arXiv:2304.02098v2 [cs.CV] 8 Sep 2024

深度学习中全景分割的不确定性估计

Michael Smith 和 Frank Ferrie

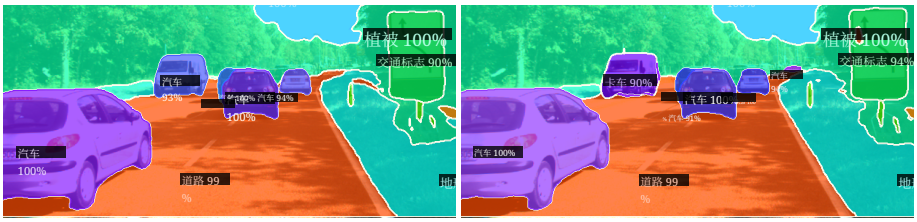
智能机器中心，麦吉尔大学，加拿大魁北克省蒙特利尔
michael.smith6@mail.mcgill.ca

摘要 随着基于深度学习的计算机视觉算法不断推进现有技术水平，它们对真实世界数据的鲁棒性仍然是一个问题，这使得将算法从实验室带到现实世界变得困难。基于集成的不确定性估计方法（如蒙特卡洛Dropout）已在许多应用中成功使用，以试图解决这一鲁棒性问题。不幸的是，并不总是清楚此类基于集成的方法是否可以应用于新的问题领域。全景分割就是这种情况，其问题结构和为解决它而设计的架构意味着，与图像分类甚至语义分割不同，使用样本均值的典型解决方案无法直接应用。在本文中，我们展示了如何在全景分割领域使用基于集成的不确定性估计方法（如蒙特卡洛Dropout），而无需对现有网络进行任何更改，从而既提高了性能，更重要的是为网络做出的预测提供了更好的不确定性度量。结果在COCO、KITTI-STEP和VIPER数据集上以定量和定性方式得到了证明。

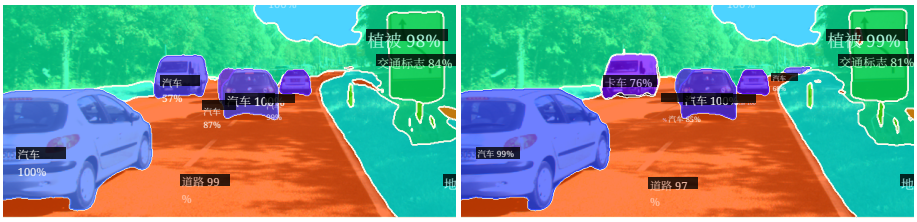
1 引言

分割，也就是解析图像内容，数十年来一直是计算机视觉领域的一项任务，应用广泛 [24, 33, 43, 48]。 [19] 近期提出了全景分割，这项任务将语义分割与实例分割（见补充材料附录 A）统一起来，可视为对较早分割任务的一种现代化重构，其目标始终不变：生成连贯的场景解读。更具体地说，全景分割要求为图像中的每个像素指派两个彼此一致的标签：类别标签（来自语义分割任务）和实例标签（来自实例分割任务）。类别标签说明对象是什么（汽车、天空、建筑物），实例标签则区分可计数的对象（例如汽车）。这一一致性要求使全景分割有别于同时进行语义分割与实例分割的多任务场景，在后者中，标签指派可能相互冲突。与其他分割任务一样，全景分割的主要用途是为更高层级的处理流程提供输入，例如车辆的自动驾驶导航。

据我们所知，目前所有全景分割算法都使用神经网络（这是由任务和数据集的复杂性所决定的），但它们遗憾地存在一个共同问题：预测可能出错，而且缺乏可靠的置信度度量来评估这些预测。图 1a 展示了这一点：在单帧画面中，我们看到模型对迎面驶来的车辆的判断，从非常确信它是汽车，变为非常确信它是卡车。这会让人难以依赖这些预测。人们已尝试为神经网络引入此类度量 [12],，其中包括集成；蒙特卡洛 Dropout [11] 便是一个例子。



(a) 基线方法。



(b) 我们的方法，使用 15 个蒙特卡洛样本。

图1： KITTI-STEP [51] 数据集中的两个连续帧（为清晰起见进行了裁剪）。请注意，对迎面驶来的车辆的识别如何从汽车变为卡车，以及我们的方法如何更好地捕捉预测中的不确定性。

通常，集成样本由模型参数的变化生成。均值等运算符可用于生成单个预测（消除噪声），也可以使用香农熵 [45] 来识别预测中的不确定性。一般而言，样本之间存在分歧（或不存在分歧）都可以为自动驾驶等更高阶的处理过程提供有用信号。要应用上述运算符，通常需要假设样本彼此匹配，即它们代表同一对象。对于某些任务，默认满足这一条件，例如图像分类或语义分割；在这些任务中，逐像素类别图提供了隐式对应关系 [15]。在我们的工作中，我们研究了某些类型的全景分割网络，它们会生成一组固定的预测。要确定每个样本中的各项预测是否代表同一对象，我们需要先计算预测之间的对应关系，然后才能应用均值等运算符。其他领域确实存在实现这一过程的方法

领域 [35,51]中会使用匈牙利方法 [21],等算法，但针对全景分割的直接实现并不显而易见。

因此，在本文中，我们提出了一种高效且定义明确的解决方案，用于解决对应问题，使集成能够应用于全景分割领域中选定的一组方法。这提供了更可靠的置信度度量，使需要此类置信度度量的高阶处理流程得以应用。图 1b 给出了一个简单示例：我们的方法能够累积多个样本（此处为蒙特卡洛样本），从而更好地捕捉预测中的不确定性。在多个数据集上，与基线或其他方法相比，我们的方法通常能提升分割性能。更重要的是，它能够通过预测熵（公式（3））等函数，从多个样本中提取有用的不确定性信息。这使自动驾驶等下游任务能够更好地利用全景分割结果，并根据需要将不确定性纳入考量。

2 相关工作

全景分割 [19] 在近期的数据集、挑战赛和框架中仍广受关注 [10, 28, 29]。由于所有像素都需要类别标签（例如 天空），但只有部分像素需要实例 ID（例如 汽车），全景分割方法的一个关键区别在于如何处理这两类标签。我们大致可以将这些方法分为三类：无提议、基于提议和通用。无提议（自底向上）方法 [3, 6, 26, 41, 50] 从语义分割开始，随后对与实例相关的像素进行聚类。另一方面，基于提议（自顶向下）的方法 [18,25,34,53]分别执行语义分割和实例分割，然后融合两者产生的（可能相互冲突的）分割区域。最后，近期有一些方法采用了我们所称的通用架构 [7,8,27,49,55,57] ，包括 DETR [4], ；我们的实验便基于这一架构展开。它们尽可能以相同方式处理语义类别和实例类别，通常通过使用一组固定的类别无关预测，再以类似目标检测的方式处理这些预测 [56]。

在不确定性估计方面，基于集成（或基于采样）的方法，例如蒙特卡洛 Dropout [11] 和深度集成 [23] ，尽管已经问世多年 [2, 32, 40]，仍是评估新方法的基准。许多方法以集成的形式呈现，或可以视为集成 [54, 58] ，并得到广泛应用 [2, 20] ；但由于样本生成过程 [37]，它们通常被认为计算开销较大。其他流行或较新的方法 [13, 37, 44] 则存在一些问题，例如无法区分不同的不确定性来源，或需要对网络架构进行较大改动。

在现代神经网络的分割任务不确定性估计方面，绝大多数研究 [15, 22, 36, 39, 59] 都针对语义分割或与之相近的任务，以避开全景分割固有的复杂性。然而，近期有两种方法 [9, 46] 确实处理了全景分割中的复杂问题。这些方法与我们的方法不同之处在于，它们采用确定性方法（而非采样），例如证据深度学习 [44]。此外，它们使用的网络结构与我们的方法有很大不同，其解决方案针对的是

基于候选区域的架构所特有的问题，而我们的方法则聚焦于通用架构。最后，我们注意到，文献 [35] 中的方法与我们的方法有些相似，因为它们使用基本顺序算法方案（BSAS）来匹配分割区域。不过，它们关注的是实例分割，而非全景分割；实例分割是全景分割的一个子集。我们还对 BSAS 进行了优化以提高效率，而它们似乎是直接应用该算法。此外，我们的结果更加全面且一致，清楚地展示了基于样本的不确定性估计方法对全景分割可能带来的益处。相比之下，它们似乎将实例分割任务与原本用于目标检测的指标混用，因此其方法的优势及更广泛的泛化能力尚不明确。最后，据我们所知，所有上述能够在全景分割中进行不确定性估计的方法，都提出了语义、实例或空间不确定性的某种变体；与我们采用的预测熵等广为人知的不确定性函数相比，这些方法更偏临时设计。

3 方法

3.1 问题

我们需要回答的首要问题是如何在全景分割领域获得可靠的不确定性度量？我们假设使用集成模型进行不确定性估计（鉴于其广泛应用），并且全景分割架构符合我们对通用架构的定义，而不是基于提议或无提议的架构。这类通用架构会生成固定数量的预测（每个预测都具有各自的 Softmax 分数、掩码和其他属性），然后通常将其筛选为少量“优质”预测。这种网络设计风格也已用于其他领域，例如目标检测或实例分割。在全景分割中，这种架构的优势在于能够保证实例标签与语义标签在**同时**生成时保持对齐。标签的同步分配还允许它们在适当情况下相互提供信息，从而可能获得更优结果，并避免错误被强制传播。例如，如果类别标签错误且被冻结，后续的实例标签分配可能会变得困难。

鉴于这些限制，当使用包含 Q 个模型（样本）的集成从神经网络获取不确定性估计时，我们有 [2,11,23,30,31]

$$\left\{P(y \mid \boldsymbol{x}^*, \boldsymbol{\theta}^{(q)})\right\}_{q=1}^{Q}, \quad \boldsymbol{\theta}^{(q)} \sim p(\boldsymbol{\theta}|\mathcal{D}). \tag{1}$$

因此，我们需要得到一个综合所有可用样本信息的单一预测。通常，这是通过取各样本的均值来获得预测后验即

$$P(y \mid \boldsymbol{x}^*, \mathcal{D}) = \frac{1}{Q}\sum_{q=1}^{Q} P(y \mid \boldsymbol{x}^*, \boldsymbol{\theta}^{(q)}). \tag{2}$$

可以通过集成的统计量以多种方式获得该预测的不确定性；例如，我们可以将总不确定性计算为预测后验的熵 [45]

预测后验:

$$\mathcal{H}\left[P(y \mid \boldsymbol{x}^*, \mathcal{D})\right] \approx \mathcal{H}\left[\frac{1}{Q}\sum_{q=1}^{Q} P(y \mid \boldsymbol{x}^*, \boldsymbol{\theta}^{(q)})\right], \quad \boldsymbol{\theta}^{(q)} \sim p(\boldsymbol{\theta} \mid \mathcal{D}). \tag{3}$$

还可以计算互信息或样本熵等其他指标，它们在不同程度上反映认知（模型）不确定性或偶然不确定性（数据不确定性）。前者涵盖多个模型都能同样良好地拟合数据的可能性；后者则涵盖标注噪声或测量噪声，例如人工标注员可能会对图像的标签产生分歧。不同任务可能需要认知和/或偶然不确定性例如，识别分布外样本需要认知不确定性 [17]。

关键在于，公式（2）隐含地假设集成中的每个成员所作的预测都可以与其他所有成员的预测相匹配。对于图像分类或语义分割，这一假设成立，因为所有集成成员的参照系（图像或像素）相同，因此公式（2）可以在该层面应用。对于全景分割（以及实例分割），情况则未必如此。

考虑 DETR 神经网络在全景分割配置下的输出 O [4]可表示为

$$O = \{L_{N\times C}, M_{N\times h\times w}\}, \tag{4}$$

其中 L 是一组逻辑值向量，可对其应用 softmax函数，从而获得每个候选区域 $N \in N$ 的类别表示；而 M 是与每个候选区域 $n \in N$ 相关联的掩码。 C 表示数据集中的类别数。每个候选区域 n代表图像中的一个潜在对象，例如车辆、建筑物，或天空等更一般的类别。由于 N 固定，在推理阶段需要进行后处理，以保留有用的候选区域，从而生成高质量的全景分割（详情见附录C.2）。使用集成时，我们现在会获得神经网络输出 $\{O_1, \ldots, O_N\}$的多个样本，但没有明确的方法来汇总这些样本，因为每个样本中的候选区域 n 可能代表不同的对象，而且我们不知道各实例之间的对应关系。

3.2 我们的方法

我们的任务是对从神经网络中收集的多个样本进行聚合，从而获得可接受的全景分割结果及相应的像素级不确定性估计，如图 2 所示。

第一步是使用 MC Dropout 从网络中收集并处理样本 [11],如图 2 上半部分所示。然后，我们执行一些从基线方法改编的最小处理，以将 N 提议减少到 $K \leq N$ 因为许多提议本质上是噪声 [4]。这在补充材料中的算法 A1 (perSampleSeg) 中显示，结果是一组 Softmax 向量和候选区域图，它们将每个像素链接到一个实例 ID 和一个类别标签。

第二步涉及语义（背景类）分割，在图 2 中以浅紫色阴影显示，并在算法 2 中展示。虽然全景分割是一个联合问题，同时分配语义和实例标签是有帮助的，但我们发现在从网络获取样本后分解问题对性能和速度都有用。因此，我们首先通过在样本上应用均值来处理更简单的语义标签情况，以识别每个像素的类别标签。如果像素映射到语义类（不需要实例 ID），我们就完成了，因为我们只需要类别标签。如果像素映射到需要实例 ID 的类，我们继续进行实例分割步骤。

实例分割步骤在图 2 中以浅粉色阴影标出，并展示于算法 3 中。我们的目标是在不丢弃相关信息或造成过高计算复杂度的前提下，为每个像素分配实例 ID。我们的流程是基本顺序算法方案 [47] 的一种改进形式：我们检查每个样本中的每个实例，计算该实例与实例列表中各实例在图像空间中的交并比得分。如果得分高于某个阈值，我们就将其视为匹配项，并合并这两个实例。如果在列表中找不到匹配项，我们就将其视为一个此前未见的新实例，并将其添加到列表中。完成这一流程后，我们会得到一个实例列表，其中大多数实例都已与其他样本中的实例匹配。根据该列表，我们可以按匹配次数的多少，为实例分配像素（从而为其匹配类别和实例 ID），生成实例分割图 S。

最后一步是撤销先前的分解，并结合语义分割步骤和实例分割步骤的输出，生成完整分割图 S_{final}，使所有像素都被分配类别标签，并在适当情况下为部分像素分配实例 ID。这一过程展示在图 2 右下角和算法 1 中。由于我们继承了基线网络容易产生大量假正例的倾向，因此可以选择通过置信度阈值对分割结果进行剪枝。或者，我们也可以在逐像素不确定性图上应用数学依据更充分的不确定性度量（例如预测熵），并在模型不确定的区域采取适当措施（例如 交由人工处理）。

4 实验结果

4.1 通用实现

我们的框架采用 DETR 模型 [4]和 ResNet50 主干网络。我们使用 COCO [29], KITTI-STEP [51] 和 VIPER [42] 数据集。对于 COCO 数据集，我们使用来自 [4] 的预训练模型。对于其他数据集，我们训练自定义模型。采用原始配置的 DETR 方法作为基线。我们进一步调整并实现了一种称为 Jonker–Volgenant 算法的匈牙利算法变体，将其作为额外的基线 [16]。所有数据集都相当复杂且多样，包含数千张图像。我们还按照 ImageNet-C [14], 创建了所有数据集的“受损”版本，但只使用一种受损类型，即将高斯噪声和散粒噪声结合起来。我们生成了该数据集的三个版本，分别应用严重程度等级 1、2 和 3 的上述受损类型。

算法 1： generateSegmentation(Net)

输入： 全景分割神经网络 *Net*
输出： 分割图 $S^{H\times W}$，逐像素不确定性图 $U^{H\times W}$

从网络 *Net* 收集 Q 个样本，得到：
Logits $L^{Q\times N\times C}$ 和掩码 $M^{Q\times N\times h\times w}$
// C: 类别
// h & w: 输入图像尺寸
// N: 候选区域数量
$S_{perSample}, I \leftarrow$ perSampleSeg(L, M)
$S_{initial}, SC, MC \leftarrow$stuffSeg($S_{perSample}$, I)
$S_{final} \leftarrow$thingSeg($S_{perSample}$, I, $S_{initial}$, MC)
// 剪枝可选
$S_{final} \leftarrow$剪枝(S_{final}, SC)
// 可使用多个不确定性函数
$U \leftarrow$不确定性(SC)
返回 S_{final}, U

算法 2: stuffSeg(S, I)

输入： 每样本提议图 $S^{Q\times H\times W}$，候选 softmax 向量列表，长度为 Q $I^{K\times C}$
输出： 语义分割图 $S^{H\times W}$，置信度图 $SC^{Q\times H\times W\times C}$，平均置信度图 $MC^{H\times W\times C}$,

// SC: 方程 (1)
$SC^{Q\times H\times W\times C} \leftarrow \mathbf{0}$
对于每个样本 $q \in Q$ **执行**
  对于每个候选区域 $k \in K$ **执行**
    $SC_{q,i,j} \leftarrow I_{q,k}\ \forall\ i, j$ 其中 $S = k$
  end
结束
// MC: 式. (2)
$MC^{H\times W\times C} \leftarrow$ 均值$_Q$ SC
// LB: 逐像素标签
$LB^{H\times W} \leftarrow$ 最大值索引$|_C$ MC
$S^{H\times W} \leftarrow \mathbf{0}$
foreach标签 $lb \in LB$ **do**
  如果 $lb \subset$背景类 **那么**
    $S_{i,\ j} \leftarrow lb\ \forall\ i,\ \ j$ 其中 $lb = LB_{i,\ j}$
  end
结束
return S, SC, MC

算法 3： thingSeg(*S*, *I*, $S_{initial}$, MC)

输入： 每样本提议图 $S^{Q\times H\times W}$，提议 softmax 向量列表长度 Q
$I^{K\times C}$，初始语义分割图 $S_{initial}^{H\times W}$，平均置信度图
$MC^{H\times W\times C}$

输出： 完整分割图 $S^{H\times W}$

$LB^{H\times W} \leftarrow$ 最大值索引$_C$ MC

$ID \leftarrow \mathbf{0}$

$ID_i^{Q\times H\times W}{}_{,\,j} \leftarrow S_{i,\,j}\ \forall\ i,\ \ j$ 其中 $LB_{i,\,j} \subseteq$ 物体类

// SEG：分割区域列表

$SEG \leftarrow \mathbf{0}$

// 以下与 BSAS 类似

对样本 $q \in Q$ **执行**

  对 提议实例 $g \in I_q$ **执行**

    匹配 $\leftarrow$假

    对每个 实例 $h \in SEG$**执行**

      如果 交并比 $\geq$ 阈值 **那么**

        匹配 $\leftarrow$ 为真

        将 g 合并到 h

        跟踪合并次数

      end

    end

    如果 匹配 $=$ *False*则

      将 g 添加到 SEG

    end

  end

结束

$SEG \leftarrow$按合并排序(SEG)

$S \leftarrow S_{initial}\ \forall\ i,\ \ j$ 其中 $S_{initial} \neq 0$，否则 **0**

遍历 实例 $h \in SEG$**执行**

  如果 $h \geq$的合并计数达到阈值 **则**

    $S \leftarrow S \cup ((S = 0) \cap h)$

  end

结束

返回 S

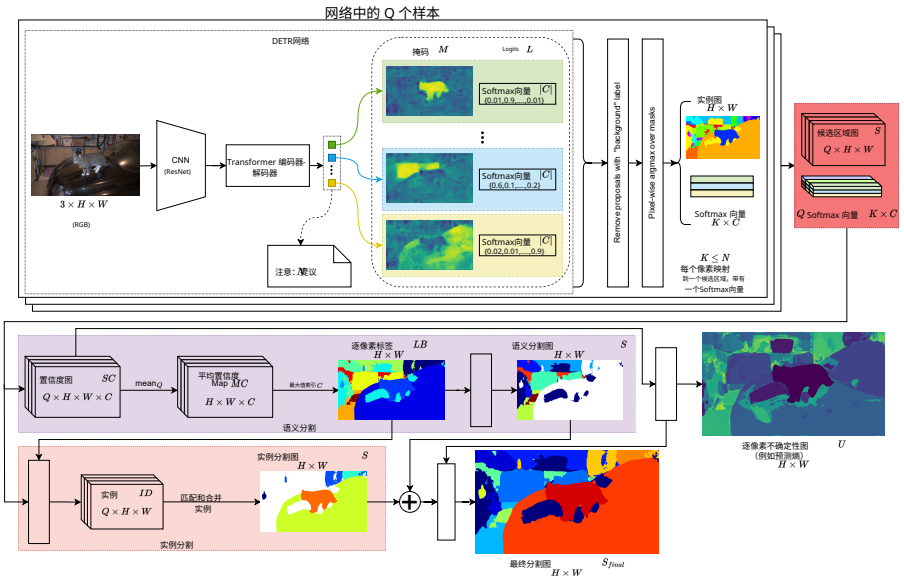


图 2： 我们方法的概览。详情请参见正文。请注意，为简洁清晰起见，我们简化了部分实现细节。

有关更多实现细节，请参见补充材料中的附录 C 和 D，其中包括不进行剪枝的基线变体，以及一些实验的补充结果。此外，请注意，代码将在论文被接收后发布。

在评估中，我们使用 [19] 提出的全景质量（PQ）指标；该指标通常用于文献中全景分割性能的评估 [5,52]。其定义如下：

$$\mathrm{PQ}_{\mathrm{class}} = \frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}. \tag{5}$$

在该指标中，真实标注与预测结果之间进行分割匹配；当交并比（IoU）≥ 0.5 时，判定为真正例。按照公式（5），按类别计算真正例、假正例和假负例。我们报告所有类别的平均结果。在此情境下不定义真负例。全景质量（PQ）指标还可分解为分割质量（SQ）和识别质量（RQ），即

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}|}}_{\mathrm{SQ}} \times \underbrace{\frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}}_{\mathrm{RQ}}. \tag{6}$$

4.2 基本性能

在我们的第一个实验中（如表 1 所示），我们在 KITTI-STEP、COCO 和 VIPER 数据集上评估了我们的方法。

		干净			受损		
	近似采样 全景质量↑		SQ↑	RQ↑	PQ↑	SQ↑	RQ↑
COCO	基线 不适用	43.4	79.3	53.8	28.9	75.4	36.8
	Hng. 5	39.5±0.1	78.8±0.1	49.0±0.1	28.1±0.0	76.2±0.2	35.7±0.1
	Hng. 15	40.7±0.1	79.1±0.1	50.4±0.1	28.9±0.1	76.3±0.2	36.7±0.1
	我们的方法 5	44.1±0.1	79.5±0.1	**54.5±0.1**	**30.6±0.1**	76.8±0.2	**38.7±0.1**
	我们的方法 15	**44.4±0.1**	**79.9±0.1**	**54.5±0.1**	**30.5±0.1**	**77.1±0.1**	38.4±0.1
KITTI	基线 不适用	39.9	**77.5**	50.2	19.3	54.9	26.0
	Hng. 5	36.1±0.2	76.5±0.2	45.4±0.2	17.1±0.1	**58.3±1.2**	23.1±0.1
	Hng. 15	37.6±0.1	76.9±0.1	47.2±0.1	17.8±0.1	**58.1±0.2**	24.1±0.1
	我们的方法 5	**41.8±0.1**	76.9±0.1	**52.7±0.2**	20.0±0.0	57.4±1.1	**27.1±0.0**
	我们的方法 15	**41.9±0.1**	77.0±0.1	**52.7±0.1**	**20.1±0.0**	57.7±0.9	**27.1±0.0**
VIPER	基线 不适用	31.1	74.0	**39.7**	**10.3**	66.8	**14.5**
	Hng. 5	27.0±0.0	73.1±0.1	34.6±0.0	8.4±0.0	68.8±0.2	12.1±0.0
	Hng. 15	28.4±0.0	73.6±0.1	36.3±0.0	9.2±0.0	69.2±0.2	13.1±0.0
	我们的方法 5	31.1±0.0	74.2±0.1	39.5±0.0	10.2±0.0	**69.5±0.1**	14.3±0.0
	我们的方法 15	**31.3±0.0**	**74.6±0.0**	39.6±0.0	10.1±0.0	**69.7±0.2**	14.2±0.0

表1： 基本性能。结果表示为在随机情况下运行10次的结果。*受损* 列表示使用高斯噪声和散粒噪声（严重程度1）。*Hng.* 指匈牙利方法。

主要发现有三点。第一，无论采用何种方法，加入高斯噪声和散粒噪声都会显著降低整体性能；从网络中收集更多蒙特卡洛 Dropout 样本也无法弥补噪声导致的网络性能不佳。第二，除一项例外，我们的方法通常比基线带来小幅性能提升；在干净数据和含噪数据上，相较于匈牙利算法，性能提升则显著得多。能够达到或超过基线表现值得肯定，因为我们只会在后续实验中使用获得的不确定性估计。这表明我们的方法能够正确整合多个样本中的信息。关于匈牙利算法的性能，请参见附录 D.1。总的来说，我们注意到，其他研究也曾发现使用多个样本时性能会下降 [9]；而我们的方法总体上能够避免这种情况。第三，虽然增加样本数量确实会略微提升性能，但在此情境下，5 个样本通常足以生成合理的分割结果；再增加样本只会提高计算复杂度，收益却有限。

4.3 不确定性估计的应用

在图3中，对于全部三个数据集，我们绘制了类ROC（受试者工作特征）曲线，针对一项实验，我们改变一个阈值，该阈值决定一个像素必须达到何种不确定程度才会被从数据集中移除（在评估期间被忽略，就好像不存在真实标注一样）。根据数据集和/或是否使用噪声，我们在验证集内最多移除60-95%的被标记为不确定的像素，

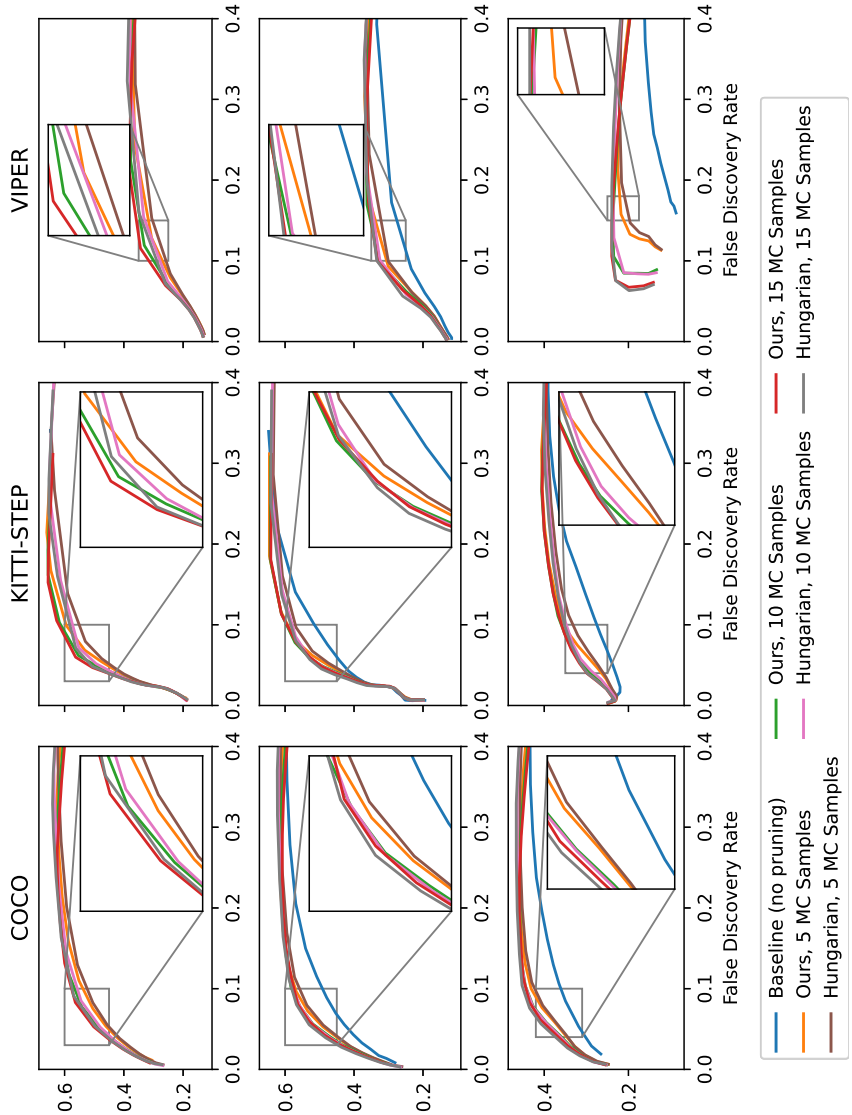


Figure 3: Using various uncertainty measures to identify and remove the most uncertain pixels in the dataset, we plot the True Positive Rate versus the False Detection Rate. Better uncertainty estimates result in a greater area under the curve. Results are collected over 10 runs and averaged; standard deviation is visually negligible. *Noisy* indicates that Gaussian and Shot Noise at severity level 1 is applied to the images provided to the network. The baseline is not shown in the mutual information plots as the measure requires more than 1 sample to calculate.

数据；再移除更多会破坏评估指标。曲线下面积越大越好。我们绘制了干净数据下的预测熵和互信息度量，以及严重程度等级1下高斯噪声和散粒噪声的预测熵。我们可以得出四个观察结果。首先，正如预期，我们使用来自蒙特卡洛样本的预测熵（公式(3)）的方法可靠地优于使用Softmax熵的基线 [38]。我们的方法表现与匈牙利算法相当或更好，尽管在少数情况下有些例外。我们还注意到，使用蒙特卡洛Dropout的性能提升在图表右侧会下降甚至略微逆转；这是预期的，因为它与其他类似工作一致 [39]。此外，我们在像素 级别移除不确定数据的方式与我们的评估依赖于分割区域 匹配之间存在不匹配。第二个观察是，添加更多样本可提供更好的不确定性估计，正如预期，但随着使用更多样本（≥ 10），收益递减。第三个观察是，与表1一致，添加噪声通常会大幅降低性能，而依赖蒙特卡洛Dropout的集成不确定性估计方法不一定对此更具韧性。最后一个观察是，捕捉认知不确定性的预测熵和互信息都表现良好。这表明认知不确定性得到了正确建模，正如预期，因为蒙特卡洛Dropout侧重于此。

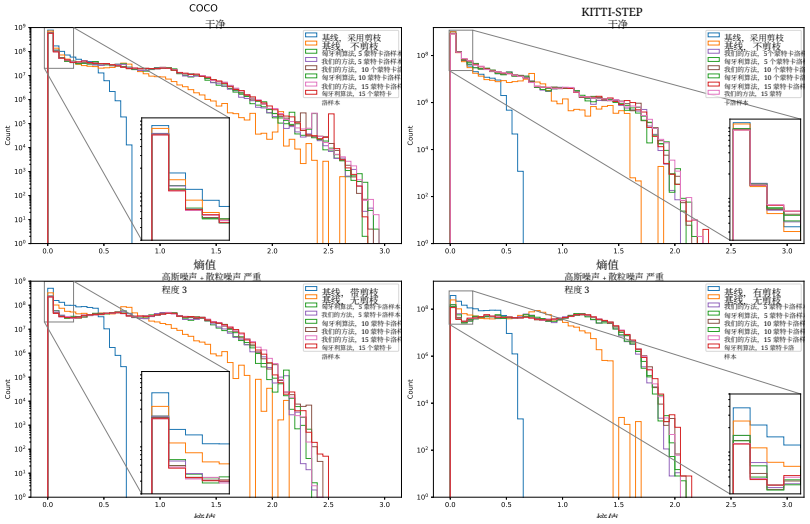


图 4： 对我们的方法（采用蒙特卡洛Dropout）和基线分别进行预测熵与 Softmax 熵的分箱和绘图。上：干净数据。下：高严重程度下的高斯噪声和散粒噪声。请注意，y 轴采用对数刻度。

在图 4 中，我们按照与 [23] 类似的流程，对 COCO 和 KITTI-STEP 数据集中所有图像的所有像素所产生的预测的预测熵值（对于基线，即 Softmax 熵 [38]）进行分箱并绘图。我们分别针对

干净数据以及添加大量高斯噪声和散粒噪声的数据进行上述操作。我们按照 [4] 绘制基线结果，也绘制禁用剪枝后的结果。采用更好的不确定性度量时，过度自信的预测应当更少；在噪声等异常条件下，不确定性应当较高，因为我们预期网络的表现会更差。具体而言，在有噪声的条件下，我们预期分布会更加均匀。总体而言，两个数据集上的结果都验证了这一假设。首先，默认基线采用的阈值会筛除预测中不够自信的部分，从而大幅削减可用的不确定性信息。禁用这些阈值后可以看到，在干净数据等正常条件下，与基线的该变体相比，我们的方法和匈牙利方法都能产生分布更均匀的数值。在有噪声的条件下，这种效果更加明显，表明对（未知）真实条件概率的表征更好；这些概率本应更加均匀。有趣的是，仅在 COCO 数据集上，我们发现添加噪声后， $2.5 - 3$ 范围内极不确定的预测变少了；但无论采用哪种方法，这一现象都一致出现，因此很可能是由基础网络模型导致的。我们还注意到，尽管我们的方法可与任何集成配合使用，但我们使用蒙特卡洛 Dropout 进行的评估自然会受到其自身局限性的影响，如 [23] 所示。

4.4 定性结果

图 5 展示了我们选取的图像及其在两个数据集上的全景预测。总体而言，正如定量实验所反映的那样，与基线相比，我们的方法略微改善了分割效果。然而，在 COCO 数据集的某些情况下，真实标注存在歧义，因此部分改进可能无法体现在 PQ 指标中。

图 6 展示了基线模型与我们的方法在 Softmax 和预测熵方面的热力图对比。从定性结果来看，我们的方法能够提供更优的不确定性估计：许多区域呈现出更高的异质性，表明模型考虑了更多样化的预测。鉴于图 4 中的定量结果，这些定性差异是可以预期的。

5 结论

我们提出了一种算法，使集成不确定性估计首次能够应用于全景分割领域。我们通过蒙特卡洛 Dropout [11] 和基于 Transformer 的 DETR 网络 [4] 展示了该算法的有效性；具体而言，借助该算法生成的全景分割结果略优于基线 DETR，且明显优于匈牙利算法。最重要的是，我们可以使用预测熵等具有数学依据的不确定性函数，生成更可靠的置信度度量，但其效果受所选样本生成过程（如我们实验中的蒙特卡洛 Dropout）的限制。这样便能够识别并妥善处理各种情况下的错误。我们的方法相当直接，适用范围也较广。按照本文所述，无需修改架构或重新训练，即可使用现有神经网络模型。

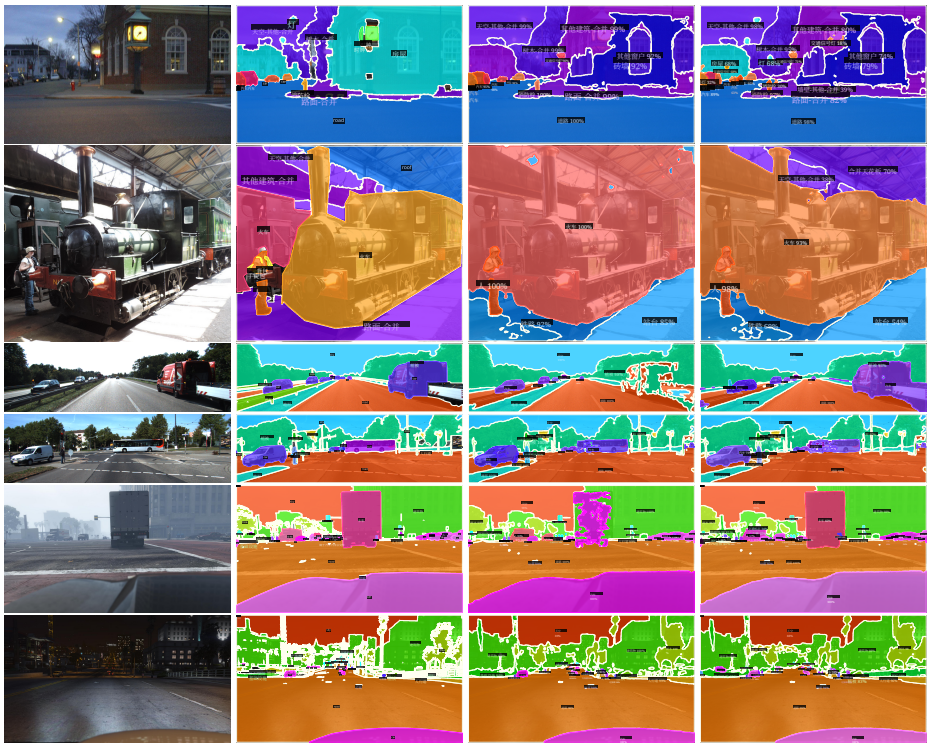


图 5： COCO（前两行）、KITTI-STEP（中间两行）和 VIPER（后两行）验证集中的定性示例。从左到右：输入图像、真实标注、基线分割结果，以及使用 15 个蒙特卡洛样本的我们的方法。

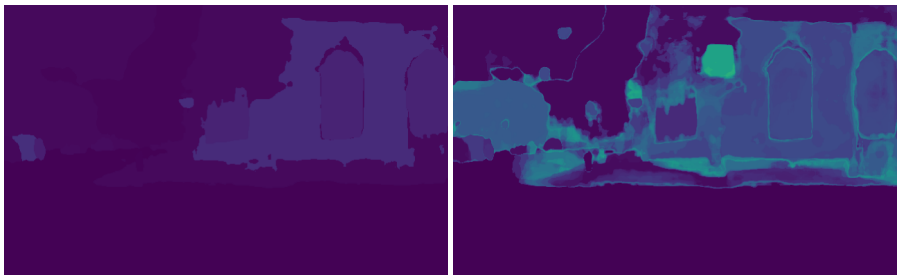


图 6： 对图 5 中最上方示例的基线方法（左）与使用 15 个蒙特卡洛样本的我们的方法（右）的预测熵不确定性热图进行并排比较。颜色越浅表示不确定性越高。

更广泛地说，我们的方法应兼容任何基于集成的不确定性估计流程（例如 [23]）以及最新的通用全景分割架构，包括当前最先进的方法 [27]。我们希望这些发现能够推动未来研究，以获得更好的全景分割不确定性估计，因为现在可以将其与蒙特卡洛Dropout [11]等已知方法进行公平比较。

6 致谢

我们感谢Nvidia提供了我们实验所用的GPU之一。

参考文献

1. Alex Kendall, V.B., Cipolla, R.: Bayesian SegNet：用于场景理解的深度卷积编码器-解码器架构中的模型不确定性。载于：Tae-Kyun Kim, Stefanos Za- feiriou, G.B., Mikolajczyk, K.（编）英国机器视觉会议论文集（BMVC）。第 57.1–57.12 页。BMVA Press（2017年9月）。https://doi.org/10.5244/C. 31 57 2 57, 10 5244 31 . https://dx.doi.org/ . /C. . . Ashukha, A., Lyzhov, A., Molchanov, D., Vetrov, D.: 深度学习中域内不确定性估计与集成的陷阱。载于：国际学习表征会议（2020年），https://openreview.net/forum?id=BJxI5gHKDr 3. Bonde, U., Alcantarilla, P.F., Leutenegger, S.: 无边界框全景分割的探索。载于：Akata, Z., Geiger, A., Sattler, T.（编）模式识别。第 316–330 页。计算机科学讲义，Springer国际出版公司，Cham（2021年）。https: //doi.org/ 10 1007 978 3 030 71278 5_23 . / - - - - . Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: 基于Transformer的端到端目标检测。载于：Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M.（编）计算机视觉—ECCV 2020。第 213–229 页。计算机科学讲义，Springer国际出版公司，Cham（2020年）。https://doi.org/10.1007/978- 3-030-58452-8 13_5. Chen, T., Li, L., Saxena, S., Hinton, G., Fleet, D.J.: 图像和视频全景分割通用框架。载于：IEEE/CVF计算机视觉国际会议论文集（ICCV）。第 909–919 页（2023年10月）6. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-DeepLab：一种简单、强大且快速的自底向上全景分割基线。载于：CVPR（2020年）7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: 用于通用图像分割的掩码注意力掩码Transformer。载于：2022年 IEEE/CVF计算机视觉与模式识别会议（CVPR）。第 1280–1289 页（2022年6月）。https://doi.org/ 10.1109/CVPR52688.2022.00135，iSSN: 2575-7075 8. Cheng, B., Schwing, A., Kirillov, A.: 逐像素分类并非语义分割所需的一切。载于：Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W.（编）神经信息处理系统进展。第 34 卷，第 17864–17875 页。Curran Associates, Inc.（2021年），https://proceedings.neurips.cc/paper/2021/file/950a4152c2b4aa3ad78bdd6b366cc179-Paper.pdf 9. Deery, J., Lee, C.W., Waslander, S.: ProPanDL：面向不确定性的全景分割模块化架构（2023年4月）。https://doi.org/10.48550/arXiv. 2304.08645, http://arxiv.org/abs/2304.08645, arXiv:2304.08645 [cs]

10. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.：CARLA：一个开放式城市驾驶模拟器。载于：第1届年度机器人学习会议论文集。第1–16页（2017） 11. Gal, Y., Ghahramani, Z.：将丢弃作为贝叶斯近似：在深度学习中表示模型不确定性。载于：国际机器学习会议。第1050–1059页。PMLR（2016年6月），http://proceedings.mlr.press/v48/gal16.html，iSSN：1938-7228 12. Ghahramani, Z.：概率机器学习与人工智能。《自然》**521**(7553)，452–459（2015年5月）。https://doi.org/10.1038/nature14541，https://www.nature.com/articles/nature14541，期号：7553，出版商：Nature Pub-lishing Group 13. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.：现代神经网络的校准。载于：第34届国际机器学习会议论文集——第70卷。第1321–1330页。ICML'17，JMLR.org，澳大利亚新南威尔士州悉尼（2017年8月） 14. Hendrycks, D., Dietterich, T.：对神经网络应对常见损坏和扰动的鲁棒性进行基准测试。《国际学习表征会议论文集》（2019） 15. Huang, P.Y., Hsu, W.T., Chiu, C.Y., Wu, T.F., Sun, M.：视频语义分割的高效不确定性估计。载于：Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y.（编）。《计算机视觉——ECCV 2018》。第536–552页。《计算机科学讲义》，Springer国际出版公司，Cham（2018）。https://doi.org/10.1007/978-3-030-01246-5 32_16. Jonker, R., Volgenant, A.：用于稠密和稀疏线性指派问题的最短增广路径算法。《计算》**38**(4)，325–340（1987年12月）。https://doi.org/10.1007/BF02278710，https://doi.org/10.1007/BF02278710 17. Kendall, A., Gal, Y.：贝叶斯深度学习用于计算机视觉时，我们需要哪些不确定性？arXiv:1703.04977 [cs]（2017年10月），http://arxiv.org/abs/1703. 04977，arXiv：1703.04977 18. Kirillov, A., Girshick, R., He, K., Dollar, P.：全景特征金字塔网络。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集（2019年6月） 19. Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P.：全景分割。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集（2019年6月） 20. Kirsch, A., van Amersfoort, J., Gal, Y.：BatchBALD：用于深度贝叶斯主动学习的高效多样化批量采集。载于：神经信息处理系统进展。第32卷。Curran Associates, Inc.（2019），https://proceedings.neurips.cc/paper/2019/hash/95323660ed2124450caaac2c46b5ed90- Abstract.html 21. Kuhn, H.W.：用于指派问题的匈牙利方法。《海军研究后勤季刊》**2**(1-2)，83–97（1955）。https://doi.org/10.1002/nav.3800020109，https : / / onlinelibrary . wiley . com / doi / abs / 10 . 1002 / nav . 3800020109 _ 10 1002 3800020109 ，22 电子印本：https://onlinelibrary.wiley.com/doi/pdf/ . /nav. . Kwon, Y., Won, J.H., Kim, B.J., Paik, M.C.：分类中使用贝叶斯神经网络进行不确定性量化：在生物医学图像分割中的应用。《计算统计与数据分析》**142**，106816（2020年2月）。https://doi.org/ 10 . 1016 / j . csda . 2019 . 106816，https : / / www . sciencedirect . com / science/article/pii/S016794731930163X 23. Lakshminarayanan, B., Pritzel, A., Blundell, C.：使用深度集成进行简单且可扩展的预测不确定性估计。载于：第31届国际会议论文集。

神经信息处理系统。第 6405–6416 页。NIPS'17，Curran Associates Inc.，雷德胡克，纽约州，美国（2017年12月）24. Levine, M.D.：面包板火星机器人在室内环境中运行的场景分析。技术报告 NASA-CR-135711，美国国家航空航天局喷气推进实验室（1973年9月），https://ntrs.nasa. gov/citations/19730023424，NASA技术报告服务器作者单位：喷气推进实验室、加利福尼亚理工学院。NASA技术报告服务器文档 ID：19730023424 NASA技术报告服务器研究中心：旧版CDMS（CDMS）25. Li, Q., Qi, X., Torr, P.H.：面向全景分割的训练与推理统一方法。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集（2020年6月）26. Li, Y., Zhao, H., Qi, X., Wang, L., Li, Z., Sun, J., Jia, J.：用于全景分割的全卷积网络。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集。第 214–223 页（2021年6月）27. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., Lu, T.：Panoptic SegFormer：利用 Transformer 深入探索全景分割。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集。第 1280–1289 页（2022年6月）28. Liao, Y., Xie, J., Geiger, A.：KITTI-360：用于城市场景理解的全新数据集与基准。arXiv 预印本 arXiv:2109.13410（2021）29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.：微软COCO：上下文中的常见物体。载于：Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T.（编）计算机视觉——ECCV 2014。第 740–755 页。《计算机科学讲义》，Springer国际出版公司，Cham（2014）。https://doi.org/10.1007/978-3-319-10602-1_4830. Malinin, A.：深度学习中的不确定性估计及其在口语评估中的应用。博士论文，剑桥大学（2019）31. Malinin, A.：监督学习中的不确定性估计（2019），https://youtu.be/ P4WUl7TDdLo，深度学习与贝叶斯方法暑期学校 32. Malinin, A., Band, N., Ganshin, Alexander, Chesnokov, G., Gal, Y., Gales, M.J.F., Noskov, A., Ploskonosov, A., Prokhorenkova, L., Provilkov, I., Raina, V., Raina, V., Roginskiy, Denis, Shmatova, M., Tigas, P., Yangel, B.：Shifts：一个涵盖多种大规模任务的真实分布偏移数据集。arXiv:2107.07455 [cs, stat]（2021年7月），http://arxiv. org/abs/2107.07455，arXiv: 2107.07455 版本：2 33. Marr, D.：视觉：对人类表征及视觉信息处理的计算研究。W.H. Freeman，旧金山（1982），http://digitool.hbz- nrw.de:1801/webclient/DeliveryManager?pid=2948165，章节：xvii，397 页：插图；24 厘米 34. Mohan, R., Valada, A.：EfficientPS：高效全景分割。《国际计算机视觉期刊》**129**(5)，1551–1579（2021年5月）。https://doi.org/10.1007/ s11263-021-01445-z，https://doi.org/10.1007/s11263-021-01445-z 35. Morrison, D., Milan, A., Antonakos, N.：使用随机失活采样估计实例分割中的不确定性。载于：CVPR 2019机器人视觉概率目标检测挑战赛（2019），https://www.amazon.science/publications/estimating-uncertainty-in-instance-segmentation-using-dropout-sampling 36. Mukhoti, J., van Amersfoort, J., Torr, P.H.S., Gal, Y.：语义分割中的深度确定性不确定性。技术报告 arXiv:2111.00079，arXiv（2021年10月）。https://doi.org/10.48550/arXiv.2111.00079，http://arxiv.org/abs/2111.00079，arXiv:2111.00079 [cs] 类型：文章

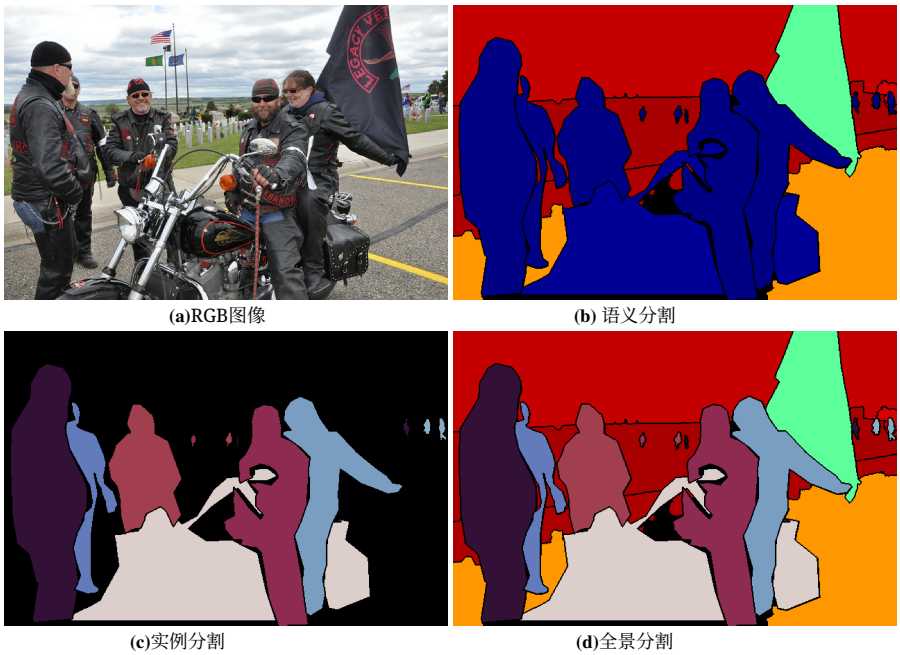
37. Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P.H.S., Gal, Y.：深度确定性不确定性：一种简单的基线（2022年1月）。https://doi.org/10.48550/arXiv.2102. 11582, 2102 11582, 2102 11582 [ ]
38
http://arxiv.org/abs/ . arXiv: . cs, stat . Mukhoti, J., Kirsch, A., Amersfoort, J.v., Torr, P.H.S., Gal, Y.：具有归纳偏置的确定性神经网络可捕获认知不确定性与偶然不确定性（2021），_电子预印本：2102.11582 39. Nair, T., Precup, D., Arnold, D.L., Arbel, T.：探索深度网络中的不确定性度量在多发性硬化症病灶检测与分割中的应用。《医学图像分析》**59**，101557（2020年1月）。https://doi.org/10.1016/j.media.2019. 101557，https : / / www . sciencedirect . com / science / article / pii / S1361841519300994 40. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.：你能相信模型给出的不确定性吗？数据集偏移下预测不确定性的评估。载于：神经信息处理系统进展。第 32 卷。Curran Associates, Inc.（2019），https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html 41. Qiao, S., Chen, L.C., Yuille, A.：DetectoRS：利用递归特征金字塔和可切换空洞卷积检测目标。载于：IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集。第 10213–10224 页（2021年6月）42. Richter, S.R., Hayder, Z., Koltun, V.：为基准测试而玩。载于：IEEE国际计算机视觉会议，ICCV 2017，意大利威尼斯，2017年10月22日至29日。第 2232–2241 页（2017）。https://doi.org/10.1109/ICCV.2017.243，https://doi.org/ 10.1109/ICCV.2017.243 43. Rosenfeld, A., Kak, A.C.：第 10 章——分割。载于：Rosenfeld, A., Kak, A.C.（编）《数字图像处理》（第二版），《计算机科学与科学计算》，第 2 卷，第 55–190 页。Academic Press，圣迭戈，第二版（1982）。https://doi.org/https://doi.org/10.1016/B978- 0- 12-597302- 1.50009-8，https://www.sciencedirect.com/science/article/pii/B9780125973021500098 44. Sensoy, M., Kaplan, L., Kandemir, M.：通过证据深度学习量化分类不确定性。载于：Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R.（编）神经信息处理系统进展。第 31 卷。Curran Associates, Inc.（2018），https://proceedings.neurips.cc/paper/2018/file/a981f2b708044d6fb4a71a1463242520-Paper.pdf 45. Shannon, C.E.：通信的数学理论。《贝尔系统技术期刊》**27**(3)，379–423（1948年7月）。https://doi.org/10.1002/j.1538-7305. 1948.tb01338.x, https://ieeexplore.ieee.org/abstract/document/ 6773024，会议名称：《贝尔系统技术期刊》46. Sirohi, K., Marvi, S., Büscher, D., Burgard, W.：不确定性感知全景分割。《IEEE机器人与自动化快报》**8**(5)，2629–2636（2023）。https://doi.org/ 10.1109/LRA.2023.3256926 47. Theodoridis, S., Koutroumbas, K.：第 12 章——聚类算法I：序贯算法。载于：Theodoridis, S., Koutroumbas, K.（编）《模式识别》（第四版），第 627–652 页。Academic Press，波士顿，第四版（2009）。https: / / doi . org / https : / / doi . org / 10 . 1016 / B978 - 1 - 59749 - 272 - 0 . 50014- 1，https://www.sciencedirect.com/science/article/pii/ B9781597492720500141 48. Tighe, J., Niethammer, M., Lazebnik, S.：利用目标实例与遮挡排序进行场景解析。载于：IEEE计算机视觉与模式识别会议（CVPR）论文集（2014年6月）

49. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: MaX-DeepLab：基于掩码Transformer的端到端全景分割。载于：IEEE/CVF计算机视觉与模式识别会议论文集（CVPR），第5463–5474页（2021年6月）50. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-DeepLab：用于全景分割的独立轴向注意力。载于：Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M.（编），计算机视觉—ECCV 2020，第108–126页。计算机科学讲义，Springer国际出版公司，Cham（2020年）。https://doi.org/10.1007/978-3-030-58548-8_751. Weber, M., Xie, J., Collins, M., Zhu, Y., Voigtlaender, P., Adam, H., Green, B., Geiger, A., Leibe, B., Cremers, D., Osep, A., Leal-Taixe, L., Chen, L.C.: STEP：分割并跟踪每个像素。arXiv:2102.11859 [cs]（2021年2月），http://arxiv.org/abs/ 2102.11859，arXiv: 2102.11859 52. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: 基于文生图扩散模型的开放词汇全景分割。载于：IEEE/CVF计算机视觉与模式识别会议论文集（CVPR），第2955–2966页（2023年6月）53. Yang, Y., Li, H., Li, X., Zhao, Q., Wu, J., Lin, Z.: SOGNet：用于全景分割的场景重叠图网络。AAAI人工智能会议论文集 **34**(07)，12637–12644（2020年4月）。https://doi.org/10.1609/aaai.v34i07. 6955, https://ojs.aaai.org/index.php/AAAI/article/view/6955，期号：07 54. Yeo, T., Kar, O.F., Zamir, A.: 通过跨域集成实现鲁棒性。arXiv:2103.10919 [ ] 2021 2103 10919, 2103 10919 cs（3月），55 http://arxiv.org/abs/。arXiv:。Yu, Q., Wang, H., Kim, D., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: CMT-DeepLab：用于全景分割的聚类掩码Transformer。载于：IEEE/CVF计算机视觉与模式识别会议论文集（CVPR），第2560–2570页（2022年6月）56. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: 现代深度学习目标检测模型综述。数字信号处理 **126**，103514（2022年6月）。https://doi.org/10.1016/j.dsp.2022.103514, https://www.sciencedirect.com/science/article/pii/S1051200422001312 57. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-Net：迈向统一的图像分割。载于：Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W.（编），神经信息处理系统进展，第34卷，第10326–10338页。Curran Associates, Inc.（2021年），https://proceedings.neurips.cc/paper/2021/file/55a7cf9c71f1c9c495413f934dd1a158-Paper.pdf 58. Zhang, Z., Dalca, A.V., Sabuncu, M.R.: 使用结构化丢弃对卷积神经网络进行置信度校准。arXiv:1906.09551 [cs, stat]（2019年6月），http:// arxiv.org/abs/1906 09551 59 09551，1906 . arXiv:。Zheng, Z., Yang, Y.: 通过不确定性估计校正伪标签学习，用于领域自适应语义分割。国际计算机视觉期刊 **129**(4)，1106–1120（2021年4月）。https://doi.org/10.1007/s11263-020-01395-y, https://doi.org/10.1007/s11263-020-01395-y

补充材料

全景分割可视化

在图A1中，我们展示了全景分割、语义分割和实例分割之间的区别。请注意，来自 COCO 数据集 [29] 的图像通常能代表我们所用数据集场景的复杂程度；在这个具体案例中，场景中的物体与人有相当程度的相似性。数据集本身的复杂性，加上全景分割任务固有的复杂性，有助于解释为什么据我们所知，所有全景分割算法都以某种方式使用神经网络。



(a)RGB图像 **(b)** 语义分割

(c)实例分割 **(d)**全景分割

图 A1： 语义分割、实例分割和全景分割的比较，以及来自 COCO 数据集的某幅图像的真实标注数据 [29]。语义分割 (b) 使用逐像素类别标签。实例分割 (c) 使用逐对象掩码和类别标签。全景分割 (d) 使用逐像素类别和实例标签。改编自 [19] 中的图 1。

B 算法详情

本文介绍的大多数算法已在论文中阐述，我们在算法 A1 中正式描述其余算法。该算法扩展了论文中提供的概述。

算法 A1： perSampleSeg(*L*, *M*)

输入： Logits $L^{Q\times N\times C}$，掩码 $M^{Q\times N\times h\times w}$
输出： 每样本提议图 $S^{Q\times H\times W}$，候选 softmax 向量列表长度 Q，其中 $I^{K\times C}$

// 此算法基于基线方法 [4]

遍历样本 $q \in Q$ **执行**
  $S \leftarrow \emptyset$
  遍历 候选区域 $n \in N \subseteq q$ **执行**
    $l^C \leftarrow \text{Softmax}\, L_{\,q,n}$
    标签 $\leftarrow$ 最大值索引$_C\, l$
    如果标签 = 背景 **则**
      移除候选区域 n 来自 L & M
    end
  end
  // 现在每个 K proposals q
  // $K \leq N$ 移除后
  $M_q^{K\times H\times W} \leftarrow$放大$(M_q)$
  $S_q^{H\times W} \leftarrow$ 最大值索引$_K(\text{Softmax}_K\, M_q)$
  $I_q^{K\times C} \leftarrow l_k\ \forall\ k \in K$
结束
$S^{Q\times H\times W} \leftarrow$ 拼接 $S_q\ \forall\ q \in Q$
$I^{Q\times K\times C} \leftarrow$ 拼接 $I_q\ \forall\ q \in Q$
return S , I

C 实现细节与原理

C.1 概述

我们选择使用 DETR [4] 模型，因为与其他方法相比，它在保持实现较为简单的同时，也能取得令人满意的性能。选择 ResNet50 主干网络，是因为与其他主干网络（如 ResNet101）相比，它所需的计算资源更少，代价是性能略有下降。由于我们的重点是不确定性估计，而非追求最先进的性能，这种权衡是可以接受的。对于 COCO 数据集，我们使用来自 [4] 的预训练模型（未作修改）。对于 KITTI-STEP 和 VIPER 数据集，我们将其视为标准的全景分割任务；由于没有可用的预训练模型，因此训练模型 100 个周期，批量大小为 1。需要注意的是，VIPER 数据集依靠游戏引擎生成模拟数据，而其他数据集收集的是真实世界图像。所有数据集的训练和评估均使用各自数据集提供的训练集和验证集划分（COCO 使用 2017 版本）。在部分实验中使用数据集的“受损”版本时，仅在推理阶段应用受损处理；所有模型都只使用“干净”数据进行训练。在适用情况下，我们会将数据集从其原始格式转换为与 COCO 全景格式一致的格式。其他训练参数均遵循 [4],，包括在训练进行到$_{2\ 3}$ 时降低学习率。我们采用迁移学习：使用 [4]为 COCO 数据集提供的预训练权重，在训练期间初始化模型。不过，一个关键区别是，我们只以全景分割模式训练模型，而不是先训练边界框模型，再训练分割模型。我们发现，这种做法在所选的全景质量指标上表现更好，且所需训练时间更短。对于基线以外的实验，我们在测试时启用丢弃，在模型 Transformer 部分的编码器和解码器中采用 [4] 的默认值 $p = 0.1$ 即模型中所有使用丢弃的层。

C.2 基线

在所有实验中，我们用于比较的基线是按照 [4] 中的参数和代码执行模型。在其他数据集上，我们使用训练好的模型。由于在推理阶段禁用了蒙特卡洛 Dropout，基线结果是确定性的。该基线体现了在不使用集成中的任何样本、并将所有混杂变量降至最低的情况下能够达到的效果。

DETR网络最基本的运行方式会产生大量假正例，因此需要进行一定程度的剪枝。基线通过要求候选区域的 Softmax 分数至少达到 85%，且像素数量不少于 4，来确定一个分割区域。任何不满足这些阈值的像素都会被分配给次优分割区域。因此，当网络对图像中的物体不确定时（例如 置信度仅为 75%），这些物体可能会被并入附近的其他区域，例如道路或天空。这种简单粗暴的方法确实有助于提高基线的全景质量指标，但当用户想知道网络何时对其预测结果不确定时，这种方法就完全不适用了。我们将这一过程称为剪枝，并将其作为标准基线。在论文的一些实验中，我们采用或纳入

关闭此剪枝功能时基线的结果，以便与我们的方法进行更恰当的比较。论文中的图 3 和图 4 展示了相关示例。

作为另一种能够从多个集成样本生成预测的基线方法，我们实现了匈牙利方法的一种特定变体 [21]，即 Jonker–Volgenant 算法 [16]。我们的实现借鉴了该算法在其他领域的用法 [51]，并尽可能保持简洁。与我们的方法一样，它能够处理多个集成样本，并生成最终的场景分割结果。按照基线方法处理每个样本后，每个样本包含数量不定的候选区域（检测到的对象）。每个候选区域都具有关联的 logit、二值掩码和 Softmax 分数。以一个随机样本作为初始样本，使用 Jonker–Volgenant 算法将所有其他样本与之匹配，匹配代价由二值掩码的交并比决定。成功匹配后，将匹配候选区域的面积并入二值掩码。在尽可能完成所有样本的匹配后（由于该算法是为经典指派问题设计的，因此某些样本可能仍未匹配），通过对各样本取均值，为每个候选区域生成新的 logit 掩码。随后按照基线方法应用 Softmax 和最大值索引运算符，生成像素与检测到的对象之间的关联。

C.3 Basic Performance

在论文第4.2节和表1所示的基本性能实验中，不考虑逐像素不确定性估计。相反，我们分别对本方法和匈牙利算法进行优化，使二者都能尽可能发挥最佳表现。由于基线会应用多种阈值来去除假正例，我们也对本方法进行基本程度的剪枝。具体而言，我们将每个像素的概率阈值设为 0.4，并将定义一个分割区域所需的最少像素数设为 4。不符合这些条件的图像像素不赋予类别或实例 ID，也就是说，我们不为这些像素提供预测。理想情况下，正如我们在后续实验中所做的那样，可利用逐像素不确定性估计去除这类假正例及其他错误预测。对于本方法中的实例分割部分，我们还将交并比阈值设为 0.6，并将最小合并计数设为蒙特卡洛（MC）样本数的 80%（例如，若有 15 个样本，则需合并 12 次）。最后，匈牙利算法不使用任何阈值，因为我们发现应用上述任何适用阈值都会降低性能。我们评估了两种不同的蒙特卡洛样本数：5 和 15。请注意，对于任何特定问题所需的样本数，目前尚无共识；但 MC Dropout 的原始研究 [11] 以及近期一项与其进行比较的研究 [36] 均使用了 5 到 10 个样本，不过也有研究使用了显著更多的样本 [1]。所有使用蒙特卡洛样本的运行均执行 10 次，并报告这些运行的均值和标准差。基线运行是确定性的，因此直接报告其结果。

C.4 不确定性估计的应用

在论文第4.3节和图3所示的本实验中，不使用概率阈值、合并次数阈值或最小像素数阈值；仅使用逐像素不确定性估计来决定何时从评估中移除像素。我们逐步调整各像素数值不确定性估计的阈值，并移除最不确定的像素（视为不存在真实标注），以模拟将不确定区域转交给人工审核员的情形。因此，对于不同的阈值，我们会得到不同水平的真正例、假正例和假负例，这符合标准评估流程 [19]。根据这些数据计算并绘制假发现率和真正率，即可得到类似 ROC 的曲线。对于图中未明确标注的参数，唯一相关的参数是交并比阈值（其他阈值均已禁用），我们将其设为 0.2。

请注意，图 4 使用与图3相同的数据生成，但由于处理整个数据集所有像素所需的内存量受到计算资源限制，对于使用蒙特卡洛样本的任一算法，只展示了随机选取的一次运行结果。

D 其他结果与讨论

D.1 匈牙利算法性能

表1显示，匈牙利方法的表现不如我们的方法，也不如基线。我们认为，这种较差的表现主要是由匈牙利算法用于求解经典指派问题的设计所致。正如我们在附录C.2中详细说明的，我们使用了一种广受欢迎的匈牙利算法变体，名为Jonker-Volgenant算法，它能够求解更一般化的指派问题，即使用矩形代价矩阵的情形。这正适用于我们的实验，因为检测到的候选区域（分割区域）数量可能会随样本而变化。然而，这种实现会强制采用尽可能少的指派，因此，某个样本中的一些候选区域可能无法与另一样本中的对应候选区域匹配。我们认为，对指派数量的这一最小化要求是限制匈牙利算法性能的关键因素。匈牙利方法的另一个限制是，不同样本中的候选区域之间只能存在一个最大匹配。相比之下，如果满足相应条件，我们的方法可以将一个样本中的一个候选区域与另一样本中的多个候选区域匹配。这适用于这样一种情形：网络可能会错误地将一个真实物体识别为另一样本中的多个物体；我们注意到，这种情况最常见于车辆，其中各个部件有时会被检测为多个独立的车辆。

D.2 基本性能

在表A1和表A2中，我们进一步细分了论文中的表1，将指标按干净数据和受损数据中的背景类 和 物体类 子类别进行展开。如前所述，物体类 包括那些可以彼此区分、并在数据集中据此标注的物体，例如汽车。背景类 包括无法

Segmentation Results - Clean Data										
App.	Samp.	PQ_{all} ↑	SQ_{all} ↑	RQ_{all} ↑	PQ_{things} ↑	SQ_{things} ↑	RQ_{things} ↑	PQ_{stuff} ↑	SQ_{stuff} ↑	RQ_{stuff} ↑
Baseline	N/A	43.4	79.3	53.8	48.2	79.8	59.5	36.3	**78.5**	45.3
Hungarian	5	39.5±0.1	78.8±0.1	49.0±0.1	45.3±0.2	79.7±0.1	55.9±0.2	30.7±0.1	77.4±0.2	38.7±0.1
Hungarian	15	40.7±0.1	79.1±0.1	50.4±0.1	46.3±0.1	79.9±0.1	57.0±0.2	32.2±0.1	77.8±0.1	40.4±0.1
Ours	5	44.1±0.1	79.5±0.1	**54.5±0.1**	48.9±0.1	80.5±0.1	**59.9±0.1**	36.9±0.1	77.9±0.1	46.3±0.2
Ours	15	**44.4±0.1**	**79.9±0.1**	**54.5±0.1**	**49.1±0.2**	**81.0±0.1**	59.7±0.2	**37.2±0.1**	78.2±0.1	**46.6±0.1**
Baseline	N/A	39.9	**77.5**	50.2	57.2	84.2	67.5	37.9	76.7	48.1
Hungarian	5	36.1±0.2	76.5±0.2	45.4±0.2	53.0±0.1	83.9±0.0	62.8±0.2	34.1±0.2	75.7±0.2	43.4±0.2
Hungarian	15	37.6±0.1	76.9±0.1	47.2±0.1	54.5±0.1	84.1±0.0	64.4±0.2	35.6±0.1	76.0±0.1	45.2±0.2
Ours	5	**41.8±0.1**	**76.9±0.1**	**52.7±0.2**	57.7±0.1	84.8±0.1	67.6±0.2	**39.9±0.1**	75.9±0.1	**50.9±0.2**
Ours	15	**41.9±0.1**	77.0±0.1	**52.7±0.1**	**58.1±0.1**	**85.2±0.0**	**67.9±0.1**	**39.9±0.1**	**76.0±0.1**	**50.9±0.1**
Baseline	N/A	31.1	73.9	**39.7**	22.6	74.4	29.7	**37.7**	73.6	**47.4**
Hungarian	5	27.0±0.0	73.1±0.1	34.6±0.0	21.0±0.1	73.9±0.1	27.8±0.1	31.6±0.0	72.5±0.1	39.8±0.1
Hungarian	15	28.4±0.0	73.6±0.1	36.3±0.0	21.9±0.0	74.2±0.1	28.9±0.1	33.4±0.0	73.1±0.1	42.0±0.0
Ours	5	31.1±0.0	74.2±0.1	39.5±0.0	22.9±0.0	75.0±0.1	**29.9±0.1**	37.4±0.0	**73.7±0.1**	46.9±0.0
Ours	15	**31.3±0.0**	**74.6±0.0**	39.6±0.0	**23.0±0.0**	**75.7±0.0**	29.7±0.1	37.6±0.0	**73.7±0.0**	47.2±0.0

Table A1: Basic performance. Results are presented as $\mu \pm \sigma$ over 10 runs when stochastic.

Segmentation Results - Gaussian and Shot Noise Severity 1										
App.	Samp.	PQ_{all} ↑	SQ_{all} ↑	RQ_{all} ↑	PQ_{things} ↑	SQ_{things} ↑	RQ_{things} ↑	PQ_{stuff} ↑	SQ_{stuff} ↑	RQ_{stuff} ↑
Baseline	N/A	29.0	75.4	36.8	33.6	76.4	42.8	21.9	73.8	27.8
Hungarian	5	28.1±0.0	76.2±0.2	35.7±0.1	33.2±0.1	76.7±0.4	42.1±0.1	20.3±0.1	75.3±0.2	26.1±0.2
Hungarian	15	28.9±0.1	76.3±0.2	36.7±0.1	34.0±0.1	76.6±0.0	43.0±0.1	21.2±0.1	75.7±0.4	27.1±0.2
Ours	5	**30.6±0.1**	76.8±0.2	**38.7±0.1**	**35.2±0.1**	77.4±0.3	**44.2±0.1**	23.7±0.1	75.8±0.1	30.3±0.1
Ours	15	30.5±0.1	**77.1±0.1**	38.4±0.1	35.0±0.1	**77.8±0.1**	43.6±0.1	**23.9±0.0**	**76.0±0.1**	**30.5±0.1**
Baseline	N/A	19.3	54.9	26.0	42.0	79.1	52.5	16.6	52.1	22.9
Hungarian	5	17.1±0.1	**58.3±1.2**	23.1±0.1	40.4±0.2	78.7±0.1	50.7±0.21	14.3±0.1	**55.9±1.4**	19.9±0.1
Hungarian	15	17.8±0.1	**58.1±0.2**	24.1±0.1	41.1±0.1	78.8±0.1	51.5±0.2	15.1±0.1	**55.7±0.2**	20.9±0.1
Ours	5	20.0±0.0	57.4±1.1	**27.1±0.0**	**42.5±0.1**	79.2±0.1	**53.1±0.2**	17.4±0.0	54.8±1.3	24.0±0.0
Ours	15	**20.1±0.0**	57.7±0.9	**27.1±0.0**	**42.5±0.1**	**79.4±0.1**	52.9±0.1	**17.5±0.0**	55.1±1.0	24.1±0.1
Baseline	N/A	**10.3**	66.8	**14.5**	8.9	74.1	12.1	**11.4**	61.1	**16.4**
Hungarian	5	8.4±0.0	68.8±0.2	12.1±0.0	9.5±0.0	73.5±0.1	13.0±0.1	7.6±0.0	65.2±0.3	11.4±0.0
Hungarian	15	9.2±0.0	69.2±0.2	13.1±0.0	**9.8±0.0**	73.9±0.2	**13.3±0.1**	8.7±0.0	65.5±0.2	12.9±0.0
Ours	5	10.2±0.0	69.5±0.1	14.3±0.0	9.4±0.0	74.4±0.2	12.7±0.0	10.8±0.0	**65.8±0.2**	15.6±0.0
Ours	15	10.1±0.0	**69.7±0.2**	14.2±0.0	9.0±0.0	**75.0±0.1**	12.1±0.0	11.0±0.0	65.6±0.2	15.9±0.1

Table A2: Basic performance. Results are presented as $\mu \pm \sigma$ over 10 runs when stochastic.

被识别为独立物体，或者数据集标注人员认为难以做到这一点；建筑物 或 天空 等类别就是常见的例子。

总体而言，表A1和A2得出的结论与论文中所述相同。此外，表A1显示，各方法之间的性能差异在物体类和背景类子类别中大体上都重现了；VIPER 数据集是个例外：我们的方法整体表现更好，但未能达到基线水平，也未能以微弱优势超越基线。表A2的情况也相同：背景类和物体类的性能提升或下降通常与 VIPER 数据集的情况一致，后者是个例外。我们还发现，添加噪声几乎全面摧毁了模型性能，并导致一些显著的例外情况，例如匈牙利方法在 PQ^{things} 类别中表现最佳；但此时性能极差（尤其是与干净数据相比），因此不宜根据这些例外得出任何结论。

D.3 定性结果

在图 A2 中，我们通过加入一个来自 VIPER 数据集的示例，对论文中的图 6 作了扩展，以说明这些差异并非仅限于单个数据集。



图 A2： 论文图 4 中最下方示例的预测熵不确定性热图并排对比：基线（左）与采用 15 个蒙特卡洛样本的方法（右）。较浅的色调表示不确定性更高。

D.4 执行时间

在图 A3 中，我们比较了基线、我们的方法和匈牙利算法在不同蒙特卡洛样本数量下的运行时间（每张图像所需秒数），并分别展示了中央处理器和图形处理器实现的结果。请注意，我们计入了将数据加载到图形处理器以及相关后处理步骤所需的时间，但不包括从磁盘加载模型和计算评估指标的时间。结果基于从 COCO 数据集中选取的 100 张图像计算得出，实验运行于配备 94GB 内存的 Intel Xeon E5-1620 v3 和 Nvidia GeForce RTX 2080 Ti 上。总体而言，该代码属于研究级而非生产级，因此此处展示的计时结果不能广泛

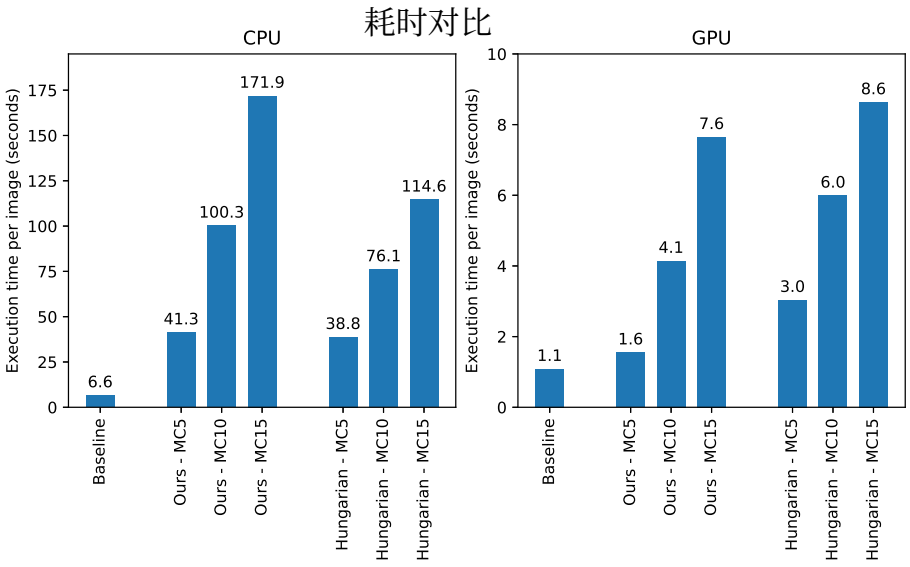


图 A3：基线方法、我们的方法和匈牙利算法在不同蒙特卡洛样本数量下的执行时间对比，结果为 5 次运行的平均值。各次运行之间的方差在视觉上可以忽略，因此未予显示。

推广，除非对可用的软件框架和相关硬件有充分了解。此外，还有一个重要注意事项：我们的匈牙利算法实现仅支持中央处理器，而我们的方法可以完全在图形处理器上运行。

从中央处理器端可以看出，我们的方法耗时略高于匈牙利方法，而且样本数量越多，差距越明显。不过，正如论文表1所示，匈牙利方法在指标表现上需要为这一优势付出代价。总的来说，除基线方法外，这里的其他方法都不太适合实际使用：在一些应用场景中，例如自动驾驶，每张图像等待超过半分钟实在太久。

从 GPU 端来看，情况稍微有趣一些。首先可以看到，蒙特卡洛样本数量与运行时间之间呈线性关系，但这种关系只在样本数量超过某个阈值后才出现。我们认为，某些环节的瓶颈和优化（例如 DETR 模型的执行）使得无论是我们的方法还是匈牙利算法，使用 5 个样本的耗时都没有比基线方法使用单个样本高出太多。这意味着，在进行一些优化后，满足时限要求的生产环境至少有可能使用几个样本。其次，匈牙利算法的表现尤其差于我们的方法，但即使使用 5 个样本，其耗时仍不到基线方法的五倍。不过，由于前述实现方面的局限，目前还无法对我们的方法与匈牙利算法进行比较并得出明确结论。

总体而言，我们可以说，使用集成和我们的方法的实际成本并不像一些人所假设的那么高（例如 [46]），但也没有另一些人所假设的那么低

(例如 [11]), 而更确切地说是在两者之间。这些发现当然取决于具体的实现方式和可用的硬件。