

AI播客：GSPO算法的突破

主持人A：嘿，咱们最近聊聊天工智能的时候，有没有发现那些聊天机器人回答问题是越来越溜了？不光能写点小诗改改代码，甚至数学题也能解那么几道。

嘉宾B：对对对，这个进步确实挺明显的。其实啊，背后有个特别关键的技术在推动，叫强化学习。不过呢，训练这些大模型可不容易，经常遇到一个让人头大的问题，业内管道叫模型崩溃。

主持人A：模型崩溃？这个说法听着就挺吓人，具体啥意思啊？是说这模型学着学着突然就废了吗？

嘉宾B：差不多就是这意思。你想啊，就像我们教小孩学骑自行车，他本来蹬得挺好，结果摔了一跤，之后怎么鼓励都不敢再骑了，彻底放弃了。模型崩溃就有点类似这种情况。

主持人A：哦哦，明白了，就是训练过程中出现了某种内伤，导致后面再怎么努力也恢复不过来了。

嘉宾B：没错没错，这个问题在大模型训练，尤其是那些需要生成常文本的场景下，表现的特别突出，是困扰大家很久的难题。

主持人A：诶，我记得之前阿里巴巴的QWAgent团队好像在这方面有了突破，是不是叫GSPO？

嘉宾B：啊，你消息还挺灵通，对，就是GSPO,全称是Global Sequence Policy Optimization,翻译过来大概叫分组串行策略优化，这名字听着有点复杂，但它的确是针对这个问题开出的咬法。

主持人A：那它取代的是谁啊？之前主流用的是啥算法？

嘉宾B：之前呢，大家觉得JRPO就是Group-Wise Relative Policy Optimization,分组相对策略优化，已经是挺大的进步了，比更早的PPO要好。

主持人A：PPO我知道，它是不是需要额外搞一个模型来评估每个词的重要性，这个特别耗资源是吧？

嘉宾B：是的，所以JRPO就把它砍掉了，不再训练那个专门的价值模型，改用一种思路，就是在同一个问题下，声称多个回答，然后比较这些回答哪个更好，哪个稍差一点，用这个相对优势来作为奖励信号。

主持人A：这听着挺合理啊，那为啥JRPO还是有问题呢？

嘉宾B：问题出在它虽然不训练价值模型了，但在计算重要性采样权重的时候，还是在一个一个单词token的级别上操作的。

主持人A：单词级别具体啥意思？

嘉宾B：就是它计算权重的时候，是看生成每个词时，新策略和旧策略的概率比值，这个权重是用来校正数据分布偏差的，让新策略能有效利用旧策略产生的样本。

主持人A：然后呢？

嘉宾B：麻烦在于，生成一个长句子或者段落，那是有很多很多个词组成的，对吧？每生成一个词，就要计算一次这个权重。你想想，如果某个词的权重计算有那么一点点误差或者噪声，这个误差会随着句子变长，一个词一个词的累积下去。

主持人A：哦，就跟滚雪球似的，前面一个小偏差，后面越滚越大。

嘉宾B：对，就是这个意思，就像你让小朋友写一篇作文，你只盯着他每个字有没有写对，有没有写公整，却完全不看整段话的意思通不通顺，上下文连不连贯，结果可能就是每个单字都挑不出毛病，但整篇作文读起来不知所云，完全跑题了。

主持人A：哈哈，这个比喻太形象了，所以就是太专注于局部细节，忽略了整体目标。

嘉宾B：精辟，这RPO的根本问题就在于它的优化单位，单个单词和最终的目标整个句子或段落的得分奖励，在层级上错配了，它在一个个单词上费劲地做校正，但奖励是针对整个串行给的。这就好比用改错别字的方法想提高作文整体分数，方向不对头。

主持人A：那这错误的方向具体会引发啥后果，除了你说的学求效应？

嘉宾B：后果很严重，首先是这种逐词计算的权重本身在数学上就不太站得住脚，因为重要性采样的原理要求要有足够多的样本来平均，才能正确估计概率比值，但JRPO在每一步生成时，实际上只用了当前这个词的一个样本来算权重。这就好比你想知道全班平均身高，结果只量了一个同学的身高就宣布结果了。差不多是这个道理，这就导致两个大问题，第一，它没能真正实现有效的分布校正，第二，更重要的是它把巨大的高方差的噪声注入了梯度更新过程里。

主持人A：梯度更新，就是模型学习时调整参数的那个方向？

嘉宾B：对对，就是那个梯度，这个噪声污染了梯度，让模型的学习方向都错了，而且随着

句子变长，声称更多单词，这个噪声就叠加的越来越多，越来越大。

主持人A：那这不就乱套了吗？模型崩溃是不是就这样发生的？

嘉宾B：完全正确，实验观察它的现象非常明确，一旦出现崩溃的苗头，比如模型性能断压时下跌，这时候你就算回到训练到一半的存档点，或者拼命微调学习率这些参数，甚至换一批问题让它继续练，或者让它声称长一点的回答，都救不回来了，模型彻底死机，学不进新东西了。

主持人A：天哪，这么严重，完全不可逆。

嘉宾B：至少在JRPU框架下，目前看是这样，问题出在方法本身有根本缺陷。理论分析证实了，根源就是优化单位单词和奖励作用单位整个串行的错配，你非得在单词这个局部层面做优化，去迎合串行层面的整体目标，这在数学原理上就有点拧巴，矛盾不可调和。所以GSPO的解法就是直接在这个整体层面，也就是串行级别上做文章。

主持人A：太对了，GSPO的内核创新，就是把重要性采样权重从token级别一下提升到了sequence级别，它不再计算每个新词和旧词的概率比值了，而是直接计算整个句子或者段落在新策略和旧策略下出现的概率比值。

主持人A：啊？整个串行的概率比值？这个怎么算啊？感觉好抽象。

嘉宾B：其实有办法，语言模型生成一个串行，本质上是逐词生成的，对吧？每个词生成的概率都依赖于前面的词，所以整个串行出现的概率就等于这些逐词生成的条件概率乘几了。

主持人A：哦，对，所以串行级的概率比值，其实就是这些词级条件概率比值连成的结果。

嘉宾B：聪明，就是这个意思。这个串行级的比值frag,帕西达y一个字，帕西达o的y一个字，在数学上有非常明确的意义。它直接反映了用旧策略采样得到的某个回答y,在新策略下出现的可能性偏离了多少，偏离的越大，这个比值就离一越远，要么远大于一，要么远小于一。

主持人A：明白了，这个比值越大，说明新策略特别喜欢这个回答，比值越小，说明新策略特讨厌这个回答。

嘉宾B：可以这么理解，有了这个在数学上更可靠的权重，JSPO就能在串行级别进行有效的分布校正了。

主持人A：那具体怎么用它来训练呢？GRPO不是还有裁减机制吗？

嘉宾B：JSPO也保留了裁减机制，但它不是在单个词上剪裁了，而是在整个串行的权重上进行裁减。

主持人A：串行级裁减？这怎么操作？

嘉宾B：还是设定一个范围，比如1是一个小值，如果一个串行的权重算出了特别大，远大于一加，或者特别小，远小于一，就说明这个回答在新旧策略下差异太大了，可能是噪声太大，或者无效样本，就直接把它裁减掉，不让它过度影响梯度的计算。

主持人A：哦，这样就能避免那些极端样本把训练带沟里去，那它怎么利用这个串行级的优势advantage呢？

嘉宾B：问得好，JSPO采用了一种叫分组优势估计的方法，简单说，它对同一个问题生成的多个回答进行组内比较。

主持人A：组内比较？

嘉宾B：对，比如一个问题生成了32个回答，把它们分成一组，然后在组内对每个回答的相对优势进行计算，具体就是把每个回答的原始奖励值减去这组回答的平均奖励，再除以标准差，进行一个标准化。

主持人A：哦，这就把不同问题的绝对分数差异去掉了，只关注同一个问题下不同回答的相对好坏？

嘉宾B：没错，这样比较更有意义，能更准确地告诉模型哪个回答相对于本组的其他回答更优。所以整个JSPO的目标函数就是把串行级的权重和分组标准化后的优势结合起来？

嘉宾B：是的，公式上看就是串行权重乘以标准化优势，然后同样做裁减处理，目标就是让模型尽量增大那些相对优势高的串行在新策略下出现的概率权重。

主持人A：听起来挺优雅的，那效果怎么样？

嘉宾B：效果非常显着，首先训练过程变得超级稳定，那种可怕的模型崩溃现象几乎消失了。

主持人A：那太好了，效率呢？训练是不是更快了？

嘉宾B：效率也大幅提升，尤其是在处理声称常文本任务时，JSPO的优势更明显，因为它处理的是一个整体的权重，而不是把误差分散到几十个上百个词上来记。

主持人A： 诶， 你刚才提到一个JSPO Token, 这又是啥？ 跟JSPO啥关系？

嘉宾B： 哦， JSPO Token是JSPO的一个变种， 或者叫增强版， JSPO本身是在串行级别做整体的优化， 但在有些特别复杂的任务里， 比如多轮对话这种， 光优化整体可能还不够精细。

主持人A： 怎么讲？

嘉宾B： 想象一下多轮对话， 你问一个问题， 回答很长一段话， 可能这段话里某几个关键的词或句子决定了回答的好坏， 而其他词可能没那么重要， 串行级别的优化把整个回答一视同仁了。

主持人A： 哦， 对， 可能有些词至关重要， 有些词相对次要。

嘉宾B： JSPO Token就是为了解决这个问题诞生的， 它把优势估计细化到每个词Token级别。

主持人A： 那它又回Token级别了， 这不是回到JRPO的老路了吗？

嘉宾B： 不不不， 别担心， 它巧妙的结合了两者， JSPO Token的内核是引入了双重控制机制。

主持人A： 双重控制？

嘉宾B： 是的， 第一重， 它仍然计算整个串行在新旧策略下的概率比值作为基础权重， 这个权重反映了串行的整体质量。然后第二重， 它在每个词的位置引入一个局部的权重， 这个权重基于声称这个词的新旧策略条件概率比值。所以它既有全局串行级的权重， 又有局部词级的权重。

嘉宾B： 差不多， 但设计的很巧妙， 它用了一个叫梯度停止或者冻结梯度的操作， 技术上叫SG操作或Detach, 让串行级权重的信息流到每个词内里， 但是反过来， 每个词的优化不会影响整个串行级权重的计算。

主持人A： 等等， 有点抽象， 能不能再通俗点？

嘉宾B： 可以这么想， 串行级权重像是一个总指挥， 给整个回答定了个基调， 判断这个回答总体上偏离旧策略多少， 然后这个信息会下达到每个词那里， 告诉他们， 喂， 你们这个词所在的这个回答总体偏离度是这样的。接着， 每个词再根据自己的上下文， 就是它前面那些

词，计算出它自己在新旧策略上的偏离度，词集权重，最后，每个词最终的调整力度，是综合考虑了总指挥给的信息串行权重，和自己局部的情况，词集权重来决定了。

主持人A：哦，明白了，这样就能做到在保持整体稳定性的前提下，对关键位置的关键词进行更精准的优化。

嘉宾B：完全正确，而且它在每个词的级别上也应用了裁剪，防止某个词的权重波动太大，影响整体。这特别适合多轮对话或者指令跟随这类任务，能精准强化那些对任务完成至关重要的关键词。

主持人A：听起来GSPO和GSPO Token都很强大，那实际应用效果如何？特别是你提到它在阿里最新的昆三模型里应用了。

嘉宾B：效果确实非常令人兴奋。首先最直观的是训练稳定性的大幅提升。以前用JRPU训练超大的混合专家模型MOE，比如Curl3系列里的某些模型，是非常棘手的。

主持人A：哦，混合专家模型，这个能简单解释下吗？我听说过，但不太懂。

嘉宾B：好，简单打个比方，普通的语言模型就像一个知识渊博的全科医生，啥都得懂点，而MOE模型则像组建了一个专家团队，有专门负责历史的，有专门负责编程的，有专门负责写诗的等等。每次遇到一个问题，模型会根据问题类型，路由到几个最相关的专家那里，主要听他们的意见，其他专家暂时休息。

主持人A：哦，这样就能让模型更庞大，参数更多，但实际计算时不会太慢。

嘉宾B：对，这就是MOE的内核优势。但是，训练MOE模型，特别是用强化学习训练时，这RPO遇到了大麻烦，因为每次模型更新，那些被激活的专家成员可能会有变化。

主持人A：为什么会有变化？

嘉宾B：因为策略更新了，模型对问题的理解方式变了，它觉得需要请教的历史专家或者编程专家可能就变了，这个变化还挺剧烈，大概每次更新，可能有百分之几到百分之十几的专家切换。

主持人A：这跟集RPO有啥冲突？

嘉宾B：冲突可大了。这RPO依赖Token集的重要性全弄，这个全弄计算需要就策略下各个Token的概率，但MOE模型中，一个Token的概率是由被激活的那几个专家决定的。如果这

「人文科学」，或「自然科学家」，得出不同的结果。也就是说，不同的专家组合都不同了。

主持人A：哦，这就好像你用A医生团队的诊断结果来评估B医生团队的处方水平，这不乱套了吗？

嘉宾B：比喻得非常好，结果就是JRPO计算出的Token集重要性全中，完全失准，噪声巨大，模型根本无法正常学习，经常崩溃。

主持人A：那之前怎么解决这个问题？

嘉宾B：之前只能用一种叫Routing Replay的策略来勉强应付。Routing Replay，听起来像重放？是的，它的内核思想就是，在训练过程中，缓存旧策略下每次生成每个词时，具体激活了哪几个专家，然后用这个缓存下来的路由信息去计算旧策略下的Token概率。就相当于把当时的专家组合记下来，后面计算全中时强行用这个组合，不管后来策略怎么变。

主持人A：哦，相当于给模型训练时带了个紧箍咒。

嘉宾B：是的，而GSPO的出现彻底解决了这个问题。

主持人A：怎么解决的？

嘉宾B：关键就在于串行级的优化，GSPO只关注整个串行的概率比值 π_c 的 y_{ix} , π_c 的 o 的 y_x ，计算这个比值只需要知道在给定问题 x 下，整个串行 y 由旧策略生成的概率是多少，而不需要关心在生成串行中每个词具体激活了哪些专家。

主持人A：哦，对，因为串行的整体概率是把所有词的生成概率乘起来得到的，这个计算过程本身就隐含了当时的专家激活情况。

嘉宾B：没错，所以当策略更新后，专家激活模式变化了，这对GSPO没有直接影响，它只需要知道在旧策略下这个回答出现的概率是多少，而不需要知道这个概率是由哪些专家具体怎么算出来的细节，这样就把对专家激活波动的敏感性大大降低了。

主持人A：太棒了，这相当于从源头上规避了问题。

嘉宾B：是的，所以实验证明，用GSPO训练Mole模型再也不用依赖那个笨重的Routing Replay策略了，训练变得非常稳定和高效。

主持人A：效率提升具体体现在哪？

嘉宾B：首先省掉了Routing Replay带来的额外内存开销和计算开销，其次也是更重要的，GSPO的裁剪虽然比例看起来高一些，图二显示GSPO裁剪掉大约0.15%的样本，而GRPO因为用了裁剪范围非常小的Token级裁剪，只裁掉0.0013,前文值1.3,但GSPO反而效率更高。

主持人A：啊？裁剪比例高反而效率高？为啥？

嘉宾B：这就是GSPO串行极优化的精妙之处了，因为它裁剪的是那些整体上偏离旧策略很远的不合格串行，一次性全扔掉，避免这些串行里的每个词带着造成污染梯度，而GRPO裁剪范围太小，很多带着噪声的样本，特别是那些只有个别词权重很极端的样本，没有被有效过滤掉，结果这些噪声词对梯度的污染持续存在，反而拖慢了有效学习。

主持人A：所以GSPO虽然表面上裁剪的多点，但换来的是更纯净的学习信号和更快的有效受点。

嘉宾B： 哟，有道理，这就像除草，GSPO是整片地看到杂草多的区域就不管了，集中精力种好草少的区域，GRPO是小心翼翼的试图拔掉每一棵杂草，结果效率低下，还容易漏拔。

嘉宾B：这个比喻非常贴切，除了稳定性，GSPO在训练大规模模型时，也展现出了超强的扩展性，Scalability,你可以通过增加训练部署，更新查询集，就是让模型回答新问题，或者允许模型生成长度更长的文本，来持续提升模型性能，不像以前那样容易碰到瓶颈。

主持人A：那最终的模型效果呢？

嘉宾B：效果提升非常显著，比如在QWE N3系列模型上应用GSPO后，再像AME24这样的数学解题测试，LiveCodeBench这样的编程能力测试，还有CodeForce这种高难编程竞赛题测试上，都取得了比用GRPO训练更好的成绩，而且是在相同训练资源下做到了。

主持人A： 哇，这真是实实在在的进步，看来GSPO确实是解决大模型强化学习训练不稳定问题的关键钥匙了。

嘉宾B：可以这么说，它不仅在QWE系列上成功了，这项技术展现出来的从局部优化Token级，转向全局优化Sequence级的思想，其实具有很强的普世性。

主持人A：你是说，这个思路可能不局限于聊天机器人，还能用到其他地方？

嘉宾B：没错，虽然论文主要集中在语言模型上，但这个内核思想关注整体的质量评价，而

非局部的微小调整，不仅能对计算机视觉产生的串行瓶颈，比如识别抽丝瓶颈，机器人的控制动作串行规划等领域都有启发意义，它代表了算法设计范式的一种重要转变。

主持人A：确实，有时候跳出细节从更高维度看问题，反而能发现更有效的解决方案。

嘉宾B：非常赞同，技术的突破往往就源于这种视角的根本性转变。好了，朋友们，今天我们深入探讨了阿里巴巴Qwen团队提出的突破性算法GSPO, 它如何通过串行级的优化策略解决了困扰大模型强化学习训练的稳定性难题。记住，在追求进步的路上，有时候退一步，海阔天空，从整体着眼，反而能找到更优的路径。

主持人A：没错，这项创新不仅提升了现有模型的性能，更为未来更大规模、更复杂模型的训练铺平了道路，继续推动人工智能能力的边界向前拓展。今天的分享就到这里，下期再见。