An Information-Theoretic Regularizer for Lossy Neural Image Compression

Supplementary Material

6. Proof of Lemma 1

For a deterministic quantization process $\mathrm{Q}(\cdot)$, the conditional probability $p(\boldsymbol{U}|\boldsymbol{X})$ can only take values of 0 or 1, i.e.,

$$p(\boldsymbol{U}|\boldsymbol{X}) = \begin{cases} 1, & \text{if } \boldsymbol{U} = \mathrm{Q}(\boldsymbol{X}) \\ 0, & \text{if } \boldsymbol{U} \neq \mathrm{Q}(\boldsymbol{X}) \end{cases}. \tag{17}$$

Similarly, for a deterministic dequantization process $\mathrm{Q}^{-1}(\cdot)$, it is by definition a bijection function, i.e., given the index $\boldsymbol{U}$ or the reconstruction $\hat{\boldsymbol{X}}$, we can uniquely determine the corresponding value of $\hat{\boldsymbol{X}}$ or $\boldsymbol{U}$, respectively. Therefore, we have

$$p(\hat{\boldsymbol{X}}|\boldsymbol{U}) = \begin{cases} 1, & \text{if } \hat{\boldsymbol{X}} = \mathrm{Q}^{-1}(\boldsymbol{U}) \\ 0, & \text{if } \hat{\boldsymbol{X}} \neq \mathrm{Q}^{-1}(\boldsymbol{U}) \end{cases}, \tag{18}$$

$$p(\boldsymbol{U}|\hat{\boldsymbol{X}}) = \begin{cases} 1, & \text{if } \boldsymbol{U} = \mathrm{Q}(\hat{\boldsymbol{X}}) \\ 0, & \text{if } \boldsymbol{U} \neq \mathrm{Q}(\hat{\boldsymbol{X}}) \end{cases}, \tag{19}$$

and overall,

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X}) = \begin{cases} 1, & \hat{\boldsymbol{X}} = \mathrm{Q}^{-1}(\mathrm{Q}(\boldsymbol{X})) \\ 0, & \hat{\boldsymbol{X}} \neq \mathrm{Q}^{-1}(\mathrm{Q}(\boldsymbol{X})) \end{cases}. \tag{20}$$

Thus, given Eqn.(18) to Eqn.(20), the following conditional entropy is by definition equal to 0:

$$H(\hat{\boldsymbol{X}}|\boldsymbol{U}) = H(\boldsymbol{U}|\hat{\boldsymbol{X}}) = H(\hat{\boldsymbol{X}}|\boldsymbol{X}) = 0. \tag{21}$$

Furthermore, from the equality of mutual information $I(\boldsymbol{X};\hat{\boldsymbol{X}})$ and the chain rule of joint entropy $H(\hat{\boldsymbol{X}},\boldsymbol{U})$, we have

$$\begin{cases} I(\boldsymbol{X};\hat{\boldsymbol{X}}) = H(\hat{\boldsymbol{X}}) - H(\hat{\boldsymbol{X}}|\boldsymbol{X}) \\ H(\boldsymbol{U}|\hat{\boldsymbol{X}}) + H(\hat{\boldsymbol{X}}) = H(\hat{\boldsymbol{X}}|\boldsymbol{U}) + H(\boldsymbol{U}) \end{cases}. \tag{22}$$

Substituting Eqn.(21) into Eqn.(22), we conclude

$$\begin{cases} I(\boldsymbol{X};\hat{\boldsymbol{X}}) = H(\hat{\boldsymbol{X}}) \\ H(\hat{\boldsymbol{X}}) = H(\boldsymbol{U}) \end{cases}, \tag{23}$$

and thus

$$H(\boldsymbol{U}) = I(\boldsymbol{X};\hat{\boldsymbol{X}}). \tag{24}$$

7. Proof of Lemma 2

Recalling the proof in Sec. 6, since both the analysis transform $\mathrm{T}_\mathrm{A}(\cdot)$ and synthesis transform $\mathrm{T}_\mathrm{S}(\cdot)$ are deterministic, the following holds:

$$p(\boldsymbol{U}|\boldsymbol{X}) = \begin{cases} 1, & \text{if } \boldsymbol{U} = \mathrm{Q}(\mathrm{T}_\mathrm{A}(\boldsymbol{X})) \\ 0, & \text{if } \boldsymbol{U} \neq \mathrm{Q}(\mathrm{T}_\mathrm{A}(\boldsymbol{X})) \end{cases}, \tag{25}$$

$$p(\hat{\boldsymbol{X}}|\boldsymbol{U}) = \begin{cases} 1, & \text{if } \hat{\boldsymbol{X}} = \mathrm{T}_\mathrm{S}(\mathrm{Q}^{-1}(\boldsymbol{U})) \\ 0, & \text{if } \hat{\boldsymbol{X}} \neq \mathrm{T}_\mathrm{S}(\mathrm{Q}^{-1}(\boldsymbol{U})) \end{cases}, \tag{26}$$

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X}) = \begin{cases} 1, & \text{if } \hat{\boldsymbol{X}} = \mathrm{T}_\mathrm{S}(\mathrm{Q}^{-1}(\mathrm{Q}(\mathrm{T}_\mathrm{A}(\boldsymbol{X})))) \\ 0, & \text{if } \hat{\boldsymbol{X}} \neq \mathrm{T}_\mathrm{S}(\mathrm{Q}^{-1}(\mathrm{Q}(\mathrm{T}_\mathrm{A}(\boldsymbol{X}))))) \end{cases}, \tag{27}$$

and therefore

$$H(\hat{\boldsymbol{X}}|\boldsymbol{U}) = H(\hat{\boldsymbol{X}}|\boldsymbol{X}) = 0. \tag{28}$$

Herein, the core distinction between the transform coding and direct coding models lies in the fact that the synthesis transform $\mathrm{T}_\mathrm{S}(\cdot)$, unlike the dequantization function $\mathrm{Q}^{-1}(\cdot)$, does not inherently guarantee a bijective mapping, particularly in the context of neural transforms. Consequently, given the reconstruction $\hat{\boldsymbol{X}}$, there may be uncertainty in $\hat{\boldsymbol{Y}}$ and thus $\boldsymbol{U}$, i.e.,

$$H(\boldsymbol{U}|\hat{\boldsymbol{X}}) \neq 0. \tag{29}$$

Substituting Eqn.(28) and Eqn.(29) into Eqn.(22), we can conclude

$$\begin{cases} I(\boldsymbol{X};\hat{\boldsymbol{X}}) = H(\hat{\boldsymbol{X}}) \\ H(\hat{\boldsymbol{X}}) = H(\boldsymbol{U}) - H(\boldsymbol{U}|\hat{\boldsymbol{X}}) \end{cases}, \tag{30}$$

and thus

$$H(\boldsymbol{U}) = I(\boldsymbol{X};\hat{\boldsymbol{X}}) + H(\boldsymbol{U}|\hat{\boldsymbol{X}}). \tag{31}$$

Table 3. BD-Rate comparison between our reproduction and the pre-trained models [44].

	Kodak	Tecnick	CLIC	Average
hyperprior	0.52%	1.88%	-1.24%	0.39%
autoregressive	0.56%	2.25%	-1.56%	0.41%
attention	-3.32%	-3.33%	-5.68%	-4.11%

8. Reproduced baselines

We retrain the *hyperprior* [4], *autoregressive* [8], and *attention* [10] models from scratch, adhering to the default implementation and training configurations of CompressAI's [44]. Four bit-rate points, i.e., $\lambda \in \{0.0018, 0.0035, 0.0067, 0.0130\}$ are trained with 2×10^6 steps. The evaluation results are summarized in Table 3. On

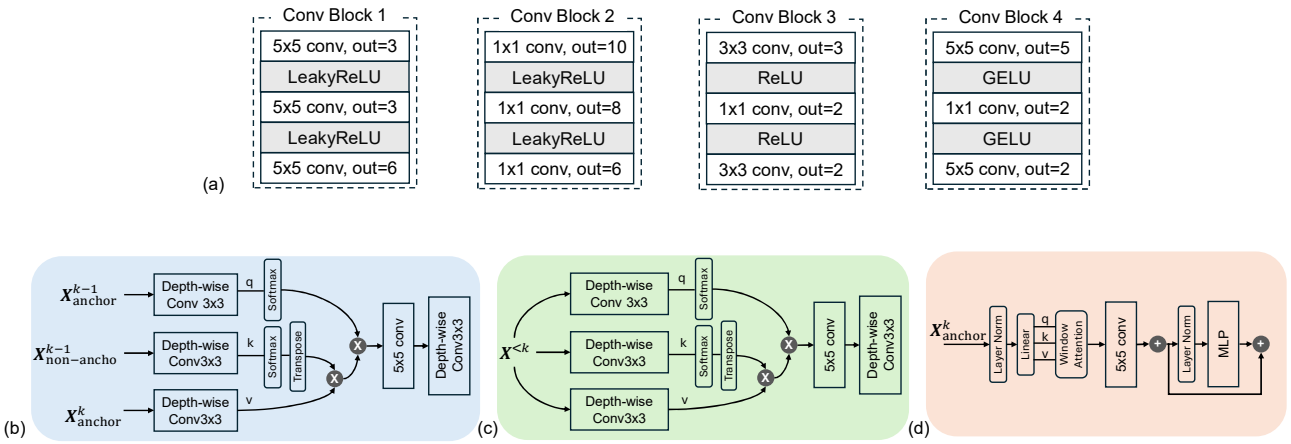


Figure 7. Details of (a) Four convolutional modules. (b) Inter attention module, (c) Intra attention module, and (d) Checkerboard attention module from MLIC++ [12].

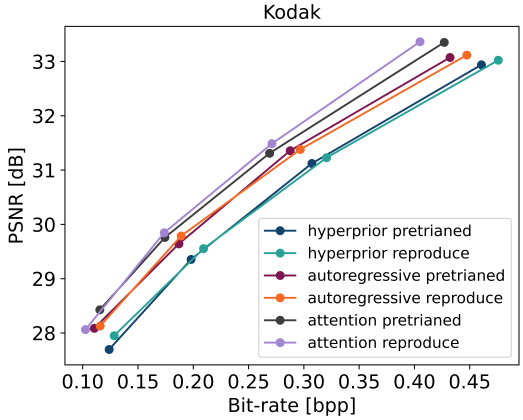


Figure 8. Comparison between the pre-trained models from CompressAI's and our reproduction on the Kodak dataset. Our reproduction achieves 0.52%, 0.56%, and -3.32% BD-Rates for the *hyperprior*, *autoregressive*, and *attention* models, respectively.

average, when compared to the pre-trained models of CompressAI's, our reproduction models yield 0.39%, 0.41%, and -4.11% BD-Rates [57] for the *hyperprior*, *autoregressive*, and *attention* models, respectively. Notably, on the CLIC dataset, our reproduction models outperform their pre-trained counterparts by -1.24%, -1.56%, and -5.68% BD-Rates for the *hyperprior*, *autoregressive*, and *attention* models, respectively. For the *attention* model, our reproduction model outperforms the pre-trained model on all three evaluation datasets of Kodak, CLIC, and Tecnick by -3.32%, -3.33%, and -5.68% BD-Rates, respectively. Note that in the original *attention* network [10], the latent is modeled as a mixture of Gaussians with 3 clusters. In CompressAI's default implementation, this cluster number is simplified to 1. We follow this latent implementation. The detailed rate-distortion curves for the Kodak dataset are visualized in Fig.8.

9. More details on source entropy models

Herein, the module designs are identical to the latent designs [4, 8, 10, 12, 43], with only minor dimension adjustment to fit $\boldsymbol{X}$. The details are depicted in Fig.7. For the attention modules from MLIC++, the depthwise separable convolution is adopted [56].

10. Regularization with $H(\boldsymbol{U}|\hat{\boldsymbol{X}})$ estimation

As shown in Fig.9(a), we further examine the regularization performance by training compression networks with the following minimization objective:

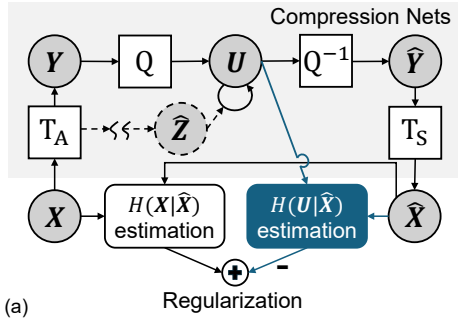
$$R + \lambda D + \alpha \underbrace{\mathbb{E}_{\boldsymbol{X}}[\log q_\theta(\boldsymbol{X}|\hat{\boldsymbol{X}}) - \log q_\varphi(\boldsymbol{U}|\hat{\boldsymbol{X}})]}_{\approx -(H(\boldsymbol{X}|\hat{\boldsymbol{X}}) - H(\boldsymbol{U}|\hat{\boldsymbol{X}}))}, \quad (32)$$

where an additional entropy model $q_\varphi(\boldsymbol{U}|\hat{\boldsymbol{X}})$ is jointly estimated with $q_\theta(\boldsymbol{X}|\hat{\boldsymbol{X}})$ by maximizing

$$\max \ \mathbb{E}_{\boldsymbol{X}}[\log q_\theta(\boldsymbol{X}|\hat{\boldsymbol{X}}) + \log q_\varphi(\boldsymbol{U}|\hat{\boldsymbol{X}})]. \quad (33)$$

The training procedure follows Algorithm 1, consisting of alternating steps: one step updates the compression network using Eqn.(32), while another step updates the regularizer using Eqn.(33). The distribution $q_\varphi(\boldsymbol{U}|\hat{\boldsymbol{X}})$ is modeled as a factorized Gaussian, with its neural architecture identical to that of the analysis transform $\mathrm{T_A}$. We evaluate this approach on the *hyperprior* model, with the regularization factor α set to 0.1 and all other training settings consistent with Sec. 4.1. The corresponding results are presented in Fig.9(b). As shown in Fig.9(b), introducing the $H(\boldsymbol{U}|\hat{\boldsymbol{X}})$ term only provides a marginal gain (approximately -0.2% BD-Rate) on the CLIC dataset at 0.5×10^6

steps, while no significant improvement is observed in other cases. One possible explanation is that learning the distribution $q_\varphi(\boldsymbol{U}|\hat{\boldsymbol{X}})$ is highly challenging, as it requires training an additional "encoder" function to accurately map $\hat{\boldsymbol{X}}$ to $\boldsymbol{U}$. Moreover, due to computational complexity considerations, the proposed Algorithm 1 updates the regularization terms with only one gradient step per iteration. Experiments reveal that even with this single-step update strategy, the overall training complexity has already increased from approximately 28% to 45%. One might suggest reusing the entire analysis transform T_A instead of learning a new mapping from $\hat{\boldsymbol{X}}$ to $\boldsymbol{U}$ to reduce complexity. We have explored this alternative, but its performance is inferior to the results presented here.



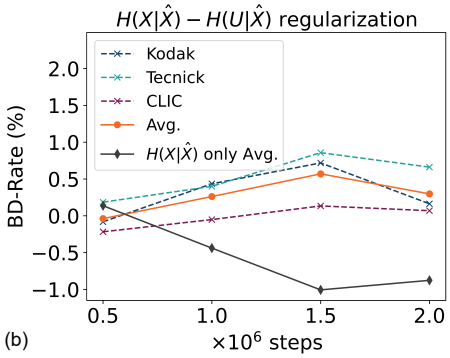


Figure 9. (a) Illustration of regularization with an additional $H(\boldsymbol{U}|\hat{\boldsymbol{X}})$ term; (b) Performance evaluation on the *hyperprior* model.