arXiv:2305.02301v2 [cs.CL] 5 Jul 2023

逐步蒸馏！凭借更少的训练数据和更小的模型规模超越更大的语言模型

Cheng-YuHsieh[1]*, **Chun-Liang Li**[2], **Chih-Kuan Yeh**[3], **Hootan Nakhost**[2],**Yasuhisa Fujii**[3], **AlexanderRatner**[1], **Ranjay Krishna**[1], **Chen-Yu Lee**[2], **TomasPfister**[2]1华盛顿大学, [2]Google Cloud AI Research, [3]Google Researchcydhsieh@cs.washington.edu

摘要

在实际应用中部署大语言模型（LLM）颇具挑战，因为它们的内存效率低且计算密集。对此，研究人员通过使用人工标签进行微调，或使用大语言模型生成的标签进行蒸馏，来训练较小的特定任务模型。然而，要取得与大语言模型相当的性能，微调和蒸馏都需要大量训练数据。我们提出了逐步蒸馏，这是一种新机制，能够：（a）训练出性能超越大语言模型的较小模型；（b）通过利用比微调或蒸馏所需更少的训练数据来实现这一目标。我们的方法提取大语言模型的推理过程，作为在多任务框架中训练小型模型的额外监督。我们在4个自然语言处理基准测试中得出三项发现：第一，与微调和蒸馏相比，我们的机制只需少得多的有标签/无标签训练样本，就能取得更好的性能。第二，与采用少样本提示的大语言模型相比，我们使用显著更小的模型规模即可取得更好的性能。第三，我们同时减少了模型规模和超越大语言模型所需的数据量；在一项基准测试中，我们微调后的7.7亿参数 T5 模型仅使用可用数据的80%，就超越了采用少样本提示的5400 亿参数 PaLM 模型，而对同一 T5 模型进行标准微调时，即使使用100%的数据集，也难以达到相同水平。[1]

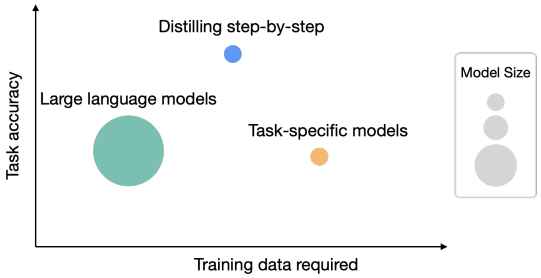

图 1：尽管大语言模型（LLM）具备出色的零样本/少样本性能，但在实际应用中部署它们颇具挑战。另一方面，传统的训练小型特定任务模型的方法需要大量训练数据。我们提出了逐步蒸馏这一新范式，从大语言模型中提取推理依据，作为信息丰富的任务知识，用于训练小型模型，从而同时降低部署模型的规模和训练所需的数据量。

1 引言

尽管大语言模型（LLM）展现出了令人瞩目的少样本能力（Brown 等人，2020年；Chowdhery 等人，2022年；Thoppilan 等人，2022年；Hoffmann 等人，2022年；Smith 等人，2022b；Zhang 等人，2022年），但由于体量庞大，这些模型难以部署于实际应用中。为单个175 0亿参数的大语言模型提供服务，至少需要 350GB 的 GPU 内存，并依赖专用基础设施（Zheng 等人，2022年）。更糟的是，如今最先进的大语言模型拥有超过 5000亿个参数（Chowdhery 等人，2022年），因此需要显著更多的内存和计算资源。这种计算需求远超大多数产品团队的承受能力，尤其是对于要求低延迟的应用而言。

为规避大型模型在部署中面临的这些挑战，实践者往往会选择部署较小的专用模型。这些较小模型通常采用两种常见范式之一进行训练：微调或蒸馏。微调使用下游人工标注数据（Howard 和 Ruder，2018）更新预训练的较小模型（例如 BERT（Devlin 等人，2020年）或 T5（Raffel 等人，2018）。蒸馏则利用大型大语言模型生成的标签来训练相同的较小模型（Tang 等人，2019年；Wang 等人，2021年；Smith 等人，2022a；Arora 等人，2022年）。遗憾的是，这些范式是以牺牲其他资源为代价来缩小模型规模：为了达到与大语言模型相当的性能，微调需要耗费大量成本获取人工标签，而蒸馏

*此项工作由作者在 Google Cloud AI Research 担任学生研究员期间完成。[1]源代码可在此处获取：https://github.com/google-research/distilling-step-by-step。

则需要大量无标注数据，而这类数据往往难以获取（Tang 等人，2019年；Liang 等人，2020年）。

在这项工作中，我们提出了逐步蒸馏（Distilling step-by-step），这是一种利用更少训练数据训练较小模型的全新简单机制。我们的机制减少了将大语言模型微调和蒸馏到较小模型规模时所需的训练数据量。该机制的核心在于改变我们对大语言模型的看法：不再将其视为产生噪声标签的来源，而是将其视为能够进行推理的智能体。大语言模型能够生成自然语言推理依据，以解释其预测标签（Wei 等，2022；Kojima 等，2022）。例如，当被问及“Jesse 的房间长 11 英尺、宽 15 英尺。如果她已经有 16 平方英尺的地毯，那么她还需要多少地毯才能铺满整个地板？”时，可以通过思维链（CoT）技术（Wei 等，2022）提示大语言模型提供中间推理依据：“面积 $=$ 长度$\times$ 宽度。Jesse 的房间面积为 11×15 平方英尺。”这些推理依据能够更好地将输入与最终答案“$(11 \times 15) - 16$”联系起来。这些推理依据可能包含相关的任务知识，例如“面积 $=$ 长度 $\times$宽度”，而这些知识原本可能需要大量数据才能让较小的特定任务模型学会。因此，我们利用提取出的这些推理依据作为额外且信息更丰富的内容，通过多任务训练方案训练小型模型，同时执行标签预测和推理依据预测任务（Raffel 等，2020；Narang 等，2020）。

逐步蒸馏使我们能够学习针对特定任务的较小模型，这些模型在使用超过 $500\times$ 更少模型参数的同时，性能优于大语言模型；与传统微调或蒸馏相比，它所需的训练样本也少得多（图 1）。我们的结果在 4 项自然语言处理基准测试中得出了三个令人振奋的经验结论。首先，与微调和蒸馏相比，我们得到的模型在各数据集上的平均表现更好，所需训练样本减少了超过 50% （最多可减少超过 85% ）。其次，我们的模型以小得多的模型规模（最多小 $2000\times$ ）超越大语言模型，大幅降低了模型部署所需的计算成本。第三，我们同时缩小了模型规模，并减少了超越大语言模型所需的数据量。我们使用一个 770M T5 模型，超越了 540B 参数大语言模型的性能；与现有微调方法所需的数据集相比，这个较小模型只需使用80% 的有标注数据集，便可达到上述效果。的微调方法所需的数据。当只有无标注数据时，我们的小模型仍能达到与大语言模型相当或更好的表现。我们仅用一个 110亿参数 T5 模型，就超越了 5400 亿参数 PaLM 的表现。我们还表明，当较小模型的表现不及大语言模型时，与标准蒸馏方法相比，逐步蒸馏能够更高效地利用额外的无标注数据，使模型达到大语言模型的表现。

2 相关工作

我们的工作利用当今大语言模型涌现出的推理能力，将其特定于任务的知识蒸馏到更小的专用模型中。我们借鉴了知识蒸馏研究，以及从人类生成的推理依据和大语言模型生成的推理依据中学习的方法。

从大型模型中进行知识蒸馏。知识蒸馏已成功用于将知识从规模更大、能力更强的教师模型迁移到更小、适用于实际应用的学生模型中（Bucilu˘a 等，2006；Hinton 等，2015；Beyer 等，2022；West 等，2021；Fu 等，2023）。它支持利用有限的带标签数据进行学习，因为较大的教师模型通常用于生成带有噪声伪标签的训练数据集（Chen 等，2020；Iliopoulos 等，2022；Wang 等，2021；Smith 等，2022a；Arora 等，2022；Agrawal 等，2022）。知识蒸馏常面临的一个局限是，创建有用的含噪训练数据集需要依赖大量无标注数据。尽管以往研究探索了使用数据增强技术来减少对数据的需求（Tang 等，2019；Liang 等，2020；Srinivas 和 Fleuret，2018；Milli 等，2019），我们提出了一种替代方法：通过蒸馏的不仅是标签，还有教师的推理依据，来减少对大量无标注数据的需求。

利用人类推理依据进行学习。 尽管利用大语言模型生成的推理依据是一个令人振奋的新研究领域，但使用人类生成的推理依据由来已久（Hase 和 Bansal，2021年）。例如，人类推理依据可用于规范模型行为（Ross 等人，2017年）；可作为额外输入来引导模型的预测（Rajani 等人，2019年）；还可用于提升模型的整体性能（Zaidan 等人，2007年；Zhang 等人，2016年；Camburu 等人，2018；Hancock 等人，2019年；Pruthi 等人，2022年）；而且人类

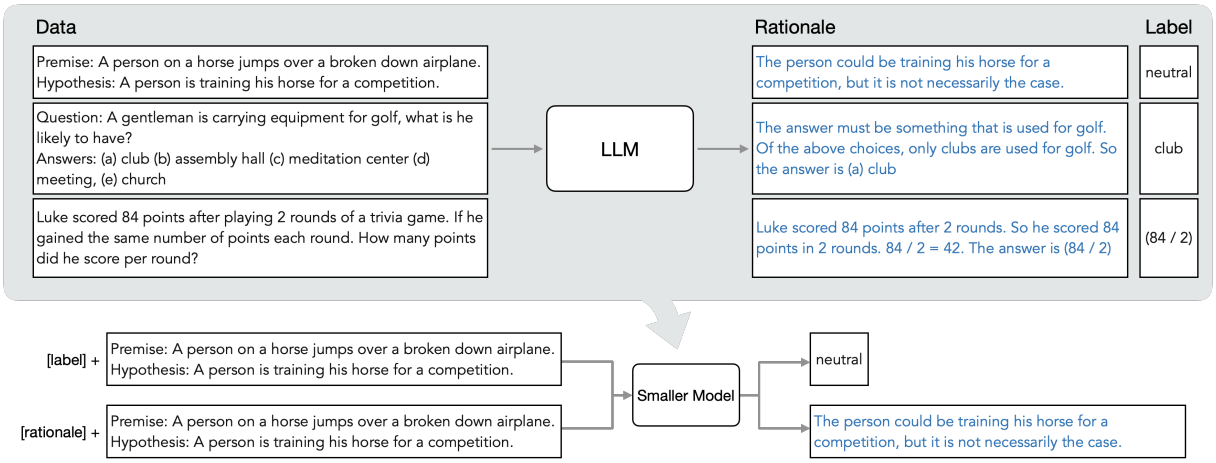

图 2：逐步蒸馏概览。首先，我们利用思维链提示从大语言模型中提取推理依据（第 3.1 节）。然后，我们在多任务学习框架中使用生成的推理依据来训练小型特定任务模型：在输入示例前添加任务前缀，并训练模型根据给定的任务前缀输出不同内容（第 3.2 节）。

类推理依据可作为金标准标签，通过生成相似的推理依据，使模型更具可解释性（Wiegreffe 等人，2021年；Narang 等人，2020年；Eisenstein 等人，2022年）。遗憾的是，获取人类推理依据的成本很高。

利用大语言模型生成的推理依据进行学习。 如今的大语言模型能够生成高质量的推理步骤，解释其预测结果（Wei 等，2022年；Kojima 等，2022年）。这些推理步骤被用于扩充大语言模型的输入提示，从而提升其少样本或零样本表现（Wei 等，2022年；Kojima 等，2022年；Wang 等，2022b）；推理步骤也被用作额外的微调数据，以“自我改进”大语言模型（Zelikman 等，2022年；Huang 等，2022年）。遗憾的是，无论如何改进大语言模型，其庞大的规模都限制了它们在大多数测试时应用中的实用性。

相比之下，我们利用生成的推理依据作为信息性监督，来训练规模更小的任务专用模型，即无需承担高昂计算或内存成本即可部署的模型。与此同时，还有几项研究提出了与我们类似的想法，即使用提取的推理依据作为监督信号（Wang 等人，2022a；Ho 等人，2022年；Magister 等人，2022年；Li 等人，2023年）。其中，PINTO（Wang 等人，2022a）依赖大语言模型在测试时生成推理依据，因此并未彻底解决部署方面的挑战。与 Ho 等人（2022年）和 Magister 等人（2022年）相比，我们进一步拓展了他们的实验，通过改变训练数据集规模、探索不同下游模型规模，并展示……

展示了我们的方法在完全无标注数据集上的有效性。

3 逐步蒸馏

我们提出了一种新范式，逐步蒸馏，利用大语言模型对其预测进行推理的能力，以数据高效的方式训练较小模型。我们的整体框架如图2所示。该范式包含两个简单步骤：首先，给定一个大语言模型和一个无标签数据集，我们提示大语言模型生成输出标签，并提供推理依据来论证这些标签。推理依据是自然语言解释，用于支持模型预测的标签（见图2）。其次，我们利用这些推理依据以及任务标签来训练较小的下游模型。直观而言，推理依据能提供更丰富、更详细的信息，说明为何某个输入会映射到特定的输出标签，而且通常包含相关的任务知识，而这些知识可能很难仅从原始输入中推断出来。

3.1 从大语言模型中提取推理依据

近期研究发现，大语言模型有一项引人注目的新兴特性：能够生成支持其预测的推Weiet al.，2022年；Kojima et al.，2022年）。尽管这些研究主要关注如何从大语言模型中激发这种推理能力（Nye etal.，2021年；Wei etal.，2022年；Kojimaetal.，2022年），我们则将其用于训练较小的下游模型。

具体而言，我们利用思维链（CoT）提示（Wei 等人，2022年）来诱导并提取

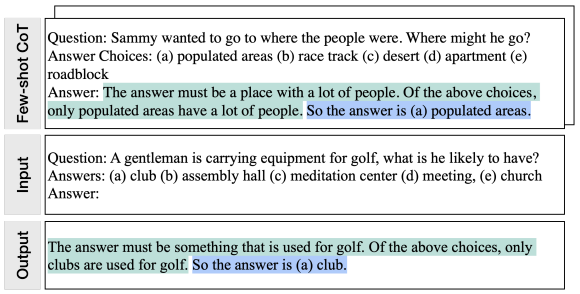

图3：我们使用少样本思维链提示，其中包含一个示例推理（以绿色突出显示）和一个标签（以蓝色突出显示），以引导大语言模型针对新的输入示例生成推理依据。

大语言模型的推理依据。如图3所示，给定一个无标签数据集 $x_i \in D$，我们首先整理出一个提示模板 p，阐明应如何解决该任务。每个提示都是一个三元组（x^{p}、r^{p}、y^{p}），其中 x^{p} 是一个示例输入，y^{p} 是其对应的标签，而 r^{p} 是用户提供的推理，用于解释为什么 x^{p} 可以被归类为 y^{p}。我们将每个输入 x_i 附加到 p，并将其作为提示大语言模型的输入，以便为每个 $x_i \in D$ 生成推理依据和标签。借助在 p 中看到的示例，大语言模型能够模仿三元组示例，为 x_i 生成推理依据 $\hat{r}_i$，并输出 $\hat{y}_i$。

3.2 使用推理依据训练较小模型

我们首先介绍当前训练特定任务模型的框架。在此框架基础上，我们进一步将推理依据纳入训练过程。形式上，我们将数据集记为 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$，其中每个 x_i 表示一个输入，而 y_i 是相应的期望输出标签。尽管我们的框架支持任意模态的输入和输出，但我们的实验将x 和 y 限定为自然语言。这个文本到文本框架（Raffel 等人，2020年）涵盖了各种自然语言处理任务：分类、自然语言推理、问答等。

标准微调与任务蒸馏。训练特定任务模型最常见的做法，是使用监督数据对预训练模型进行微调（Howard 和 Ruder，2018年）。缺少人工标注标签时，特定任务蒸馏（Hinton 等人，2015；Tang 等人，2019年）使用大语言模型教师生成带噪声的伪训练标签，$\hat{y}_i$ 以替代 y_i (Wang 等人，2021年；Smith 等人，2022a；Arora 等人，2022年）。

对于这两种情况，较小模型 f 都会通过训练来最小化标签预测损失：

$$\mathcal{L}_{\text{label}} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), \hat{y}_i), \tag{1}$$

其中 ℓ 是预测 token 与目标 token 之间的交叉熵损失。为便于阐述，我们将式 $\hat{y}_i$ 1 中的 1 过载使用，既表示标准微调情况下的人工标注标签 y_i，也表示模型蒸馏情况下由大语言模型预测的标签 $\hat{y}_i$。

基于推理依据的多任务学习。为了更明确地建立 x_i的输出与 $\hat{y}_i$的输出之间的联系，我们将提取出的推理依据 $\hat{r}_i$ 用作额外监督信号。将推理依据纳入下游模型训练过程有多种方法。一种简单直接的方法是将 $\hat{r}_i$ 作为额外输入，这也是其他同期研究提出的方法（Rajani 等人，2019年；Wang 等人，2022a）。换言之，$f(x_i, \hat{r}_i) \rightarrow \hat{y}_i$ 将文本和推理依据 $[x, r]$ 都作为输入进行训练：

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i, \hat{r}_i), \hat{y}_i). \tag{2}$$

遗憾的是，这种设计要求大语言模型先生成推理依据，然后 f 才能进行预测。部署期间仍然需要大语言模型，这限制了其部署能力。

在这项工作中，我们不将推理依据用作额外的模型输入，而是将基于推理依据的学习构建为一个多任务问题。具体而言，我们训练模型 $f(x_i) \rightarrow (\hat{y}_i, \hat{r}_i)$，使其不仅能预测任务标签，还能根据文本输入生成相应的推理依据：

$$\mathcal{L} = \mathcal{L}_{\text{label}} + \lambda \mathcal{L}_{\text{rationale}}, \tag{3}$$

其中，$\mathcal{L}_{\text{label}}$ 是式1中的标签预测损失，$\mathcal{L}_{\text{rationale}}$ 是推理依据生成损失：

$$\mathcal{L}_{\text{rationale}} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), \hat{r}_i). \tag{4}$$

推理依据生成损失使模型能够学会生成预测所需的中间推理步骤，因此可以引导模型更好地预测最终标签。这就是我们提出的逐步蒸馏。与公式2 相比，推理依据 $\hat{r}_i$ 在测试时并非必需，因此无需在测试时使用大语言模型。

我们在输入示例前添加“任务前缀”（[标签]，[推理依据]），并训练小模型在提供$\hat{}y_i$ 时输出[标签]，以及生成$\hat{}r_i$ 及其[推理依据]（Raffel 等人，2020年）。

4 实验

我们通过实证验证了逐步蒸馏的有效性。首先，与标准微调和任务蒸馏方法相比，我们展示了逐步蒸馏仅需少得多的训练样本即可取得更好的性能，从而显著提升学习小型特定任务模型时的数据效率（第 4.1节）。其次，我们展示了逐步蒸馏能够以小得多的模型规模超越大语言模型的性能，与大语言模型相比大幅降低部署成本（第 4.2节）。第三，我们研究了逐步蒸馏要超越大语言模型所需的最少资源，包括训练样本数量和模型规模。我们展示了逐步蒸馏通过使用更少的数据和更小的模型超越了大语言模型，同时提升了数据与可部署性效率（第 4.3节）。最后，我们进行了消融研究，以了解逐步蒸馏框架中不同组件和设计选择的影响（第4.4节）。

设置。 在实验中，我们将 540B PaLM 模型（Chowdhery 等人，2022年）作为大语言模型。对于特定任务的下游模型，我们使用 T5 模型（Raffel 等人，2020年），并使用从公开来源获得的预训练权重初始化这些模型[2]。对于思维链提示，我们在可用时遵循 Wei 等人（2022年）的方法，并为新的数据集整理自己的示例。更多实现细节请参见附录A.1。

数据集。 我们在 4 个涵盖 3 项不同自然语言处理任务的热门基准数据集上开展实验：用于自然语言推理的 *e-SNLI*（Camburu 等人，2018年）和*ANLI*（Nie 等人，2020年）；用于*CQA*（Talmor 等人，2019年；Rajani 等人，2019年）的常识问答；以及用于算术数学应用题的 *SVAMP*（Patel 等人，2021年）。更多数据集详情请参见附录 A.2。

4.1 减少训练集中的数据量

我们将逐步蒸馏与学习特定任务模型时最常用的两种方法进行比较：

(1) 标准微调 适用于有人工标注示例的情况；(2) 标准任务蒸馏 适用于只有未标注示例的情况。具体而言，标准微调指的是当前普遍采用的预训练后微调范式，即通过标准标签监督，利用真实标签对模型进行微调（Howard 和 Ruder，2018年）。另一方面，当只有未标注示例时，标准任务蒸馏会将教师大语言模型预测的标签视为真实标签，以此训练特定任务模型（Hinton 等人，2015；Chen 等人，2020年；Wang 等人，2021年；Smith 等人，2022a；Arora 等人，2022年）。

在下面这组实验中，我们将特定任务模型固定为 2.2 亿参数的 T5-Base 模型，并比较不同方法在可用训练样本数量各异时所取得的任务表现。

逐步蒸馏仅需少得多的标注示例，便能超越标准微调。使用人工标注示例进行微调时，图4 显示，使用数量不同的标注示例时，逐步蒸馏始终能取得比标准微调更好的性能。此外，我们看到，逐步蒸馏仅需少得多的标注示例，便能达到与标准微调相同的性能。具体而言，仅使用完整 e-SNLI 数据集的12.5% 时，逐步蒸馏便能超越使用完整数据集100% 进行训练的标准微调。类似地，在 ANLI、CQA 和 SVAMP 上，我们分别减少了75%、25%和20% 的训练样本需求，即可超越标准微调。

逐步蒸馏仅需少得多的未标注示例，性能便优于标准蒸馏。仅有无标注数据可用时，我们将逐步蒸馏与标准任务蒸馏进行比较。在图 5中，我们观察到整体趋势与微调设置相似。具体而言，在使用不同数量无标注数据的情况下，逐步蒸馏在全部 4 个数据集上的表现均优于标准任务蒸馏。我们还发现，逐步蒸馏只需少得多的无标注数据，便能胜过标准任务蒸馏。例如，在 e-SNLI 数据集上，我们仅需使用完整无标签数据集的 12.5%，其表现便优于标准任务蒸馏使用 100% 训练样本所达到的表现。

[2]https://huggingface.co/

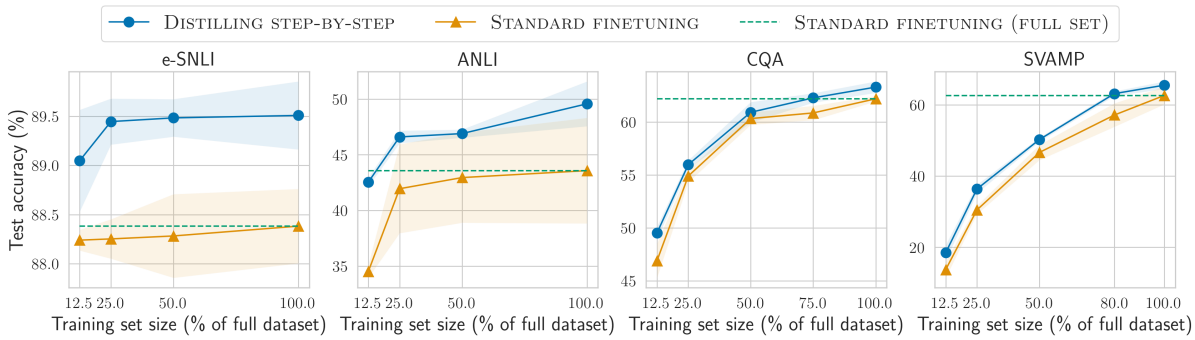

图 4：我们使用 2.2亿参数 T5 模型，在不同规模的人工标注数据集上比较逐步蒸馏和标准微调。在所有数据集上，逐步蒸馏都能够超越使用完整数据集训练的标准微调，同时仅需使用少得多的训练样本（例如，仅使用完整 e-SNLI 数据集的 12.5%）。

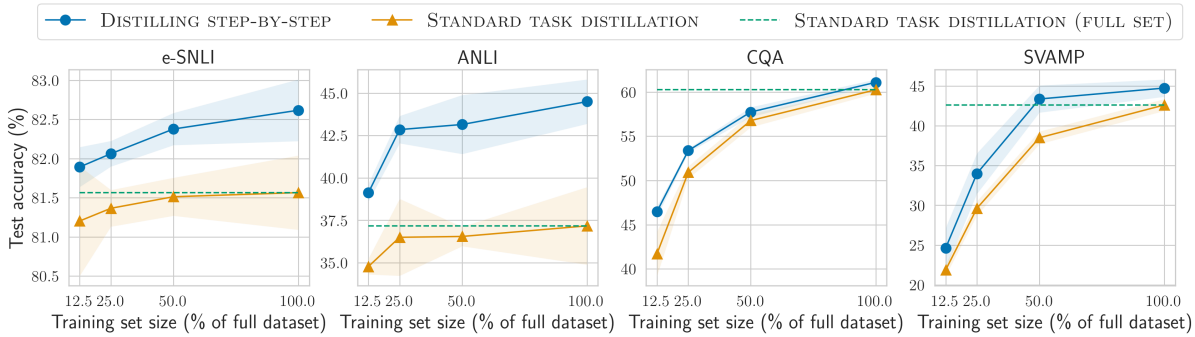

图 5：与上面的图类似，我们使用 2.2亿参数 T5 模型，在不同规模的无标注数据集上比较逐步蒸馏和标准任务蒸馏。逐步蒸馏仅使用完整无标注数据集的一小部分，就能够超越标准任务蒸馏（例如，在 ANLI 数据集上仅使用 12.5%）。

4.2 缩小模型规模

在以下一组实验中，我们固定训练集的大小（使用100% 的数据集），并比较通过逐步蒸馏和标准方法训练的不同规模小型 T5 模型与大语言模型。具体来说，我们考察3种不同规模的 T5 模型，即 220M T5-Base、770M T5-Large 和 11B T5-XXL。对于大语言模型，我们纳入两种基线方法：（1）少样本-思维链（Wei 等人，2022年），以及（2）PINTO 微调（Wang 等人，2022a）。少样本思维链直接利用思维链示例提示 5400 亿参数 PaLM，在预测最终标签之前生成中间步骤，无需对大语言模型进行进一步微调。PINTO 微调指的是我们将 Wang 等人（2022a）的方法扩展到处理问答以外的任务，而这些任务并未由 Wang 等人（2022a）研究。在这里，我们在 PaLM 模型生成的输出之上微调一个 220M T5-Base 模型，这可视为一种通过增加参数来微调大语言模型的方法（Zhang 等人，2020年；Lester 等人，2021年）。

我们展示了以下条件下的实验结果：

在图中展示的两种主要场景：可使用有标注数据集或无标注数据集6 和图 7，分别。我们根据各方法用于预测的部署模型规模绘制结果（x轴），并展示其相应的任务表现（y轴）。

在所使用的不同模型规模下，逐步蒸馏均优于标准基线方法。在图6 和图7 中，我们分别看到，逐步蒸馏在所有规模的 T5 模型上都始终优于标准微调和标准蒸馏。在 ANLI 上，这种提升最为明显：逐步蒸馏在任务准确率上平均分别超越标准微调和标准蒸馏8% 和13% 。

逐步蒸馏通过使用小得多的特定任务模型，超越了大语言模型。在图6 当有人工标注数据集可用时，逐步蒸馏在所有4个所考虑的数据集上，始终优于少样本思维链和 PINTO 微调，而且使用的 T5 模型小得多。例如，我们可以在 e-SNLI 上使用比 5400 亿参数 PaLM 模型少

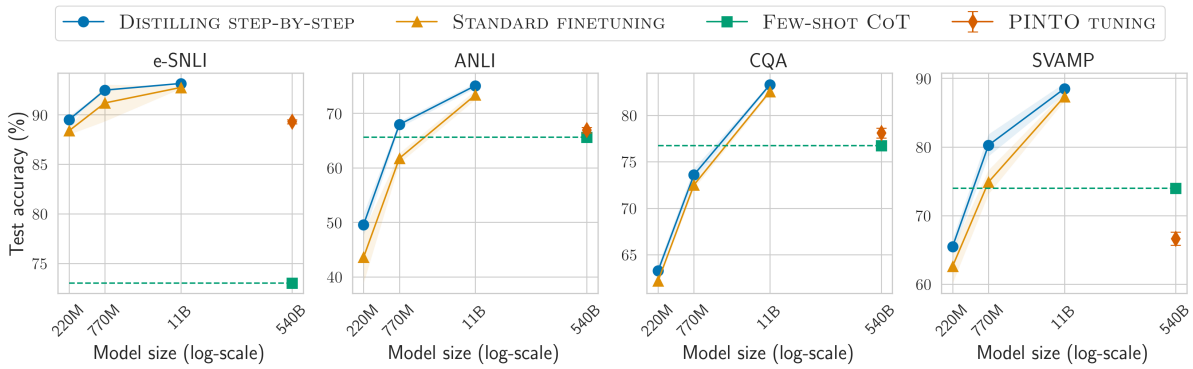

图 6：我们在不同规模的 T5 模型上，使用完整的人工标注数据集进行逐步蒸馏和标准微调，并将其性能与大语言模型基线（即少样本思维链和 PINTO 微调）进行比较。逐步蒸馏能够使用小得多的模型超越大语言模型基线，例如，在 ANLI 上使用的模型规模小了 700×。标准微调使用相同规模的模型时，无法达到大语言模型的性能。

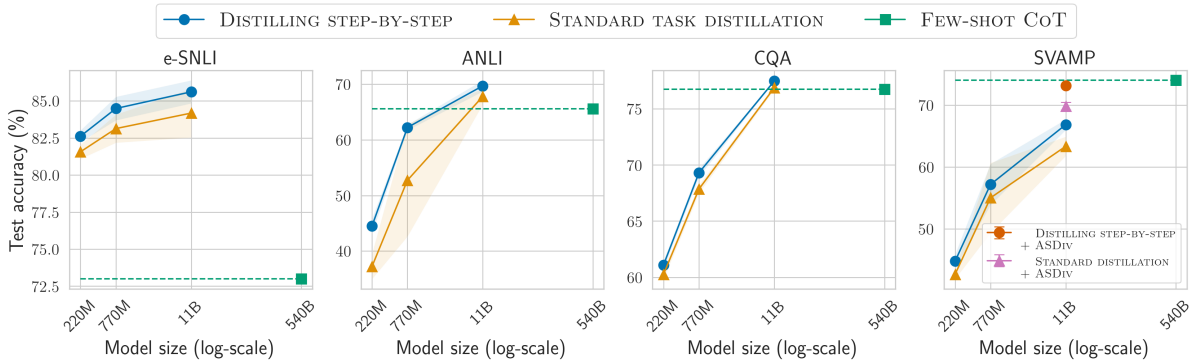

图 7：我们使用无标注数据集，在不同规模的 T5 模型上进行逐步蒸馏和标准任务蒸馏，并将它们与少样本思维链进行比较。在 e-SNLI 上，逐步蒸馏使用的模型规模小了 2000×，在 ANLI 和 CQA 上则小了 45×，但性能仍优于少样本思维链。在 SVAMP 上，通过添加来自 ASDiv 的未标注示例，我们缩小了与少样本思维链之间的差距，而标准蒸馏仍难以赶上。

小220M（小了 2000×）的 T5 模型；在 ANLI 和 SVAMP 上使用比770M（小了 700× ）的 T5 模型；在 CQA 上使用比11B（小了 45×）的 T5 模型，取得了比 5400 亿参数 PaLM 模型使用少样本思维链更好的表现。即使进一步使用可用的有标注数据，通过 PINTO 微调 5400 亿参数 PaLM 模型，这些结果依然成立[3]。

在图 7中，仅利用未标注示例，逐步蒸馏在 4 个数据集中的 3 个上也优于教师大语言模型。具体而言，逐步蒸馏使用 11B T5（其规模不到 PaLM 的3% ），超越了 540B PaLM 模型的少样本思维链表现。对于蒸馏模型表现较差的 SVAMP，我们推测这是由于该数据集中的数据点相对较少（即 800）所致。为此，我们提出用额外的未标注示例扩充数据集，以缩小性能差距，如下所示。

利用无标注数据进行增强，能进一步提升逐步蒸馏的效果。 我们使用来自 *AS-Div* 数据集的未标注示例来扩充 SVAMP 训练集（Miao 等人，2020年）。ASDiv 数据集共包含 2,305 个示例，每个示例都是一道与 SVAMP 中类似的数学应用题。在图 7 中，我们展示了使用 ASDiv 扩充训练集后，逐步蒸馏和标准任务蒸馏在 SVAMP 上的表现，所用模型为 11B T5。可以看到，数据增强显著提升了逐步蒸馏和标准任务蒸馏的表现。然而，即便加入了未标注示例，标准任务蒸馏的表现仍不及少样本思维链。另一方面，逐步蒸馏能够更高效地利用新增示例的价值，达到少样本思维链的同等表现；而且再次只需使用规模不到 3% 的 540B PaLM。

[3]我们注意到，PETuning 方法可能优于 PINTO 微调。然而，这些方法在训练和部署阶段都需要大量资源，而这并非本文的重点。

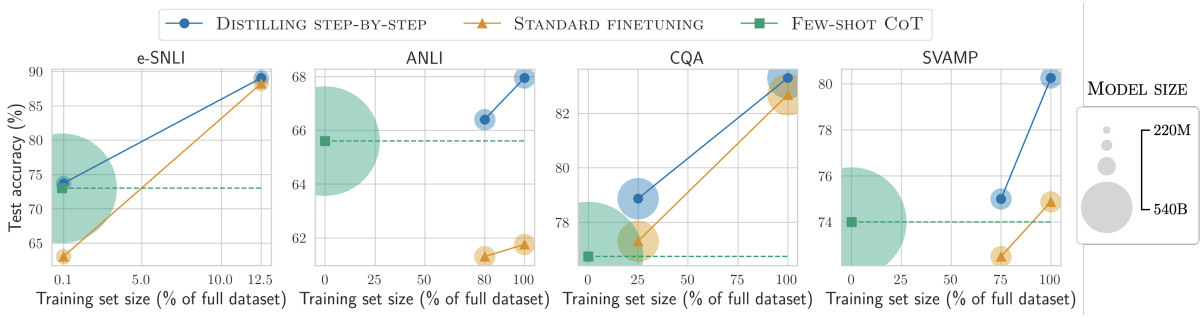

图8：我们通过粗粒度搜索，展示了逐步蒸馏要超越大语言模型的少样本思维链所需的最小 T5 模型规模和最少人工标注示例数量。逐步蒸馏不仅能够使用小得多的模型超越少样本思维链，而且与标准微调相比，只需少得多的训练样本即可达到这一效果。在 ANLI 上，我们仅使用 80% 的数据集，便通过一个 770M 参数模型超越了大语言模型的思维链，而标准微调即使使用 100% 的数据集，也难以达到同等水平。

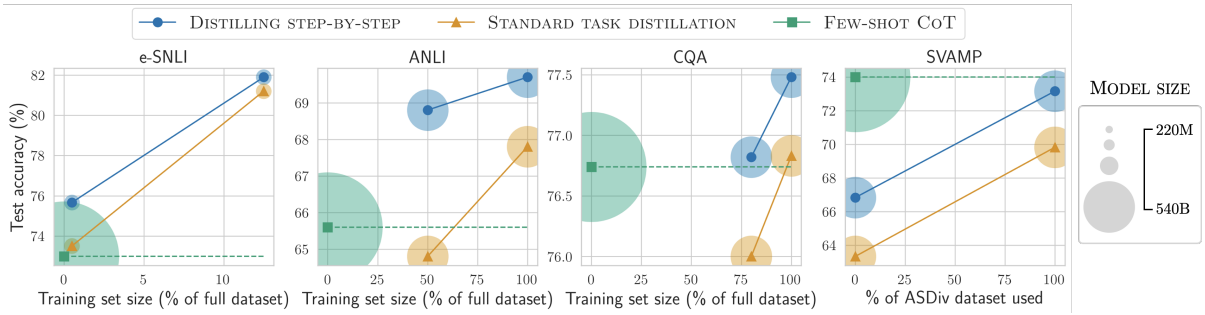

图9：类似于图8 ，但仅使用未标注示例，逐步蒸馏便能够以小得多的模型和少得多的示例，超越少样本思维链和标准任务蒸馏。在 SVAMP 上， x轴表示用于扩充原始 SVAMP 数据集的 ASDiv 数据集规模，即$x = 0$ 表示不进行扩充，而 $x = 100$表示添加完整的 ASDiv 数据集。

4.3 使用最小模型规模和最少训练数据超越大语言模型

在此，我们以大语言模型的表现为基准，探究逐步蒸馏与标准微调/蒸馏要超越该大语言模型，在训练样本数量和部署模型规模这两个方面所需的最高效资源配置。我们再次分别在人工标注设置和无标注设置下展示结果，见图 8 和图 9 。我们通过绘制不同的结果模型来呈现结果，并以以下维度进行对比：（1）使用的训练样本数量（x轴），（2）最终达到的任务表现（y轴），以及（3）模型规模（以阴影区域的大小表示）。

逐步蒸馏仅用更少的数据，就能凭借小得多的模型超越大语言模型。 在图 8所示的所有数据集上，我们看到，逐步蒸馏只使用部分可用训练样本，就能凭借小得多的 T5 模型超越 PaLM 的少样本思维链。具体来说，在 e-SNLI 上，逐步蒸馏能够取得更优的性能，使用的模型甚至小了 2000× （220M T5），且只需 0.1% 的完整数据集。在图 9中，在只有无标注数据集可用的情况下，我们观察到同样的趋势：逐步蒸馏在大多数时候都能用较小的模型和更少的数据超越少样本思维链。例如，在 ANLI 上，逐步蒸馏使用一个 45× 更小的模型和50% 的完整无标注数据集，就超越了大语言模型。

标准微调和蒸馏需要更多数据和更大的模型。 最后，在图 8 和图9中，我们看到，标准微调和蒸馏通常需要更多数据或更大的模型，才能达到大语言模型的性能。例如，在图8中，对于 e-SNLI，我们发现，逐步蒸馏仅使用数据集的 0.1%便能超越大语言模型，而标准微调则需要更多数据才能达到相同性能。此外，在图8中，对于 ANLI，我们发现，逐步蒸馏仅使用训练集的 770M 模型和 80% 即可超越 PaLM，而标准微调甚至难以达到大语言模型的性能

表1：逐步蒸馏适用于不同规模的大语言模型。从20B GPT-NeoX 模型中提取推理依据时，与在 2.2亿参数 T5 模型上进行标准微调相比，逐步蒸馏仍能提升性能。

方法	LLM	数据集 e-SNLI	ANLI	CQA	SVAMP
标准微调	N/A	88.38	43.58	62.19	62.63
逐步蒸馏	20B	89.12	48.15	63.25	63.00
逐步蒸馏	540B	89.51	49.58	63.29	65.50

使用完整数据集，因此需要更大的模型才能缩小性能差距。

4.4 进一步的消融研究

到目前为止，我们主要展示了逐步蒸馏在减少微调或蒸馏较小型特定任务模型所需训练数据方面的有效性。本节将进一步研究逐步蒸馏框架中不同组成部分的影响。具体而言，我们研究：(1) 提取推理依据所用的不同大语言模型如何影响逐步蒸馏的效果；(2) 在利用大语言模型推理依据训练小型特定任务模型时，多任务训练方法与其他潜在设计选择相比有何表现。这里，我们将小型特定任务模型固定为 2.2亿参数 T5 模型，并使用所有数据集 100% 的数据。

逐步蒸馏适用于经过良好训练、规模不同的大语言模型。除了使用540B PaLM 作为大语言模型外，我们还考虑一种规模相对较小的大语言模型，20B GPT-NeoX 模型（Black 等人，2022年），并从中提取推理依据用于逐步蒸馏。在表1中可以看到，结合不同规模的大语言模型时，与标准微调相比，逐步蒸馏仍能提升性能。不过，使用20B GPT-NeoX 模型而非540B PaLM 提取推理依据时，性能提升幅度较小。这可能是因为规模更大的 PaLM 模型能够提供质量更高的推理依据，从而更有利于学习任务。

多任务训练比单任务推理依据与标签联合预测有效得多。使用大语言模型推理依据作为输出监督来训练特定任务模型，有多种不同的方法。一种直接的方法是将推理依据 $\hat{r}_i$ 和标签 $\hat{y}_i$ 拼接成一个

表2：我们提出的多任务训练框架始终比将推理依据和标签预测视为单一任务带来更好的性能。单任务训练有时会导致性能不如标准微调。

方法	数据集 e-SNLI	ANLI	CQA	SVAMP
标准微调	88.38	43.58	62.19	62.63
单任务训练	88.88	43.50	61.37	63.00
多任务训练	**89.51**	**49.58**	**63.29**	**65.50**

序列 $[\hat{r}_i, \hat{y}_i]$ 并将整个序列作为训练小模型的目标输出，正如（Magister 等人，2022年；Ho 等人，2022年）中所述：

$$\mathcal{L}_{\text{single}} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), [\hat{r}_i, \hat{y}_i]). \quad (5)$$

在表格 2中，我们比较了这种单任务训练方法与我们提出的利用大语言模型推理依据的多任务训练方法。我们发现，多任务训练不仅能持续带来更好的表现，使用大语言模型推理依据进行单任务训练有时还会比标准微调表现更差，例如在 ANLI 和 CQA 上。事实上，(Wiegreffe 等人, 2021年; Magister 等人, 2022年;Ho 等人, 2022年) 中也观察到了类似结果：将推理依据和标签预测简单地视为单一联合任务，可能会损害模型的标签预测表现。这验证了我们采用多任务训练方法的合理性，也凸显了谨慎处理推理依据的必要性，从而充分发挥其实际优势。

5 讨论

我们提出逐步蒸馏，从大语言模型中提取推理依据，作为训练小型特定任务模型的信息性监督。我们证明，逐步蒸馏能够减少构建特定任务较小模型所需的训练数据集；同时，还能降低达到乃至超越原始大语言模型性能所需的模型规模。与现有方法相比，逐步蒸馏提出了一种资源高效的训练到部署范式。进一步的研究展示了逐步蒸馏的泛化能力及其设计选择。最后，我们将在下文讨论本工作的局限性、未来方向和伦理声明。

局限性

我们的方法存在若干局限性。首先，为了使用少样本思维链（~ 10所有任务均采用少样本设置）提示机制，我们需要用户提供少量示例演示（Wei 等人，2022年）。近期研究表明，无需用户标注的演示也能引出推理依据，这一进展有望改善此局限（Kojima 等人，2022年）。其次，使用推理依据训练特定任务模型会带来少量训练时计算开销。不过，在测试集上，我们的多任务设计能够自然地避免这类计算开销，因为它允许模型只预测标签，而不生成推理依据。最后，尽管我们观察到大语言模型推理依据取得了成功，但有证据表明，大语言模型在更复杂的推理和规划任务上的推理能力有限（Valmeekam 等人，2022年）。未来的工作应探究推理依据的质量如何影响逐步蒸馏。

伦理声明

值得注意的是，我们的下游较小模型的行为会受到从较大的教师大语言模型继承而来的偏差影响。我们设想，在减少大语言模型中的反社会行为方面取得的同样研究进展，也可以应用于改进较小的语言模型。

参考文献

Priyanka Agrawal、Chris Alberti、Fantine Huot、Joshua Maynez、Ji Ma、Sebastian Ruder、Kuzman Ganchev、Dipanjan Das 和 Mirella Lapata。2022年。Qameleon：仅用 5 个示例实现多语言问答。*arXiv* 预印本 *arXiv:2211.08264*。

Simran Arora、Avanika Narayan、Mayee F Chen、Laurel J Orr、Neel Guha、Kush Bhatia、Ines Chami、Frederic Sala 和 Christopher Ré。2022年。《有问必答：提示语言模型的简单策略》。*arXiv* 预印本 *arXiv:2210.02441*。

Lucas Beyer、Xiaohua Zhai、Amélie Royer、Larisa Mar-keeva、Rohan Anil 和 Alexander Kolesnikov。2022年。知识蒸馏：优秀的教师应当耐心且始终如一。发表于*IEEE/CVF*计算机视觉与模式识别会议论文集，第10925–10934页。

Sid Black、Stella Biderman、Eric Hallahan、Quentin Anthony、Leo Gao、Laurence Golding、Horace He、Con- nor Leahy、Kyle McDonell、Jason Phang、Michael Pieler、USVSN Sai Prashanth、Shivanshu Purohit、Laria Reynolds、Jonathan Tow、Ben Wang，以及 Samuel Weinbach。2022年。GPT-NeoX-20B：一种开源的自回归语言模型。载于*ACL*创建大语言模型的挑战与展望研讨会论文集。

Tom Brown、Benjamin Mann、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，33：1877–1901。Cristian Bucilu˘a、Rich Caruana和Alexandru Niculescu-Mizil。2006年。模型压缩。载于第*12*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，第535–541页。

Oana-Maria Camburu、Tim Rocktäschel、Thomas Lukasiewicz 和 Phil Blunsom。2018年。e-SNLI：基于自然语言解释的自然语言推理。神经信息处理系统进展，第31卷。

Ting Chen、Simon Kornblith、Kevin Swersky、Mohammad Norouzi 和 Geoffrey E Hinton。2020年。大型自监督模型是强大的半监督学习器。神经信息处理系统进展，33：22243–22255。

Aakanksha Chowdhery、Sharan Narang、Jacob Devlin、Maarten Bosma、Gaurav Mishra、Adam Roberts、Paul Barham、Hyung Won Chung、Charles Sutton、Sebastian Gehrmann 等。2022年。PaLM：利用 pathways 扩展语言建模。*arXiv*预印本 *arXiv:2204.02311*。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2018年。Bert：用于语言理解的深度双向 Transformer 预训练。*arXiv* 预印本 *arXiv:1810.04805*。

Jacob Eisenstein、Daniel Andor、Bernd Bohnet、Michael Collins 和 David Mimno。2022年。来自不可信教师的诚实学生：从预训练语言模型中学习可解释的问答流程。*arXiv* 预印本 *arXiv:2210.02498*。

Yao Fu、Hao Peng、Litu Ou、Ashish Sabharwal 和 Tushar Khot。2023年。让较小的语言模型专门用于多步推理。*arXiv* 预印本 *arXiv:2301.12726*。Braden Hancock、Antoine Bordes、Pierre-Emmanuel Mazare 和 Jason Weston。2019年。部署后从对话中学习：喂养自己，聊天机器人！*arXiv* 预印本 *arXiv:1901.05415*。Peter Hase 和 Mohit Bansal。2021年。模型何时能够从解释中学习？理解解释数据作用的形式化框架。*arXiv* 预印本 *arXiv:2102.02201*。Geoffrey Hinton、Oriol Vinyals、Jeff Dean 等。2015年。提炼神经网络中的知识。*arXiv* 预印本 *arXiv:1503.02531*，2(7)。

Namgyu Ho、Laura Schmid 和 Se-Young Yun。2022年。大型语言模型是推理教师。*arXiv* 预印本 *arXiv:2212.10071*。

Jordan Hoffmann、Sebastian Borgeaud、Arthur Mensch、Elena Buchatskaya、Trevor Cai、Eliza Ruther- f ord、Diego de Las Casas、Lisa Anne Hendricks、Johannes Welbl、Aidan Clark 等。2022年。训练计算最优的大语言模型。*arXiv* 预印本 *arXiv:2203.15556*。

Jeremy Howard 和 Sebastian Ruder。2018年。用于文本分类的通用语言模型微调。载于计算语言学协会第 *56* 届年会论文集（第 *1* 卷：长篇论文），第 328–339 页，澳大利亚墨尔本。计算语言学协会。

Jiaxin Huang、ShixiangShane Gu、Le Hou、Yuexin Wu、Xuezhi Wang、Hongkun Yu 和 Jiawei Han。2022年。大语言模型能够自我改进。*arXiv* 预印本 *arXiv:2210.11610*。

Fotis Iliopoulos、Vasilis Kontonis、Cenk Baykal、Gaurav Menghani、Khoa Trinh 和 Erik Vee。2022年。利用未标注示例进行加权蒸馏。载于神经信息处理系统进展。

Takeshi Kojima、Shixiang Shane Gu、Machel Reid、Yutaka Matsuo 和 Yusuke Iwasawa。2022年。大型语言模型是零样本推理器。*arXiv* 预印本 *arXiv:2205.11916*。

Brian Lester、Rami Al-Rfou 和 Noah Constant。2021年。规模对参数高效提示调优的力量。*arXiv*预印本 *arXiv:2104.08691*。

Liunian Harold Li、Jack Hessel、Youngjae Yu、Xi- ang Ren、Kai-Wei Chang 和 Yejin Choi。2023。符号化思维链蒸馏：小模型也能逐步“思考”。*arXiv* 预印本 *arXiv:2306.14050*。

Kevin J Liang、Weituo Hao、Dinghan Shen、Yufan Zhou、WeizhuChen、ChangyouChen 和 Lawrence Carin。2020年。Mixkd：迈向大规模语言模型的高效蒸馏。*arXiv*预印本 *arXiv:2011.00593*。

Lucie CharlotteMagister、Jonathan Mallinson、Jakub Adamek、Eric Malmi 和 Aliaksei Severyn。2022年。教会小型语言模型进行推理。*arXiv*预印本 *arXiv:2212.08410*。

Shen-yun Miao、Chao-Chun Liang 和 Keh-Yih Su。2020年。用于评估和开发英语数学文字题求解器的多样化语料库。载于计算语言学协会第*58*届年会论文集，第975–984页。

Smitha Milli、Ludwig Schmidt、Anca D Dragan 和 Moritz Hardt。2019年。根据模型解释进行模型重构。载于公平性、问责制与透明度会议论文集，第 1–9 页。

Sharan Narang、Colin Raffel、Katherine Lee、Adam Roberts、Noah Fiedel 和 Karishma Malkan。2020年。Wt5?!：训练文本到文本模型来解释其预测。*arXiv* 预印本 *arXiv:2004.14546*。Yixin Nie、Adina Williams、Emily Dinan、Mohit Bansal、Jason Weston 和 Douwe Kiela。2020年。对抗性自然语言推理：用于自然语言理解的新基准测试。载于计算语言学协会第*58*届年会论文集。计算语言学协会。Maxwell Nye、Anders Johan Andreassen、Guy Gur-Ari、Henryk Michalewski、Jacob Austin、David Bieber、David Dohan、Aitor Lewkowycz、Maarten Bosma、David Luan 等。2021年。展示你的工作：语言模型中间计算的草稿纸。*arXiv* 预印本 *arXiv:2112.00114*。

Arkil Patel、Satwik Bhattamishra 和 Navin Goyal。2021年。自然语言处理模型真的能解答简单的数学文字题吗？载于计算语言学协会北美分会 *2021*年会议：人类语言技术论文集，第2080–2094 页，线上。计算语言学协会。

Danish Pruthi、Rachit Bansal、Bhuwan Dhingra、Livio Baldini Soares、Michael Collins、Zachary C Lipton、Graham Neubig 和 William W Cohen。2022年。评估解释：教师提供的解释对学生有多大帮助？计算语言学协会汇刊，10：359–375。

Colin Raffel、Noam Shazeer、Adam Roberts、Kather- ine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。2020年。探索统一文本到文本转换器的迁移学习极限。机器学习研究杂志，21(140)：1–67。

Nazneen Fatema Rajani、Bryan McCann、Caiming Xiong 和 Richard Socher。2019年。解释一下你自己！利用语言模型进行常识推理。载于第 *57*届计算语言学协会年会论文集，第4932–4942页，意大利佛罗伦萨。计算语言学协会。

Andrew Slavin Ross、Michael C Hughes 和 Finale Doshi-Velez。2017年。理由恰当：通过约束解释来训练可微模型。*arXiv* 预印本 *arXiv:1703.03717*。

Ryan Smith、Jason A Fries、Braden Hancock 和 Stephen H Bach。2022a。循环中的语言模型：将提示融入弱监督。*arXiv* 预印本 *arXiv:2205.02318*。

Shaden Smith、Mostofa Patwary、Brandon Norick、Patrick LeGresley、Samyam Rajbhandari、Jared Casper、Zhun Liu、Shrimai Prabhumoye、George Zerveas、Vijay Korthikanti 等。2022b。使用 DeepSpeed 和 Megatron 训练 Megatron-TuringNLG

530b，一种大规模生成式语言模型。*arXiv* 预印本 *arXiv:2201.11990*。

Suraj Srinivas 和 François Fleuret。2018年。通过雅可比匹配进行知识迁移。载于国际机器学习大会，第 4723–4731 页。PMLR。

Alon Talmor、Jonathan Herzig、Nicholas Lourie 和 Jonathan Berant。2019年。CommonsenseQA：一项针对常识知识的问答挑战。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第 *1* 卷（长篇和短篇论文），第 4149–4158 页，明尼阿波利斯，明尼苏达州。计算语言学协会。

Raphael Tang、Yao Lu、Linqing Liu、Lili Mou、Olga Vechtomova 和 Jimmy Lin。2019年。将任务特定知识从 BERT 蒸馏到简单神经网络中。*arXiv* 预印本 *arXiv:1903.12136*。

Romal Thoppilan、Daniel De Freitas、Jamie Hall、Noam Shazeer、Apoorv Kulshreshtha、Heng-Tze Cheng、Alicia Jin、Taylor Bos、Leslie Baker、Yu Du 等。2022年。LaMDA：面向对话应用的语言模型。*arXiv* 预印本 *arXiv:2201.08239*。

Karthik Valmeekam、Alberto Olmo、Sarath Sree dharan 和 Subbarao Kambhampati。2022年。大语言模型仍然无法进行规划（针对大语言模型在规划和推理变化方面的基准测试）。*arXiv* 预印本 *arXiv:2206.10498*

Peifeng Wang、Aaron Chan、Filip Ilievski、Muhao Chen 和 Xiang Ren。2022a。PINTO：使用提示生成的推理依据进行忠实的语言推理。*arXiv* 预印本 *arXiv:2211.01562*。

ShuohangWang、YangLiu、YichongXu、Chenguang Zhu 和 Michael Zeng。2021年。想要降低标注成本？GPT-3 可以提供帮助。*arXiv* 预印本 *arXiv:2108.13487*。

Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc Le、Ed Chi 和 Denny Zhou。2022b。自洽性提升了语言模型中的思维链推理能力。*arXiv* 预印本 *arXiv:2203.11171*。

Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Ed Chi、Quoc Le 和 Denny Zhou。2022年。《思维链提示能够激发大语言模型的推理能力》。*arXiv* 预印本 *arXiv:2201.11903*。

Peter West、Chandra Bhagavatula、Jack Hessel、Jena D Hwang、Liwei Jiang、Ronan Le Bras、Ximing Lu、Sean Welleck 和 Yejin Choi。2021年。《符号知识蒸馏：从通用语言模型到常识模型》。*arXiv* 预印本 *arXiv:2110.07178*。

Sarah Wiegreffe、Ana Marasovi´c 和 Noah A. Smith。2021年。衡量标签与自由文本推理依据之间的关联。载于*2021*年自然语言处理实证方法会议论文集论文集，第10266–10284页，线上和多米尼加共和国蓬塔卡纳。计算语言学协会。

Omar Zaidan、Jason Eisner 和 Christine Piatko。2007年。使用“标注者理由”改进文本分类的机器学习。载于《*2007*年人类语言技术会议：北美计算语言学协会分会会议论文集；主会议论文集》，第260–267页，纽约州罗切斯特。计算语言学协会。

Eric Zelikman、Yuhuai Wu 和 Noah D Goodman。2022年。STAR：通过推理进行自举。*arXiv* 预印本 *arXiv:2203.14465*。

Jeffrey O Zhang、Alexander Sax、Amir Zamir、Leonidas Guibas 和 Jitendra Malik。2020年。侧向调优：通过添加式侧网络实现网络适配的基线。载于欧洲计算机视觉会议，第698–714页。施普林格。

Susan Zhang、Stephen Roller、Naman Goyal、Mikel Artetxe、Moya Chen、Shuohui Chen、Christopher De-wan、MonaDiab、Xian Li、Xi Victoria Lin 等人。2022年。OPT：开放式预训练变换器语言模型。*arXiv*预印本 *arXiv:2205.01068*。

Ye Zhang、Iain Marshall 和 Byron C. Wallace。2016年。用于文本分类的推理依据增强型卷积神经网络。载于*2016*年自然语言处理实证方法会议论文集，第795–804页，得克萨斯州奥斯汀。计算语言学协会。

Lianmin Zheng、Zhuohan Li、Hao Zhang、Yonghao Zhuang、Zhifeng Chen、Yanping Huang、Yida Wang、Yuanzhong Xu、Danyang Zhuo、Joseph E Gonza- lez 等。2022年。Alpa：为分布式深度学习自动化算子间和算子内并行。*arXiv* 预印本 *arXiv:2201.12023*。

A 实验详情

A.1 实现

我们在 cloudA 上进行实验100×16GPU 实例。我们使用以下超参数训练 T5 模型，所用软件包来自公开可用的 https://github.com/huggingface/transformers：

• T5-Base（220M）和 T5-Large（770M）：我们使用学习率 $= 5 \times 10^{-5}$、批量大小 $= 64$、最大输入长度 $=1024$ 训练模型，最多训练 10000 步。

•T5-XXL（11B）：我们使用学习率 $= 5 \times 10^{-5}$、批量大小 $= 32$、最大输入长度 $= 1024$ 训练模型，最多训练 4000 步。

我们报告了 4 次随机运行的所有结果，并在所展示的图表中包含标准误差。

A.2 数据集

我们对实验中使用的数据集进行了更详细的说明。我们列出了获取这些数据集的来源，以及作者最初发布数据集的来源。关于数据集的许可或使用和/或分发条款，请读者参阅这些来源。据我们所知，所用数据集不包含可指名或唯一识别个人的信息，也不包含冒犯性内容。

•e-SNLI：该数据集最初发表于（Camburu 等人，2018年），并公开发布于 https://github.com/OanaMariaCamburu/e-SNLI。我们从以下地址获取该数据集：https://huggingface.co/datasets/esnli。

• ANLI：该数据集最初发表于（Nie 等人，2020年），并公开发布于 https://github.com/facebookresearch/anli。我们从以下地址获取该数据集：https://huggingface.co/datasets/anli。实验中使用了 R1 划分。

• CQA：该数据集最初由（Talmor 等人，2019年）发布，并在以下网址公开提供：https://www.tau-nlp.sites.tau.ac.il/commonsenseqa。随后，研究人员为其补充了人工标注的解释（Rajani 等人，2019年）提供，网址为 https://github.com/salesforce/cos-e。我们从 https://huggingface.co/datasets/cos_e 获取了实验中使用的数据集。

表3：我们实验中使用的数据集统计信息。

数据集	训练集	验证集	Test
e-SNLI	549,367	9,842	9,824
ANLI	16,946	1,000	1,000
CQA	8,766	975	1,221
SVAMP	720	80	200

•SVAMP：该数据集最初发布于（Patel 等人，2021年）。我们从 https://github.com/arkilpatel/SVAMP获取该数据集。

•ASDiv：该数据集最初于（Miao 等人，2020年）发布。我们从 https://github.com/chaochun/nlu-asdiv-dataset获取该数据集。

对于每个数据集，如果原本未提供验证集，我们会从原始训练集中随机抽取10%作为验证集。对于 CQA，由于原始测试集没有真实标签，我们将原始验证集用作测试集。数据集统计信息见表3。