Content Adaptive Optimization for Neural Image Compression

Joaquim Campos Meierhans Simon Abdelaziz Djelouah Christopher Schroers
DisneyResearch|Studios
abdelaziz.djelouah@disney.com christopher.schroers@disney.com

Abstract

The field of neural image compression has witnessed exciting progress as recently proposed architectures already surpass the established transform coding based approaches. While, so far, research has mainly focused on architecture and model improvements, in this work we explore content adaptive optimization. To this end, we introduce an iterative procedure which adapts the latent representation to the specific content we wish to compress while keeping the parameters of the network and the predictive model fixed. Our experiments show that this allows for an overall increase in rate-distortion performance, independently of the specific architecture used. Furthermore, we also evaluate this strategy in the context of adapting a pre-trained network to other content that is different in visual appearance or resolution. Here, our experiments show that our adaptation strategy can largely close the gap as compared to models specifically trained for the given content while having the benefit that no additional data in the form of model parameter updates has to be transmitted.

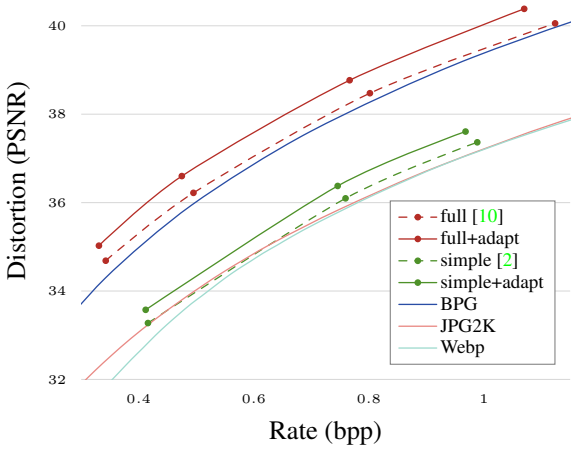

Figure 1: **Latent Adaptation.** The evaluation on the Tecnick dataset [1] shows that per image latent representation adaptation is complementary to existing neural image compression methods. It allows to improve rate-distortion performance while keeping the neural network and the computing time on the decoder side unchanged.

1. Introduction

The share of video content in today's internet traffic is colossal and will only increase in the foreseeable future [5]. Since image compression is at the core of all video coding approaches, improvements concerning image data are expected to have a significant impact on video as well. In the recent years, several neural network-based approaches for image compression [2, 9, 10, 12, 13] have been developed and rapidly caught up with several decades of work in transform coding. They are already able to outperform traditional image compression codecs, which rely on handcrafting the individual components. Instead, these methods can be trained end-to-end and leverage large amounts of data to learn an optimal non-linear transform, along with the probabilities required for entropy coding the latent representation into a compact bit stream.

While previous work has mainly focused on more efficient architectures and predictive models, in this work, we adopt a different approach by optimizing the latent representation *individually*, on a per-image basis, during the encoding process. Thanks to this per-image adaptation, our refined representation is more efficient in terms of rate-distortion performance compared to the latent representation obtained with a simple forward pass through the autoencoder. The method is general and, as such, can be applied to improve a number of different architectures for learned image compression. The key benefit of the proposed solution lies in the ability to achieve an improved compression performance while the neural compression network and the predictive model are kept fixed and the computing time on the decoder side remains unchanged. We demonstrate the general applicability of the adaptation scheme by providing results on two different image compression architectures (Fig. 1). The second contribution is a detailed evaluation on the use of the proposed adaptation scheme for adapting a given pre-trained model to other content that is different in visual appearance or resolution. Our

evaluation includes comparisons to strategies that update only the predictive model or both the network parameters and the predictive model. Experiments show the advantages of the latent space adaptation, as this largely allows to close the gap with the models specifically trained on the new content and does not require to transmit any updated model parameters.

2. Preliminaries and Related Work

The objective of *lossy* image compression is to find a mapping or encoding function $\psi : \mathcal{X} \rightarrow \mathcal{Y}$ from the image space $\mathcal{X}$ to a latent space representation $\mathcal{Y}$ and its reverse mapping or decoding function $\phi : \mathcal{Y} \rightarrow \mathcal{X}$ back to the original image space, with the competing constraints that, on the one hand, the latent representation should occupy as little storage as possible while, on the other hand, the reconstructed image should closely resemble the original image.

In neural image compression, this mapping is realized with a neural encoder-decoder pair, where the bottleneck values constitute the latent representation. An image x is first mapped to its latent representation $y = \psi(x)$. After quantization, the resulting latents $\hat{y}$ are coded losslessly to a bit stream that can be decoded into the image $\hat{x} = \phi(\hat{y})$.

Image compression can be formally expressed as the minimization of both the expected length of the bitstream, as well as the expected distortion of the reconstructed image compared to the original, which leads to the optimization of the following rate-distortion trade-off:

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

Here, $d(x, \hat{x})$ 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. The weight λ steers the rate distortion trade-off, e.g. reducing λ leads to a higher compression rate at the cost of a larger distortion of the reconstructed image.

In order to achieve good compression results that can deal with a vast distribution of images, two main problems arise: First, finding a powerful encoder/decoder transformation and second, properly modeling the distribution in the latent space.

Existing works such as [2, 12] have made contributions to the first problem by proposing neural network architectures to parameterize the encoding and decoding functions; more recently, the main focus of the research community has been on the second problem which not only allows to capture remaining redundancy in the latent representation for efficient entropy coding but also regularizes the encoder [3, 9, 10].

3. Content Adaptive Compression

In existing approaches, equation 1 is optimized over a corpus of potentially millions of images in order to find optimal functions for encoding and decoding (ϕ and ψ), along with a suitable probability model $p_{\hat{y}}$ for the latent space. Although the network has been trained over a large corpus of images to find what should ideally be an optimal encoding function ψ over the whole data set, the encoding can still be improved by adapting to each single image. In our work, we perform this per-image adaptation, without changing the encoder/decoder or the parameters of the latent space probability model, but by changing the latent values themselves at test time. As such, we are effectively trying to solve the following optimization problem, during test time, for a single image x:

$$\arg\min_{\hat{y}} \; -\log_2 p_{\hat{y}}(\hat{y}) + \lambda \, d(x, \hat{x}) \,. \quad (2)$$

The fact that we do not change the decoder and the probability model in the optimization respects the assumption that both have been trained and deployed at the receiver. Therefore, the ideal strategy at this point is to find the best discrete latent representation by varying only the latent values themselves.

There are several options to practically solve this problem, including both discrete and continuous optimization approaches. In this work, we solve it through an iterative procedure, similar as during training, where gradient descent is applied on the latents according to

$$y_{t+1} = y_t - \eta \nabla_y L(\psi, \phi, p_{\hat{y}}, x). \quad (3)$$

Here $L(\psi, \phi, p_{\hat{y}}, x)$ is the rate-distortion objective for a particular image x:

$$L(\psi, \phi, p_{\hat{y}}, x) = \log_2 p_{\hat{y}}(\hat{y}) + \lambda \, d(x, \hat{x}), \quad (4)$$

and η is the weighting applied on the gradient. This requires a differentiable approximation of the quantization operation performed in the bottleneck and we use additive uniform noise for this purpose [2]. Adopting the notation $\mathcal{U}$ for an independent uniform noise of width 1, the density function $p_{\tilde{y}}$ of the random variable $\tilde{y} = y + \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ becomes a continuous differentiable relaxation of the probability mass function $p_{\hat{y}}$.

The final image compression pipeline is described by Algorithm 1. The lossless arithmetic encoding/decoding operations are represented by AE/AD. The step function corresponds to updating the latent representation according to the gradient step obtained from the Adam [7] optimizer with a learning rate of $1e^{-3}$. In all our experiments there were no noticeable improvements after 1500 update steps and we used this as maximum number of iterations in Algorithm 1. On a Titan Xp GPU with 12Gb of memory, optimizing the latents for an HD image requires approximately 5min.

Algorithm 1 Compression with Per Image Adaptation

procedure REFINELATENTS(y, x)
loop for MAX steps:
  $\tilde{y} := y + \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$
  $\tilde{x} := \phi(\tilde{y})$
  $L(\tilde{y}) := \sum_i -\log_2 p_{\tilde{Y}_i}(\tilde{y}_i) + \lambda d(x, \tilde{x})$
  $y := y + \text{step}(L(\psi, \phi, p_{\hat{y}}, x))$
return :
  y
procedure ENCODE(x)
  $y := \psi(x)$
  $y := \text{RefineLatents}(y, x)$
  $\hat{y} := \text{quantize}(y)$
  $b := \text{AE}(\hat{y})$
return :
  b
procedure DECODE(b)
  $\hat{y} = AD(b)$
  $\hat{x} = \phi(\hat{y})$

4. Experimental Results

In order to show the benefits of the proposed latent adaptation strategy, we consider two experimental setups: First, we explore the applicability of the proposed approach on different image compression architectures. Second, we evaluate how our per-image adaptation compares to other forms of content adaptation.

4.1. Latent adaptation on different architectures

The image-adaptive optimization is independent of the particular neural compression algorithm used. To demonstrate this, we use two existing architectures; A (*simple*) model using a single latent space and a factorized probability model [2], and a more complex model (*full*) with a hierarchical latent representation in which hyperpriors and context are used for modeling the probabilities of the latents [10]. Both models were trained on images from the COCO segmentation dataset [8]. After training, the models were tested on the Tecnick dataset [1]. Figure 1 showcases the rate distortion performance averaged over the test set together with JPEG2000 [11], WEBP [6] and BPG [4] rate-distortion curves. As the rate-distortion curves show, in each case, the per-image optimized testing procedure can yield a significant increase in rate-distortion performance. Figure 2 illustrates the change in the likelihood of latent space values.

4.2. Latent adaptation vs Model retraining

In order to evaluate the benefit of an image-specific adaptation, we perform a comparative study with other forms of adaptation. A first option consists of retraining the *entire* model on the particular data to encode. Another option is to only update the probability model during the retraining procedure, i.e. to fix the feature extraction part of the network while refining the probability model on the content we wish to compress. From a practical point of view, these options are problematic as additional model updates have to be transmitted whereas the latent refinement offers a way of investing more capacity into the encoding process while all model parameters remain unchanged. In the following, we compare rate-distortion performance of latent adaptation with these two model retraining options without taking into account the cost of transmitting their updated weights. Our objective is to get some insights regarding the capabilities of latent adaptation. In its current form, model retraining is not a realistic alternative. Next, we evaluate adaptation in two scenarios; first in the case of content that is different in visual appearance and then on content that is different in resolution. For the remainder of the experiments, we will only consider the *full* model [10].

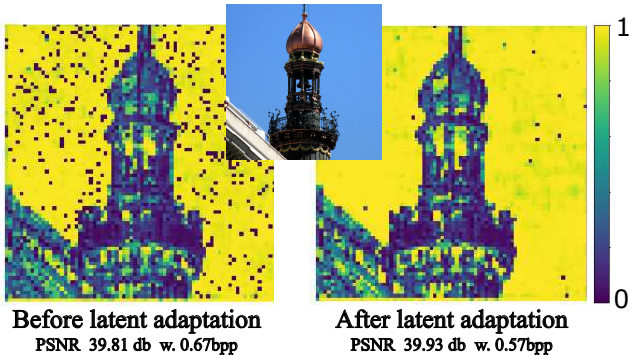

Figure 2: **Latent space adaptation.** Visualization of latent space likelihoods for one channel after adaptation.

Adapting to different visual appearance. In this experiment, we use two short movies with very different visual appearance: *Lucid Dreams* [1] and *Meridian* [2]. This allows to experiment two different cases in terms of content correlation. In *Lucid Dreams*, the frames share some similarity in terms of environment and characters, but the correlation is stronger in the second movie given the style and the lower number of scenes. In both cases, the videos are resized to a resolution of 1280×720. A common video streaming configuration is to use a key-frame every 2 seconds, allowing for robustness and adaptability to the network speed. As a result, in each case, our image test set consists of 70 frames extracted with regular spacing. Using the full model trained on the COCO dataset, we obtain the rate-distortion curve (dotted red curve) in figure 3. This constitutes our base result. Next, we describe the methodology used for

[1] https://www.youtube.com/watch?v=3zfV0Y7rwoQ
[2] https://www.netflix.com/title/80141336

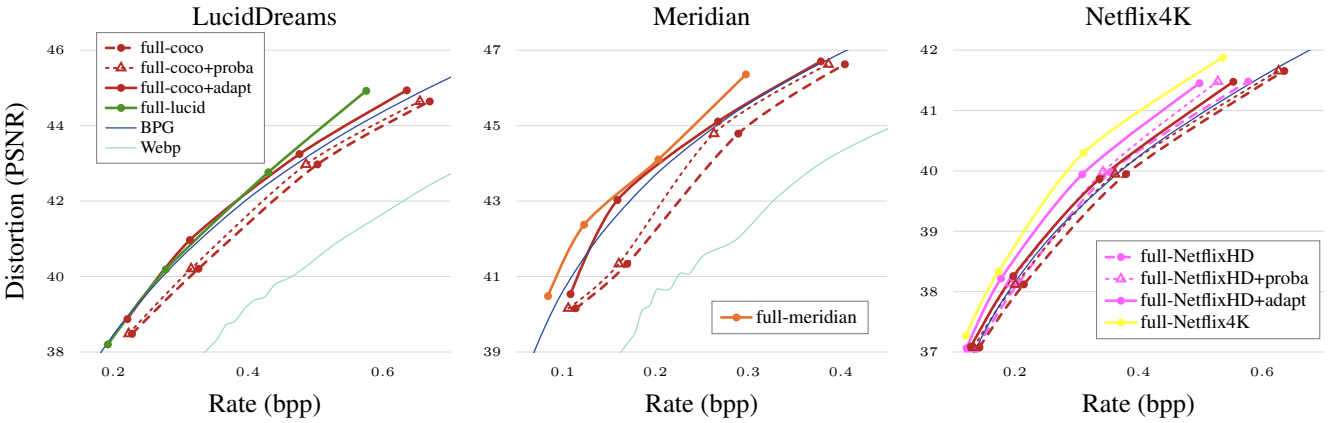

Figure 3: **Quantitative evaluation.** We compare the effect of various adaptation strategies in different experimental setups (from left to right), by adapting to different content and resolutions (see text for details).

comparing the different adaptation strategies.

From each movie we have extracted around 1200 frames (including the test frames). In a first setup, these images are used to train a compression network from scratch. The corresponding models are named *full-lucid* and *full-meridian*. In a second setup, the training is limited to the probability model. Starting from the pre-trained model, we only refine the probability model (hyper-encoder and hyper-decoder) using the new training sets. This is indicated in the model name. For example, in the LucidDream test, the model *full-coco+proba* corresponds to a model pre-trained on the COCO dataset for which the latent probability model was fine-tuned on *Lucid Dreams* test images. Finally, we apply our iterative algorithm to adapt the latent representation of each image while keeping the original compression network unchanged. We can see that, in both cases, adapting the latents on a per-image basis always outperforms fine-tuning the probability model. On the *Lucid Dreams* sequence, the latent adaptation even reaches the quality of the network specialized for this sequence. Given the stronger correlation on the *Meridian* frames, the specialized network performs better but adapting the latents still represents a significant improvement.

Adapting to different resolution. The objective of this experimental setup is to obtain insights regarding the behavior of models trained for different resolutions in terms of probability models and features. We extract a small set of 4K videos from the (*Netflix-4K*) collection, referenced by xiph.org [3]. On average, 20 frames per video are used as training data. For the test set, only 2 frames per video are used. In total, there are 25 test frames and 500 training frames. The model trained on the COCO dataset (*full-coco*) is tested first. We then explore adaptation, first by fine-tuning the probability model on the 4K training set, and second by adapting the latents on a per-image basis. To single out the effect of resolution, we train a new model from scratch, *full-NetflixHD*, using the frames from the set *Netflix-4K* resized to HD resolution. Then we similarly test fine-tuning the probability model and adapting the latents. Consistent with our previous experiments, adapting the latents always outperforms fine-tuning the probability model and represent a significant improvement (Fig. 3).

[3] https://media.xiph.org/video/derf/

5. Conclusion

In this work we have investigated content adaptive compression strategies which can be seen as a complementary approach of improving neural image coding besides architecture refinements. More specifically, we have presented a latent space refinement algorithm that allows to improve quality by roughly 0.5 dB at the same bit rate on the Tecnick data set. This strategy also allows to significantly close the gap between generic pre-trained models and models that are specifically trained for a given target content.Thus, the latent space adaptation can be an effective strategy to make a given encoding process more powerful and content adaptive. This is particularly beneficial in situations such as streaming, where the encoding complexity is not the limiting factor when compared to the transmission and decoding. As the gap towards models that are *entirely* trained on the specific target content cannot fully be closed, it would be interesting to further investigate which more complex but still practically viable form of adaptation may achieve this. Also currently, neural image compression models are typically trained for each rate-distortion point and it would be similarly beneficial to investigate strategies that allow automatic adaptation to each quality level.

References

[1] N. Asuni and A. Giachetti. Testimages: a large-scale archive for testing visual devices and basic image processing algorithms. In *Eurographics Italian Chapter Conference*, 2014. 1, 3

[2] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. *ICLR*, 2017. 1, 2, 3

[3] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. *ICLR*, 2018. 2

[4] F. Bellard. Bpg specification version 0.9.5, 2015. 3

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

[6] Google. Webp, 2010. 3

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[9] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool. Conditional probability models for deep image compression. In *CVPR*, 2018. 1, 2

[10] D. Minnen, J. Ballé, and G. D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*. 2018. 1, 2, 3

[11] D. S. Taubman and M. W. Marcellin. Jpeg2000: Standard for interactive imaging. *Proceedings of the IEEE*, 2002. 3

[12] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. *ICLR*, 2016. 1, 2

[13] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks. In *CVPR*, 2017. 1