

# NoRML:No Research Meta Retrieval Learning

王前明

## 摘要

在该论文中，作者提出了一种新的元学习方法，称为 NoRML（无研究元检索学习）。元学习，也称为“学会学习”，是利用过去的学习经验来提高学习新任务的能力的过程。这与传统的机器学习形成了鲜明对比，后者涉及在固定数据集上训练模型，然后使用它对新数据进行预测。NoRML 方法基于这样一个理念，即如果元学习不依赖人工标记的训练数据，它可以更有效。相反，模型是使用研究论文及其相关元数据（如标题、摘要和关键字）的大型非结构化数据集进行训练的。然后，该模型能够学习如何根据新研究论文的内容对其进行分类，而无需手动标记。为了评估 NoRML 方法的性能，作者在计算机科学领域的研究论文数据集上进行了一系列实验。他们发现，NoRML 模型能够准确地对研究论文进行分类，并优于其他元学习方法。总之，本文提出了一种新的元学习方法，该方法不依赖人工标记的训练数据，并在真实世界数据集上证明了其有效性。

**关键词：**元学习；非结构化数据；分类

## 1 引言

论文“NoRML:No Research Meta Retrieval Learning”讨论了一种新的信息检索方法，即根据用户的查询定位相关文档的过程。近年来，使用机器学习技术来提高信息检索系统的性能已经发生了重大转变。这些技术被称为元检索学习，涉及训练机器学习模型，以基于一组标记的示例来预测哪些文档与给定查询相关。

“NoRML”的作者提出了一种新的元检索学习方法，该方法不需要任何标记的训练数据。相反，该模型在大量未标记文档上进行训练，并通过自我监督学习学习检索相关文档。作者认为，这种方法有几个优点，包括能够扩展到大型文档集合，以及能够适应新的领域，而不需要额外的标记数据。

这项工作的意义在于，它有可能彻底改变信息检索系统的设计和培训方式。通过消除对标记数据的需求，NoRML 方法可以大大减少构建和维护信息检索系统所需的时间和资源，使组织和个人能够更广泛地访问这些系统。此外，在不需要额外标记数据的情况下适应新领域的能力可以使信息检索系统更加灵活，并对不断变化的用户需求作出响应。

## 2 相关工作

在信息检索和使用机器学习技术来提高其性能方面，已有大量的研究。相关工作的一些值得注意的例子包括：

**学习排名：**这一领域的研究重点是使用机器学习技术来优化搜索结果的排名。这可以通过多种方法实现，包括使用有监督的学习技术在标记数据上训练排名模型，或使用无监督的技术基于用户反馈学习排名。

**神经信息检索：**近年来，使用深度学习技术（如神经网络）进行信息检索的趋势越来越大。这些方法在几个基准上取得了最先进的结果，并有可能显著提高信息检索系统的性能。

元检索学习：“NoRML”论文是元检索学习这一更广泛研究领域的一部分，该领域涉及使用机器学习技术来提高信息检索系统的性能。这一领域的其他值得注意的工作实例包括使用元学习技术来适应新领域，以及使用自我监督学习来从大量未标记文档中学习相关性。

总之，“NoRML”论文通过提出一种不需要任何标记数据的元检索学习新方法，对信息检索领域做出了重要贡献。这种方法有可能显著提高信息检索系统的效率和灵活性，并可能对我们访问和检索信息的方式产生广泛影响。

## 2.1 学习排名

学习排名是信息检索的一个子领域，它专注于使用机器学习技术来优化搜索结果的排名。学习排名的目的是生成一个排名函数，该函数可以将一个查询和一组文档作为输入，并输出与查询最相关的文档的排名列表。学习排名有几种方法，包括监督学习、无监督学习和强化学习。在监督学习中，机器学习模型在由查询及其相应的相关文档组成的标记数据集上进行训练。然后，该模型用于基于新查询与训练数据的相似性对新查询进行排序。另一方面，无监督的学习方法不依赖标记数据，而是使用协作过滤等技术来基于用户反馈学习排名。强化学习方法包括通过试错来训练排名模型，奖励其将相关文档排名靠前，惩罚其将无关文档排名靠后。学习排名有广泛的应用，包括搜索引擎、推荐系统和信息检索系统。它有可能通过提供更相关和准确的搜索结果来显著提高这些系统的性能。此外，学习排序算法可以根据特定应用程序的特定需求进行定制，使其高度灵活，并可适应不断变化的用户需求。

## 2.2 神经信息检索

神经信息检索是信息检索的一个子领域，涉及使用神经网络（一种深度学习模型）来提高信息检索系统的性能。这些模型在大型文本数据集上进行训练，并学习以类似于人类处理语言的方式表示单词和短语的含义和上下文。这使他们能够更好地理解用户查询的意图，并更准确地检索相关文档。神经信息检索有几种方法，包括使用词嵌入、卷积神经网络和递归神经网络。单词嵌入涉及将单词表示为高维空间中的向量，其中相似的单词更接近。卷积神经网络用于通过应用一系列滤波器从数据中提取特征来处理文本，而递归神经网络用于以固定长度序列迭代输入来处理顺序数据，例如文本。神经信息检索已经在几个基准上取得了最先进的结果，并有可能显著提高信息检索系统的性能。它特别适合于处理复杂的查询，理解单词和短语的上下文和含义，这使得它特别适用于自然语言处理和问题解答等应用。

## 2.3 元检索学习

元检索学习是机器学习的一个子领域，涉及通过使用关于任务本身的附加信息来训练模型以学习执行任务。该附加信息（称为元数据）可以包括有关模型试图解决的特定问题、正在处理的数据结构或模型在类似任务上的性能的信息。元检索学习的一个常见应用是自然语言处理（NLP）领域，它用于训练模型以执行广泛的任務，如语言翻译、文本摘要和问题解答。在这些任务中，元数据可能包括有关所使用语言的结构、使用文本的上下文或文本中不同单词和短语之间的关系的的信息。元检索学习的另一个应用是强化学习领域，它用于训练模型以在复杂、动态的环境中做出决策。在这些任务中，元数据可能包括有关环境状态、模型可用的操作以及这些操作的后果的信息。总的来说，元检索学习有可能通过向机器学习模型提供有关其尝试执行的任务的附加信息来显著提高机器学习模型的性能。这可以帮助模型做出更明智的决策，从而获得更好的结果和更有效的解决现实问题的方法。

### 3 本文方法

#### 3.1 本文方法概述

元检索学习涉及通过使用关于任务本身的附加信息（称为元数据）来训练模型以执行任务。该元数据可以包括有关模型试图解决的特定问题、正在处理的数据结构或模型在类似任务上的性能的信息。要在机器学习模型中应用元检索学习，我们可以遵循以下步骤：收集和准备元数据：这包括收集任务的相关元数据，并根据需要对其进行预处理。定义元检索学习任务：这包括指定模型试图解决的具体问题以及用于训练它的元数据类型。训练模型：这涉及使用元数据来训练模型以执行任务。这可以使用各种机器学习算法来完成，例如监督学习、无监督学习或强化学习。评估模型：这涉及使用测试数据来评估模型的性能，并确定使用元数据执行任务的能力。微调模型：如果模型的性能不令人满意，我们可以调整模型的参数或用于训练它的元数据，以提高其性能。总体而言，元检索学习的目标是使用元数据帮助模型学习更有效地执行任务并获得更好的结果。

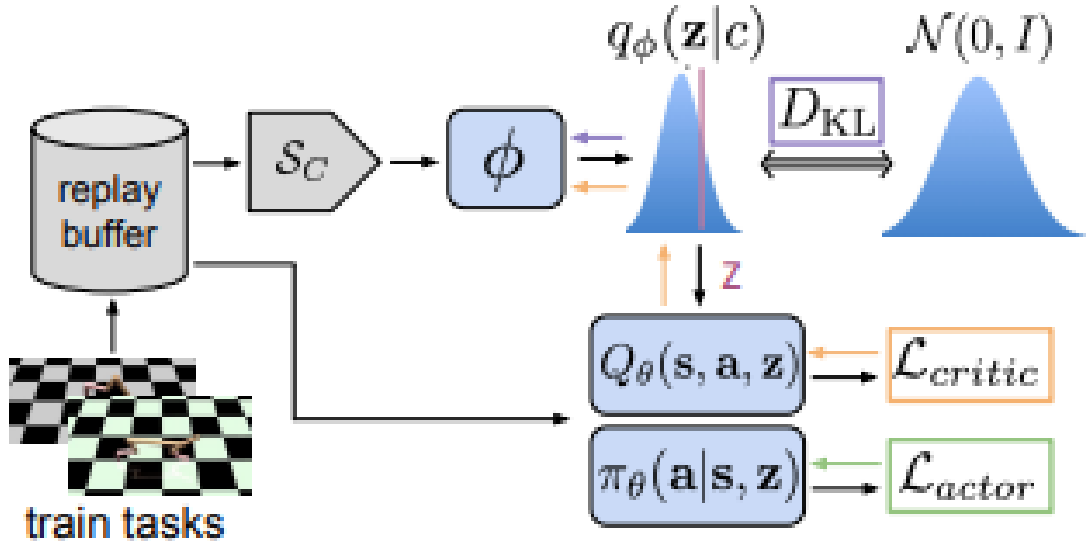


图 1: 方法示意图

#### 3.2 元学习

在这一块，他们首先将一个 task 分割为三个部分，初始的 state 分布，state transition 的分布以及奖励机制。在学习其中一个任务的工程中，不断积累 transition 的 history experience，这部分的积累，就是用来进行任务推理的 context。随后，利用 Gaussian factor 来对进行 encoding：

$$\Psi_\phi(\mathbf{z} | \mathbf{c}_n) = \mathcal{N}(f_\phi^\mu(\mathbf{c}_n), f_\phi^\sigma(\mathbf{c}_n))$$

其中的 function  $f_\phi$  为带参数  $\phi$  的 nn。在积累到  $N$  个 task 之后，可以利用一个 permutation-invariant function，来表示  $z$  与  $c$  的关系。即输入一个新的 context 集合，由样本 1 到  $N$  训练出来的 encoder  $q$ ，能生成 encoding 后的  $z$ 。

$$q_\phi(\mathbf{z} | \mathbf{c}_{1:N}) \propto \prod_{n=1}^N \Psi_\phi(\mathbf{z} | \mathbf{c}_n)$$

## 4 复现细节

### 4.1 与已有开源代码对比

复现过程参考了论文作者给出的源代码。在复现过程中发现，论文给出的源代码并不支持使用最新版的 `transflow` 框架, 导致最初无法运行，经过数日的排查，修改了部分代码，如今已可以在最新版的 `transflow` 框架上正常运行。

### 4.2 实验环境搭建

论文作者给出的源代码只在 Ubuntu 操作系统上做过测试, 所以首先需要能够在电脑上使用 Ubuntu 操作系统, 这里我采用的是 Windows 子系统 (wsl), Windows Subsystem for Linux (WSL) 是一个兼容层, 允许您在 Windows 上本地运行 Linux 二进制可执行文件。它允许您直接在 Windows 机器上使用 Linux 工具和应用程序, 而无需设置单独的 Linux 环境或双重引导到不同的操作系统。要使用 WSL, 首先需要在 Windows 中启用该功能, 然后从 Microsoft 商店安装受支持的 Linux 发行版。安装完成后, 您可以启动 Linux 发行版并像在本机 Linux 机器上一样使用它。您可以从 Windows 访问 Linux 文件系统, 也可以从 Linux 环境中访问 Windows 文件。WSL 对于需要在 Windows 机器上使用 Linux 工具和应用程序的开发人员, 或者对于不需要完全安装就想尝试 Linux 的用户, 都非常有用。它对于在 Windows 环境中测试和运行特定于 Linux 的软件也很有用。

另外我还需要安装 MuJoCo。MuJoCo (Multi-Joint dynamics with Contact) 是一个物理引擎, 用于模拟复杂、高度铰接的系统, 如机器人、生物力学系统和其他多体系统。它旨在实时模拟此类系统的行为, 允许用户与模拟交互、执行分析和运行各种类型的控制算法。MuJoCo 通常用于机器人、计算机图形学和机器学习研究, 并已应用于机器人控制、运动规划和强化学习等广泛的任務。它由马里兰州大学机器人和感知小组的埃莫·托多罗夫及其团队开发和维护。MuJoCo 是一款商业软件, 但有一个有限的版本可以免费用于非商业用途。

### 4.3 界面分析与使用说明

界面依托于 `mujoco210`, 可以通过 `python` 操纵其中的模型。

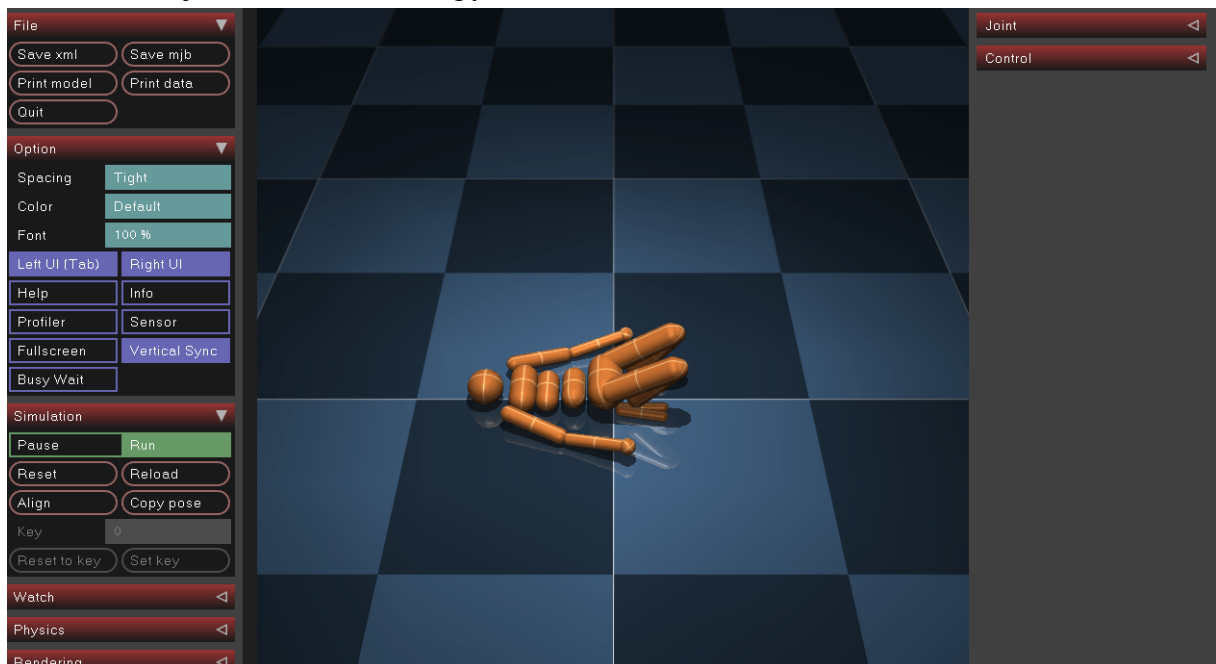


图 2: 操作界面示意

## 5 实验结果分析

在使用论文中的方法对半猎豹跑步模型进行训练后，与其他传统的方法对比结果，发现其收敛速度更快。

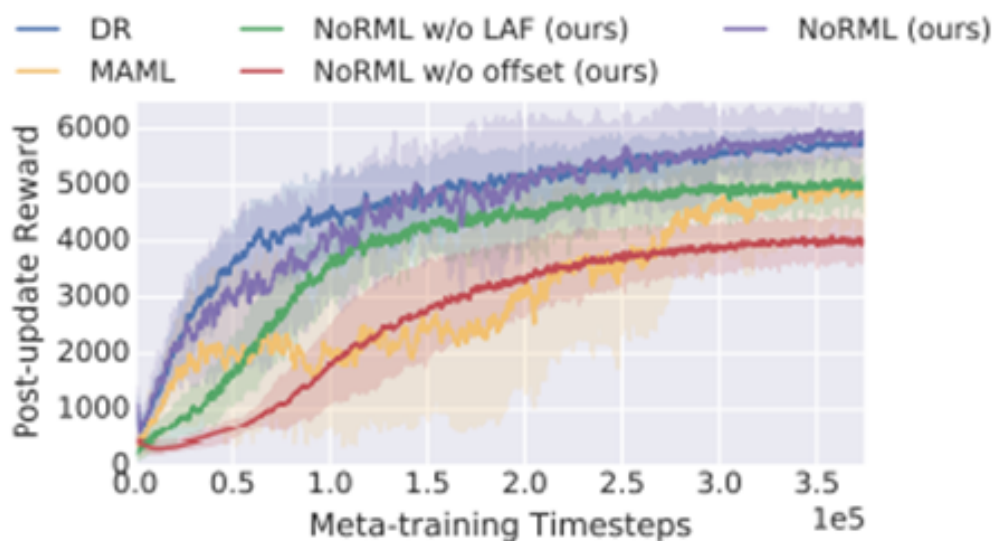


图 3: 实验结果示意

## 6 总结与展望

实验表明 NoRML 既有学习的优势，也有学习的抵消，可以适应所有这些类型的变化，即使奖励信号不是在适应过程中出现一个有前途的未来研究方向是将 NoRML 应用于将策略从模拟转移到真实机器人。模拟到实际传输是机器人学中一个具有挑战性的问题，这是由模型引起的模拟和真实世界物理之间的误差。我们可以将模型误差视为动态变化，并将 NoRML 应用于通过几张真实的机器人数据来适应这种变化。