Neural Inter-Frame Compression for Video Coding

Abdelaziz Djelouah[1] Joaquim Campos[1] Simone Schaub-Meyer[1,2] Christopher Schroers[1]
[1]DisneyResearch|Studios [2]Department of Computer Science, ETH Zurich
abdelaziz.djelouah@disney.com christopher.schroers@disney.com

Abstract

While there are many deep learning based approaches for single image compression, the field of end-to-end learned video coding has remained much less explored. Therefore, in this work we present an inter-frame compression approach for neural video coding that can seamlessly build up on different existing neural image codecs. Our end-to-end solution performs temporal prediction by optical flow based motion compensation in pixel space. The key insight is that we can increase both decoding efficiency and reconstruction quality by encoding the required information into a latent representation that directly decodes into motion and blending coefficients. In order to account for remaining prediction errors, residual information between the original image and the interpolated frame is needed. We propose to compute residuals directly in latent space instead of in pixel space as this allows to reuse the same image compression network for both key frames and intermediate frames. Our extended evaluation on different datasets and resolutions shows that the rate-distortion performance of our approach is competitive with existing state-of-the-art codecs.

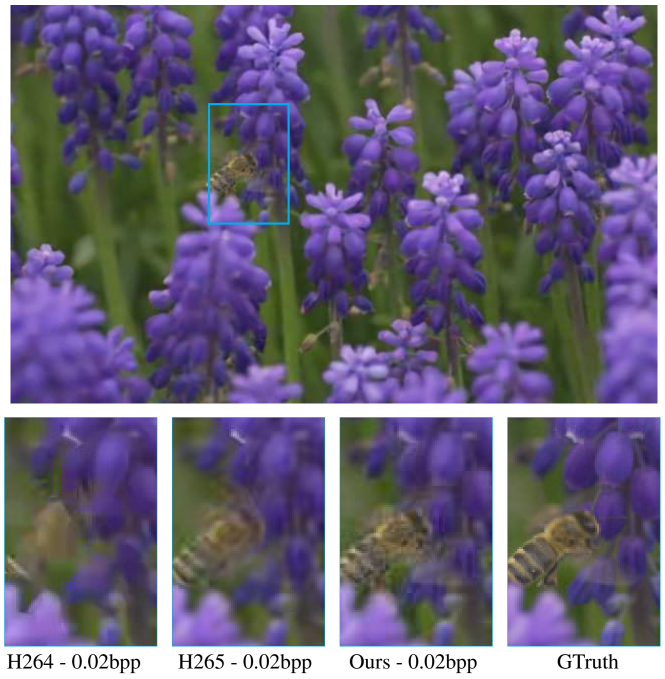

Figure 1: **Learning based video compression.** Compared to existing video codecs, our method achieves better results at a similar or lower bit rate. Less visual artifacts are present and colors are closer to the original (best viewed on screen).

1. Introduction

In 2017 video content already represented 75% of the total internet traffic and it is projected to reach 82% by 2022 [7]. This is due to an expected increase in subscribers to streaming services, higher resolution, frame rate, and dynamic range. As a result, video compression techniques are challenged in handling this data efficiently and with low loss of visual quality.

Although important progress has been made with the different generations of video codecs through thorough testing and exploration of variations [24, 31], they all follow similar strategies. However, very recently some fundamentally different directions to video coding that rely on deep learning have been proposed [11, 14, 23, 32]. For example Wu *et al.* [32] propose a recurrent strategy for interpolation based compression but optical flow is still encoded with a traditional method [10]. The variational approach of Han *et al.* [11] considers video compression from the variational inference perspective but focuses on small resolution videos. The most recent works [23, 14] target the low latency setup that aims at reducing the amount of delay in encoding by only considering frames from the past during motion compensation. These methods require a complex spatial adaptation of the bit-rate [23] or more computation on the decoder side with a refinement network during motion compensation [14].

In this work, we propose a framework for interpolation based video compression that is compatible with neural image compression methods [4, 17, 15]. The key element is an inter-frame compression method that can seamlessly build up on different existing neural image autoencoders. It consists of two stages; interpolation and residual compression. First, the interpolation problem is framed in the context of video compression and we combine motion compression and image synthesis in a single network. Secondly, we express the residual information between the original frame and the interpolated frame in latent space directly. Our objective is to use the same autoencoder for images (key-frames) and residuals (interpolated frames), with the constraint of learning a representation that performs well for both. Our contributions can be summarized as follows:

- The interpolation model combines motion compression and image synthesis while reducing computation at decoding time. This joint approach offers the possibility to reduce the motion code size (for example when motion does not help to produce a good interpolated image).
- We show how the same neural autoencoder can be used for images and residuals. In addition to reducing the number of parameters, it has the advantage of automatically achieving the same image quality for both outputs. There is no need for separately tuning the residual quality to match the key-frames.
- Our extensive evaluation on different datasets and resolutions shows the benefits of our approach, which achieves competitive results with existing state-of-the art video codecs. This encompasses a comparative study with different interpolation approaches, demonstrating the benefits of our method.

2. Related Work

As our approach to video coding compresses per frame data with an auto-encoder, we regard both image and video compression approaches as related works and structure this section accordingly. Our proposed codec is lossy and therefore we will only focus on lossy compression methods.

Image Compression. One of the oldest but widely used lossy images codecs is JPEG [29]. It partitions the image into smaller patches and encodes the data using a discrete cosine transform. The resulting coefficients are then scaled, quantized, and entropy coded to form the final bitstream. In newer formats different directions have been explored such as using other transforms - wavelets in JEPG2000 [26] - or intra prediction, and in-loop filtering derived from video codecs in BPG [6] and Webp [10].

Neural Image Compression. Recently, there has been significant work on applying deep learning to image compression [4, 15, 17, 22, 28]. Instead of hand crafting the individual components of the codec, these models can learn an optimal non-linear transform from data along with the probabilities required for entropy coding the latent representation into a bitstream in an end-to-end fashion. While the first methods [4, 27, 28] showed improved results over JPEG or JPEG2000, subsequent methods [5, 15, 17] are now on par or surpassing BPG [6]. The best performing methods [5, 17] refine the prior model for entropy coding by transmitting side information and applying an autoregressive model.

Video Compression. In the 1960s, video compression research started with codecs that compress each frame individually [3] while todays codecs all leverage the temporal redundancy of video data by using motion estimation and motion compensation for inter frame prediction. They also rely on a hand-crafted block based hybrid structure [19] combining inter with intra frame predictions. Currently, H.264 (AVC) is the most commonly used standard [31], but in the near future more recent codecs such as H.265 (HEVC) [24], VP9 [18], and AV1 [8] will replace it.

Neural Video Compression. After recent successes in image compression and the promising advances in frame interpolation [13, 16, 20, 21, 33] and optical flow estimation [12, 25], some recent works proposed using neural networks for video compression. To the best of our knowledge, Wu *et al.* [32] proposed the only deep learning approach for interpolation based video compression. Their work focuses on encoding residuals after warping the reference frames and their context. It offers interesting insights on the interplay between interpolation and compression. The solution is however computationally expensive as several iterations are required to reach higher quality levels and multiple models are needed for different interpolation intervals. In a very different approach to the problem, Han *et al.* [11], keep all predictive spatial and temporal modeling in the latent space. But this method is limited to low resolution videos. More recently, the low latency setting was explored [14, 23] where only preceding reference frames are used for inferring temporal information. Rippel *et al.* [23] maintain a latent state learned by the model instead of using frames from the past. Their model does not however address the interpolation case. Lu *et al.* [14] replace traditional video codec blocks with neural network components, but contrary to our join strategy, they sequentially compute and compress optical flow, estimate the compensation and finally compress image residuals. This requires a dedicated networks for motion compensation and residuals which is unnecessary in our case.

3. Neural Inter-Frame Compression

The main objective of *lossy* image and video compression is to find an encoding of the input frames satisfying the competing constraints that it should occupy as little storage as possible, while on the other hand, the reconstruction

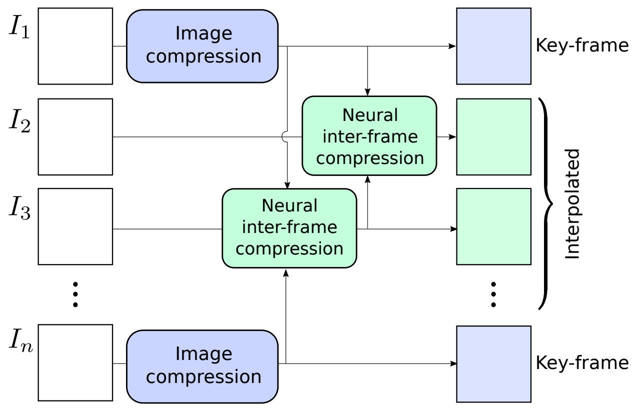

Figure 2: **Overview.** To encode a sequence of n frames, the first and last frames are encoded as images (called key-frames). The rest of the frames in the interval are encoded recursively using our *neural inter-frame compression* block.

should have as little distortion as possible. In particular, video compression takes advantage of temporal redundancy in the image sequence. In our interpolation based compression setting (see Fig. 2), videos are divided into segments of lengths n. First and last frames of the segment are encoded and decoded as independent images (*key-frames*), while motion information is used to generate the intermediate images from these already decoded frames through interpolation. We also define the notion of *reference frames*, which are the images used by the neural inter-frame compression block. These include the key-frames but also any interpolated frame that is itself used as input for another inter-frame block as illustrated in Figure 2.

In this work we propose an interpolation based compression algorithm based on neural networks, that is compatible with image compression architectures, to build an efficient video compression pipeline. The solution we propose consists of two subtasks; interpolation (Sec. 3.1) and residual compression (Sec. 3.2). We first address interpolation by introducing a single network which combines motion encoding and image synthesis. In the second step, we correct for distortions by encoding latent residuals obtained from a neural image compression network. As neural image compression is a key building block, we first review some of its elements.

Neural image compression. A mapping from image to latent space is realized with a neural encoder-decoder pair, where the bottleneck values constitute the latent representation. We denote g_ϕ the function mapping from image space to latent space and $g_{\phi'}$ the reverse mapping. The learned parameters are ϕ and ϕ'. An image x is first mapped to its latent representation $y = g_\phi\ (x)$. After quantization, the resulting latents $\hat{y}$ are coded losslessly to a bitstream that can be decoded into the image $\hat{x} = g_{\phi'}(\hat{y})$. Image compression can formally be expressed as minimizing the expected length of the bitstream as well as the expected distortion of the reconstructed image compared to the original, formulated as optimizing the following rate-distortion objective functional:

$$L(g_\phi, g_{\phi'}, p_{\hat{y}}) = \mathbb{E}_{x\sim p_x}[\underbrace{-\log_2 p_{\hat{y}}(\hat{y})}_{\text{rate}} + \lambda \underbrace{d(x,\hat{x})}_{\text{distortion}}] , \quad (1)$$

where $d(x, \hat{x})$ is the distortion measure, e.g. mean squared error. The rate corresponds to the length of the bitstream needed to encode the quantized representation $\hat{y}$, based on a learned entropy model $p_{\hat{y}}$ over the unknown distribution of natural images p_x. By reducing the weight λ, better compression can be achieved at the cost of larger distortion on the reconstructed image.

By design, our framework is compatible with any neural autoencoder. In this work we use the encoder-decoder pair proposed by Ballé *et al.* [4]. Several possibilities [4, 5, 15] exist to learn the entropy model $p_{\hat{y}}$. We follow [17] and use neural networks to predict the probabilities for latent space values, based on side information additionally sent. The distributions of latent space values are modeled as Gaussians, and a hyper-parameter and a context model networks are used to predict the probabilities.

3.1. Interpolation with Compression Constraints

To take advantage of temporal redundancy in video coding, our solution relies on information transfer through motion compensation. More precisely, an intermediate frame x can be predicted from its set of reference images $\mathcal{K}_x = \{x_1, \ldots, x_k\}$ by considering motion information. Contrary to the standard setup of frame interpolation, the original frame x is available during encoding. Our solution takes advantage of this by internally estimating displacement maps f_i w.r.t. the ground truth frame x, then computing a quantized latent representation $\hat{q}$, that can be directly decoded into displacement maps $\hat{f}_i$ and blending coefficients $\hat{\alpha}_i$. If we denote w the warping function that transforms the reference x_i according to motion, the interpolation result is

$$x_{\text{intrp}} = \sum_{i=1}^{k} \hat{\alpha}_i w(x_i, \hat{f}_i) \text{ with } \sum_{i=1}^{k} \hat{\alpha}_i = 1. \quad (2)$$

We propose an encoder-decoder pair $(h_\rho, h_{\rho'})$ to solve this interpolation problem. In our method we use two reference frames ($n = 2$) and Figure 3 shows the corresponding model. The encoder h_ρ maps the input data into a latent representation q :

$$q = h_\rho(x, x_1, x_2, f_1, f_2) , \quad (3)$$

which is quantized into $\hat{q}$.

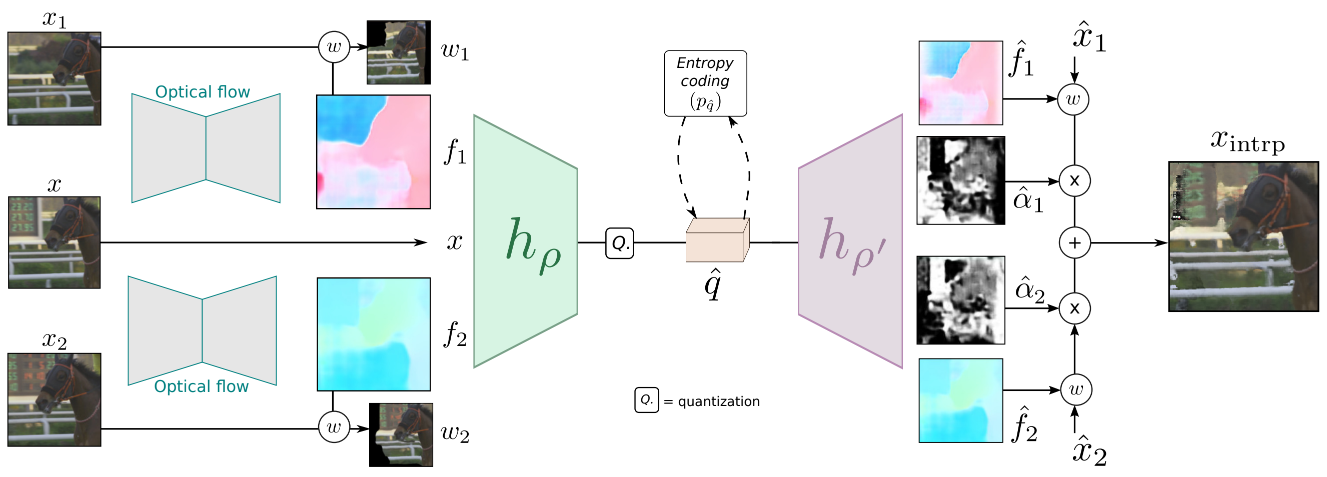

Figure 3: **Interpolation with compression constraints.** We propose combining the two tasks of interpolation and optical flow compression. Given two reference frames x_1 and x_2, we first use an optical flow network to compute the 2d displacement maps f_1 and f_2. An important difference with classic frame interpolation, is the availability of the ground truth image x at encoding time. We take advantage of this by providing as input to the encoder h_ρ the original frame x, the optical flow fields f_1 and f_2, and the the warped frames $w_1 = w(x_1, f_1)$ and $w_2 = w(x_2, f_2)$. The resulting representation is quantized ($\hat{q}$) before entropy coding. The decoder $h_{\rho'}$ can directly synthesize the displacement maps ($\hat{f}_1$, $\hat{f}_2$) and the blending coefficients ($\hat{\alpha}_1$, $\hat{\alpha}_2$) to compute the intermediate frame x_{intrp} of the decoded reference frames $\hat{x}_1$ and $\hat{x}_2$.

From $\hat{q}$, the decoder $h_{\rho'}$ can both reconstruct optical flow fields and blending coefficients:

$$(\hat{\alpha}_1, \hat{\alpha}_2, \hat{f}_1, \hat{f}_2) = h_{\rho'}(\hat{q}). \quad (4)$$

and x_{intrp} is directly computed according to Equation (2).

Similarly to image compression, the proposed latent representation should respect compression objectives, that is to satisfy the competing constraints of occupying as little storage as possible, while minimizing distortion on the interpolation result. Formally this can be expressed as optimizing the following rate-distortion problem:

$$L(\Theta_{\text{intrp}}) = \mathbb{E}_{x\sim p_x}[-\log_2 p_{\hat{q}}(\hat{q}) + \lambda_{\text{intrp}}\, d(x, x_{\text{intrp}})] \ , \quad (5)$$

where $\Theta_{\text{intrp}} = \{\rho, \rho', p_{\hat{q}}\}$ consists of the encoder-decoder network parameters (ρ, ρ') and the entropy model $(p_{\hat{q}})$.

A first advantage of our approach comes from providing warping results together with the original frame x during compression, such that a better prediction for the blending coefficients can be made. Furthermore we only penalize distortion on the reconstructed intermediate frame and not on the reconstructed motion field itself. This enables the network to identify flow vector importance and to infer where a faithful motion reconstruction is unnecessary or less important w.r.t to the final result. Another advantage of our approach is the reduced computation time as complex frame interpolation is avoided on the decoding side; The network $h_{\rho'}$ does not have any more complexity than the image decoder $g_{\phi'}$, and its output is directly used to synthesize the interpolation result.

3.2. Latent Space Residuals

The previous section described how we obtain an estimate x_{intrp} of the original image x based on motion compensation. However, the interpolation result can still contain noticeable errors, which can be reduced by additionally transmitting residual information between x_{intrp} and x. Instead of designing a distinct network for image space residuals, we propose to take advantage of the compression encoder-decoder pair $(g_\phi, g_{\phi'})$ used for the key-frames and to compute residual information in latent space. Figure 4 illustrates latent residual estimation when interpolating from two key-frames x_1 and x_2. The key-frames are encoded with an image compression strategy, then used as reference frames for interpolation, as described in Section 3.1. The remaining residual information between x and x_{intrp} is represented by computing the residual in latent space between y and y_{intrp}:

$$r = y - y_{\text{intrp}} = g_\phi(x) - g_\phi(x_{\text{intrp}}) \ . \quad (6)$$

The residual is quantized and the final decoded image $\hat{x}$ can be computed as:

$$\hat{x} = g_{\phi'}(y_{\text{intrp}} + \hat{r}) \ . \quad (7)$$

Estimating the residual in latent space allows to use the same encoder and decoder for both the key-frames and the

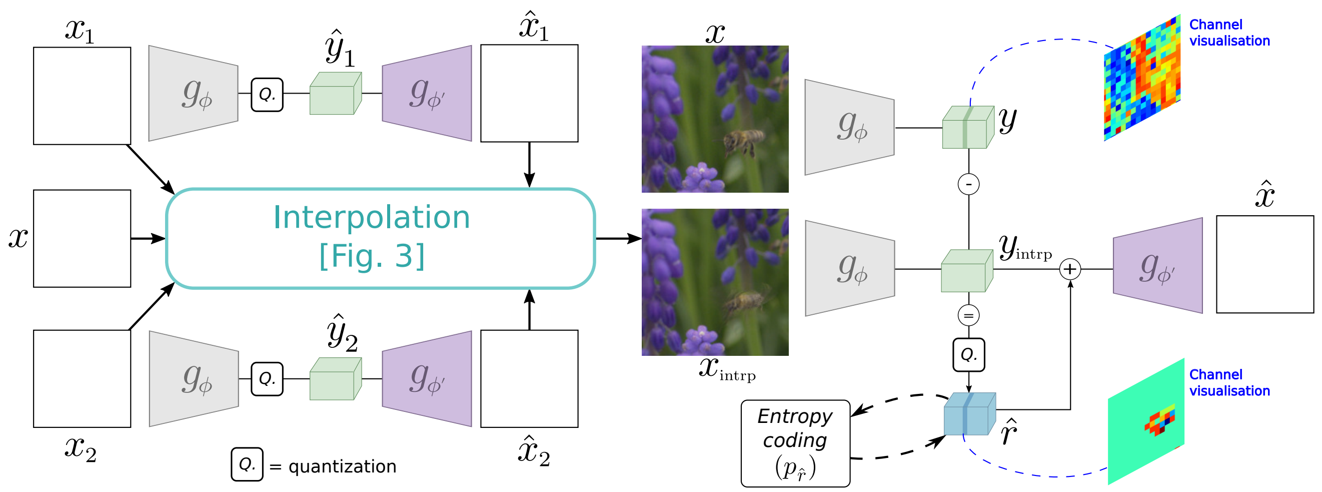

Figure 4: **Latent space residual.** When compressing a video segment, the key-frames x_1 and x_2 are encoded first with an image compression autoencoder. The decoded frames $\hat{x}_1$ and $\hat{x}_2$ are then used to estimate the interpolation result x_{intrp}. The same image encoder g_ϕ is used to compute the latents y and y_{intrp} and only the residual $\hat{r}$ has to be transfered (visualization for one channel is provided). With residuals in latent space, we only need to additionally estimate a probability model for $\hat{r}$.

residuals which reduces the number of parameters. This also offers the advantage of achieving the same reconstruction quality both for key-frames ($\hat{x}_i$) and predicted frames $\hat{x}$ by design.

As the image compression network is also used for the residuals, some adaptations are required. The loss function described in Equation (1) is limited to image compression. We extend this objective function to take into account the residuals when training the encoder-decoder pair $(g_\phi, g_{\phi'})$. Furthermore, in addition to the model $p_{\hat{y}}$, we need to build a probability model $p_{\hat{r}}$ for entropy coding residual values. These modifications amount to optimizing jointly the rate-distortion objective functional for key-frames and interpolation residual together:

$$L(\Theta_{\text{img}}) = \mathbb{E}_{x\sim p_x}\Bigg[\underbrace{-\log_2 p_{\hat{r}}(\hat{r}) + \lambda_{\text{img}}\, d(x,\hat{x})}_{\text{residual}} + \sum_{i=1}^{2}\frac{1}{2}\underbrace{\big(-\log_2 p_{\hat{y}}(\hat{y}_i) + \lambda_{\text{img}}\, d(x_i,\hat{x}_i)\big)}_{\text{key-frame}}\Bigg], \tag{8}$$

where $\Theta_{\text{img}} = \{\phi, \phi', p_{\hat{y}}, p_{\hat{r}}\}$ are the learned parameters.

3.3. Balancing Side Information and Residuals

In the previous sections we have described all the constituting elements of the video compression framework illustrated in Figure 2. If we consider a fixed target quality for the video, the ideal inter-frame compression should achieve the lowest bit-rate for interpolated frames, while having a perceptually similar quality as the key-frames. Since the image compression network for the key-frames is also used for the residuals in our inter-frame compression block, the final interpolated image quality is similar. This quality is determined by training the image compression network for a particular value of λ_{img} in the loss function (Eq. 8). As a result, the only remaining degree of freedom comes from selecting λ_{intrp} in Equation (5) for the interpolation autoencoder $(h_\rho, h_{\rho'})$. For example, by choosing a lower quality for the interpolation the bit-rate is largely reduced but this could negatively impact the bit-rate needed for the residual. On the opposite end, choosing a higher quality for the interpolation may not be ideal either, as large motion can be present in which case interpolation is difficult and allocating a larger fraction of bits to encode the residuals becomes a better choice overall.

In the proposed framework, the best compromise can be found by comparing several interpolation configurations at test time and using the one resulting in the lowest total bit-rate after residual computation. To be able to freely combine the different version of the two networks, we train them independently. The interpolation part is trained first for different λ_{intrp} values in the loss function (Eq. 5) and only using ground truth images. The image compression network is then trained in a second step, using one of the obtained interpolation models, keeping its parameters fixed.

3.4. Network Architectures

We use the encoder-decoder pair proposed by Ballé *et al.* [4] for both the image and the interpolation networks and we develop next some of the details. The encoders g_ϕ and g_ρ have 5 blocks each consisting of a convolutional and

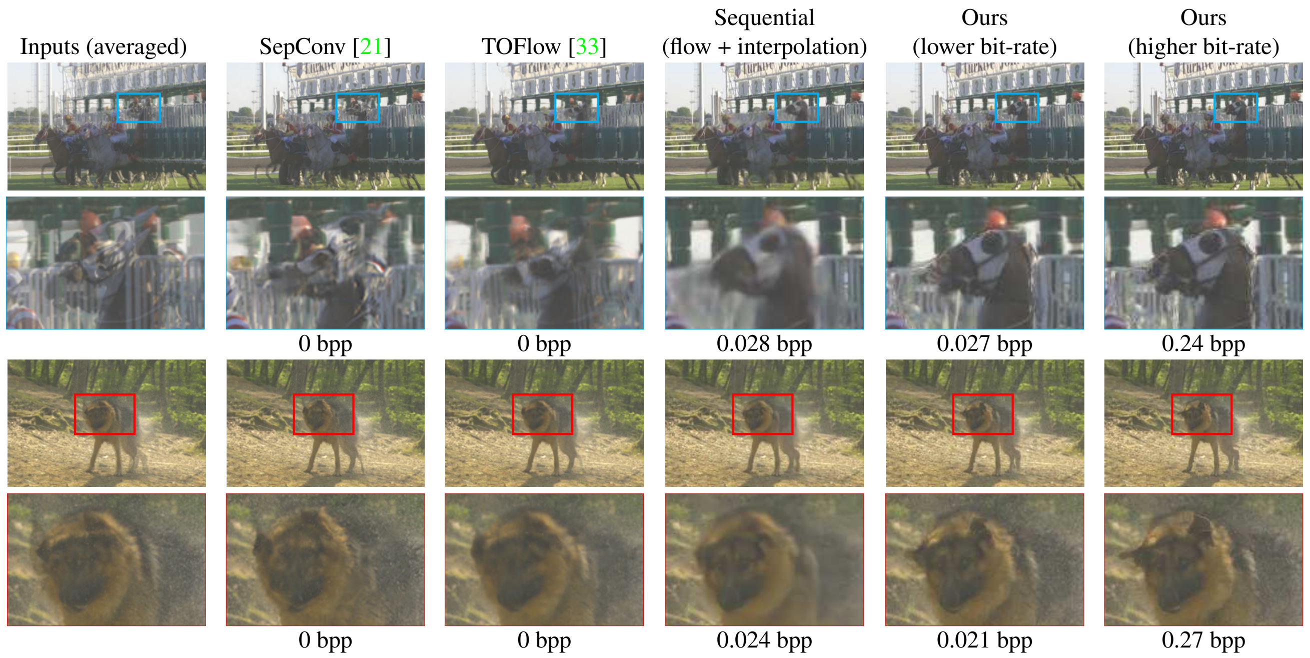

Figure 5: **Interpolation results.** Using compression constraints for frame interpolation, we are able to achieve good interpolation results even at low bit-rates. The left-most column shows the averaged input frames. Frame interpolation techniques [21, 33] do not anticipate to transfer motion data. In the sequential approach, the optical flow is first decompressed, then the interpolation is estimated. All these methods require more computation power at decoding time and perform worse than our joint strategy.

a Generalized Normalization Transformation (GDN) layer. The convolutional layers have a kernel size $k = 5$ and a stride $s = 2$. The decoders $g_{\phi'}$ and $g_{\rho'}$ also share the same architecture of 5 blocks each with upsampled convolutions ($k = 5$ and $s = 2$) and an inverse GDN layer. The image decoders $g_{\phi'}$ final number of output channels is 3, corresponding to an RGB image. The decoder $g_{\rho'}$ used for the interpolation part, has 5 output channels. Four channels correspond to the two motion fields $\hat{f}_1$ and $\hat{f}_2$, while a sigmoid is applied to the 5-th channel to obtain the mixing coefficient $\hat{\alpha}_1$, implying $\hat{\alpha}_2 = 1 - \hat{\alpha}_1$, respectively. For the optical flow estimation we use the pretrained PWC-Net [25] and keep its weights fixed during all our trainings.

To approximate the quantization operation performed in the bottleneck, we add an independent uniform noise to the latent space values during training. This was shown to be a continuous differentiable relaxation of the rounding operation [4].

Training Procedure. We use all the frames from the Vimeo-90K septuplets dataset [33] for training. Video compression is applied on short video segments and requires interpolation from a different range of intervals. To simulate this during training, we randomly sample triplet of images with intervals of $1, 2$ or 3 frames. We use the mean squared error (MSE) as image distortion loss d (Eq. 5,8). We achieve different rate-distortion ratios by training with different weights ($\lambda_{\text{intrp}}, \lambda_{\text{img}}$). For entropy coding, we used the probability model proposed by Minnen *et al.* [17] to model image latents $p_{\hat{y}}$, latent residual values $p_{\hat{r}}$ and motion information $p_{\hat{q}}$. During test time, we reach different rate-distortion points by varying the quantization step size used in latent representation [9].

4. Experimental Results

We present here a detailed evaluation of the proposed model. We analyze the advantages of the individual parts of our framework, namely the compression constrained frame interpolation as well as the latent space residual followed by a comparative study of the full model with respect to standard video compression codecs H.264 [31] and H.265[24]. To measure the distortion we use the Peak Signal to Noise Ratio (PSNR). For our experiments we use three different datasets. The first dataset is the Video Trace Library [2] further referenced as VTL. We only used the highest resolution clips (352×288) in our experiments and setting the maximum length to 300 frames for all clips. Besides VTL we also used raw videos from the Ultra Video Group [1] and the MCL-JVC dataset [30] which all have a resolution of 1920×1080 with a large variety of content and motion.

Advantages of the proposed interpolation. In order to evaluate the benefits of the proposed compression constrained interpolation method we use the UVG dataset and the following experiment setup: Given a frame to interpolate $x = I_i$, we use the frames $x_1 = I_{i-k}$ and $x_2 = I_{i+k}$ as reference frames. For this experiment we use the original frames x_i as reference frames and do not apply any image compression. Our method follows the diagram of Figure 3. For comparison, we implement a sequential approach where optical flow is first separately compressed and decompressed followed by a interpolation method on the accordingly warped reference frames (see supplementary material for details). In addition to this, we also test two frame interpolation methods; TOFlow [33] and SepConv [21], using the implementations available online, where no additional motion information needs to be encoded. Figure 5 shows examples of the resulting images. The first example corresponds to an interval of 12 frames between the reference frame whereas the second one corresponds to 6 frames. Our method is able to noticeably outperform frame interpolation techniques even at low bit-rates, while using more bits also keeps improving the results (last column).

A quantitative evaluation on the UVG dataset (Fig. 6) with frame intervals of 12 frames shows the evolution of image quality with respect to bit-rate. We evaluate our interpolation method using the *simple* factorized model [4] as well as the *full* model [17]. For reference, frame interpolation techniques are plotted with horizontal lines as they do not require any data to be transmitted. In this case the interpolated image is directly predicted from the available reference frames. However, computation cost is higher and the task is more challenging, especially for large motion that is nonlinear. The sequential approach benefits from decoded motion fields but improvement is limited when increasing bit-rate. Our method, on the other hand has access to the ground truth image during encoding and can leverage the additional data more efficiently.

Analysis of latent space residuals. To better understand latent space residuals, Figure 7 illustrates the obtained values and probabilities. The first row corresponds to the key-frame compression case. The input image x is mapped to a latent representation y that is quantized into $\hat{y}$. The middle column shows the result for one of the bottleneck channels. Latent space values are represented by a temperature whereas probabilities are represented by gray-scale values. The second row corresponds to the result x_{intrp} obtained by interpolation and a deviation from the ground truth is visible which appears in the quantized latent space residual $\hat{r}$ as well. Because the range of values of $\hat{r}$ is smaller than of $\hat{y}$, we can achieve a much lower entropy, allowing more efficient encoding.

Video codec comparisons. To show the advantage of the proposed video compression framework, we compare with existing video codecs, in particular H.264 [31] and H.265 [24]. Our key-frames are positioned every 12 frames and interpolation is performed recursively inbetween them. This is also reflected in the used video codecs configuration for comparison. We use *ffmpeg* tools to compress the videos and the exact command lines are provided in the supplemental material. We consider two setups for the video codecs: *fast zero latency* and *medium*. In addition to standard video codecs, we compare with two neural video compression methods from state of the art [32, 14].

Our evaluation (Fig. 8) using the *full* model shows competitive results with existing codecs, especially on the high resolution, high quality datasets UVG and MCL-JVC. Figure 1 shows an example of the image quality achieved by

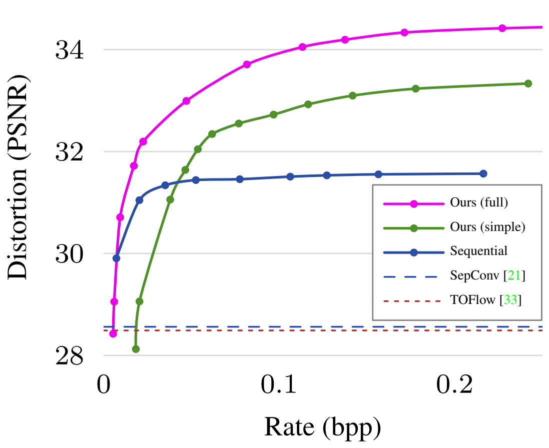

Figure 6: **Quantitative evaluation for interpolation.** Rate-distortion curves for our interpolation model using the factorized (green) and full model (pink). We also evaluate the sequential approach decompressing the flow before interpolation. Horizontal lines correspond to frame interpolation techniques, which do not require any encoded data.

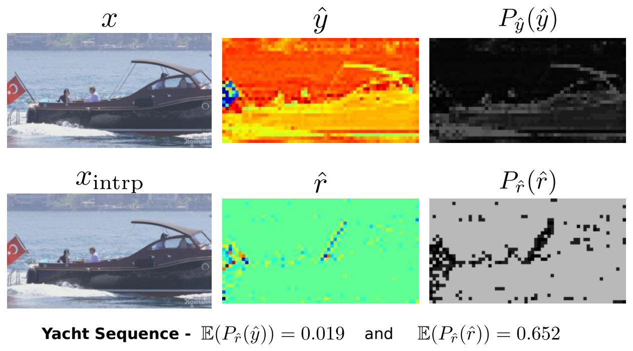
x
$\hat{y}$
$P_{\hat{y}}(\hat{y})$
x_{intrp}
$\hat{r}$
$P_{\hat{r}}(\hat{r})$
Yacht Sequence - $\mathbb{E}(P_{\hat{r}}(\hat{y})) = 0.019$ and $\mathbb{E}(P_{\hat{r}}(\hat{r})) = 0.652$

Figure 7: **Analysis of latent space residuals.** Using image compression, the image x is mapped to its quantized latent representation $\hat{y}$ with associated probabilities $P_{\hat{y}}(\hat{y})$ (illustrated for one channel). In the second row, the interpolation result x_{intrp} has some errors that appear in latent space residual $\hat{r}$. The residual will mostly have values centered around 0 which results in higher probabilities (i.e. low entropy).

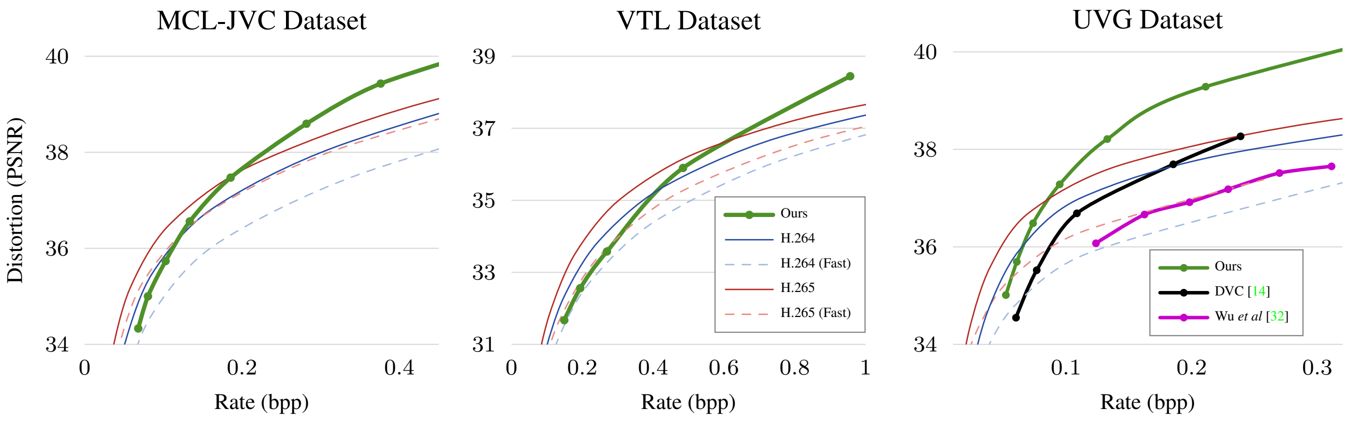

Figure 8: **Final evaluation.** Compression results on three different video datasets using a key-frame interval of 12 frames. We compare with H.264 [31] and H.265 [24] in two different configurations.

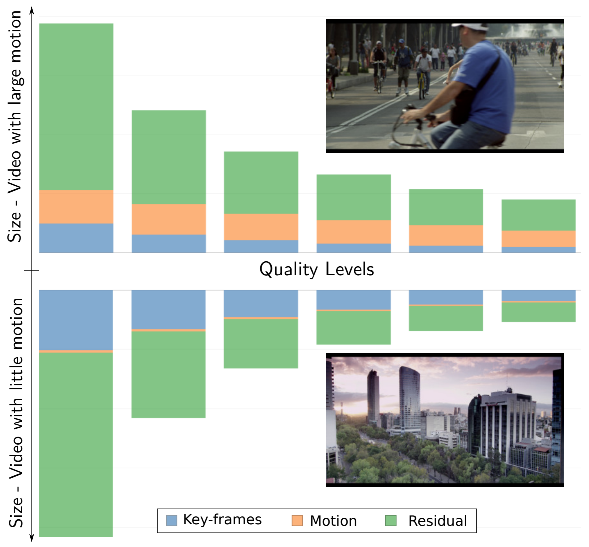

Figure 9: **Bit-rate distribution.** The bottom video is a time-lapse clip, where very few bits are allocated to motion. For the top video, motion is very important and can represent close to half the data for the lower quality levels. As quality increases, residual data becomes more important.

our approach. Compared to existing video codecs, less visual artifacts are present and colors are closer to the original. As the VTL dataset includes many noisy low resolution videos which are different from the training set, our solution does not perform as well on the lower bit-rates, however, still on par with existing codecs.

Comparisons with existing neural video compression methods is provided for the UVG dataset. Our approach outperforms both DVC [14] and interpolation based video compression [32]. In addition to this, our approach has the advantage of using only two networks for compression while DVC [14] has 4 different networks. We also coherently process all frame intervals and motion amplitudes with a single autoencoder, while [32] has models to handle different interpolation intervals.

Bit-rate distribution. We investigate the bits repartition between the different types of data. For each quality level, we compute the repartition of bits in terms of key-frames, residual and motion. Figure 9 illustrates this for two video clips with very different types of content in terms of motion. The top video contains many moving elements whereas the second is a time-lapse with very little motion. This is reflected in the number of bits allocated to motion compensation. It is also interesting to note the relative importance of motion and residual data for different quality levels. In the lowest levels, motion compensation is very efficient and so allocating a larger portion of the bit-rate to it is beneficial. For higher quality levels, it is better to increase the proportion of residual data. This is in line with the results in Figure 5 where quality difference between lower and higher bit-rate results is marginal compared to size difference.

5. Conclusion

Our neural video coding framework is able to achieve results competitive with existing video codecs that have witnessed several decades of engineering improvements. This is in particular due to the interpolation approach that embeds compression constraints and takes advantage of all the available information at encoding time. In addition to this, expressing residuals in latent space simplifies the video compression task as the same network is used both for key-frames and residuals. In this work, we have focused on compressing intermediate frames that rely on key frames in the past and future. However, our approach is also compatible with other settings such as only frames from the past. Thus finding optimal strategies for key frame selection would be an interesting branch of future work.

References

[1] Ultra video group test sequences. http://ultravideo.cs.tut.fi. Accessed: 2019-03-22. 6

[2] Video trace library. http://trace.kom.aau.dk/yuv/index.html. Accessed: 2019-03-22. 6

[3] HC Andrews and WK Pratt. Television bandwidth reduction by encoding spatial frequencies. *Journal of the SMPTE*, 1968. 2

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *ICLR*, 2017. 1, 2, 3, 5, 6, 7

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *ICLR*, 2018. 2, 3

[6] Fabrice Bellard. Bpg specification version 0.9.5, 2014. 2

[7] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 2018. 1

[8] Peter de Rivaz and Jack Haughton. Av1 bitstream & decoding process specification. *The Alliance for Open Media*, page 182, 2018. 2

[9] Thierry Dumas, Aline Roumy, and Christine Guillemot. Autoencoder based image compression: can the learning be quantization independent? In *ICASSP*, 2018. 6

[10] Google. Webp, 2010. 1, 2

[11] Jun Han, Salvator Lombardo, Christopher Schroers, and Stephan Mandt. Deep probabilistic video compression. *arXiv*, 2018. 1, 2

[12] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018. 2

[13] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik G. Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018. 2

[14] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *CVPR*, 2019. 1, 2, 7, 8

[15] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *CVPR*, 2018. 1, 2, 3

[16] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *CVPR*, 2018. 2

[17] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*. 2018. 1, 2, 3, 6, 7

[18] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, and R. Bultje. The latest open-source video codec vp9 - an overview and preliminary results. In *Picture Coding Symposium (PCS)*, 2013. 2

[19] Hans Georg Musmann, Peter Pirsch, and H-J Grallert. Advances in picture coding. *Proceedings of the IEEE*, 1985. 2

[20] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, 2018. 2

[21] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2, 6, 7

[22] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *ICML*, 2017. 2

[23] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G Anderson, and Lubomir Bourdev. Learned video compression. *arXiv*, 2018. 1, 2

[24] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al. Overview of the high efficiency video coding(hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 2012. 1, 2, 6, 7, 8

[25] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2, 6

[26] David S Taubman and Michael W Marcellin. Jpeg2000: Standard for interactive imaging. *Proceedings of the IEEE*, 2002. 2

[27] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *ICLR*, 2016. 2

[28] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *CVPR*, 2017. 2

[29] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 1992. 2

[30] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset. In *ICIP*. IEEE, 2016. 6

[31] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 2003. 1, 2, 6, 7, 8

[32] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. *ECCV*, 2018. 1, 2, 7, 8

[33] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *arXiv*, 2017. 2, 6, 7