arXiv:2301.08752v1 [eess.IV] 20 Jan 2023

面向实用型基于神经网络的图像和视频压缩的优化学习熵编码参数

Amir Said Reza Pourreza HoangLe
Qualcomm AI Research，* 美国圣迭戈

摘要

将权重和激活值量化为低精度整数，可显著提升基于神经网络的图像和视频编解码器的能效。虽然已有通用技术可减轻量化效应，但若未考虑特定的熵编码特性，仍可能造成较大损失。本文分析参数量化对熵编码的影响，并提出一种将损失降至最低的方法。研究表明，通过采用某种可学习的编码参数类型，均匀量化在实践中可达到最优，同时也简化了代码存储需求的最小化过程。本文介绍了这种新表示的数学性质，并通过编码实验验证其有效性：每个网络输出的精度低至 4 位时仍可获得良好结果，而精度为 8 位时几乎没有损失。

索引词——学习式图像和视频压缩、神经网络量化、熵编码。

1. 引言

经过多年研究，神经网络技术在视频编码方面的优势已得到证实 [1, 2], ，近期也展示了其在移动设备上实现实时解码的能力 [3, 4], 。人们对其标准化和商业部署的兴趣日益浓厚 [5], ，而实际应用中的一项主要挑战是尽可能降低解码所需的功耗。

使用低精度整数实现神经网络（NN）计算可显著降低功耗，但需要尽量减小对网络权重和激活值进行量化所产生的影响 [6, 7, 8]。对于神经网络图像和视频编解码器，其性能很大程度上取决于熵编码阶段及相关网络，并可能因此严重下降。

分析神经网络中的量化效应可能颇为困难，因为神经网络执行的是非线性过程，且学习得到的变量具有未知属性。基于量化约束进行重新训练的技术有助于减少损失，但效果有限。

本文表明，用于熵编码的神经网络输出具有明确的含义和性质，可用于预测量化效应。此外，这些性质还可用于修改学习过程中

*Qualcomm AI Research 是 Qualcomm Technologies, Inc. 的一项计划。

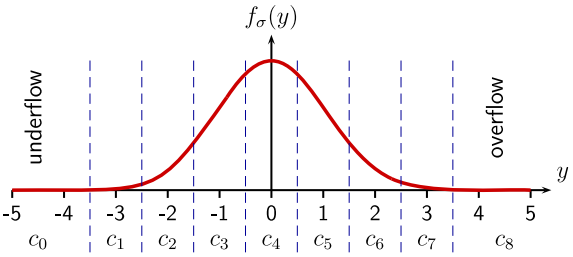


图 1。正态分布中经量化的数值，标准差为 σ，常用于基于神经网络的编解码器。

使用的损失函数，并且研究表明，可以训练网络，使其对量化不敏感。

进一步分析表明，这种优化后的表示对量化具有显著的独立性，因此，即使熵编码设计选项发生重大变化，也无需重新训练网络。因此，在这些网络中，均匀量化也可以扩展到超出硬件要求的范围，并用于最大限度地减少存储熵编码表所需的存储器。

实验结果表明，所提出的方案在精度低至 4 位时仍能保证良好的性能（冗余为 ≈ 2%）；每增加一位，相对冗余便可降低至原来的四分之一，精度达到 8 位时，冗余低于 0.01%。

2. 基于神经网络的编解码器的熵编码

传统编解码器中的熵编码基于对待编码变量概率的估计。例如，在 HEVC 和 VVC 等视频标准中，使用 [9, 10, 11], 自适应二进制算术编码上下文来估计与数据元素相关的概率。

基于神经网络的编解码器采用不同的原理：训练时选择一个参数化概率分布（pdf），然后使用学习得到的方案（表或神经网络）确定用于编码量化随机变量的参数值。图 1 展示了常用正态分布和单位步长量化的示例。

随机变量 $y \sim \mathcal{N}(0, \sigma^2)$ 量化为值 $n \in \mathbb{Z}$ 时的概率序列为

$$p_n(\sigma) = \frac{1}{2}\left[\operatorname{erfc}\left(\frac{2n-1}{\sqrt{8}\sigma}\right) - \operatorname{erfc}\left(\frac{2n+1}{\sqrt{8}\sigma}\right)\right], \quad (1)$$

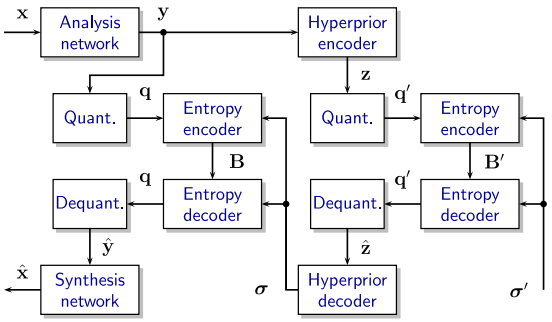


图 2。基于神经网络的编解码器架构示例，其中包含用于熵编码的尺度超先验网络。

其熵为

$$H(\sigma) \stackrel{\text{def}}{=} -\sum_{n=-\infty}^{\infty} p_n(\sigma)\log_2(p_n(\sigma))\,, \tag{2}$$

其中，互补误差函数为

$$\operatorname{erfc}(x) \stackrel{\text{def}}{=} \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2}\,\mathrm{d}t. \tag{3}$$

在实际实现中，会使用特殊符号对下溢和上溢无限集合中的值进行编码。这些值的概率虽极小但不为零，如图 1 所示。在这些集合中，值会以非最优方式编码，例如使用 Elias 通用码 [12]。

图 1 还展示了码向量 $\mathbf{c}(\sigma)$（CV）中的元素，这些元素包含在给定范围 R 内用于算术编码（AC）的概率

$$c_i(\sigma) = \begin{cases} \sum_{n=-\infty}^{-R} p_n(\sigma), & i = 0, \\ p_{i-R}(\sigma), & i = 1, 2, \ldots, 2R-1, \\ \sum_{n=R}^{\infty} p_n(\sigma), & i = 2R. \end{cases} \tag{4}$$

为简化符号表示，我们忽略实现细节，例如将概率转换为用于 AC 的整数累积和 [13, 14]。假设 R 和 AC 精度的选取能够使压缩效果非常接近熵。

这种熵编码方式用于基于神经网络的编解码器中，如图 2 所示，对应于尺度超先验架构 [15]。还有其他编解码器配置，但我们主要关注图 3 所示的逐元素编码阶段，该阶段适用于所有因子化先验。

待编码的潜变量 y 采用 $Q_{y\to q}$进行量化，并在浮点训练期间将其纳入考量。此处讨论的问题由量化阶段 $Q_{\sigma\to\hat{\sigma}}$引起，该阶段意味着编码时使用低精度网络输出，而训练时不使用。

由于编码器和解码器 [16], 必须获得逐位完全一致的码向量 $\mathcal{U} = [\sigma_{\min}, \sigma_{\max}]$，我们假设它们共享一个单调序列 $\{t_k\}_{k=0}^{N}$ 分区 $\mathcal{U}$，并且在 $\hat{\sigma} \in [t_k, t_{k+1})$ 时，使用 $\mathbf{c}(\rho_k^*)$ 对潜变量 y 进行编码。

为了确定用于码向量计算的 ρ_k^* 最优值，我们测量平均冗余，其定义为

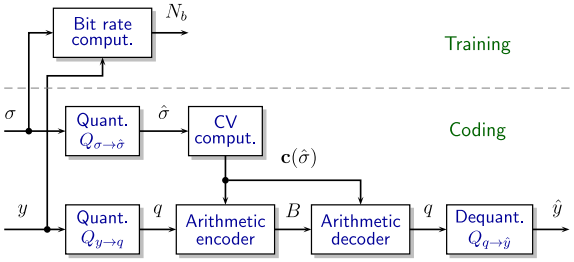


图 3。逐元素熵编码，包括概率密度函数参数 σ的量化。

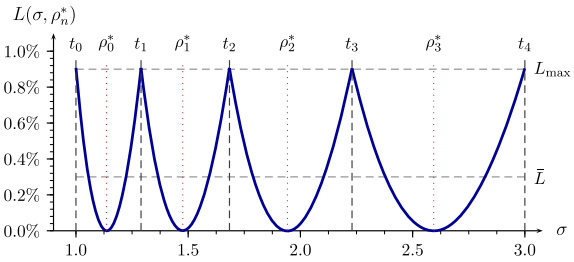


图 4。参数 σ的非均匀量化示例，旨在实现恒定的最大相对冗余。

Kullback-Leibler散度

$$R(\sigma,\rho) \stackrel{\text{def}}{=} \sum_{n=-\infty}^{\infty} p_n(\sigma)\log_2\left(\frac{p_n(\sigma)}{p_n(\rho)}\right). \tag{5}$$

由于实用的图像和视频编解码器旨在适用于广泛的比特率，因此采用旨在最小化相对冗余的熵编码颇具意义

$$L(\sigma,\rho) \stackrel{\text{def}}{=} \frac{R(\sigma,\rho)}{H(\sigma)}. \tag{6}$$

并针对所有使用场景启用良好性能

$$\rho_k^* = \arg\min_{\rho} \int_{t_k}^{t_{k+1}} L(\sigma,\rho)\,\mathrm{d}\sigma. \tag{7}$$

3. 新的 PDF 参数化

根据这些定义，可以优化非均匀量化方案，如图 4 所示的示例。与其他量化方案类似，所有区间都具有相同的最大相对冗余和平均值

$$\bar{L}_k \stackrel{\text{def}}{=} \frac{1}{t_{k+1}-t_k}\int_{t_k}^{t_{k+1}} L(\sigma,\rho_k^*)\,\mathrm{d}\sigma. \tag{8}$$

对 σ 进行足够精细的量化后，$L(\sigma,\rho_k^*)$ 近似为二次函数，因此我们可以使用

$$\rho_k^* \approx \{\rho : L(t_k,\rho) = L(t_{k+1},\rho)\}\,, \tag{9}$$

$$\bar{L}_k \approx \frac{L(t_k,\rho_k^*)}{3} = \frac{L(t_{k+1},\rho_k^*)}{3}. \tag{10}$$

基于神经网络的编解码器可以使用预先计算的码向量数组来实现精确的浮点运算

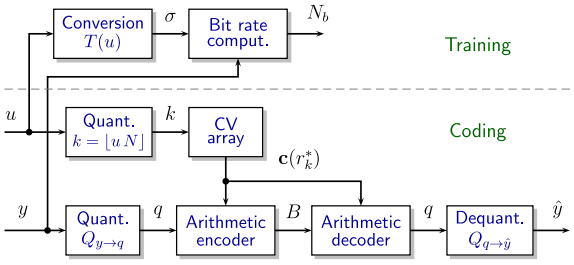


图 5。图 3 中编解码器的拟议改进：将概率密度函数参数 σ 替换为针对量化优化的参数 u。

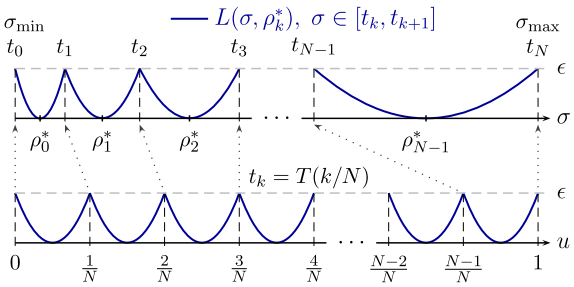


图 6。从非均匀量化到均匀量化的映射。

如图 4 示例所示的方法。然而，对于必须处理广泛比特率的实用编解码器，有必要支持例如 $\sigma \in [0.1, 1000]$，同时将超先验网络输出 $\hat{}\sigma$ 的精度限制为仅 8 位。这意味着 $\{t_k\}_{k=0}^N$ 也只能使用 8 位，从而严重限制设计并造成显著损失。

为了解决这个问题，我们可以利用这样一个事实：σ 仅在训练期间用于码率估计以及计算码向量。拟议的解决方案如图 5 所示：训练超先验网络以计算新的概率密度函数参数$u \in [0, 1]$，并将其转换为 $\sigma = T(u)$。为便于分析，我们定义 $u \in \mathbb{R}$；但请注意，在实际编码过程中，量化网络只需生成整数 k。此方法用于 [16],，其中转换形式选为$\sigma = e^{\alpha u+\beta}$。

图 6 展示了如何针对给定的 N 定义最优的 $T(u, N)$，以获得恒定的最大相对冗余，并可使用以下算法进行计算。

1. 给定 N和 $[\sigma_{\min}, \sigma_{\max}]$，选择初始最大冗余度 ϵ；初始化 $t_0 = \sigma_{\min}$。
2. 对于 $n = 1$、2、.....、N:
   (a) Set $\rho_{n-1}^* = \{\rho > t_{n-1} : L(t_{n-1}, \rho) = \epsilon\}$;
   (b) Set $t_n = \{t > \rho_{n-1}^* : L(t, \rho_{n-1}^*) = \epsilon\}$.
3. 如果 $t_N \approx \sigma_{max}$ 则停止。
4. 若 $t_N < \sigma_{max}$，则使用单维搜索方法增加 ϵ；否则减少 ϵ。5. 转到步骤 2。

对若干个 N 值计算 $T(u, N)$ 并在采样值之间进行插值，得出的结论令人惊讶：所有结果实际上都几乎相同。这意味着，如果我们计算

$$T(u) = \lim_{N\to\infty} T(u, N), \tag{11}$$

我们可以得到一个单一的参考函数，可用于所有网络训练，并且 N的值可以稍后选择和修改，而无需重新训练。

确定 $T(u)$ 的依据可从图 6 的下半部分看出。随着 N 的增大，所有区间中的冗余曲线都应近似于关于 u 的二次方程，并具有相同的二阶导数。如果我们能够找到 $T(u)$ 和常数 α，使得

$$\frac{\partial^2}{\partial u^2} L(T(u), \rho)\bigg|_{\rho=T(u)} = \alpha, \quad u \in [0, 1]. \tag{12}$$

定义

$$\psi(\sigma) \stackrel{\text{def}}{=} \frac{\partial^2}{\partial \sigma^2} L(\sigma, \rho)\bigg|_{\rho=\sigma} \tag{13}$$

可以证明，得益于 $L(\sigma, \rho)$ 的性质，式子等价于 (12)

$$\frac{\mathrm{d}}{\mathrm{d}u} T(u) = \sqrt{\frac{\alpha}{\psi(T(u))}}, \tag{14}$$

因此，可以通过求解以下常微分方程（ODE），并使用边界条件 $T(0) = \sigma_{\min}$、$T(1) = \sigma_{\max}$，来确定函数 $T(u)$。

对于任意一组可微的参数化概率，可以证明

$$\psi(\sigma) = \frac{1}{\ln(2)H(\sigma)} \sum_{n=-\infty}^{\infty} \frac{1}{p_n(\sigma)} \left[\frac{\mathrm{d}p_n(\sigma)}{\mathrm{d}\sigma}\right]^2, \tag{15}$$

对于正态概率密度函数，有

$$\psi(\sigma) = \frac{1}{\ln(2)H(\sigma)} \left\{ \frac{[\phi_1(\sigma)]^2}{1-\omega_1(\sigma)} + \sum_{n=1}^{\infty} \frac{[\phi_{2n-1}(\sigma) - \phi_{2n+1}(\sigma)]^2}{\omega_{2n-1}(\sigma) - \omega_{2n+1}(\sigma)} \right\}, \tag{16}$$

其中

$$\omega_k(\sigma) \stackrel{\text{def}}{=} \operatorname{erfc}\left(\frac{k}{\sqrt{8}\sigma}\right), \quad \phi_k(\sigma) \stackrel{\text{def}}{=} \frac{k\, e^{-k^2/(8\sigma^2)}}{\sqrt{2\pi}\sigma^2}. \tag{17}$$

4. 实验结果

方程（14）的常微分方程使用四阶龙格–库塔法 [17, §17.1]，求解，边界条件为 $\sigma \in [0.1, 1000]$。解 $T(u)$ 如图 7 所示。粗略观察可能会得出错误结论，认为它可以用类似 10^{4u^2-1} 的函数近似（红线）。

由于 $T(u)$ 由微分方程定义，因此必须通过导数相对误差来衡量近似的质量。图 8 展示了这些误差，我们可以

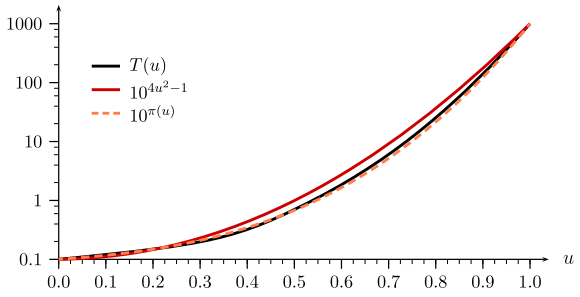


图 7。最优的概率密度函数参数转换函数 $T(u)$ 和两种近似。

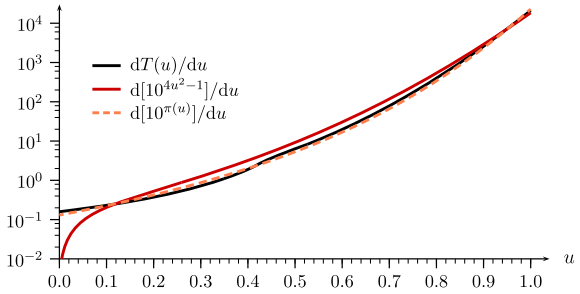


图 8。概率密度函数参数转换函数 $T(u)$ 及其近似的导数。

还可以看到一个好得多的近似 $T_\pi(u) = 10^{\pi(u)}$，其中多项式

$$\pi(u) = 2.49284\,u^3 + 0.93703\,u^2 + 0.57013\,u - 1, \quad (18)$$

使最大相对导数误差最小。

图 9 展示了对概率密度函数参数 u 进行均匀量化，并使用 N 码向量对潜变量 y 进行熵编码时，平均相对冗余 $L(\sigma)$ 随 σ 的变化情况，如图 5 所示。根据公式（13）定义的条件，使用 $T(u)$ 进行概率密度函数参数转换时，$L(\sigma)$ 不会改变。另一方面，使用近似 $T_\pi(u)$ 时会出现相对较小的变化，这些偏差对应于导数近似中的误差（参见图 8）。

我们还观察到，这些曲线几乎完全平行，证实了相对冗余的泰勒级数中二次项占主导地位。当 N 减半时，$L(\sigma)$ 增至四倍，这一事实也验证了这一点。

对柯达图像数据集中的尺度超先验编解码器 [15]，浮点实现所产生的值，使用由变换$T(u)$定义的非均匀量化进行熵编码。表 1 展示了冗余结果，平均比特率范围为 0.25 至 2.5 比特/像素。

当 N 的值大于约 128（7 位表示）时，量化带来的冗余为 0.01% 或更低，处于实验误差范围内。对于较小的 N 值，我们可以观察到

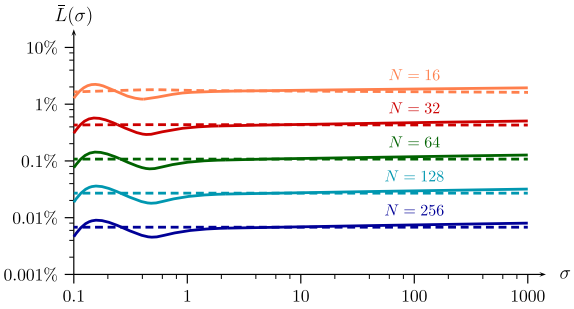


图 9。不同量化区间数 N下，$T(u)$的平均相对冗余（虚线）与近似值 $10^{\pi(u)}$（实线）。

表 1。将尺度超先验编解码器应用于柯达数据集中的 24 幅图像所得的编码结果。

N	CV 内存. (K 字节)	Rel. 比特率对应的冗余（%）				
		0.25	0.50	1.00	1.50	2.50
256	51.6	0.00	0.00	0.00	0.00	0.01
192	38.7	0.00	0.00	0.00	0.01	0.01
128	25.8	0.04	0.04	0.04	0.03	0.03
96	19.4	0.03	0.04	0.04	0.04	0.05
64	12.9	0.13	0.12	0.12	0.12	0.11
48	9.7	0.24	0.23	0.22	0.22	0.20
32	6.5	0.38	0.38	0.40	0.41	0.42
24	4.9	0.85	0.81	0.81	0.82	0.79
16	3.4	1.79	1.76	1.76	1.78	1.75

- 对于所有比特率，冗余实际上都保持不变，这意味着所提出的编码方法实现了相对冗余保持恒定这一设计目标。
- 平均冗余水平与图 9 相符，表明理论预测的特征与实际实现相吻合。

表 1 的第二列给出了使用 16 位/元素时存储码向量所需的总存储量。这表明，所提出的方法可以直接用于减少存储器占用。例如，即使网络能够支持 8 位输出，且可以接受低于 0.5% 的冗余，也可以进一步将概率密度函数参数量化为 5 位，只需舍弃最低的 3 位有效位，即可将码向量存储器从 51.6 Kbytes 降至 6.5 Kbytes。

5. 结论

研究表明，可以通过评估编码冗余来分析量化对熵编码的影响，并据此调整待学习的概率密度函数参数，使平均相对冗余在实际应用中保持恒定。因此，更改概率密度函数参数的量化方式时无需重新训练网络；我们还表明，可以有意调整量化方式，以尽量减少码表所占用的存储器。

6. 参考文献

[1]S. Ma、X. Zhang、C. Jia、Z. Zhao、S. Wang 和 S. Wang，《基于神经网络的图像和视频压缩：综述》，IEEE Trans. Circuits Syst. Video Technol.，第 30 卷，第 6 期，第 1683–1698 页，2020 年 6 月，arXiv:1904.03567v2。

[2]D. Ding、Z. Ma、D. Chen、Q. Chen、Z. Liu 和 F. Zhu，“基于深度神经网络的视频压缩系统进展：综述与案例研究”，《IEEE 论文集》，第 109 卷，第 9 期，第 1494–1520 页，2021 年 3 月，arXiv:2101.06341v1。

[3]Qualcomm AI Research，“全球首款基于软件的神经视频解码器，可在商用智能手机上实时解码高清格式”，网址：
h
t
t
p
s://
www.qualcomm.com/news/onq/2021/06/17/worlds-first-software-based-neural-video-decoder-running-hd-format-real-time，2021 年 6 月。

[4] Qualcomm AI Research，“在移动设备上实时进行神经视频解码的演示”，网址：https://www.youtube.com/watch?v=WUnlSHenr08，2021 年 12 月。[5]J. Ascenso、P. Akyazi、F. Pereira 和 T. Ebrahimi，“基于学习的图像编码：早期方案综述与主观质量评估”，载于《SPIE 会议录 11353：成像应用光学、光子学与数字技术 VI》，2020 年 4 月，第 164–176 页。[6]M. Nagel、M. van Baalen、T. Blankevoort 和 M. Welling，“通过权重均衡和偏置校正实现无数据量化”，载于《IEEE/CVF 国际计算机视觉会议论文集》，2019 年 10 月，第 1325–1334 页，arXiv:1906.04721v3。[7]M. Nagel、M. Fournarakis、R. A. Amjad、Y. Bondarenko、M. van Baalen 和 T. Blankevoort，“神经网络量化白皮书”，arXiv:2106.08295v1，2021 年 6 月。[8] T. Liang、J. Glossner、L. Wang、S. Shi 和 X. Zhang，“深度神经网络加速中的剪枝与量化：综述”，《神经计算》，第 461 卷，第 370–403 页，2021 年 10 月，arXiv:2101.09671。[9]G. J. Sullivan、J.-R. Ohm、W.-J. Han 和 T. Wiegand，“高效视频编码（HEVC）标准概述”，《IEEE 电路与系统视频技术汇刊》，第 22 卷，第 12 期，第 1649–1668 页，2012 年 12 月。

[10] V. Sze 和 D. Marpe，“高效视频编码中的熵编码”，载于《高效视频编码（HEVC）：算法与架构》，V. Sze、M. Budagavi 和 G. J. Sullivan 编，第 8 章，第 209–274 页。Springer，2014 年。

[11] B. Bross、Y.-K. Wang、Y. Ye、S. Liu、J. Chen、G. J. Sulli- van 和 J.-R. Ohm，“多功能视频编码（VVC）标准概述及其应用”，IEEE 电路与系统视频技术汇刊，第 31 卷，第 10 期，第 3736–3764 页，2021 年 10 月。[12] P. Elias，“通用码字集与整数表示”，IEEE 信息论汇刊，第 21 卷，第 2 期，第 194–203 页，1975 年 3 月。[13] A. Said，“算术编码”，载于《无损压缩手册》，K. Sayood 编，第 5 章，第 101–152 页。学术出版社，圣迭戈，加利福尼亚州，2003 年。[14] A. Said，“算术编码导论——理论与实践”，技术报告 HPL-2004-76，惠普实验室，帕洛阿尔托，加利福尼亚州，美国，2004 年 4 月，（http://www.hpl.hp.com/techreports/2004/HPL-2004-76.pdf）。[15] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang 和 N. John- ston，“采用尺度超先验的变分图像压缩”，载于第六届国际学习表征会议，加拿大温哥华，2018 年 4 月，arXiv 预印本 arXiv:1802.01436v2。[16]J. Ball´e、N. Johnston 和 D. Minnen，“用于潜变量模型数据压缩的整数网络”，载于国际学习表征会议，美国路易斯安那州新奥尔良，2019 年 5 月，网址：https://openreview.net/forum?id=S1zz2i0cY7。[17] W. H. Press、S. A. Teukolsky、W. T. Vetterling 和 B. P. Flannery，《数值计算方法：科学计算的艺术》，剑桥大学出版社，英国剑桥，第三版，2007 年。[18] 柯达图像数据集，网址：http://www.cs.albany.edu/ xy-pan/research/snr/Kodak.html。