



微信扫一扫
关注该公众号



文 | 小伟
编 | 小铁

前言

自从Google石破天惊地发布Bert以来，NLP就进入了预训练语言模型的时代。众所周知，我们可以用预训练语言模型来学习各种各样的任务，即使它们的特征空间有比较大的差异。那么预训练语言模型为什么会有这种泛化能力呢？或者说预训练阶段学习到的通用表示为什么可以很容易地适应广泛的下游NLP任务呢？

今天介绍的这篇文章从近来愈火火的 prompt tuning 的角度出发，对这个问题进行了初步的探索。它经验性地发现：预训练模型在各类下游任务上泛化的过程其实就是在优化各类任务的公共低维本征子空间（common low-dimensional intrinsic task subspace）中非常少量的几个自由参数。为了证明这一观点，作者在100多个 few-shot 任务上进行了实验，发现仅仅优化子空间中的 5个自由参数，就可以获得 full prompt tuning 87% 的性能。

至于，何为“任务的公共低维本征子空间”？作者又是如何论证得到上述结论的？在下文中我们将为大家仔细解读。

论文标题:

Exploring Low-dimensional Intrinsic Task Subspace via Prompt Tuning

论文链接:

<https://arxiv.org/pdf/2110.07867.pdf>

概览

本文对任务低维本征子空间的探索是基于 prompt tuning，而不是fine-tuning。原因是预训练模型的参数实在是太多了，很难找到这么多参数的低维本征子空间。作者基于之前的工作提出了一个基本的假设：预训练模型在不同下游任务上学习的过程，可以被重新参数化（reparameterized）为在同一个低维本征子空间上的优化过程。如下图所示，模型在不同的任务上学习的参数虽然不同，但这些参数共享了同一个低维本征子空间。

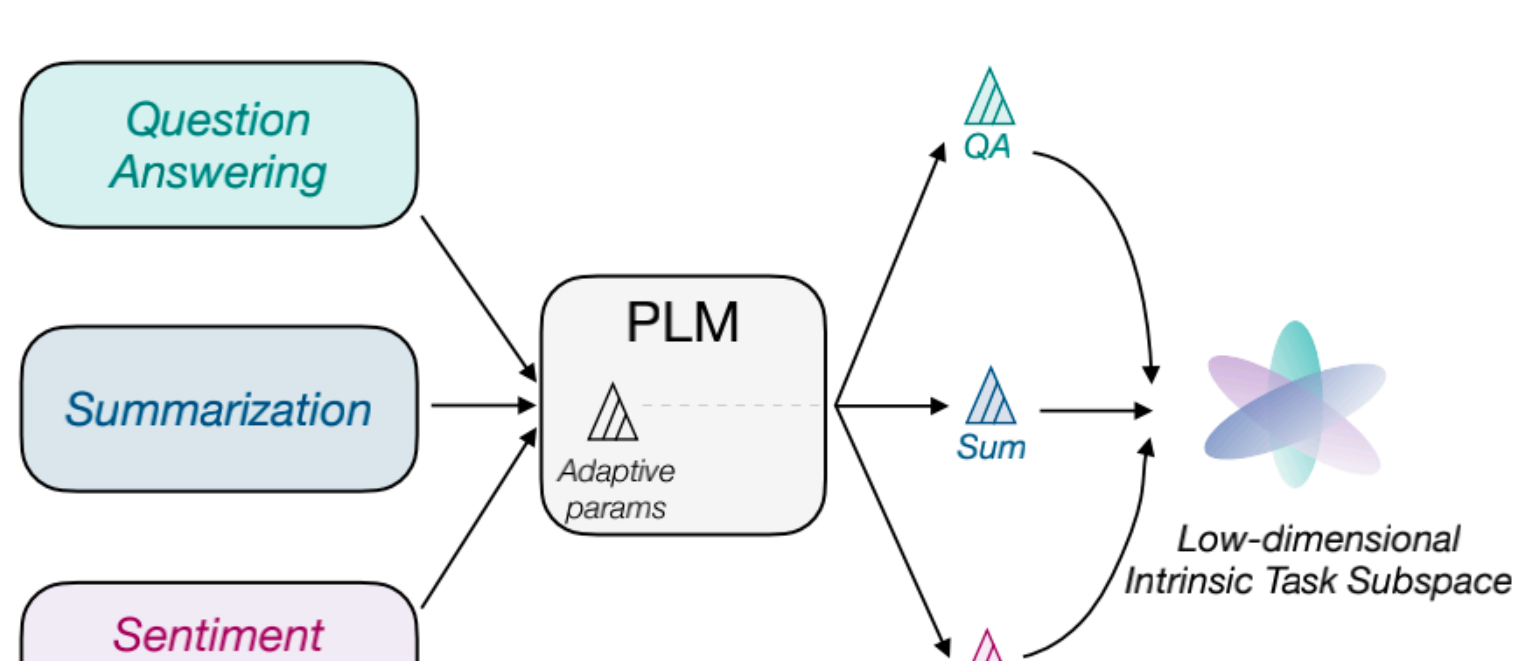


Figure 1: An illustration of a common low-dimensional intrinsic task subspace for diverse NLP tasks. During adaptation, PLMs learn tunable adaptive parameters for each task.

基于这一假设，作者提出了探索公共低维本征子空间的方法：intrinsic prompt tuning (IPT)。

IPT由两个阶段组成:

- Multi-task Subspace Finding (MSF): 寻找多个任务的公共子空间，这是一个低维的、更为本征的一个空间
- Intrinsic Subspace Tuning (IST): 在找到的公共本征子空间上进行模型优化

下图展示了 IPT 与 fine-tuning 和 prompt tuning 的对比。

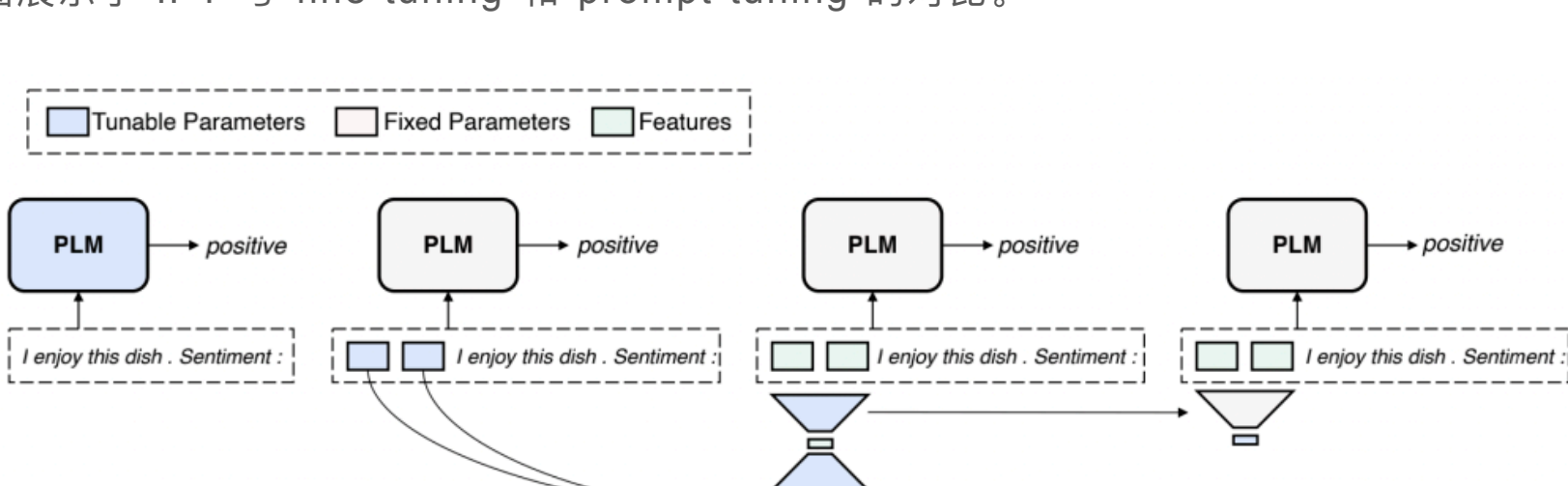


Figure 2: Illustrations for fine-tuning (a), prompt tuning (b) and two components of IPT (c,d). We discriminate tunable parameters, fixed parameters and intermediate features with different colors.

下面我们具体来了解一下IPT的两个阶段

IPT

作者使用 intrinsic prompt tuning (IPT) 来验证本文的基本假设：预训练模型对多个不同下游任务的学习可以被重新参数化为在同一个低维本征子空间上的优化。

第一个阶段是 multi-task subspace finding (MSF)。

1. 寻找公共本征子空间 (MSF)

MSF阶段旨在通过对多个任务进行学习，来找到公共的低维本征子空间。如上图(c)所示，本质上就是在学习一个自编码器。

我们用 **Proj** 来代表自编码器的Encoder部分(上图(c)中处于下方的梯形)，用 **Proj_b** 来代表自编码器的Decoder部分(上图(c)中处于上方的梯形)，那么自编码器会先用**Proj**把Prompt参数**P_i**映射为一个低维(d_f 维)的向量(向量所在的 d_f 维空间就是我们想要的低维本征子空间)，然后再用**Proj_b**把该低维向量重新映射回原始的prompt空间，得到 **P_i^{*}**这样我们就可以使用 **P_i** 和 **P_i^{*}**的距离来计算自编码器的重建loss \mathcal{L}_{AE}^i ，形式化表述就是:

$$\begin{aligned} \mathbf{P}_i^* &= \text{Proj}_b(\text{Proj}(\mathbf{P}_i)) \\ \mathcal{L}_{AE}^i &= \|\mathbf{P}_i^* - \mathbf{P}_i\|_2^2 \end{aligned}$$

另外，使用自编码器来学习公共低维本征子空间的最终目的还是为了解决多个任务，所以作者引入了面向任务的语言模型loss \mathcal{L}_{LM} 来提供任务相关的监督(例如图(c)中模型生成的结果"positive"和正确标签之间的交叉熵)。那么MSF阶段最终的loss就是:

$$\mathcal{L}_{proj} = \frac{1}{|T_{train}|} \sum_{i=1}^{|T_{train}|} (\mathcal{L}_{LM}^i + \alpha \mathcal{L}_{AE}^i)$$

其中 θ_{proj} 代表 **Proj** 和 **Proj_b** 的参数，这也是我们在MSF阶段要学习的参数。

2. 本征子空间优化 (IST)

在MSF阶段中，我们通过对多个任务的学习找到了 d_f 维的公共本征子空间，然后就进入了第二个阶段IST。在这一阶段中，我们想评价我们在MSF阶段中找到的低维本征子空间是不是能够很好的泛化到 (a) MSF阶段训练过的任务的新数据，以及 (b) MSF阶段没有训练过的任务。如果该低维本征子空间在这两种情况下都有比较好的泛化性能的话，那么我们在一定程度上就成功地找到了想要的本征子空间。

在本阶段中，如上图 (d) 所示，我们只保留自编码器的Decoder部分**Proj_b**并冻结它的参数。对于每个测试任务，我们只微调本征子空间中的 d_f 个自由参数 θ_d ，**Proj_b**会将 θ_d 解码回原始的prompt空间中来计算loss:

$$\mathcal{L}_{\theta_d}^i = \mathcal{L}_{LM}^i$$

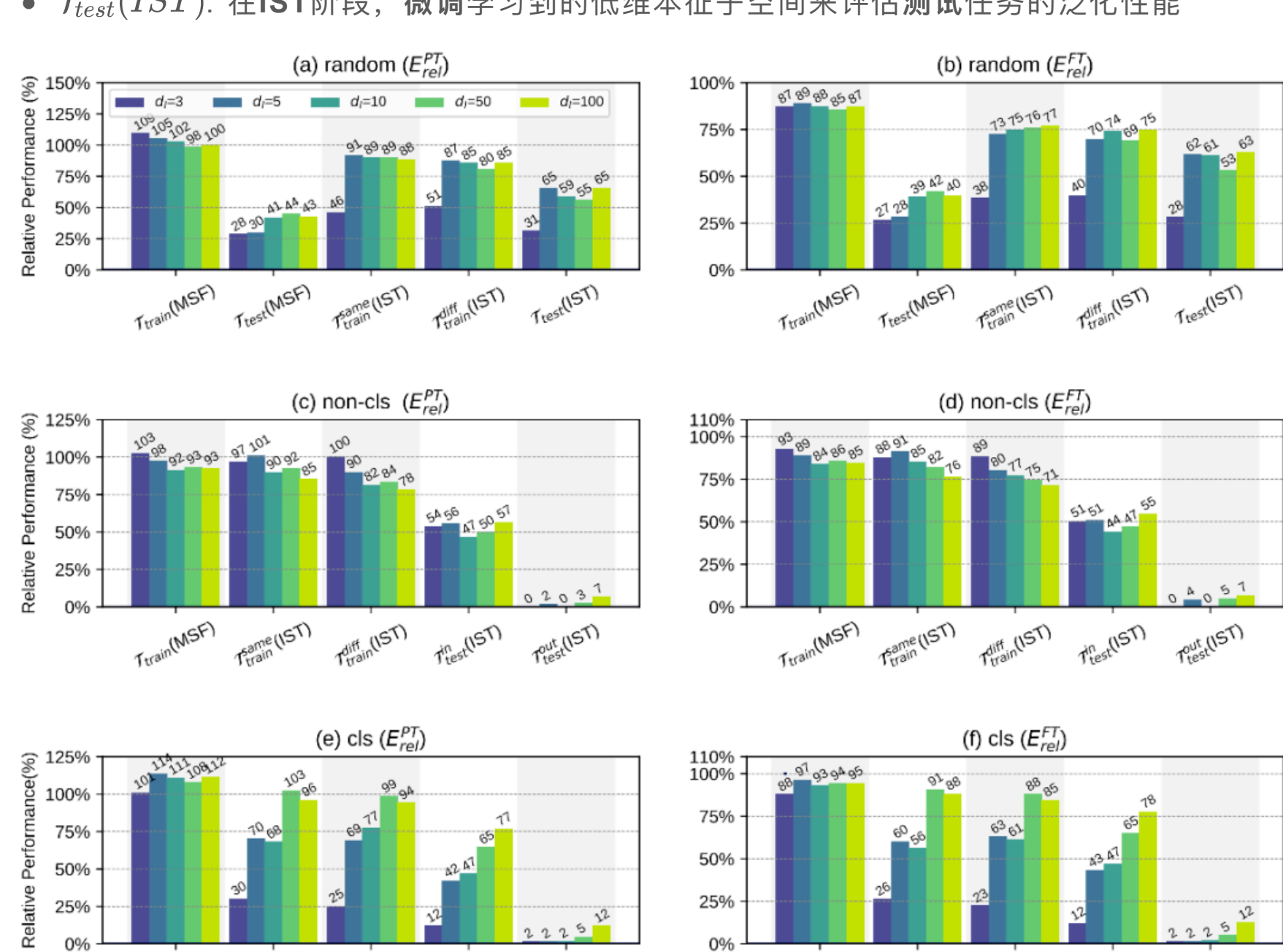
实验

作者使用了120个few-shot任务来进行实验，并进行了三种不同的训练-测试任务划分

- random: 随机选择100个任务作为训练任务，其余20个任务作为测试任务
- non-clis: 随机选择非分类任务中的35作为训练任务，其余所有任务作为测试任务
- cls: 随机选择分类任务中的35个作为训练任务，其余所有任务作为测试任务

同时，对每一种任务划分，作者进行了5种不同的实验

- $\mathcal{T}_{train}(MSF)$: 在MSF阶段，直接使用学习到的低维本征子空间来评估训练任务在训练数据上的性能
- $\mathcal{T}_{test}(MSF)$: 在MSF阶段，直接使用学习到的低维本征子空间来评估测试任务(0-shot)的泛化性能
- $\mathcal{T}_{train}^{same}(IST)$: 在IST阶段，微调学习到的低维本征子空间来评估训练任务在训练数据上的性能
- $\mathcal{T}_{train}^{diff}(IST)$: 在IST阶段，微调学习到的低维本征子空间来评估训练任务在新数据上的泛化性能
- $\mathcal{T}_{test}(IST)$: 在IST阶段，微调学习到的低维本征子空间来评估测试任务的泛化性能



整体的实验结果如上图所示，作者通过分析不同实验的结果，得出了一些比较重要的结论：

在random划分中，仅仅微调低维本征子空间中的5个自由参数，就可以分别获得full prompt tuning 87%(训练过的任务，不同训练数据)以及65%(未训练过的任务)的性能，这证明我们在MSF阶段中找到的低维本征子空间是比较有效的。但从另一个方面来讲，使用低维本征子空间无法获得和full prompt tuning相当的性能，所以我们不能直接得出预训练模型对多个任务的学习可以被重新参数化为在完全相同的子空间中的优化的结论。

训练-测试任务的划分会对结果有很大的影响。比如在cls划分中，训练时找到的本征子空间可以在分类的测试任务上有比较合理的表现，但在非分类的测试任务上表现很差。

随着MSF阶段中训练任务数量的增加，找到的本征子空间的泛化能力会有所提高。这反映了增加MSF阶段中训练任务的覆盖范围和多样性可以帮助IPT找到更通用的本征子空间。

结论

本文设计了IPT框架来验证提出的假设：预训练模型对多个不同下游任务的学习可以被重新参数化为在同一个低维本征子空间上的优化。详尽的实验为假设提供了一定的积极证据，也帮助大家对如何更高效地使用预训练语言模型有了更好的了解。

思考

虽然文章中的实验结果不能直接验证“预训练模型对多个任务的学习可以被重新参数化为在完全相同的子空间中的优化”这一假设是完全正确的，但起码它证明了各种任务重新参数化后的低维子空间是有比较大的交集的，而且我们可以通过MSF来找到这个交集。

感觉接下来的工作应该是设计更好的框架来验证这个公共的低维本征子空间是不是真的存在；以及如果真的存在，怎么更好的提高它的泛化性能。PS: 如果真的存在这样的跨任务低维本征子空间，是不是意味着彻底不需要fine-tuning了呢？

鸣谢

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈

感谢以下朋友的支持和反馈