Coordinate Attention for Efficient Mobile Network Design

Qibin Hou[1] Daquan Zhou[1] Jiashi Feng[1]
[1]National University of Singapore
{andrewhoux,zhoudaquan21}@gmail.com

本文提出 **Coordinate Attention，CA**，可以插入到 Mobile Network 中，可以使 MobileNetV2、EfficientNet 等网络涨点，性能优于 SE、CBAM 等注意力模块，同时还可以提高检测、分割任务的性能，代码即将开源！ **作者单位**：南洋理工大学

1 简介

Mobile Network 设计的最新研究成果表明，**通道注意力**（例如，SE 注意力）对于提升模型性能具有显著效果，但它们通常会忽略位置信息，而位置信息对于生成空间选择性 attention maps 是非常重要。

因此在本文中，作者通过将位置信息嵌入到通道注意力中提出了一种新颖的移动网络注意力机制，将其称为“**Coordinate Attention**”。

与通过 2 维全局池化将特征张量转换为单个特征向量的通道注意力不同，coordinate 注意力将通道注意力分解为两个 1 维特征编码过程，分别沿 2 个空间方向聚合特征。这样，可以沿一个空间方向捕获远程依赖关系，同时可以沿另一空间方向保留精确的位置信息。然后将生成的特征图分别编码为一对方向感知和位置敏感的 attention map，可以将其互补地应用于输入特征图，以增强关注对象的表示。

本文所提的 Coordinate 注意力很简单，可以灵活地插入到经典的移动网络中，例如 MobileNetV2，MobileNeXt 和 EfficientNet，而且几乎没有计算开销。大量实验表明，Coordinate 注意力不仅有益于 ImageNet 分类，而且更有趣的是，它在下游任务（如目标检测和语义分割）中表现也很好。

2 相关工作

2.1 Mobile Network

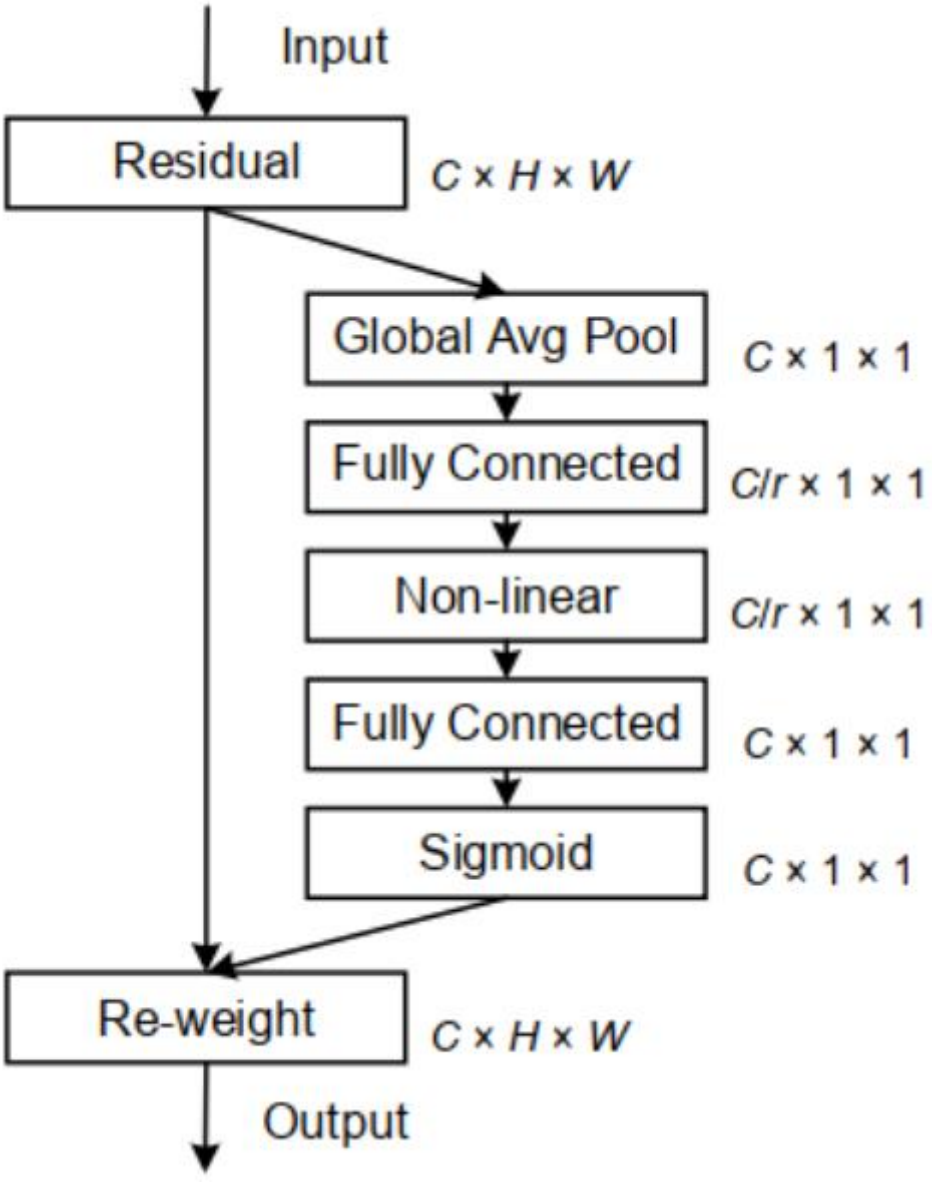
最近的很多关于 Mobile Network 的工作大多数都是基于**深度可分离卷积**和 **inverted 残差模块**：

- **HBONet**：在每个 inverted 残差模块中引入下采样操作，用于建模具有代表性的空间信息。
- **ShuffleNetV2**：在 inverted 残差模块之前和之后使用通道分割模块和通道 shuffle 模块。
- **MobileNetV3**：结合神经网络结构搜索算法，寻找最优激活函数和不同深度的 inverted 残差块的扩展比。

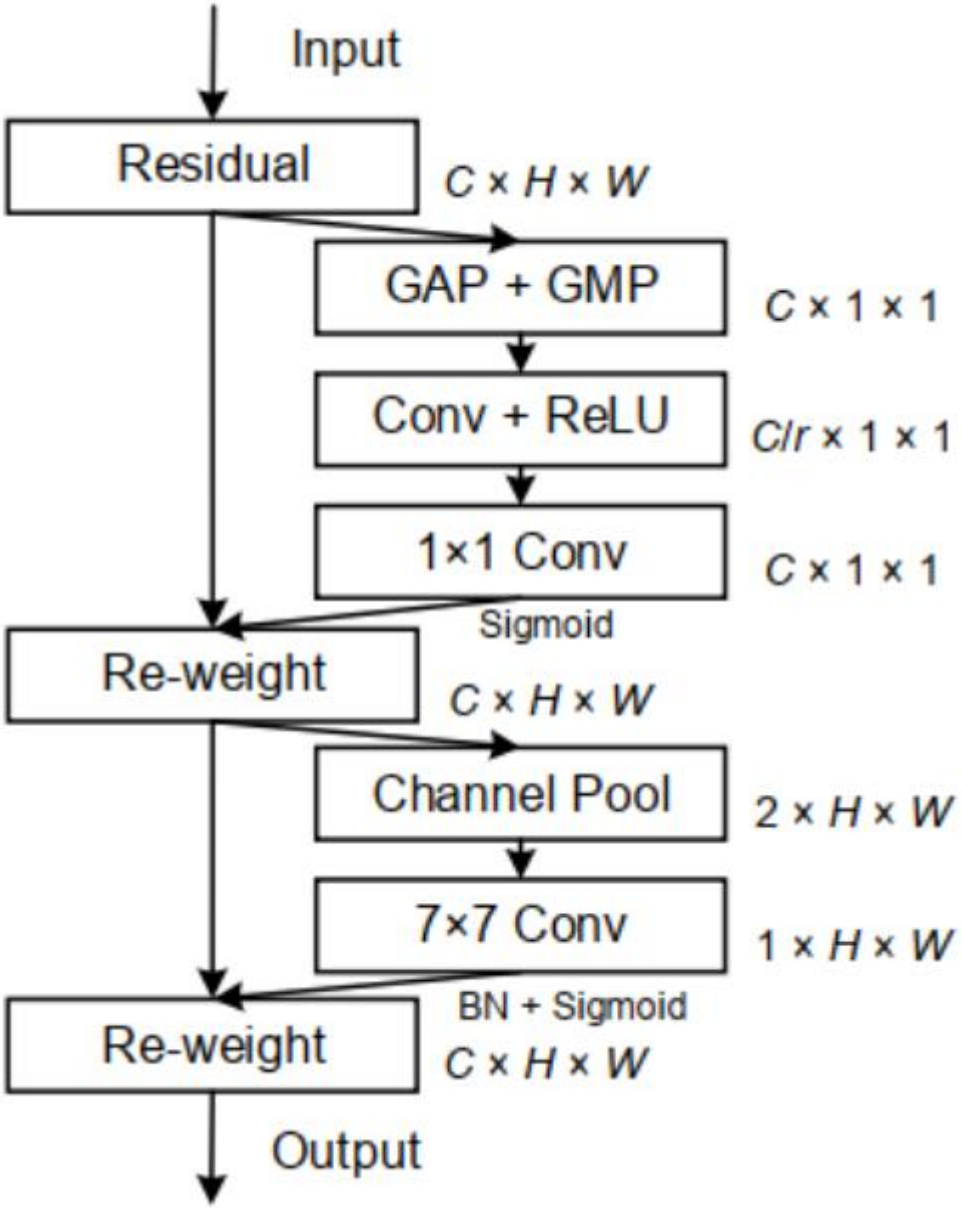
- **MixNet、EfficientNet 和 ProxylessNAS**：也采用不同的搜索策略来搜索深度可分卷积的最优核大小或标量，从而从扩展比、输入分辨率、网络深度和宽度等方面控制网络权值。
- 最近，有学者重新思考了基于深度可分离卷积的方法，专门设计了基于 Mobile Network 的 bottleneck 结构，并基于此设计 **MobileNeXt**。

2.2 注意力机制

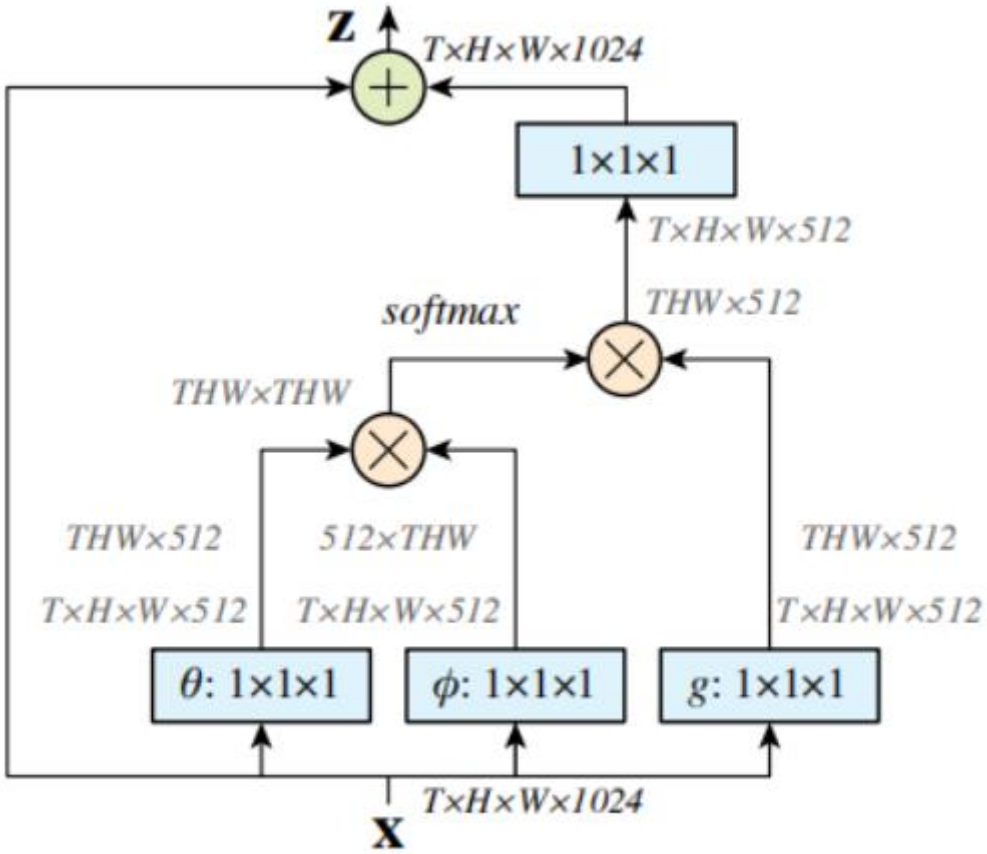
想必大家都已经知道注意力机制在各种计算机视觉任务中都是有帮助，如图像分类和图像分割。其中最为经典和被熟知的便是 **SENet**，它通过简单地 squeeze 每个 2 维特征图，进而有效地构建通道之间的相互依赖关系。



CBAM 进一步推进了这一思想，通过大尺度核卷积引入空间信息编码。后来的研究如 GENet、GALA、AA、TA，通过采用不同的空间注意力机制或设计高级注意力块，扩展了这一理念。



Non-local/self-attention Network 则着重于构建 spatial 或 channel 注意力。典型的例子包括 NLNet、GCNet、A2Net、SCNet、gsopnet 和 CCNet，它们都利用 Non-local 机制来捕获不同类型的空间信息。然而，由于 self-attention 模块内部计算量大，常被用于大型模型中，不适用于 Mobile Network。



A spacetime **non-local block.**

与 Non-local/self-attention 的方法不同，CA 方法考虑了一种更有效的方法来捕获位置信息和通道关系，以增强 Mobile Network 的特征表示。通过将二维全局池操作分解为两个一维编码过程，本文方法比其他具有轻量级属性的注意力方法(如 SENet、CBAM 和 TA)运行得更好。

3 Coordinate Attention

一个 coordinate attention 块可以被看作是一个计算单元，旨在增强 Mobile Network 中特征的表达能力。它可以将任何中间特征张量作为输入并通过转换输出了与张量具有相同 size 同时具有增强表征的。为了更加清晰的描述 CA 注意力，这里先对 SE block 进行讨论。

3.1 Revisit SE Block

在结构上，SE block 可分解为 **Squeeze** 和 **Excitation** 2 步，分别用于全局信息嵌入和通道关系的自适应 Re-weight。

Squeeze

在输入的条件下，第通道的 squeeze 步长可表示为:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j),$$

式中，是与第通道相关的输出。

输入来自一个固定核大小的卷积层，因此可以看作是局部描述符的集合。Sqeeze 操作使模型收集全局信息成为可能。

Excitation

Excitation 的目的是完全捕获通道之间的依赖，它可以被表述为：

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \sigma(\hat{\mathbf{z}}),$$

其中为通道乘法，为激活函数，为变换函数生成的结果，公式如下:

$$\hat{\mathbf{z}} = T_2(\mathrm{ReLU}(T_1(\mathbf{z}))).$$

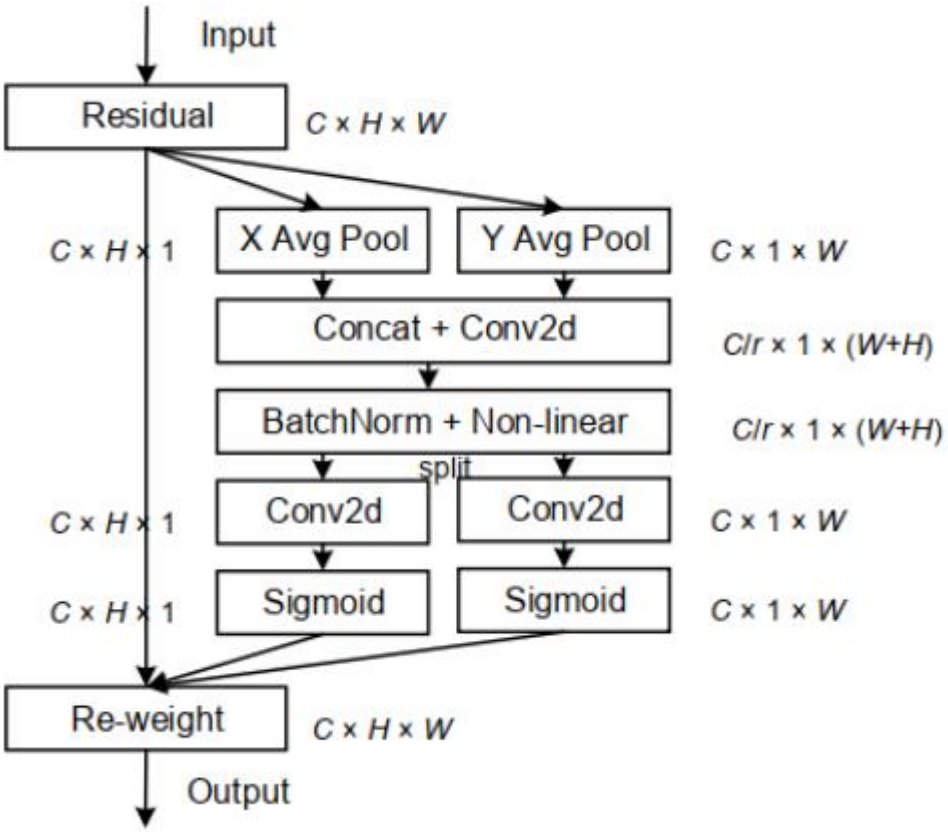
这里，和是 2 个线性变换，可以通过学习来捕捉每个通道的重要性。

为什么 SE Block 不好？

SE Block 虽然近 2 年来被广泛使用；然而，它只考虑通过建模通道关系来重新衡量每个通道的重要性，而忽略了位置信息，但是位置信息对于生成空间选择性 attention maps 是很重要的。因此作者引入了一种新的注意块，它不仅仅考虑了通道间的关系还考虑了特征空间的位置信息。

3.2 Coordinate Attention Block

Coordinate Attention 通过精确的位置信息对通道关系和长期依赖性进行编码，具体操作分为 **Coordinate 信息嵌入**和 **Coordinate Attention 生成** 2 个步骤。



3.2.1 Coordinate 信息嵌入

全局池化方法通常用于通道注意编码空间信息的全局编码，但由于它将全局空间信息压缩到通道描述符中，导致难以保存位置信息。为了促使注意力模块能够捕捉具有精确位置信息的远程空间交互，本文按照以下公式分解了全局池化，转化为一对一维特征编码操作：

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j)$$

具体来说，给定输入，首先使用尺寸为(H,1)或(1,W)的 pooling kernel 分别沿着水平坐标和垂直坐标对每个通道进行编码。因此，高度为的第通道的输出可以表示为:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i).$$

同样，宽度为的第通道的输出可以写成:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w).$$

上述 2 种变换分别沿两个空间方向聚合特征，得到一对方向感知的特征图。这与在通道注意力方法中产生单一的特征向量的 SE Block 非常不同。这 2 种转换也允许注意力模块捕捉到沿着一个空间方向的长期依赖关系，并保存沿着另一个空间方向的精确位置信息，这有助于网络更准确地定位感兴趣的目标。

3.2.2 Coordinate Attention 生成

通过 3.2.1 所述，本文方法可以通过上述的变换可以很好的获得全局感受野并编码精确的位置信息。为了利用由此产生的表征，作者提出了第 2 个转换，称为 **Coordinate Attention 生成**。这里作者的设计主要参考了以下 3 个标准：

- **首先**，对于 Mobile 环境中的应用来说，新的转换应该尽可能地简单；
- **其次**，它可以充分利用捕获到的位置信息，使感兴趣的区域能够被准确地捕获；
- **最后**，它还应该能够有效地捕捉通道间的关系。

通过信息嵌入中的变换后，该部分将上面的变换进行 concatenate 操作，然后使用卷积变换函数对其进行变换操作：

$$\mathbf{f} = \delta(F_1([\mathbf{z}^h, \mathbf{z}^w])),$$

式中为沿空间维数的 concatenate 操作，为非线性激活函数，为对空间信息在水平方向和垂直方向进行编码的中间特征映射。这里，是用来控制 SE block 大小的缩减率。然后沿着空间维数将分解为 2 个单独的张量和。利用另外 2 个卷积变换和分别将和变换为具有相同通道数的张量到输入，得到:

$$\mathbf{g}^h = \sigma(F_h(\mathbf{f}^h)),$$
$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w)).$$

这里是 sigmoid 激活函数。为了降低模型的复杂性和计算开销，这里通常使用适当的缩减比(如 32)来减少的通道数。然后对输出和进行扩展，分别作为 attention weights。

最后，Coordinate Attention Block 的输出可以写成:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j).$$

3.2.3 CA Block 的 PyTorch 实现

```python
import torch
from torch import nn



class CA_Block(nn.Module):
    def __init__(self, channel, h, w, reduction=16):
        super(CA_Block, self).__init__()

        self.h = h
        self.w = w

        self.avg_pool_x = nn.AdaptiveAvgPool2d((h, 1))
        self.avg_pool_y = nn.AdaptiveAvgPool2d((1, w))

        self.conv_1x1 = nn.Conv2d(in_channels=channel, out_channels=channel//reduction, kernel_size=1, stride=1, bias=False)

        self.relu = nn.ReLU()
        self.bn = nn.BatchNorm2d(channel//reduction)

        self.F_h = nn.Conv2d(in_channels=channel//reduction, out_channels=channel, kernel_size=1, stride=1, bias=False)
        self.F_w = nn.Conv2d(in_channels=channel//reduction, out_channels=channel, kernel_size=1, stride=1, bias=False)

        self.sigmoid_h = nn.Sigmoid()
        self.sigmoid_w = nn.Sigmoid()

    def forward(self, x):

        x_h = self.avg_pool_x(x).permute(0, 1, 3, 2)
        x_w = self.avg_pool_y(x)
```

```
        x_cat_conv_relu = self.relu(self.conv_1x1(torch.cat((x_h, x_w), 3)))

        x_cat_conv_split_h, x_cat_conv_split_w = x_cat_conv_relu.split([self.h, self.w], 3)

        s_h = self.sigmoid_h(self.F_h(x_cat_conv_split_h.permute(0, 1, 3, 2)))
        s_w = self.sigmoid_w(self.F_w(x_cat_conv_split_w))

        out = x * s_h.expand_as(x) * s_w.expand_as(x)

        return out



if __name__ == '__main__':
    x = torch.randn(1, 16, 128, 64)    # b, c, h, w
    ca_model = CA_Block(channel=16, h=128, w=64)
    y = ca_model(x)
    print(y.shape)
```

复制

4. 实验

4.1 消融实验

当水平注意力和垂直注意力结合时得到了最好的结果，如表 1 所示。实验结果表明，Coordinate Information Embedding 在图像分类中，可以在保证参数量的情况下提升精度。

4.2 与其他 Attention 进行比较

可以看出，添加 SE attention 已经使分类性能提高了 1%以上。对于 CBAM 与 SE 注意相比，似乎在 Mobile Network 中没有提升。然而，当使用本文所提出的 CA 注意力时，取得了最好的结果。

在上图中，作者还将使用不同注意力方法的模型生成的特征图进行了可视化。显然，CA 注意力比 SE 和 CBAM 更有助于目标的定位。

4.3 Stronger Baseline

为了检验所提 CA 注意力在 EfficientNet 上的表现，作者简单地用 CA 注意力代替 SE。对于其他设置遵循原始文件。结果如表 5。与原有的含 SE 的 EfficientNet-b0 方法以及其他与 EfficientNet-b0 的方法相比，CA 注意力络获得了最好的结果。这也所提出的 CA 注意力在强大的 Mobile Network 中仍然具有良好的性能。

4.4 目标检测实验

作者通过实验观察到 SE 和 CBAM 并不能改善 Baseline 的性能。然而，增加 CA 注意力可以很大程度上提高平均 AP 从 71.7%到 73.1%。在 COCO 和 Pascal VOC 数据集上的检测实验都表明，与其他注意力方法相比，具有 CA 注意力的分类模型具有更好的迁移能力。

4.5 语义分割实验

从表 8 可以看出，具有 CA 注意力的模型比 vanilla MobileNetV2 使用其他注意力的模型的表现要好得多。