ALP-KD: Attention-Based Layer Projection for Knowledge Distillation

ALP-KD: 基于注意力的层投影用于知识蒸馏

Peyman Passban [2,*] , Yimeng Wu [1] , Mehdi Rezagholizadeh [1] , Qun Liu [1]
Peyman Passban [2,*] , Yimeng Wu [1] , Mehdi Rezagholizadeh [1] , Qun Liu [1]
[1] Huawei Noah' s Ark Lab, [2] Amazon
[1] 华为诺亚方舟实验室, [2] 亚马逊
passban.peyman@gmail.com
{yimeng.wu, mehdi.rezagholizadeh, qun.liu}@huawei.com
{yimeng.wu, mehdi.rezagholizadeh, qun.liu}@huawei.com

Abstract

摘要

Knowledge distillation is considered as a training and compression strategy in which two neural networks, namely a teacher and a student, are coupled together during training. The teacher network is supposed to be a trustworthy predictor and the student tries to mimic its predictions. Usually, a student with a lighter architecture is selected so we can achieve compression and yet deliver high-quality results. In such a setting, distillation only happens for final predictions whereas the student could also benefit from teacher's supervision for internal components.

知识蒸馏被视为一种训练和压缩策略，其中两个神经网络，即教师网络和学生网络，在训练过程中相互耦合。教师网络被认为是一个可靠的预测者，而学生则试图模仿其预测。通常，选择一个架构较轻的学生网络，以便实现压缩并提供高质量的结果。在这种设置中，蒸馏仅发生在最终预测上，而学生也可以从教师的监督中受益于内部组件。

Motivated by this, we studied the problem of distillation for intermediate layers. Since there might not be a one-to-one alignment between student and teacher layers, existing techniques skip some teacher layers and only distill from a subset of them. This shortcoming directly impacts quality, so we instead propose a combinatorial technique which relies on attention. Our model fuses teacher-side information and takes each layer's significance into consideration, then performs distillation between combined teacher layers and those of the student. Using our technique, we distilled a 12-layer BERT (Devlin et al. 2019) into 6-, 4-, and 2-layer counterparts and evaluated them on GLUE tasks (Wang et al. 2018). Experimental results show that our combinatorial approach is able to outperform other existing techniques.

受到此启发，我们研究了中间层的蒸馏问题。由于学生层和教师层之间可能没有一一对应的对齐，现有技术跳过了一些教师层，仅从其子集进行蒸馏。这个缺陷直接影响了质量，因此我们提出了一种依赖于注意力的组合技术。我们的模型融合了教师端信息，并考虑了每一层的重要性，然后在组合的教师层和学生层之间进行蒸馏。使用我们的技术，我们将一个 12 层的 BERT(Devlin 等，2019) 蒸馏为 6 层、4 层和 2 层的对应模型，并在 GLUE 任务 (Wang 等，2018) 上进行了评估。实验结果表明，我们的组合方法能够超越其他现有技术。

Introduction

引言

Knowledge distillation (KD) (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015) is a commonly-used technique to reduce the size of large neural networks (Sanh et al. 2019). Apart from this, we also consider it as a complementary and generic add-on to enrich the training process of any neural model (Furlanello et al. 2018).

知识蒸馏 (KD)(Buciluǎ, Caruana 和 Niculescu-Mizil 2006；Hinton, Vinyals 和 Dean 2015) 是一种常用的技术，用于减少大型神经网络的规模 (Sanh 等人 2019)。除此之外，我们还将其视为一种补充性和通用的附加工具，以丰富任何神经模型的训练过程 (Furlanello 等人 2018)。

In KD, a student network(S)is glued to a powerful teacher ($\mathcal{T}$) during training. These two networks can be trained simultaneously or $\mathcal{T}$ can be a pre-trained model. Usually, $\mathcal{T}$ uses more parameters than $\mathcal{S}$ for the same task, therefore it has a higher learning capacity and is expected to provide reliable

predictions. On the other side, $\mathcal{S}$ follows its teacher with a simpler architecture. For a given input, both models provide predictions where those of the student are penalized by an ordinary loss function (using hard labels) as well as predictions received from $\mathcal{T}$ (also known as soft labels).

在 KD 中，一个学生网络 (S) 在训练期间与一个强大的教师（$\mathcal{T}$）相连。这两个网络可以同时训练，或者 $\mathcal{T}$ 可以是一个预训练模型。通常，$\mathcal{T}$ 在同一任务中使用的参数比 $\mathcal{S}$ 多，因此它具有更高的学习能力，并预计能够提供可靠的预测。另一方面，$\mathcal{S}$ 采用更简单的架构跟随其教师。对于给定的输入，这两个模型都提供预测，其中学生的预测受到普通损失函数 (使用硬标签) 以及来自 $\mathcal{T}$ 的预测 (也称为软标签) 的惩罚。

Training a (student) model for a natural language processing (NLP) task can be formalized as a multi-class classification problem to minimize a cross-entropy (ce) loss function, as shown in Equation 1:

为自然语言处理 (NLP) 任务训练一个 (学生) 模型可以形式化为一个多类分类问题，以最小化交叉熵 (ce) 损失函数，如方程 1 所示:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N}\sum_{w\in V}[\mathbb{1}(y_i = w)\times \log p_{\mathcal{S}}(y_i = w \mid x_i, \theta_{\mathcal{S}})] \tag{1}$$

where $\mathbb{1}(\,.\,) is the indicator function, V$ is a vocabulary set (or different classes in a multi-class problem), N is the number of tokens in an input sequence, and y is a prediction of the network $\mathcal{S}$ with a parameter set $\theta_{\mathcal{S}}$ given an input x .

其中 $\mathbb{1}(\,.\,) is the indicator function, V$ 是词汇集 (或多类问题中的不同类别)，N 是输入序列中的标记数量，y 是网络 $\mathcal{S}$ 在给定输入 x 的情况下的预测，参数集为 $\theta_{\mathcal{S}}$ 。

To incorporate teacher' s supervision, KD accompanies $\mathcal{L}_{ce}$ with an auxiliary loss term, $\mathcal{L}_{KD}$, as shown in Equation 2:

为了结合教师的监督，KD 伴随 $\mathcal{L}_{ce}$ 具有一个辅助损失项 $\mathcal{L}_{KD}$ ，如方程 2 所示:

$$\mathcal{L}_{KD} = -\sum_{i=1}^{N}\sum_{w\in V}[p_{\tau}(y_i = w \mid x_i, \theta_{\tau})\times \log p_{\mathcal{S}}(y_i = w \mid x_i, \theta_{\mathcal{S}})] \tag{2}$$

Since $\mathcal{S}$ is trained to behave identically to $\mathcal{T}$, model compression can be achieved if it uses a simpler architecture than its teacher. However, if these two models are the same size KD would still be beneficial. What $\mathcal{L}_{KD}$ proposes is an ensemble technique by which the student is informed about teacher's predictions. The teacher has better judgements and this helps the student learn how much it deviates from true labels.

由于 $\mathcal{S}$ 被训练为与 $\mathcal{T}$ 行为完全相同，如果它使用比其教师更简单的架构，则可以实现模型压缩。然而，如果这两个模型的大小相同，知识蒸馏仍然是有益的。$\mathcal{L}_{KD}$ 提出了一种集成技术，通过该技术，学生可以获得教师的预测信息。教师具有更好的判断力，这有助于学生学习其与真实标签的偏差程度。

This form of KD that is referred to as Regular KD (RKD) throughout this paper, only provides $\mathcal{S}$ with external supervision for final predictions, but this can be extended to other components such as intermediate layers too. The student needs to be aware of the information flow inside teacher's layers and this becomes even more crucial when distilling from deep teachers. Different alternatives have been proposed to this end, which compare networks' internal layers in addition to final predictions (Jiao et al. 2020; Sun et al. 2020, 2019), but they suffer from other types of problems. The main goal in this paper is to study such models and address their shortcomings.

本文中所称的这种知识蒸馏形式为常规知识蒸馏 (RKD)，仅为 $\mathcal{S}$ 提供最终预测的外部监督，但这可以扩展到其他组件，例如中间层。学生需要了解教师层内部的信息流，当从深层教师中提取知识时，这一点变得尤为重要。为此，已经提出了不同的替代方案，这些方案除了最终预测外，还比较网络的内部层 (Jiao et al. 2020; Sun et al. 2020, 2019)，但它们存在其他类型的问题。本文的主要目标是研究此类模型并解决其缺陷。

Problem Definition

问题定义

To utilize intermediate layers' information (and other components in general), a family of models exists that defines a dedicated loss function to measure how much a student diverges from its teacher in terms of internal representations. In particular, if the goal is to distill from an n -layer teacher into an m -layer student, a subset of m (out of n) teacher layers is selected whose outputs are compared to those of student layers (see Equation 3 for more details). Figure 1 illustrates this concept.

为了利用中间层的信息 (以及一般的其他组件)，存在一类模型定义了专门的损失函数，以衡量学生在内部表示方面与教师的偏离程度。特别是，如果目标是从一个 n 层的教师提取知识到一个 m 层的学生，则选择一组 m (来自 n) 教师层，其输出与学生层的输出进行比较 (有关更多细节，请参见方程 3)。图 1 说明了这一概念。

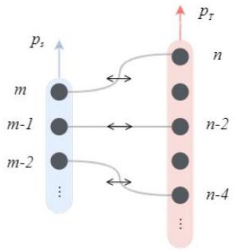

Figure 1: Student and teacher models have m and n layers, respectively. Each node is an intermediate layer and links are cross-model connections. In this example, every other layer of the teacher is skipped in order to match the size of the student. The output of nodes connected to each other are compared via a loss function (shown with $\leftrightarrow$) to ensure that the student model has similar internal representations as its teacher.

图 1: 学生和教师模型分别具有 m 和 n 层。每个节点是一个中间层，链接是跨模型连接。在这个例子中，教师的每隔一层被跳过，以匹配学生的大小。连接在一起的节点的输出通过损失函数 (用 $\leftrightarrow$ 显示) 进行比较，以确保学生模型具有与其教师相似的内部表示。

As the figure shows, each student layer is connected to a single, dedicated peer on the teacher side, e.g. the n -th teacher layer corresponds to the m -th student layer. Since outputs of these two layers are compared to each other, we hope that both models generate as similar outputs as possible at points n and m . With this simple technique, teacher' s knowledge can be used to supervise student's intermediate layers.

如图所示，每个学生层连接到教师端的一个专用对等层，例如，第 n 层教师对应于第 m 层学生。由于这两个层的输出彼此比较，我们希望两个模型在 n 和 m 点生成尽可能相似的输出。通过这种简单的技术，教师的知识可以用来监督学生的中间层。

Experimental results show that intermediate layer matching could be quite effective, but in our study we realized that it may suffer from two shortcomings:

实验结果表明，中间层匹配可能非常有效，但在我们的研究中，我们意识到它可能存在两个缺点:

- If $n \gg m$, multiple layers in $\mathcal{T}$ have to be ignored for distillation but we know that those layers consist of precious information for which we spend expensive resources to learn. This issue is referred to as the skip problem in this paper.

- 如果 $n \gg m$ ，则必须忽略 $\mathcal{T}$ 中的多个层进行蒸馏，但我们知道这些层包含了我们花费大量资源学习的宝贵信息。本文将此问题称为跳过问题。

- Moreover, it seems the way teacher layers are kept/skipped is somewhat arbitrary as there is no particular strategy behind it. Before training, we lack enough knowledge to judge which subset of teacher layers contributes more to the distillation process, so there is a good chance of skipping significant layers if we pick them in an arbitrary fashion. Finding the best subset of layers to distill from requires an exhaustive search or an expert in the field to signify connections. We refer to this issue as the search problem.

- 此外，教师层的保留/跳过方式似乎有些任意，因为没有特别的策略。在训练之前，我们缺乏足够的知识来判断哪一部分教师层对蒸馏过程贡献更大，因此如果我们以任意方式选择它们，就很可能

会跳过重要层。找到最佳的蒸馏层子集需要进行穷举搜索或由领域专家来指明连接。我们将此问题称为搜索问题。

In order to resolve the aforementioned issues we propose an alternative, which is the main contribution of this paper. Our solution does not skip any layer but utilizes all information stored inside $\mathcal{T}$. Furthermore, it combines teacher layers through an attention mechanism, so there is no need to deal with the search problem. We believe that the new notion of combination defined in this paper is as important as our novel KD architecture and can be adapted to other tasks too.

为了解决上述问题，我们提出了一种替代方案，这是本文的主要贡献。我们的解决方案不跳过任何层，而是利用 $\mathcal{T}$ 中存储的所有信息。此外，它通过注意力机制结合教师层，因此无需处理搜索问题。我们相信，本文定义的新组合概念与我们新颖的 KD 架构同样重要，并且可以适应其他任务。

The remainder of this paper is organized as follows: First, we briefly review KD techniques used in similar NLP applications, then we introduce our methodology and explain how it addresses existing shortcomings. We accompany our methodology with experimental results to show whether the proposed technique is useful. Finally, we conclude the paper and discuss future directions.

本文的其余部分组织如下: 首先，我们简要回顾在类似 NLP 应用中使用的 KD 技术，然后介绍我们的方法论并解释它如何解决现有的不足。我们通过实验结果来支持我们的方法论，以展示所提议的技术是否有用。最后，我们总结本文并讨论未来的方向。

Related Work

相关工作

KD was originally proposed for tasks other than NLP (Bu-ciluǎ, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015). Kim and Rush (2016) adapted the idea and proposed a sequence-level extension for machine translation. Freitag, Al-Onaizan, and Sankaran (2017) took a step further and expanded it to a multi-task scenario. Recently, with the emergence of large NLP and language understanding (NLU) models such as ELMO (Peters et al. 2018) and BERT (Devlin et al. 2019) KD has gained extra attention. Deep models can be trained in a better fashion and compressed via KD, which is favorable in many ways. Therefore, a large body of work in the field such as Patient KD (PKD) (Sun et al. 2019) has been devoted to compressing/distilling BERT (and similar) models.

KD 最初是为 NLP 以外的任务提出的 (Bu-ciluǎ, Caruana, 和 Niculescu-Mizil 2006; Hinton, Vinyals, 和 Dean 2015)。Kim 和 Rush(2016) 适应了这一思想，并为机器翻译提出了序列级扩展。Freitag, Al-Onaizan, 和 Sankaran(2017) 进一步扩展了这一概念，应用于多任务场景。最近，随着大型 NLP 和语言理解 (NLU) 模型如 ELMO(Peters et al. 2018) 和 BERT(Devlin et al. 2019) 的出现，KD 受到了更多关注。深度模型可以通过 KD 以更好的方式进行训练和压缩，这在许多方面都是有利的。因此，该领域的大量工作，如患者 KD(PKD)(Sun et al. 2019)，致力于压缩/提炼 BERT(及类似) 模型。

PKD is directly related to this work, so we discuss it in more detail. It proposes a mechanism to match teacher and student models' intermediate layers by defining a third loss function, $\mathcal{L}_P$, in addition to $\mathcal{L}_{ce}$ and $\mathcal{L}_{KD}$, as shown in Equation 3:

PKD 与本研究直接相关，因此我们将其详细讨论。它提出了一种机制，通过定义第三个损失函数 $\mathcal{L}_P$ 来匹配教师和学生模型的中间层，此外还有 $\mathcal{L}_{ce}$ 和 $\mathcal{L}_{KD}$ ，如方程 3 所示:

$$\mathcal{L}_P = -\sum_{i=1}^{N}\sum_{j=1}^{m}\left\| \frac{h_{\mathcal{S}}^{i,j}}{\left\|h_{\mathcal{S}}^{i,j}\right\|_2} - \frac{\mathcal{A}(j)^i}{\left\|\mathcal{A}(j)^i\right\|_2}\right\|_2^2 \tag{3}$$

where $h_{\mathcal{S}}^{i,j}$ is the output [1] of the j -th student layer for the i -th input. A subset of teacher layers selected for distillation is denoted with an alignment function $\mathcal{A}$, e.g. $\mathcal{A}(j) = h_{\mathcal{T}}^{l}$ implies that the output of the j -th student layer should be compared to the output of the l -th teacher layer $\left(h_{\mathcal{S}}^{i,j} \leftrightarrow h_{\mathcal{T}}^{i,l}\right)$.

其中 $h_{\mathcal{S}}^{i,j}$ 是第 j 个学生层对第 i 个输入的输出 [1] 。为蒸馏选择的教师层的子集用对齐函数 $\mathcal{A}$ 表示，例如 $\mathcal{A}(j) = h_{\mathcal{T}}^{l}$ 意味着第 j 个学生层的输出应与第 l 个教师层的输出进行比较 $\left(h_{\mathcal{S}}^{i,j} \leftrightarrow h_{\mathcal{T}}^{i,l}\right)$ 。

PKD is not the only model that utilizes internal layers' information. Other models such as TinyBERT (Jiao et al. 2020) and MobileBERT (Sun et al. 2020) also found it crucial for training competitive student models. However, as Equation 3 shows, in these models only m teacher layers (the number of teacher layers returned by $\mathcal{A}$) can contribute to distillation. In the presence of deep teachers and small students, this limitation can introduce a significant amount of information loss. Furthermore, what is denoted by $\mathcal{A}$

directly impacts quality. If $\mathcal{A}$ skips an important layer the student model may fail to provide high-quality results.

PKD 并不是唯一利用内部层信息的模型。其他模型如 TinyBERT (Jiao et al. 2020) 和 MobileBERT (Sun et al. 2020) 也发现这对于训练具有竞争力的学生模型至关重要。然而，如方程 3 所示，在这些模型中，只有 m 教师层 (由 $\mathcal{A}$ 返回的教师层数量) 可以贡献于蒸馏。在深层教师和小型学生的情况下，这一限制可能导致大量信息损失。此外，$\mathcal{A}$ 所表示的内容直接影响质量。如果 $\mathcal{A}$ 跳过了一个重要层，学生模型可能无法提供高质量的结果。

To tackle this problem, Wu et al. (2020) proposed a combinatorial technique, called CKD. In their model, $\mathcal{A}(j)$ returns a subset of teacher layers instead of a single layer. Those layers are combined together and distillation happens between the combination result and the j -th student layer, as shows in equation 4:

为了解决这个问题，Wu 等人 (2020) 提出了一种组合技术，称为 CKD。在他们的模型中，$\mathcal{A}(j)$ 返回一组教师层的子集，而不是单个层。这些层被组合在一起，并且在组合结果与第 j 个学生层之间进行蒸馏，如方程 4 所示:

$$\widehat{\mathcal{C}}^j = \mathcal{F}_c\left(h_T^k\right); h_T^k \in \mathcal{A}(j)$$

$$\mathcal{C}^j = \mathcal{F}_r\left(\widehat{\mathcal{C}}^j\right) \tag{4}$$

$$\cup_{j=1}^{m}\mathcal{A}(j) = \left\{h_T^1, \ldots, h_T^n\right\}$$

where $\widehat{\mathcal{C}}^j$ is the result of a combination produced by the function $\mathcal{F}_c$ given a subset of teacher layers indicated by $\mathcal{A}(j)$. In Wu et al. (2020), $\mathcal{F}_c$ is implemented via a simple concatenation. Depending on the form of combination used in Equation 4, there might be a dimension mismatch between $\widehat{\mathcal{C}}^j$ and the student layer h_S^j . Accordingly, there is another function, $\mathcal{F}_r$, to reform the combination result into a comparable shape to the student layer. CKD uses a single projection layer to control the dimension mismatch.

其中 $\widehat{\mathcal{C}}^j$ 是由函数 $\mathcal{F}_c$ 生成的组合结果，给定由 $\mathcal{A}(j)$ 指示的教师层子集。在 Wu 等人 (2020) 中，$\mathcal{F}_c$ 通过简单的连接实现。根据方程 4 中使用的组合形式，可能会出现 $\widehat{\mathcal{C}}^j$ 和学生层 h_S^j 之间的维度不匹配。因此，还有另一个函数 $\mathcal{F}_r$ ，用于将组合结果重新调整为与学生层可比较的形状。CKD 使用单个投影层来控制维度不匹配。

With the combination technique (concatenation+projection), CKD could solve the skip problem but the search problem still remains unanswered. Similar to PKD, CKD also requires a search process, but it looks for the best subset of teacher layers instead of the best single layer. These two models are directly related to this research so we consider them as baselines in our experiments.

通过组合技术 (连接 + 投影)，CKD 可以解决跳过问题，但搜索问题仍然没有答案。与 PKD 类似，CKD 也需要一个搜索过程，但它寻找的是最佳教师层子集，而不是最佳单层。这两个模型与本研究直接相关，因此我们将它们视为实验中的基准。

The application of KD in NLP and NLU is not limited to the aforementioned models. Aguilar et al. (2020) followed the same architecture as PKD but they introduced a new training regime, called progressive training. In their method, lower layers are trained first and training is progressively shifted to upper layers. They claim that the way internal layers are trained during KD can play a significant role. Liu et al. (2019) investigated KD from another perspective. Instead of focusing on the compression aspect, they kept the size of student models equal to their teachers and showed how KD could be treated as a complementary training ingredient.

KD 在自然语言处理 (NLP) 和自然语言理解 (NLU) 中的应用并不限于上述模型。Aguilar 等人 (2020) 采用与 PKD 相同的架构，但他们引入了一种新的训练机制，称为渐进训练。在他们的方法中，首先训练较低层，然后逐步转移到较高层。他们声称，在 KD 过程中内部层的训练方式可以发挥重要作用。Liu 等人 (2019) 从另一个角度研究了 KD。他们没有关注压缩方面，而是保持学生模型的大小与教师模型相同，并展示了如何将 KD 视为一种补充训练成分。

Tan et al. (2019) squeezed multiple translation engines into one transformer (Vaswani et al. 2017) and showed that knowledge can be distilled from multiple teachers. Wei et al. (2019) introduced a novel training procedure where there is no need for an external teacher. A student model can learn from its own

[1] By the output, we mean the output of the layer for the CLS token. For more details about CLS see Devlin et al. (2019).
[1] 这里的输出是指 CLS 标记的层输出。有关 CLS 的更多细节，请参见 Devlin et al. (2019)。

checkpoints. At each validation step, if the current checkpoint is better than the best existing checkpoint, student learns from it otherwise the best stored checkpoint is considered as a teacher.

Tan 等人 (2019) 将多个翻译引擎压缩到一个变换器 (Vaswani 等人 2017) 中，并展示了知识可以从多个教师中提取。Wei 等人 (2019) 引入了一种新颖的训练程序，其中不需要外部教师。学生模型可以从其自身的检查点中学习。在每个验证步骤中，如果当前检查点优于现有最佳检查点，则学生从中学习，否则将最佳存储检查点视为教师。

Methodology

方法论

For a given student model $\mathcal{S}$ and a teacher model $\mathcal{T}$ we show all intermediate layers with sets $H_{\mathcal{S}} = \left\{h_{\mathcal{S}}^1, \ldots, h_{\mathcal{S}}^m\right\}$ and $H_{\mathcal{T}} = \left\{h_{\mathcal{T}}^1, \ldots, h_{\mathcal{T}}^n\right\}$, respectively. Based on the pipeline designed by current models for intermediate layer KD, there must be a connection between $H_{\mathcal{S}}$ and $H_{\mathcal{T}}$ during training and each student layer can only correspond to a single peer on the teacher side. As previously mentioned, layer connections are denoted by $\mathcal{A}$.

对于给定的学生模型 $\mathcal{S}$ 和教师模型 $\mathcal{T}$ ，我们展示所有中间层及其对应的集合 $H_{\mathcal{S}} = \left\{h_{\mathcal{S}}^1, \ldots, h_{\mathcal{S}}^m\right\}$ 和 $H_{\mathcal{T}} = \left\{h_{\mathcal{T}}^1, \ldots, h_{\mathcal{T}}^n\right\}$ 。根据当前模型为中间层 KD 设计的管道，在训练过程中 $H_{\mathcal{S}}$ 和 $H_{\mathcal{T}}$ 之间必须存在连接，并且每个学生层只能对应教师端的一个同级层。如前所述，层连接用 $\mathcal{A}$ 表示。

A common heuristic to devise $\mathcal{A}$ is to divide teacher layers into m buckets with approximately the same sizes and pick only one layer from each (Jiao et al. 2020; Sun et al. 2019). Therefore, for the j -th layer of the student model, $\mathcal{A}(j)$ returns a single teacher layer among those that reside in the j -th bucket. Figure 2a illustrates this setting. Clearly, this is not the best way of connecting layers, because they are picked in a relatively arbitrary manner. More importantly, no matter what heuristic is used there still remain $n - m$ layers in this approach whose information is not used in distillation.

一种常见的启发式方法是将教师层划分为具有大致相同大小的 m 桶，并仅从每个桶中选择一层 (Jiao et al. 2020; Sun et al. 2019)。因此，对于学生模型的 j -th 层，$\mathcal{A}(j)$ 从位于 j -th 桶中的层中返回一个单一的教师层。图 2a 说明了这一设置。显然，这并不是连接层的最佳方式，因为它们是以相对任意的方式选择的。更重要的是，无论使用何种启发式方法，这种方法仍然存在 $n - m$ 层，其信息在蒸馏中未被使用。

To address this issue, we simply propose a combinatorial alternative whereby all layers inside buckets are taken into consideration. Our technique is formulated in Equation 5:

为了解决这个问题，我们简单地提出了一种组合替代方案，其中考虑了桶内的所有层。我们的技术在方程 5 中进行了公式化:

$$\begin{aligned} \mathcal{C}^j &= \sum_{h_{\mathcal{T}}^k \in \mathcal{A}(j)} \alpha_{jk} h_{\mathcal{T}}^k \\ \alpha_{jk} &= \frac{\exp\left(h_{\mathcal{S}}^j \cdot h_{\mathcal{T}}^k\right)}{\sum_{h_{\mathcal{T}}^{k'} \in \mathcal{A}(j)} \exp\left(h_{\mathcal{S}}^j \cdot h_{\mathcal{T}}^{k'}\right)} \\ \cup_{j=1}^{m} \mathcal{A}(j) &= H_{\mathcal{T}} = \left\{h_{\mathcal{T}}^1, \ldots, h_{\mathcal{T}}^n\right\} \end{aligned} \tag{5}$$

This idea is similar to that of CKD, but we use an attention mechanism (Bahdanau, Cho, and Bengio 2015) instead of a concatenation for layer combination. Experimental results demonstrate that this form of combination is more useful. We refer to this idea as Attention-based Layer Projection for KD or ALP-KD in short.

这个想法类似于 CKD，但我们使用注意力机制 (Bahdanau, Cho, and Bengio 2015) 而不是连接来进行层组合。实验结果表明，这种组合形式更为有效。我们将这个想法称为基于注意力的层投影用于知识蒸馏，简称 ALP-KD。

According to the equation, if a student layer associates with a particular bucket, all layers inside that bucket are combined/used for distillation and $\mathcal{C}^j$ is a vector representation of such a combination. Our model benefits from all n teacher layers and skips none as there is a dedicated $\mathcal{C}$ vector for each student layer. Figure 2 b visualizes this setting.

根据方程，如果学生层与特定桶相关联，则该桶内的所有层都被组合/用于蒸馏，$\mathcal{C}^j$ 是这种组合的向量表示。我们的模型受益于所有 n 教师层，并且没有跳过任何层，因为每个学生层都有一个专用的 $\mathcal{C}$ 向量。图 2 b 可视化了这一设置。

Weights (α values) assigned to teacher layers are learnable parameters whose values are optimized during training. They show the contribution of each layer to the distillation process. They also reflect the correlation between student and teacher layers, i.e. if a student layer correlates more with a set of teacher layers weights connecting them should receive higher values. In other words, that specific layer is playing the role of its teacher peers on the student side. To measure the correlation, we use the dot product in our experiments but any other function for similarity estimation could be used in this regard.

分配给教师层的权重 (α 值) 是可学习的参数，其值在训练过程中被优化。它们显示了每一层对蒸馏过程的贡献。它们还反映了学生层与教师层之间的相关性，即如果一个学生层与一组教师层的相关性更高，则连接它们的权重应获得更高的值。换句话说，该特定层在学生端扮演着其教师同伴的角色。为了测量相关性，我们在实验中使用点积，但在这方面可以使用任何其他相似性估计函数。

Equation 5 addresses the skip problem with a better combination mechanism and is able to provide state-of-the-art results. However, it still suffers from the search problem as it relies on buckets and we are not sure which bucketing strategy works better. For example, in Figure 2b the first bucket consists of the first three layers of the teacher but it does not mean that we cannot append a fourth layer. In fact, a bucket with four layers might perform better. Buckets can also share layers; namely, a teacher layer can belong to multiple buckets and can be used numerous times in distillation. These constraints make it challenging to decide about buckets and their boundaries, but it is possible to resolve this dilemma through a simple modification in our proposed model.

方程 5 通过更好的组合机制解决了跳过问题，并能够提供最先进的结果。然而，它仍然面临搜索问题，因为它依赖于桶，而我们不确定哪种分桶策略效果更好。例如，在图 2b 中，第一个桶由教师的前三层组成，但这并不意味着我们不能添加第四层。实际上，包含四层的桶可能表现得更好。桶也可以共享层；即，一个教师层可以属于多个桶，并且可以在蒸馏中多次使用。这些限制使得决定桶及其边界变得具有挑战性，但通过对我们提出的模型进行简单修改，可以解决这一困境。

To avoid bucketing, we span the attention mask over all teacher layers rather than over buckets. To implement this extension, $\mathcal{A}(j)$ needs to be replaced with $H_{\mathcal{T}}$ in Equation 5. Therefore, for any student layer such as $h_{\mathcal{S}}^{j}$ there would be a unique set of n attention weights and $\mathcal{C}^{j}$ would be a weighted average of all teacher layers, as shown in Equation 6:

为了避免分桶，我们将注意力掩码扩展到所有教师层，而不是分桶。为了实现这一扩展，需要在方程 5 中将 $\mathcal{A}(j)$ 替换为 $H_{\mathcal{T}}$ 。因此，对于任何学生层，例如 $h_{\mathcal{S}}^{j}$ ，将会有一组独特的 n 注意力权重，而 $\mathcal{C}^{j}$ 将是所有教师层的加权平均，如方程 6 所示:

$$\mathcal{C}^{j} = \sum_{h_{\mathcal{T}}^{k} \in \mathcal{A}(j)} \alpha_{jk} h_{\mathcal{T}}^{k} \tag{6}$$

$$\mathcal{A}(j) = H_{\mathcal{T}} \forall j \in \{1, 2, \ldots, m\}$$

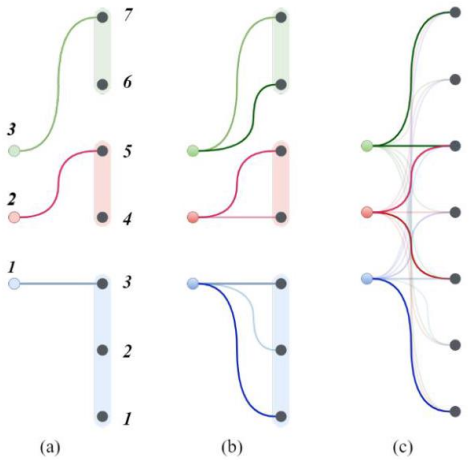

Figure 2: Three pairs of $\mathcal{S}$ and $\mathcal{T}$ networks with different forms of layer connections. In Figure 2a, teacher layers are divided into 3 buckets and only one layer from each bucket is connected to the student side, e.g. $h_{\mathcal{T}}^{5}$ is the source of distillation for $h_{\mathcal{S}}^{2}$ $\left(h_{\mathcal{T}}^{5} \leftrightarrow h_{\mathcal{S}}^{2}\right)$. In Figure 2b, a weighted average of teacher layers from each bucket is considered for distillation, e.g. $\mathcal{A}(2) = \left\{h_{\mathcal{T}}^{4}, h_{\mathcal{T}}^{5}\right\}$ and $\mathcal{C}^{2} = \alpha_{24} h_{\mathcal{T}}^{4} + \alpha_{25} h_{\mathcal{T}}^{5}$ $\left(\mathcal{C}^{2} \leftrightarrow h_{\mathcal{S}}^{2}\right)$. In Figure 2c , there is no bucketing and all teacher layers are considered for projection. Links with higher color intensities have higher attention weights.

图 2: 三对具有不同层连接形式的 $\mathcal{S}$ 和 $\mathcal{T}$ 网络。在图 2a 中，教师层被分为 3 个桶，每个桶中只有一层连接到学生端，例如 $h_{\mathcal{T}}^{5}$ 是 $h_{\mathcal{S}}^{2}\left(h_{\mathcal{T}}^{5} \leftrightarrow h_{\mathcal{S}}^{2}\right)$ 的蒸馏源。在图 2b 中，考虑了来自每个桶的教师层的加权平均进行蒸馏，例如 $\mathcal{A}(2)=\left\{h_{\mathcal{T}}^{4}, h_{\mathcal{T}}^{5}\right\}$ 和 $\mathcal{C}^{2}=\alpha_{24} h_{\mathcal{T}}^{4}+\alpha_{25} h_{\mathcal{T}}^{5}\left(\mathcal{C}^{2} \leftrightarrow h_{\mathcal{S}}^{2}\right)$ 。在图 2c 中，没有分桶，所有教师层都被考虑用于投影。颜色强度较高的链接具有更高的注意权重。

This new configuration, which is illustrated in Figure 2c, proposes a straightforward way of combining teacher layers and addresses both skip and search problems at the same time.

这种新配置在图 2c 中进行了说明，提出了一种简单的方式来组合教师层，并同时解决跳过和搜索问题。

To train our student models, we use a loss function which is composed of $\mathcal{L}_{ce}, \mathcal{L}_{KD}$, and a dedicated loss defined for ALP-KD, as shown in Equation 7:

为了训练我们的学生模型，我们使用一个由 $\mathcal{L}_{ce}, \mathcal{L}_{KD}$ 组成的损失函数，以及为 ALP-KD 定义的专用损失，如方程 7 所示:

$$\mathcal{L}=\beta \mathcal{L}_{ce}+\eta \mathcal{L}_{KD}+\lambda \mathcal{L}_{ALP}$$

$$\mathcal{L}_{ALP}=\sum_{i=1}^{N} \sum_{j=1}^{m} \operatorname{MSE}\left(h_{\mathcal{S}}^{i, j}, \mathcal{C}^{i, j}\right) \tag{7}$$

where $MSE()$ is the mean-square error and $\mathcal{C}^{i,j}$ shows the value of $\mathcal{C}^{j}$ when the teacher is fed with the i -th input. β, η , and λ are hyper-parameters of our model to minimize the final loss.

其中 $MSE()$ 是均方误差，$\mathcal{C}^{i,j}$ 显示当教师输入 i -th 输入时 $\mathcal{C}^{j}$ 的值。β, η 和 λ 是我们模型的超参数，用于最小化最终损失。

Experimental Study

实验研究

A common practice in our field to evaluate the quality of a KD technique is to feed $\mathcal{T}$ and $\mathcal{S}$ models with instances of standard datasets and measure how they perform. We followed the same tradition in this paper and selected a set of eight GLUE tasks (Wang et al. 2018) including CoLA, MNLI, MRPC, QNLI, QQP, RTE, SST-2, and STS-B datasets to benchmark our models. Detailed information about datasets is available in the appendix section.

在我们领域中，评估 KD 技术质量的常见做法是用标准数据集的实例对 $\mathcal{T}$ 和 $\mathcal{S}$ 模型进行测试，并测量它们的表现。我们在本文中遵循了这一传统，选择了一组八个 GLUE 任务 (Wang et al. 2018)，包括 CoLA、MNLI、MRPC、QNLI、QQP、RTE、SST-2 和 STS-B 数据集，以基准我们的模型。关于数据集的详细信息可在附录部分找到。

In NLP/NLU settings, $\mathcal{T}$ is usually a pre-trained model whose parameters are only fine-tuned during training. On the other side, $\mathcal{S}$ can be connected to $\mathcal{T}$ to be trained thoroughly or can alternatively be initialized with $\mathcal{T}$ ' s parameters to be fine-tuned similar to its teacher. This helps the student network generate better results and converge faster. Fine-tuning is more common than training in our context and we thus fine-tune our models rather than training. This concept is comprehensively discussed by Devlin et al. (2019) so we skip its details and refer the reader to their paper. We have the same fine-tuning pipeline in this work.

在自然语言处理/自然语言理解的环境中，$\mathcal{T}$ 通常是一个预训练模型，其参数仅在训练期间进行微调。另一方面，$\mathcal{S}$ 可以连接到 $\mathcal{T}$ 进行全面训练，或者可以选择用 $\mathcal{T}$ 的参数进行初始化，以便类似于其教师进行微调。这有助于学生网络生成更好的结果并更快收敛。在我们的背景下，微调比训练更为常见，因此我们选择微调我们的模型，而不是进行训练。Devlin 等人 (2019) 对这一概念进行了全面讨论，因此我们省略其细节，并将读者引导至他们的论文。我们在本工作中采用相同的微调流程。

In our experiments, we chose the original BERT model [2] (also known as BERT $_{\text{Base}}$) as our teacher. We are faithful to the configuration proposed by Devlin et al. (2019) for it. Therefore, our in-house version also has 12 layers with 12 attention heads and the hidden and feed-forward dimensions are 768 and 3072, respectively. Our students are also BERT models only with fewer layers ($|H_{\mathcal{S}}|=m; m<12$) . We use the teacher BERT to initialize students, but because the number of layers are different ($12 \neq m$) we only consider its first m layers. We borrowed this idea from PKD (Sun et al. 2019) in the interest of fair comparisons.

在我们的实验中，我们选择了原始的 BERT 模型 [2] (也称为 BERT $_{\text{Base}}$) 作为我们的教师。我们忠实于 Devlin 等人 (2019) 为其提出的配置。因此，我们的内部版本也具有 12 层，12 个注意力头，隐藏

层和前馈维度分别为 768 和 3072。我们的学生也是 BERT 模型，只是层数较少 ($|H_{\mathcal{S}}| = m; m < 12$) 。我们使用教师 BERT 来初始化学生，但由于层数不同 ($12 \neq m$) ，我们仅考虑其前 m 层。我们借鉴了 PKD(Sun et al. 2019) 的这一思路，以便进行公平比较。

In order to maximize each student's performance we need to decide about the learning rate, batch size, the number of fine-tuning iterations, and β, η , and λ . To this end, we run a grid search similar to Sun et al. (2019) and Wu et al. (2020). In our setting, the batch size is set to 32 and the learning rate is selected from $\{1e-5, 2e-5, 5e-5\}$.η and λ take values from $\{0, 0.2, 0.5, 0.7\}$ and $\beta = 1 - \eta - \lambda$.

为了最大化每个学生的表现，我们需要决定学习率、批量大小、微调迭代次数，以及 β, η 和 λ 。为此，我们进行了类似于 Sun 等人 (2019) 和 Wu 等人 (2020) 的网格搜索。在我们的设置中，批量大小设置为 32，学习率从 $\{1e-5, 2e-5, 5e-5\}$.η 中选择，而 λ 的取值范围为 $\{0, 0.2, 0.5, 0.7\}$ 和 $\beta = 1 - \eta - \lambda$ 。

We trained multiple models with different configurations and compared our results to RKD- and PKD-based students. To the best of our knowledge, these are the only alternatives that use BERT as a teacher and their students' architecture relies on ordinary Transformer blocks (Vaswani et al. 2017) with the same size as ours, so any comparison to any other model with different settings would not be fair. Due to CKD's similarity to our approach we also re-implemented it in our experiments. The original CKD model was proposed for machine translation and for the first time we evaluate it in NLU tasks. Table 1 summarizes our experiments.

我们训练了多个具有不同配置的模型，并将我们的结果与基于 RKD 和 PKD 的学生进行了比较。根据我们所知，这些是唯一使用 BERT 作为教师的替代方案，其学生的架构依赖于与我们相同大小的普通 Transformer 块 (Vaswani 等人 2017)，因此与任何其他具有不同设置的模型进行比较是不公平的。由于 CKD 与我们的方法相似，我们在实验中也重新实现了它。原始 CKD 模型是为机器翻译提出的，我们首次在 NLU 任务中对其进行了评估。表 1 总结了我们的实验。

The teacher model with 12 layers and 109M parameters has the best performance for all datasets. [3] This model can be compressed, so we reduce the number of layers to 4 and train another model ($\mathcal{S}_{\text{NKD}}$) . The rest of the configuration (attention

具有 12 层和 109M 参数的教师模型在所有数据集上表现最佳。[3] 该模型可以被压缩，因此我们将层数减少到 4 并训练了另一个模型 ($\mathcal{S}_{\text{NKD}}$) 。其余配置 (注意力

Problem	Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
N/A	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
N/A	$\mathcal{S}_{\text{NKD}}$	31.05	76.83	77.70	85.13	88.97	61.73	88.19	87.29	74.61
skip, search	$\mathcal{S}_{\text{RKD}}$	29.22	79.31	79.41	86.77	90.25	65.34	90.37	87.45	76.02
skip, search	$\mathcal{S}_{\text{PKD}}$	32.13	79.26	80.15	86.64	90.23	65.70	90.14	87.26	76.44
search	$\mathcal{S}_{\text{CKD-NO}}$	31.23	79.42	80.64	86.93	88.70	66.06	90.37	87.62	76.37
search	$\mathcal{S}_{\text{CKD-PO}}$	31.95	79.53	80.39	86.75	89.89	67.51	90.25	87.55	76.73
search	$\mathcal{S}_{\text{ALP-NO}}$	34.21	79.26	79.66	87.11	90.72	65.70	90.37	87.52	76.82
search	$\mathcal{S}_{\text{ALP.P.P.O}}$	33.86	79.74	79.90	86.95	90.25	66.43	90.48	87.52	76.89
none	$\mathcal{S}_{\text{ALP}}$	33.07	79.62	80.72	87.02	90.54	67.15	90.37	87.62	77.01

问题	模型	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	平均
不适用	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
不适用	$\mathcal{S}_{\text{NKD}}$	31.05	76.83	77.70	85.13	88.97	61.73	88.19	87.29	74.61
跳过，搜索	$\mathcal{S}_{\text{RKD}}$	29.22	79.31	79.41	86.77	90.25	65.34	90.37	87.45	76.02
跳过，搜索	$\mathcal{S}_{\text{PKD}}$	32.13	79.26	80.15	86.64	90.23	65.70	90.14	87.26	76.44
搜索	$\mathcal{S}_{\text{CKD-NO}}$	31.23	79.42	80.64	86.93	88.70	66.06	90.37	87.62	76.37
搜索	$\mathcal{S}_{\text{CKD-PO}}$	31.95	79.53	80.39	86.75	89.89	67.51	90.25	87.55	76.73
搜索	$\mathcal{S}_{\text{ALP-NO}}$	34.21	79.26	79.66	87.11	90.72	65.70	90.37	87.52	76.82
搜索	$\mathcal{S}_{\text{ALP.P.P.O}}$	33.86	79.74	79.90	86.95	90.25	66.43	90.48	87.52	76.89
无	$\mathcal{S}_{\text{ALP}}$	33.07	79.62	80.72	87.02	90.54	67.15	90.37	87.62	77.01

Table 1: Except the teacher ($\mathcal{T}_{\text{RERT}}$) which is a 12-layer model, all other models have 4 layers. Apart from the number of layers, all students have the same architecture as the teacher. The first column shows

[2] https://github.com/google-research/bert

[2] https://github.com/google-research/bert

[3] Similar to other papers, we evaluate our models on validation sets. Testset labels of GLUE datasets are not publicly available and researchers need to participate in leaderboard competitions to evaluate their models on testsets.

[3] 与其他论文类似，我们在验证集上评估我们的模型。GLUE 数据集的测试集标签不可公开获取，研究人员需要参与排行榜竞赛以在测试集上评估他们的模型。

what sort of problems each model suffers from. NKD stands for $NoKD$ which means there is no KD technique involved during training this student model. NO and PO are different configurations for mapping internal layers. Boldfaced numbers show the best student score for each column over the validation set. Scores in the first column are Matthew's Correlations. SST-B scores are Pearson correlations and the rest are accuracy scores.

表 1: 除了教师模型 ($\mathcal{T}_{\text{RERT}}$) 是一个 12 层模型外，所有其他模型均为 4 层。除了层数之外，所有学生模型与教师模型具有相同的架构。第一列显示每个模型所面临的问题类型。NKD 代表 $NoKD$ ，这意味着在训练该学生模型时没有涉及 KD 技术。NO 和 PO 是用于映射内部层的不同配置。粗体数字显示每列在验证集上的最佳学生得分。第一列的得分为马修相关系数。SST-B 得分为皮尔逊相关系数，其余为准确率得分。

heads, hidden dimension etc) remains untouched. There is no connection between the teacher and $\mathcal{S}_{\text{NKD}}$ and it is trained separately with no KD technique. Because of the number of layers, performance drops in this case but we still gain a lot in terms of memory as this new model only has 53M parameters. To bridge the performance gap between the teacher and $\mathcal{S}_{\text{NKD}}$, we involve KD in the training process and train new models, $\mathcal{S}_{\text{RKD}}$ and $\mathcal{S}_{\text{PKD}}$, with RKD and PKD techniques, respectively.

头部、隐藏维度等保持不变。教师模型与 $\mathcal{S}_{\text{NKD}}$ 之间没有连接，并且它是单独训练的，没有 KD 技术。由于层数的原因，性能在这种情况下下降，但我们在内存方面仍然获得了很大的收益，因为这个新模型只有 53M 个参数。为了缩小教师模型与 $\mathcal{S}_{\text{NKD}}$ 之间的性能差距，我们在训练过程中引入 KD，并分别使用 RKD 和 PKD 技术训练新模型 $\mathcal{S}_{\text{RKD}}$ 和 $\mathcal{S}_{\text{PKD}}$ 。

$\mathcal{S}_{\text{RKD}}$ is equivalent to a configuration known as DistilBERT in the literature (Sanh et al. 2019). To have precise results and a better comparison, we trained/fine-tuned all models in the same experimental environment. Accordingly, we do not borrow any result from the literature but reproduce them. This is the reason we use the term equivalent for these two models. Furthermore, DistilBERT has an extra Cosine embedding loss in addition to those of $\mathcal{S}_{\text{RKD}}$. When investigating the impact of intermediate layers in the context of KD, we wanted $\mathcal{L}_P$ to be the only difference between RKD and PKD, so incorporating any other factor could hurt our investigation and we thus avoided the cosine embedding loss in our implementation.

$\mathcal{S}_{\text{RKD}}$ 等价于文献中称为 DistilBERT 的配置 (Sanh et al. 2019)。为了获得精确的结果和更好的比较，我们在相同的实验环境中训练/微调了所有模型。因此，我们没有借用文献中的任何结果，而是重现了它们。这就是我们将这两个模型称为等价的原因。此外，DistilBERT 除了 $\mathcal{S}_{\text{RKD}}$ 的损失外，还有额外的余弦嵌入损失。在研究 KD 中间层的影响时，我们希望 $\mathcal{L}_P$ 是 RKD 和 PKD 之间唯一的区别，因此任何其他因素的引入可能会影响我们的研究，因此我们在实现中避免了余弦嵌入损失。

PKD outperforms RKD with an acceptable margin in Table 1 and that is because of the engagement of intermediate layers. For $\mathcal{S}_{\text{PKD}}$, we divided teacher layers into 3 buckets (4 layers in each) and picked the first layer of each bucket to connect to student layers, i.e. $\mathcal{A}(1) = h_{\mathcal{T}}^1, \mathcal{A}(2) = h_{\mathcal{T}}^5$, and $\mathcal{A}(3) = h_{\mathcal{T}}^9$. There is no teacher layer assigned to the last layer of the student. This form of mapping maximizes PKD's performance and we figured out this via an empirical study.

PKD 在表 1 中以可接受的幅度超越了 RKD，这是由于中间层的参与。对于 $\mathcal{S}_{\text{PKD}}$ ，我们将教师层分为 3 个桶 (每个桶 4 层)，并选择每个桶的第一层与学生层连接，即 $\mathcal{A}(1) = h_{\mathcal{T}}^1, \mathcal{A}(2) = h_{\mathcal{T}}^5$ 和 $\mathcal{A}(3) = h_{\mathcal{T}}^9$ 。最后一层的学生没有分配教师层。这种映射形式最大化了 PKD 的性能，我们通过实证研究发现了这一点。

Results discussed so far demonstrate that cross-model layer mapping is effective, but it can be improved even more if the skip issue is settled. Therefore, we trained two other students using CKD. The setting for these models is identical to PKD, namely teacher layers are divided into 3 buckets. The first 4 teacher layers reside in the first bucket. The fifth to eighth layers are in the second bucket and the rest are covered by the third bucket. Layers inside the first bucket are concatenated and passed through a projection layer to match the student layers' dimension. The combination result for the first bucket is assigned to the first student layer $\left(\mathcal{C}^1 \leftrightarrow h_{\mathcal{S}}^1\right)$. The same procedure is repeated with the second and third buckets for $h_{\mathcal{S}}^2$ and $h_{\mathcal{S}}^3$. Similar to PKD, there is no teacher layer connected to the last student layer. This configuration is referred to as No Overlap (NO), that indicates buckets share no layers with each other.

到目前为止讨论的结果表明，跨模型层映射是有效的，但如果解决跳过问题，它可以进一步改进。因此，我们使用 CKD 训练了另外两个学生。这些模型的设置与 PKD 相同，即教师层被分为 3 个桶。前 4 个教师层位于第一个桶中。第五到第八层位于第二个桶中，其余层则包含在第三个桶中。第一个桶中的层被串联并通过一个投影层，以匹配学生层的维度。第一个桶的组合结果被分配给第一个学生层 $\left(\mathcal{C}^1 \leftrightarrow h_{\mathcal{S}}^1\right)$ 。对于 $h_{\mathcal{S}}^2$ 和 $h_{\mathcal{S}}^3$ ，相同的过程在第二个和第三个桶中重复。与 PKD 类似，最后一个学生层没有连接到教师层。此配置称为无重叠 (NO)，表示桶之间没有共享层。

In addition to **NO** we designed a second configuration, **PO** , which stands for Partial Overlap. In PO,

each bucket shares its first layer with the preceding bucket, so the first bucket includes the first to fifth layers, the second bucket includes the fifth to ninth layers, and from the ninth layer onward reside in the third bucket. We explored this additional configuration to see the impact of different bucketing strategies in CKD.

除了 **NO** ，我们设计了第二种配置 **PO** ，代表部分重叠。在 PO 中，每个桶与前一个桶共享其第一层，因此第一个桶包括第一到第五层，第二个桶包括第五到第九层，从第九层开始的层则位于第三个桶中。我们探索了这个额外的配置，以观察不同分桶策略在 CKD 中的影响。

Comparing $\mathcal{S}_{\text{CKD}}$ to $\mathcal{S}_{\text{PKD}}$ shows that the combination (concatenation+projection) idea is useful in some cases, but for others the simple skip idea is still better. Even defining different bucketing strategies did not change it drastically, and this leads us to believe that a better form of combination such as an attention-based model is required.

比较 $\mathcal{S}_{\text{CKD}}$ 和 $\mathcal{S}_{\text{PKD}}$ 显示，组合 (串联 + 投影) 思想在某些情况下是有用的，但在其他情况下，简单的跳过思想仍然更好。即使定义不同的分桶策略也没有显著改变这一点，这使我们相信需要一种更好的组合形式，例如基于注意力的模型。

In $\mathcal{S}_{\text{ALP}}$ extensions, we replace the CKD' s concatenation with attention and results improve. ALP-KD is consistently better than all other RKD, PKD, and CKD variations and this justifies the necessity of using attention for combination. $\mathcal{S}_{\text{ALP-NO}}$ and $\mathcal{S}_{\text{ALP-PO}}$ also directly support this claim. In $\mathcal{S}_{\text{ALP}}$, we followed Equation 6 and spanned the attention mask over all teacher layers. This setting provides a model that requires no engineering adjustment to deal with skip and search problems and yet delivers the best result on average.

在 $\mathcal{S}_{\text{ALP}}$ 扩展中，我们用注意力替代了 CKD 的连接，结果得到了改善。ALP-KD 始终优于所有其他 RKD、PKD 和 CKD 变体，这证明了使用注意力进行组合的必要性。$\mathcal{S}_{\text{ALP-NO}}$ 和 $\mathcal{S}_{\text{ALP-PO}}$ 也直接支持这一说法。在 $\mathcal{S}_{\text{ALP}}$ 中，我们遵循了方程 6，并在所有教师层上展开了注意力掩码。这个设置提供了一个模型，无需工程调整即可处理跳过和搜索问题，并且在平均情况下提供了最佳结果。

Training Deeper/Shallower Models Than 4-Layer Students

训练比 4 层学生更深/更浅的模型

So far we compared 4-layer ALP-KD models to others and observed superior results. In this section, we design additional experiments to study our technique's behaviour from the size perspective. The original idea of PKD was proposed to distill from a 12-layer BERT to a 6-layer student (Sun et al. 2019). In such a scenario, only every other layer of the teacher is skipped and it seems the student model should not suffer from the skip problem dramatically. We repeated this experiment to understand if our combination idea is still useful or its impact diminishes when student and teacher models have closer architectures. Table 2 summarizes findings of this experiment.

到目前为止，我们将 4 层 ALP-KD 模型与其他模型进行了比较，并观察到更优的结果。在本节中，我们设计了额外的实验，从规模的角度研究我们技术的表现。PKD 的原始思想是从 12 层 BERT 蒸馏到 6 层学生 (Sun et al. 2019)。在这种情况下，只有每隔一层的教师被跳过，似乎学生模型不应该在跳过问题上受到显著影响。我们重复了这个实验，以了解我们的组合思想是否仍然有用，或者当学生和教师模型具有更接近的架构时，其影响是否减弱。表 2 总结了这个实验的发现。

Problem	Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
N/A	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
N/A	$\mathcal{S}_{\text{NKD}}$	40.33	79.91	81.86	87.57	90.21	65.34	90.02	88.49	77.97
skip, search	$\mathcal{S}_{\text{RKD}}$	45.51	81.41	83.82	88.21	90.56	67.51	91.51	88.70	79.65
skip, search	$\mathcal{S}_{\text{PKD}}$	45.78	82.18	85.05	89.31	90.73	68.23	91.51	88.56	80.17
search	$\mathcal{S}_{\text{CKD-NO}}$	48.49	81.91	83.82	89.53	90.64	67.51	91.40	88.73	80.25
search	$\mathcal{S}_{\text{CKD-PO}}$	46.99	81.99	83.82	89.44	90.82	67.51	91.17	88.62	80.05
search	$\mathcal{S}_{\text{ALP-NO}}$	46.40	81.99	85.78	89.71	90.64	68.95	91.86	88.81	80.52
search	$\mathcal{S}_{\text{ALP-PO}}$	46.02	82.04	84.07	89.16	90.56	68.23	91.74	88.72	80.07
none	$\mathcal{S}_{\text{ALP}}$	46.81	81.86	85.05	89.67	90.73	68.59	91.86	88.68	80.41

问题	模型	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	平均
不适用	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
不适用	$\mathcal{S}_{\text{NKD}}$	40.33	79.91	81.86	87.57	90.21	65.34	90.02	88.49	77.97
跳过，搜索	$\mathcal{S}_{\text{RKD}}$	45.51	81.41	83.82	88.21	90.56	67.51	91.51	88.70	79.65
跳过，搜索	$\mathcal{S}_{\text{PKD}}$	45.78	82.18	85.05	89.31	90.73	68.23	91.51	88.56	80.17
搜索	$\mathcal{S}_{\text{CKD-NO}}$	48.49	81.91	83.82	89.53	90.64	67.51	91.40	88.73	80.25
搜索	$\mathcal{S}_{\text{CKD-PO}}$	46.99	81.99	83.82	89.44	90.82	67.51	91.17	88.62	80.05
搜索	$\mathcal{S}_{\text{ALP-NO}}$	46.40	81.99	85.78	89.71	90.64	68.95	91.86	88.81	80.52
搜索	$\mathcal{S}_{\text{ALP-PO}}$	46.02	82.04	84.07	89.16	90.56	68.23	91.74	88.72	80.07
无	$\mathcal{S}_{\text{ALP}}$	46.81	81.86	85.05	89.67	90.73	68.59	91.86	88.68	80.41

Table 2: The teacher model $\mathcal{T}_{\text{BERT}}$ has 12 and all other student models have 6 layers.

表 2: 教师模型 $\mathcal{T}_{\text{BERT}}$ 有 12 层，所有其他学生模型都有 6 层。

Among 6-layer students, $\mathcal{S}_{\text{ALP-NO}}$ has the best average score which demonstrates that the combinatorial approach is still useful. Moreover, the supremacy of attention-based combination over the simple concatenation holds for this setting too. $\mathcal{S}_{\text{ALP}}$ is the second best and yet our favorite model as it requires no layer alignment before training.

在 6 层学生中，$\mathcal{S}_{\text{ALP-NO}}$ 具有最佳的平均得分，这表明组合方法仍然是有用的。此外，基于注意力的组合优于简单连接的优势在这个设置中也成立。$\mathcal{S}_{\text{ALP}}$ 是第二好的模型，但仍然是我们最喜欢的模型，因为它在训练前不需要层对齐。

The gap between PKD and ALP-KD is narrowed in 6- layer models compared to 4-layer students, and this might be due to an implicit relation between the size and need for combining intermediate layers. We focused on this hypothesis in another experiment and this time used the same teacher to train 2-layer students. In this scenario, student models are considerably smaller with only 39M parameters. Results of this experiment are reported in Table 3.

与 4 层学生相比，6 层模型中 PKD 和 ALP-KD 之间的差距缩小，这可能是由于中间层的大小和组合需求之间的隐含关系。我们在另一个实验中集中研究了这个假设，这次使用相同的教师来训练 2 层学生。在这种情况下，学生模型的参数显著较小，仅有 39M 个参数。该实验的结果在表 3 中报告。

For CKD and ALP-KD, we combine all teacher layers and distill into the first layer of the student. Similar to previous experiments, there is no connection between the last layer of 2-layer students and the teacher model and KD happens between $h_{\mathcal{S}}^1$ and $H_{\mathcal{T}}$. For PKD, we need to decide which teacher layers should be involved in distillation, for which we assessed three configurations with the first $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^1\right)$, sixth $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^6\right)$, and twelfth $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^{12}\right)$ layers. $\mathcal{S}'_{\text{ALP}}$ outperforms other students in this case too and this time the gap between PKD and ALP-KD is even more visible. This result points out to the fact that when teacher and student models differ significantly, intermediate layer combination becomes crucial.

对于 CKD 和 ALP-KD，我们将所有教师层结合并蒸馏到学生的第一层。与之前的实验类似，2 层学生的最后一层与教师模型之间没有连接，KD 发生在 $h_{\mathcal{S}}^1$ 和 $H_{\mathcal{T}}$ 之间。对于 PKD，我们需要决定哪些教师层应参与蒸馏，因此我们评估了三种配置，分别是第一层 $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^1\right)$ 、第六层 $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^6\right)$ 和第十二层 $\left(h_{\mathcal{S}}^1 \leftrightarrow h_{\mathcal{T}}^{12}\right)$ 。在这种情况下，$\mathcal{S}'_{\text{ALP}}$ 也优于其他学生，这次 PKD 和 ALP-KD 之间的差距更加明显。这个结果指出，当教师和学生模型差异显著时，中间层的组合变得至关重要。

Qualitative Analysis

定性分析

We tried to visualize attention weights to understand what happens during training and why ALP-KD leads to better performance. Figure 3 illustrates results related to this experiment. From the SST-2 dataset, we randomly selected 10 examples and stimulated both teacher and student models to emit attention weights between the first layer of the student $\left(h_{\mathcal{S}}^1\right)$ and all teacher layers $(H_{\mathcal{T}})$. We carried out this experiment with 2-,4-, and 6-layer $\mathcal{S}_{\text{ALP}}$ models. The x and y axes in the figure show the attention weights and 10 examples, respectively.

我们尝试可视化注意力权重，以理解训练过程中发生了什么以及为什么 ALP-KD 导致更好的性能。图 3 展示了与该实验相关的结果。我们从 SST-2 数据集中随机选择了 10 个示例，并刺激教师和学生模型在学生的第一层 $\left(h_{\mathcal{S}}^1\right)$ 和所有教师层 $(H_{\mathcal{T}})$ 之间发出注意力权重。我们对 2 层、4 层和 6 层 $\mathcal{S}_{\text{ALP}}$ 模型进行了此实验。图中的 x 和 y 轴分别显示注意力权重和 10 个示例。

As seen in Figure 3a, the first half of the teacher model is more active, which is expected since we distill into the first layer of the student. However, h_S^1 receives strong signals from other layers in the second half too, e.g. in Example-

如图 3a 所示，教师模型的前半部分更为活跃，这是可以预期的，因为我们将信息提炼到学生的第一层。然而，h_S^1 在后半部分也接收到了来自其他层的强信号，例如在示例 10 中，h_S^1 和 [latex1] 之间存在强连接。

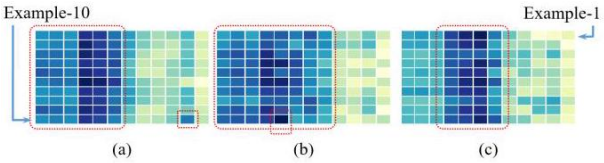

Figure 3: Visualizing attention weights between the first layer of the student model and all teacher layers for 10 samples from SST-2. Weights belong to $\mathcal{S}_{\text{ALP}}$ with 2 (a),4 (b), and 6 (c) layers.

图 3: 可视化 SST-2 中 10 个样本的学生模型第一层与所有教师层之间的注意力权重。这些权重属于 $\mathcal{S}_{\text{ALP}}$ ，具有 2(a)、4(b) 和 6(c) 层。

10 there is a strong connection between $h_{\mathcal{T}}^{11}$ and $h_{\mathcal{S}}^1$. This visualization demonstrates that all teacher layers participate in distillation and defining buckets or skipping layers might not be the best approach. A similar situation arises when distilling into the 4-layer model in Figure 3b as the first half is still more active. For the 6-layer model, we see a different pattern where there is a concentration in attention weights around the middle layers of the teacher and $h_{\mathcal{S}}^1$ is mainly fed by layers $h_{\mathcal{T}}^4$ to $h_{\mathcal{T}}^7$.

在示例 10 中，$h_{\mathcal{T}}^{11}$ 和 $h_{\mathcal{S}}^1$ 之间存在强连接。该可视化表明，所有教师层都参与了信息提炼，定义桶或跳过层可能不是最佳方法。当提炼到图 3b 中的 4 层模型时，类似的情况出现，因为前半部分仍然更为活跃。对于 6 层模型，我们看到不同的模式，其中注意力权重集中在教师的中间层，而 $h_{\mathcal{S}}^1$ 主要由层 $h_{\mathcal{T}}^4$ 到 $h_{\mathcal{T}}^7$ 提供信息。

Considering the distribution of attention weights, any skip-or even concatenation-based approach would fail to reveal the maximum capacity of KD. Such approaches assume that a single teacher layer or a subset of adjacent layers affect the student model, whereas almost all of them participate in the process. Apart from previously reported results, this visualization again justifies the need for an attention-based combination in KD.

考虑到注意力权重的分布，任何跳过或甚至基于连接的方法都无法揭示知识蒸馏的最大能力。这些方法假设单个教师层或相邻层的子集会影响学生模型，而几乎所有层都参与了这一过程。除了之前报告的结果外，这一可视化再次证明了在知识蒸馏中基于注意力的组合的必要性。

Our technique emphasizes on intermediate layers and the necessity of having similar internal representations between student and teacher models, so in addition to attention weights we also visualized the output of intermediate layers. The main idea behind this analysis is to show the information flow inside student models and how ALP-KD helps them mimic their teacher. Figures 4a and 4 b illustrate this experiment.

我们的技术强调中间层以及学生和教师模型之间具有相似内部表示的必要性，因此除了注意力权重外，我们还可视化了中间层的输出。这一分析的主要思想是展示学生模型内部的信息流，以及 ALP-KD 如何帮助它们模仿教师。图 4a 和 4 b 说明了这一实验。

We randomly selected 100 samples from the SST-2 dataset and visualized what hidden representations of $\mathcal{S}_{\text{ALP}}, \mathcal{S}_{\text{PKD}}$, and $\mathcal{T}$ models (from Table 1) look like when stimulated with these inputs. Student models have 4 layers but due to space limitations we only show middle layers' outputs, namely h_S^2 (Figure 4a) and h_S^3 (Figure 4b). h_S^1 and h_S^4 also expressed very similar attitudes.

我们从 SST-2 数据集中随机选择了 100 个样本，并可视化了 $\mathcal{S}_{\text{ALP}}, \mathcal{S}_{\text{PKD}}$ 和 $\mathcal{T}$ 模型 (见表 1) 在这些输入刺激下的隐藏表示。学生模型有 4 层，但由于空间限制，我们仅展示中间层的输出，即 h_S^2 (图 4a) 和 h_S^3 (图 4b)。h_S^1 和 h_S^4 也表现出非常相似的态度。

Problem	Model	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
N/A	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
N/A	$\mathcal{S}_{\text{NKD}}$	14.50	72.73	72.06	79.61	86.89	57.04	85.89	40.80	63.69
skip, search	$\mathcal{S}_{\text{RKD}}$	24.50	74.90	73.53	81.04	87.40	59.21	87.39	41.87	66.23
skip, search	$\mathcal{S}_{\text{PKD-1}}$	23.09	74.65	72.55	81.27	87.68	57.40	88.76	43.37	66.1
skip, search	$\mathcal{S}_{\text{PKD-6}}$	22.48	74.57	73.04	80.74	87.70	57.40	88.65	42.92	65.94
skip, search	$\mathcal{S}_{\text{PKD-12}}$	22.46	74.33	72.79	81.22	87.88	57.40	88.76	45.39	66.28
search	$\mathcal{S}_{\text{CKD}}$	24.69	74.67	73.04	81.60	87.10	58.84	88.65	43.71	66.54
none	$\mathcal{S}_{\text{ALP}}$	24.61	74.78	73.53	81.24	88.01	59.57	88.88	46.04	67.08

问题	模型	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	平均
不适用	$\mathcal{T}_{\text{BERT}}$	57.31	83.39	86.76	91.25	90.96	68.23	92.67	88.82	82.42
不适用	$\mathcal{S}_{\text{NKD}}$	14.50	72.73	72.06	79.61	86.89	57.04	85.89	40.80	63.69
跳过，搜索	$\mathcal{S}_{\text{RKD}}$	24.50	74.90	73.53	81.04	87.40	59.21	87.39	41.87	66.23
跳过，搜索	$\mathcal{S}_{\text{PKD-1}}$	23.09	74.65	72.55	81.27	87.68	57.40	88.76	43.37	66.1
跳过，搜索	$\mathcal{S}_{\text{PKD-6}}$	22.48	74.57	73.04	80.74	87.70	57.40	88.65	42.92	65.94
跳过，搜索	$\mathcal{S}_{\text{PKD-12}}$	22.46	74.33	72.79	81.22	87.88	57.40	88.76	45.39	66.28
search	$\mathcal{S}_{\text{CKD}}$	24.69	74.67	73.04	81.60	87.10	58.84	88.65	43.71	66.54
none	$\mathcal{S}_{\text{ALP}}$	24.61	74.78	73.53	81.24	88.01	59.57	88.88	46.04	67.08

Table 3: The teacher model $\mathcal{T}_{\text{BERT}}$ has 12 and all other student models have 2 layers. $\mathcal{S}_{\text{RKD}-i}$ indicates that $h^l_{\mathcal{T}}$ is used for distillation.

表 3: 教师模型 $\mathcal{T}_{\text{BERT}}$ 有 12 层，而所有其他学生模型都有 2 层。$\mathcal{S}_{\text{RKD}-i}$ 表示使用 $h^l_{\mathcal{T}}$ 进行蒸馏。

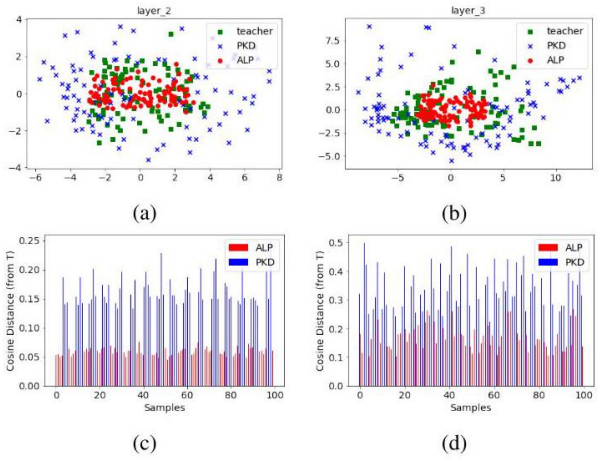

Figure 4: Visualizing intermediate layers' outputs and their distance from the teacher in ALP-KD and PKD students. Teacher-, ALP-KD-, and PKD-related information is visualized with green, red, and blue colors, respectively. Figures 4a and 4c provide information about $h^2_{\text{ALP}}, h^2_{\text{PKD}}$, and $h^5_{\mathcal{T}}$, and Figures 4 b and 4 d report information about $h^3_{\text{ALP}}, h^3_{\text{PKD}}$, and $h^9_{\mathcal{T}}$. In the bottom figures, the x axis shows samples and the y axis is the Cosine distance from the teacher.

图 4: 可视化 ALP-KD 和 PKD 学生的中间层输出及其与教师的距离。教师、ALP-KD 和 PKD 相关信息分别用绿色、红色和蓝色可视化。图 4a 和 4c 提供了关于 $h^2_{\text{ALP}}, h^2_{\text{PKD}}$ 和 $h^5_{\mathcal{T}}$ 的信息，而图 4 b 和 4 d 报告了关于 $h^3_{\text{ALP}}, h^3_{\text{PKD}}$ 和 $h^9_{\mathcal{T}}$ 的信息。在底部图中，x 轴显示样本，y 轴是与教师的余弦距离。

The output of each intermediate layer is a 768-dimensional vector, but for visualization purposes we consider the first two principal components extracted via PCA (Wold, Esbensen, and Geladi 1987). During training, $h^5_{\mathcal{T}}$ and $h^9_{\mathcal{T}}$ are connected to $h^2_{\mathcal{S}}$ and $h^3_{\mathcal{S}}$ as the source of distillation in PKD, so we also include those teacher layers' outputs in our visualization. As the figure shows, ALP-KD's representations are closer to teacher's and it demonstrates that our technique helps train better students with closer characteristics to teachers.

每个中间层的输出是一个 768 维的向量，但为了可视化目的，我们考虑通过 PCA 提取的前两个主成分 (Wold, Esbensen 和 Geladi 1987)。在训练过程中，$h^5_{\mathcal{T}}$ 和 $h^9_{\mathcal{T}}$ 连接到 $h^2_{\mathcal{S}}$ 和 $h^3_{\mathcal{S}}$ 作为 PKD 中的蒸馏源，因此我们还在可视化中包含了这些教师层的输出。正如图所示，ALP-KD 的表示更接近教师的，这表明我们的技术有助于训练出更好、特征更接近教师的学生。

We conducted another complementary analysis where we used the output of the same teacher and student layers from the previous experiment and measured their distance for all 100 examples. Results of this experiment are illustrated in Figures 4c and 4 d for the second and third student layers, respectively. Internal representations generated by PKD are more distant from those of the teacher compared to ALP-KD' s representations, e.g. the distance between $h^{20,2}_{\text{PKD}}$ (the output of the second PKD layer for the 20-th example in Figure 4c) and $h^{20,5}_{\mathcal{T}}$ is around 0.20 whereas this number is only 0.05 for ALP-KD. This is an indication that the ALP-KD student follows its teacher better than the PKD student. To measure distance, we used the Cosine similarity in this experiment.

我们进行了另一个补充分析，使用了前一个实验中相同教师和学生层的输出，并测量了它们在所有 100 个示例中的距离。该实验的结果在图 4c 和 4 d 中分别展示了第二和第三个学生层。PKD 生成的内部

表示与教师的表示相比更为遥远，例如，$h^{20,2}_{\text{PKD}}$ (图 4c 中第 20 个示例的第二个 PKD 层的输出) 与 $h^{20,5}_{\mathcal{T}}$ 的距离约为 0.20，而 ALP-KD 的这个数值仅为 0.05。这表明 ALP-KD 学生比 PKD 学生更好地跟随其教师。在本实验中，我们使用了余弦相似度来测量距离。

Conclusion and Future Work

结论与未来工作

In this paper, we discussed the importance of distilling from intermediate layers and proposed an attention-based technique to combine teacher layers without skipping them. Experimental results show that the combination idea is effective. Our findings in this research can be summarized as follows:

在本文中，我们讨论了从中间层提取的重要性，并提出了一种基于注意力的技术来组合教师层，而不跳过它们。实验结果表明，这种组合思想是有效的。我们在这项研究中的发现可以总结如下:

- It seems to distill from deep teachers with multiple internal components combination is essential.
- 从具有多个内部组件组合的深层教师中提取似乎是至关重要的。
- The more teacher and student models differ in terms of the number of layers, the more intermediate layer combination becomes crucial.
- 教师和学生模型在层数上差异越大，中间层组合就越重要。
- Although a simple concatenation of layers is still better than skipping in many cases, to obtain competitive results an attention-based combination is required.
- 尽管在许多情况下，简单的层连接仍然优于跳过，但为了获得竞争力的结果，需要基于注意力的组合。
- ALP-KD can be tuned to combine layers inside buckets and this approach is likely to yield state-of-the-art results, but if there is no enough knowledge to decide about buckets, a simple attention mask over all teacher layers should solve the problem.
- ALP-KD 可以调优以在桶内组合层，这种方法可能会产生最先进的结果，但如果没有足够的知识来决定桶的划分，简单的注意力掩码覆盖所有教师层应该能解决问题。

As our future direction, we are interested in applying ALP-KD to other tasks to distill from extremely deep teachers into compact students. Moreover, we will work on designing better attention modules. Techniques that are able to handle sparse structures could be more useful in our architecture. Finally, we like to adapt our model to combine other internal components such as attention heads.

作为我们的未来方向，我们有兴趣将 ALP-KD 应用到其他任务中，以从极深的教师模型中提炼出紧凑的学生模型。此外，我们将致力于设计更好的注意力模块。能够处理稀疏结构的技术在我们的架构中可能更为有用。最后，我们希望调整我们的模型，以结合其他内部组件，例如注意力头。

Acknowledgments

致谢

We would like to thank our anonymous reviewers as well as Chao Xing and David Alfonso Hermelo from Huawei Noah's Ark Lab for their valuable feedback.

我们要感谢我们的匿名评审以及来自华为诺亚方舟实验室的 Chao Xing 和 David Alfonso Hermelo 对我们宝贵反馈的支持。

Appendix

附录

GLUE Datasets

GLUE 数据集

Datasets used in our experiments are as follows:
我们实验中使用的数据集如下:

- CoLA: A corpus of English sentences drawn from books and journal articles with 8,551 training and 1,043 validation instances. Each example is a sequence of words with a label indicating whether it is a grammatical sentence (Warstadt, Singh, and Bowman 2019).

- CoLA: 一个来自书籍和期刊文章的英语句子语料库，包含 8,551 个训练实例和 1,043 个验证实例。每个示例都是一个单词序列，带有标签指示它是否是一个语法正确的句子 (Warstadt, Singh, and Bowman 2019)。

- MNLI: A multi-genre natural language inference corpus including sentence pairs with textual entailment annotations (Williams, Nangia, and Bowman 2018). The task defined based on this dataset is to predict whether the premise entails the hypothesis, contradicts it, or neither, given a premise sentence and a hypothesis. The dataset has two versions, matched (test and training examples are from the same domain) and mismatched, that we use the matched version. This dataset has 392,702 training and 9,815 validation examples.

- MNLI: 一个多类型的自然语言推理语料库，包括带有文本蕴含注释的句子对 (Williams, Nangia, and Bowman 2018)。基于该数据集定义的任务是预测前提是否蕴含假设、与之矛盾，或两者皆非，给定一个前提句和一个假设。该数据集有两个版本，匹配 (测试和训练示例来自同一领域) 和不匹配，我们使用匹配版本。该数据集有 392,702 个训练示例和 9,815 个验证示例。

- MRPC: A corpus of sentence pairs with human annotations. The task is to decide whether sentences are semantically equivalent (Dolan and Brockett 2005). The training and validation sets have 3,668 and 408 examples, respectively.

- MRPC: 一个带有人类注释的句子对语料库。任务是判断句子是否在语义上等价 (Dolan and Brockett 2005)。训练集和验证集分别有 3,668 和 408 个示例。

- QNLI: A dataset built for a binary classification task to assess whether a sentence contains the correct answer to a given query (Rajpurkar et al. 2016). The set has 104,743 training and 5,463 validation examples.

- QNLI: 一个用于二元分类任务的数据集，用于评估一个句子是否包含给定查询的正确答案 (Rajpurkar 等，2016)。该数据集包含 104,743 个训练示例和 5,463 个验证示例。

- QQP: A set of question pairs with 363,849 training and 40,430 validation instances collected from the well-known question answering website Quora. The task is to determine if a given pair of questions are semantically equivalent (Iyer, Dandekar, and Csernai 2017).

- QQP: 一组问题对，包含 363,849 个训练实例和 40,430 个验证实例，收集自著名的问答网站 Quora。任务是确定给定的问题对是否在语义上等价 (Iyer, Dandekar 和 Csernai 2017)。

- RTE: A combined set of 2,490 training and 277 validations examples collected from four sources for a series of textual entailment challenges (Dagan, Glickman, and Magnini 2005; Bar-Haim et al. 2006; Giampiccolo et al. 2007; Bentivogli et al. 2009).

- RTE: 一个结合了 2,490 个训练示例和 277 个验证示例的集合，收集自四个来源，用于一系列文本蕴涵挑战 (Dagan, Glickman 和 Magnini 2005；Bar-Haim 等，2006；Giampiccolo 等，2007；Bentivogli 等，2009)。

- SST-2: A sentiment analysis dataset with sentence-level (positive/negative) labels. The training and validation sets include 67,349 and 872 sentences, respectively (Socher et al. 2013).
- SST-2: 一个情感分析数据集，具有句子级别 (正面/负面) 标签。训练集和验证集分别包含 67,349 和 872 个句子 (Socher 等，2013)。
- STS-B: A collection of sentence pairs used for semantic similarity estimation. Each pair has a similarity score from 1 to 5 . This dataset has 5,749 training and 1,500 validation examples (Cer et al. 2017).
- STS-B: 用于语义相似度估计的句子对集合。每对句子都有一个从 1 到 5 的相似度评分。该数据集包含 5,749 个训练示例和 1,500 个验证示例 (Cer 等，2017)。

Hardware

硬件

Each model is fine-tuned on a single NVIDIA 32GB V100 GPU. The fine-tuning time, based on the dataset size, can vary from a few hours to one day on a single GPU.

每个模型在单个 NVIDIA 32GB V100 GPU 上进行微调。微调时间根据数据集大小的不同，可能从几个小时到一天不等，具体取决于单个 GPU。

References

参考文献

Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; and Guo, C. 2020. Knowledge Distillation from Internal Representations. In AAAI, 7350-7357.

Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; 和 Guo, C. 2020. 从内部表示中进行知识蒸馏。在 AAAI，7350-7357。

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In ICLR.

Bar-Haim, R.; Dagan, I.; Dolan, B.; Ferro, L.; Giampiccolo, D.; Magnini, B.; and Szpektor, I. 2006. The second pascal recognising textual entailment challenge. In Proceedings of the second PASCAL challenges workshop on recognising textual entailment, volume 6, 6-4. Venice.

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In TAC.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 535-541.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 1-14.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL recognising textual entailment challenge. In Machine Learning Challenges Workshop, 177-190. Springer.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.

Dolan, W. B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005).

Freitag, M.; Al-Onaizan, Y.; and Sankaran, B. 2017. Ensemble distillation for neural machine translation. arXiv preprint arXiv:1702.01802 .

Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; and Anand-kumar, A. 2018. Born again neural networks. In International Conference on Machine Learning, 1607-1616. PMLR.

Giampiccolo, D.; Magnini, B.; Dagan, I.; and Dolan, B. 2007. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, 1-9. Association for Computational Linguistics.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 .

Iyer, S.; Dandekar, N.; and Csernai, K. 2017. First Quora Dataset Release: Question Pairs. https://data.quora.com/Fi
Quora-Dataset-Release-Question-Pairs .

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 4163-4174.

Kim, Y.; and Rush, A. M. 2016. Sequence-Level Knowl-

Kim, Y.; 和 Rush, A. M. 2016. 序列级知识-

edge Distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1317- 1327.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019. Improving multitask deep neural networks via knowledge distillation for natural language understanding. arXiv preprint arXiv:1904.09482

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextual-ized Word Representations. In NAACL-HLT, 2227-2237. Association for Computational Linguistics.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In EMNLP, 2383-2392. The Association for Computational Linguistics.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. Distil-BERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 .

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A .; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, 1631-1642.

Sun, S.; Cheng, Y.; Gan, Z.; and Liu, J. 2019. Patient Knowledge Distillation for BERT Model Compression. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 4314-4323.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2158-2170.

Tan, X.; Ren, Y.; He, D.; Qin, T.; and Liu, T.-Y. 2019. Multilingual Neural Machine Translation with Knowledge Distillation. In International Conference on Learning Representations. https://openreview.net/forum?id=S1gU

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In Advances in neural information processing systems, 5998-6008.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In BlackboxNLP@EMNLP, 353-355. Association for Computational Linguistics.

Warstadt, A.; Singh, A.; and Bowman, S. R. 2019. Neural network acceptability judgments. Transactions of the Association for Computational Linguistics 7: 625-641.

Wei, H.-R.; Huang, S.; Wang, R.; Dai, X.; and Chen, J. 2019. Online Distilling from Checkpoints for Neural Machine Translation. In Proceedings of the 2019 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 1932-1941.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 1112-1122.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. Chemometrics and intelligent laboratory systems 2(1-3): 37-52.

Wu, Y.; Passban, P.; Rezagholizadeh, M.; and Liu, Q. 2020. Why Skip If You Can Combine: A Simple Knowledge Distillation Technique for Intermediate Layers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).