

基于中间层对齐的异构师生模型知识蒸馏

翟飞燕, 王任之, 李丕绩*

南京航空航天大学计算机科学与技术学院
模式分析与机器智能工业和信息化部重点实验室 (南京航空航天大学)
江苏, 南京, 211106
{fyzhai, rzhwang, pjli}@nuaa.edu.cn

摘要

知识蒸馏技术作为大语言模型时代的一项前沿模型压缩策略, 通过将复杂模型的知识有效迁移至简单模型, 显著降低了模型的参数规模和计算成本。尽管如此, 目前主流的生成式大语言模型蒸馏算法主要集中于优化师生模型间的最后输出层损失, 而忽视了对模型中间层的探索。此外, 针对中间层蒸馏的研究往往对师生模型的结构一致性有着严格的要求, 无法处理异构模型间的蒸馏问题, 从而存在明显的局限性。针对这些问题, 我们提出了一种新的知识蒸馏算法: 引入了中间层蒸馏损失的异构生成式师生大语言模型知识蒸馏算法。该算法首先提取师生模型的中间层信息作为蒸馏对象, 随后通过专门设计的中间层映射规则和对齐模块, 实现异构模型间基于中间层知识对齐与损失计算。最后, 联合优化各个蒸馏损失的比例。通过在五个相关数据集上进行实验验证, 我们的方法在提高蒸馏效果方面展现出显著优势。

关键词: 生成式大语言模型; 白盒知识蒸馏; 中间层; 异构性

Knowledge distillation of heterogeneous teacher-student model with intermediate layer loss

Feiyan Zhai, Renzhi Wang, Piji Li*

College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
Nanjing, 211106, China
{fyzhai, rzhwang, pjli}@nuaa.edu.cn

Abstract

As a cutting-edge model compression strategy in the era of large language models, knowledge distillation technology significantly reduces the parameter scale and computational cost of models by effectively transferring the knowledge of complex models to simple models. However, the current mainstream generative large language model distillation algorithms mainly focus on optimizing the loss of the final output layer between the teacher-student models, and neglect the exploration of the model middle layer. In addition, the research on interlayer distillation often has strict requirements on the consistency of the structure of the teacher-student model, and can not deal with the distillation problem between heterogeneous models, so there are obvious limitations. To solve these problems, we propose a new knowledge distillation algorithm, which introduces the heterogeneous generative teacher-student large language model

©2024 中国计算语言学大会

根据《Creative Commons Attribution 4.0 International License》许可出版

*Corresponding author.

knowledge distillation algorithm with intermediate layer distillation loss. The algorithm first extracts the middle layer information of the teacher-student model as the distillation object, and then realizes the knowledge alignment and loss calculation between heterogeneous models based on the middle layer through the specially designed middle layer mapping rules and alignment module. Finally, the ratio of each distillation loss is optimized jointly. Through experimental validation on five relevant data sets, our method shows significant advantages in improving the distillation efficiency.

Keywords: White Box Large Language Model , Knowledge Distillation , Intermediate Layer , Isomerism

1 引言

近年来, 大语言模型 (Large Language Model, 简称LLM) 蓬勃发展(Brown et al., 2020; Achiam et al., 2023; Chowdhery et al., 2023; Anil et al., 2023; Thoppilan et al., 2022), 在对话生成、翻译、问答、文本摘要等多项自然语言处理任务中均展现出了强大的性能(Zhao et al., 2023; Chang et al., 2023)。然而由于参数规模庞大, 给资源受限环境下的训练和推理带来了巨大的困难(Strubell et al., 2019)。

相比之下, 具有较少参数量的语言模型为资源受限的研究环境提供了一种更为可行的替代方案。但由于规模较小, 其所涵盖的知识和语言能力相对有限。而知识蒸馏(Hinton et al., 2015; Adriana et al., 2015)便是一种将知识从参数规模大而复杂的深度模型转移到轻量而高效的小模型中的模型压缩技术(Deng et al., 2020; Han et al., 2015), 它通过将大模型学习到的丰富知识蒸馏到较小的模型中, 使得小模型能够从大模型的优越性能中受益, 平衡了小参数与高性能(Gou et al., 2021), 是大语言模型时代的一项极具发展前景的技术。

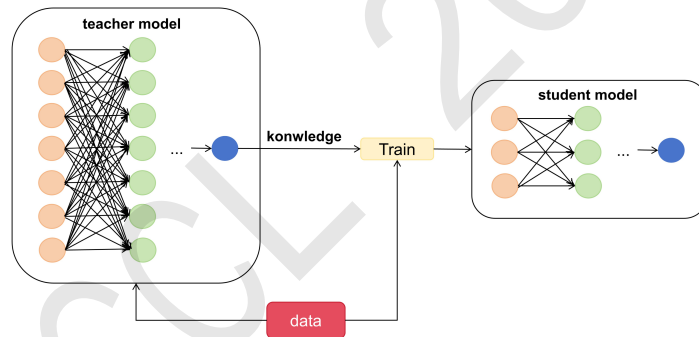


Figure 1: 知识蒸馏算法简图。知识蒸馏是将知识从一个庞大复杂的教师模型转移到一个轻量高效的学生模型中。

如图1所示, 在大语言模型时代来临之前, 传统的知识蒸馏研究(Adriana et al., 2015; Hinton et al., 2015)重点放在特定神经网络结构和为任务量身定做的训练目标设计上(Xu et al., 2024)。而在大语言模型时代, 根据教师模型参数的可获取与否, 知识蒸馏主要分为: 对黑盒LLM的蒸馏(Jiang et al., 2023; Wu et al., 2023)和对白盒LLM的蒸馏(Timiryasov and Tastet, 2023; Liang et al., 2023; Gu et al., 2023)。相比于黑盒模型在蒸馏过程中只有教师预测结果可用, 白盒大语言模型易于获取教师中间层结果和所有参数信息(Gou et al., 2021)。此外, 白盒大语言模型根据结构的不同分为由编码器组成的语言理解模型(Devlin et al., 2018)和由解码器组成的生成式语言模型(Radford et al., 2019; Touvron et al., 2023a; Touvron et al., 2023b)。由于现有的白盒蒸馏方法主要应用于编码器结构的语言理解模型, 而针对解码器结构的生成式大语言模型尚未有深度的探究, 因此本文的主要研究对象是生成式大语言模型的白盒蒸馏。

现有的生成式大语言模型的白盒蒸馏方法大多仅考虑教师模型与学生模型的最后输出层蒸馏损失(Gu et al., 2023; Agarwal et al., 2023; Sanh et al., 2019; Wen et al., 2023; Kim and Rush, 2016), 忽略了对教师模型中间层信息的蒸馏。而在大语言模型的生成过程中, 中间层占据了它

结构的一大部分,包含着许多信息:丰富的语义信息、对输入句子的注意力分配等(Clark et al., 2019; Sun et al., 2019; Wang et al., 2020b; Wang et al., 2020a)。因此,对于大语言模型中间层信息的蒸馏具有很大的研究意义。

据调研,针对Transformer架构模型的中间层蒸馏方法(Jiao et al., 2019; Sun et al., 2019; Wang et al., 2020b)往往对教师模型与学生模型的内部结构要求严格一致性:教师模型的Transformer块数需为学生的整倍数、学生模型的注意力头数及中间层隐状态维度需与教师模型完全相同,而现实中许多蒸馏场景中教师模型与学生模型都是异构的。因此,现有的针对中间层的蒸馏算法具有明显局限性,需要一个结构无关的更普适的中间层蒸馏算法。此外,虽然针对中间层的蒸馏方法在编码器结构的语言理解模型中很常见(Sun et al., 2019; Jiao et al., 2019; Wang et al., 2020a; Hou et al., 2020; Liang et al., 2020),但迄今为止还没有结构无关的Transformer架构的生成式大语言模型间的中间层蒸馏算法研究(Xu et al., 2024)。

为解决上述问题,我们提出了一种针对生成式大语言模型的中间层蒸馏算法,该算法放松了对教师LLM与学生LLM中间层内部结构的一致性要求,更加普适于异构生成式大语言模型之间的蒸馏场景。此外,我们将设计的中间层蒸馏算法应用于仅在输出层对教师LLM进行蒸馏的三种蒸馏算法上:标准知识蒸馏(Sanh et al., 2019; Wen et al., 2023)、序列化知识蒸馏(Kim and Rush, 2016)以及Minillm(Gu et al., 2023),并分类地对不同的中间层输出结果进行了多项实验,最后优化蒸馏损失权重配比以最大化模型蒸馏效果。具体地,我们选用开源的GPT2系列模型(Brown et al., 2020)作为蒸馏任务中的教师与学生,其中,学生模型的参数量大小从120M (gpt2-base)到760M (gpt2-large)不等,教师模型选用参数量为1.5B的gpt2-xlarge。实验涵盖了大范围的NLP任务,使用5个指令跟随数据集: DollyEval, SelfInst(Wang et al., 2022a), VicunaEval(Chiang et al., 2023), S-NI(Wang et al., 2022b)以及UnNI(Honovich et al., 2022),结合GPT4-score(Achiam et al., 2023)和Rouge-L(Lin, 2004)两种机器指标以及人工评测对蒸馏后的学生模型进行综合评估。多项实验表明,引入中间层蒸馏损失后得到的学生模型在几乎所有数据集上的表现都优于仅针对最后输出层蒸馏得到的学生模型,验证了我们提出的更普适于异构生成式LLM的中间层蒸馏算法的有效性。

2 相关工作

知识蒸馏(Knowledge Distillation)旨在从复杂的教师模型中提取知识训练简化的学生模型,从而使学生模型在保持相对较高性能的同时拥有更小的模型体积和更低的计算成本。

早期的知识蒸馏致力于将知识从庞大复杂的模型转移到轻量高效的网络中。Geoffrey Hinton利用教师模型的输出概率分布(Soft Target)(Hinton et al., 2015)作为学生模型的训练目标,相比于最大化正确答案的平均对数概率(Hard Target)能得到更多信息以更好地训练小网络,让小模型获得大模型的泛化能力。后来的序列级知识蒸馏(简称SeqKD)(Kim and Rush, 2016)通过最大化教师模型生成序列的可能性来微调学生模型,使学生的预测与教师的预测保持一致。这些传统的知识蒸馏思想为后续大语言模型蒸馏奠定了基础。

近年来,随着大语言模型的蓬勃发展,知识蒸馏越来越多地应用于参数量巨大的大语言模型压缩上。主要分为两类:一是针对黑盒大语言模型(Achiam et al., 2023)的蒸馏,二是针对白盒大语言模型(Radford et al., 2019; Le Scao et al., 2022; Touvron et al., 2023a; Touvron et al., 2023b; Zhang et al., 2022; Team and others, 2023)的蒸馏。其中,黑盒知识蒸馏研究在大语言模型API生成的即时响应对的小模型微调中显示出不错的结果(Jiang et al., 2023; Wu et al., 2023)。但由于黑盒蒸馏需要调用API,这些模型成本高昂,并且在某种程度上不可复制。而相比之下,白盒蒸馏为研究人员提供了一种更透明、更容易接近的方法。

现有的白盒蒸馏研究多是仅针对教师模型和学生模型的最后输出层计算蒸馏损失。标准知识蒸馏(简称KD)(Sanh et al., 2019)用标记级概率分布标记固定的序列数据集来提取教师特征知识。具体分为两步:监督学习、教师模型学习。第一步监督学习中学生模型根据标注数据进行监督学习,此过程中的损失函数为语言模型交叉熵损失。第二步教师模型学习中学生模型以教师模型的Soft Target为目标计算蒸馏损失进行训练。最后把这两个损失结果相加即为KD损失。上述标准知识蒸馏和序列化知识蒸馏方法本质上都是最小化教师和学生分布之间的近似前向KL散度: $KL(p \parallel q_\theta)$ (Sanh et al., 2019; Wen et al., 2023; Timiryasov and Tastet, 2023; Liang et al., 2023),24年提出的Minillm方法(Gu et al., 2023)揭示了这一本质,并深入研究了白盒生成式LLM的蒸馏,提出对于开放文本生成任务,让学生模型去学习教师模型的概率分布会导

致学生模型学到过多教师模型错误的判断结果。因此，为解决这个问题，Minillm提出让教师模型的输出分布去拟合学生模型的概率分布，即最小化教师和学生分布之间的近似反向KL散度： $KL(q_\theta \parallel p)$ ，并在整体模型的训练中加入了PPO强化学习算法(Schulman et al., 2017)，使用策略梯度优化(Sutton et al., 1999)策略选择，进一步优化了模型的蒸馏效果。

除了仅针对最后输出层的白盒蒸馏，近年来许多针对Transformer编码器架构的模型中间层蒸馏算法（简称ILD）提出，这些研究根据蒸馏的中间层对象的不同分为两类：对隐藏层的蒸馏、对自注意力模块的蒸馏。第一类中，Patient Teacher(Sun et al., 2019)不仅让学生网络从教师的Soft Target学习，还提出patient teacher-student机制，让学生模型耐心地从教师模型的多个中间层学习增量知识提取。第二类中，MiniLM(Wang et al., 2020b)提出深度自注意力精馏，通过提取教师模型最后一个Transformer层的自注意力模块让小模型通过深度模仿大模型的自注意力模块进行训练，并实验验证了对教师模型中间隐藏层蒸馏的有效性。紧随其后的MiniLMv2(Wang et al., 2020a)在此基础上进行了推广，它不再要求师生模型的自注意力头数一致，且深入研究了教师模型的中间层层次选择策略。此外，Tinybert(Jiao et al., 2019)针对BERT系列模型设计了ILD，兼具了对教师模型两种中间层对象的蒸馏，减少了模型层数但同时几乎保持了原教师模型的性能，但它对师生模型的中间层结构要求严格一致性，因此有明显的局限性。ALP-KD(Passban et al., 2021)提出了一种依赖于中间层注意力的组合蒸馏技术，它融合了教师端信息并考虑了教师模型每一层的重要性，解决了以往ILD会跳过一些教师层而仅从其子集中提取知识的缺陷。针对ILD无法轻松处理层映射搜索和师生模型架构不匹配等问题，Universal-KD(Wu et al., 2021)通过基于注意力的层投影来匹配输出空间中教师和学生的中间层（在中间层上添加伪分类器）。RAIL-KD(Haidar et al., 2021)提出了一种新的师生模型之间中间层映射方式：在每一轮训练时，随机选取教师模型的中间层与学生模型的中间层建立映射以进行知识蒸馏，其在BERT等系列教师模型的蒸馏任务中验证了通用性与有效性。最新的研究中，CR-ILD(Ko et al., 2023)发现了ILD在训练数据集上的过度拟合问题，通过仅提取教师模型最后一个Transformer层并在补充任务上进行ILD有效缓解了此种现象。针对当前的知识蒸馏方法依赖于模型的中间层特征和标签信息，因而通常需要对齐的模型架构和足够的标记数据这一问题，2023年提出的GLMD(Tan et al., 2023)方法执行两阶段单词预测蒸馏和词汇压缩，通过消除师生模型之间的维度和结构的限制以及通过没有中间层和硬标签来消除对标记数据集的需求来支持更通用的应用场景。综上，现有的白盒蒸馏方法多用于Transformer编码器架构的语言理解模型蒸馏，而对于生成式大语言模型的异构师生模型的中间层蒸馏算法尚有待探索，而这正是本文的研究主题。

3 模型介绍

本文算法主要包含四个模块：中间层对象选择模块、中间层结果对齐模块、中间层损失计算模块以及各蒸馏损失融合优化模块。

整体流程框架如图2所示，其中，中间层对象选择模块阐述了我们选择的不同中间层结果种类及其含义；中间层结果对齐模块是我们算法的核心，它实现了异构师生模型中间层结果的对齐。主要包含两个部分：一是中间层映射规则设计，这部分详细介绍了我们为不同层数关系的教师与学生模型设计的层对应规则；二是维度转换模块，这部分说明了当面对中间层输出结果维度不一致时，我们是如何进行维度转换的；中间层损失计算模块讲述了我们对于最终损失函数的设计；最后的各蒸馏损失融合优化模块介绍了我们是如何调节各部分蒸馏的损失权重配比，以达到优化蒸馏效果这一目标的。

3.1 中间层蒸馏对象选择

考虑到基于Transformer架构的开源大语言模型的中间层输出结果主要包含了两类：注意力模块输出、隐藏层输出。这里隐藏层输出包括输入模型主体部分前的词嵌入层、模型每个Transformer层的中间隐藏层输出以及模型最后一层输出。实验中，我们仅使用了每个Transformer Layer的中间隐藏层输出作为蒸馏对象。考虑到这些中间层输出结果代表的含义不同，蕴含的信息也不同，在中间层信息上的重要程度占比未知，蒸馏效果也有待考量(Clark et al., 2019)。因此，我们针对以上两种不同类型的中间层结果，分别进行了蒸馏实验。使用两种不同的中间层输出作为蒸馏对象后得到的实验结果对比见表3,表4,表5。

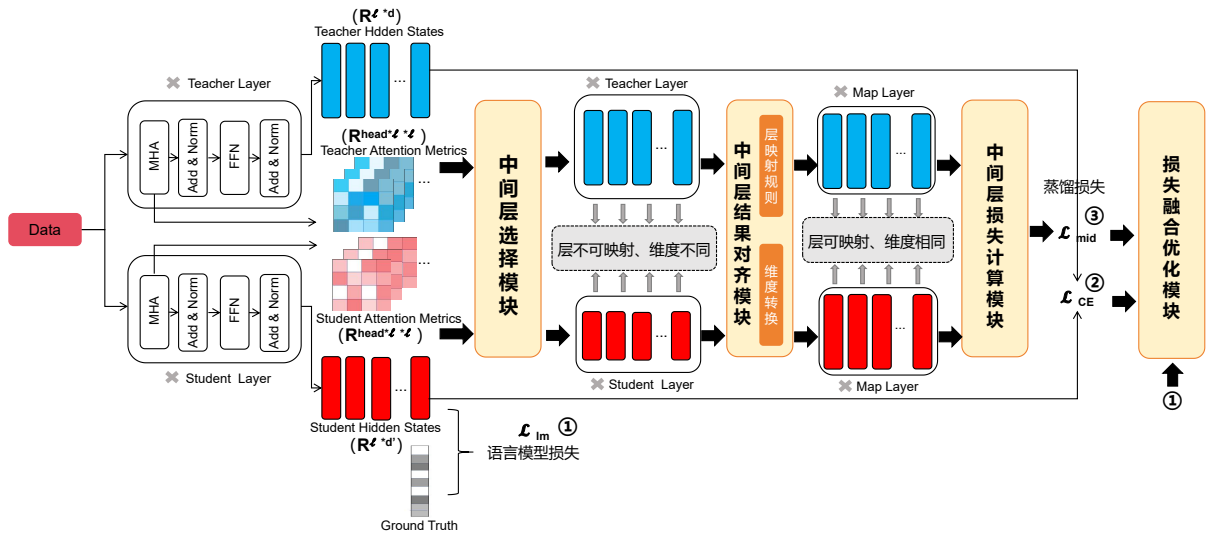


Figure 2: 引入中间层损失的异构师生模型蒸馏算法框架图。包括中间层对象选择、中间层结果对齐、中间层损失计算、各蒸馏损失融合优化四个模块。

3.2 中间层结果对齐模块

与Tinybert(Jiao et al., 2019)不同的是，本文研究的教师模型与学生模型的中间层输出在结构上不同，主要体现在组成教师模型和学生模型的Transformer块的隐藏层状态维度不同、多头注意力模块中的注意力头数不同、中间层Transformer块的个数不成倍数关系等结构细节上，如图3。以上原因直接导致了在中间层蒸馏的实现过程中不能直接利用教师模型和学生模型的中间层输出结果进行蒸馏损失计算，而要首先进行中间层结果的对齐。为此，我们设计了更普适的异构师生模型中间层映射规则和维度转换模块。

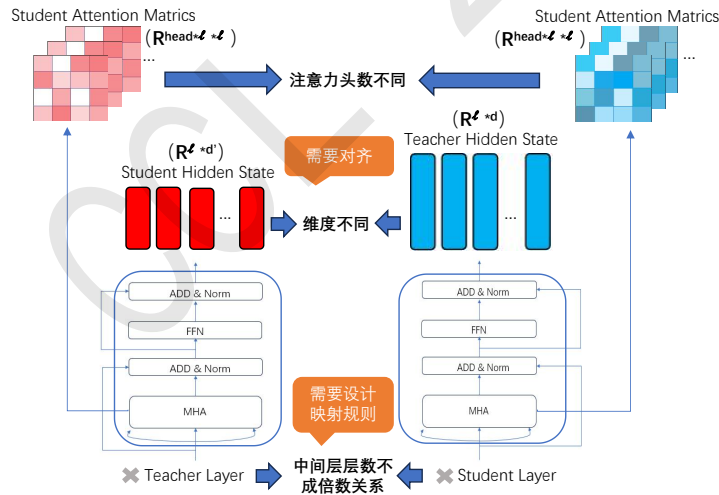


Figure 3: 师生模型中间层异构性示意图。展示了本文研究的师生模型在中间结构上的异构细节

3.2.1 中间层映射规则

具体的，层映射规则设计为：取教师模型与学生模型的中间层层数的最大公因数为后续中间层对齐时的Transformer组合块数，若最大公因数不为1，则最大公因数的值即为教师模型和学生模型划分出的中间层组合块数，为后续的中间层蒸馏一一对应。当遇到最大公因数为1的情况时，就依次将教师模型、学生模型的中间层数循环减1，直到最大公因数不为1时停止循环，剩余的中间层按照上述规则进行组合对齐，减掉的部分中间层也同样相应对齐。举个例子，当教

师模型中间层为48层，学生模型为20层时，属于情况一：取师生模型层数的最大公因数4，将教师模型和学生模型的中间层都分别划分为四个大块，这样教师模型每块有12层，学生模型每块有5层，使用以块为单位的中间输出结果作为蒸馏对象进行后续的中间层蒸馏损失计算。而当教师模型为49层，学生模型为19层时，属于情况二，则循环依次将教师模型、学生模型层数减一，直到师生模型的层数最大公因数不为1停止循环，操作后剩余的层之间和减去的对应层之间均参考情况一的处理方法，这样设计是为了后续能层次对应均匀地进行中间层蒸馏。这部分的可解释性实验见5.2.2，具体的处理如下伪代码：

Algorithm 1 师生模型中间层映射规则

Input: 教师模型中间层注意力矩阵 t_atts , 学生模型中间层注意力矩阵 s_atts

Output: 教师模型最终映射矩阵 map_t , 学生模型最终映射矩阵 map_s

```

1:  $max\_gcd \leftarrow$  师生模型中间层层数的最大公因数
2: while  $max\_gcd == 1$  do
3:   if 第偶数次 (初始为0) then
4:     教师中间层层数减一
5:   else
6:     学生中间层层数减一
7:   end if
8:    $max\_gcd \leftarrow$  当下师生模型中间层层数的最大公因数
9: end while
10: 获取教师模型中间层参与映射的总块数 $t\_block \leftarrow \lfloor t\_layer / max\_gcd \rfloor$ 
11: 获取学生模型中间层参与映射的总块数 $s\_block \leftarrow \lfloor s\_layer / max\_gcd \rfloor$ 
12: 计算教师模型最终的中间层映射矩阵 $map\_t \leftarrow [t\_atts[i \times t\_block + t\_block - 1]$ 
13: 计算学生模型最终的中间层映射矩阵 $map\_s \leftarrow [s\_atts[i \times s\_block + s\_block - 1]$ 
14: if 属于情况2 then
15:   被减去的师生模型中间层部分按照同样的规则得到对应映射，并加入最终映射矩阵中
16: else
17:   保持原样
18: end if
19: 最终输出 $map\_t, map\_s$ 

```

3.2.2 维度转换模块

中间层映射规则确定后，就需要统一教师模型与学生模型的中间层输出维度了，让他们可以进行蒸馏损失值的计算。具体的，我们在模型层面设计了一个线性映射层，负责将教师模型的中间层结果通过这一线性映射网络转换为与学生模型的中间层输出结果维度一致的张量，然后进行中间层损失计算。这个线性映射网络与学生模型一起参与训练优化，尽可能地让维度转换后的教师模型中间层结果保留其语义信息，以传递给学生模型。最终的中间层蒸馏目标如下：

$$\mathcal{L}_{mid} = \sum_{i=1}^N \mathcal{L}_F(\Phi(f_i^T(x)), f_i^S(x)) \quad (1)$$

其中， N 指进行蒸馏的中间层对应映射的层数， i 的取值取决于3.2.1中讲述的中间层映射规则， $f_i^T(x)$ 和 $f_i^S(x)$ 分别是教师模型和学生模型第 i 个Transformer块的中间层输出结果， mid 与 f 的种类取决于3.1中的中间层蒸馏对象选择，转换函数 Φ_T 应用于教师模型的中间层，进行维度转换，以确保教师模型与学生模型的中间层输出具有相同的形状，便于直接比较，而损失函数类型 \mathcal{L}_F 将在3.3中选择与确定。

3.3 中间层损失计算模块

在中间层损失函数的选择过程中，为了与标准知识蒸馏方法中的仅输出层的蒸馏损失 \mathcal{L}_{CE} 与语言模型损失 \mathcal{L}_{lm} 在数值量级上保持对齐。我们尝试了多种不同的中间层损失函数来对教师模型和学生模型进行中间层输出结果的损失计算。具体地，我们尝试了余弦相似性损

失函数（最大化师生模型中间层输出张量的相似度）、平均绝对误差MAE（最小化师生模型中间结果的差）、均方误差MSE（最小化师生模型中间层输出结果的平方差），结果对比见表7，最终根据实际实验效果选择了MSE作为损失函数，因此最终的中间层损失函数为：

$$\mathcal{L}_{\text{mid}} = \frac{1}{N} \sum_{i=1}^N (\Phi(f_i^T(x)) - f_i^S(x))^2 \quad (2)$$

3.4 各蒸馏损失融合优化模块

在获取了中间层蒸馏损失后，我们设计了各部分损失的融合优化模块，包括：语言模型损失 \mathcal{L}_{lm} 、最后输出层的蒸馏损失 \mathcal{L}_{CE} 、我们引入的中间层蒸馏损失 \mathcal{L}_{mid} ，各部分的损失计算公式如下：

语言模型损失：

$$\mathcal{L}_{\text{lm}} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{i,j} \log(p_{i,j}) \quad (3)$$

其中， N 是指样本数量， C 是类别数量（词汇表大小）， $y_{i,j}$ 是样本 i 的真实标签， $p_{i,j}$ 是模型对于样本 i 中第 j 类别的预测概率。

最后输出层的蒸馏损失：

$$\mathcal{L}_{\text{CE}} = \sum_i t_i \log(s_i) \quad (4)$$

其中， t_i 是指教师模型的预测分布， s_i 是指学生模型的预测分布，两者计算交叉熵损失。

引入的中间层蒸馏损失：

$$\mathcal{L}_{\text{mid}} = \begin{cases} \frac{1}{N} \sum_{i=1}^N \text{MSE}(\Phi_{\text{attn}}(A_i^T), A_i^S), & \text{mid} = \text{attention} \\ \frac{1}{N} \sum_{i=1}^N \text{MSE}(\Phi_{\text{layer}}(o_i^T(x)), o_i^S(x)), & \text{mid} = \text{hidden layer} \end{cases} \quad (5)$$

其中， MSE 是指MSE损失函数， Φ_{attn} (Φ_{layer})是指针对注意力模块（隐藏层）设计的线性映射， $o_i(x)$ 是指中间隐藏层输出， A_i 是指中间注意力模块输出。

我们使用网格搜索方法根据模型蒸馏效果自动优化配比三种不同损失在总损失上所占比重，最终得到融合了三种不同损失的优化目标，公式如下：

$$\mathcal{L}_{\text{sum}} = \alpha \mathcal{L}_{\text{lm}} + \beta \mathcal{L}_{\text{CE}} + \sum_{i=0}^N \lambda_i \mathcal{L}_{\text{layer}} \quad (6)$$

其中， N 指进行蒸馏的中间层对应映射的层数， $\mathcal{L}_{\text{layer}}$ 是第 i 层的中间层蒸馏损失， λ_i 是第 i 层的蒸馏损失占整个中间层蒸馏中的重要性参数， α 和 β 都是损失重要性系数。

4 实验设置

4.1 实验训练设置

实验选用的蒸馏数据集来自 *databricks - dolly - 15k*¹，由15K对人类书写的指令-响应组成。我们使用其中随机分割的14K个样本作为训练集 D ，剩下的500个作为验证集和测试集。对于模型预训练数据集 \mathcal{D}_{PT} ，我们使用OpenWebText(Gokaslan et al., 2019)。实验过程中，我们首先在 D 上微调一个生成式大语言模型作为教师 p ，然后通过评估蒸馏模型在五个指令跟随数据集上的表现比较不同知识蒸馏方法的蒸馏效果。实验流程与训练细节如下：

- **实验流程：**我们首先按照4.1节中所述选取并划分数据集，接着参考之前的研究(Gu et al., 2023)在预处理好的指令-回复数据集 D 上微调大语言模型GPT2-xlarge作为后续蒸馏的教师模型；然后分别对三种学生模型GPT2-base、GPT2-medium、GPT2-large在数据集 D 上使用不同的蒸馏方案蒸馏：仅在 D 上使用标签数据监督微调（SFT w/o KD）、标准蒸馏（KD）、序列化蒸馏（SeqKD）、MiniLLM方法、引入中间层蒸馏损失的蒸馏方法；比较这些不同的知识蒸馏方法蒸馏训练后得到的学生模型在预先划分出来的测试集上生成的回复内容的Rouge-L与GPT4得分，并最终评估蒸馏效果的好坏。

¹<https://github.com/databrickslabs/dolly/tree/master>

- **训练与参数设置**: 对于基线方法, 学习率取值区间设为 $[5e-4, 1e-4, 5e-5]$, 总批大小取值区间设为 $[32, 64]$, 蒸馏训练迭代20轮。对于标准知识蒸馏(KD), 以0.5的混合率将蒸馏损失与语言模型损失混合得到最终损失。每个基线方法的检查点由验证集的Rouge-L得分选择, 选择分数最高的为最优的蒸馏模型。对于MiniLLM方法, 首先使用数据集D对学生模型进行监督微调后选择验证损失最小的检查点作为后续MiniLLM训练的初始模型, 使用Adamw优化器, 学习率设为 $5e-6$, 总批大小设为64, 采样时温度设为1, 模型允许的最大文本长度为512, 同样的, 使用验证集上的Rouge-L分数选择最后的检查点。对于引入中间层蒸馏损失的方法, 本文提出的方法分别加在了上述基线方法中进行中间层蒸馏实验, 因此超参数设置与上述设置一致, 除了最后得到的中间层蒸馏损失在最终的损失函数占比上的权重配比使用网格搜索算法进行了优化, 各损失的权重搜索范围设为 $[0, 1]$, 选择0.1的间隔进行搜索, 使用交叉验证的平均损失值作为评价指标, 遍历所有可能的权重组合以寻找最佳的权重配比方案。以上实验基于8卡NVIDIA RTX A5000, 并使用deepspeed进行模型分布式训练。

4.2 对比方法

为了评估在生成式大语言模型中引入中间层蒸馏的有效性, 我们与以下基线方法作对比:

- **SFT w/o KD**: 直接在数据集D上进行监督微调, 不使用知识蒸馏。
- **KD(Sanh et al., 2019)**: 词级别的知识蒸馏。使用教师分布作为每个token级别的监督, 在数据集D上对学生模型进行微调, 最小化师生模型的正向KL散度。
- **SeqKD(Kim and Rush, 2016)**: 使用教师模型生成的数据对学生模型进行微调。
- **MiniLLM(Gu et al., 2023)**: 类似于KD, 不同点在于最小化师生模型的反向KL散度, 且加入了模型优化策略。

4.3 评测指标

我们使用以下两个指标评估模型生成的响应:

- **Rouge-L(Lin, 2004)**: 用来评估模型生成内容的精度, (Wang et al., 2022b)表明Rouge-L适用于大规模指令跟踪评估。具体地, Rouge-L指标结果是对各个蒸馏方法在每个数据集的五个不同随机种子下生成的回答内容计算Rouge-L值最后取平均得到的。
- **GPT4(Achiam et al., 2023)**: 设计prompt将蒸馏模型生成的结果与ground truth标签数据进行比较打分, 分值为1-10分, 关于使用GPT4评分时的prompt设计详见附录A。具体地, 对于每个蒸馏模型, 我们使用五个不同的随机种子生成回复, 最后的GPT4打分结果是将各个蒸馏方法在各个数据集的不同随机种子设置下生成的结果均匀采样100条数据后调用GPT4接口对回复内容进行打分后获得的分数取平均值得到的。

5 实验结果与分析

5.1 对比实验

我们复现了KD、SeqKD、MiniLLM这三种仅在模型最后输出层蒸馏的算法作对照, 分别在这些方法上引入了我们提出的中间层蒸馏算法进行实验, 并在DollyEval、SelfInst、Vicuna、S-NI、UnNI五个指令跟随数据集上进行了蒸馏模型的效果评测, 包括机器评测和人工评测。

Model	Params	Method	DollyEval		SelfInst		VicunaEval		S-NI		UnNI	
			R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4
GPT2	1.5B	Teacher	27.1	60.9	14.7	55.9	16.4	64.0	27.3	40.5	31.5	69.0
		SFT w/o KD	23.0	41.2	10.4	36.4	14.2	35.5	16.5	25.0	18.4	34.3
	120M	KD	22.4	20.6	9.4	28.3	13.7	33.2	15.0	25.8	15.5	39.4
		OurMethod	23.6	20.9	9.8	33.0	15.0	36.1	15.8	30.5	18.0	32.8
		SeqKD	17.8	13.4	7.3	17.1	11.9	22.9	9.5	16.5	11.0	27.6
		OurMethod	18.4	14.7	8.5	23.9	12.5	23.7	12.6	20.3	13.6	26.7
		Minillm	19.3	17.4	9.7	31.3	11.9	36.4	17.0	25.9	18.7	31.0
		OurMethod	19.0	22.2	9.8	35.9	12.2	38.8	16.9	29.8	19.2	37.6
	340M	SFT w/o KD	25.2	61.7*	12.2	42.3	16.1	49.4	20.9	49.7*	24.5	33.9
		KD	22.2	43.1	11.0	42.4	14.2	39.6	18.5	32.2	21.0	36.2
		OurMethod	25.5	53.4	12.3	46.3	15.7	43.5	21.1	37.2	25.9	40.4
		SeqKD	25.0	55.6	12.5	49.4	16.0	46.4	23.1	35.6	26.9	43.8
		OurMethod	25.4	58.2	12.7	45.3	16.8*	46.5	23.1	43.8*	27.4	47.5
		Minillm	21.1	46.8	11.9	40.6	12.6	44.8	20.0	22.8	23.8	47.5
		OurMethod	22.5	50.1	12.2	43.4	14.1	45.5	22.8	35.6	26.3	48.3
		SFT w/o KD	25.7	61.6*	12.1	39.6	16.5*	45.5	21.2	44.5*	24.9	55.6
		KD	24.9	57.0	12.0	40.7	15.9	48.4	20.8	41.8*	23.5	31.5
		OurMethod	25.6	59.7	12.8	45.6	16.2	39.3	23.3	44.5*	26.0	35.2
760M	SeqKD	25.1	58.0	12.6	36.9	16.3	42.2	21.3	42.0*	23.9	24.2	
	OurMethod	26.1	60.1	12.5	44.0	16.4*	42.7	22.5	41.6*	26.2	30.0	
	Minillm	21.8	58.9	13.2	57.5*	13.8	52.2	23.3	55.1*	25.5	33.3	
	OurMethod	23.1	60.4	14.6	54.0	15.9	53.5	25.6	58.3*	26.3	39.8	

Table 1: 机器评测结果。GPT4和R-L分别代表GPT-4和Rouge-L在5个随机种子中的平均反馈分数。加入我们的方法后得分有提升的加粗表示，学生模型的表现优于教师的标记*。

Method	DollyEval	SelfInst	VicunaEval	S-NI	UnNI
KD	4.33	5.53	4.32	4.73	3.16
SeqKD	3.98	4.74	4.21	5.2	2.63
MiniLLM	5.08	6.94	5.4	6.67	3.54
OurMethod	5.13	7.12	6.71	6.75	4.42

Table 2: 人工评测结果。分值范围为1-10分，结果是随机选取指定方法在指定数据集下的生成内容打分后取平均得到的。

5.1.1 机器评测结果

使用Rouge-L和GPT4-score这两个机器评价指标得到的结果如表1。根据结果分析可得：首先，相比于监督微调（SFT w/o KD），引入本文提出的中间层蒸馏损失后的三种基线方法均在大多数情况下获得了更高的RougeL和GPT4分数，且在相同规模的师生蒸馏场景中基本获得了所有蒸馏方案中最高的分数，成功实现了对教师模型的知识蒸馏效果的优化，验证了本文提出算法的有效性。其次，通过比较引入我们提出的中间层蒸馏算法后的三种基线方法（OurMethod）与未引入的基线方法（KD、SeqKD、MiniLLM）在五个数据集上的各指标得分，引入中间层蒸馏损失后的得分普遍获得了提升，进一步验证了本文方法有效提取了输出层之外额外的教师模型信息（中间层），进而实现了蒸馏效果的提升，也说明了本方法对仅输出层蒸馏方法的普适性、可叠加性。

5.1.2 人工评测结果

针对每种蒸馏方法，我们随机抽取了部分生成样本进行人工打分(1-10)，最后取样本平均

分为每种蒸馏算法得分。打分过程中，以ground truth为标准答案，并综合考虑生成内容的流畅性、完整性、正确性以及内容多样性等因素。人工评测结果如表2，可以看出，相对于仅针对输出层蒸馏的算法，引入了中间层损失的蒸馏方法获得了更高的分数，说明使用我们的算法蒸馏得到的学生模型在语言流畅性、语义完整性和灵活性等角度性能得到了提升。

5.2 分析实验

5.2.1 不同中间层对象的蒸馏实验结果

在中间层蒸馏细节上，我们分别利用了两种不同的中间层蒸馏对象进行实验：中间层注意力输出、隐藏层输出。综合表3,4,5的各评测指标得分可得，无论是何种中间层蒸馏对象，引入了中间层蒸馏损失之后，原基线蒸馏方法的指标得分均得到了一定的提升，进一步证明了本文提出的中间层蒸馏算法能有效提升异构生成式大语言模型的蒸馏效果。此外，相较于选择隐藏层输出作为蒸馏对象，引入中间层注意力模块的蒸馏损失在多数情况下得到的学生模型得分更高，蒸馏效果更好，也一定程度上说明了注意力模块可能蕴含更多的中间语义有效信息。

Model	Params	Method	DollyEval		SelfInst		VicunaEval		S-NI		UnNI	
			R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4
GPT2	1.5B	Teacher	27.1	60.9	14.7	55.9	16.4	64.0	27.3	40.5	31.5	69.0
		KD	22.4	20.6	9.4	28.3	13.7	33.2	15.0	25.8	15.5	39.4
		KD+attn	23.6	20.9	9.8	28.4	15.0	34.9	15.8	30.5	18.0	31.6
	120M	KD+layer	23.1	20.7	9.6	33.0	14.5	36.1	15.2	27.0	17.2	32.8
		KD	22.2	43.1	11.0	42.4	14.2	39.6	18.5	32.2	21.0	36.2
		KD+attn	25.5	53.4	12.3	46.3	15.7	43.5	21.1	32.7	25.9	40.4
	340M	KD+layer	24.4	50.9	11.8	43.5	14.3	41.4	20.2	37.2	25.1	39.8
		KD	24.9	57.0	12.0	40.7	15.9	48.4	20.8	41.8	23.5	31.5
		KD+attn	25.6	58.2	12.8	44.5	16.0	39.3	23.3	44.5	26.0	35.2
	760M	KD+layer	25.2	59.7	12.5	45.6	16.2	39.2	21.2	38.2	24.4	33.9

Table 3: 不同中间层蒸馏对象应用于KD实验结果表。其中加粗表示的部分是指我们的方法超过原KD方法。

Model	Params	Method	DollyEval		SelfInst		VicunaEval		S-NI		UnNI	
			R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4
GPT2	1.5B	Teacher	27.1	60.9	14.7	55.9	16.4	64.0	27.3	40.5	31.5	69.0
		SeqKD	17.8	13.4	7.3	17.1	11.9	22.9	9.5	16.5	11.0	27.6
		SeqKD+attn	18.4	14.7	8.5	22.4	12.5	23.1	12.6	20.3	13.6	26.7
	120M	SeqKD+layer	18.0	13.6	7.7	23.9	12.2	23.7	9.7	17.3	11.2	24.8
		SeqKD	25.0	55.6	12.5	49.4	16.0	46.4	23.1	35.6	26.9	43.8
		SeqKD+attn	25.3	58.2	12.7	44.9	16.2	46.5	22.9	43.8	27.4	44.3
	340M	SeqKD+layer	25.4	57.7	12.7	45.3	16.8	46.5	23.2	37.3	27.1	47.5
		SeqKD	25.1	58.0	12.6	36.9	16.3	42.2	21.3	42.0	23.9	24.2
		SeqKD+attn	26.1	59.7	12.8	44.0	16.4	42.7	22.5	41.0	26.2	30.0
	760M	SeqKD+layer	25.8	60.1	12.2	40.3	16.3	42.5	21.6	41.6	24.5	25.8

Table 4: 不同中间层蒸馏对象应用于SeqKD实验结果表。

5.2.2 不同中间层映射规则实验结果

在中间层映射规则的设计上本文主要从两大角度考虑，最后选择最大公因数异构对齐策略：一是参考了之前工作中已有实验验证有效性的倍数映射策略(Jiao et al., 2019)，二是在实际实验中，我们设计了纯倍数映射(Jiao et al., 2019)、随机映射(Haidar et al., 2021)、指定层映射等多种不同的映射策略，具体地，它们的含义是：

Model	Params	Method	DollyEval		SelfInst		VicunaEval		S-NI		UnNI	
			R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4	R-L	GPT4
GPT2	1.5B	Teacher	27.1	60.9	14.7	55.9	16.4	64.0	27.3	40.5	31.5	69.0
	120M	Minillm	19.3	17.4	9.7	31.3	11.9	36.4	17.0	25.9	18.7	31.0
		Minillm+attn	19.5	17.9	9.8	35.9	12.2	38.8	16.9	26.6	19.2	37.6
		Minillm+layer	19.0	22.2	9.8	35.2	12.0	38.7	16.9	29.8	18.8	36.3
	340M	Minillm	21.1	46.8	11.9	40.6	12.6	44.8	20.0	22.8	23.8	47.5
		Minillm+attn	22.5	50.1	12.2	43.4	14.1	45.5	22.8	24.4	26.3	48.4
		Minillm+layer	22.4	49.6	12.4	42.8	13.1	45.2	22.5	35.6	26.1	47.9
	760M	Minillm	21.8	58.9	13.2	57.5	13.8	52.2	23.3	55.1	25.5	33.3
		Minillm+attn	23.1	59.3	14.6	52.4	15.9	53.5	25.6	58.3	26.3	33.4
		Minillm+layer	22.5	60.4	13.6	57.8	15.9	53.3	25.0	56.7	25.8	39.8

Table 5: 不同中间层蒸馏对象应用于MiniLLM实验结果表。

- **纯倍数映射**: 仅适用于教师模型与学生模型的中间层层数成整数倍的情况。是本文设计的最大公因数异构对齐策略中的一个特殊情况。因此实验中没有单独测试。
- **随机映射**: 在每一轮训练时, 随机选取教师模型的中间层与学生模型的中间层建立映射以进行知识蒸馏。
- **指定层映射**: 手动规定教师模型的哪几个中间层对应学生模型的哪几个中间层, 训练前指定, 不同于随机映射, 这里每轮训练所使用的层映射规则不变。
- **最大公因数异构对齐策略**: 详见3.2.1。

我们选取教师模型为GPT2-xlarge, 学生模型为GPT2-base, 蒸馏方法为标准蒸馏 (KD) 上加入本文提出的中间层蒸馏算法, 选取数据集DollyEval, 分别应用上述不同的中间层映射规则开展蒸馏实验, 得到的最终实验结果对比如表6所示, 因此最终选择了效果最好的最大公因数异构对齐策略。

Table 6: 中间层映射规则蒸馏效果对比。加粗表示的为蒸馏效果最好的方案。

中间层映射规则	Rouge-L(\uparrow)	GPT4-score(\uparrow)
随机映射	22.9847	18.66
指定层映射	16.9077	15.14
最大公因数异构对齐策略	23.6245	20.86

5.2.3 不同的中间层损失函数实验结果

为确定3.3节的中间层损失计算模块中使用何种损失函数计算中间层蒸馏损失, 我们分别应用余弦相似性Cosine Similarity、平均绝对误差MAE以及均方误差MSE三种损失函数, 以GPT2-base作为学生模型, GPT2-xlarge作为教师模型, 得到的Rouge-L实验效果对比如表7, 因此, 最终选择了效果最好的MSE作为中间层蒸馏损失函数。

Loss Function	Method		
	KD+OurMethod	SeqKD+OurMethod	MiniLLM+OurMethod
Cosine	22.1	17.2	17.8
MAE	17.4	12.4	15.6
MSE	23.6	18.4	19

Table 7: 不同中间层蒸馏损失函数结果表。加粗的是Rouge-L指标最高的

6 总结

总的来说, 本文提出了一种针对生成式大语言模型的在仅输出层蒸馏之外引入中间层损失的蒸馏方法, 实现上主要包括四部分: 中间层蒸馏对象选择、中间层结果对齐、中间层损失计算以及各损失函数优化融合, 并将此中间层蒸馏方法融入多种现有的仅输出层蒸馏的白盒生成式大语言模型蒸馏方法中。我们首次将中间层蒸馏思想应用于同为Transformer解码器架构但中间层内部结构不同的师生模型中, 设计了一种更加普适于异构师生模型, 蒸馏效果更好的白盒生成式大语言模型蒸馏方法。实验中, 我们提出的蒸馏方法在五个数据集三种不同学生模型规模的多个实验中均获得了超过原基线蒸馏方法的效果, 验证了我们设计的引入中间层蒸馏方法的有效性, 为未来针对生成式大语言模型的异构师生模型间的中间层蒸馏提供了思路与技术借鉴。

7 致谢

该研究得到了国家自然科学基金(62106105)、CCF-百度松果基金(CCF-Baidu202307)、南京航空航天大学科研启动基金(YQR21022)和南京航空航天大学高性能计算平台的支持。

参考文献

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Alvenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua. 2015. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2(3):1.
- Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. Gkd: Generalized knowledge distillation for auto-regressive sequence models. *arXiv preprint arXiv:2306.13649*.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*.
- Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. 2020. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.
- Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.
- Md Akmal Haidar, Nithin Anchuri, Mehdi Rezagholizadeh, Abbas Ghaddar, Philippe Langlais, and Pascal Poupard. 2021. Rail-kd: Random intermediate layer mapping for knowledge distillation. *arXiv preprint arXiv:2109.10164*.
- Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.
- Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793.
- Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of closed-source large language model. *arXiv preprint arXiv:2305.12870*.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.
- Jongwoo Ko, Seungjoon Park, Minchan Jeong, Sukjin Hong, Euijai Ahn, Du-Seong Chang, and Se-Young Yun. 2023. Revisiting intermediate layer distillation for compressing language models: An overfitting perspective. *arXiv preprint arXiv:2302.01530*.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model.
- Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. *arXiv preprint arXiv:2011.00593*.
- Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, pages 13657–13665.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. Are intermediate layers and labels really necessary? a general language model distillation method. *arXiv preprint arXiv:2306.06625*.
- MosaicML NLP Team et al. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL www.mosaicml.com/blog/mpt-7b. Accessed, pages 05–05.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *arXiv preprint arXiv:2308.02019*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020a. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hananeh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2.
- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*.
- Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. 2021. Universal-kd: Attention-based output-grounded intermediate layer knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7649–7661.
- Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *arXiv preprint arXiv:2304.14402*.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A 实验细节

对于GPT-4反馈，我们应用如图4中的提示词模板，并将温度设置为0.7。

```
We would like to request your feedback on the performance of two AI assistants in response to the user instruction and input displayed above. Please rate the helpfulness, relevance, accuracy, and level of detail of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output a single line containing only two values indicating the scores for Assistant 1 and 2 below, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

### Instruction:
{Instruction}

### Input:
{Input}

### Assistant1:
{ground_truth}

### Assistant2:
{Model_response}
```

Figure 4: GPT4提示词模板

B 实例分析

我们在五个实验数据集的测试集上均随机选择了不同的问答数据对作为参照（见表8,10,12,14,16），对比分析了三种仅考虑输出层蒸馏损失的基线方法与本文提出的引入了中间层蒸馏损失的蒸馏方法生成回复的内容，见表9,11,13,15,17，可以发现多数情况下本文所提出的蒸馏方法得到的蒸馏模型生成内容在语义上更接近教师模型给出的回答，且回复内容更完整、连贯、正确性更高（相较于正确答案），进一步验证了本文提出的中间层蒸馏算法对生成式大语言模型蒸馏效果的提升与在多种任务多个数据集上的泛化能力。下面是具体生成内容展示与分析：

- **DollyEval数据集**：根据表8,9分析可得，引入本文提出的中间层蒸馏算法后生成的回复（KD+OurMethod、SeqKD+OurMethod、MiniLLM+OurMethod）都比较精准而完整地回答了提问中四个国家的关联，与教师模型（Teacher）给出的回复更相似，且相比于仅计算输出层损失的基线方法（KD、SeqKD、MiniLLM）语义内容更加完整、更贴合题意。
- **SelfInst数据集**：根据表10,11分析可得，我们提出的蒸馏方法回复内容更符合原问题的答案，且相较于基线方法语义内容更丰富、具有多样性。
- **VicunaEval数据集**：根据表12,13分析可得，在面对需要逻辑推理的计算题类型问答中，我们提出的蒸馏方法更趋向于去推理算式计算的过程，而原基线方法的回答就相较之下略显杂乱、缺乏逻辑。
- **S-NI数据集**：根据表14,15分析可得，在面对摘要类型问答时，我们提出的蒸馏方法回复更精炼，且与原基线方法比较下来更正确地捕获了需提炼句子的真实含义。
- **UnNI数据集**：根据表16,17分析可得，在不考虑回复正确性的前提下，我们提出的方法基本上会以“Here...that have the most anagrams: ”为开头展开回复，相较与原基线方法而言对问题的任务要求理解比较正确，回复格式也更清晰。

Table 8: 实例一, DollyEval数据集指令-回复对

DataSet	DollyEval
Instruction	What links Brazil, Uruguay, Mozambique and Angola?
Input	-
ground truth	Colonies of Portugal

Table 9: 实例一各蒸馏方法生成回复。加粗表示的是指引入了我们提出的中间层蒸馏方法。

Model	Response
Teacher	Brasil, Uruguay, Mozambican and Angola are all countries in South America.
SFT w/o KD	African lion hunting.
KD	Regions of the World.
KD+OurMethod	All of them are famous Portuguese islanders.
SeqKD	Brazil and Uruguay
SeqKD+OurMethod	These 5 countries are all part of the South America.
MiniLLM	The African continent is made up of countries that share a border. Brazil and Mozambique are on the Atlantic and African sea and Uruguay and Angola are on the Pacific and South American sea.
MiniLLM+OurMethod	Brazil, Uruguay and Angola form the African Free Trade Area (AFFTA).

Table 10: 实例二, SelfInst数据集指令-回复对

DataSet	SelfInst
Instruction	What sound does this make?
Input	好
ground truth	hǎo

Table 11: 实例二各蒸馏方法生成回复。

Model	Response
Teacher	This does not make any sound.
SFT w/o KD	This makes a sound when the strings are pulled together.
KD	This sound is a mixture of two other sounds: a truck horn and a working noise machine.
KD+OurMethod	This makes a lot of sense. It makes the listening experience more enjoyable. When you are driving on a windy road, you hear the woodwinds and also the small wailing of babies. The sound reflects off the wind and makes you want to get out of the car faster.
SeqKD	This makes a close approximation of the sound of a car horn.
SeqKD+OurMethod	This makes a sound similar to a dog's bark.
MiniLLM	Aah
MiniLLM+OurMethod	Hai

Table 12: 实例三, VicunaEval数据集指令-回复对

DataSet	VicunaEval
Instruction	Given that $f(x) = 5x^3 - 2x + 3$, find the value of $f(2)$.
Input	-
ground truth	To find the value of $f(2)$, we simply substitute 2 in place of x in the given equation of $f(x)$: $f(2) = 5(2)^3 - 2(2) + 3$ $f(2) = 5(8) - 4 + 3$ $f(2) = 40 - 1$ Therefore, the value of $f(2)$ is 39.

Table 13: 实例三各蒸馏方法生成回复。

Model	Response
Teacher	The value of $f(2)$ is 7.
SFT w/o KD	The value of $f(x)$ is 2. The reason it can be found in two ways is that the square of both those numbers is 4 with 5 as a reminder, which is 4 with x as a capital, so the value of x is 5.
KD	$2 * f = 5x^3$
KD+OurMethod	Given that fx is 4, we can subtract sum of 4's value from the number 2 and get the result of four's value. Therefore, $f(x)$ is 4.
SeqKD	$1a = 2x/c$, so $5x = 5x$
SeqKD+OurMethod	find the value of $f(x - 2, x - 3, a - 1) = 5x^4$
MiniLLM	$F(x)=5x^3 - 2$
MiniLLM+OurMethod	$f(2) = 4$.

Table 14: 实例四, S-NI数据集指令-回复对

DataSet	S-NI
Instruction	In this task, you are given a text of the article. Your task is to generate a headline (title) for this article.
Input	americans have to wait probably another year before they can relish succulent indian mangoes .
ground truth	time not ripe yet for indian mangoes to hit us

Table 17: 实例五各蒸馏方法生成回复。

Model	Response
Teacher	'lives', 'debt card', 'cried'
SFT w/o KD	'lives', 'debts', 'cried', "shoulder", and "impatient" are all anagrams. 'Lives', 'cried', and 'shoulder' are not anagrams.
KD	words: 'lives', 'cried', 'shirt', 'shirt', 'shirt', 'cried', 'shank', 'caught', 'caught', "Vote". You will be able to find the word that best fits in the given set of words.
KD+OurMethod	Here are the words that have the most anagrams:- 'lives', 'Debit card', 'crying', 'shoulder', and 'impatient'.
SeqKD	Here are the words that have the most anagrams: Here is a list of words that have the most anagrams: ['lives', 'debit card', 'broken', 'impatient']; The word 'cried' has an anagrams: 'cried', 'broken', 'cried'. Words that do not have anagrams include: 'crying', 'shoulder', and 'impatient'.
SeqKD+OurMethod	
MiniLLM	The most anagrams can be found are: "lives", "debit card", "cried", "shoulder". Popular anagrams for 'lives' include 'prest', 'happy', 'slaughtered', 'alive'. Anagrams for 'debit card' include 'stayed', 'somewhat', 'purchased', 'watched', 'attracted', 'behaved'. Anagrams for all others include 'dead', 'different', 'error', 'found', 'never', 'forgot', 'decide', 'likes', 'unaware', 'sees', 'more'.
MiniLLM+OurMethod	