A. Details of NIC codecs

Encoding consists of a learnable nonlinear mapping $\mathbf{z} = \tilde{f}(\mathbf{x}; \theta)$ where the feature encoder is parametrized by θ that maps an input image $\mathbf{x} \in \mathbb{R}^N$ to a latent representation $\mathbf{z} \in \mathbb{R}^M$. However, the latent representation is still redundant and in continuous space, which is not amenable to transmission through digital communication channels. Thus, the latent representation is quantized as $\mathbf{q} = Q(\mathbf{z})$, where $\mathbf{q} \in \mathbb{Z}^M$ is a discrete-valued symbol vector (in this paper we use quantization to the nearest integer, i.e. $\mathbf{q} = \lfloor \mathbf{z} \rceil$). Finally, the symbol vector $\mathbf{q}$ is binarized and serialized into the bitstream $\mathbf{b}$, using a (lossless) entropy encoder that exploits its statistical redundancy. Entropy coding is reversed during the process of decoding to obtain $\mathbf{q}$, which is then processed by the feature decoder in order to obtain the reconstructed image. This is be summarized by $\hat{\mathbf{x}} = \tilde{g}(\mathbf{q}; \phi)$, where $\hat{\mathbf{x}} \in \mathbb{R}^N$ is the reconstructed image, and the feature decoder, $\tilde{g}$ is parametrized by ϕ. In particular, we follow the framework of Balle *et al.* [3, 4], which combines convolutional layers, generalized divisive normalization (GDN) [2] and inverse GDN layers, scalar quantization and arithmetic coding.

During training, quantization is replaced by a differentiable proxy to allow end-to-end training via backpropagation [3]. In this paper, additive uniform noise $\tilde{\mathbf{z}} = \mathbf{z} + \Delta\mathbf{z}$, with $\Delta\mathbf{z} \sim \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)$ is used as the proxy. Rate is estimated as the entropy of the quantized symbol vector $R(\mathbf{b}) \approx H[P_{\mathbf{q}}] \approx H[p_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}}; \nu)]$, where ν are the parameters of the entropy model. The entropy model is known by both encoder and decoder. After training, the full model is thus determined by parameters $\psi = (\theta, \phi, \nu)$, with the full (including arithmetic coding) encoder $f(\mathbf{x}; \theta, \nu)$ and decoder $g(\mathbf{b}; \phi, \nu)$.

The parameters ψ are learned by minimizing a combination of rate and distortion over a training set $\mathcal{X}^{\text{tr}}$ sampled from the domain of interest $\mathcal{X}$

$$J\left(\mathcal{X}^{\text{tr}}, \psi; \lambda\right) = R\left(\mathcal{X}^{\text{tr}}, \psi\right) + \lambda D\left(\mathcal{X}^{tr}, \psi\right), \qquad (2)$$

where λ is the (fixed by design) tradeoff between rate and distortion. In this paper, distortion is measured as the average reconstruction mean square error (MSE)

$$D\left(\mathcal{X}^{\text{tr}}, \psi\right) = \mathbb{E}_{\mathbf{x} \in \mathcal{X}^{\text{tr}}} \left\|\hat{\mathbf{x}} - \mathbf{x}\right\|^2 \qquad (3)$$

and rate estimated as the average entropy

$$R\left(\mathcal{X}^{\text{tr}}, \psi\right) = \mathbb{E}_{\mathbf{x} \in \mathcal{X}^{\text{tr}}} H\left[p_{\tilde{\mathbf{z}}}\left(\tilde{\mathbf{z}}; \nu\right)\right]. \qquad (4)$$

B. Related work

Image compression Widely used image and video coding standards are based on the successful combination of prediction of pixels in small blocks, linear transforms (typically DCT), quantization and entropy coding [15]. Most advanced standards include carefully designed prediction and coding tools that greatly improve coding efficiency and quality [17, 11, 5]. Due to the pervasiveness of strictly defined decoder implementations in consumer devices, backward compatibility with legacy standards has always been a desirable characteristic in applications using image and video coding. A consequence of such approach is very limited adaptability of underlying codecs, leading to suboptimal use of compression technology in many application domains.

In contrast, NIC uses highly flexible parametric architectures whose parameters are learned from data. A typical architecture consists of a deep autoencoder followed by quantization and entropy coding (using differentiable proxies during optimization) [12, 3, 4, 14]. Recent methods include hyperpriors [4] and autoregressive probability models [14] to improve performance. However, the application of adapting learned image codecs to new domains while keeping the performance and compatibility with the original domain has not been considered.

Transfer learning and domain adaptation Deep neural networks (pre)trained on large datasets, can be reused and adapted (e.g. fine tuned) to specific target tasks, even with limited data (i.e.transfer learning) [1]. Domain adaptation [16] is the specific case when the task remains the same across domains (e.g. source and target tasks have the same categories). Here, we address domain adaptation in the context of image compression.

Continual learning Continual learning [10] addresses challenging scenarios where the model has to learn from continually arriving non-i.i.d. data from new unseen categories, domains or tasks. The most characteristic problem in this setting is the (catastrophic) forgetting [9] of previously learned skills, due to the interference between previous domains or tasks. It is often addressed using specific regularization, (pseudo)rehearsal of previous data or using task-specific parameters. Here, we study and address interference and forgetting in NIC.

Table 3. Comparison of the number of parameters between base model and *CAwF*

	Filters	Encoder	Entropy model	Decoder	Total parameters
Base	64	324736	6208	324675	655619
CAwF	80	505760	7760	505683	1019203