IEEE 国际多媒体会议 W 机器编码研讨会，澳大利亚布里斯班 ，2023。

© 2023 IEEE。允许个人使用本材料。对于所有其他用途，包括在任何当前或未来媒体中重印/再版本材料以用于广告或宣传目的、创作新的汇编作品、转售或重新分发至服务器或列表，或在其他作品中重复使用本作品的任何受版权保护部分，均须获得 IEEE 的许可。

arXiv:2305.02562v2 [eess.IV] 4 Jul 2023

用于人类与机器可伸缩编码的条件与残差方法

Anderson de Andrade、Alon Harell、Yalda Foroutan 和 Ivan V. Baji´c

工程科学学院，西蒙弗雷泽大学，加拿大本拿比{anderson de andrade,alon harell,yalda foroutan}@sfu.ca,ibajic@ensc.sfu.ca

摘要

我们介绍了面向人类与机器的可伸缩编码中的条件编码和残差编码方法。我们重点利用计算机视觉任务中可用的信息，优化重建任务的率失真性能。我们对这两种方法都进行了信息分析，以提供基线；此外，我们还提出了一种适用于条件编码的熵模型，其建模容量更高，同时可处理性与先前工作相近。我们将这些方法应用于图像重建：一个实例使用在 Cityscapes 数据集上为语义分割创建的表征，另一个实例使用在 COCO 数据集上为目标检测创建的表征。在两项实验中，条件方法与残差方法取得了相近的性能，所得率失真曲线均处于我们的基线范围内。

索引词——可学习压缩、可伸缩编码、条件编码、残差编码、熵建模

1. 引言

随着人工智能日益普及，数字内容不仅供人类使用，也供计算机程序使用。这些软件通常会根据自身目的，以不同方式分析内容。根据任务的不同，可能只需使用可用信息的一个子集。此外，所需信息可以用一种更适合计算机程序的方式来表示，而这种表示未必类似于人类消费此类内容时通常需要的原始自然表示。

在一种协作场景中， [1], 边缘设备采集信号，并将其处理后传输至云服务，以完成一组任务。为了高效地完成这些任务，只需传输必要的信息即可。针对每个任务子集分别创建表示，难以随着任务数量的增加而实现良好的扩展。此外，如果某些任务所需的信息已经传输，而现在又需要针对相同输入完成原始任务集合的一个超集，那么传输新的相应表示将因冗余信息而产生额外开销。因此，我们希望以可扩展的方式组合任务所需的信息， [2], 以这些基础表示由多个任务共享，而对于更具体的任务，只需要增量增加少量信息。

创建能够利用不同信息流、并针对不同用途进行适配的可学习任务是一项挑战 [3]。特定任务所诱导的低维流形，未必能直接用于另一项任务。将表征从一个流形转换到另一个流形，以便在次要任务中尽可能多地利用信息，受到变换的建模能力和可用数据的限制 [4, 5]。在可学习压缩场景中，条件编码和残差编码一直是引入边信息的两种不同方法。这些方法可以利用专门的可学习变换，将信息显式传递到目标域。

我们的研究结果仅限于一种常见设置：我们有一个图像重建任务，以及一个与前者共享表示的计算机视觉任务。这种配置称为面向人类和机器的可扩展图像编码 [3]。我们提出了用于可扩展可学习压缩的条件方法和残差方法，在这些方法中，我们变换表示，使其共享一个共同的特征空间。我们为这些方法推导出基线，并通过实验进行比较。我们的实验使用语义图像分割和目标检测的表示，对不同数据集进行重建。我们还提出了一种建模能力更强、适用于条件编码的熵模型。

2. 相关工作

在可学习压缩中，输入与输出之间的中间表示上会形成一个信息瓶颈 [6]。成功的方法遵循变分框架 [7] ，其中，超先验表示学习潜在表示中不同因子之间的依赖关系，并作为边信息 [8, 9, 10, 11]发挥作用。通过使用自回归熵模型来形成信息瓶颈，并对学习得到的表征进行熵编码。

近期关于面向人类和机器的可伸缩编码的研究，将可学习压缩的思想应用于重建和计算机视觉任务 [12, 3, 13]。在

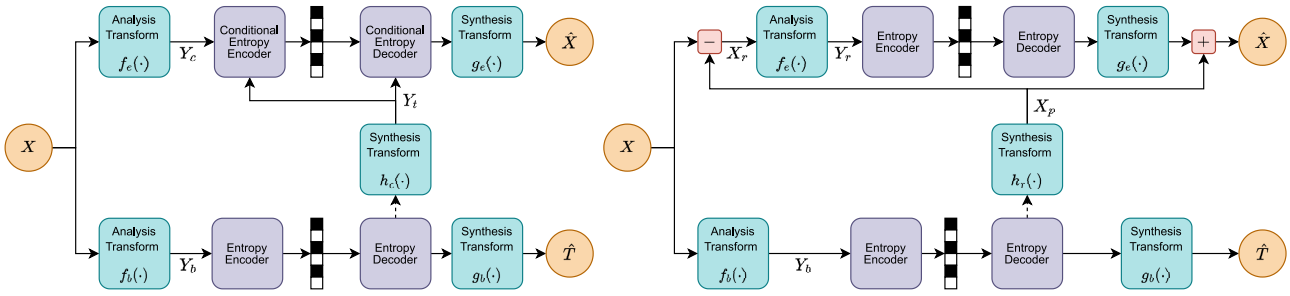


图1：残差方法和条件方法的整体架构。虚线表示增强网络不会影响基础网络。条件熵解码器对 $H(Y_c|Y_t)$ 进行建模。

这些方法中，重建任务使用专用表示和共享表示。解码后，这些表征会被拼接起来，并作为重建模型的输入。通过率失真优化，这种方法可以生成彼此独立、信息冗余较少的表示，但 [3] 和 [13] 的结果显示存在相当大的冗余。本研究主要关注如何高效地对重建任务的专用表示进行编码。在条件方法中，专用表示包含与重建相关的全部信息，但编码时会利用共享表示所消除的不确定性。在残差方法中，编码前先从目标表示中移除共享表示的信息，解码后再在处理流程中将其加回。

在可学习压缩领域，先前已有研究探索过视频压缩中的残差编码和条件编码 [14, 15, 16, 17]。我们的公式不同之处在于，预测完全由原始信号解释，因此残差的信息量不会增加；而在视频压缩中，由于使用前一帧来计算预测，这种情况可能发生。 [16], 中表明，可学习的条件编码通常需要对边信息进行变换，这可能导致一定程度的信息损失，以至于残差方法可能优于条件方法。在这项工作中，我们提出在两种方法中都变换边信息，并展示这种做法如何提升残差方法相对于条件方法的性能。

现有许多用于可学习压缩的熵模型都支持在给定超先验表示 [8] 的条件下，对目标表示进行条件编码。近期的熵模型通过将多个因子分组 [18], , 并重新组织和并行化空间位置的解码顺序 [19], 进一步扩展了这些方法，以高效捕获这些表示在空间和维度上的依赖关系。在本研究中，我们以类似的方式利用条件信息，但通过扩大其感受野并添加尺度化残差连接，提升了其建模容量。

3. 提出的方法

对于输入图像 $X \in \mathbb{R}^{C_x \times H \times W}$，学习有损压缩的基础表示 $Y_b = f_b(X)$，以最小化相对于给定计算机视觉目标 T、任务失真函数 $d_b(\cdot,\cdot)$ 和可学习解码函数 $g_b(\cdot)$ 的失真 $D_b = \mathbb{E}_X[d_b(g_b(Y_b)\ T)]$。

在条件设置下，学习有损压缩的增强表示 $Y_c = f_e(X)$，以最小化$\widehat{\ }$ $\widehat{\ }$失真 $D_c = \mathbb{E}[d_e(X, X)]$; $X = g_e(Y_c)$，使用图像重建失真函数 $d_e(\cdot,\cdot)$ 和可学习解码器 $g_e(\cdot)$。重建任务所使用的全部信息都包含在 Y_c 中，而 Y_b 中包含的信息则用于高效编码 Y_c。条件编码有效地建模了 $H(Y_c|Y_t)$，其中 $Y_t = h_c(Y_b)$ 是对 Y_b 的可学习变换，直观上具有与增强表示 Y_c 相似的特征空间，因此可以利用二者之间的相似性。任何能够降低条件熵的信息都应保留在 Y_t 中，因为其码率不会受到惩罚。

在残差方法中，创建了类似的表示$Y_r = f_e(X_r)$; $X_r = X - X_p$; $X_p = h_r(Y_b)$，以最小化 $D_r = \mathbb{E}[d_e(g_e(Y_r) + X_p, X)]$。这里， $h_r(\cdot)$ 是对 Y_b 的可学习变换，可隐式地重建图像。预测 X_p 在重建过程结束时被加上。图1展示了两种配置的架构图。[1]

3.1. 条件编码的界限

我们的理论分析在无损情况下进行，以便为我们有损方法提出的基线提供依据。在条件编码中，我们将 $H(Y_b) + H(Y_c|Y_t)$ 建模为具有 $H(Y_c)$ 的下界：

$$\begin{aligned} H(Y_c) &\le H(Y_c) + H(Y_t|Y_c) = H(Y_c, Y_t) \\ &= H(Y_t) + H(Y_c|Y_t) \le H(Y_b) + H(Y_c|Y_t), \end{aligned} \quad (1)$$

[1]官方代码发布：https://github.com/adeandrade/research

其中，我们根据数据处理不等式得出 $H(Y_t) \leq H(Y_b)$。当 $H(Y_t|Y_c) = 0$ and$H(Y_b|Y_t) = 0$时，此界是紧的；等价地，当 $H(Y_b) = I(Y_c; Y_t)$ 时，此界也是紧的。这对应于 $H(Y_c|Y_t)$ 中的信息减少了 $H(Y_b)$。上界可通过以下方式得到：

$$\begin{aligned} H(Y_b) + H(Y_c|Y_t) &= H(Y_b) + H(Y_c) - I(Y_c; Y_t) \\ &\leq H(Y_b) + H(Y_c). \end{aligned} \tag{2}$$

当 $I(Y_c; Y_t) = 0$时，此界是紧的，这对应于 Y_c 和 Y_t 相互独立。

我们通过使用独立的增强表示 Y_e （其生成不依赖任何边信息），并测量 $\hat{}H(Y_e)$，为条件方法提供一个上基线，其中 $\hat{}H(\cdot)$ 是熵估计。作为下基线，我们使用 $\hat{}H(Y_b) + \hat{}H(Y_e)$。这基于以下考虑： Y_e 作为任务表示可能比 Y_c 更高效，以及式 (1) 和 (2) 中的界。

3.2. 残差编码的界限

已证明，条件编码是残差编码的一个上界 [16, 17]:

$$\begin{aligned} H(X|X_p) &= H(X_r + X_p|X_p) = H(X_r|X_p) && (3) \\ &= H(X_r) - I(X_p; X_r) \leq H(X_r) && (4) \\ &= H(X|X_p) + I(X_p; X_r). && (5) \end{aligned}$$

这里，(3) 利用了这样一个事实：在观察到 X_p 后，$X_r + X_p$ 中唯一的不确定性来自 X_r。(4) 中的不等式利用了互信息的非负性。(5) 中使用互信息的定义对 $H(X_r)$ 进行了改写，并再次利用了这样一个事实：给定 X_p 后，$X - X_p$ 中唯一的不确定性来自 X。

式（4）中的项 $I(X_p; X_r)$ 对残差形式起到惩罚项的作用。为了使其最小化，残差 X_r 与预测 X_p 必须尽可能相互独立。当 X_p 将 X 中的数值合并，从而使 $H(X_r)$ 减小时，便可实现这一点；或者，当 X_p 针对 X中的不同数值产生恒定值，从而降低 $H(X_p)$ 时，也可实现这一点。降低 $H(X_p)$ 会增大 $H(X|X_p)$，进而可能产生不利影响，使 $H(X_r)$增大，如式（5）所示。

在我们提出的方法中，我们通过训练来最小化 D_r 和 $\hat{}H(Y_r)$。由此，我们也最小化 $\hat{}H(X_r)$。如式（5）所示，这会同时降低 $H(X|X_p)$ 和 $I(X_p; X_r)$。因此，这一优化过程促使可学习函数$h_r(\cdot)$创建表示 X_p ，尽可能准确地还原输入 X，同时使其与由此产生的残差 X_r尽可能相互独立。

需要注意的是，强制要求 X_p 与原始输入 X 相似可能并非最佳做法，因为尽管这种优化会降低 $H(X|X_p)$，却可能导致 $I(X_p; X_r)$上升。这解释了为什么在初步实验中，我们发现采用函数 $h_r(\cdot)$显式重建 X ，其表现不如我们提出的方法。

基于我们提出的方法所涉及的上述考量，我们发现实现 $H(Y_c|Y_t) = H(Y_r)$ $\hat{}$可能更容易。这促使我们将 $H(Y_b) + \hat{}H(Y_r)$ 与条件方法中使用的相同基线进行比较。

3.3. 熵建模

为了对表示 Y_c 进行条件编码，以尽可能利用来自 Y_t 的信息，我们使用 CNN [18]对表示之间及表示内部的空间和维度依赖关系进行建模。我们提出的熵模型在复杂性和准确性之间取得了平衡。我们将通道分组，每组大小固定为 K [18]。在每个组内，跨通道的相同位置使用前面各组中的所有位置，以及当前组所有通道中位于当前接收野内的所有先前位置作为上下文，并行处理。与 [8], 类似，空间域中的位置按照从上到下、从左到右的方式处理。通过对卷积核应用掩码来强制执行马尔可夫性质。图 2a 展示了某层单个输出通道的卷积核掩码。

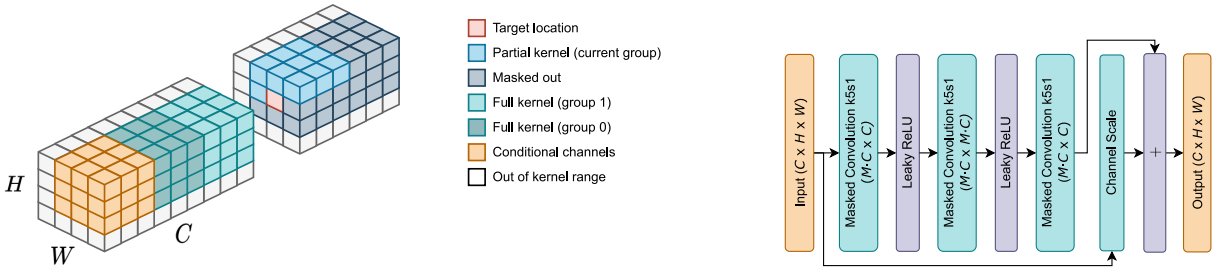
与以往的研究不同，我们的熵模型采用了具有可扩展残差连接的 CNN 架构，并在自回归卷积中使用了更深的层以及尺寸大于 1 的卷积核。这消除了类似熵模型所受到的一些建模限制。CNN 架构由多个包含三层的块组成，其中输入通道数首先增加，在更高维的空间中进行变换，然后再缩减回原始通道数。这些块之间引入了残差连接，使得输入可以沿通道维度进行不同程度的重新缩放。为了在通道数发生变化时保持马尔可夫性质，组大小会相应地重新缩放，并且通道数只能以 M 的倍数变化。图 2b 展示了 CNN 中单个块的架构概览。

条件表示可作为另一组通道使用，并以与其他组相同的方式由 CNN 进行变换，但其上下文仅限于该组内的感受野。条件表示中的所有元素都可以访问这一组。

与 [8] 和较新的研究类似，熵模型的预测对应于赋予表示中每个元素的单变量高斯分布的均值 W 和尺度 Σ 。符号通过 $Q = \lfloor Y - W \rceil$ 获得，相应的概率为 $P_{\mathcal{N}(0,\Sigma)}[|Q| \trianglelefteq |Q| \pm 1/2]$。训练期间，通过添加均匀噪声 $\mathcal{U}(-1/2, 1/2)$ 来模拟取整操作。

3.4. 可学习的可扩展压缩

作为可学习压缩的架构，我们使用了 [11] 中工作的简化版本。我们去除了边信息



(a) 卷积掩码　　(b) CNN 块

图 2：熵模型概览。(a) 卷积核大小为 3×3，输入为 $12 \times 4 \times 4$。条件输入大小为 $3 \times 4 \times 4$，共有 $K = 4$ 组。采用输入填充且步幅为 1 时，这是卷积的第 7 步。

组件，该组件在 [8] 中作为超先验引入编码器。我们还移除了 [10] 中引入的注意力层。为了减小内存占用并加快训练过程，我们逐步减少分析器前几层和合成器最后几层中的通道数量。

基础任务和增强任务具有相同的架构。基础任务生成的表示与输入 X 具有相同的维度和分辨率，但不强制重建输入。该表示是计算机视觉模型的输入。编码器和计算机视觉模型通过端到端方式联合训练。

在条件方法中，$h_c(\cdot)$ 是一种传统 CNN，由通过残差连接相连的块组成。每个块包含三个卷积层，用于在更高维度上执行变换，并将输出缩放回较低维度。前半部分的块维度与 Y_b 相同，后半部分则过渡到与 Y_c 相同的维度，从而得到 Y_t。由于 Y_b 和 Y_c 的分辨率相同，整个网络的分辨率保持不变。在残差方法中，$h_r(\cdot)$ 使用与我们的合成器相同的架构，将 Y_b 变换为 X_p。合成器会对表示进行上采样，使其分辨率和维度与 X 相匹配。

针对 D_b 优化的表示 Y_b 可能包含一些单独来看对重建没有帮助的信息。此外，Y_b 中的信息以适合计算机视觉任务 T 的方式表示，而通过 $h_r(\cdot)$ 将其全部带回图像特征空间可能颇具挑战。为克服这些障碍，我们在率失真拉格朗日最小化公式中，对经过变换的 Y_b 添加一个较小的重建惩罚项：

$$\mathcal{L}_b = D_b + \lambda_b \hat{H}(Y_b) + \beta\, \mathbb{E}[d_e(\hat{h}_r(Y_b), X)], \qquad (6)$$

其中，$h_r(\cdot)$ 是一个辅助网络，其架构与其他合成器相同，λ_b 和 β 是超参数。对于增强表示，我们使用传统的率失真损失函数 [6]：

$$\mathcal{L}_c = D_c + \lambda_e \hat{H}(Y_c|Y_t), \qquad \mathcal{L}_r = D_r + \lambda_r \hat{H}(Y_r),$$

分别对应条件方法和残差方法。在训练过程中，基础网络要么保持冻结状态，要么重建网络产生的梯度不会反向传播至基础网络。

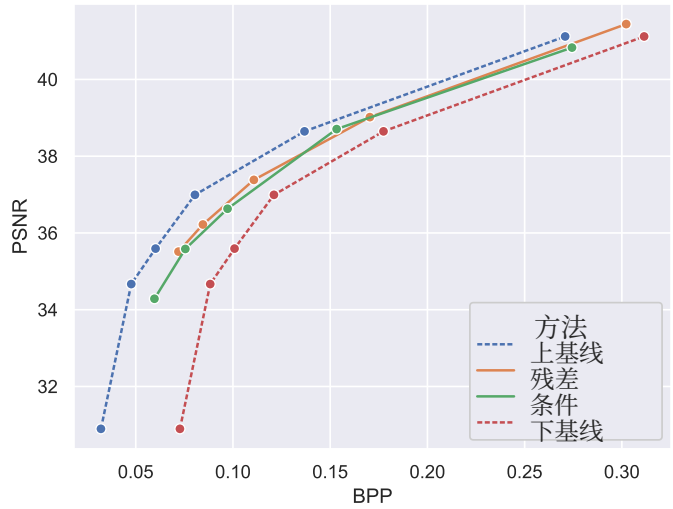
4. 实验

我们进行了实验，以分析所提出的可伸缩编码条件方法和残差方法的率失真性能，并将其与所提出的基线进行比较。我们开展了两组实验：一组以语义分割作为计算机视觉任务，使用 Cityscapes 数据集 [20]，；另一组以目标检测作为计算机视觉任务，使用 COCO 2017年数据集 [21]。

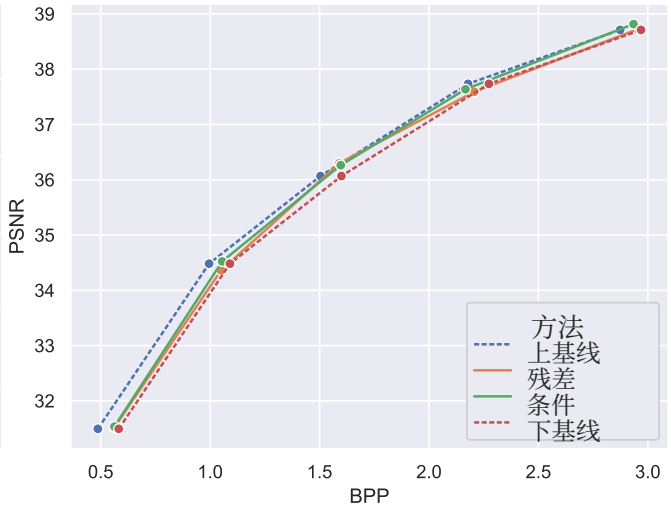
我们首先针对计算机视觉任务训练基础表示，以获得不同 λ_b取值下的率失真点 $f_b(\cdot)$ 和 $g_b(\cdot)$。我们选择率失真曲线上一个失真程度合理的点所对应的模型，随后用它生成条件方法和残差方法的表示 Y_b。上基线通过在没有边信息的情况下训练重建任务来构建。分析变换和合成变换采用与其他模型相同的架构。上基线的熵模型与残差方法所用的熵模型具有相同的架构。下基线则通过加入条件方法和残差方法所用基础表示的码率来获得。

在所有实验中，我们为基础表示分配了 $C_b = 32$ 个通道，为增强表示分配了 $C_e = 256$ 个通道。在分析变换中，四个下采样操作的输出通道数分别为 24、48、192 和 C_b 或 C_e，而合成变换中的上采样操作的输出通道数分别为 192、48、24 和 C_x。熵模型由 5 个块组成，包含 $K = 16$ 和 $M = 1$。

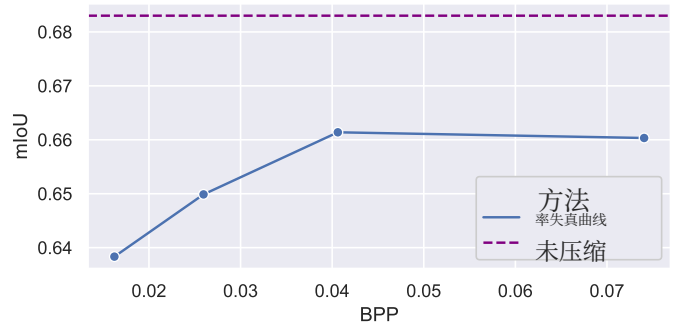
在所有实验中，为了训练重建任务，我们使用均方根误差函数作为失真函数 $d_e(\cdot,\cdot)$。我们根据熵估计计算并报告每像素比特数（BPP）；在若干实验中，该数值与实际达到的每像素比特数相差不超过 0.5%。此外，为了加快率失真曲线的计算，我们通常会训练一个模型



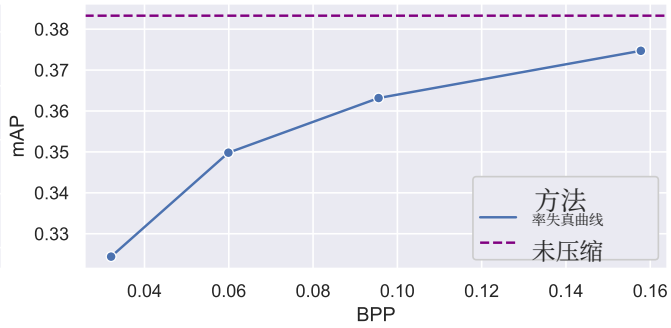
(a) Cityscapes 的可伸缩编码率失真曲线



（b）COCO 的可伸缩编码率失真曲线



(c) Cityscapes 上语义分割的率失真曲线



(d) COCO 上目标检测的率失真曲线

图 3：可伸缩编码结果。紫色线条表示使用 $\lambda_b = 0$ 和 $\beta = 0$时达到的性能。

该模型采用低压缩设置，并将其权重用作训练其他模型以获得曲线其余部分的初始值。使用 Adam 以10^{-4}的学习率更新参数。我们采用提前停止机制训练模型，但如果训练进入平台期，会先将学习率降低至原来的 0.75 倍。

4.1. Cityscapes 上的图像语义分割

Cityscapes 是一组用于理解城市场景的图像 [20]。我们使用 DeepLabV3 [22] 作为语义分割的计算机视觉模型，并以 MobileNetV3 [23] 作为后端。此处， $d_b(\cdot,\cdot)$表示逐像素多类别交叉熵，但我们报告平均交并比（ mIoU）指标。我们设置 $\beta = 0.1$ ，并报告验证集上的结果。数据增强方面，我们使用随机裁剪的 768×768 像素图像、随机水平翻转和颜色抖动。与编码器对应的模型前端使用 Adam 进行训练，学习率为 10^{-4}；分类器则使用随机梯度下降进行训练，并采用动量和 10^{-2}的学习率。我们在分类器的权重上添加 ℓ_2 损失，以防止过拟合，缩放因子为 10^{-4}。

图 3a 展示了条件方法和残差方法的率失真曲线。我们注意到，这些曲线位于各基线之间，所得率失真点也符合这一关系。与下基线的率失真曲线相比，条件方法的 BD-Rate[24] 为 -16.56%，而残差方法实现了 -14.6%的码率降低。因此，就 BD-Rate 而言，条件方法的表现略优于残差方法。通过比较这些 BD-Rate 得分与上基线的 BD-Rate 得分，我们可以计算出基础表示的利用率。因此，条件方法使用了 43.01% 的边信息码率，而残差方法使用了 37.91%。在两种方法中压缩程度最低的设置下，利用率更高。

图3c展示了基础任务的率失真性能。所选的 β 值会对码率和任务性能施加惩罚，但允许架构利用基础表示。我们认为，这条率失真曲线中的细微不完美之处源于 β 的选择以及训练算法的局限性。

4.2. COCO 上的目标检测

COCO 2017年包含 123,287 张用于目标检测和分割的领域无关图像 [21]。我们使用 Faster R-CNN [25]进行目标检测，并以 ResNet-50 [26] 作为后端。对于此任务，$d_b(\cdot,\cdot)$ 是该架构所采用的不同损失函数之和。我们报告根据 [21], 和$\beta = 0.05$计算得出的平均精度均值（ mAP）指标。在训练的数据预处理中，我们采用随机水平翻转，并按照相似的宽高比将图像分为 3 个簇，以生成批次。在训练计算机视觉任务时，将一个批次中的图像调整为最小尺寸，并相应地调整边界框。在进行重建训练时，将一个批次中的图像中心裁剪至其最小尺寸。所有权重均使用 Adam 进行训练，学习率为 10^{-4}。

如图 3b 所示，两种方法的性能相当；与下基线相比，条件方法和残差方法的 BD-Rate 分别提升了 -4.14 % 和 -2.47%。就 BD-Rate 而言，条件方法和残差方法对基础任务的利用率分别达到 49.24% 和 29.32%。图 3d 展示了基础任务的基础失真性能。与 Cityscapes 上的语义分割相比，该任务模型更能接近未压缩时的任务性能。此外，在失真惩罚相近的情况下，该任务使用的码率更高。COCO 数据集上重建任务得到的率失真曲线，比 Cityscapes 上的曲线大近一个数量级。这可以归因于 Cityscapes 中图像内容较为简单，以及压缩导致 COCO 数据集中出现了更多伪影。

5. 结论

我们提出了面向人类与机器的可扩展编码中的条件方法和残差方法。实验表明，所提出的条件编码与残差编码架构表现相近，且其率失真性能处于所给出的基线或运行界限范围内。此外，所提出的条件熵模型能够达到与残差方法相当的性能。

6. 参考文献

[1] I. V. Baji´c、W. Lin 和 Y. Tian，“协同智能：挑战与机遇”，载于 IEEE ICASSP，2021 年，第 8493–8497 页。[2] H. Schwarz、D. Marpe 和 T. Wiegand，“H.264/AVC 标准可扩展视频编码扩展概述”，IEEE TCSVT，第 17 卷，第 9 期，第 1103–1120 页，2007 年。[3] H. Choi 和 I. V. Baji´c，“面向人类和机器的可扩展图像编码”，IEEE TIP，第 31 卷，第 2739–2754 页，2022 年。[4] B. J. Culpepper 和 B. A. Olshausen，“学习图像流形的传输算子”，载于 NeurIPS，2009 年，第 423–431 页。

[5] M. Connor、G. Canal 和 C. Rozell，“具有学习潜在结构的变分自编码器”，发表于 AISTATS，2021年，第 130 卷，第 2359–2367 页。[6] N. Tishby、F. C. N. Pereira 和 W. Bialek，“信息瓶颈方法”，CoRR，第 physics/0004057 卷，2000 年。[7] D. P. Kingma 和 M. Welling，“自编码变分贝叶斯”，发表于 ICLR，2014 年。[8] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang 和 N. Johnston，“采用尺度超先验的变分图像压缩”，发表于 ICLR，2018 年。[9] D. Minnen、J. Ball´e 和 G. Toderici，“用于学习式图像压缩的联合自回归先验与分层先验”，发表于 NeurIPS，2018 年，第 10794–10803 页。[10] Z. Cheng、H. Sun、M. Takeuchi 和 J. Katto，“采用离散化高斯混合似然和注意力模块的学习式图像压缩”，发表于 IEEE CVPR，2020 年，第 7936–7945 页。[11] D. He、Z. Yang、W. Peng、R. Ma、H. Qin 和 Y. Wang，“ELIC：采用不均匀分组空间通道上下文自适应编码的高效学习式图像压缩”，发表于 IEEE CVPR，2022 年，第 5708–5717 页。[12] H. Choi 和 I. V. Baji´c，“面向多任务协同智能的潜在空间可扩展性”，发表于 IEEE ICIP，2021 年，第 3562–3566 页。[13] E. Ozyilkan、M. Ulhaq、H. Choi 和 F. Racape，“用于面向人类和机器的可扩展图像编码的学习式解耦潜在表示”，arXiv 2301.04183，2023。[14] A. Djelouah、J. Campos、S. Schaub-Meyer 和 C. Schroers，“用于视频编码的神经帧间压缩”，发表于 IEEE ICCV，2019 年，第 6420–6428 页。[15] T. Ladune、P. Philippe、W. Hamidouche、L. Zhang 和 O. D´eforges，“用于灵活学习式视频压缩的条件编码”，发表于 ICLR，2021 年。[16] F. Brand、J. Seiler 和 A. Kaup，“从信息论角度看条件帧间视频编码的优势与挑战”，发表于 PCS，2022 年，第 289–293 页。[17] H. Hadizadeh 和 I. V. Baji´c，“ LCCM-VC：用于视频编码的学习式条件编码模式”，arXiv 2210 15883，2022。。[18] D. Minnen 和 S. Singh，“用于学习式图像压缩的逐通道自回归熵模型”，发表于 IEEE ICIP，2020 年，第 3339–3343 页。[19] D. He、Y. Zheng、B. Sun、Y. Wang 和 H. Qin，“用于高效学习式图像压缩的棋盘格上下文模型”，发表于 IEEE CVPR，2021 年，第 14771–14780 页。[20] M. Cordts、M. Omran、S. Ramos、T. Rehfeld、M. Enzweiler、R. Benenson、U. Franke、S. Roth 和 B. Schiele，“用于城市场景语义理解的 Cityscapes 数据集”，发表于 IEEE CVPR，2016年，第 3213–3223 页。[21] T.-Y. Lin、M. Maire、S. J. Belongie、J. Hays、P. Perona、D. Ra-manan、P. Doll´ar 和 C. L. Zitnick，“Microsoft COCO：情境中的常见对象”，发表于 ECCV，2014 年，第 740–755 页。[22] L.-C. Chen、G. Papandreou、F. Schroff 和 H. Adam，“重新思考用于语义图像分割的空洞卷积”，CoRR，第 abs/1706.05587 卷，2017年。[23] A. Howard、R. Pang、H. Adam、Q. V. Le、M. Sandler、B. Chen、W. Wang、L.-C. Chen、M. Tan、G. Chu、V. Vasudevan 和 Y. Zhu，“探索 MobileNetV3”，发表于 IEEE ICCV，2019 年，第 1314–1324 页。

[24] G. Bjontegaard，“率失真曲线之间平均峰值信噪比差异的计算”，国际电信联盟电信标准化部门 SC16/Q6 VCEG-M33，2001年。[25] S. Ren、K. He、R. B. Girshick 和 J. Sun，“更快的区域卷积神经网络：借助区域建议网络实现实时目标检测”，IEEE模式分析与机器智能汇刊，第 39 卷，第 6 期，第 1137–1149 页，2017年。[26] K. He、X. Zhang、S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，发表于 IEEE CVPR，2016年，第 770–778 页。