Three Gaps for Quantisation in Learned Image Compression

Shi Pan
Department of Computing
Imperial College London
London, UK
span@imperial.ac.uk

Chris Finlay
DeepRender,
London, UK
chris.finlay@deeprender.ai

Chri Besenbruch
DeepRender,
London, UK
chri.besenbruch@deeprender.ai

William Knottenbelt
Department of Computing
Imperial College London
London, UK
wjk@imperial.ac.uk

Abstract

Learned lossy image compression has demonstrated impressive progress via end-to-end neural network training. However, this end-to-end training belies the fact that lossy compression is inherently not differentiable, due to the necessity of quantisation. To overcome this difficulty in training, researchers have used various approximations to the quantisation step. However, little work has studied the mechanism of quantisation approximation itself. We address this issue, identifying three gaps arising in the quantisation approximation problem. These gaps are visualised, and show the effect of applying different quantisation approximation methods. Following this analysis, we propose a Soft-STE quantisation approximation method, which closes these gaps and demonstrates better performance than other quantisation approaches on the Kodak dataset.

1. Introduction

Image compression is a fundamental area of computer vision. The human visual system is more sensitive to noise in lower spatial frequencies, and ignores most higher frequencies. To exploit this phenomenon, traditional image compression methods, based upon a module-based block diagram, exploit quantisation to remove redundant information. For example, quantisation approaches based on the rounding function balance the compression rate with visual distortion by minimising quantisation residuals. However, in learned image compression, the situation is different.

In 2015, Toderici et al. first introduced neural networks in image compression [17]. In following works end-to-end learned image compression has demonstrated excellent results. Indeed, recent works (e.g. [9, 5]) have shown that end-to-end learned image compression methods delivers outstanding performance on various benchmark datasets, such as the Kodak and CLIC datasets. However, there are two main difficulties in end-to-end learned lossy image compression [2]: (1) quantisation is not differentiable, and (2) spatial redundancy must be efficiently reduced. This paper addresses the former: Ballé et al. [2, 3] has highlighted that the quantisation component used in conventional image compression methods is not differentiable. This non-differentiability prevents gradients from flowing to the encoder during end-to-end training. As a work-around, they added uniform noise as a simple approximation of quantisation, which makes the network end-to-end trainable. More recent work builds on this approach and delivers better performance by adding complicated neural blocks or autoregressive context models [18].

Despite this encouraging progress, there is still little insight into quantisation approximation itself, and in particular how it achieves such good performance by simply adding uniform noise during training. This lack of theoretical understanding of quantisation approximation has been noted as unsatisfactory by some researchers [8]. This paper addresses the underpinnings of the quantisation approximation with the following contributions:

1. **Three gaps theory**: We propose a framework of three gaps (**Discrete gap**, **Entropy Estimation gap** and **Local Smoothness gap**) to characterise the approximation of quantisation in learned image compression. We highlight that designing better quantisation approximations relies on an understanding of its inner work-

ings. In the visualisations of our experiments, different quantisation approximation methods are shown to significantly change the latent distribution, which in turn highly affects compression performance.

2. **Dequantisation**: We introduce a dequantisation step into the STE quantisation approximation to make entropy estimation meaningful (so-called Soft-STE). Adding noise to the latent variables [2] can be explained as a naïve dequantisation process. Our experiments show that models utilising the dequantisation approach achieve better performance.

3. **Adversarial perturbation**: We suggest that a quantisation approximation function will significantly change the loss landscape. The local Lipschitz constant can be used to measure these changes. We further suggest that models with small local Lipschitz constants should have better performance. This observation offers a possible direction to design better quantisation approximation methods. By following this idea, we introduce an adversarial perturbation to the proposed Soft-STE quantisation approximation. Experimentally, we see that adversarial perturbations can decrease the estimated score of the local Lipschitz constant, and results in better compression performance on the Kodak test set.

To evaluate the proposed methods, we adopt the widely used architecture in [2], and change only the quantisation approximation method. In particular, following [3], we use a hyper-prior network, but other widely-used pipeline components, such as context modelling and attention modules, are removed to keep the analysis simple and clear.

2. Related Work

Learned image compression has made tremendous progress. Toderici et al. first showed neural networks could be used for image compression tasks in 2015 [17]. Now, end-to-end trained compression models outperform traditional codecs such as JPEG2000[14] and BPG [4].

However, practically it is difficult to train deep neural networks for image compression tasks in an end-to-end fashion. The entropy coding step, which relies on Shannon Entropy (or discrete Cross-Entropy to be precise), requires a *discrete* representation of the bottleneck latent space. This introduces the need for some form of quantisation. At inference, common practice is to use integer rounding, which is not differentiable. Unfortunately, using integer rounding also in training prevents gradient flow, thus making optimisation of the Encoder impossible. Ballé et al. [2, 3] recognised this conflict and introduced an approximation of quantisation during training based on the addition of uniform noise, which makes the model end-to-end trainable. However, identifying the 'right' choice of quantisation approximation still remains a challenge. Despite significant follow-up work in recent years, such as the inclusion of auto-regressive components [10], and mixture-models [6], there has been little innovation on quantisation and quantisation approximation itself.

Any innovations in quantisation must address two major challenges: (1) the rounding function is not differentiable; and (2) the discrete probability (ground truth) mass function cannot be easily estimated by optimising the ELBO of its continuous approximation. To address the first problem, a simple approximation is to add uniform noise to the continuous latent variables [2, 3]. For example Ballé et al. claim this approach is based on a relaxed probability model: the discrete probability mass function (ground truth) is expected to be equal to the probability mass of the continuous probability density within the corresponding quantisation bin. This analysis may explain why their entropy model works in some cases. However, some questions remain unaddressed: namely why should the continuous latent variables require this additive noise, and how does this uniform noise help the decoder reconstruct the image from the quantised latent? To address the second problem, Ballé et al. use the CDF of the Gaussian distribution to estimate the relaxed continuous distribution of the latent space [3]. However, the selection of the quantisation approximation function significantly affects the distribution of the latent variables. As a result, the difference between the estimated distribution and the ground truth of the latent distribution affects compression performance.

There are many other quantisation approximation methods in the literature. Theis et al. introduced the straight-through estimator (STE) to approximate the quantisation step [16]. STE uses the rounding function in the forward pass, but an identity function replaces the gradient function in the backward pass. Unfortunately models trained with STE perform worse than models trained with a noise approximation. Hu et al. suggest that this is due to the fact that during training, optimisers require a *continuous* approximation of the quantised coefficient distribution [8]. But models trained with STE suffer from discontinuities (even if their gradients are approximated smoothly), and so are difficult to optimise. Other approaches are designed with a modified objective function. For example Yang et al. designed a continuous proxy by using a Bernoulli variable with a "tempered" distribution [19]. Their method directly optimises a discrete latent by sampling with the Gumbel-softmax trick. Agustsson et al. used soft-to-hard vector quantisation to replace the scalar quantisation approximation [1]. Dumas et al. introduced an additional model to learn the quantisation parameters [7]. Despite these innovations, almost all recent works utilise noise quantisation approximation due to its simplicity and superior performance.

3. Three Gaps in quantisation approximation

In this section, a theoretical framework is introduced for studying different quantisation approximation methods. We analyse the effect of selecting different quantisation approximations in the learned image compression pipeline.

3.1. Discrete Gap for noise approximation

In the literature, noise quantisation approximation (also known as additive noise) is understood as a relaxation of the rounding function. Instead of directly using discrete data, the optimisation process stays in the continuous domain, subject to a noisy perturbation with properties similar to rounding. However, there is an obvious gap in models trained with a noise quantisation approximation: The approximated latent distribution is fundamentally different from the ground truth latent distribution. We call this difference the **Discrete gap**. We suggest the Discrete gap changes the loss landscape. To be precise, with noise approximation [2], the gap is defined as follows:

$$\mathrm{GAP} = R(y) - Q_{\mathrm{noise}}(y) \tag{1}$$

$$Q_{\mathrm{noise}}(y) = \hat{y}_{\mathrm{noise}} = y + u \qquad u \sim U(-0.5, 0.5) \tag{2}$$

where $R(y) = \tilde{y}$ is the rounded latent, u is the uniform noise and $Q_{\mathrm{noise}}(y)$ indicates the noise quantisation approximation in end-to-end training. The formula shows that the data for inference will always be different from the data used in training. The fact that $\tilde{y} \neq \hat{y}$ partially explains the performance gap between training and validation. In general, with an arbitrary quantisation training function, we define the Discrete gap as follows:

$$G_d = \sum |R(y) - Q(y)|_p \tag{3}$$

where $Q(.)$ is an arbitrary quantisation function, and $|.|_p$ indicates the p-norm. In what follows, we use G_d to measure how close the quantisation approximation is to the ground truth. Large G_d will lead to a large performance gap between training and validation [8].

The discrete gap is obvious, especially in models that use noise quantisation approximation during training. Naturally, a good quantisation approximation method should be expected to simulate the actual rounding process in the forward pass, which will in turn help the decoder learn faithful reconstruction images from quantised latent. Note that the rounding function does indeed perturb the latents [2]. Additive uniform noise is used to mimic these rounding perturbations. However the uniform noise perturbations lead to extra difficulties for the decoder when trying to reconstruct images. During training, the decoder is expected to "denoise" these noisy perturbations. However, at inference, there is no reason to expect the actual quantisation residuals $\tilde{y} - y$ to follow the uniform distribution.

3.2. Discrete latent and the Entropy Estimation Gap

Of course, approximating quantisation with STE easily closes the Discrete gap, whereby the rounding function is used in the forward pass and the identity function is used as a stand-in for the gradient. However, simply applying STE approximation leads to *worse* compression performance [16]. We suggest that simply applying STE leads to failure to properly estimate the continuous entropy, which we call the **Entropy Estimation Gap**. This is the difference between the real distribution of the discrete latent and its continuous estimation during training, defined as:

$$G_{EE} = \mathrm{dist}(P_{\tilde{y}}(\tilde{y} = i) \; || \int_{\mathcal{B}} Q_{\hat{y}}(\hat{y}) \; d\hat{y}) \tag{4}$$

$P_{\tilde{y}}(.)$ is the probability mass function of the rounded, discrete, latent $\tilde{y}$, and $Q_{\hat{y}}(.)$ is the estimated density function for approximated latent $\hat{y}$. $\mathrm{dist}(. \; || \; .)$ can be any distance measurement between two distributions (such as the KL-divergence or Wasserstein Distance). The domain of integration $\mathcal{B}$ is used to represent the bin of quantisation (or hypercube associated if quantising vectors). In [2], $\int_{\mathcal{B}} Q(\hat{y}) d\hat{y}$ is represented as $\int_{\mathcal{B}} Q(u|y) du$ where u is sampled from a flexible continuous distribution (e.g. uniform distribution).

In the STE approximation, $P_{\tilde{y}}(\cdot)$ is a discrete distribution, and is not a continuous random variable. However, when we embed the STE distribution in a continuous space, we reasonably expect that the learned continuous entropy models will, during training, collapse onto point masses supported on the discrete points [15]. In particular, this collapse will bring additional difficulties in optimising the rate term. Thus, the gap G_{EE} of a model trained with STE approximation should be larger than models trained with noise approximation, due to the worse rate estimation.

3.3. Local Smoothness Gap

Quantisation approximation methods are used to produce "valid" gradient information. But well-defined gradients are not in themselves enough to ensure good compression performance. The gradient of the approximate quantisation can affect the compression performance by altering the loss landscape. In turn, the properties of this loss landscape determine how the training optimiser oscillates around a local optimum. We propose that the "local smoothness" of the loss landscape can be considered an indicator of the local optima. Moreover, good quantisation approximations should produce "smooth" loss landscapes which will help the optimisation converge.

The Lipschitz constant is a tool to measure the "smoothness" of the loss landscape. However, it is not feasible to compute the global Lipschitz constant within polynomial time. Fortunately, in learned image compression, quantisation can be understood as a small perturbation around the latent variables, and is local rather than global. Therefore we suggest that the *local* Lipschitz constant may be a more important metric in learned image compression. Following the definition of (the local) Lipschitz constant, we define the Local Smoothness Score (LSS) to measure the **Local Smoothness Gap**:

$$\mathrm{LSS} = \mathbb{E}_{\mathbb{D}}((L(\hat{y}+\xi) - L(\hat{y}))/(\xi)) \tag{5}$$

Here the perturbation ξ is added to the latent $\hat{y}$ *after* quantisation approximation. $L = L_D + \lambda L_R$ is the rate-distortion loss. $L_D = \mathbb{E}(p(x|\hat{y}))$ is the distortion loss for the reconstruction quality and $L_R = \mathbb{E}(-\log P(\hat{y}))$ is the rate loss measuring bitstream length. LSS is a measure of the loss's susceptibility to perturbations, in particular to quantisation perturbations. If the LSS is small, then a reasonable expectation is that the compression pipeline will be insensitive to quantisation perturbations. The LSS is well known elsewhere as a proxy for perturbation sensitivity; for example Miyato et al. [11] advocate the importance of the local Lipschitz constant in the adversarial robustness literature. We remark that the particular quantisation approximation function will highly affect the Local Lipschitz constant. For instance, STE approximation will lead to very high LSS. We suggest that smaller Lipschitz constants can improve compression performance.

4. Soft-STE quantisation approximation

The three gaps identified above characterise problems associated with various quantisation approximations in learned image compression. The Entropy Estimation Gap and the Local Smoothness gap are proposed to be reasons which limit the performance of the models trained with STE. To overcome these problems, we propose, **Soft-STE** quantisation, which we show reduces or eliminates these two gaps, and lead to a better compression performance.

Soft-STE is motivated as a remedy for the Entropy Estimation Gap, so that the rate term of the model with STE approximation will work in a meaningful way, and will be prevented from collapsing onto point masses. Therefore, in the rate-term, we do not use the STE approximation alone. Instead, we add continuous noise to the output of the STE approximation. In the literature, this approach is termed **Dequantisation** [12]. The total loss will be changed as:

$$L = \mathbb{E}(p(x|\hat{y})) + \lambda\mathbb{E}(-\log P(\hat{y}+\xi)) \text{ s.t. } \hat{y} = \mathrm{STE}(y) \tag{6}$$

Here STE(.) denotes STE quantisation approximation, and ξ is some continuous perturbation (possibly noisy), which relaxes the discrete coefficient. Uniform noise or Gaussian noise are possible noise distributions. Note, however, that perturbation is *only added in the rate term*, but not to the distortion term. Of course, noisy perturbations could be learned, which may lead to better dequantisation results. However, we also note that the selected perturbation will affect the Local Smoothness Gap, and will therefore impact performance.

For this reason, in our implementation, we implement Soft-STE using *adversarial perturbations*. In detail, we use the Fast Sign Gradient Method (FSGM), which is widely used in adversarial training and network robustness [13]. This method adds the 'noise' extracted from the Jacobian matrix to the original input during training. Thus in our implementation of Soft-STE, we calculate the **Adversarial perturbation** $\xi = \epsilon * \nabla L_R^*$. Here ∇L_R^* is the Jacobian matrix of rate loss without the dequantisation step. As a result, we hypothesise that the model so learned should be robust to perturbations within each quantisation bin. To ensure the ξ within quantisation bins, the ϵ is selected as the variance of $U(-0.5, 0.5)$.

Further to Soft-STE, we use **Spectral normalisation** to further constrain the local Lipschitz constant of the learned networks[11]. Since the rate term ∇L_R only backpropagates through the encoder, we only apply spectral normalisation to the encoder network. Our implementation adopts the fast approximation for the spectral norm of all weight matrices in the convolution layers [11].

5. Experiments

In this section, we empirically visualise the three gaps for models trained with different quantisation approximations. The compression performance of the proposed method is evaluated by using the Kodak dataset[1].

5.1. Experimental Setup

We use the common architecture first proposed in Ballé et al. [3]. A hyperprior network [10] is applied to predict mean and scale for the latent. We do not use context modelling, for simplicity. Various quantisation approximation methods are applied during training. The fully factorised distribution placed on the latent variable is assumed to be Gaussian $p(\hat{y}|z) = \mathcal{N}(\hat{y}; \mu_{\hat{y}}(z), \sigma_{\hat{y}}(z))$ where $\mu_{\hat{y}}(z)$ and $\sigma_{\hat{y}}(z)$ are the mean and scale predicted by the hyperprior network. The training examples are 256×256 pixel patches randomly cropped from a set of 1M high-resolution PNG images scraped from the internet. All models are optimised for mean squared error (MSE) as the distortion loss, and the Adam optimiser is used in training. We train for 2M iterations with a batch size of 2, and the learning rate starts as 10^{-4} and drops to 10^{-5} after 1.5M iterations. All runs are performed on one NVIDIA Tesla V100 GPU.

[1] Can be downloaded from `http://www.cs.albany.edu/~xypan/research/snr/Kodak.html`

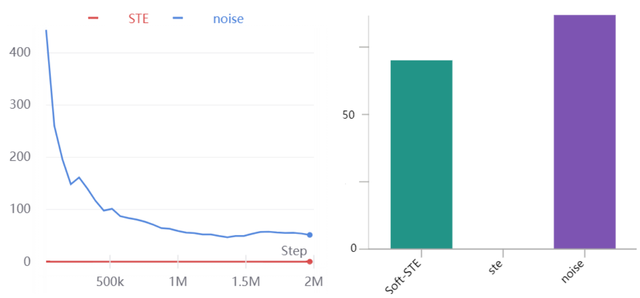


Figure 1. **Left:** Visualisation for the discrete gap during training using STE (Red) and noise (Blue). **Right:** Visualisation for the average discrete gap as per Equation 3 for model trained with Soft-STE (Green), STE (Pink) and noise (Purple). Note that STE does not have a discrete gap, and so the STE bar is not visible.

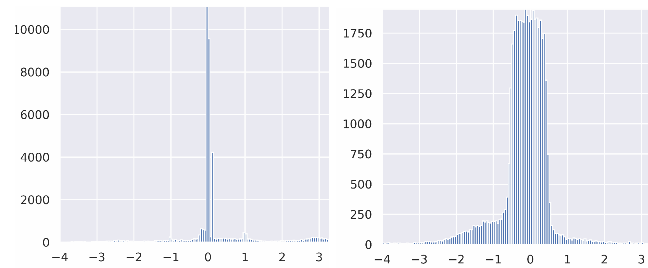


Figure 2. Average histogram for the latent at 300k iterations over Kodak dataset. **Left:** The model trained with STE approximation. **Right:** The model trained with noise approximation. Note that the STE model is collapsing towards a discrete distribution with mass centred on integer points.

5.2. Visualisation Results

Discrete Gap is defined in Eq. 3. A natural idea is to visualise this gap for different quantisation approaches during training. Figure 1 (left) visualises the discrete gap from the beginning of training to 2M iterations. We consider two models in this figure: the model trained with STE approximation [16] (red line) and noise approximation [2] (blue line). From this figure, it is apparent that the model with STE approximation maintains a discrete gap of zero. The discrete gap for the noise approximation decreases during training, but is non-negligible. We suggest that the decoder with noise approximation learns to denoise the uniform noise. However, the difference between uniform noise and the rounded residual $(R(y) - y)$ leads to a performance gap between training and validation.

Figure 1 (right) shows average G_d over the last 1 000 iterations during training. We consider three models: the proposed Soft-STE approximation (green), the model trained with STE approximation (pink), and noise approximation (purple). The model with Soft-STE approximation has a smaller average discrete gap G_d than the model with noise approximation. The proposed Soft-STE slightly closes this discrete gap during training.

Entropy Estimation Gap is defined in Eq. 4 and directly relates to the selected density function in the entropy (rate) loss. The quantisation approximation method changes the distribution of latent. This can be seen in Figure 2, which visualises the average histogram of the latent y at 300K iterations, comparing the model trained with STE approximation against noise approximation. The histogram is calculated over the entire Kodak dataset. We use 0.05 as the width of each histogram bin and fix our plot to the values between -4 and 3. Figure 2 illustrates the effect on the latent variable distribution from the two quantisation approximation methods. Noise approximation encourages the model to have a balanced distribution within the bin of quantisation $\mathcal{B}$. But for the model with STE approximation, bars around integer values are significantly higher than others. We suggest that this makes the latent y closer and closer to a discrete distribution during training. Using discrete data may force a continuous model to produce a degenerate solution that places all probability mass on discrete data points [15]. This difference partly explains why directly applying STE leads to poor compression performance.

Visualising the approximated entropy estimation gap G_{EE} is difficult. Entropy Estimation Gap is designed to illustrate the difference between the discrete probability mass for the rounded coefficients and the probability density distribution estimated by the model. Directly calculating G_{EE} is not tractable in practice, because the distribution of the rounded latent is unknown. We estimate G_{EE} by:

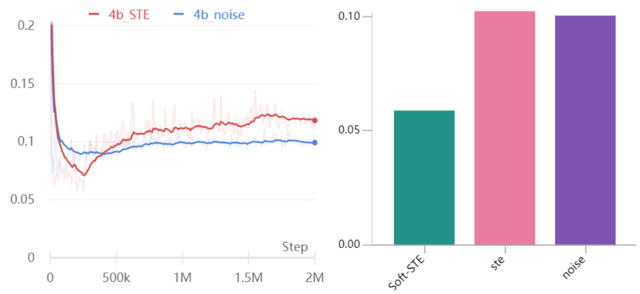
$$\hat{G}_{EE} = \mathbb{E}_{\hat{y}\sim P(\hat{y}), \check{y}\sim \mathcal{N}(\mu_y, \sigma_y)} \text{Wasserstein}(P(\hat{y}), \mathcal{N}(\mu_y, \sigma_y))$$

Here, $\tilde{y}$ is the rounded latent and $\check{y}$ is generated by a reparameterisation trick $\check{y} = \mu_y + \delta * \sigma_y$ where $\delta \sim \mathcal{N}(0, I)$.

Figure 3(a) is a smoothed line chart to show the approximated $\hat{G}_{EE}$ during training. We have also considered the model with STE approximation and the model with noise approximation. In this figure, the Wasserstein distance does not decrease monotonically throughout training.

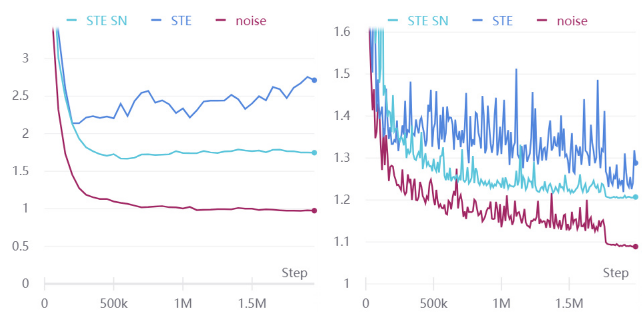
Applying STE quantisation approximation actually grows $\hat{G}_{EE}$ in the later stages of training. Linking this observation to the distribution of latent, we suggest that degenerate solutions enlarge $\hat{G}_{EE}$, making the model unstable.

The average $\hat{G}_{EE}$ for the model with STE approximation (pink), noise approximation (purple) and Soft-STE (green) can be found in Figure 3(b). Each model in this figure is fully trained to 2 million iterations. We then use another ten thousand iterations to estimate average $\hat{G}_{EE}$. We study three models: the proposed Soft-STE approximation (green), the model trained with STE approximation (pink), and noise approximation (purple). We suggest that the proposed Soft-STE approximation in part closes the entropy estimation gap for the image compression pipeline.



(a) $\hat{G}_{EE}$ (b) $\text{avg}(\hat{G}_{EE})$

Figure 3. Visualisation for the entropy estimation gap using Wasserstein distance. **3(a)** The model trained with STE (Red) and noise (Blue), visualising the entropy estimation gap using Wasserstein distance during training. The estimation of G_{EE} is calculated from the rounded latent and a data sampled from Gaussian distribution by using reparameterisation trick with the predicted μ_y, σ_y. **3(b)** Visualisation of the average Entropy Estimation gap in the fully trained model: Soft-STE (Green), STE (Pink) and noise (Purple).



(a) LSS (b) Loss

Figure 4. Visualise the LSS and related loss value for the model trained with noise approximation and STE approximation.

Local Smoothness Gap is also difficult to visualise. As per the analysis of Section 3.3, we calculate the Local Smoothness Score by using Monte Carlo simulation. Figure 4 demonstrate the Local Smoothness Score and the loss value for three different models. One model uses noise approximation (Maroon) and two models use STE approximation (Indigo and Blue). We apply spectral normalisation to all convolution layers in the STE SN model (Indigo).

From Figure 4(a), we see that both models with STE approximation have higher LSS during training. Spectral normalisation stabilises training and encourages a lower LSS for the model trained with STE approximation. However spectral normalisation alone cannot close the local smoothness gap. From Figure 4(b), we see that the model with higher LSS also has a higher loss value. This shows that the proposed LSS and local smoothness of the loss surface are highly related to compression performance.

5.3. Performance Comparison

To evaluate the effect of different quantisation approximations, four different approximation methods for quantisation are tested in this section. **Uniform approximation** is the baseline of this experiment [2, 9]. The implementation of this method follows the original paper [2]. At inference, the rounding function is applied to the pipeline to get the discrete coefficient as per the realised useage. **Uniform +** is a modified version of uniform approximation. The training and quantisation approximation part will not be changed, but $\tilde{y} = R(y - \mu) + \mu$ is applied in the validation/testing part. This means we can train a single model and evaluate it for both settings.

STE models use straight-through-estimator [16] to replace noise quantisation approximation during training. The rounded latent in the validation/test pass will be calculated by using the forward pass of STE. **Soft-STE** is our own proposed quantisation approximation method. We replace the naïve STE approximation with the proposed Soft-STE. For the validation/testing part, the Soft-STE approximation block is replaced by a rounding function $\tilde{y} = R(y - \mu) + \mu$. From the previous section, we have seen that Soft-STE can eliminate the entropy estimation gap and the local smoothness gap during training. In this section we will show that the quantisation approximation with smaller gaps leads to better compression performance.

PSNR-BPP is a common evaluation metric over the Kodak dataset. In our visualisations, different values of hyper-parameter λ in range $[10^{-4}, 0.3]$ are chosen to reach different bit rates. All models are initially trained with $\lambda = 0.01$, and we apply 100K iterations for different λs to fine-tune the model with different bit-rate settings. The learning rate of this fine-tuning step will be fixed to 10^{-4}. This design significantly decreases the training cost of this set of experiments; however, the performance score will be slightly lower than models trained with different λs from scratch.

Results of the performance comparison can be found in Figure 5. The blue line (noise) and the green line (STE) are two models which follow Ballé et al. [3] and Theis et al. [16]. The yellow line (noise +) denotes the model trained with noise approximation (same as blue line) but use $y_{\text{decoder}} = R(y - \mu) + \mu$ in the validation to minimise the quantisation residuals. The red line (S-ste) denotes the proposed model trained with soft-STE quantisation approximation. From the comparison in Figure 5, we can see the effectiveness of the proposed Soft-STE approximation methods. By comparing the model with noise (blue) and Soft-STE (red), the proposed Soft-STE approximation offers an extra performance gain of the pipeline described in [2] with hyper-prior [3]. If we only consider the model with STE function (red and green lines), the proposed Soft-STE significantly improve the performance of the model, in line with our hypothesis of the three gaps theory. Soft-STE ap-

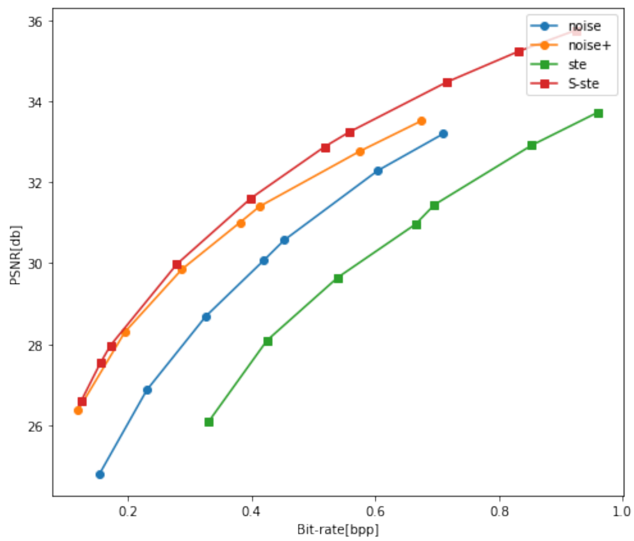


Figure 5. Evaluation of proposed quantisation with a hyperprior model on Kodak dataset. “noise” and “noise+” means the model trained with uniform noise approximation. “ste” and “S-ste” denote the model trained with Straight Through Estimator and the proposed Soft-STE approximation methods. The latter leads to improved performance by eliminating the entropy estimation gap and the local smoothness gap.

proximation eliminates the discrete gap (Figure 2 (Right)), entropy estimation gap (Figure 3(b)) and local smoothness gap during training. We highlight that using the rounding function $y_{\text{decoder}} = R(y - \mu) + \mu$ in the validation can decrease the quantisation residual (red and yellow lines).

6. Conclusion

In this paper, we have explored the quantisation approximation problem for learned image compression tasks. We defined a set of three gaps that are related to compression performance, and visualised these gaps in practice. The proposed theoretical framework can be used to analyse different quantisation approximation methods. We proposed a novel quantisation approximation method, Soft-STE, which demonstrates better performance than noise approximation on a benchmark dataset.

References

[1] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017.

[2] J. Ballé, V. Laparra, and E. P. Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015.

[3] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2016.

[4] F. Bellard. Bpg image format (http://bellard. org/bpg/), accessed: 2017-01-30.

[5] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Learning image and video compression through spatial-temporal energy compaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10071–10080, 2019.

[6] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020.

[7] T. Dumas, A. Roumy, and C. Guillemot. Autoencoder based image compression: can the learning be quantization independent? In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1188–1192. IEEE, 2018.

[8] Y. Hu, W. Yang, Z. Ma, and J. Liu. Learning end-to-end lossy image compression: A benchmark. *arXiv preprint arXiv:2002.03711*, 2020.

[9] J. Lee, S. Cho, and S.-K. Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[10] D. Minnen, J. Ballé, and G. D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.

[11] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[12] D. Nielsen and O. Winther. Closing the dequantization gap: Pixelcnn as a single-layer flow. *arXiv preprint arXiv:2002.02547*, 2020.

[13] A. Nøkland. Improving back-propagation by adding an adversarial gradient. *arXiv preprint arXiv:1510.04189*, 2015.

[14] M. Rabbani and R. Joshi. An overview of the jpeg 2000 still image compression standard. *Signal processing: Image communication*, 17(1):3–48, 2002.

[15] L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[16] L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

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

[18] M. Tschannen, E. Agustsson, and M. Lucic. Deep generative models for distribution-preserving lossy compression. In *Advances in Neural Information Processing Systems*, pages 5929–5940, 2018.

[19] Y. Yang, R. Bamler, and S. Mandt. Improving inference for neural image compression. *arXiv preprint arXiv:2006.04240*, 2020.