arXiv:2304.02654v1 [cs.LG] 5 Apr 2023

采用双监督器以高效使用大规模远程深度神经网络

MICHAEL WEISS,瑞士意大利语区大学，瑞士PAOLO
TONELLA, 瑞士意大利语区大学，瑞士

近几十年来，大规模深度神经网络（DNNs）兴起，在各种人工智能任务中达到了与人类竞争的性能。这些深度神经网络通常由数亿甚至数千亿个参数组成，太大而无法部署到或高效运行在资源受限的设备上，如手机或物联网微控制器。因此，依赖大规模深度神经网络的系统必须通过网络调用相应的模型，这导致托管和运行大规模远程模型的成本高昂，这些成本通常按使用次数收费。在本文中，我们提出了 BiSupervised，一种新颖的架构，在该架构中，系统在依赖大型远程深度神经网络之前，会尝试在小型本地模型上进行预测。一个深度神经网络监督器监控所述预测过程，并识别出可以信任本地预测的简单输入。对于这些输入，无需调用远程模型，从而节省成本，同时仅对系统整体准确率产生轻微影响。我们的架构还预见了第二个监督器来监控远程预测，并识别出即使这些也无法信任的输入，从而允许引发异常或运行回退机制。我们在四个不同的案例研究中评估了成本节省以及检测错误预测输入的能力：IMDb 电影评论情感分类、Github 问题分类、ImageNet 图像分类和 SQuADv2 自由文本问答。

CCS 概念： • **计算机系统组织** → **嵌入式系统**； 冗余；机器人技术； • **网络** →网络可靠性。

其他关键词和短语：数据集、神经网络、注视检测、文本标注

1 引言

近几十年来，现代软件系统中深度神经网络 （深度神经网络）组件的数量持续增长，这类系统被称为基于深度学习的系统 （DLS）：例如，买家在网上购物时能获得智能辅助[1], 客服工作由机器人执行并分析 [6], Github 问题可自动分类 [25] 医疗诊断也由自动化流程作出或辅助完成 [20] （有时甚至能在智能手机上运行 [69]）。

为了实现越来越高的预测准确率，许多任务中表现最佳的深度神经网络模型规模都大幅增加，如今有些模型甚至包含数万亿个参数。大型模型必须托管在服务器中心，其中一个原因是它们对硬件的要求非常苛刻。因此，面向客户端的应用必须在以下两者之间做出选择：在客户端设备（例如浏览器或移动电话）上本地运行的小型、低准确率模型，或将预测任务卸载到云端的大型模型。后者除了其他缺点之外，还会引入延迟，并给软件提供商托管其模型或使用第三方模型带来高昂的财务成本。例如，任何人都可以使用的、由非营利组织OpenAI 托管的热门GPT-3 模型 [2], ，单次请求的费用最高可达 $0.48[2]。

即使是最先进的模型，由于其输入空间通常大得难以处理且具有随机性，也偶尔会做出错误预测。在许多风险规避场景中，诸如高准确率之类的典型机器学习评估指标不足以证明纳入该模型是合理的。相反，软件架构师和软件工程师必须防范这些风险。

[1]例如：https://hellorep.ai/[2]4000 tokens，使用经过微调的 davinci 模型（每 1000 个 tokens 收费 0.12 美元）。请访问 beta.openai.com。

作者地址：Michael Weiss，michael.weiss@usi.ch，意大利瑞士大学，Via Giuseppe Buffi 13，瑞士卢加诺，6900；Paolo Tonella，paolo.tonella@usi.ch，意大利瑞士大学，ViaGiuseppe Buffi 13，瑞士卢加诺，6900。

最先进的深度神经网络模型存在缺陷，即使面对罕见且预测错误的输入，也能防止系统故障。这促成了近期软件工程文献中对各种深度神经网络监督器的提出与评估 [例如 4, 40, 54, 55, 59, 62, 64]。深度神经网络监督器是一种在运行时监控深度神经网络，并检测模型预测结果不可信的输入的软件组件。对于此类输入，系统将默认采用预定义的特定领域备份策略（例如，将请求转发给人工操作员），或拒绝生成任何输出。我们的工作表明，在资源受限设备上运行的应用中，使用此类监督器不仅可以防止系统故障，还可以降低成本和延迟。

具体而言，我们提出了 BiSupervised，这是一种新颖的软件架构，结合了廉价且快速的本地深度神经网络的优势，以及使用高精度但成本高昂的远程深度神经网络的优势。具体而言，BiSupervised 利用第一个监督器区分那些可以由小型模型在本地可靠预测的输入，仅当输入对本地模型而言过于复杂时才利用大型远程模型。随后，第二个监督器用于过滤掉高度不确定的远程预测。具体而言，我们的贡献如下：

架构我们提出了一种新颖的架构，将小型本地模型与大型远程模型相结合，使我们能够微调出最佳的成本—可靠性权衡。

成本节省评估我们通过实证评估了我们的架构所实现的成本节省（以向远程模型发出的请求数量计）与系统准确率之间的权衡。具体而言，我们评估了我们的架构是否能够在保持系统级准确率至多略低于远程模型准确率的同时，实现显著的成本节省。我们发现，对于所有案例研究，在将成本降低50%时，BiSupervised 实现的系统级准确率仅略低于大型远程模型的准确率。此外，在两个案例研究中，由于本地模型和远程模型能力之间的互补性，BiSupervised 甚至实现了超过大型远程模型的准确率。

监督性能评估 我们在监督设置中评估 BiSupervised，即包括检测那些即使远程深度神经网络预测也不可靠到足以信任的输入，因此应触发系统特定的修复策略。我们将我们系统的监督性能与本地监督模型进行比较。我们发现，在绝大多数测试配置中，BiSupervised 的监督性能与本地监督模型相似或超过后者。

2 背景

2.1 迈向巨大深度神经网络的趋势。

33 年间从 *1000* 个参数增长到 *1.6* 万亿个参数。 1989 年，LeCun等人 [34] 提出了一种深度神经网络，在手写数字分类方面达到了当时最先进的性能，标志着最早一批采用反向传播端到端训练的神经网络在现实世界中的应用之一 [26]。它仅由 1000 个参数组成，比我们如今为在具有挑战性的任务上实现最先进（SOTA）性能而使用的模型小了几个数量级。1989 年之后，我们距离深度神经网络在实际任务中的广泛应用仍相去甚远：训练深度神经网络是一项计算密集型任务，因此在很长一段时间里，支持向量机等更快速的经典人工智能方法在非平凡任务上持续取得更好的结果；例如，它们在 2010 年和 2011 [47] 仍然赢得了图像分类基准测试（ ILSVRC ）。随着一项创新的出现，这种情况发生了改变，它使深度神经网络的训练速度大幅提升，从而能够训练规模大得多的深度神经网络：从 2010 年开始，在 GPU上首次成功应用普通反向传播，实现了相对于中央处理器 50 倍的加速 [7], ，训练出了参数量最高达1200 万的网络。不久之后，拥有6000万个参数的深度神经网络 AlexNet [32] 赢得了 ILSVRC2012，标志着“大规模目标

识别”[47]。截至本文撰写时，用于同一任务的 SOTA 模型 CoCa 含有 21 亿个参数[3] [70]。

在其他领域，例如自然语言处理（NLP），我们也能看到更大的规模：GPT-2（15亿个参数） 2019 [43] 和 GPT-3（1750亿个参数） 2020 [2] 因其生成自然流畅文本的出色表现而受到全球媒体关注。在过去两年中，它们又被规模更大的模型超越，例如 Megatron（5300亿个参数，2022 年） [50]。据我们所知，目前广为人知的最大深度神经网络是 SwitchTransformer，参数量达 1.6 万亿 [12]。

大型模型的成本。 这类大型模型在推理（即预测）时所需的巨大计算资源，是其推广应用的一大挑战 [39], ，因为它们需要大量内存和能源，以及功能强大的中央处理器或图形处理器才能进行预测。如果面向客户的设备和平台资源受限，例如网页浏览器、物联网（IOT）微控制器或手机，那么部署这类模型可能并不实际，甚至不可能。因此，高精度的大规模深度神经网络通常部署在服务器中心，以提供远程推理服务。已有多种服务支持此类预测，例如 Huggingface 推理 API[4], 、用于 GPT-3[5] 的 OpenAI API，或 Google 的 Vertex AI[6]。在此类服务上，即使只对大型模型进行一次推理，也可能成本高昂；而且远程推理的成本随请求数量增加而近乎同比例上升，无法有效实现规模经济。

2.2 在资源受限环境中运行深度神经网络

更小的本地模型的优势。因此，一种截然不同的方法是在客户端设备上本地运行深度神经网络，这会限制模型的规模和复杂度，最终导致准确率较低。[7] 此类本地部署的模型具有诸多优势，包括能耗大幅降低、推理速度更快（这既得益于模型更小，也因为无需进行网络往返）、不依赖稳定的互联网连接，而且如上所述，软件提供商无需为每次推理付费。

实现高效模型的途径。因此，人们十分关注如何尽可能提高本地模型的效率，目标是让资源受限系统能够容纳更多参数，或者在尽可能减少模型参数数量的同时优化模型性能。以下介绍了近期部分相关研究和框架，并非详尽无遗，旨在推动深度神经网络模型在资源受限环境中的应用。

- 小型模型架构：许多研究提出了特定的深度神经网络架构或训练流程，旨在专门促进其在资源受限设备上的部署。例如 TinyBERT [22] 和 MobileNet [18, 19, 49]。这些模型对于各自的任务（自然语言处理和 ImageNet 分类）而言规模相对较小，旨在作为起点，通过微调适配其他特定用例，从而广泛发挥作用。

- 较小兄弟模型： 近期发布的各种新型大规模SOTA模型，通常都伴随着一组相似但规模更小的模型。用于创建旗舰级大规模SOTA模型的架构选择与创新

[3]根据 https://paperswithcode.com/sota/image-classification-on-imagenet（访问于 7 月 21, 2022
[4]
https://huggingface.co/inference-api[5]https://openai.com/api/ [6]
https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform[7]BiSupervised不仅限于分类问题，因此也可以使用准确率以外的其他指标来评估预测的质量。为简洁和易读起见，在本文讨论预测质量时，我们仅提及准确率。

同样也应用于一组规模更小的模型。这使用户能够在准确率与模型大小之间进行权衡。示例包括efficientnet [56], OPT [72] 和GPT-3[8] [2]。

- 模型调优与后处理：文献提供了一系列技术，使给定的大型深度神经网络架构适应较低的系统要求，从而可将其部署在资源受限设备上。最流行的此类方法是量化，即将全精度深度神经网络转换为低位宽整数版本，从而降低计算和内存需求 [68]。另一种技术是深度神经网络剪枝，以引导方式减少权重数量，旨在仅最小程度地降低模型准确率 [35]。
- 自定义框架：用于构建和训练深度神经网络的典型软件栈，在将深度神经网络部署到移动电话或网页浏览器等环境时无法轻松应用；除了硬件限制外，这些环境还对软件栈施加了限制。为此，人们提出了专用框架，其中最受欢迎的是tf-lite9，它可以将标准 TensorFlow 模型转换为能够部署到特定平台上的模型，例如 Android、iOS 和 Linux 边缘设备。一个专用版本tf-lite micro 则面向尺寸特别小的嵌入式设备 [8]。

自然地，这些技术中没有一种能让我们在资源受限环境中部署巨大的SOTA模型，因此在追求最佳预测准确率时，并不能取代对远程预测的需求。然而，它们确实支持和扩展了在次优准确率可接受的情况下，在本地或边缘设备上部署较小模型的可能性。

3 相关工作

3.1 边缘计算中的智能卸载

将计算密集型任务从资源受限设备委托给边缘或云的想法，称为卸载，并不限于深度神经网络。近期工作——并非特定于深度神经网络——讨论了在本地设备上智能决定哪些任务可以在本地计算，以及哪些输入必须转发到远程计算API的想法。在其他方法中，一些工作提出使用本地深度神经网络对输入进行分类以做出上述决策 [Surveys 3, 48]。因此，这些架构自然与BiSupervised表现出一些高层相似性，特别是与一级监督器。然而，我们的架构特定于深度神经网络推理任务，包括一个与本地模型紧密集成的一级监督器，使我们能够高效地计算本地预测以及是否需要远程调用的决策。具体来说，我们利用了在推理时量化深度神经网络不确定性的可能性——这种可能性并不适用于本地设备中的任意计算。据我们所知，我们的方法是第一个支持在本地资源受限设备上运行的深度神经网络进行智能卸载的方法。

3.2 网络监督

为了应对深度神经网络预测中实际上不可避免的不完美，对可能产生错误预测的输入进行检测，是机器学习安全的关键组成部分 [15]。这项任务称为监督 （有时也称为监控 [15], 预测验证 [5] 或不当行为预测 [54, 55]），能够忽略深度神经网络的预测，转而运行回退流程（有时称为修复流程 [54]）。

[8]模型未公开发布，但可通过 API 访问。[9]
https://www.tensorflow.org/lite

3.2.1 相关文献中使用的监督器。文献提供了多种可用于监督的技术。下面我们将简要概述这些技术。虽然我们会在第 4.2 小节提出一些建议，但本概述旨在帮助读者充分了解情况，从而为 BiSupervised 选择合适的监督器。

基于Softmax似然的 大多数深度神经网络分类器使用*Softmax*归一化的输出层：令 sm表示此类softmax输出层的激活，令 C 表示类别集合。我们有 $length(sm) = |C|$和 $\sum_{c \in C} sm_c = 1$。模型的预测是 $argmax_{c \in C} sm_c$，其分配的似然为 $max_{c \in C} sm_c$。[10]给定一个softmax输出数组，最简单的基于softmax的监督器，*Vanilla Softmax*或 *MaxSoftmax* [16], 如果其softmax值高于某个阈值，则简单地信任该预测。在此基础上，其他监督器使用整个softmax数组的函数，例如预测置信度分数（两个最高似然之间的差值） [73], 预测似然中的熵 [65] 或预测似然中的基尼不纯度 [13]。Outlier Exposure [17] 是一种在模型训练期间进一步增强基于Softmax似然的监督器能力的技术。

蒙特卡洛丢弃法（MC-Dropout）MC-Dropout 是实现贝叶斯神经网络的一种简单方法，即一种可以预测输出分布，而非仅给出单一“点”预测的网络 [23]。为此，它利用多个随机化的预测来估计预测结果的离散程度，从而衡量其不确定性 [14]。预测过程中的随机性来自于在丢弃层 [51] 中启用丢弃机制，即在预测时随机丢弃部分节点的激活值；这类层通常用于深度神经网络训练过程中的正则化。随后，可以采用多种技术，根据样本量化模型的不确定性，例如预测变异率或互信息。MC-Dropout 已在文献中得到广泛应用 [，例如 4, 5, 40, 73]。

集成与 MC-Dropout 类似，集成 [33] 通过衡量针对给定输入所采样预测结果之间的一致性来估计模型的不确定性。不过，在这里，预测过程中的随机性来自使用一系列不同的深度神经网络；这些网络具有相同的架构，但训练时采用了不同的权重和偏置初始化。直观而言，不同模型意见分歧越大的输入，其不确定性就越高。这里可以使用与 MC-Dropout 相同的量化指标。

惊奇充分性与上述旨在量化模型不确定性的各类方法不同，惊奇充分性（SA） [28] 旨在量化相对于监督模型中的激活轨迹而言，输入有多令人惊讶（即新颖或分布外）。惊奇充分性有不同变体，例如基于距离的 *SA*（*DSA*） [28],基于似然的 *SA*（*LSA*） *[28]* 以及基于马氏距离的 SA（MDSA） *[29, 30],*，其中一些可用于多模态场景 [31],，详细说明请参阅相应论文。

自动编码器 自动编码器监督 [52–55]采用了一种更直接的黑盒 [45] 监督方法。自动编码器是一种深度神经网络，旨在先压缩输入，再对其进行重构。如果在与受监督深度神经网络相同的数据上训练，自动编码器就能在运行时检测分布外输入：由于这类数据按定义在训练数据中未得到充分表示，因此很可能具有较高的重构误差。

其他监督器 此外，使用频率较低的方法包括，解剖器 [59] 该方法进行逐层预测并衡量其差异，*DeepRoad* [71] 一种基于模型激活轨迹的主成分分析（PCA）方法，学习断言 [37]旨在自动学习识别可能被错误预测的输入

[10]值得注意的是，sm_c 并不是对 c 实际上正确这一点的恰当近似；众所周知，如果没有经过刻意校准，深度神经网络会高估此类可能性 [16]。

来自相关激活值，并且*NIRVANA* [5] 将多个MC-Dropout 量化器的输出结合到支持向量机（支持向量机）中。

3.2.2 监督器的计算开销比较。在资源受限环境中，监督器带来的开销，即运行监督器所产生的额外计算成本，可能是一个重要的考量因素。这里可以看到，各监督器之间存在显著差异：显然，基于 Softmax 的方法速度最快，因为它们不需要额外的预测。相反，从 Softmax 数组中量化不确定性只需进行少量浮点运算。对于 MaxSoftmax，实际上只需读取数组中的一个值（预测类别的 Softmax 似然值），因此相关开销显然极小。MC-Dropout（因此 NIRVANA 也是如此）和集成则需要更多时间，因为必须收集同一输入的多个预测。尽管我们在现有工具 [65],中提供了能够高效执行此类预测的实用工具，但这仍然是资源极度受限应用中的一大障碍。

自动编码器也可以被认为是相当快速且高效的，因为只需要进行一次额外的预测（自动编码器前向传播）。此外，这种额外的预测实际上可以与监督模型的预测同时进行，甚至可以在其之前进行。因此，如果基于自动编码器的监督器判断某个给定输入的预测结果不可信，那么甚至无需进行该预测。最初提出时，与当时最先进的 DeepRoad 相比，使用基于自动编码器的监督所带来的性能提升是其主要优势之一 [54]。惊奇充分性的开销在很大程度上取决于其具体类型：当使用大型训练集时，DSA 和 LSA 都会遭遇严重的性能瓶颈。有一些方法可以缓解这些问题，例如仅使用训练集的子样本以及快速的向量化实现 [62], 但即使进行了此类改进，在许多现实世界应用中使用这些技术，尤其是在微控制器等设备上，仍然是难以接受的。在这类领域中，只有最新的 SA 变体 MDSA 能够提供足够低的开销 [66]。

3.2.3 监督器在监督性能方面的比较。 面对可用于监督的广泛技术，已进行了一系列实证研究来比较它们 [1, 63, 64, 66, 67, 73]。概括而言，它们都发现没有单一技术能作为主导监督器，即没有技术能始终优于所有其他方法。一些研究强调，集成往往——但并非总是——比所有其他技术提供更好的监督性能 [41, 64, 67] ，因为它们不仅给出准确的不确定性估计，而且通常首先提供更好的预测。基于Softmax的方法也经常提供良好的监督性能，在最近的一项研究中，我们发现不同变体的基于Softmax的监督器在检测错误分类输入的性能之间没有统计差异。因此，简单的MaxSoftmax在许多情况下可能是一个不错的选择。惊喜充分性虽然通常比基于Softmax的监督器表现稍差 [63, 66], ，但具有可应用于任何DNN的优势，而不仅仅是分类模型。最后，只要训练集在输入特征空间中相对稀疏，自动编码器就能表现良好。相反，对于特征丰富的数据集，自动编码器开始过度泛化而不适合监督，因为它们学会了也重建由相同特征组成的异常值 [46, 63]

3.2.4 监督器在鲁棒性方面的比较。 监督器能够检测出 DNN失效的根本原因非常重要，例如对抗性输入、损坏输入、无效输入或模糊输入。在最近的一项研究中，我们比较了各种监督器检测这些特定不确定性根本原因的能力 [63]。总的来说，我们发现所有监督器都存在一些弱点。例如，惊奇充分性在检测损坏输入、对抗性输入和无效输入方面通常表现良好，但无法检测分布内的模糊输入（即仅包含偶然性，

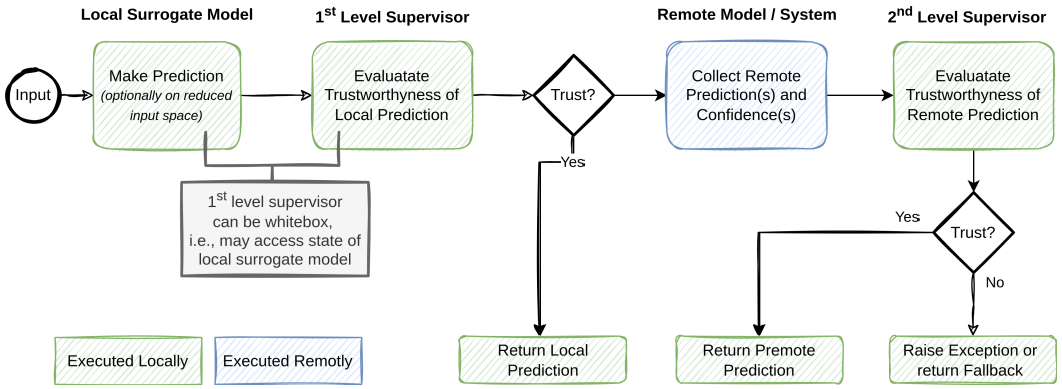


图 1. BiSuper 监督：为监督组合本地和远程预测而提出的架构

但不包括认知不确定性）。基于Softmax的监督器原则上可以检测所有根本原因，但容易受到对抗攻击的影响；这类攻击通常旨在修改输出层的值，因此可以轻易地被设计成产生置信度很高但错误的预测。

4 BISUPERVISED：我们提出的架构

BiSupervised 的概述如图 1 所示。我们架构的目标如下：(1) 尝试在小型、快速且免费的本地代理模型 （简称：本地模型）上进行预测。(2) 利用一个监督器，记为 一级监督器，以识别本地预测不可信的输入。(3) 对于这些输入，且仅针对这些输入，从大型、高精度的远程模型收集预测和置信度。(4) 识别即使远程预测也不可信的预测，基于报告的预测和置信度在 第二级监督器 中，并对这些输入发出警告或返回默认（回退机制）值。直观上，此架构仅尝试对 困难 输入进行昂贵的远程预测，其中简单的本地模型训练不足。第二级监督器还允许处理即使高精度远程模型也无法做出可信预测的输入，例如无效或模糊的输入。

算法 1 以伪代码形式概述了采用基于阈值的监督器的 BiSupervised：在第 2–3 行，本地模型进行预测，一级监督器则量化该预测的置信度。为减少执行第 2–3 行所需的计算量，可以降低相应输入的维度（例如缩小图像尺寸），以便使用更小的本地模型（第 1 行）。如果本地计算所得预测的置信度高于某个预定义阈值 t_l，则返回该预测（第 4–6 行）。否则，将原始输入发送到远程预测服务器，服务器会返回远程预测及一些元数据（例如预测 logits），第二级监督器可据此计算置信度（第 7–8 行）。如果相应的置信度足够高，则返回远程预测；否则，抛出异常（第 9–13 行）。

示例 4.1 给出了一个具体示例，说明如何在实际应用中实现和使用 BiSupervised。

示例 *4.1*（在*BiSupervised* 野生动物相机中的使用）。让我们考虑一个国家公园中的野生动物相机，安装用于监测野生动物的存在并将其报告给公园管理部门。每15秒，它拍摄一张图像，根据图像上可见的动物进行分类，例如 *no-animal*、*deer*、*wolfe*、*human* 等。使用 BiSupervised，分类工作

算法 1双重监督算法

要求： 双重监督输入 x、本地和远程置信度阈值 t_l 和 t_r

$x_{small} \leftarrow reduce_dimensionality(x)$
$pred,\ metadata \leftarrow local_model.predict\ (x_{small})$
$conf \leftarrow 1st_level_supervisor.calc_confidence(metadata)$
如果 $conf > t_l$ **则**
    return $pred$
结束 if
$pred, metadata \leftarrow collect_remote_prediction(x)$
$conf \leftarrow 2nd_level_supervisor.calc_confidence(metadata)$
如果 $conf > t_r$ **那么**
    返回 $pred$
否则
    引发 异常 ▷ 未找到高置信度的预测。
结束 if

如下：首先，野生动物相机本地尝试对图像进行分类。在此过程中，它收集完整的 Softmax 输出数组，一级监督器可以从中轻松读取模型的置信度 [16]。如果置信度高，则将预测报告给野生动物公园的服务器。否则，利用远程模型：野生动物相机将图像发送到高度可靠的第三方图像分类服务，例如 IBM 的 Watson[11]，，该服务返回其预测以及预测 logits。然后，第二级监督器根据远程模型置信度（可根据预测logits确定）确定远程预测是否可信。如果远程预测可信，则将预测报告给野生动物公园服务器。否则，向公园管理部门提交警告，以及无法分类的图像。因此，对于不同的图像，可能会发生以下场景：

- **名义图像** 假设摄像头前没有动物。本地模型以高置信度检测到无动物，且惊奇充分性较低。无需调用远程模型，从而节省费用。
- **罕见图像**假设出现了雾，这种情况在本地模型的训练集中代表性不足，因此导致对海狸的预测不可信。本地监督器检测到这个令人意外的输入，因此忽略本地预测，而是将图像转发给大规模远程模型；该模型对这类罕见情况的泛化能力更强，能够清楚地检测到雾中的鹿 。于是，公园管理部门收到了“鹿”这一报告。
- **无效图像** 一场暴风雨导致摄像头部分被泥土覆盖，使得本地模型和远程模型都无法做出有把握的预测，而这两种监督器都能检测到这一点。系统会向公园管理部门发送一条警告。在短时间内收到多条此类警告后，公园管理员会再次检查所报告的图像，并通知护林员清理摄像头。

下面，我们将更详细地解释我们架构中的每个组件。

4.1 LocalSurrogateModel 的设计

本地模型的目标是在非常受限的环境中，尽可能为更多输入做出高置信度的预测：我们的架构旨在适用于微控制器等超小型设备，因此本地模型的设计可能性有限，尤其是agaduhan

[11]https://www.ibm.com/no-en/cloud/watson-visual-recognition/pricing

关于模型参数的数量。关于此类模型创建技术的概述，请参阅第 2.2 小节；但在我们的架构中，还有一种可选择采用的额外措施，可进一步缩小模型尺寸，我们称之为输入域缩减：我们可以选择设计一个本地模型，使其处理缩减后的输入空间，而非原始的、可能很大的输入空间：我们可以降低输入空间的维度，这可能会损失一些信息，导致模型无法对少数输入作出最优预测，但通常能显著缩小本地模型的尺寸。例如，在自然语言处理任务中，我们可以使用小得多的词典，或截断输入长度。对于简单输入，输入域缩减造成的信息损失不太可能引起太多不确定性。对于其他所有输入，我们则利用针对较大原始输入域的远程预测。例如，在示例 4.1 中，我们可以用小尺寸的灰度图像训练本地代理模型，并相应地压缩其输入。简单的分类任务仍然可行，例如判断画面中是否没有动物，此时置信度较高；同时，这也使我们能够使用小得多的深度神经网络。

4.2 一级监督器的设计

原则上，我们的架构不受任何特定监督技术的限制，因此第 3.2 节中列出的所有技术都可能用作一级监督器。然而，由于一级监督器在本地、资源受限的环境中运行，因此应选择一种在推理过程中不会带来显著计算开销的技术。在我们近期的研究中 [63, 64, 66]，我们对多种监督技术进行了实证比较。结果表明，以下两种特定技术值得采用，因为它们都具有良好的监督准确率，同时只会带来可忽略不计的额外计算开销：对于本地模型输出经 Softmax 归一化的预测的分类问题，直接取最高的 Softmax 值（即预测类别对应的 Softmax 值） [16] 作为置信度分数，能够实现良好的实际监督性能。对于没有 Softmax 输出的其他模型，基于马氏距离的惊奇充分性（MDSA） [29]能够可靠地检测分布外输入。关于不同监督器的详细讨论，以及它们之间的互补性、弱点、优势和开销，请参阅第 3.2 小节。

4.3 远程模型或系统的使用及二级监督器的设计

我们的架构对远程模型的架构没有任何特定要求。事实上，远程模型被视为一个“黑盒”，其组成可能不止是一个深度神经网络：一般而言，它可能是一个功能完备、复杂的多组件系统（为简便起见，本文采用直观的术语远程模型 ）。不过，如果远程模型是一个还能在返回预测的同时给出置信度（或不确定性等相关指标）的深度神经网络，则可大大简化第二级监督。并非所有远程模型都能提供这些信息，有时还需要进行专门配置。例如，使用 OpenAI 的 GPT-3 时，可以配置请求，使其不仅返回表示模型最大后验概率的词元序列，还为返回的每个词元提供一组最有可能的选项及其负对数似然。模型预测的这些附加信息不会产生额外费用，因此可以直接实现第二级监督器。另一种做法是采用黑盒或数据盒 [45] 监督器（前者只能访问远程模型的输入/输出，后者还可以访问用于训练远程模型的训练数据）作为第二级监督器，从而筛选出连远程模型也无法处理的输入。尽管这种做法还能进一步节省成本（监督器可以在远程模型运行之前执行，从而避免收集之后会被拒绝的远程预测），但在此用例中，我们不建议使用此类黑盒和数据盒监督器：它们可能会大幅增加本地资源受限平台的负载，而且通常无法检测由

歧义导致的偶然不确定性 [63] 。最后，对于数据盒监督而言，它们需要访问远程模型的训练数据，而这类数据通常无法获得，或者规模过大而难以处理。

4.4 超高准确率性能

我们的架构的主要目标是在廉价、低准确率模型与昂贵、高准确率的远程模型之间实现权衡。因此，我们预计组合后的成本和准确率始终会处于本地模型和远程模型各自对应指标之间。然而，就准确率而言，在特定情况下，我们的架构甚至可能达到高于远程模型的准确率，同时仍然降低成本。如果本地模型和远程模型的预测能力具有互补性（即，本地模型能够正确预测的输入集合不是远程模型能够预测的输入集合的子集），并且一级监督器能够可靠地信任本地模型的正确预测，那么一些原本会被远程模型错误预测的输入，在我们的架构下将得到正确预测。较强的互补性可能会以较低的成本大幅提高系统整体准确率。我们将这种改进后的系统整体准确率称为超高准确率性能。然而，由于对能够合理运行在本地设备上的模型选择存在限制，追求本地模型和远程模型之间的互补性可能并不可行：在本文中，我们的主要目标是尽可能减少成本，同时尽可能少地降低准确率。因此，尽管我们有时可能会在实证中观察到超高准确率性能，但它的出现更多是一个（理想情况下希望具备的）副作用，而不是明确的设计目标。

4.5 监督器的阈值选择

我们强烈建议将单阈值监督器与 BiSupervised 配合使用。此类监督器基于单一置信度（或不确定性[12]）分数，以及一个可配置的阈值。当且仅当输入的置信度高于可配置的阈值时，该输入才会被视为可信。修改阈值即可逐步改变不可信/可信输入的比例。在 BiSupervised 中，仔细调优第一级监督器中的阈值，即可选择所需比例的输入来收集远程预测（从而确定成本节省）。同样，仔细调优第二级监督器中的阈值，即可调节系统拒绝进行预测的输入比例。将第二级监督器视为错误预测检测器时，调整阈值便可在假阳性与假阴性之间进行权衡。尤其是在非安全关键领域，我们可以想象，不同用户或不同用例的风险厌恶程度各不相同。因此，在风险厌恶程度较高的场景中，可以调高阈值，以允许触发更多异常，尽管这可能需要人工干预；而在其他用例中，与频繁触发异常或频繁调用回退机制所带来的负面影响（例如警告过于频繁导致用户体验下降）相比，用户可能更愿意接受一定程度的错误预测。

阈值选择的现有技术。 Stocco 等人 [54] 提出拟合名义验证集输入的分布监督分数，这些输入均被正确分类 [54]。这样便可以选择具有特定假阳性率（即预期误报百分比）的阈值，前提是生产环境中的输入分布与验证集中的相同。这种方法的一大优势是无需异常值数据集，只需一个名义验证集即可计算阈值。Dola 等人（2021年） [10]提出了另一种用于测试输入验证器的阈值选择方法。他们建议分别计算名义数据集和无效数据集的监督分数，然后选择一个阈值。

[12]我们将置信度和不确定性视为完美的互补量 [45], , 并且就深度神经网络监督而言，只需对该值应用任意保序反转变换，即可轻松将二者之一转换为另一个 [64]

该方法能够最优地区分这两个分布。尽管他们提出这些技术是用于测试数据验证的，但显然也可以将其用于监督器。不过，在此值得注意的是，现实中的无效数据集可能并不总是可用。BiSupervised不受阈值设置方式的影响。关于本文讨论的两种方法的更深入比较，请参阅 Riccio 等人（2023） [46] 。

阈值的运行时配置。 上述两种方法都将阈值视为一个在部署前确定的常量。对于 BiSupervised，我们认为在实践中，采用更动态的方式设置阈值可能会有所帮助：阈值可以视为一个配置变量，在生产环境中进行更改或调整，例如利用远程信息 [60] ，或专门根据特定用户本地设备上观察到的输入来设置。动态选择阈值能够快速调整监督器的行为，例如应对输入分布发生变化、需要采用更保守或更宽松的监督方式（例如远程预测的成本发生变化），或用户在用户设置中指定偏好的情况。为体现这种多样性，我们在论文中使用一系列不同的阈值来评估我们的架构。

4.6 BiSupervised 的扩展

BiSupervised 的架构和使用方式可以进行一些扩展，虽然这些内容超出了本文的讨论范围，但仍值得一提。首先，如果有边缘节点，就可以进行扩展，在边缘节点上增加第三个受监督的深度神经网络，其能力介于精简的本地模型与庞大的远程模型之间。这样一来，BiSupervised（实际上将变为 TriSupervised）会在无法信任本地预测时调用边缘节点上的模型。如果边缘模型也存在较高的不确定性，边缘节点还可以将预测转发给远程模型。

其次，我们注意到，主动学习可以轻松集成到BiSupervised中：由于一级监督器会检测出本地模型尚未得到充分训练的输入，因此收集这些输入、为其添加标签（甚至可以使用远程模型自动完成），并将其加入本地模型的训练集，将能够更加稳健地生成下一版本的本地模型：在我们最近的工作 [66], 中，我们已经展示了 MaxSoftmax 和 MDSA等简单的监督器——我们在 BiSupervised 中推荐将其用作一级监督器——也是实现这一目的的出色技术。因此，这样的主动学习设置本质上能够实现本地模型的持续改进，从而进一步减少未来使用 BiSupervised 的软件版本所需的远程预测占比。

5 BiSupervised 的实证评估

我们的实证评估旨在回答以下研究问题：

RQ1: 成本节省 使用 BiSupervised，我们能否减少远程预测的数量，同时仅轻微影响整体预测准确率?

研究问题2：监督性能 在给定百分比的输入用于远程预测的情况下，与独立的监督式本地模型相比，BiSupervised 在监督性能 [64],方面表现如何?

RQ1 是本研究背后的主要研究问题，旨在量化成本节省与准确率可能降低之间的权衡。该 RQ忽略二级监督，因为其重点仅在于减少高成本的远程调用，而不考虑远程生成的预测的置信度。

研究问题2旨在比较单个本地监督器的性能与 BiSupervised 所设想的两个本地/远程监督器所提供的性能。理想的监督器是能够阻止】【。

所有且仅有那些模型原本会表现出较低准确率的输入。通过研究问题2，我们检验所提出的架构除了降低远程执行成本之外，是否还能提升监督性能。

5.1 RQ1 的评估指标

我们案例研究中使用的所有监督器都基于单个不确定性分数运行；如果该分数高于某个阈值，则表示不应信任某个预测。我们以与阈值无关的方式回答 RQ1：我们的结果展示了所有可能阈值下的成本-准确率权衡，即对于任意选定比例的输入，都会收集远程预测。

请求-准确率曲线。改变一级监督器的阈值，可以在转发以进行远程预测的输入百分比（越少意味着成本越低）与较高的系统准确率（通常与大量远程预测相关）之间进行权衡。我们使用名为请求-准确率曲线的图来可视化RQ1的结果，其中 x 轴显示转发至远程模型的输入百分比，而 y 轴显示系统整体准确率（忽略二级监督）。因此，当使用采用特定一级监督器阈值的架构时，该曲线可以快速展示成本节省和准确率变化。我们还将使用相应数值标注选定的点（例如拐点和超高准确率性能点，如有），以进一步提高图表的可读性。

曲线下面积与随机基线。请求-准确率曲线可以直观地与随机基线进行比较。随机基线是在使用全部本地预测或全部远程预测时的性能之间的一条直线：如果效果良好，我们的架构所产生的曲线应大部分位于这条直线之上。为了对随机基线与 BiSupervised 进行定量比较，我们测量请求-准确率曲线下的面积（AUC-RAC）。具体而言，我们将阈值设为每个输入对应的值，计算所有准确率的平均值，并根据仅使用远程预测和仅使用本地预测时观察到的准确率进行归一化。令 $\mathrm{ACC}_{\mathrm{SYS}}(r)$ 表示给定系统和测试集的系统性能，其中有 $r \in [0,1]$ 的输入被转发给远程模型（因此，r 是请求-准确率曲线上 x 轴上的一个点），并令 n 表示测试集中的输入数量。AUC-RAC 的形式化定义如下：

$$\frac{\frac{1}{n}\sum_{i=0}^{n}\mathrm{ACC}_{\mathrm{SYS}}(\frac{i}{n}) - \mathrm{ACC}_{\mathrm{SYS}}(0)}{\mathrm{ACC}_{\mathrm{SYS}}(1) - \mathrm{ACC}_{\mathrm{SYS}}(0)} \tag{1}$$

这假设远程模型的准确率高于本地模型，即 $\mathrm{ACC}_{\mathrm{SYS}}(1) > \mathrm{ACC}_{\mathrm{SYS}}(0)$，这是我们实验的前提条件。随机监督的预期 AUC-RAC 为 0.5。需要注意的是，与 AUC-ROC 等其他 AUC 不同，由于经过了归一化，所提出的 AUC-RAC 在监督效果极差时可能低于 0，在超高准确率性能极高时也可能高于 1。这两种情况都是极端情形，我们预计不会经常出现。

5.2 研究问题2的评估指标

研究问题2考察 BiSupervised 在监督设置下的性能，即同时使用两个监督器来识别本地和远程预测均不可信的输入。为了将我们的架构与监督式本地模型进行比较，我们采用以下三项指标；这些指标是我们专门提出并设计的，用于比较不同监督模型的成对表现：

- 监督准确率 (*ACC*)：仅考虑可信预测的准确率。越高越好。通常，监督者越保守，*ACC*就越高。

表 1. 我们实证评估中所考虑的案例研究概览

Name	Domain		Architecture	#Params	Metric	Value
imdb	文本分类	Local Remote	Custom GPT-3-curie [2]	79K 13B *	Accuracy	79.4% 89.5%
issues	文本分类	Local Remote	Custom CatIss [21] **	223K 125M	微平均 F1	71.1% 82.3%
imagenet	图像类别。	Local Remote	MobileNetV3 [18] EfficientNetV2L [57]	2.9M 119M	Accuracy	67.8% 85.2%
SQuADv2 问答		Local Remote	Bert-Tiny [58] *** GPT-3-davinci [2]	6M 175B	Exact Match	28.0% 30.8%

* = 估计的 [42],** =微调后的 roberta [36],*** =微调 ed

- 接受率(Δ)：可信输入的百分比。越高越好。通常，监督器越保守， Δ 越低。
- *S-Beta* 分数(S_β)。它是（归一化后的）监督准确率与接受率的几何平均值。 β 是几何平均值中的权重参数。我们将评估 $S_{0.5}$、 S_1、S_2。

我们在 Weiss 和 Tonella [64] 中对这些指标作了详细说明。

阈值选择。因此，对于研究问题2，我们需要为第一级和第二级监督器选择阈值。对于前者，我们采用在 RQ1 中识别出的有意义拐点。对于后者，按照相关文献 [5, 54, 64]，的做法，我们选择一个阈值，使其达到目标（整体）假阳性率（FPR）。具体而言，对于先前识别出的每个拐点，我们将进行三次独立评估，分别调整第二级监督器的阈值，使 FPR 达到 0.01、0.05 和 0.1。作为基线，我们将评估仅由本地模型和监督器组成的系统的监督性能，并再次选择能够达到相同 FPR 的阈值。这也反映了我们的目标使用场景：阈值是可在运行时调整的参数，因此在部署后观察到输入（以及输入的变化）时，可以对其进行微调和调整，同时仍能与基线进行公平比较。

5.3 案例研究

我们在四个案例研究中评估了我们的架构。这些案例旨在体现多样的问题类型（文本分类、图像分类和问答），并纳入各种不同的架构作为本地模型和远程模型。表1 概述了这些案例研究，下面将对其进行介绍。关于实现细节，例如所使用的超参数和确切的数据预处理方式，请参阅我们的复现实验包，详见第 8 节。

5.3.1 IMDB 情感分类（文本分类）。IMDB 情感分类基准 [38] 是自然语言处理分类方法常用的案例研究。给定一条英文电影评论，分类器需要判断用户的评分是正面还是负面。我们的架构使用以下组件：

- **本地模型：** 对于输入域缩减，我们将输入截短为仅包含前100 个单词，并使用一个仅包含 2000 个最高频词的小型词典。我们使用一个小型模型，

由一个 Transformer 块、一个池化层和两个全连接层组成。为进行正则化，在全连接层之前添加了 dropout 层。最终的本地模型有 78’558 个参数。

- **1一级监督器：** 由于我们的本地模型是基于Softmax的分类器，因此我们可以使用原始 Softmax 作为一级监督器。
- **远程模型：** 作为远程模型，我们在 GPT-3 [2] 系列的 *curie* 模型上采用少样本学习方法。该模型估计拥有 130 亿个参数，比我们的本地模型大 160,000多倍。我们使用一个简单的提示，如清单1 所示。这使准确率达到 89.5%，比我们的本地模型高出 9% 以上。
- **2二级监督器：** 对我们的提示，任何有效的响应都只包含一个额外的 token。我们将远程请求配置为返回最可能的 100 个 token 的列表及其对应的负对数似然（可轻松转换为似然）。我们将已知的硬编码等价 token（例如 *“Negative”*、*“negative”*和*“bad”*）的似然相加。随后使用这一总似然来确定监督器是否应拒绝输入。

我们在 ESEC/FSE 2022 学生研究竞赛上展示的一项试点实验 [61], 中使用了 IMDB案例研究。尽管该试点实验取得了令人鼓舞的结果，并促使我们撰写本文，但它只回答了 RQ1 的部分问题，且采用了更简单的架构（不包含第二级监督器），因此也未涉及研究问题2。

```
请将以下电影评论分类为以下情感：负面、正面。--- 文本：我讨厌这部电影 标签：负面---文本：我喜欢这
部电影 标签：正面 --- 文本： < 我们想要分类的评论 >标签：
```

列表 1. 用于 IMDB案例研究的少样本 GPT-3 提示。

5.3.2 GitHub 问题分类器（文本分类）。GitHub 问题分类器反映了一个现实世界中的用例：当新的问题被报告到GitHub 仓库时，维护者通常会对这些问题进行分类处理（它们是疑问、错误报告、改进请求，还是其他类型）。TicketTagger [25] 是一个开源机器人，通过自动为每个输入添加标签来执行此类分类。在CI/CD 环境中运行时，这类自动分类器的“本地环境”通常受到相当严格的资源限制。最近，Kallis 等人 [24] 发布了一个相应的基准数据集，作为NLBSE’22工具竞赛的一部分：给定GitHub问题的若干属性（包括标题和描述），需要将问题分类为以下类别之一：*{bug, enhancement, question}*。该数据集是不平衡的、异质的（例如，其中包含非英语问题等异常值），并且包含一些确实模棱两可的输入（即，即使是经验丰富的开发人员也无法确定其真实标签的问题）。这使得该案例研究成为研究问题2中特别有充分动机的研究场景，因为不明确的输入最好保持未分类状态，并由仓库维护者进行人工分类处理。在采用BiSupervised后，TicketTagger 可以为被两个监督器同时拒绝的输入分配一个回退机制标签“需要人工分类”。

为了衡量此案例研究中的模型和系统性能，我们采用微平均F1分数，它会考虑此案例研究中的类别不平衡情况，具体如工具竞赛 [24] 所述。然后，我们的架构采用以下组件：

• **本地模型：** 我们使用了与 IMDb 类似的基于 Transformer 的分类器，采用相同类型的层，但选择了能够考虑该问题更高复杂度的超参数（例如，隐藏稠密层包含 100 个节点，而不是 20 个），并将词典大小增加到 2000。作为输入，我们使用与问题标题连接的问题描述，并将其截断为前 200 个单词。
• **远程模型：** 作为远程模型，我们使用CatIss，即NLBSE'22 工具竞赛的获胜模型 [21]。该模型基于经过微调的 Roberta 模型 [36]。据我们所知，这是目前适用于该问题分类数据集的最佳模型。我们预计，通过对更大规模的 Roberta 实例进行微调，可以进一步提升整体性能。

• **1st 一个 nd 第二级监督器：** 我们使用原生 Softmax 作为第一级和第二级监督器。

5.3.3 ImageNet（图像分类）。 ImageNet [47] 是文献中使用最广泛的图像分类基准之一。其图像代表1000个不同的类别，其中一些彼此密切相关（例如119种狗的品种 [11]）。因此，它代表了一个具有挑战性的机器学习问题，即使是大规模的最先进模型也尚未达到91%以上的准确率 [70]。鉴于通过标准模型仓库可以获得广泛的ImageNet模型，并且此类模型通常是具有Softmax输出层的深度神经网络，为我们的架构选择组件并不困难：

• **本地模型：** 我们使用MobilenetV3 [18] 作为本地模型。MobileNetV3是第三代也是最新一代的MobileNet ImageNet模型，专为在资源受限环境中运行而设计。MobileNetV3存在各种版本（大小不同）。我们使用'mobilenetv3_small_1.0_224'，即具有290万个参数且准确率为67.8%的模型[13]
• **远程模型：** 我们使用EfficientNetV2L [57] 作为大规模远程模型。凭借其1.19 亿个参数和 85.2 的准确率，它是 Keras应用中可用的第三大且准确率最高的 ImageNet 分类器[14]。

• **1第一和第二级监督器：** 我们使用原生 Softmax 作为第一级和第二级监督器。

5.3.4 斯坦福问答数据集（自由文本问答）。我们的第四个案例研究考察问答，具体来说是斯坦福问答数据集的第2个版本（SQuADv2）[44]。面对一篇英语维基百科文章的标题和段落，以及一列关于该段落的问题，模型应当为每个问题提供一个简短答案，该答案由段落中的子字符串组成（因此本质上是自由文本）。SQuADv2为每个问题提供了一组正确答案。例如，问题"诺曼人在哪个世纪首次获得了独立的身份？"列出了"*10*世纪"、"*10*世纪上半叶"和"*10*世纪"作为正确答案。[15] 我们使用精确匹配（*EM*） [44] 指标来评估模型和架构的性能，即预测结果与SQuADv2真实答案中的某个答案完全匹配的百分比。除了上述问题外，SQuADv2还包含无法回答的问题，即在所提供的段落中找不到答案的问题。它们非常适合在监督设置下测试模型。对于RQ1（其明确不评估第二级监督器），我们忽略这些无法回答的问题；但对于RQ2，我们将其纳入其中，因为这些问题应由第二级监督器检测出来。在此，值得一提的是，许多使用SQuADv2的相关文献认为，如果模型的不确定性高于阈值（即，如果

[13]参见https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV3Small [14] https://keras.io/api/applications/，截至2022年7月31日[15]从技术上讲，答案数据集还包含答案在段落中的位置，因此，在段落中出现多次的答案会被多次列出。在我们的评估中，我们只考虑答案，而不考虑其位置。

输入被监督器检测为无法回答）。我们认为，这在实践中并不是一种现实的设置（拒绝预测当然不等同于正确预测）。我们对RQ1（仅在可回答问题上进行精确匹配）和RQ2（ACC、Δ 和 S_β）的评估更为合适。然而，正因如此，我们在RQ1中发现的准确率，以及在RQ2中发现的监督准确率，与相关文献中针对相同模型所提及的准确率并不一致。

- **本地模型：**我们使用了 Bert-Tiny 的微调版本 [58], 这是一种流行的 基于变换器的双向编码器表示 （BERT）模型的微型版本 [9], 非常适合在资源受限环境中进行部署。Huggingface团队已完成针对 SQuADv2 数据集的微调[16]。

- **1st 级监督器：**除了表示本地模型答案的预测词元之外，我们还收集每个预测词元的不确定性分数。我们考虑了多种适合将这些分数缩减为单个分数的方法，该分数可用于基于阈值的监督，其中乘积和最小值是最明显的候选方法。实际上，我们发现文献大多使用这些不确定性的乘积。我们认为这并不适合监督：鉴于词元数量非常庞大，即使对于最简单的输入，预测概率也很可能远低于1，这自然会使由多个词元组成的答案的整体置信度呈指数级低于单词元预测的置信度。因此，在我们的实验中，我们使用所有预测词元的所有预测似然中的最小值 。

- **远程模型：**作为远程模型，我们在GPT-3 [2] 系列的*davinci*模型上使用单样本学习方法。它拥有1650亿参数，是GPT-3系列中最大的模型，因此通常被理解为*the* GPT-3模型。在首次发布时，GPT-3的作者在SQuADv2数据集上评估了该模型，结果显示出良好的性能，特别是考虑到单样本（未微调）设置。我们构建的提示与评估GPT-3时构建的提示等效。简而言之，提示以标题开始，然后是段落和一个带有正确答案的可回答问题（该问题因此从测试集中移除），最后是我们评估回答能力的问题（参见Brown等人 [2]的图G.28中的示例提示）。

- **2二级监督器：**如IMDB案例研究中所述，我们可以配置对GPT-3的请求，以收集每个预测令牌的似然度。然后，我们可以等效于一级监督器，将此似然度数组简化为单个置信度分数，从而实现基于阈值的监督。

对于 GitHub 问题分类和 ImageNet案例研究，我们在用于实验的高性能工作站上运行远程模型。因为在我们的实验设置中，我们能够负担所需的硬件，而我们提出 BiSupervised 所针对的大多数基于客户端的应用并不具备这一条件。

5.4 结果

5.4.1 评估设置的变更。上述评估设置在收集结果之前，已作为注册报告经过同行评审[17]。实验已按计划开展，但有以下两项轻微偏差：

(1) **IMDb 远程监督器:** 我们计划使用等效标记的似然之和（例如 "负面","负面" 和 "坏"）在 **100** 最可能的标记中。然而，OpenAI API 将此信息限制为 **5** 最可能标记[18] -因此我们使用

[16]https://huggingface.co/mrm8488 2 17
5/bert-tiny--finetuned-squadvhttps://dl.acm.org/journal/tosem/registered-papers [18]
https://beta.openai.com/docs/api-reference/completions/create#completions/create-logprobs

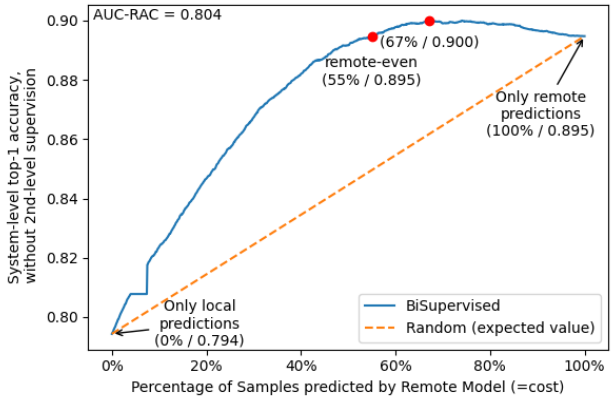


图2. IMDb的请求-准确率曲线 (RAC)

用于我们的实验。总体而言，我们观察到，后续的等效词元通常被赋予极低的概率。这是意料之中的，因为我们的提示（见清单 1）指定了一种特定拼写。因此，我们预计此处描述的更改不会对结果产生明显影响。

(2) **SQuADv2 提示：**我们预期在进行SQuADv2评估时使用与提出GPT-3的原始论文作者相同的提示（参见Brown等人 [2]的图G.28中的示例提示）。在开发过程中，我们发现使用此提示时，GPT-3经常给出冗长的完整句子答案。由于此类答案不属于SQuADv2答案数据集，这可能导致误导性的低EM（精确匹配）指标。因此，我们扩展了提示，明确指出答案应为"从背景中提取，尽可能简短"。这里，背景指的是提示中给出的上下文，与最初预期的提示一致。这个明显的修复是任何从业者实际上都会做的，因此我们决定包含它。由于GPT-3是闭源模型，具有未知的提示预处理，我们不知道当我们观察到的问题时，Brownet. al. [2] 进行实验时是否已经存在。鉴于他们不错的结果，我们假设当我们重用他们的提示时观察到的GPT-3冗长句子可能是由对GPT-3服务的故意近期更改引起的（在大多数用例中，冗长答案可能被视为更好），否则，我们会期望Brown et. al. [2] 以类似于我们的方式调整他们的提示。虽然不清楚此更改对我们的结果有何影响，但很可能我们的更改实际上为BiSupervised带来了更强的基线，并且总体上是一个更具雄心的实验设置：随着独立远程模型预期更高的EM，超越或表现同样好——如RQ1中所衡量的——成为一项更具挑战性的任务。

5.4.2 RQ1 的结果：成本节省。

IMDb。IMDB案例研究的RAC如图2所示。它展现出温和的超高准确率性能：在68%的远程请求下，系统级准确率（忽略第二级监督器）达到峰值，这比仅进行远程预测略好。在55%的远程请求下，即远程预测成本节省达到45%时，BiSupervised实现了与仅使用远程模型时相同的系统级性能。

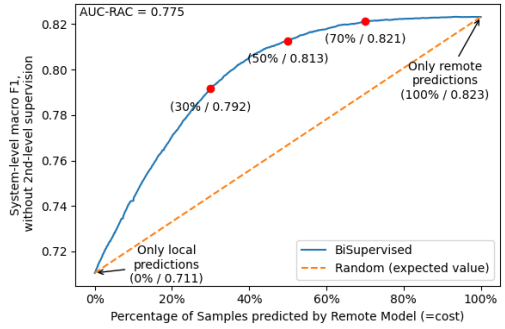


图 3. 问题的请求-准确率曲线 (RAC)

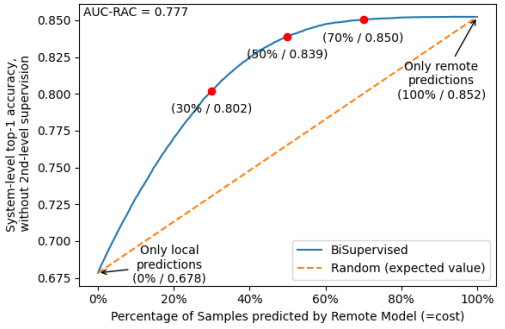
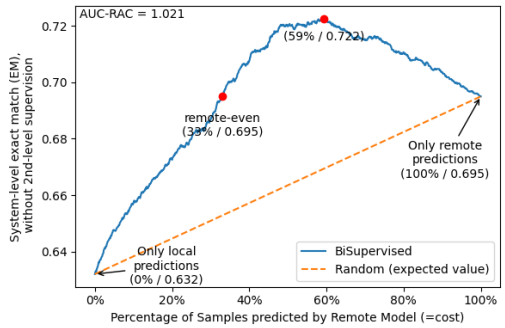
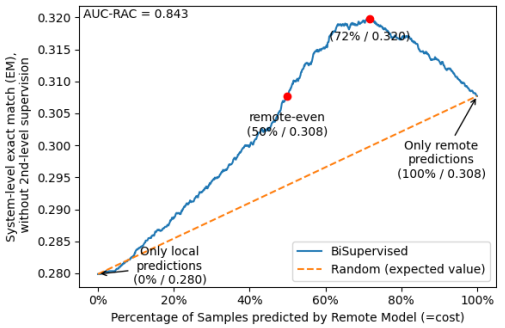


图 4. ImageNet 的请求-准确率曲线 (RAC)



(a) 仅有效（可回答）输入



(b) 所有输入

图 5. SQuADv2 数据集的请求-准确率曲线（RAC）。BiSupervised 在仅由有效（即可回答）问题组成的干净数据集以及完整数据集上均实现了超高准确率的性能。

GitHub 问题分类。GitHub 问题分类案例研究的 RAC如图 3所示。在此，BiSupervised能够在保持与独立远程模型上观察到的系统级F1 相近的同时，节省很大一部分远程预测成本：当任何阈值至少带来 60% 的远程预测时，系统级F1 基本保持不变。因此，BiSupervised 能够在几乎不对系统级 F1 产生负面影响的情况下，节省40% 的远程预测成本。总体而言，该图的AUC-RAC为0.775，因而明显优于随机基线（预期 AUC-RAC 为 0.5）。

ImageNet。 图 4 展示的 ImageNet 案例研究中的 RAC 与 GitHub Issue 案例研究中的 RAC 相似。虽然没有显著的超高准确率性能，但我们再次观察到，BiSupervised 能够节省 40 % 的远程预测成本，同时几乎不会对系统级准确率产生负面影响。与此同时，远程预测占比很低时，RAC 的陡增凸显了 BiSupervised 对目前使用 MobileNetV3（我们采用的、适用于生产环境且适配移动设备的本地模型）的应用所具有的潜力：只需将少量预测交由更大的模型（该模型可部署在边缘设备或远程服务器上）处理，就能轻松提升系统级准确率。

SQuADv2。如实证设置中所预期，在 RQ1 的研究中，SQuADv2 数据集不含无效输入的部分最为重要。相应的 RAC 如图 5a 所示。显然，BiSupervised 展现出显著的超高准确率性能：其峰值 EM 达到 0.722，比独立远程模型（0.695）高出 0.02 以上。此时只需 59% 的远程预测，因此还可节省 41% 的远程预测成本。BiSupervised 仅需 38% 的成本，就能达到独立远程模型的相同性能，显然实现了成本节省的目标。总体而言，BiSupervised 的 AUC-RAC 为 1.021。为求完整，我们还计算了包含所有无效问题的完整 SQuADv2 数据集的 RAC，如图 5b 所示。虽然这些结果略逊于不含无效输入时的结果，但图中仍显示出超高准确率性能。

RQ1（成本节省）总结

结果清楚地表明，BiSupervised 能够在保持高准确率的同时节省预测成本：

在两个案例研究中，我们实现了超高准确率性能：与仅使用大规模远程模型相比，BiSupervised 能以低得多的成本获得更高的系统级预测准确率；或者，以分别相当于远程预测成本 38% 或 55% 的代价，实现相当的性能。在另外两个模型互补性较弱的案例研究中，BiSupervised 则能节省约 40% 的远程预测成本（具体取决于案例研究），同时仍达到相当的系统级性能。

在所有案例研究中，BiSupervised 的 AUC-RAC 均显著高于随机基线对应的 AUC-RAC。

*5.4.3*研究问题*2*的结果：监督性能。研究问题2的结果列于表2、3、4、5和6中。研究问题1中观察到的RAC是平滑的，这有利于我们对研究问题2进行评估，因为一级监督器的具体选择变得不那么关键：阈值的微小偏差，即进行远程预测的输入所占的百分比的微小偏差，不会对观察到的结果产生重大影响。这使我们能够以一种通用的方式为研究问题2的评估选择一级监督器的阈值：

- 对于我们观察到超高准确率性能的案例研究（IMDb和 SQuADv2），我们采用能够带来最高性能的阈值，以及对应于尽可能少的远程预测、且总体性能等同于独立远程模型性能的阈值（记作 远程均衡）。
- 对于不具备超高准确率性能的案例（问题分类和ImageNet），我们为一级监督器选择了对应于 30%、50%和 70% 远程预测的阈值。通过同时考虑一个准确率明显降低的点（30%）和一个整体准确率与单独远程模型相当的点（70%），这些阈值能够很好地覆盖观测到的曲线。

*IMDB*案例研究。IMDb基础研究的结果如表2所示。对于所有目标假阳性率（FPR）和所有经过测试的局部阈值，即分别对应于在RQ1中实现远程均匀系统级接受率和最佳准确率的阈值，BiSupervised显然优于局部基线。在所有情况下，系统级接受率（Δ）与局部基线相当，但使用BiSupervised时，监督准确率（obj）最高可提升11%。因此，对于 Δ 和 obj的所有权重，其算术平均值，即 S_β，也优于基线。

FPR	远程请求	远程 Δ	Δ	$\overline{obj}$	$S_{0.5}$	S_1	S_2
0.01	0.00（基线）	n/a	0.98	0.80	0.83	0.88	0.94
	0.55（远程均衡）	0.96	0.98	0.91	0.92	0.94	0.96
	0.68（最佳）	0.97	0.98	0.91	0.92	0.94	0.96
0.05	0.00 (基线)	n/a	0.93	0.82	0.83	0.87	0.90
	0.55（远程均衡）	0.86	0.92	0.92	0.92	0.92	0.92
	0.68 (最佳)	0.88	0.92	0.93	0.93	0.92	0.92
0.10	0.00 (基线)	n/a	0.86	0.83	0.83	0.85	0.86
	0.55 (远程均衡)	0.76	0.87	0.93	0.91	0.90	0.88
	0.68（最佳）	0.8	0.87	0.94	0.92	0.90	0.88

表2。IMDb 的系统级监督评估

FPR	remote requests	远程 Δ	Δ	$\overline{obj}$	$S_{0.5}$	S_1	S_2
0.01	0.00 (基线)	n/a	0.98	0.72	0.76	0.83	0.91
	0.30	0.94	0.98	0.80	0.83	0.88	0.94
	0.50	0.96	0.98	0.82	0.85	0.89	0.94
	0.70	0.97	0.98	0.83	0.86	0.90	0.95
0.05	0.00 (基线)	n/a	0.91	0.74	0.77	0.82	0.87
	0.30	0.75	0.92	0.82	0.84	0.87	0.90
	0.50	0.84	0.92	0.84	0.86	0.88	0.90
	0.70	0.88	0.92	0.86	0.87	0.89	0.91
0.10	0.00 (基线)	n/a	0.84	0.77	0.78	0.80	0.83
	0.30	0.55	0.86	0.82	0.83	0.84	0.86
	0.50	0.71	0.85	0.86	0.86	0.86	0.86
	0.70	0.79	0.85	0.88	0.87	0.87	0.86

表3. 针对问题的系统级监督评估

Github 问题和 *ImageNet*案例研究。 与 RQ1 一样，Github 问题和 ImageNet案例研究在RQ2 中显示出非常相似的结果——分别列于表3和表 4 中。对于所有测试的阈值以及不同的 β 中 S_β，BiSupervised 都明显优于基线。与独立的监督式本地模型相比，在 BiSupervised 中使用远程模型不仅提高了系统级监督准确率 obj，而且在大多数情况下还带来了更高的系统级接受率 Δ。

*SQuADv2*案例研究。 仅考虑 SQuADv2 的有效答案（表 5），结果显示，与IMDb案例研究类似，BiSupervised始终明显优于基线。然而，对于涵盖所有输入的情况，即包括任何模型都不可能给出正确答案的无效输入，情况则不太明确。在这里，对于 63% 的输入，本地模型和远程模型都无法做出正确预测。随后，BiSupervised 的监督性能取决于目标假阳性率：当假阳性率较低时

FPR	remote requests	remote Δ	Δ	$\overline{obj}$	$S_{0.5}$	S_1	S_2
0.01	0.00（基线）	n/a	0.96	0.70	0.74	0.81	0.89
	0.30	0.93	0.98	0.81	0.84	0.89	0.94
	0.50	0.96	0.98	0.85	0.87	0.91	0.95
	0.70	0.97	0.98	0.86	0.88	0.92	0.95
0.05	0.00（基线）	n/a	0.86	0.75	0.77	0.80	0.84
	0.30	0.73	0.92	0.83	0.85	0.87	0.90
	0.50	0.84	0.92	0.87	0.88	0.89	0.91
	0.70	0.89	0.92	0.87	0.88	0.90	0.91
0.10	0.00（基线）	n/a	0.77	0.79	0.79	0.78	0.78
	0.30	0.53	0.86	0.84	0.84	0.85	0.86
	0.50	0.69	0.85	0.89	0.88	0.87	0.85
	0.70	0.78	0.85	0.90	0.89	0.87	0.86

表 4. ImageNet 系统级监督评估

FPR	远程请求	远程 Δ	Δ	$\overline{obj}$	$S_{0.5}$	S_1	S_2
0.01	0.00（基线）	n/a	0.97	0.64	0.69	0.77	0.88
	0.33（远程均衡）	0.94	0.98	0.70	0.74	0.82	0.91
	0.58（最佳）	0.96	0.98	0.73	0.77	0.84	0.92
0.05	0.00 (基线)	n/a	0.90	0.67	0.70	0.77	0.84
	0.33 (远程均衡)	0.79	0.93	0.71	0.75	0.81	0.88
	0.58 (最佳)	0.87	0.93	0.74	0.77	0.82	0.88
0.10	0.00 (基线)	n/a	0.81	0.70	0.72	0.75	0.79
	0.33（远程均衡）	0.62	0.87	0.72	0.74	0.79	0.84
	0.58（最佳）	0.78	0.87	0.75	0.77	0.80	0.84

表 5。SQuADv2 的系统级监督评估（仅可能）

of 0.01，BiSupervised 仍然优于局部基线，因为它成功地保持了更高的接受率 Δ，同时具有与独立本地模型相似甚至略好的系统级准确率。然而，当我们提高目标假阳性率时，特别是 FPR 为 0.1 时，使用 BiSupervised 的监督准确率的提升不如本地模型，同时保持了高得多的系统级接受率 Δ。据推测，这是因为我们通过仅操纵远程监督器的阈值同时保持一级监督器的阈值不变来实现给定的目标假阳性率：因此，我们移除了一些正确的远程预测（迅速达到目标假阳性率），而错误但被接受的本地预测则产生负面影响 obj。因此，在本案例研究中，whichoftheapproachesperforms 在监督设置中更好取决于用户的风险厌恶程度，即 βchosen in S_β：虽然 BiSupervised 在我们测试的配置中，对于 FPR 0.1，在 S_2 上优于基线，但在 $S_{0.5}$ 上未能击败基线。总体而言，SQuADv2（包括无效输入）是唯一一个结果并非结论性的案例研究

FPR	远程请求	远程 Δ	Δ	$\overline{obj}$	$S_{0.5}$	S_1	S_2
0.01	0.00（基线）	n/a	0.89	0.31	0.36	0.46	0.65
	0.51（远程均衡）	0.93	0.96	0.32	0.37	0.48	0.68
	0.72 (最佳)	0.94	0.96	0.33	0.38	0.49	0.69
0.05	0.00 (基线)	n/a	0.76	0.35	0.39	0.48	0.61
	0.51（远程均衡）	0.79	0.89	0.33	0.38	0.48	0.66
	0.72 (最佳)	0.86	0.90	0.34	0.39	0.49	0.68
0.10	0.00 (基线)	n/a	0.65	0.39	0.42	0.49	0.57
	0.51（远程均衡）	0.62	0.81	0.34	0.39	0.48	0.64
	0.72（最佳）	0.78	0.84	0.34	0.39	0.49	0.65

表6. SQuADv2（全部）的系统级监督评估

倾向于支持 BiSupervised，同时仍呈现出积极趋势：在18种评估设置（目标 FPR、S_β 和一级监督器阈值）中，BiSupervised在13种情况下的结果优于基线；在5种情况下，其得分较低S_β。

研究问题2（监督性能）总结

在大多数配置下，所有案例研究中使用 BiSupervised 获得的监督性能都高于我们的基线，即独立的监督式本地模型。这主要归因于更高的系统准确率，以及两级监督流程检测通常较难输入的能力。

5.5 威胁

内部有效性威胁：两个案例研究，即 IMDb 和 SQuADv2，使用 GPT-3 作为远程模型，该模型是在通过网络爬取收集的大量（未知）文本上训练的。这些文本中可能包含来自我们的测试集的条目（这有时称为测试集污染）。首先，我们认为这种情况不太可能发生，并且这与现实中使用 GPT-3 这类大规模远程模型的情形并无不同。因此，预计这不会影响 BiSupervised 在真实世界使用场景下的结果有效性。其次，对于 SQuADv2（GPT-3 最初就是在该数据集上进行评估的），作者测量了测试集污染的影响，发现它对观察到的能力没有任何重大影响 [2]。

受我们使用场景的启发，尤其是监督器阈值可在运行时自适应配置的可能性，我们没有在开展实验之前为评估选定阈值。在 RQ1 中，我们进行了与阈值无关的分析（AUC-RAC）并且还绘制了曲线。平滑的曲线表明，在实践中选择阈值很容易（不存在极端的本地最优解）。事实上，我们在 RQ1 中观察到的曲线没有大的波动，并且相对平滑。关于 RQ2，由于无法使用与阈值无关的指标，我们根据RQ1 的发现，为两个监督器选择了范围广泛的不同阈值，并为第二级监督器固定了FPR。这样一来，它们中的大多数最终落入小型本地最优解的可能性（这将使我们的结果失效）微乎其微。

构念效度威胁：成本与准确率权衡以及监督器性能的量化并不简单，且不存在用于此类任务的统一指标。对于前者的量化，我们定义了AUC-RAC指标，因为它提供了一种与阈值无关的量化方式，适合以请求-准确率曲线的形式进行可视化，并且人类可以轻松理解。对于后者，我们采用了S-Beta 分数，该方法由近期关于监督器评估的研究提出 [64]。

外部效度威胁：我们仅考察了四个案例研究。为选择能够代表多种应用领域的模型，我们纳入了面向文本分类、图像分类和问答任务设计的模型。然而，对于本文所考察案例研究以外的其他领域或模型，仍应通过在不同领域或模型上复现本研究，以实证方式评估我们研究结果的有效性。

5.6 超越成本节省的优势

除了架构带来的成本节省外，减少远程预测还具有许多其他优势。不过，这些优势很大程度上取决于所使用的基础设施，因此我们仅从概念层面进行讨论，而不作实证分析。

更低的延迟。对于我们的架构这类分布式系统，延迟取决于许多因素，例如网络速度、带宽、所用硬件和系统负载。我们可以合理地假设，通过网络发送请求会耗费相当多的时间，而且大规模深度神经网络进行预测所需的时间比小规模深度神经网络更长。虽然使用速度极快的硬件运行大规模模型，在一定程度上可以缓解后一问题，但前一问题是当今网络条件下无法避免的现实。在 BiSupervised 的延迟抽象模型中，令 t_l 表示在本地预处理输入、使用本地模型进行预测并运行第一级监督器所需的平均时间。同理，令 t_r 表示进行远程预测所需的平均时间（包括网络通信以及对远程预测进行监督的时间）。令 r 表示 BiSupervised 触发远程预测的输入所占比例。若满足以下条件，则 BiSupervised 的平均延迟更低：

$$t_r > t_l + rt_r \tag{2}$$

只要 $(1 - r)t_r > t_l$，这一点就成立。

我们测量了使用本地模型和远程模型所产生的延迟，以量化与使用远程模型相比，使用 BiSupervised 可能带来的延迟改善。所有测量均涵盖所需的全部计算，包括数据预处理、推理、监督以及适用情况下的网络请求。为收集结果，我们使用了一台配备 AMD Ryzen Threadripper 1920X CPU 并通过有线网络连接的工作站。整个过程中未使用 GPU。显然，测得的时间高度依赖这一配置，也会受到远程服务器负载等其他因素的影响。因此，我们报告的时间仅供说明，并不代表结果具有普遍适用性。值得注意的是，虽然我们报告了所有案例研究的时间测量结果，但这些结果对于 IMDb 和 SQuADv2 案例研究最有参考意义，因为它们使用的是通过 OpenAI API 访问的真正远程模型。对于 GitHub 问题分类和 ImageNet，大规模“远程”模型是在本地模拟的，因此没有网络开销，但也未能利用专用硬件。为求完整，我们仍报告了后者的结果。

对于所有案例研究，我们的测量结果都证实，BiSupervised 在降低延迟方面具有巨大潜力：当一级监督器的配置使得转发至远程模型的请求比例低于 85% 时，其平均延迟低于仅使用远程模型的基线，因此完全抵消了本地推理和监督所带来的额外开销。对于 ImageNet 和 SQuADv2 案例研究，这一盈亏平衡点甚至高于 96%。因此，我们在 RQ1 和研究问题2 中讨论的所有评估点，在实际应用中都能降低平均延迟。由此可见，BiSupervised 不仅能够节省成本，还能缩短平均推理时间，

Case Study	Local Only	Remote Only	Break Even (*)	Eval Points (vs. 仅远程)	备注
imdb	0.05s	0.32s	85.68%	55%：0.23秒（-30.7%） 67%：0.26秒（-18.5%）	
Issues	0.09s	1.08 秒	91.53%	30%：0.41秒（-61.5%） 50%：0.63秒（-41.5%） 70%: 0.84s (-21.5%)	在本地模拟远程模型。
ImageNet	0.02s	0.68秒	96.35%	30%：0.23 秒（-66.3%） 50%：0.36秒（-46.3%） 70%: 0.50s (-26.3%)	远程模型在本地模拟。 两个模型均使用批量推理。
SQuADv2 (仅可能)	0.02秒	0.71s	97.42%	33%：0.25秒（-64.4%） 59%：0.44 秒（-38.1%）	
SQuADv2 （全部）	0.02 秒	0.74 秒	97.53%	49%：0.39 秒（-47.6%） 71%：0.55秒（-25.9%）	

(*) 盈亏平衡点：远程调用占比达到此值时，BiSupervised 的延迟等于该延迟仅远程方法的。对于任何导致更少远程预测的一级监督器阈值，BiSupervised 比使用独立远程模型更快。因此，数值越高越好。

表 7. 每个输入的平均预测和监督延迟

同时在监督设置下提升鲁棒性，并保持准确率不变；在某些情况下，甚至还能提高系统级准确率。

降低能耗。大模型进行预测所需的能量自然比小模型更多，因为完成单次预测需要执行的算术运算大幅增加。此外，将请求发送到远程服务器也会产生额外能耗。因此，与上文讨论的延迟类似，我们可以预期，即使完全由本地完成的预测所占比例相对较低，本地推理和第一级监督所带来的少量开销（这两者均旨在提高能效），也能通过减少远程请求而得到摊销。

6 未来工作

目前，BiSupervised 假设，对于所有本地预测不确定性较高的输入，远程架构的不确定性可能更低。然而，情况并非总是如此：在某些领域，偶然不确定性，即预测所针对的输入本身固有的不确定性（例如噪声或歧义），是导致预测错误的主要原因 [27]。在这里，我们注意到，只有当本地模型表现出认知不确定性时，例如由训练数据不足或模型架构引起的不确定性，将输入转发给规模大得多、因而推测能力更强的深度神经网络才是合理的。因此，作为未来工作，我们建议尝试在本地监督器中区分偶然不确定性与认知不确定性，并且只对认知不确定性较高的输入进行远程预测，而不是对一般不确定性较高的输入进行远程预测。

我们的实验是在常规数据上进行的，即大部分属于训练数据集分布的数据。需要使用异常值数据复现我们的结果，以评估 BiSupervised 的鲁棒性。

7 结论

本文提出了 BiSupervised，这是一种新颖的架构，旨在解决资源受限设备使用大规模深度神经网络（深度神经网络）所面临的挑战——这一问题通常通过在远程服务器上进行高成本、高延迟的预测来解决。我们提出的架构结合了小规模本地模型与大规模远程模型，并包含两个监督器，用于监控预测过程，识别哪些输入可以信任本地预测，哪些则需要调用远程模型。

我们评估了 BiSupervised 在四个不同案例研究中的有效性，包括 IMDb 电影评论情感分类、Github 问题分诊、ImageNet 图像分类和 SQuADv2 自由文本问答。我们的结果表明，BiSupervised 可以在 IMDb 和 SQuADv2 上实现超高准确率性能，即 BiSupervised 实现的准确率超过了大规模远程模型的准确率。对于其他案例研究，远程推理成本可降低高达 50%，同时保持相当的系统级准确率。

为了评估 BiSupervised 在监督设置中的能力，即当识别即使远程预测也无法信任的输入时，我们将其与独立的监督式本地模型进行了比较，发现 BiSupervised 在大多数测试配置中表现相似或更好。

此外，我们测量了平均预测延迟，并表明即使避免了少量远程调用，BiSupervised 也能降低延迟，这是因为本地和远程推理之间的延迟差异很大。总体而言，我们的结果表明，BiSupervised 可以成为依赖大规模远程模型的系统的一种经济高效且准确的解决方案，特别是在资源受限的设备上。

8 数据可用性

完整的复制 包可在 Github 上获取，并存档于 Zenodo o:

- https://github.com/testingautomated-usi/bisupervised
- https://zenodo.org/record/7767383

该论文的复现包包含了我们用于计算结果的所有脚本，以及一个 Docker 文件，便于我们轻松重建运行这些脚本所需的环境。复现包还包含用于自动下载和处理第三方工件（如模型和数据集）的脚本，只要条件允许即可。我们的所有代码和工件均采用宽松的 MIT 许可证。与之对应的 TOSEM 复现计算结果（RCR）报告目前正在接受评审。

致谢

本研究部分得到了 H2020 项目 PRECRIME 的资助，该项目由 ERC 2017 年高级资助计划资助（ERC 资助协议编号：787703）。

参考文献

[1] David Berend、Xiaofei Xie、Lei Ma、Lingjun Zhou、YangLiu、Chi Xu 和 Jianjun Zhao。2020年。猫不是鱼：深度学习测试需要分布外感知。在第*35*届*IEEE/ACM* 自动化软件工程国际会议。计算机协会，美国纽约州纽约市。[2] Tom Brown、BenjaminMann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell等。2020年。语言模型是少样本学习者。神经信息处理系统进展33（2020年），1877–1901。[3] Gonçalo Carvalho、Bruno Cabral、VascoPereira 和Jorge Bernardino。2020年。使用人工智能技术在边缘计算环境中进行计算卸载。人工智能工程应用 95（2020年），103840。[4] Ferhat OzgurCatak、Tao Yue 和 ShaukatAli。2021年。自动驾驶目标检测模型中的预测表面不确定性量化。（2021年）。arXiv:2107.04991v1 [cs.CV]

[5] Ferhat Ozgur Catak、Tao Yue 和 Shaukat Ali。2022年。用于网络物理系统数据的深度学习模型中不确定性感知预测验证器。《ACM 软件工程与方法学汇刊》31 卷，4 期，文章 79（2022年7月），31 页。https://doi.org/ 10 1145 3527451 . / Cen Chen、Xiaolu Zhang、Sheng Ju、Chilin Fu、Caizhi Tang、Jun Zhou[6]和 Xiaolong Li。2019年。AntProphet：支付宝智能客服机器人背后的意图挖掘系统。。载于 IJCAI，第 8 卷。6497–6499 页。[7] Dan Claudiu Cireşan、Ueli Meier、Luca Maria Gambardella 和 J ürgen Schmidhuber。2010年。用于手写数字识别的深层、大型、简单神经网络。《神经计算》22 卷，12 期（2010年），3207–3220 页。[8] Robert David、Jared Duke、Advait Jain、Vijay Janapa Reddi、Nat Jeffries、Jian Li、Nick Kreeger、Ian Nappier、Meghna Natraj、Shlomi Regev、Rocky Rhodes、Tiezhen Wang 和 Pete Warden。2020年。TensorFlow Lite Micro：TinyML 系统上的嵌入式机器学习。https://doi.org/ 10 48550 2010 08678 . /ARXIV. . Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。. Bert：用于语言理解的深度双向 Transformer 预训练。arXiv 预印本 arXiv:1810.04805（2018年）。[10] Swaroopa Dola、Matthew B Dwyer 和 Mary Lou Soffa。2021年。使用生成模型进行神经网络的分布感知测试。载于《2021年 IEEE/ACM 第 43 届国际软件工程会议（ ICSE ）》。IEEE，226–237 页。[11] Noam Eshed。2020 年。卷积神经网络中的新颖性检测与分析。硕士论文。康奈尔大学。[12] William Fedus、Barret Zoph 和 Noam Shazeer。2021年。Switch Transformer：通过简单高效的稀疏性扩展至万亿参数模型。[13] Yang Feng、Qingkai Shi、Xinyu Gao、Jun Wan、Chunrong Fang 和 Zhenyu Chen。2020年。Deepgini：优先处理海量测试以增强深度神经网络的鲁棒性。载于《第 29 届 ACM SIGSOFT 国际软件测试与分析研讨会论文集》。177–188 页。[14] Yarin Gal 和 Zoubin Ghahramani。2016年。将 Dropout 作为贝叶斯近似：表示深度学习中的模型不确定性。载于《国际机器学习会议》。PMLR，1050–1059 页。[15] Dan Hendrycks、Nicholas Carlini、John Schulman 和 Jacob Steinhardt。2021年。机器学习安全领域尚未解决的问题。arXiv 预印本 arXiv:2109.13916（2021年）。[16] Dan Hendrycks 和 Kevin Gimpel。2016年。用于检测神经网络中错误分类和分布外样本的基线。（2016年）。arXiv:1610.02136v3 [cs.NE][17] Dan Hendrycks、Mantas Mazeika 和 Thomas Dietterich。2019年。通过暴露离群值进行深度异常检测。（ 2019 1812 04606 3 [ ] ）。arXiv: . v cs.LG Andrew Howard、Mark Sandler、Grace Chu、Liang-Chieh Chen[18]、Bo Chen、Mingxing Tan、Weijun Wang、Yukun Zhu、Ruoming Pang、Vijay Vasudevan 等。2019年。探索 MobileNetV3。载于《IEEE/CVF 国际计算机视觉会议论文集》。1314–1324 页。[19] Andrew G Howard、Menglong Zhu、Bo Chen、Dmitry Kalenichenko、Weijun Wang、Tobias Weyand、Marco Andreetto 和 Hartwig Adam。2017年。MobileNets：面向移动视觉应用的高效卷积神经网络。arXiv 预印本 arXiv:1704.04861（2017年）。[20] Ibrahim Ibrahim 和 Adnan Abdulazeez。2021年。机器学习算法在疾病诊断中的作用。《应用科学与技术趋势杂志》2 卷，01 期（2021年），10–19 页。[21] Maliheh Izadi。2022年。CatIss：一种使用 Transformer 对问题报告进行分类的智能工具。arXiv 预印本 arXiv:2203.17196（2022年）。[22] Xiaoqi Jiao、Yichun Yin、Lifeng Shang、Xin Jiang、Xiao Chen、Linlin Li、Fang Wang 和 Qun Liu。2019年。TinyBERT：为自然语言理解精简 BERT。arXiv 预印本 arXiv:1909.10351（2019年）。[23] Laurent Valentin Jospin、Hamid Laga、Farid Boussaid、Wray Buntine 和 Mohammed Bennamoun。2022年。贝叶斯神经网络实践指南——面向深度学习用户的教程。《IEEE 计算智能杂志》17 卷，2 期（2022年），29–48 页。[24] Rafael Kallis、Oscar Chaparro、Andrea Di Sorbo 和 Sebastiano Panichella。2022年。NLBSE'22 工具竞赛。载于《2022年 IEEE/ACM 第 1 届自然语言软件工程国际研讨会（NLBSE）》。IEEE，25–28 页。[25] Rafael Kallis、Andrea Di Sorbo、Gerardo Canfora 和 Sebastiano Panichella。2019年。Ticket tagger：机器学习驱动的问题分类。载于《2019年 IEEE 国际软件维护与演化会议（ICSME）》。IEEE，406–409 页。[26] Andrej Karpathy。2022年。深度神经网络：33 年前与 33 年后。http://karpathy.github.io/2022/03/14/ lecun1989/[27] Alex Kendall 和 Yarin Gal。2017年。计算机视觉贝叶斯深度学习需要哪些不确定性？《神经信息处理系统进展》第 30 卷（2017年）。[28] Jinhan Kim、Robert Feldt 和 Shin Yoo。2019年。使用惊奇充分性指导深度学习系统测试。载于《2019年 IEEE/ACM 第 41 届国际软件工程会议（ICSE）》。1039–1049 页。[29] Jinhan Kim、Jeongil Ju、Robert Feldt 和 Shin Yoo。2020年。使用惊奇充分性降低深度神经网络标注成本：自动驾驶的工业案例研究。载于《第 28 届 ACM 欧洲软件工程会议与软件工程基础研讨会联合会议论文集》。1466–1476 页。

[30] Seah Kim和 Shin Yoo。2020年。评估问答任务的惊奇充分性。载于IEEE/ACM第42届国际软件工程大会研讨会论文集。197–202。 [31] S. Kim和 S.Yoo。2021年。面向自然语言处理深度神经网络模型的输入多模态惊奇充分性分析。载于 2021年IEEE/ACM软件测试自动化国际会议（AST）。IEEE 计算机学会，美国加利福尼亚州洛斯阿拉米托斯，
80 89 10 1109 52587 2021 00017 – 。 https://doi.org/ 。 /AST 。 。 Alex
[32] 2012
Krizhevsky、Ilya Sutskever 和 Geoffrey E Hinton。 。使用深度卷积神经网络进行 ImageNet 分类。载于《神经信息处理系统进展》，F. Pereira、C.J. Burges、L. Bot- tou 和 K.Q. Weinberger（编），第25卷。Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf [33] Balaji Lakshminarayanan、AlexanderPritzel 和 Charles Blundell。2017年。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《神经信息处理系统进展》。6402–6413。[34] Yann LeCun、Bernhard Boser、John S Denker、Donnie Henderson、Richard E Howard、WayneHubbard 和Lawrence DJackel。1989年。将反向传播应用于手写邮政编码识别。《神经计算》1，4（1989），541–551。 [35] Ning Liu、Xiaolong Ma、Zhiyuan Xu、YanzhiWang、Jian Tang和 JiepingYe。2020年。Autocompress：一种面向超高压缩率的自动深度神经网络结构化剪枝框架。载于AAAI 人工智能会议论文集，第34卷。4876–4883。[36] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer和 VeselinStoyanov。2019年。RoBERTa：一种经过稳健优化的BERT 预训练方法。《CoRR》abs/1907.11692（2019）。arXiv: 1907 11692 1907 11692 。
[37]
http://arxiv.org/abs/ 。Haochuan Lu、Huanlin Xu、NanaLiu、YangfanZhou 和 XinWang。2019年。通过学习到的断言对深度学习系统进行数据健全性检查。(2019 1909 03835 3 [ ])。arXiv： 。 v
[38]
cs.LG AndrewL. Maas、RaymondE. Daly、Peter T. Pham、Dan Huang、AndrewY. Ng和 Christopher Potts。2011年。学习用于情感分析的词向量。载于第49届计算语言学协会年会论文集：人类语言技术。计算语言学协会，美国俄勒冈州波特兰，142–150。 http://www.aclweb.org/anthology/P 11 1015 -Matthew Mattina。 。使用
[39] 2020
残数系统降低神经网络推理成本。
https://community.arm.com/arm-research/b/articles/posts/reducing-the-cost-of-neural-network-inference-with-residue-number-systems [40] Rhiannon Michelmore、MartaKwiatkowska和 Yarin Gal。2018年。评估端到端自动驾驶控制中的不确定性量化。《CoRR》（ 2018 1811 06817 1 [ ] ）。arXiv:
[41]
。 vcs.LG Yaniv Ovadia 、 Emily Fertig 、 Jie Ren 、 Zachary Nado 、 D. Sculley 、 Sebastian Nowozin 、 Joshua Dillon、Balaji Lakshmi-narayanan 和JasperSnoek。2019年。你能相信你的模型不确定性吗？评估数据集偏移下的预测不确定性。《神经信息处理系统进展》（2019），13991–14002。 [42] Zilun Peng 和 Akshay Budhkar。2021年。GPT-Neo 对比 GPT-3：商业化的自然语言处理模型真的好那么多吗？
https://medium.com/georgian-impact-blog/gpt-neo-vs-gpt-3-are-commercialized-nlp-models-really-that-much-better-f4c73ffce10b[43] Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever 等。2019年。语言模型是无监督的多任务学习者。OpenAI 博客 1，8（2019），9。 [44] Pranav Rajpurkar、Robin Jia 和 Percy Liang。2018年。了解你不知道什么：SQuAD 中无法回答的问题。arXiv 预印本 arXiv:1806.03822（2018）。 [45] Vincenzo Riccio、Gunel Jahangiroba、Andrea Stocco、Nargiz Humbatova、MichaelWeiss 和Paolo Tonella。2020年。测试基于机器学习的系统：系统性映射。《实证软件工程》（2020）。 [46] Vincenzo Riccio和 Paolo Tonella。2023年。深度学习测试生成器何时以及为何会产生无效输入：一项实证研究。载于 2023年 IEEE/ACM 第45届国际软件工程大会（ICSE）。即将出版。[47] Olga Russakovsky、Jia Deng、Hao Su、JonathanKrause、Sanjeev Satheesh、Sean Ma、Zhiheng Huang、Andrej Karpathy、AdityaKhosla、Michael Bernstein 等。2015年。ImageNet大规模视觉识别挑战赛。《国际计算机视觉杂志》115，3（2015），211–252。[48] Firdose Saeik、Marios Avgeris、Dimitrios Spatharakis、Nina Santi、Dimitrios Dechouniotis、John Violos、Aris Leivadeas、Nikolaos Athanasopoulos、Nathalie Mitton 和Symeon Papavassiliou。2021年。边缘计算和云计算中的任务卸载：关于数学、人工智能和控制理论解决方案的综述。《计算机网络》195（2021），108177。[49] Mark Sandler、Andrew Howard、Menglong Zhu、Andrey Zhmoginov 和 Liang-ChiehChen。2018年。MobileNetV2：倒置残差和线性瓶颈。载于IEEE 计算机视觉与模式识别会议论文集。4510–4520。[50] Shaden Smith、Mostofa Patwary、Brandon Norick、Patrick LeGresley、Samyam Rajbhandari、Jared Casper、Zhun Liu、Shrimai Prabhumoye、GeorgeZerveas、Vijay Korthikanti等。2022年。使用 deepspeed 和 Megatron 训练 Megatron-Turing NLG530B：大规模生成式语言模型。arXiv 预印本 arXiv:2201.11990 （2022）。

[51] Nitish Srivastava、Geoffrey Hinton、Alex Krizhevsky、Ilya Sutskever 和 Ruslan Salakhutdinov。2020年。Dropout：一种防止神经网络过拟合的简单方法。《机器学习研究期刊》15，56（2020年），1929–1958。http://jmlr.org/papers/v15/srivastava14a.html [52] Andrea Stocco 和 Paolo Tonella。2020年。迈向能够持续学习的异常检测器。载于《2020年 IEEE 软件可靠性工程国际研讨会（ISSREW）》。IEEE，201–208。[53] Andrea Stocco 和 Paolo Tonella。2022 年。用于预测自动驾驶系统安全关键故障的置信度驱动加权再训练。《软件：演化与过程》34，10（2022年），e2386。[54] Andrea Stocco、Michael Weiss、Marco Calzana 和 Paolo Tonella。2020年。自动驾驶系统中的错误行为预测。载于《第42届国际软件工程大会论文集》。ACM，12页。[55] Andrea Stocco、Michael Weiss、Marco Calzana 和 Paolo Tonella。2020年。使用自动编码器预测自动驾驶系统中的安全关键错误行为。载于《第42届国际软件工程大会论文集——海报专场》。[56] Mingxing Tan 和 Quoc Le。2019年。EfficientNet：重新思考卷积神经网络的模型缩放。载于《国际机器学习大会》。PMLR，6105–6114。[57] Mingxing Tan 和 Quoc Le。2021年。EfficientNetV2：更小的模型和更快的训练。载于《国际机器学习大会》。PMLR，10096–10106。[58] Iulia Turc、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。博学的学生学得更好：论预训练紧凑模型的重要性。arXiv预印本 arXiv:1908.08962（2019年）。[59] Huiyan Wang、Jingwei Xu、Chang Xu、Xiaoxing Ma 和 Jian Lu。2020年。DISSECTOR：通过跨层剖析对深度学习应用进行输入验证。载于《第42届国际软件工程大会论文集》。ACM。[60] Rainer Weinreich 和 Reinhold Plösch。2003年。基于代理的组件系统的远程配置。《对象技术期刊》2，6（2003年），67–84。[61] Michael Weiss。2022年。CheapET-3：经济高效地使用远程深度神经网络模型。载于《第30届 ACM 欧洲软件工程大会与软件工程基础研讨会联合会议论文集》（ESEC/FSE '22）。[62] Michael Weiss、Rwiddhi Chakraborty 和 Paolo Tonella。2021年。惊奇充分性的评述与改进。载于《2021年 IEEE/ACM 深度学习测试与深度学习测试国际第三届研讨会（ DeepTest ）》。IEEE，17–24。[63] Michael Weiss、André García Gómez 和 Paolo Tonella。2022年。深度神经网络监督测试中被遗忘的危险：生成和检测真正的歧义。https://doi.org/

10 48550 2207 10495 . /ARXIV。。Michael

[64] 2021 Weiss 和 Paolo Tonella。。通过不确定性监控实现基于深度学习系统的故障安全执行。载于《2021年 IEEE 第14届软件测试、验证与确认国际会议（ICST）》。IEEE。IEEE，24–35。[65] Michael Weiss 和 Paolo Tonella。2021年。Uncertainty-Wizard：快速且用户友好的神经网络不确定性量化。载于《2021年第14届 IEEE 软件测试、验证与确认会议（ICST）》。436–441。https: //doi.org/ 10 1109 49551 2021 00056 . /ICST 。

。 Michael Weiss 和 Paolo Tonella。。简单技术在神经网络测试优先级排序和主动学习中表现出乎意料地好。[66] 2022 载于《第31届 ACM SIGSOFT 软件测试与分析国际研讨会论文集》。[67] Michael Weiss 和 Paolo Tonella。2023年。深度神经网络的不确定性量化：实证比较与使用指南。《软件测试、验证与可靠性》（2024年1月

2023 10 1002 1840 ）。

[68] https://doi.org/ 。 /stvr。Jiwei Yang、Xu Shen、Jun Xing、Xinmei Tian、Houqiang Li、Bing Deng、Jianqiang Huang 和 Xian-sheng Hua。2019年。量化网络。载于《IEEE/CVF 计算机视觉与模式识别大会论文集》。7308–7316。[69] Hang Yu、Feng Yang、Sivaramakrishnan Rajaraman、Ilker Ersoy、Golnaz Moallem、Mahdieh Poostchi、Kannappan Palaniappan、Sameer Antani、Richard J Maude 和 Stefan Jaeger。2020年。Malaria Screener：用于自动疟疾筛查的智能手机应用。《BMC 传染病》20，1（2020年），1–8。[70] Jiahui Yu、Zirui Wang、Vijay Vasudevan、Legg Yeung、Mojtaba Seyedhosseini 和 Yonghui Wu。2022年。CoCa：对比式字幕器是图像-文本基础模型。arXiv预印本 arXiv:2205.01917（2022年）。[71] Mengshi Zhang、Yuqun Zhang、Lingming Zhang、Cong Liu 和 Sarfraz Khurshid。2018年。DeepRoad：面向自动驾驶系统的基于 GAN 的变形测试与输入验证框架。载于《第33届 ACM/IEEE 自动化软件工程国际会议论文集》（法国蒙彼利埃）（ASE 2018）。ACM，美国纽约州纽约，132–142。https://doi.org/10.1145/3238147.3238187[72] Susan Zhang、Stephen Roller、Naman Goyal、Mikel Artetxe、Moya Chen、Shuohui Chen、Christopher Dewan、Mona Diab、Xian Li、Xi Victoria Lin 等。2022年。OPT：开放的预训练 Transformer 语言模型。arXiv预印本 arXiv:2205.01068（2022年）。

[73] Xiyue Zhang、XiaofeiXie、Lei Ma、Xiaoning Du、Qiang Hu、YangLiu、JianjunZhao 和MengSun。2020年。从不确定性的角度刻画深度学习软件的对抗性缺陷。载于 第*42*届国际软件工程大会论文集。ACM。