用于神经视频编解码增强的可微比特率估计

Amir Said 高通人工智能研究*圣迭戈，CA，美国
asaid@qti.qualcomm.com

Manish Kumar Singh 高通人工智能研究*圣迭戈，CA，美国
masi@qti.qualcomm.com

Reza Pourreza 高通人工智能研究*圣迭戈，CA，美国
pourreza@qti.qualcomm.com

arXiv:2301.09776v1 [eess.IV] 24 Jan 2023

摘要——神经网络（NN）可通过对编码视频进行预处理和后处理来提升标准视频压缩效果。为实现最佳神经网络训练，需要用编解码器代理替代标准编解码器，使其能够提供比特率估计值和失真的导数，以用于梯度反向传播。由于标准编解码器的熵编码旨在考虑变换系数之间的非线性依赖关系，因此简单的逐系数估计器无法准确近似比特率。本文提出一种新的比特率估计方法，该方法类似于端到端神经编解码器训练中采用的方法，并能够高效地考虑这些统计依赖关系。该方法基于数学模型定义，可提供估计值及其梯度的闭式公式，从而降低计算复杂度。实验结果表明，该方法能够准确估计高效视频编码（HEVC/H.265）编解码器的比特率。

索引词——视频编码、神经网络视频增强、比特率估计

I. 引言

在消费设备中，视频编解码器通常由定制硬件（专用集成电路）实现，这类硬件性能高，但灵活性较低，因为修改需要耗时且昂贵的重新设计和部署。

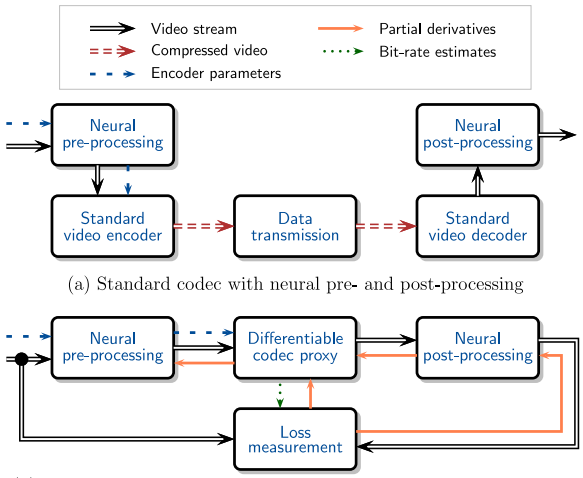
无需更改专用集成电路，即可通过在编码前和解码后处理视频来提升编解码器性能；如图 1(a) 所示，最新趋势是采用神经网络（NN）。应用示例包括去噪、伪影去除、分辨率变换等。 [1]–[3]。

采用端到端优化有望取得最佳效果，即训练神经网络时将编解码器参数和性能纳入考量。一个根本问题是，若能利用性能测量值的导数，神经网络训练会有效得多 [4], [5]，；但通过常见的标准视频编解码器实现无法直接获得这些导数。

解决方案是采用编解码器代理（例如 [6]–[9]），以准确估计性能因素及相应的导数，如图 1(b) 所示，从而实现神经网络梯度的反向传播 [10], [11]。在视频编码中，损失函数必须同时考虑最小化失真和比特率这两个相互冲突的目标。

对于失真估计，为训练端到端神经编解码器（EENC）开发的方法提供了良好的

*高通人工智能研究是高通技术公司的一项计划。



(a) Standard codec with neural pre- and post-processing

(b) Network training with codec proxy and gradient back-propagation

图 1. (a) 使用神经网络进行视频压缩增强的系统，以及 (b) 相应的网络训练。

可微近似，并可与用于估计主观质量的方法结合使用 [8], [9], [12]。

另一方面，为 EENC 开发的可微码率估计方法无法直接改用于标准编解码器，因为这些方法所处理的数据类型大不相同，因此在熵编码优化方面采用了截然不同的策略。

视频压缩的码率估计问题广为人知，因为码率估计是码率控制所必需的，而码率控制对于实际视频编码至关重要 [13], [14]。例如，若干方法使用基于量化器步长来估计比特率的模型 Q [15], [16]。

然而，这些方法大多旨在直接用于标准编解码器，因此可能不可微，也不适用于神经网络训练的条件；在这种条件下，需要在变换块的精细尺度上获得估计值，并计算逐像素导数。

像广为人知的 ρ域码率控制 [17],这类方法更适用于小规模估计，但修改为可微版本后（参见式 (1)），准确性会有所降低。

本文提出了一种码率估计方法，该方法旨在适用于标准编解码器所使用的数据，同时通过采用一种类似于神经网络编解码器训练中所用的数据建模方式，实现更高的准确度。我们证明

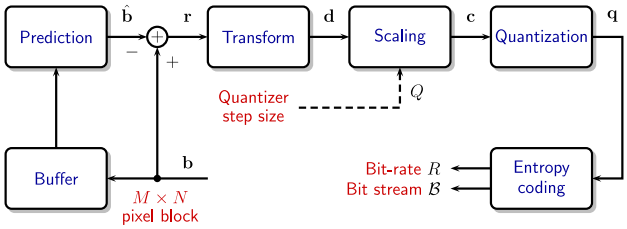


图 2. 标准编解码器中使用的混合视频编码简化示意图。

由于该方法基于数学表述，因此可以推导出估计值及其相应导数的闭式方程，从而提高计算效率并加快神经网络训练。

下一节将讨论为何先进的熵编码会使准确的码率估计变得困难；第三节将介绍所提出的统计模型及其与 EENC 中所用模型的相似之处。第四节将给出公式和计算方法，第五节则展示并讨论实验结果。

II. 视频编解码器中的熵编码

图 2 展示了标准视频编解码器所用混合编码架构的示意图，并介绍了本文档中使用的符号。在给定的编码阶段，会对一个包含 $M \times N$ 个像素的块进行预测，对预测残差向量 $\mathbf{r}$ 应用正交变换，将所得向量 $\mathbf{d}$ 除以量化器步长 Q，以获得缩放系数 $\mathbf{c}$，最后对其进行量化和熵编码。

在此符号体系中，所有向量的维数均为 $K = MN$；为与信号处理符号体系保持一致，所有向量和矩阵的索引均从零开始。

估算比特率的一种简单方法是将每个系数的估计值相加。例如， [8] 中使用的对 ρ域估计的可微近似为

$$\hat{R}_d(\mathbf{c}) = \mu \sum_{k=0}^{K-1} \log_2(1 + |c_k|), \tag{1}$$

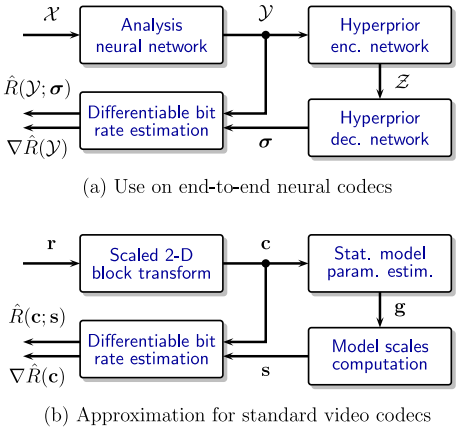
其中， μ 是根据 JPEG 比特率得出的一个因子。

这种方法的主要问题在于，变换系数并非单独编码，因为它们在统计上并不独立。事实上，利用幅值依赖关系可获得显著的压缩增益，这也促使人们采用相当复杂的熵编码形式 [18]–[21]。

因此，许多实现会使用标准规定的实际熵编码方法来估算比特率。虽然这种方法最为精确，但在将其改造成可微版本时，会遇到许多实际问题。

为了解决量化导数几乎处处为零的问题，人们开发了可微近似 [7], [8], [22]。虽然这些近似对失真估计很有用，但要将其用于基于量化值的复杂熵编码过程则困难得多。

例如，量化为零的系数通常会一起编码，或者通过指示最后一个非零



(a) Use on end-to-end neural codecs

(b) Approximation for standard video codecs

图 3. 基于统计模型的码率估计系统比较。

元素的位置来编码。非零值会被二值化，并可通过多次遍历进行编码，每个二进制符号使用不同的编码上下文 [19], [20]。

这些困难促使我们寻找更好的码率估计方法，这些方法基于相同的统计特性，但采用不同的方法论。

III. 基于模型的估计

图 3(a) 展示了使用超先验架构训练 EENC 的基本结构 [12]。该结构基于为非线性变换元素（量化后将进行熵编码）建立统计模型，该模型由分布类型（通常为高斯分布）以及包含标准差的数组 $\boldsymbol{\sigma}$ 定义。

这种方法可以自然地纳入许多数据元素之间的统计依赖关系，并将其转化为标准差值的变化；实验结果表明，该方法能够得到相当准确的码率估计值。

基于这些原因，我们提出了一种类似的方法，如图 3(b) 所示，其区别如下

- 它应用于标准编解码器所使用的正交变换（如离散余弦变换或正弦变换）的系数。
- 假设变换系数是服从拉普拉斯分布的零均值随机变量，其标准差由一个包含少量参数的向量 $\mathbf{g}$ 所定义的模型给出。
- 该统计模型基于一项经验观察：变换系数的方差往往会随频率 [23], 呈指数下降，下降速率取决于像素模式的方向。
- 对于每个块，使用所有系数值计算最大似然（ML）参数 $\mathbf{g}^*$，并使用该模型和 $\mathbf{g}^*$ 估算比特率和梯度。

IV. 实际实现

编解码器代理的实现涉及许多实际细节，超出了本文的讨论范围。如图 4 所示，我们假设图 2 所示的混合编码器的主要预测部分均采用近似处理。同样，

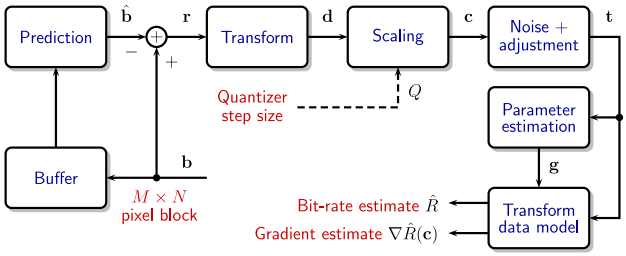


图 4. 提出的码率估计系统。

选择 Q 可以固定，也可以随机改变 [9], ，具体取决于训练目标。

主要策略如上一节所述，本节将介绍实际实现所需的近似，以及一些经验证能够提高准确性和数值稳定性的启发式方法。

为了使用与二维变换相关的索引，定义向量 m 和 n 如下

$$m_k \stackrel{\text{def}}{=} \lfloor k/N \rfloor, \quad n_k \stackrel{\text{def}}{=} k - N\, m_k, \tag{2}$$

并用 1 表示全 1 向量，定义以下 $K \times 3$ 矩阵

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{1} & \mathbf{m} & \mathbf{n} \end{bmatrix}. \tag{3}$$

A. 系数调整

为了近似常见的死区 [24] 或经 R-D 优化的 [25] 量化，首先使用图 5 所示的函数及其导数对变换系数进行“调整”，以减小较小的幅值

$$\begin{aligned} \psi(c) &\stackrel{\text{def}}{=} \frac{c^3}{c^2+\tau}, \\ \Psi(c) &\stackrel{\text{def}}{=} \frac{\mathrm{d}\psi(c)}{\mathrm{d}c} = 1 + \frac{\tau\left(c^2-\tau\right)}{\left(c^2+\tau\right)^2}. \end{aligned} \tag{4}$$

B. 噪声添加

添加均匀噪声，以避免所有系数均为零或非常小时出现数值不稳定。给定一个由均匀分布随机变量 $\eta_k \sim U(-\epsilon, \epsilon)$ 组成的数组，我们定义向量 t 和 w，用于估计

$$t_k \stackrel{\text{def}}{=} \psi(c_k), \quad w_k \stackrel{\text{def}}{=} |t_k + \eta_k|. \tag{5}$$

C. 概率分布模型

为估计模型参数，假设向量 w 的元素服从指数概率分布，且其标准差按照三维参数向量 g 随频率呈指数衰减，如下所示

$$\sigma_k(\mathbf{g}) = \exp(-[g_0 + m_k g_1 + n_k g_2]). \tag{6}$$

为简化符号表示，我们定义标准差倒数构成的向量

$$s_k(\mathbf{g}) \stackrel{\text{def}}{=} 1/\sigma_k(\mathbf{g}) = \exp(g_0 + m_k g_1 + n_k g_2), \tag{7}$$

以获得概率分布函数

$$f(w_k; s_k(\mathbf{g})) = s_k(\mathbf{g}) e^{-s_k(\mathbf{g}) w_k}. \tag{8}$$

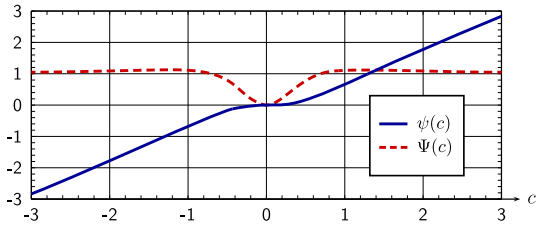


图 5。 用于变换系数调整的函数及其 导数。

D. 最大似然参数

由式（8）中的分布定义的似然函数为

$$\mathcal{L}(\mathbf{g}) = \prod_{k=0}^{K-1} f(w_k; s_k(\mathbf{g})), \tag{9}$$

且对数似然的相反数为

$$L(\mathbf{g}) \stackrel{\text{def}}{=} -\log(\mathcal{L}(\mathbf{g})) = \mathbf{w}^{\mathsf{T}}\mathbf{s}(\mathbf{g}) - \mathbf{1}^{\mathsf{T}}\mathbf{A}\mathbf{g}. \tag{10}$$

用 $\circ$ 表示逐元素向量乘法，并考虑到梯度

$$\nabla L(\mathbf{g}) = \mathbf{A}^{\mathsf{T}}\left[\mathbf{w} \circ \mathbf{s}(\mathbf{g}) - \mathbf{1}\right], \tag{11}$$

以及 3×3 对称海森矩阵

$$\mathbf{H}(\mathbf{g}) = \mathbf{A}^{\mathsf{T}} \mathrm{diag}[\mathbf{w} \circ \mathbf{s}(\mathbf{g})]\, \mathbf{A}, \tag{12}$$

易于计算，例如，可以通过牛顿迭代求得最大似然解

$$\mathbf{g} \leftarrow \mathbf{g} - [\mathbf{H}(\mathbf{g})]^{-1} \nabla L(\mathbf{g}), \tag{13}$$

若实现得当，应能 [26], 收敛到最优解$\mathbf{g}^*$。

请注意， 3×3 对称矩阵求逆或乔列斯基分解都可以轻松计算。

E. 码率估计

给定最大似然概率分布参数 $\mathbf{s}^* \stackrel{\text{定义}}{=} \mathbf{s}(\mathbf{g}^*)$，我们可以采用为端到端神经编解码器开发的技术 [12] ，在假设调整后的参数t_k 服从拉普拉斯概率分布且其累积分布函数形式如下的情况下，获得可微的比特率估计值

$$F(t; s) = \begin{cases} \frac{1}{2} e^{st}, & t < 0, \\ 1 - \frac{1}{2} e^{-st}, & t \geq 0. \end{cases} \tag{14}$$

以及参数 s_k^*。由于式 (5) 中添加了噪声，这在数学上并不精确，但作为一种近似方法很方便。

量化变换系数的可微估计概率为

$$p_k = F(t_k + 1/2; s_k^*) - F(t_k - 1/2; s_k^*), \tag{15}$$

并根据熵方程估算比特率

$$\hat{R}(\mathbf{c}) = -\frac{\alpha}{K} \sum_{k=0}^{K-1} \log_2(p_k), \tag{16}$$

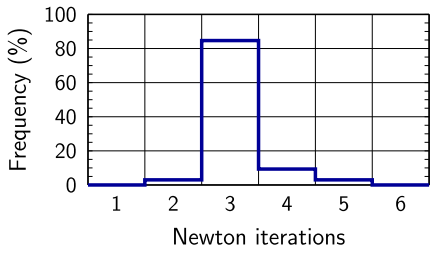


图 6. 达到高精度所需迭代次数的分布。

其中添加乘法因子 α 进行校准，与公式 (1) 中的参数 μ 类似。

请注意，在训练期间，将比特率与失真相加之前，必须先乘以一个系数。该系数取决于训练目标，最优值只能通过验证测试来确定。

例如，实验测试可以表明，对于 H.264/AVC， α 的某个值最为合适，而对于 H.265/HEVC，则另一个值更为合适。主要目标是确保估计值的一致性，从而根据视频特征正确地做出设计选择。

F. 偏导数计算

由于式 (16) 推导过程中的所有阶段都是可微的，因此可以使用自动微分 [10], [11]有效且轻松地计算 ˆgradient ∇R(c)。

然而，使用牛顿迭代来确定 $\mathbf{g}^*$ 需要创建向量序列 $\mathbf{g}^{(0)}$、$\mathbf{g}^{(1)}$、. ..，这会在梯度反向传播过程中增加额外的计算。

利用模型公式的数学性质，可以省去这些计算。可以证明，定义函数

$$\begin{aligned}\gamma(k,\delta) &\stackrel{\text{def}}{=} \frac{\alpha s_k^* \exp(-s_k^*|t_k+\delta|)}{2\ln(2)Kp_k}, \\ \phi(k,\delta) &\stackrel{\text{def}}{=} (t_k+\delta)\,\gamma(k,\delta),\end{aligned} \tag{17}$$

和向量

$$\begin{aligned}u_k &\stackrel{\text{def}}{=} \gamma(k,1/2)-\gamma(k,-1/2), \\ v_k &\stackrel{\text{def}}{=} \phi(k,1/2)-\phi(k,-1/2), \\ y_k &\stackrel{\text{def}}{=} \Psi(c_k), \\ z_k &\stackrel{\text{def}}{=} \operatorname{sign}(t_k+\eta_k)\,s_k^*,\end{aligned} \tag{18}$$

后，可以直接使用方程计算比特率梯度，效率也更高

$$\nabla\hat{R}(\mathbf{c}) = \mathbf{y}\circ\left[\mathbf{z}\circ\left(\mathbf{A}\left[\mathbf{H}(\mathbf{g}^*)\right]^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{v}\right)-\mathbf{u}\right]. \tag{19}$$

请注意，尽管 ∇ˆR(c) 的每一项都取决于 c 的所有元素，但通过高效计算中间结果，可以将计算复杂度从 O(M^2N^2) 降低至 O(MN)；此外，还可以轻松优化实现，并对向量和矩阵运算进行并行化。

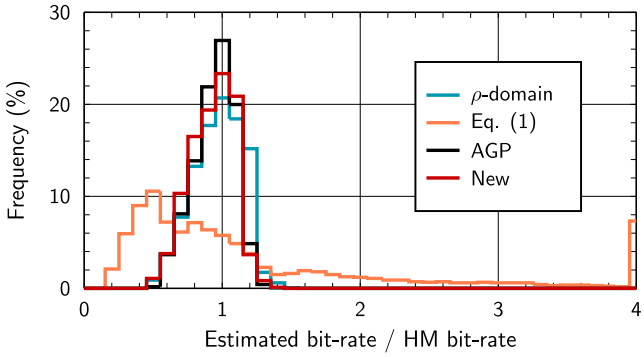


图 7. 不同可微和不可微方法所估计的比特率与 HM 比特率之间比率的直方图。

V. 实验结果

所提出的方法经过测试，用于估计 H.265/HEVC 编解码器的比特率。HM 16.20 参考实现 [27]经过修改，以输出块残差的离散余弦变换（DCT），并将所得估计值与每帧实际使用的比特数进行了比较。

实验采用低延迟 P 配置，对 VVC 标准化测试视频中的 A1、A2、B 和 E 类视频进行测试，量化参数为 = 22、27、32 和 37。为保持一致，所有 14 个视频均转换为 1280×720 高清分辨率，每个视频 250 帧，共测试 14,000 帧。

所有测试方法的估计都仅使用亮度分量。由于篇幅有限，这里仅报告 8×8 块的结果，该块采用了强制变换尺寸。

该方法使用式（4）中的 $\tau=0.4$ 实现，并使用 $\eta=0.05$ 生成均匀噪声（其规模显著小于 EENC 训练所用的噪声）。

在所有测试中，初始解均为 $g_1=g_2=0.05$，并且

$$g_0 = -\ln\left(\frac{1}{K}\sum_{k=0}^{K-1} w_k\, e^{g_1 m_k + g_2 n_k}\right). \tag{20}$$

图 6 展示了使用该初始化时牛顿迭代次数（13）的观测分布。可以看出，在大多数情况下，仅需 3 次迭代即可达到足够高的精度。

图 7 展示了若干估计方法得到的比特率与实际 HM 比特率之比的直方图。在这类图中，理想的估算器应使 100% 的比率集中在 1 附近。所有测试方法均使用针对四个量化参数值优化的校准系数，以衡量其在较宽比特率范围内的准确性。

使用 AGP 方法 [28]进行基于上下文的熵编码（一种较简单的熵编码方法）可获得最佳结果，而使用 ρ 域估算器 [17] 得到的结果稍差一些。不过，这些都是不可微估算器。

所提出的方法在 AGP 和ρ域之间实现了准确度，同时具有可微性。另一方面，式 (1) [8], 中的可微估算器性能明显较差（请注意，约有 8% 的比率

表 I 根据 HM 量化参数得出的估计比率标准差。

QP	平均值	估计方法			
	比特率	ρ-域	AGP	公式 (1)	所提出的
22	0.257	0.158	0.121	0.817	0.126
27	0.120	0.131	0.125	0.708	0.129
32	0.059	0.149	0.131	0.678	0.147
37	0.030	0.252	0.222	0.709	0.243
All	—	0.179	0.156	0.730	0.168

实际上超出了量程（大于 4），这表明所有形式的逐系数估计都存在缺陷。

在高比特率设置下，码率估计更容易；通过测量不同量化参数下测得比特率与实际比特率之比的标准差，可以观察到这一点，如表 I 所示。我们可以看到，尽管平均比特率相差约一个数量级，但总体趋势与图 7 中观察到的相同。

AGP 在所有比特率下都能提供最准确、最稳定的结果，所提出的方法紧随其后；准确度主要在较低比特率（量化参数 = 37）时有所下降。ρ 域估算器的稳定性稍逊，而逐系数估算器在所有测试中的标准差都显著更大。

VI. 结论

实验结果证实了第三节所提出方法的优势

- 当比特率估计像熵编码方法那样考虑变换系数幅值之间的统计依赖关系时，其精确度会大幅提高。

- 采用统计模型，并通过标准差分布定义幅值依赖关系（与训练端到端神经编解码器时所用的方法相同），可大幅提高估计精度。

- 采用恰当的数学表述，可以直接计算估计值及其导数，并降低计算复杂度。

参考文献

[1] S. Ma、X. Zhang、C. Jia、Z. Zhao、S. Wang 和 S. Wang，“神经网络图像和视频压缩：综述”，IEEE电路与系统视频技术汇刊，第30卷，第6期，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”，Proc. IEEE，第109卷，第9期，1494–1520页，2021年3月，arXiv:2101.06341v1。[3] S. Anwar、S. Khan 和 N. Barnes，“超分辨率探索之旅：一项调查”，ACM计算调查，第53卷，第3期，1–34页，2020年，arXiv:1904.07523v3。[4] L. Bottou，“随机梯度下降的大规模机器学习”，载于 COMPSTAT'2010 会议论文集，2010年，177–186页，施普林格。[5] I. Goodfellow、Y. Bengio 和 A. Courville，《深度学习》，麻省理工学院出版社，2016年。

[6] K. Qiu、L. Yu 和 D. Li，“用于联合优化视频编码及其预处理与后处理的编解码器仿真网络”，IEEE Open J. Circuits Systems，第2卷，648–659页，2021年。[7] X. Luo、H. Talebi、F. Yang、M. Elad 和 P. Milanfar，“码率–失真–准确度权衡：JPEG案例研究”，2020年。[8] O. G. Guleryuz、P. A. Chou、H. Hoppe、D. Tang、R. Du、P. Davidson 和 S. Fanello，“夹层式图像压缩：在标准编解码器周围封装神经网络”，载于 IEEE 国际图像处理会议论文集，美国阿拉斯加州安克雷奇，2021年9月。[9] A. Chadha 和 Y. Andreopoulos，“面向视频编码的深度感知预处理”，载于计算机视觉与模式识别会议论文集，2021年6月。[10] A.G. Baydin、B.A. Pearlmutter、A.A. Radul 和 J.M. Siskind，“机器学习中的自动微分：综述”，《机器学习研究杂志》，第18卷，1–43页，2018年。[11] A. Paszke、S. Gross、S. Chintala、G. Chanan、E. Yang、Z. DeVito、Z. Lin、A. Desmaison、L. Antiga 和 A. Lerer，“PyTorch中的自动微分”，载于神经信息处理系统会议（NIPS）研讨会，2017年。[12] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang 和 N. Johnston，“基于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。[13] B. Li、H. Li、L. Li 和 J. Zhang，“面向高效视频编码的λ 域码率控制算法”，IEEE图像处理汇刊，第23卷，第9期，3841–3854页，2014年。[14] A.A. Ramanand、I. Ahmad 和 V. Swaminathan，“高效视频编码和可扩展高效视频编码视频编码中的码率控制综述”，载于2017年IEEE国际多媒体与博览会研讨会，2017年，145–150页。[15] S. Ma、W. Gao 和 Y. Lu，“H.264/AVC视频编码的率失真分析及其在码率控制中的应用”，IEEE电路与系统视频技术汇刊，第15卷，第12期，1533–1544页，2005年。[16] D.-K. Kwon、M.-Y. Shen 和 C.-C. J. Kuo，“采用增强码率与失真模型的H.264视频码率控制”，IEEE电路与系统视频技术汇刊，第17卷，第5期，517–529页，2007年。[17] Z. He 和 S.K. Mitra，“通过 ρ域源建模实现视频编码的最优比特分配和精确码率控制”，IEEE电路与系统视频技术汇刊，第12卷，849，2002年10，840期，第页–。[18] I. E. Richardson，《H.264高级视频压缩标准》，约翰·威立父子出版公司有限公司，英国西萨塞克斯，第二版，2010年。[19] V. Sze 和 M. Budagavi，“高效视频编码中的高吞吐量基于上下文的自适应二进制算术编码熵编码”，IEEE电路与系统视频技术汇刊，第22卷，第12期，1778–1791页，2012年12月。[20] 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，209–274页。施普林格，2014年。[21] M. Wien，《高效视频编码：编码工具与规范》，施普林格出版社，柏林，2015年。[22] R. Shin 和 D. Song，“抗JPEG对抗图像”，载于2017年NIPS机器学习与计算机安全研讨会，2017年。[23] A. Said 和 D. Mukherjee，“利用数据幅值模式实现高效图像编码”，载于IEEE国际图像处理会议论文集，美国加利福尼亚州圣迭戈，2008年10月，2900–2903页。[24] G. J. Sullivan 和 S. Sun，“关于死区加均匀阈值标量量化”，载于国际光学工程学会第5960卷：视觉通信与图像处理会议论文集，中国北京，2005年7月。[25] M. Karczewicz、P. Chen、Y. Ye 和 R. L. Joshi，“H.264中的基于率失真的量化”，载于国际光学工程学会第7443卷：数字图像处理应用会议论文集，美国加利福尼亚州圣迭戈，2009年9月。[26] 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。[27] “HEVC HM参考软件”，https://vcgit.hhi.fraunhofer.de/jvet/HM。[28] A. Said 和 W. A. Pearlman，“通过字母表和样本集划分实现低复杂度波形编码”，载于国际光学工程学会第3024卷：视觉通信与图像处理会议论文集，美国加利福尼亚州圣何塞，1997年2月，25–37页。