通过知识选择改进预训练语言模型的知识蒸馏

Chenglong Wang[1], Yi Lu[1]*, Yongyu Mu[1], Yimin Hu[1], Tong Xiao[1,2]†和 Jingbo Zhu[1,2]东北大学计算机科学与工程学院自然语言处理实验室，中国沈阳[2]
NiuTrans Research，中国沈阳 {clwang1119,yilu001102}@gmail.com
{xiaotong,zhujingbo}@mail.neu.edu.cn

arXiv:2302.00444v1 [cs.CL] 1 Feb 2023

摘要

知识蒸馏旨在解决将知识从教师模型迁移到学生模型的问题。在这一过程中，我们通常会从教师模型中提取多种类型的知识。问题在于如何充分利用这些知识来训练学生模型。我们的初步研究表明：（1）并非所有知识都是训练出优质学生模型所必需的；（2）知识蒸馏可以在不同训练步骤中受益于特定知识。为此，我们提出一种演员-评论家方法，在知识蒸馏过程中选择适当的知识进行迁移。此外，我们还改进了训练算法，以减轻计算负担。在 GLUE数据集上的实验结果表明，我们的方法显著优于多个强有力的知识蒸馏基线方法。

1 引言

预训练语言模型（PLM）显著提升了多种自然语言处理任务的最新水平，例如情感分析（Bataa 和 Wu，2019年；Baert 等人，2020年）、文本分类（Sun 等人，2019a；Arslan 等人，2021年）以及问答（Yang 等人，2019年）。尽管取得了显著成果，预训练语言模型的参数数量庞大，因此部署成本高昂（Yang 等人，2019年）。

为解决这一问题，近期研究采用知识蒸馏（Hinton 等人，2015）来压缩并加速预训练语言模型（Sun 等人，2019b；Sanh 等人，2019；Li 等人，2021b；Liang 等人，2021）。其核心思想是将大型预训练语言模型（即，教师模型）中的知识迁移到轻量级模型（即，学生模型）中，同时避免显著的性能损失。在此过程中，我们通常会从教师模型中提取多种类型的知识，例如响应知识、特征知识和关系知识（Gou 等人，2021）。近期研究主要关注如何让学生模型更高效地学习迁移而来的知识，例如设计训练方案（Sun 等人，2019b；Mirzadeh 等人，2020；Jafari 等人，2021；Li 等人，2021a）或扩充任务专用数据（Jiao 等人，2020；Liang 等人，2021）。然而，很少有研究考虑如何在学生模型的训练中充分利用多种类型的知识。

我们的初步研究（见第 4.1 节）表明，在利用多种知识进行蒸馏时，并非所有知识都是训练出优秀学生模型所必需的。此外，受动态知识蒸馏的启发（Li 等人，2021b），我们假设在不同训练步骤中从特定知识中学习有利于知识蒸馏。我们在第4.2 节进行探测实验，以验证这一假设。具体而言，我们重复进行知识蒸馏 200 次，并在每个训练步骤随机选择知识。我们发现，不同蒸馏学生模型的性能存在明显差异。例如，在 RTE 数据集上，最佳学生模型的准确率比最差模型高出近 10%。此外，它的表现明显优于从固定知识中学习的学生模型。基于初步研究，我们提出以下建议：如果蒸馏学生模型在每个训练步骤都能学习适当的知识，就能取得更优异的性能。这一建议促使我们研究如何在知识蒸馏过程中为学生模型选择适当的知识。

基于上述发现，我们提出了一种演员-评论家方法，用于在不同训练步骤选择要传递的适当知识（Bhatnagar 等人，2009年）。该方法首先通过长期奖励优化，利用演员-评论家算法实现知识选择模块。这种优化可以考虑知识选择对未来训练步骤的影响

*本研究是在东北大学 NLPLab 实习期间完成的。

†通讯作者。

。此外，我们提出了一种多阶段训练方法，将蒸馏过程划分为多个阶段，并在阶段结束时提供特定奖励。与通常的演员-评论家算法相比，该方法可以减轻计算奖励的负担。之后，我们利用训练好的知识选择模块在每个训练步骤选择知识，从而执行知识蒸馏。

我们在六个 GLUE数据集（Wang 等人，2019年）上测试了所提出的方法，这些数据集涵盖问答、情感分析和文本蕴含。实验结果表明，我们的方法显著优于原始知识蒸馏方法（Hinton 等人，2015）和其他具有竞争力的知识蒸馏方法。值得注意的是，结果还表明，我们的 BERT_6 （ 6 层 BERT ）学生模型的性能可达到教师 $\text{BERT}_{\text{BASE}}$的 98.5% 以上，同时参数量大幅减少（~61%）。此外，我们证明，结合数据增强后，我们的方法能够取得进一步提升，并优于 TinyBERT （ Jiao 等人，2020 年），从而在采用数据增强的情况下建立了一个强有力的基线。

2 相关工作

知识蒸馏（KD）（Hinton 等人，2015）被广泛用于压缩和加速预训练语言模型（ PLMs）（Jiao 等人，2020；Sun 等人，2020；Wang 等人，2020；Li 等人，2021b）。其核心思想是将大型预训练语言模型（即教师模型）中的知识迁移到轻量级模型（即学生模型）中。近期关于知识蒸馏的研究可分为三类。第一类研究侧重于利用教师模型中的各种知识来蒸馏学生模型。例如，利用多个教师模型为学生模型提供更加多样且准确的知识（Liu 等人，2019a；Yuan 等人，2021）。Sun 等人（2019b）在蒸馏过程中利用了教师模型中间层的知识。此外，权重蒸馏（Lin 等人，2021）将教师模型参数中的知识迁移到学生模型中。第二类研究倾向于设计有效的策略，促进学生模型从不同类型的知识中学习，例如丰富任务专用数据（Jiao 等人，2020；Liang 等，2021）以及采用两阶段学习框架（Turc 等人，2019；Jiao 等人，2020）。第三类研究受到的关注较少，通常探索知识蒸馏中适合学生模型的数据和教师模型。例如， Li 等人 （2021b）提出了一种数据选择策略，根据学生模型的能力，仅选择对学生模型至关重要的部分数据。 Yuan 等人 （2021）尝试在蒸馏过程中调整多个教师模型的权重。

与这些方法不同，本文关注的是在知识蒸馏过程中如何为学生模型选择合适的知识。为此，我们通过演员-评论家算法训练一个有效的知识选择模型，并利用它在每个训练步骤中选择合适的知识进行迁移。

3 背景

3.1 知识蒸馏

在知识蒸馏中，学生模型通过模仿教师模型相应的行为来学习知识。在蒸馏过程中，这种模仿可以通过最小化以下损失函数来实现：

$$\mathcal{L}_{KD} = \sum_{x \in \mathcal{X}} \mathcal{L}_{diff}(f^S(x), f^T(x)) \tag{1}$$

其中， $\mathcal{X}$ 是训练数据集， x 是一个输入样本， $f^S(\cdot)$和 $f^T(\cdot)$分别是描述学生模型和教师模型行为的函数。 $\mathcal{L}_{diff}(\cdot)$是一个用于评估二者行为差异的损失函数。在知识蒸馏过程中，由于学生模型和教师模型都存在多种行为（例如特征提取），因此通常会涉及多种类型的知识。以往的工作主要着眼于设计精巧的 $f^S(\cdot)$、 $f^T(\cdot)$和 $\mathcal{L}_{diff}(\cdot)$，以鼓励学生模型更好地学习某些固定的知识（Sun 等，2019b；Jiao 等，2020；Liang 等，2021）。与这些工作不同，本文充分利用多种类型的知识，并在知识蒸馏过程中为学生模型选择合适的知识。

3.2 知识类型

学生模型可以在知识蒸馏中学习从教师模型提取的各种知识。根据Gou 等人（2021年），知识类型通常包括响应知识（*ResK*）、特征知识（*FeaK*）和关系知识（*RelK*）。此外，在压缩预训练语言模型时，我们还考虑一种从训练数据集中获得的微调知识（*FinK*）。这些知识类型的概述如下：

知识	RTE		SST-2	
	$BERT_6$	$BERT_3$	$BERT_6$	$BERT_3$
FinK	65.8	58.1	89.7	88.0
ResK	**67.1**	56.8	90.2	87.5
FeaK	66.1	57.8	90.4	87.1
RelK	64.6	**58.9**	88.9	88.2
FinK&ResK	65.2	55.6	90.0	**88.3**
FinK&ResK&FeaK	64.6	56.7	90.0	87.0
FinK&ResK&FeaK&RelK	64.5	57.4	**90.7**	87.7

表1：蒸馏学生模型在 RTE 和 SST-2 开发集上的准确率（%）。教师模型为 $BERT_{BASE}$。学生模型分别为 6 层 $BERT_6$ 和 3 层 $BERT_3$，。请注意，在知识蒸馏过程中，知识是固定的。

• **ResK**：ResK 指学生模型通过模仿教师模型最后一层的神经响应所学到的知识。它包含更多关于预测结果的信息（Hinton 等人，2015年）。

• **FeaK**：FeaK 表示学生模型从教师模型中间层的输出中学习到的特征表示。与 ResK 不同，它包含更多关于教师模型计算过程的信息。

• **RelK**：RelK 表示不同层或样本之间的关系（Yim等人，2017年）。它更关注模型层之间的关系。

• **FinK**：FinK 是学生模型在训练数据集上微调时从真实标签中学到的知识（Hinton 等人，2015年；Devlin等人，2019年）。在此，我们也将其视为知识蒸馏中的一种知识类型。

在这项工作中，我们探讨如何从这些类型中选择适当的知识，以便在知识蒸馏过程中进行迁移。为确保将上述类型的知识迁移到学生模型中，我们分别设计了损失函数 $\mathcal{L}_{ResK}$， $\mathcal{L}_{FeaK}$，$\mathcal{L}_{RelK}$，以及 $\mathcal{L}_{FinK}$。附录A 介绍了损失函数的具体设计细节。

4 初步研究

由于每种知识都具有独特的信息和特性，我们假设不同类型的知识对知识蒸馏结果的影响各不相同。基于此，我们开展了初步研究，以探究知识与知识蒸馏之间的关系。

Task	方法	S_{best}	S_{worst}	Δ
RTE	全程随机	68.4	58.3	10.1
	Random-One	67.2	58.9	8.7
	全程固定	67.1	64.5	2.6
SST-2	全程随机	91.3	86.1	5.2
	Random-One	90.7	87.9	2.8
	全程固定	90.7	88.9	1.8

表2：准确率（%） S_{best} 和 S_{worst} 在开发集上的表现。全程固定表示在每个训练步骤中，我们都使用相同的知识来训练学生模型。其结果取自表1。

4.1 知识蒸馏的不同知识

我们使用不同的知识分别对学生模型进行蒸馏实验。在实验中，我们在开发集上调整每种知识的权重，这与调整超参数类似。表 1对这些蒸馏后的学生模型进行了比较。我们可以看到，使用不同知识时，学生模型的准确率存在明显差距。这表明并非所有知识都是训练出优秀学生模型所必需的。有趣的是，我们还发现，适当的知识会随着学生模型或数据集的变化而变化。我们推测，学生模型容量和样本复杂度等因素可能会影响知识迁移，这与Stanton 等人(2021年）的发现一致。

4.2 不同训练步骤中的知识

受动态知识蒸馏（Li 等人，2021年b）的启发，我们假设，由于学生模型和样本在训练过程中不断变化，适当的知识也可能动态变化。我们开展探测试验以验证这一假设。在每个训练步骤中，我们为学生模型随机选择知识。具体而言，我们分别在所有训练轮次的训练步骤（全程随机）和第一个训练轮次的训练步骤（*Random-One*）中采用这一随机策略。为使实验更具普适性，全程随机和 Random-One 均重复 200 次。

我们在表 S_{best} 和S_{worst} 中报告了蒸馏学生模型的最佳和最差表现，分别记为 2。从全程随机的结果可以看出， S_{best} 与 S_{worst}的准确率存在明显差距。值得注意的是，S_{best} 在 RTE 数据集上的表现比 S_{worst} 高出 10%。我们推测， S_{best}表示学生模型在大多数训练步骤中都使用了恰当的知识进行训练，而 S_{worst} 则相反。此外，我们注意到 S_{best} 的准确率达到 68.4%，显著

超过了所有使用固定知识的学生模型。这表明知识蒸馏能够从不同训练步骤中的特定知识中获益。这也促使我们研究如何选择恰当知识的方法。

对于 Random-One 的结果，S_{best}和S_{worst}之间也存在显著的性能差异，尽管只有知识蒸馏第一个训练轮次中的训练步骤涉及知识的随机选择。这表明，当前训练步骤中的知识选择会影响后续训练步骤。因此，在选择合适的知识时，我们应考虑其未来影响。

5 方法

5.1 概述

在这项工作中，我们的目标是在知识蒸馏过程中选择合适的知识进行迁移。鉴于当前的选择可能会影响未来的训练步骤，我们提出了演员-评论家方法，通过长期奖励优化来解决知识选择问题。图 1 展示了我们方法的概述。我们的方法包含两个阶段：训练知识选择模块（KSM），以及使用训练好的 KSM 对学生模型进行蒸馏。在第一阶段，我们使用由演员和评论家组成的演员-评论家算法来实现 KSM。演员根据给定的状态输出一个动作，以选择知识。评论家网络预测动作价值，即未来奖励之和。在第二阶段，我们使用训练好的 KSM 选择合适的知识，并将其迁移到学生模型中。

5.2 定义

在这项工作中，我们使用演员-评论家算法来实现 KSM。这里，我们首先介绍演员-评论家算法的一些关键概念，包括状态、动作和奖励。

状态。在训练步骤 t，我们用 s_t 表示相应的状态。这里，s_t 应包含足够的信息，以便选择合适的知识。为此，我们使用学生和教师模型最后一层中信息丰富的 [CLS]嵌入（Devlin等人，2019年）来实现这一点。具体而言，对于训练步骤 i中的第t个样本，我们用 $\boldsymbol{c}_i^S$和 $\boldsymbol{c}_i^T$ 分别表示学生和教师模型最后一层的 [CLS] 嵌入。随后，我们利用两个可训练的特征网络 $\mathcal{N}_{fea}^S$和 $\mathcal{N}_{fea}^T$来提取有用的特征向量 $\boldsymbol{v}$:

$$\boldsymbol{v}_i^S = \mathcal{N}_{fea}^S(\boldsymbol{c}_i^S), \quad \boldsymbol{v}_i^T = \mathcal{N}_{fea}^T(\boldsymbol{c}_i^T) \tag{2}$$

其中，$\mathcal{N}_{fea}^S$和 $\mathcal{N}_{fea}^T$均由一个两层多层感知机（MLP）网络构成。我们将给定批次中提取的所有特征向量拼接为 s_t。

动作。给定状态 s_t，可以生成一个动作 a_t，用于在训练步骤 t选择合适的知识进行迁移。这里，我们提出软动作和硬动作，以灵活地选择合适的知识。软动作决定从每种知识类型中学习多少，而硬动作则选择一种或多种知识类型来训练学生模型。

奖励。这里，我们将即时奖励r_t 定义为学生模型使用选定知识训练后，在开发集上的交叉熵损失差。不同奖励的比较见附录中的表 6 。

5.3 知识选择模块

为了提升蒸馏模型的性能，我们需要在不同的训练步骤中选择适当的知识进行迁移。为此，我们通过演员-评论家算法实现了一个知识选择模块，该算法由演员和评论家组成。在这里，我们利用长期奖励对其进行优化，以考虑知识选择对未来训练步骤的影响。

5.3.1 演员和评论家

演员。在这项工作中，演员网络 μ_θ 由一个包含四个输出神经元的三层多层感知机组成。它接收状态 s_t，并通过 Sigmoid 函数输出软动作:

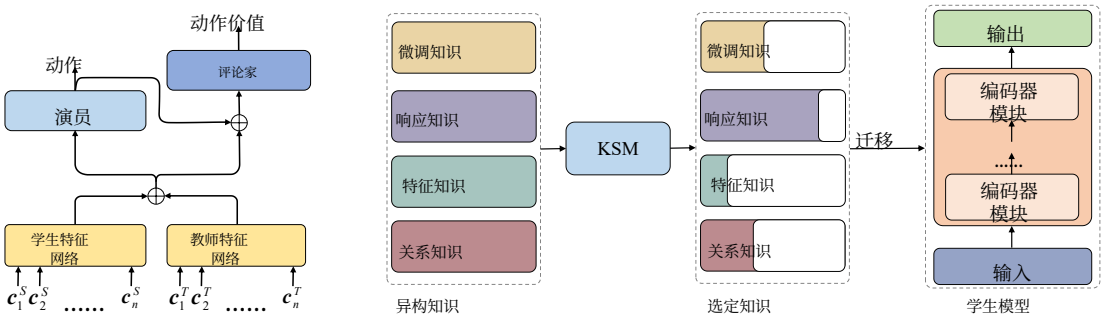
$$a_t = \text{Sigmoid}(\mu_\theta(s_t)) \tag{3}$$

其中，$a_t = \{a_1^t, a_2^t, a_3^t, a_4^t\}$ 包含四个属于 $[0, 1]$ 的实数。我们用 a_m^t 表示 a_t 中选定的第 m 类知识*的百分比。在训练步骤 t，学生模型的损失按如下方式计算:

$$\mathcal{L}_t^{soft} = a_1^t\mathcal{L}_{FinK} + a_2^t\mathcal{L}_{ResK} + a_3^t\mathcal{L}_{FeaK} + a_4^t\mathcal{L}_{RelK} \tag{4}$$

此外，为了直接选择一种或多种类型的知识，我们通过一个条件函数获得硬动作

*在这里，我们将*FinK*、*ResK*、*FeaK*和*RelK* 分别视为第 1 类、第 2 类、第 3 类和第 4 类知识。



（a）KSM 的架构　　(b) 通过 K 选择适用于知识蒸馏的知识　　SM

图1：所提出方法的概览。我们采用演员-评论家方法设计知识选择模块（KSM），旨在选择知识蒸馏过程中要迁移的适当知识。除了图1(b)所示的软动作外，我们还为演员模块设计了硬动作（参见第5.3.1节）。

$g(\cdot)$:

$$g(a_m^t) = \begin{cases} 1 & a_m^t \geq \lambda \\ 0 & otherwise \end{cases} \tag{5}$$

其中 $\lambda \in [0,1]$ 是一个阈值。采用硬动作时，学生模型的损失如下:

$$\begin{aligned} \mathcal{L}_t^{hard} = g(a_1^t)\mathcal{L}_{FinK} + g(a_2^t)\mathcal{L}_{ResK} + \\ g(a_3^t)\mathcal{L}_{FeaK} + g(a_4^t)\mathcal{L}_{RelK} \end{aligned} \tag{6}$$

此处，如果 $a_m^t \geq \lambda$，则第 m 类知识适用于学生模型；否则，该知识不适用。

评论家。评论家网络 Q_ϕ 同样由一个三层多层感知机组成。它计算给定状态-动作对 (s_t, a_t) 的动作价值 $Q_\phi(s_t, a_t)$，其中 s_t 和 a_t 拼接后作为输入。动作价值用于估计演员在状态 s_t 下采取动作 a_t 后获得的奖励总和。此处，我们将动作价值定义为长期奖励，即学生模型从训练步骤 t 到训练结束所获得的性能提升。

5.3.2 优化

演员优化。遵循深度确定性策略梯度算法（Silver 等人，2014年），我们通过更新演员参数，利用采样策略梯度来优化长期奖励:

$$\nabla_\theta J \approx \frac{1}{N}\sum_{t=1}^{N} \nabla_\theta Q_\phi(s_t, a_t) \tag{7}$$

评论家优化。在每个训练步骤中，我们只能获得即时奖励 r_t。然而，评论家提供的是长期奖励，而非即时奖励。因此，我们采用时序差分（Tesauro，1991年）来近似用 r_t估计实际的长期奖励:

$$Q_\phi^*(s_t, a_t) = \gamma^t r_t + Q_\phi(s_{t+1}, a_{t+1}) \tag{8}$$

其中 γ 是折扣因子（Sutton 和 Barto，2018年）。随后，我们可以通过近似长期奖励给出评论家网络的优化目标:

$$\mathcal{L}_Q = \frac{1}{N}\sum_{t=1}^{N} MSE(Q_\phi(s_t, a_t), Q_\phi^*(s_t, a_t)) \tag{9}$$

其中，$MSE(\cdot)$ 是均方误差损失函数，N 是知识蒸馏中的训练步数。

特征网络优化。在训练 KSM 期间，我们分别使用演员网络更新学生特征网络，并使用评论家网络更新教师特征网络，以降低 KSM 训练的不稳定性。

5.3.3 多阶段训练

由于某些数据集的训练步骤数量较多，在训练 KSM 时通过开发集上的损失来计算奖励会产生相当大的开销。例如，在 QQP 数据集上训练 5 个训练轮次，需要在一个回合中计算奖励约 50,000 次。为减轻计算负担，我们提出了一种多阶段训练方法来训练 KSM。具体而言，我们将完整的知识蒸馏过程划分为多个阶段，每个阶段包含 k 个训练步骤。学生模型在一个阶段完成训练后，我们将开发集上的交叉熵损失差作为阶段奖励 r^p。它表示一个阶段内所有奖励的总和。在这里，我们可以使用阶段优化目标 $\mathcal{L}_Q^p$ 来训练评论家网络，而不是使用 $\mathcal{L}_Q$:

$$\mathcal{L}_Q^p = \frac{1}{N_p}\sum_{j=1}^{N_p} MSE(r_j^p, \sum_{t=b(j)}^{e(j)} \hat{r}_t) \tag{10}$$

算法1我们的方法

输入： 训练良好的教师模型 $\mathcal{M}_T$；初始学-
生模型 $\mathcal{M}_S$；训练数据集 $\mathcal{X}$
输出： 蒸馏后的 $\mathcal{M}_S$；
1：%%% 第一阶段
2：通过算法训练 KSM 2；
3：
4：%%% 第二阶段
5：**对** $t = 1$ 至 N **执行**
6：通过 s_t 式 2计算状态；
7：使用动作 $a_t = \mu_\theta(s_t)$ 来选择合适的
知识；
8：通过 $\mathcal{M}_S$ 式 4或式 6进行训练；
9：**结束循环**
10：返回 $\mathcal{M}_S$

算法2KSM 的训练

输入：训练完成的教师模型 $\mathcal{M}_T$；初始学生模型 $\mathcal{M}_S$；训练数据集 $\mathcal{X}$；初始 KSM（一个演员 μ_θ、一个评论家 Q_ϕ、两个特征网络 $\mathcal{N}^S_{fea}$ 和 $\mathcal{N}^T_{fea}$） 输出：训练完成的 KSM 1：对于 $episode = 1$ 到 K，执行 2：重置 $\mathcal{M}_S$；3：将整个知识蒸馏过程划分为 N_p 个阶段；4：对于 $j = 1$ 到 N_p，执行 5：对于 $t = b\,(j)$ 到 $e\,(j)$，执行 6：通过公式 2 计算状态 s_t；7：采样一个动作 $a_t = \mu_\theta\,(s_t)$；8：利用先前阶段中的动作，通过公式 12 或公式 13 计算 r^e_t；9：计算动作价值 $Q_\phi\,(s,\ a)$；10：通过公式4或公式6训练 $\mathcal{M}_S$；11：结束循环 12：通过损失差异计算 r^p_j；13：通过公式 9 和公式 10 训练 KSM；14：结束循环 15：结束循环 16：返回 KSM

其中 N_p 是知识蒸馏中的阶段数， $b(j)$ 和$e(j)$ 分别是第j阶段的起始和结束训练步骤。 $\hat{}r_t$ 是根据公式计算的估计奖励8：

$$\hat{r}_t = \frac{Q_\phi(a_t, s_t) - Q_\phi(a_{t+1}, s_{t+1})}{\gamma^t} \tag{11}$$

通过划分阶段，我们可以有效减轻计算负担。通常，演员-评论家算法的常规训练需要计算奖励 N 次，而我们的多阶段训练只需计算奖励$\lceil N/k \rceil$ 次。

5.3.4 探索奖励

为了加快 KSM 的训练，我们还设计了一种探索奖励 r^e_t，在训练步骤 t中鼓励演员采取更多不同的动作（Tang 等人，2017年）。对于软动作，我们考虑动作之间的相似度为 r^e_t:

$$r^e_t = \alpha \times (1 - \frac{1}{k}\sum_{l=1}^{k} Sim(a_t, a_l)) \tag{12}$$

其中， $Sim(\cdot)$ 是余弦相似度函数， a_l 是上一阶段的第 l 个动作。此外，我们根据硬动作的重复次数计算其探索奖励 r^e_t：

$$r^e_t = \alpha \times (1 - \frac{count(a_t)}{k}) \tag{13}$$

其中 $count(a_t)$ 是该动作a_t 在上一阶段被采取的次数，而 α 是一个缩放因子。我们通过 r^e_t（根据公式 9 计算）为评论家网络提供一个额外的优化目标 $\mathcal{L}^e_Q$。

5.4 知识蒸馏

算法 1 展示了我们提出的整体方法。第一阶段，我们使用演员-评论家算法训练 KSM。如算法 2 所述，我们首先将完整的知识蒸馏过程划分为一个回合中的 N_p 个阶段（第 3 行）。在一个阶段的每个训练步骤中，我们计算状态、动作和探索奖励（第 6-9 行），并使用选定知识训练学生模型（第 10 行）。随后，在阶段结束时计算阶段奖励。接着，我们分别通过公式 $\mathcal{L}^e_Q$ 和 $\mathcal{L}^p_Q$ 所计算的损失目标来优化 KSM，公式为 9 和公式 10（第 13 行）。这一过程在多个回合中迭代，直至学生模型的性能收敛。第二阶段，我们采用训练好的 KSM 选择合适的知识，用于蒸馏学生模型。

6 实验

6.1 数据集与设置

数据集。我们遵循Sun 等人（2019b），在六个GLUE数据集上进行实验（Wang 等人，2019）：MNLI（Williams 等人，2018年）、QQP（Chen 等人，2018年）、QNLI（Rajpurkar 等人，2016年）、SST-2（Socher 等人，2013年）、MRPC（Dolan 和 Brockett，2005）和 RTE（Bentivogli 等人，2009年）。

设置。我们使用 $\text{BERT}_{\text{BASE}}$ 作为教师模型，使用 BERT_6 和 BERT_3 作为学生模型。训练设置的详细信息见附录B。所有实验均重复三次，并报告使用不同随机种子进行三次运行的平均结果。

6.2 基线方法

我们将所提出的方法与**微调**(Devlin等人，2019年）、**原始知识蒸馏**(Hinton 等人，

方法	学生	#参数	RTE (2.5k)	MRPC (3.7k)	多体裁自然语言推理（匹配／不匹配） (393k)	SST-2 (67k)	QNLI (105k)	QQP (364k)	Avg.
$BERT_{BASE}$（教师）	-	109.0M	69.3	87.6/83.5	84.1/83.1	94.3	90.5	71.0/89.2	84.9
PKD	$BERT_6$	67.0M	65.5	85.0/*79.9*	81.5/81.0	92.0	89.0	*70.7/88.9*	82.5
动态知识蒸馏	$BERT_6$	67.0M	-	86.5/-	81.8/81.0	-	-	-	-
DistilBERT	$BERT_6$	67.0M	58.4	86.9/-	*82.6*/81.3	*92.5*	88.9	70.1/-	-
微调	$BERT_6$	67.0M	65.2	85.1/79.2	81.1/79.8	91.7	87.1	69.4/88.2	81.8
原始知识蒸馏	$BERT_6$	67.0M	65.1	85.5/79.8	82.4/81.6	91.4	86.9	70.0/88.4	82.2
随机困难样本	$BERT_6$	67.0M	65.1	85.2/79.0	82.1/81.3	91.2	86.7	69.8/88.3	82.0
随机软动作	$BERT_6$	67.0M	64.8	85.0/79.2	82.2/81.3	91.5	86.5	69.7/88.1	81.9
我们的方法（硬动作）	$BERT_6$	67.0M	*66.6*	*87.7*/**82.2**	*82.6/81.8*	92.1	*89.0*	70.5/*88.9*	*83.3*
我们的方法（软动作）	$BERT_6$	67.0M	**66.8**	**87.9/82.2**	**83.1/82.1**	**92.6**	**89.3**	**71.1/89.1**	**83.6**
PKD	$BERT_3$	45.7M	58.2	80.7/*72.5*	76.7/76.3	87.5	*84.7*	*68.1*/87.8	77.7
随机困难样本	$BERT_3$	45.7M	54.3	80.7/71.8	76.8/76.1	87.4	82.9	66.4/86.4	76.5
随机软动作	$BERT_3$	45.7M	55.7	78.9/69.7	77.1/76.3	87.7	82.3	66.4/86.5	76.5
我们的方法（硬动作）	$BERT_3$	45.7M	*58.4*	*81.4*/71.5	*77.5/76.8*	**88.4**	84.6	*68.1/88.0*	*77.9*
我们的方法（软动作）	$BERT_3$	45.7M	**58.7**	**81.9/72.7**	**78.0/77.3**	*88.3*	**85.1**	**68.7/88.2**	**78.3**

表 3：GLUE 测试服务器上的结果。每组学生模型中排名第一和第二的结果分别以**粗体**和斜体显示。每个数据集下方的数字表示对应训练数据集的数量。对于 MRPC 和 QQP，我们报告 F1值/准确率。我们还在“平均值”列中报告每个数据集的平均准确率。动态知识蒸馏的结果通过不确定性-熵策略获得（Li 等人，2021b）。DistilBERT 和 PKD 的结果分别取自Jiao 等人（2020年）和Sun 等人（2019b）。

2015），以及其他具有竞争力的预训练模型知识蒸馏方法，例如**患者知识蒸馏（PKD）**(Sun等人，2019b)、**DistilBERT** (Sanh 等人，2019年)，以及**动态知识蒸馏**(Li 等人，2021b)。

此外，我们还设计了**随机软动作**和**随机困难样本** 基线方法，以评估我们方法的有效性。随机软动作和随机困难样本分别表示 KSM 随机采取软动作和硬动作。

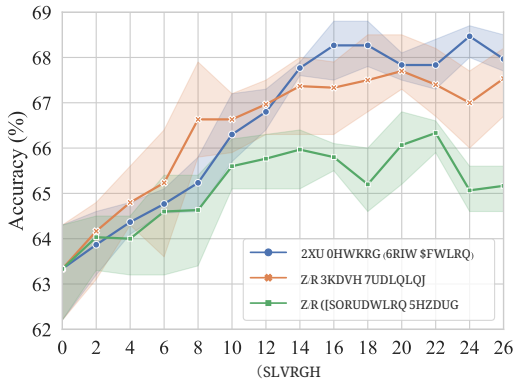


图2：消融研究。我们绘制了使用三个不同随机种子蒸馏得到的学生模型在 RTE开发集上的平均准确率。

6.3 主要结果

我们将模型预测结果提交至 GLUE 官方评测服务器，结果汇总于表3。首先，与所有基线方法相比，我们的方法能够取得最佳结果，同时还

在所有数据集上对学生模型 $BERT_6$ 和 $BERT_3$ 进行蒸馏。其次，与随机硬动作（或随机软动作）相比，我们可以发现，我们的方法在 F1 值和准确率方面取得了显著提升。这表明 KSM系统优于随机选择知识的基线方法。第三，通过我们的方法使用软动作进行蒸馏的 $BERT_6$学生模型，显著优于广受欢迎的原始知识蒸馏，至少领先 1.4%。我们认为，这是因为知识蒸馏能够从不同训练步骤中选出的知识中受益。第四，我们的方法得到的 $BERT_6$ 学生模型参数量为 ~61%，其性能与教师模型 $BERT_{BASE}$相近。例如，在 QQP 数据集上，使用我们的方法通过软动作训练的 $BERT_6$ 学生模型具有

准确率为89.1%，仅与教师模型相差0.1%。此外，与教师模型 $BERT_{BASE}$相比，$BERT_3$ 学生模型的参数仅为其41%，但性能仍达到92%。

此外，我们进一步考察了两种不同动作带来的性能提升。从结果中可以看出，软动作的表现优于硬动作，尽管两者都能使我们的方法在大多数数据集上优于基线方法。一个可能的解释是，软动作空间比硬动作空间更大。因此，相较于硬动作，软动作更有可能探索出更好的知识选择。

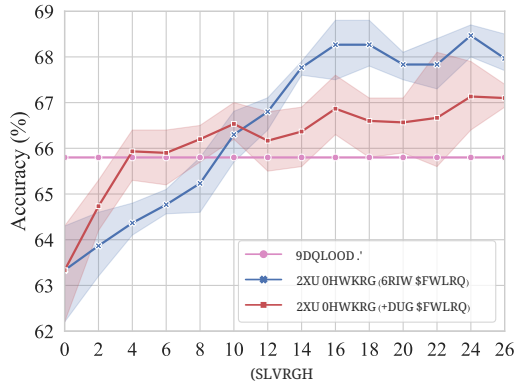


图 3：软动作与硬动作的比较。我们的方法使用硬动作可以实现更快的 KSM 学习，使用软动作则能取得更好的结果。

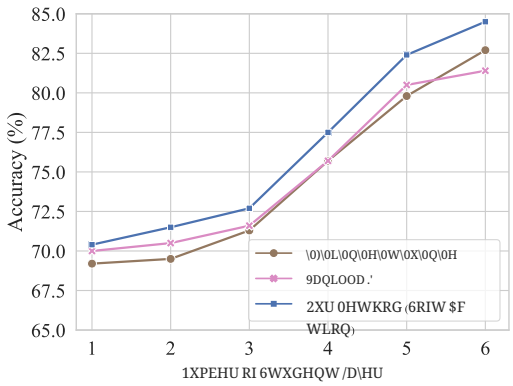


图4：在 MRPC 开发集上，不同层数的蒸馏学生模型的性能。

方法	RTE (52k)	MRPC (80k)	SST-2 (1,084k)	QNLI (2,237k)
微调	64.8	86.6/81.3	81.8	84.0
原始知识蒸馏	67.3	*88.0/82.8*	91.2	87.0
TinyBERT	*70.0*	87.3/82.6	*93.1*	*90.4*
我们的方法	**70.4**	**88.9/83.9**	**93.8**	**90.8**

表 4：我们的方法在 GLUE 测试服务器增强数据集上的结果。Tiny- BERT 的结果取自 Jiao 等人(2020年)。

6.4 消融研究

我们通过消融研究考察所提出的多阶段训练和探索奖励对准确率与效率的影响。图2展示了在移除多阶段训练或探索奖励后，使用 KSM 在 RTE开发集上蒸馏得到的学生模型的准确率。我们可以看到，采用多阶段训练更有可能训练出优秀的 KSM，尽管提供奖励的次数有所减少。我们推测，这是因为阶段奖励可能比每个训练步骤的即时奖励更加准确、稳定。此外，我们还观察到，如果没有探索奖励，KSM 就无法更快地探索出更好的知识选择。

6.5 讨论

数据增强的性能 尽管训练好的 KSM 能够选择适当的知-识，但训练数据集中的样本较少，无法为学生模型提供学习这些知识的机会（Jiao 等人，2020年；Liang 等人，2021年）。在这种情况下，我们认为，如果配合数据增强，学生模型就能更充分地学习 KSM 分配的知识，从而取得更好的性能。为此，我们通过数据增强流程将训练数据集（即，RTE、MRPC、SST-2 和 QNLI 数据集）扩充约 20 倍（Jiao 等人，2020年）。为进行公平比较，我们使用通过通用蒸馏（Jiao 等人，2020年）发布的 BERT_6† 蒸馏得到的模型初始化我们的学生模型。表4 比较了我们的学生模型（软动作）与原始知识蒸馏、微调和 TinyBERT。结果表明，我们的方法始终优于所有基线方法。这说明，数据增强能够显著提升通过我们的方法蒸馏得到的学生模型的性能。此外，与原始知识蒸馏相比，我们的方法能够更高效地利用数据，并向学生模型迁移更多知识。

数据增强方面的这一性能表明，我们的方法与 Tiny-BERT 正交，因为我们解决的是知识蒸馏过程中的知识选择问题。这也表明，我们的方法可以与其他知识蒸馏方法互为补充，例如 MixKD（Liang 等人，2021年）和 MiniLM（Wang 等人，2020年）。

软动作与硬动作的比较为了直观地展示不同动作对性能的影响，我们在 RTE 数据集上比较了软动作和硬动作。图3 展示了在开发集上采用不同动作策略蒸馏得到的学生模型的平均准确率。可以发现，我们的方法能够与不同策略无缝结合。此外，我们注意到，硬动作能够加快 KSM 学习，而软动作能够取得更好的结果。在其他数据集上的结果中，我们也观察到了类似现象，例如，在规模相对较大的 QQP 数据集上（见附录中的图5 ）。

†https://huggingface.co/huawei-noah/TinyBERT_通用_6L_768D

不同学生模型上的性能 为了探究不同学生模型的性能，我们使用不同层数的学生模型进行蒸馏，并绘制其性能，以便与基线方法（即微调和原始知识蒸馏）进行比较，如图4所示。结果表明，在使用不同层数的学生模型进行蒸馏时，我们的方法始终优于基线方法。这说明我们的方法能够很好地适用于不同学生模型的蒸馏。

从正则化视角看知识选择 在实践中，我们认为知识选择可以发挥正则化作用，从而防止知识蒸馏中的协同适应（Grisogono，2006年；Sabiriet al.，2022年），即，对学生模型进行知识蒸馏高度依赖于教师的某种行为。如果蒸馏后的学生模型从教师的依赖行为中接收到不恰当的知识，就可能显著改变学生模型的性能，这正是过拟合可能导致的情况（Hawkins，2004；Phaisan-gittisagul，2016年）。然而，在知识蒸馏过程中，知识选择使学生模型能够在每个训练步骤中适当地从多种不同的行为中学习。因此，与通过单一或固定知识对学生模型进行知识蒸馏的传统知识蒸馏方法相比，我们的方法能够减轻来自某一行为的不恰当知识所产生的影响，从而防止协同适应。

更多讨论请参见附录C。

7 结论

本文着重于充分利用多种类型的知识来蒸馏学生模型。我们提出了一种演员—评论家方法，用于在知识蒸馏过程中选择适当的知识进行迁移。为减轻训练期间计算奖励的负担，我们提出了一种多阶段训练方法和一种探索奖励。在GLUE 数据集上的实验表明，我们的方法显著优于若干强大的知识蒸馏基线方法。

局限性

本节将讨论这项工作的若干局限性，具体如下：

- 我们训练一个模型，在知识蒸馏过程中选择合适的知识进行迁移。在这一过程中，训练该模型耗时较长且资源消耗较大。例如，在 RTE 数据集上，如果使用一块 TITAN V GPU，并将蒸馏训练轮次设为 5，则训练该模型需要 40 分钟。

- 很难将所提出的方法扩展到大型数据集。其根本原因是：(*i*) 超大型数据集需要更多训练步骤，这会使知识选择问题更加困难。(*ii*) 数据集越大，状态就越复杂，可能会使知识选择模块感到困惑。

致谢

本研究部分得到了中国国家自然科学基金（项目编号：61876035 和 61732005）、中国高技术研究发展中心项目（项目编号：2020AAA0107904）、云南省重大科技专项（项目编号：202002AD080001 和 202103AA080015）、国家工业智能与系统优化前沿科学中心（中国东北大学，项目编号：B16009）以及中央高校基本科研业务费的资助。我们还感谢匿名审稿人提供的宝贵意见。

参考文献

Yusuf Arslan、Kevin Allix、Lisa Veiber、Cedric Lothritz、Tegawendé F Bissyandé、Jacques Klein 和 Anne Goujon。2021年。金融领域多类别文本分类预训练语言模型的比较。载于万维网论文集。

Gaétan Baert、Souhir Gahbiche、Guillaume Gadek 和 Alexandre Pauchet。2020 年。用于情感分析的阿拉伯语拉丁字母转写语言模型。载于计算语言学国际会议论文集。

Enkhbold Bataa 和 Joshua Wu。2019年。日语中基于迁移学习的情感分析研究。载于*ACL* 论文集。

Luisa Bentivogli、Peter Clark、Ido Dagan 和 Danilo Giampiccolo。2009年。第五届 PASCAL 文本蕴含识别挑战。载于文本分析会议。

Shalabh Bhatnagar、Richard S Sutton、Mohammad Ghavamzadeh 和 Mark Lee。2009年。自然演员–评论家算法。《自动化》。

ZihanChen、Hongbo Zhang、Xiaoji Zhang 和 Leqi Zhao。2018年。Quora问题对。滑铁卢大学。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 KristinaToutanova。2019年。BERT：语言理解的预训练

深度双向 Transformer。*NAACL* 论文集。

William B. Dolan 和 Chris Brockett。2005年。自动构建句子释义语料库。载于*IWP* 论文集。

Jianping Gou、Baosheng Yu、Stephen J Maybank 和 Dacheng Tao。2021年。知识蒸馏：综述。《国际计算机视觉杂志》。

Anne-Marie Grisogono。2006年。协同适应。载于复杂系统，第6039卷，第23–37页。

Douglas M Hawkins。2004年。过拟合问题。化学信息与计算机科学杂志，44(1):1–12。

Geoffrey Hinton、Oriol Vinyals、Jeff Dean 等。2015年。神经网络中的知识蒸馏。*NeurIPS* 论文集。

Aref Jafari、Mehdi Rezagholizadeh、Pranav Sharma 和 Ali Ghodsi。2021年。退火式知识蒸馏。载于欧洲计算语言学协会会议论文集。

Xiaoqi Jiao、Yichun Yin、Lifeng Shang、Xin Jiang、Xiao Chen、Linlin Li、Fang Wang 和 Qun Liu。2020年。TinyBERT：通过知识蒸馏使 BERT 理解自然语言。载于*EMNLP Findings* 论文集。

Bei Li、Ziyang Wang、Hui Liu、Quan Du、Tong Xiao、Chunliang Zhang 和 Jingbo Zhu。2021a。从深度 Transformer 中学习轻量级翻译模型。载于美国人工智能协会人工智能会议论文集，第 35 卷，第 13217–13225 页。Lei Li 、 Yankai Lin 、Shuhuai Ren、Peng Li、Jie Zhou 和 Xu Sun。2021b。面向预训练语言模型的动态知识蒸馏。载于*EMNLP* 论文集。Kevin J. Liang、Weituo Hao、Dinghan Shen、Yufan Zhou、Weizhu Chen、Changyou Chen 和 Lawrence Carin。2021。MixKD：迈向大规模语言模型的高效蒸馏。载于国际学习表征会议论文集。Ye Lin、Yanyang Li、Ziyang Wang、Bei Li、Quan Du、Tong Xiao 和 Jingbo Zhu。2021。权重蒸馏：迁移神经网络参数中的知识。第 2076–2088 页。ACL 论文集。Xiaodong Liu、Pengcheng He、Weizhu Chen 和 Jianfeng Gao。2019a。用于自然语言理解的多任务深度神经网络。载于*ACL* 论文集。Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019b。RoBERTa：一种稳健优化的 BERT 预训练方法。*arXiv* 预印本 *arXiv:1907.11692*。

Seyed-Iman Mirzadeh、Mehrdad Farajtabar、Ang Li、Nir Levine、Akihiro Matsukawa 和 Hassan Ghasemzadeh。2020年。通过教师助理改进知识蒸馏。载于*AAAI* 论文集。

Ekachai Phaisangittisagul。2016年。单隐藏层神经网络中 L2 正则化与随机失活之间的分析。载于*ISMS*，第 174–179 页。IEEE。

Pranav Rajpurkar、Jian Zhang、KonstantinLopyrev 和 Percy Liang。2016年。SQuAD：100，000+ 道用于文本机器阅读理解的问题。载于*EMNLP* 论文集。

Bihi Sabiri、Bouchra El Asri 和 Maryem Rhanoui。2022年。深度神经网络训练中避免过拟合的技术机制。载于*ICEIS(1)*，第 418–427 页。

Victor Sanh、Lysandre Debut、Julien Chaumond 和 Thomas Wolf。2019年。DistilBERT：BERT 的蒸馏版本，更小、更快、更经济、更轻量。*arXiv* 预印本。

David Silver、Guy Lever、Nicolas Heess、Thomas De-gris、Daan Wierstra 和 Martin A.Riedmiller。2014年。确定性策略梯度算法。载于*ICML* 会议论文集。

Richard Socher、Alex Perelygin、Jean Wu、Jason Chuang、Christopher D. Manning、Andrew Ng 和 Christopher Potts。2013年。用于情感树库语义组合性的递归深度模型。载于*EMNLP*论文集。Samuel Stanton、Pavel Izmailov、Polina Kirichenko、Alexander A Alemi 和 Andrew G Wilson。2021年。知识蒸馏真的有效吗？*NeurIPS*论文集。Chi Sun、Xipeng Qiu、Yige Xu 和 Xuanjing Huang。2019a。如何对BERT进行微调以用于文本分类？载于*CCL*论文集。Siqi Sun、Yu Cheng、Zhe Gan 和 Jingjing Liu。2019b。用于BERT模型压缩的渐进式知识蒸馏。载于*EMNLP*论文集。Zhiqing Sun、Hongkun Yu、Xiaodan Song、Renjie Liu、Yiming Yang 和 Denny Zhou。2020年。MobileBERT：面向资源受限设备的紧凑型任务无关BERT。载于*ACL*论文集。Richard S Sutton 和 Andrew G Barto。2018年。强化学习：导论。麻省理工学院出版社。Haoran Tang、Rein Houthooft、Davis Foote、Adam Stooke、Xi Chen、Yan Duan、John Schulman、Filip De Turck 和 Pieter Abbeel。2017年。#探索：深度强化学习中基于计数的探索研究。载于*NeurIPS*论文集。Gerald Tesauro。1991年。时序差分学习中的实际问题。*NeurIPS*论文集，4。

Iulia Turc、Ming-Wei Chang、KentonLee 和 Kristina Toutanova。2019年。博览群书的学生学得更好：预训练紧凑模型的重要性。*arXiv*预印本 *arXiv:1908.08962*。

Alex Wang、Amanpreet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel R. Bowman。2019年。GLUE：用于自然语言理解的多任务基准与分析平台。收录于*EMNLPWorkShop* 论文集。

Wenhui Wang、Furu Wei、Li Dong、Hangbo Bao、Nan Yang 和 Ming Zhou。2020年。MiniLM：面向任务无关的预训练变换器压缩的深度自注意力蒸馏。*NeurIPS* 论文集，33：5776–5788。

Adina Williams、Nikita Nangia 和 Samuel Bowman。2018年。用于通过推理理解句子的广覆盖挑战语料库。收录于*NAACL* 论文集。

WeiYang、Yuqing Xie、Aileen Lin、XingyuLi、Luchen Tan、Kun Xiong、Ming Li 和 Jimmy Lin。2019年。基于 BERTserini 的端到端开放域问答。收录于*NAACL* 论文集（演示环节），第 72–77 页。

Junho Yim、Donggyu Joo、Ji-Hoon Bae 和 Junmo Kim。2017年。知识蒸馏的馈赠：快速优化、网络最小化与迁移学习。发表于*CVPR* 论文集。

Fei Yuan、Linjun Shou、Jian Pei、Wutao Lin、Ming Gong、Yan Fu 和 Daxin Jiang。2021年。用于知识蒸馏的强化多教师选择。载于 *AAAI* 论文集。

A 各类知识的学习设计细节

A.1 响应知识

为了学习响应知识，我们使用 vanillaKD 损失函数：

$$\mathcal{L}_{ResK} = \sum_{x \in B} KL(\sigma(\frac{f^S(x)}{\tau}), \sigma(\frac{f^T(x)}{\tau})) \quad (14)$$

其中 $KL(\cdot)$ 是库尔贝克-莱布勒散度函数，$\sigma(\cdot)$ 是Softmax函数，而 τ 是温度超参数（Hinton 等人，2019年）。对于样本 x，$f^S(x)$ 和 $f^T(x)$ 分别是学生模型和教师模型的最终输出。

A.2 特征知识

为了学习特征知识，我们采用 PKD-Skip 方法（Sun 等人，2019b），学生模型在该方法中学习指定教师模型各层的输出。相应的损失函数可定义为：

$$\mathcal{L}_{FeaK} = \sum_{x \in B} \sum_{i=1}^{L_S} \left\| \frac{\boldsymbol{h}^S_{x,i}}{\left\|\boldsymbol{h}^S_{x,i}\right\|_2} - \frac{\boldsymbol{h}^T_{x,I(i)}}{\left\|\boldsymbol{h}^T_{x,I(i)}\right\|_2} \right\|_2 \quad (15)$$

其中 $I(i)$ 是教师模型中对应学生模型第 i 层的指定层索引，L_S 是学生模型的层数，$\boldsymbol{h}^S_{x,i}$ 和 $\boldsymbol{h}^T_{x,i}$ 分别是样本 x中学生模型和教师模型第 i 层的 [CLS]嵌入。

A.3 关系知识

为了学习关系知识，我们采用解题流程（FSP）方法（Yim等人，2017年）。具体而言，我们首先将FSP矩阵定义为：

$$G(\boldsymbol{h}_1, \boldsymbol{h}_2) = \frac{\boldsymbol{h}_1 \times \boldsymbol{h}_2^\top}{|\boldsymbol{h}_1|} \quad (16)$$

其中，$\boldsymbol{h}_1$ 和 $\boldsymbol{h}_2$ 是 [CLS] 嵌入。令$G^S(\cdot)$和 $G^T(\cdot)$分别表示学生模型和教师模型的FSP矩阵。基于这两个FSP矩阵，我们可以得到用于学习关系知识的损失函数：

$$\mathcal{L}_{RelK} = \sum_{x \in B} \sum_{i=1}^{L_S - 1} MSE(G^S(\boldsymbol{h}^S_{x,i}, \boldsymbol{h}^S_{x,i+1}), G^T(\boldsymbol{h}^T_{x,I(i)}, \boldsymbol{h}^T_{x+1,I(i+1)})) \quad (17)$$

其中 $MSE(\cdot)$ 是均方误差损失函数。这里，我们对教师模型采用与 PKD-Skip 方法相同的层选择策略。

A.4 微调知识

为了学习微调知识，我们使用以下损失函数：

$$\mathcal{L}_{FinK} = -\sum_{x \in B} \mathbf{1}\{y = \hat{y}\} \log p(y|x) \quad (18)$$

其中，$\mathbf{1}\{\cdot\}$ 是指示函数，y 是样本 x的预测标签，$\hat{y}$ 是其真实标签。

B 训练设置

B.1 KSM 的训练

我们基于 PyTorch 实现第 5节所述的 KSM[‡]。具体而言，我们使用一个 2 层多层感知机来构建学生和教师特征网络。输入层大小和输出层大小分别为 [CLS]嵌入维度和 8。演员网络和评论家网络均由一个 3 层多层感知机构成。演员网络中的隐藏层维度和输出层大小分别为 256 和 4。评论家网络中的隐藏层维度和输出层大小分别为 256 和 1。此外，训练 KSM 时有六个超参数。这些超参数的候选值见表 5。训练 KSM 时，我们可以选择开发集准确率最高的超参数配置。

超参数	值
阈值 λ	{0.1, 0.2, 0.3}
训练步数 k	{32, 64, 96, 128}
缩放因子 α	{0.1, 0.2}
折扣因子 γ	0.98
演员的学习率ϵ_ϕ	0.0002
评论家的学习率 ϵ_θ	0.0002

表5：训练 KSM 的超参数。

B.2 训练教师模型

在这项工作中，我们将层数（即 Transformer 块的数量）记为 L，将隐藏层维度记为 H，并将 BERT 的自注意力头数量记为A 。我们使用预训练语言模型**BERT**$_{\textbf{BASE}}$ ($L = 12$，$H = 768$，$A = 12$，$TotalParameters = 110M$)（Devlin等人，2019年）作为教师模型。我们使用$_{\text{BASE}}$ bert-base-uncased[§]初始化 BERT。教师模型使用随机种子进行训练。此外，句子长度为 128，学习率为 5e-5。我们将批次大小和训练轮次分别设为 32 和 5，

‡ https://github.com/pytorch/pytorch

§ https://github.com/google-research/bert

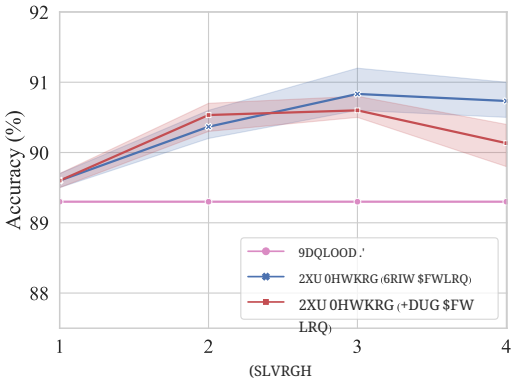


图 5：QQP 开发集上软动作与硬动作的比较。对于 QQP 数据集，我们只需经过两个回合就能训练出性能良好的 KSM。这是因为大型数据集能够为每个回合中的 KSM 更新提供更多机会。

奖励	RTE	MRPC	QNLI	Avg.
Loss	**68.4**	**89.3/84.7**	**89.0**	**80.7**
F1	*66.9*	87.8/82.9	*88.1*	*79.3*
准确率	66.7	*88.1/83.2*	87.3	79.1

表6：使用不同奖励训练 KSM 的结果。

。请注意，我们的方法可以接入任意大型预训练模型，例如 $\text{BERT}_{\text{LARGE}}$ (Devlin等人，2019年) 和 RoBERTa(Liu等人，2019b)。

B.3 学生模型的训练

我们主要报告两种学生模型的结果：**BERT**$_6$($L = 6, H = 768, A = 12, TotalParameters = 67.0M$) 和 **BERT**$_3$($L$ =3, $H = 768, A = 12, TotalParameters$ =45.7M)。我们分别使用 $\text{BERT}_{\text{BASE}}$ 最底部的 6 层和 3 层 Transformer 层来初始化 BERT_6 和 BERT_3。训练学生模型时，我们将批次大小设为 32，句子长度设为 128。我们从学习率集合 {2e-5, 3e-5, 5e-5} 中，根据模型得分选择训练良好的学生模型。

C 讨论

不同阶段大小下的性能 在第5.3.3节中，我们将整个知识蒸馏过程划分为多个阶段。由此可能产生一个问题：阶段大小是否会影响 KSM 的性能。为探究这一问题，我们使用不同的阶段大小，通过我们的方法对学生模型 BERT_6 进行知识蒸馏。如图 6所示，我们观察到，过大的阶段大小会损害训练后 KSM 的性能，我们认为这是由于阶段奖励不足所致

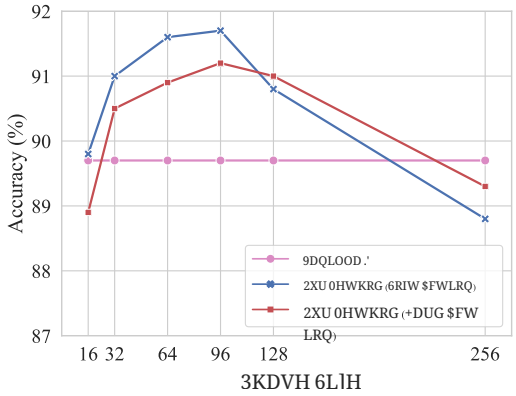


图6：使用不同阶段大小的方法蒸馏得到的学生模型的性能。

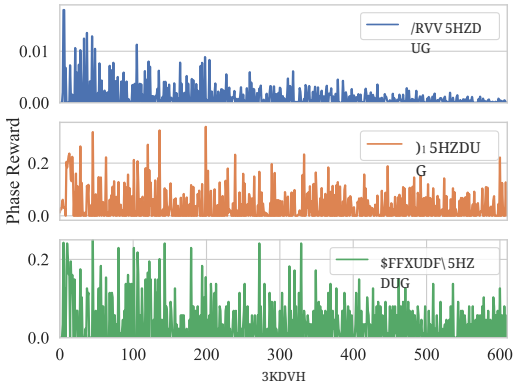


图 7：一个回合中不同阶段奖励的对比。损失奖励比 F1 奖励和准确率奖励更加稳定、平滑。

。此外，过小的阶段大小对我们的方法也无益，因为其阶段奖励与即时奖励相似。

不同阶段奖励下的性能 我们研究了采用任务指标（即，F1 值或准确率）差异作为阶段奖励时我们的方法的性能。如表 6所示，我们观察到，使用损失奖励训练 KSM 的整体性能最佳。为探究这一现象的原因，我们进一步比较了一个回合中不同奖励的阶段奖励，如图 7所示。结果表明，损失奖励比 F1 值奖励和准确率奖励更加稳定、平滑。基于此，我们得出结论：这种稳定性和平滑性使损失奖励能够训练出更好的 KSM。