated Laplace distribution that factorizes over entries with zero mean and scale determined by the empirical standard deviation:

$$P(\widehat{\theta}) = \textstyle\prod_i P(\widehat{\theta}_i; \mu = 0; \sigma = \frac{1}{\sqrt{2}}\mathrm{std}(\widehat{\theta})) \quad (3)$$

and similarly for $\widehat{\psi}$. Entropy coding for the latents and network parameters is performed using a range coder [65].

Quantization-aware gradient-based optimization. The latent ($\mathbf{z}$) and parameters (ψ, θ) are fit to an image $\mathbf{x}$ by jointly optimizing the following RD objective that trades off better reconstructions and more compressible latents with an RD-weight λ:

$$\mathcal{L}_{\theta,\psi}(\mathbf{z}) = \|\mathbf{x} - f_\theta\left(\mathrm{Up}(\mathbf{z})\right)\|_2^2 - \lambda \textstyle\sum_n \log_2 P_\psi(\mathbf{z}^n). \quad (4)$$

The optimization is made quantization-aware in several ways and proceeds in two stages (cf. Tab. 1): in the (longer) first stage, uniform noise $\mathbf{u}$ is added to the continuous latents $\mathbf{z}$; in the (shorter) second stage with low learning rate, the latents $\mathbf{z}$ are quantized and their gradients are approximated with the straight-through estimator, which is biased. Moreover, the rate term uses an integrated Laplace distribution.

Stage 1	$\nabla_{\mathbf{z},\theta,\psi}\mathcal{L}_{\theta,\psi}(\mathbf{z}+\mathbf{u})$; $\quad \mathbf{u} \sim \mathrm{Uniform}(0,1)$
Stage 2	$\nabla_{\theta,\psi}\mathcal{L}_{\theta,\psi}(\lfloor\mathbf{z}\rceil)$ and $\widetilde{\nabla}_{\mathbf{z}}\mathcal{L}_{\theta,\psi}(\lfloor\mathbf{z}\rceil)$

Table 1. Two-stage optimization; $\widetilde{\nabla}_{\mathbf{z}}$ is straight-through estimation.

3. C3: Improving Cool-chic

We first present a series of simple and effective improvements to Cool-chic, which we collectively refer to as C3 (Cooler-ChiC), that lead to a significant increase in RD performance with similar decoding complexity. Maintaining the core model structure (cf. Fig. 2), most of our improvements fall into one of two categories: (1) improvements to the quantization-aware optimization, and (2) improvements to the model architecture. See App. A for full details on all improvements. Subsequently, we introduce the modifications necessary to apply C3 to videos. We confirm with extensive ablations in Sec. 5 and App. D that each contribution is beneficial and that their improvements are cumulative.

3.1. Optimization improvements

We maintain the same two-stage optimization structure of Cool-chic (see Tab. 1) but make several improvements in both stages, most notably how quantization is approximated.

Soft-rounding (stage 1). We apply a soft-rounding function before and after the addition of noise [2]. Let s_T be a smooth approximation of the rounding function whose smoothness is controlled by a temperature parameter T. For large T, s_T approaches the identity while for small T, s_T approaches the rounding function so that

$$\lim_{T\to 0} s_T(s_T(\mathbf{z}) + \mathbf{u}) = \lfloor\lfloor\mathbf{z}\rceil + \mathbf{u}\rceil = \lfloor\mathbf{z}\rceil. \quad (5)$$

By varying T we can interpolate between rounding and the simple addition of uniform noise $\mathbf{u}$. Note that the soft-rounding does not create an information bottleneck as it is an invertible function. Therefore, adding noise is still necessary for reliable compression [2].

Small T leads to a better approximation of rounding but increases the variance of gradients for $\mathbf{z}$. Following previous work using soft-rounding, we therefore anneal the temperature over the course of the optimization. See Fig. 3 for a visualization and App. A.2.4 for details.

Kumaraswamy noise (stage 1). The addition of uniform noise as an approximation to rounding has been motivated by pointing out that for sufficiently smooth distributions, the marginal distribution of the quantization error $(\mathbf{z} - \lfloor\mathbf{z}\rceil)$ is approximately uniform [5]. The approximation further assumes that the quantization error and the input are uncorrelated. In practice, these assumptions may be violated,

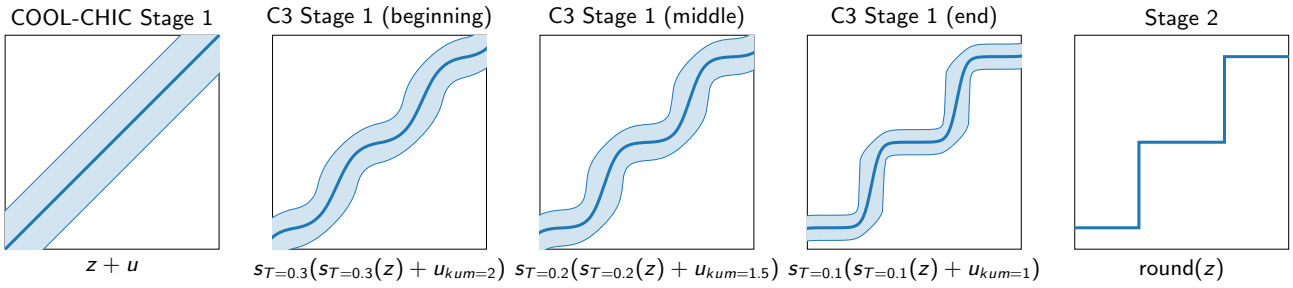

Figure 3. Approximating the $\mathrm{round}(\mathbf{z})$ function during Stage 1 of optimization. Cool-chic adds uniform noise $\mathbf{u}$, whereas C3 uses soft-rounding s_T with varying temperatures T and Kumaraswamy noise of different strengths, $\mathbf{u}_{\text{kum}}$. We plot the mean and 95% interval.

suggesting that other forms of noise are worth exploring. To that end, we replace uniform noise with samples from the Kumaraswamy distribution [41] whose support is compact on $[0, 1]$. This distribution is very similar to the Beta distribution but has an analytic cumulative distribution function that allows for more efficient sampling. By controlling its shape parameters we can interpolate between a peaked (lower noise) distribution at beginning of stage 1 and a uniform distribution at the end. See Fig. 3 for a visualization and App. A.2.5 for details.

Cosine decay schedule (stage 1). We found that a simple cosine decay schedule for the learning rate of the Adam optimizer performed well during the first stage of optimization.

Smaller quantization step (stages 1 & 2). Cool-chic quantizes the latents by rounding their values to the nearest whole integer; as a result the inputs to the synthesis and entropy networks can become large (exceeding values of 50), which can lead to instabilities or suboptimal optimization. We found that quantizing the latents in smaller steps than 1 (and correspondingly rescaling the soft-rounding in both stages) empirically improved optimization.

Soft-rounding for gradient (stage 2). We apply hard rounding/quantization to the latents $\mathbf{z}$ for the forward pass of stage 2 following Cool-chic. For the backward pass, Cool-chic v2 uses a straight-through gradient estimator and multiplies the gradient by a small ϵ. This has the effect of replacing rounding by a linear function (cf. Fig. 3) and downscaling the learning rate of the latents. Instead we use soft-rounding to estimate the gradients (with a very low temperature) and start stage 2 with a low learning rate.

Adaptive learning rate (stage 2). We adaptively decrease the learning rate further when the RD-loss does not improve for a fixed number of steps.

3.2. Model improvements

We make a number of changes to the network architectures to increase their expressiveness, support the optimization, and allow for more adaptability depending on the bitrate.

Conditional entropy model. Cool-chic uses the same entropy network to independently model latent grids of starkly varying resolutions. We explored several options to increase the expressiveness of the entropy model: first, we optionally allow the context at a particular latent location to also include values from the previous grid, $P(\mathbf{z}^n|\mathbf{z}^{n-1})$, as this information may be helpful for prediction when different grids are correlated. Second, we optionally allow the network to be resolution-dependent by either using a separate network per latent grid or using FiLM [67] layers to make the network resolution-dependent in a more parameter-efficient way.

ReLU $\rightarrow$ GELU. As we are constrained to use very small networks, we replace the simple ReLU activation function with a more expressive activation; empirically we found that GELU activations [33] worked better.

Shift log-scale of entropy model output. Small changes in how quantities are parameterized can affect optimization considerably. For example, how the scale of the entropy distribution is computed from the raw network output strongly affects optimization dynamics, in particular at initialization; we found that shifting the predicted log-scale prior to exponentiation consistently improves performance. With improved optimization we can also use larger initialization scales than Cool-chic to improve performance.

Adaptivity. We optionally sweep over several architecture choices per image or video patch to find the best RD-trade-off on a per-instance basis. We refer to this as *C3 adaptive*. This setting includes an option to vary the relative latent resolutions; *e.g.*, it may be beneficial not to use the highest resolution latent grid for low bitrates. Note that such adaptive settings are also common in traditional codecs [37, 77].

3.3. Video-specific methodology

Cool-chic has been successfully applied to images but not videos. Here we describe our methodology for applying C3 (and Cool-chic) to video, which we use on top of the improvements in Sec. 3.1 and Sec. 3.2.

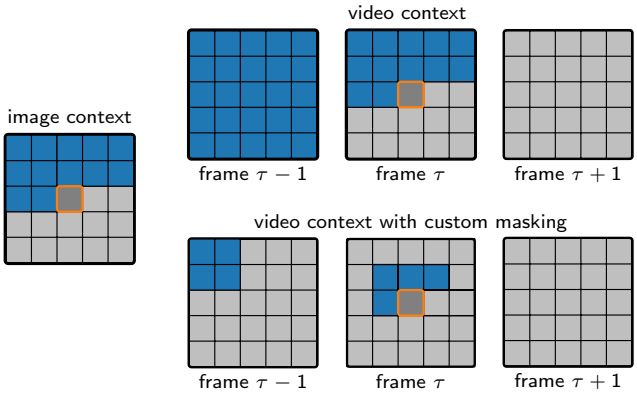

Figure 4. Visualization of entropy model's context for images and video (with and without custom masking).

2D $\rightarrow$ 3D. Given that videos have an extra time dimension compared to images, a natural way to extend C3 to video is to convert 2D parameters and operations to their 3D counterparts. Namely, we use 3D latent grids $\mathbf{z}^1, \mathbf{z}^2, \ldots$ of shapes $(t, h, w), (\frac{t}{2}, \frac{h}{2}, \frac{w}{2}), \ldots$, and the entropy model's context $\text{context}(\mathbf{z}^n, (\tau, i, j))$ is now a 3D causal neighborhood of the latent entry $\mathbf{z}^n_{\tau ij}$ (cf. Fig. 4 video context).

Using video patches. Videos have orders of magnitude more pixels than images, and a full HD video does not fit into the memory of modern GPUs. We therefore split the video into smaller video patches, and fit a separate C3 model to each patch. We find that larger patches work best for lower bitrates and smaller patches work best for higher bitrates. Our patch sizes range from $(30, 180, 240)$ to $(75, 270, 320)$.

Wider context to capture fast motion. For video patches with fast motion, the small context size that works well for images (5-7 latent pixels wide) can be smaller than the displacement of a particular keypoint in consecutive frames. This means that for a target latent pixel, the context in the previous latent frame does not contain the relevant information for the entropy model's prediction. Hence we use a wider spatial context (up to 65 latent pixels wide) to enhance predictions for videos with faster motion.

Custom masking. Naïvely increasing the context width also increases the parameter count of the entropy model, which scales linearly with the context size. However, most of the context dimensions are irrelevant for prediction and can be masked out. We use a small causal mask centered at the target latent pixel for the current latent frame, and a small rectangular mask for the previous latent frame whose position is learned during encoding time (cf. Fig. 4 video context with custom masking). See App. A.3 for details of how the position of this mask is learned.

4. Related work

Neural compression by overfitting to a single instance. COIN [20] introduced the idea of overfitting a neural network to a single image as a means for compression. This has since been improved with reduced encoding times through meta-learning [21, 71, 76] and increased RD performance via better architectures [13] or more refined quantization [18, 26]. Further improvements to RD performance have been achieved by pruning networks [46, 69, 71] and incorporating traditional compressive autoencoders [68, 72]. Recent approaches using Bayesian neural fields directly optimize RD losses, further improving performance [28, 32]. Despite this progress, no approach yet matches the RD performance of traditional codecs such as VTM.

For video, NeRV [14] overfits neural fields to single videos, using a deep convolutional network to map time indices to frames. Various follow-ups have greatly improved compression performance [4, 15, 25, 42, 47, 53, 58], among which HiNeRV [42] shows impressive RD performance that closely matches HEVC (HM-18.0, random access setting) on standard video benchmarks. While these models are typically smaller than autoencoder-based neural codecs, the model size (and hence decoding complexity) is directly correlated with the bitrate (each point on the RD curve corresponds to a different model size), making it challenging to design a low-complexity codec at high bitrates. Further, these models are typically unsuitable for video-streaming applications, as the entire bitstream needs to be transmitted before the first frame can be decoded [81]. Note that C3 does not suffer from this limitation – the very small synthesis and entropy models can be transmitted first with little overhead, and then be used to decode the bitstream for the latents that can be synthesized into frames in a streaming fashion.

Given the generality of neural fields, codecs applicable to multiple modalities have been developed [21, 24, 71, 72]. There also exist methods specialized to other modalities: climate data [34], 3D shapes [19, 35, 55], NeRF scenes [24, 52, 78], audio [28, 44] and medical images [21, 23, 59, 73].

Instance adaptive neural compression. Several autoencoder-based approaches adapt the encoder to each instance through optimization but leave the decoder fixed [12, 27, 54, 88]. Such methods generally perform worse than approaches that optimize both the encoder and decoder w.r.t. an RD loss [57, 62, 81, 82]. In particular, Van Rozendaal et al. [81] introduce Insta-SSF, an instance adaptive version of the scale-space flow (SSF) model [3] (a popular autoencoder model for neural video compression). For a fixed RD performance, the decoder of Insta-SSF is much smaller and has lower complexity than the shared decoder of SSF. Note that C3 and Cool-chic follow the same principle for low complexity decoding. However, there are key differences between C3/Cool-chic and the

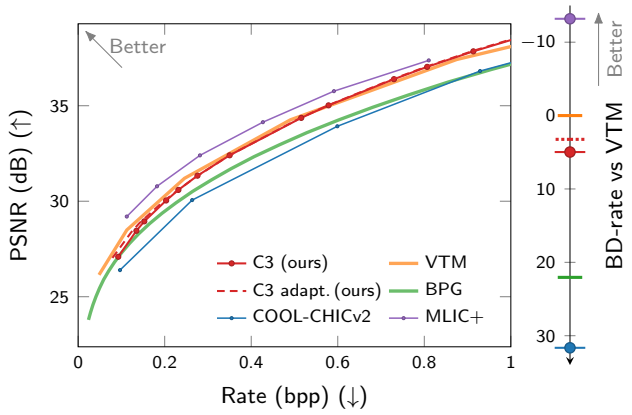

Figure 5. Rate-distortion curve and BD-rate on Kodak.

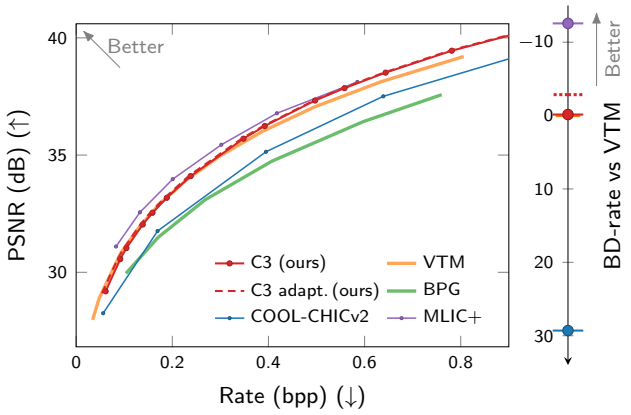

Figure 6. Rate-distortion curve and BD-rate on CLIC2020.

aforementioned instance adaptive methods: 1. we train from scratch rather than learning an initialization from a dataset; 2. we use a neural field model (without encoder) instead of an autoencoder, and show an order of magnitude lower decoding complexity; 3. for videos, there is no explicit motion compensation based on flows in our model.

Low complexity neural codecs. While the problem of high decoding complexity in neural compression is well established [89], most works to mitigate it are relatively recent. Early methods reduced complexity at little cost in RD performance by pruning network weights [39]. More recently, He et al. [30, 31] replace traditional autoregressive entropy models with checkerboard-based designs that allow for more efficient and parallelizable entropy coding. Further, Yang and Mandt [87] use shallow decoders to reduce decoding complexity and offset the resulting decrease in RD performance with iterative encoding. EVC [29] achieves RD performance surpassing VTM on images with decoding at 30FPS on a GPU, by carefully choosing architectures and using sparsity-based mask decay. Despite these impressive results, the decoding complexity required for these models is still an order of magnitude higher than C3.

For video, some prior works [45, 83] focus on providing efficient neural components and entropy coding that run on mobile devices. Due to these constraints, their RD performance is not yet competitive with most neural video codecs and their decoding complexity is an order of magnitude higher than C3. ELF-VC [70], based on autoencoders and flows, provides gains in efficiency by encoder/decoder asymmetry and an efficient convolutional architecture. However they do not report decoding complexity and are outperformed by VCT [60] in terms of RD. AlphaVC [75] introduces a technique to skip latent dimensions for entropy coding, improving efficiency in flow-based autoencoder models and surpassing VTM (low-delay) in terms of RD performance, albeit with a high decoding complexity of 1M MACs/pixel.

5. Results

5.1. Image compression

We evaluate our model on the Kodak [40] and CLIC2020 [80] benchmarks. Kodak contains 24 images at a resolution of 512×768. For CLIC2020, we use the professional validation dataset split containing 41 images at various resolutions from 439×720 to 1370×2048, following Cool-chic [43, 48]. We compare C3 against a series of baselines, including classical codecs (BPG [8], VTM [11]), autoencoder based neural codecs (BMS [6], a standard neural codec; CST [16], a strong neural codec; EVC [29], a codec optimized for RD performance and low decoding complexity; MLIC+ [38], the state of the art in terms of RD performance) and Cool-chic v2 [48]. We measure PSNR on RGB and quantify differences in RD performance with the widely used BD-rate metric. See App. A for full experimental details and App. B for full evaluation details.

Rate-distortion and decoding complexity. On CLIC2020, C3 (with a single setting for its architecture and hyperparameters) significantly outperforms Cool-chic v2 across all bitrates (-20.8% BD-rate) and matches VTM (-0.1% BD-rate), cf. Fig. 6. When adapting the model per image, C3 even outperforms VTM (-2.9% BD-rate). *To the best of our knowledge, this is the first time a neural codec has been able to match VTM while having very low decoding complexity (below 3k MACs/pixel).* While C3 does not yet match the RD performance of state of the art neural codecs such as MLIC+, it uses two orders of magnitude fewer operations at decoding time, making it substantially cheaper. Results are also strong on Kodak (see Fig. 5), although, as is the case for Cool-chic, we perform slighly worse on this dataset relative to VTM. In Fig. 1 we compare the decoding complexity (measured in MACs/pixel) and the achieved BD-rate for C3 and other neural baselines. C3 has a similar complexity to Cool-chic but much better

BD-rate, and codecs achieving similar BD-rate to ours require at least an order of magnitude more MACs (even ones optimized for low decoding complexity such as EVC). See App. C.1 for comparisons with additional baselines (including other autoencoder based and overfitted codecs) in terms of RD performance and decoding complexity.

Decoding time. A concern with using autoregressive models is that runtimes may be prohibitive despite low computational complexity [63]. To address this, we time the decoding process, which includes a full iterative roll-out of the autoregressive entropy model (and the upsampling and application of the synthesis network). On CPU (Intel Xeon Platinum, Skylake, 2GHz) these together take < 100 ms (~ 55 ms and ~ 30 ms, respectively) for an image of size 768×512. This does not account for the cost of range-decoding the bitstream (which is also a component of every classical codec). We emphasize that these numbers are based on unoptimized research code and can likely be improved substantially.

Encoding time. C3 faces the same limitations as Cool-chic, in that it has very long encoding times. Here we report encoding times on an NVIDIA V100 GPU. The largest CLIC image at 1370×2048 resolution takes 48s per 1000 iterations of optimization (*i.e.*, excluding range-encoding) with the slowest setting (largest architecture), and 22s per 1000 iterations with the fastest setting (smallest architecture). While we train for a maximum of 110k iterations, we show in App. D.3 that we can approach similar RD performance with much fewer iterations. As we run unoptimized research code, we believe the runtime can be greatly improved.

Ablations. In Tab. 2, we ablate our methodological contributions on Kodak by starting with our best performing model and sequentially removing each of our improvements, one after another. We show the resulting BD-rate with respect to the top row, demonstrating that our contributions stack to yield significant improvements in RD performance. In Tab. 3, we show BD-rate with respect to C3 when disabling individual features. We find that soft-rounding, Kumaraswamy noise and using GELU activations are responsible for the majority of the improvement. For the corresponding ablations on CLIC2020, please refer to App. D.1.

Qualitative comparisons In Fig. 7, we compare reconstructions from C3 and Cool-chic v2 on an image from CLIC2020, showing that C3 has fewer artifacts. See App. E for a more thorough comparison.

5.2. Video compression

We evaluate C3 on the UVG-1k dataset [61] containing 7 videos at HD resolution (1080×1920) with a total of 3900 frames. We evaluate PSNR on RGB, and compare against a series of baselines, including classical codecs (HEVC medium, no B-frames [77]), neural codecs based on overfitting (HiNeRV [42], FFNeRV [47]) and autoencoder based neural codecs (DCVC [49], VCT [60], Insta-SSF [82], MIMT [85]), among which MIMT reports state of the art RD performance on the UVG-1k dataset. Note that extensions of DCVC [50, 51, 74] also show strong RD performance but report results on a subset of UVG frames, hence we do not compare against them. See App. A for full experimental details and App. B for full evaluation details.

Model variant	BD-rate vs. C3 Adaptive
C3 (adaptive)	0.0%
C3	2.2%
✗ Quantiz. step < 1	2.6%
✗ Adaptive lr (stage 2)	3.4%
✗ Shift log-scale	4.2%
✗ GELU	12.6%
✗ Kumaraswamy noise	23.6%
✗ Soft-rounding	39.8%

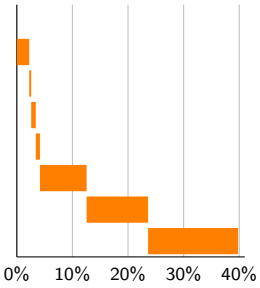

Table 2. Kodak ablation *sequentially* removing one improvement after another. Note higher BD-rate means worse RD performance.

Removed feature	BD-rate vs. C3
Soft-rounding	22.18%
Kumaraswamy noise	3.90%
GELU	3.27%
Shifted log-scale	0.87%
Adaptive lr (stage 2)	0.68%
Quantization step < 1	0.40%

Table 3. Kodak ablation knocking out individual features from C3 (fixed hyperparameters across all images). Note higher BD-rate means worse RD performance.

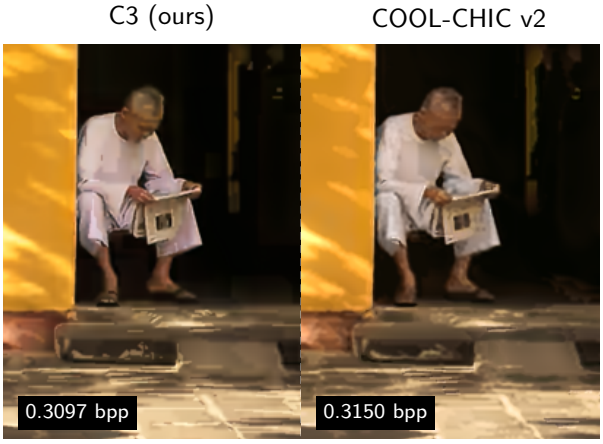

Figure 7. Qualitative comparison of compression artifacts for C3 and Cool-chic v2 at around 0.31 bpp with a PSNR of 30.28dB and 28.98dB, respectively. See App. E for the full image.

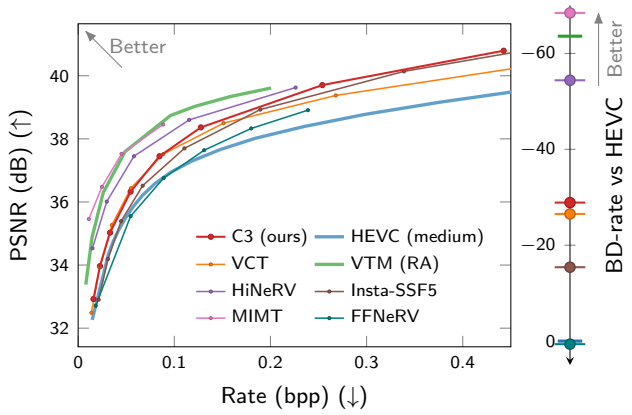

Figure 8. Rate-distortion curve and BD-rate on UVG.

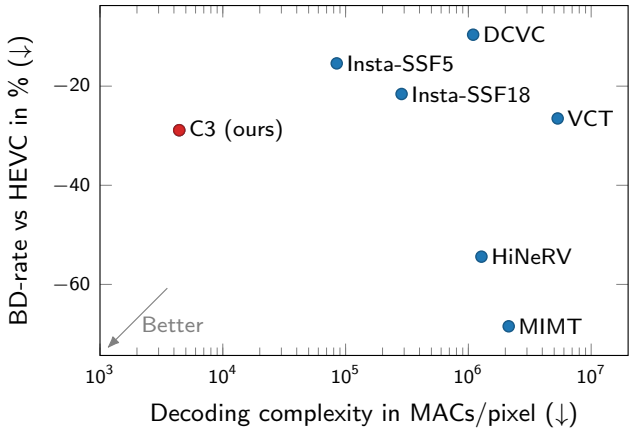

Figure 9. BD-rate vs decoding complexity relative to HEVC (medium) on UVG. For methods with varying MACs for different bitrates (*e.g.*, C3 and HiNeRV), we report the largest MACs/pixel.

Rate-distortion and decoding complexity In Fig. 8, we show the RD performance of C3 compared to other baselines, with more baselines shown in App. C.1. In Fig. 9, we show the MACs/pixel count of each method vs the BD-rate using HEVC (medium, no B-frames) [77] as anchor. In terms of RD performance, we are on par with VCT [60], a competitive neural baseline, while requiring 4.4k MACs/pixel, which is less than 0.1% of VCT's MACs/pixel. Among the baselines that overfit to a single video instance (NeRV and its followups) we are second best in terms of RD, widely outperforming FFNeRV [47], the previous runner up. Although C3 is behind stronger neural baselines such as HiNeRV and MIMT in terms of RD performance, our decoding complexity is orders of magnitude lower. Note that NeRV-based methods have different model sizes (and hence different MACs/pixel) for each point on the RD curve. For example, the 5 points on the RD curve for HiNeRV correspond to MACs/pixel values between 87k-1.2M [42]. In App. D.4 we show ablation studies showing the effectiveness of our video-specific methodology.

Encoding times. We also report encoding times for video patches on an NVIDIA V100 GPU. The slowest setting on the largest video patch of size $75 \times 270 \times 320$ resolution takes 457s per 1000 iterations of optimization, whereas the fastest setting on the smallest video patch of size $30 \times 180 \times 240$ takes 29s per 1000 iterations. We train for a maximum of 110k iterations but show in App. D.3 that we can approach similar RD performance with much fewer iterations.

6. Conclusion, limitations and future work

We propose C3, the first low complexity neural codec on single images that is competitive with VTM while requiring an order of magnitude fewer MACs for decoding than state of the art neural codecs. We then extend C3 to the video setting, where we match the RD performance of VCT with less than 0.1% of their decoding complexity. Our contributions are a step towards solving one of the major open problems in neural compression — achieving high compression performance with low decoding complexity — and ultimately towards making neural codecs a practical reality.

Limitations. In this paper, we focused on maximizing RD performance while minimizing decoding complexity. As a result, the encoding of C3 is slow, making it impractical for use cases requiring real time encoding. Yet, there are several use cases for which paying a significant encoding cost upfront can be justified if RD performance and decoding time are improved. For example, a popular video on a streaming service is encoded once but decoded millions of times [1]. Further, the autoregressive entropy model used during decoding is inherently sequential in nature, posing challenges for efficient use of hardware designed for parallel computing. However, as shown in Sec. 5, even with unoptimized research code, an image can be decoded relatively quickly on CPU due to the very small network sizes. Moreover, further optimizations and specialized implementations such as wavefront decoding [17] can likely speed up decoding times significantly. Nevertheless, it would be interesting to explore alternative probabilistic models that can be efficiently evaluated on relevant hardware.

Future work. There are several promising avenues for future work. Firstly, it would be interesting to accelerate encoding via better initializations or meta-learning [21, 71, 76]. Secondly, improving decoding speed through the use of different probabilistic models or decoding schemes is an important direction. Further, while we took an extreme view of using only a single image or video to train our models, it is likely that some level of sharing across images or videos could be beneficial. For example, sharing parts of the entropy or synthesis model may improve RD performance.

References

[1] Anne Aaron, Zhi Li, Megha Manohara, Jan De Cock, and David Ronca. Per-title encode optimization. https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2, 2015. [Online; accessed 26-Feb-2021]. 8

[2] Eirikur Agustsson and Lucas Theis. Universally quantized neural compression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 3, 17

[3] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2020. 5

[4] Yunpeng Bai, Chao Dong, Cairong Wang, and Chun Yuan. Ps-nerv: Patch-wise stylized neural representations for videos. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 41–45. IEEE, 2023. 5

[5] J Ballé, V Laparra, and E P Simoncelli. End-to-end optimized image compression. In *Int'l Conf on Learning Representations (ICLR)*, Toulon, France, 2017. Available at http://arxiv.org/abs/1611.01704. 1, 3

[6] 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. 1, 6, 15, 27

[7] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 28

[8] Fabrice Bellard. Bpg image format. *URL https://bellard.org/bpg*, 1(2):1, 2015. 2, 6

[9] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001. 2, 26

[10] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 27

[11] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 2, 6, 28

[12] Joaquim Campos, Meierhans Simon, Abdelaziz Djelouah, and Christopher Schroers. Content adaptive optimization for neural image compression. In *CVPR Workshop and Challenge on Learned Image Compression*, 2019. 5

[13] Lorenzo Catania and Dario Allegra. Nif: A fast implicit image compression with bottleneck layers and modulated sinusoidal activations. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9022–9031, 2023. 5

[14] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 5, 28

[15] Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav 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. 5, 28, 38

[16] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro 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*, pages 7939–7948, 2020. 1, 6, 27

[17] Gordon Clare, Félix Henry, and Stéphane Pateux. Wavefront parallel processing for hevc encoding and decoding. *document JCTVC-F274*, 2011. 8

[18] Bharath Bhushan Damodaran, Muhammet Balcilar, Franck Galpin, and Pierre Hellier. Rqat-inr: Improved implicit neural image compression. In *2023 Data Compression Conference (DCC)*, pages 208–217. IEEE, 2023. 5

[19] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 5

[20] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 1, 5, 28

[21] 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. 5, 8, 28

[22] The fvcore contributors. fvcore. https://github.com/facebookresearch/fvcore. 27

[23] 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)*, pages 1–5. IEEE, 2023. 5

[24] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17513–17524, 2023. 5

[25] Carlos Gomes, Roberto Azevedo, and Christopher Schroers. Video compression with entropy-constrained neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18497–18506, 2023. 5

[26] 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*, pages 341–350, 2023. 5

[27] Tiansheng Guo, Jing Wang, Ze Cui, Yihui Feng, Yunying Ge, and Bo Bai. Variable rate image compression with content adaptive optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 122–123, 2020. 5

[28] 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. 5

[29] Wang Guo-Hua, Jiahao Li, Bin Li, and Yan Lu. EVC: Towards real-time neural image compression with mask decay. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 6, 27

[30] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 6

[31] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5718–5727, 2022. 1, 6, 28

[32] Jiajun He, Gergely Flamich, Zongyu Guo, and José Miguel Hernández-Lobato. Recombiner: Robust and enhanced compression with bayesian implicit neural representations. *arXiv preprint arXiv:2309.17182*, 2023. 5, 28

[33] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4, 13

[34] Langwen Huang and Torsten Hoefler. Compressing multidimensional weather and climate data into neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. 5

[35] Berivan Isik, Philip A Chou, Sung Jin Hwang, Nick Johnston, and George Toderici. Lvac: Learned volumetric attribute compression for point clouds using coordinate based networks. *Frontiers in Signal Processing*, 2:1008812, 2022. 5

[36] Itseez. Open source computer vision library. https://github.com/opencv/opencv, 2015. 19

[37] ITU-T. Recommendation ITU-T T.81: Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines, 1992. 4

[38] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang. Mlic: Multi-reference entropy model for learned image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, page 7618–7627, New York, NY, USA, 2023. Association for Computing Machinery. 1, 6, 27

[39] Nick Johnston, Elad Eban, Ariel Gordon, and Johannes Ballé. Computationally efficient neural image compression. *arXiv preprint arXiv:1912.08771*, 2019. 6

[40] Kodak. Kodak Dataset. http://r0k.us/graphics/kodak/, 1991. 6

[41] P. Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1):79–88, 1980. 4, 18

[42] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. Hinerv: Video compression with hierarchical encoding based neural representation, 2023. 5, 7, 8, 28, 38

[43] Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. Cool-chic: Coordinate-based low complexity hierarchical image codec. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13515–13522, 2023. 1, 2, 6, 13, 15, 16, 17, 27

[44] Luca A Lanzendörfer and Roger Wattenhofer. Siamese siren: Audio compression with implicit neural representations. *arXiv preprint arXiv:2306.12957*, 2023. 5

[45] Hoang Le, Liang Zhang, Amir Said, Guillaume Sautiere, Yang Yang, Pranav Shrestha, Fei Yin, Reza Pourreza, and Auke Wiggers. Mobilecodec: neural inter-frame video compression on mobile devices. In *Proceedings of the 13th ACM Multimedia Systems Conference*, pages 324–330, 2022. 1, 6

[46] 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. 5

[47] Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Ffnerv: Flow-guided frame-wise neural representations for videos. 2023. 5, 7, 8

[48] Thomas Leguay, Théo Ladune, Pierrick Philippe, Gordon Clare, and Félix Henry. Low-complexity overfitted neural image codec. *arXiv preprint arXiv:2307.12706*, 2023. 2, 6, 13, 16, 27, 33

[49] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021. 7, 28

[50] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1503–1511, 2022. 7

[51] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023. 7

[52] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Liefeng Bo. Compressing volumetric radiance fields to 1 mb. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4222–4231, 2023. 5

[53] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022. 5

[54] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. Content adaptive and error propagation aware deep video compression. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 456–472. Springer, 2020. 5

[55] Yuzhe Lu, Kairong Jiang, Joshua A Levine, and Matthew Berger. Compressive neural representations of volumetric scalar fields. In *Computer Graphics Forum*, pages 135–146. Wiley Online Library, 2021. 5

[56] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, pages 674–679, 1981. 19

[57] Yue Lv, Jinxi Xiang, Jun Zhang, Wenming Yang, Xiao Han, and Wei Yang. Dynamic low-rank instance adaptation for universal neural image compression. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 632–642, 2023. 5

[58] Shishira R Maiya, Sharath Girish, Max Ehrlich, Hanyu Wang, Kwot Sin Lee, Patrick Poirson, Pengxiang Wu, Chen Wang, and Abhinav Shrivastava. Nirvana: Neural implicit representations of videos with adaptive networks and autoregressive patch-wise modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14378–14387, 2023. 5

[59] Matteo Mancini, Derek K Jones, and Marco Palombo. Lossy compression of multidimensional medical images using sinusoidal activation networks: An evaluation study. In *International Workshop on Computational Diffusion MRI*, pages 26–37. Springer, 2022. 5

[60] Fabian Mentzer, George Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and Eirikur Agustsson. VCT: A video compression transformer. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 6, 7, 8, 26, 28

[61] Alexandre Mercat, Marko Viitanen, and Jarno 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. 2, 7

[62] Yu Mikami, Chihiro Tsutake, Keita Takahashi, and Toshiaki Fujii. An efficient image compression method based on neural network: An overfitting approach. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2084–2088. IEEE, 2021. 5

[63] David Minnen and Nick Johnston. Advancing the rate-distortion-computation frontier for neural image compression. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2940–2944. IEEE, 2023. 7

[64] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 1, 27, 28

[65] G. Nigel and N. Martin. Range encoding: An algorithm for removing redundancy from a digitized message. In *Video & Data Recording Conference*, 1979. 3

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 27

[67] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 4

[68] Tuan Pham, Yibo Yang, and Stephan Mandt. Autoencoding implicit neural representations for image compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023. 5

[69] Juan Ramirez and Jose Gallego-Posada. L_0onie: Compressing coins with l_0-constraints, 2022. 5

[70] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14479–14488, 2021. 6, 28

[71] Jonathan Schwarz and Yee Whye Teh. Meta-learning sparse compression networks. *Transactions on Machine Learning Research*, 2022. 5, 8, 28

[72] Jonathan Richard Schwarz, Jihoon Tack, Yee Whye Teh, Jaeho Lee, and Jinwoo Shin. Modality-agnostic variational compression of implicit neural representations. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023. 5, 28

[73] Armin Sheibanifard and Hongchuan Yu. A novel implicit neural representation for volume data. *Applied Sciences*, 13 (5):3242, 2023. 5

[74] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 2022. 7

[75] Yibo Shi, Yunying Ge, Jing Wang, and Jue Mao. Alphavc: High-performance and efficient learned video compression. In *European Conference on Computer Vision*, pages 616–631. Springer, 2022. 6

[76] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022. 5, 8, 28

[77] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. 2, 4, 7, 8, 28

[78] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 5

[79] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 1

[80] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020. 2, 6

[81] Ties Van Rozendaal, Johann Brehmer, Yunfan Zhang, Reza Pourreza, Auke Wiggers, and Taco S Cohen. Instance-adaptive video compression: Improving neural codecs by training on the test set. *arXiv preprint arXiv:2111.10302*, 2021. 5, 28

[82] Ties van Rozendaal, Iris AM Huijben, and Taco Cohen. Overfitting for fun and profit: Instance-adaptive data compression. In *International Conference on Learning Representations*, 2021. 5, 7

[83] Ties van Rozendaal, Tushar Singhal, Hoang Le, Guillaume Sautiere, Amir Said, Krishna Buska, Anjuman Raha, Dimitris Kalatzis, Hitarth Mehta, Frank Mayer, et al. Mobilenvc: Real-time 1080p neural video compression on a mobile device. *arXiv preprint arXiv:2310.01258*, 2023. 1, 6

[84] Adam Wieckowski, Jens Brandenburg, Tobias Hinz, Christian Bartnik, Valeri George, Gabriel Hege, Christian Helmrich, Anastasia Henkel, Christian Lehmann, Christian Stoffers,

Ivan Zupancic, Benjamin Bross, and Detlev Marpe. Vvenc: An open and optimized vvc encoder implementation. In *Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–2. 28

[85] Jinxi Xiang, Kuan Tian, and Jun Zhang. Mimt: Masked image modeling transformer for video compression. In *The Eleventh International Conference on Learning Representations*, 2023. 7

[86] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 1

[87] Yibo Yang and Stephan Mandt. Computationally-efficient neural image compression with shallow decoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 530–540, 2023. 6

[88] Yibo Yang, Robert Bamler, and Stephan Mandt. Improving inference for neural image compression. *Advances in Neural Information Processing Systems*, 33:573–584, 2020. 5

[89] Yibo Yang, Stephan Mandt, Lucas Theis, et al. An introduction to neural data compression. *Foundations and Trends® in Computer Graphics and Vision*, 15(2):113–200, 2023. 1, 2, 6

[90] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17492–17501, 2022. 28