

清华大学自然语言处理与社会人文计算实验室

学生科研工作及成果发表规范

2022 年 10 月

严谨遵守国际通行的学术规范是从事一切科研工作的底线要求，实事求是、诚实守信、尊重他人的研究成果是科研人员必须恪守的基本学术道德，是治学的起码要求，也是一辈子为人处世、安身立命需要遵循的根本准则。如果在这方面出了问题，对学生、导师和单位的学术声誉将构成严重的损害，对个人的名誉乃至个人今后的长远发展也将带来严重的负面影响。本实验室全体师生必须对此问题的严肃性始终保持高度的警惕，防患于未然，常抓不懈。为彻底杜绝此类问题的可能发生，在学校、院系、实验室多个层面常态化进行学术规范和学术道德教育以及师生自觉自律的基础上，现重申有关纪律，并制定相应措施，要求同学们认真体会并以一种“如临深渊、如履薄冰”的态度切实严格遵守之。

科研规范

根据《[清华大学预防与处理学术不端行为办法](#)》精神，本实验室学生开展科研工作的全过程中严禁出现以下行为：

1. 剽窃、抄袭、侵占他人学术成果；
2. 篡改他人研究成果；
3. 伪造科研数据、资料、文献、注释，或者捏造事实、编造虚假研究成果；
4. 未参加研究或创作而在研究成果、学术论文上署名，未经他人许可而不当使用他人署名，虚构合作者共同署名，或者多人共同完成研究而在成果中未注明他人工作、贡献；
5. 在申报奖励（如奖学金）、申请学位等过程中提供虚假学术信息；
6. 买卖论文、由他人代写或者为他人代写论文；
7. 其他根据学校或者有关学术组织、相关科研管理机构制定的规则，属于学术不端的行为。

对出现以上行为者，将按照《[清华大学预防与处理学术不端行为办法](#)》及《[清华大学学生纪律处分管理规定](#)》进行严肃处置。同时，[The IEEE PSPB Operations Manual](#) 第 8.2 节（即《IEEE 论文抄袭、剽窃判定准则》）的相关要点，也是必须予以重视并引以为戒的参

照系。

成果发表自查及检查规范

1. **论文自查**：论文（含其他类型文章）投稿截止前，论文第一作者和第一学生作者需对照自查清单进行论文检查，签字交给**本实验室科研助理**（现为王宇星）确认后，方可提交论文。

提示：需提前两周，将论文初稿发送给导师、高年级研究生进行审阅；需提前使用 grammarly 等写作辅助工具，进行论文语法检查，避免论文中出现低级错误。

2. **论文查重**：录用论文（含其他类型文章）最终版正式提交前，发本实验室科研助理利用软件进行查重，根据重复率按以下三种情况处理：

- a. 若查重率高于 10%，则本次投稿直接终止，并按可能存在学术不端问题核查；
- b. 若查重率高于 5%且小于 10%，需认真修改并交导师确认合格后方可提交；
- c. 若查重率不高于 5%，对重复处进行认真修改后可直接提交。

3. **代码复现**：录用论文最终版正式提交前一周，论文第一作者和第一学生作者按照“代码开源规范”整理代码及文档，并提交**本实验室科研助理**（现为许诚强）审核，实验室安排抽查主要成果，科研助理参照文档进行结果复现，应确保复现结果与论文中表格保持一致，复现结果确认一致后方可提交。

成果发表自查清单

- 论文中不存在《清华大学预防与处理学术不端行为办法》中列明的学术不端行为。
- 论文投稿、署名排序等已获得所有作者（首先是导师）知情并同意。
- 论文已申请并通过保密审查，或已确认不涉及保密审查问题。
- 论文实验纪录清晰，代码和相关结果保存完整，可追溯所有论文报告结果的原始信息。

论文写作规范

1. 我组论文严禁出现《[IEEE 论文抄袭、剽窃判定准则](#)》所规定的如下任何一个等级的抄袭现象：

- a. 等级 1：在未引用的情况下，逐字逐句地抄袭他人的整篇文章；抄袭文章的大部分（比例超过 50%）；抄袭自己之前发表过的作品（比例超过 50%）；
- b. 等级 2：在未引用的情况下，大量抄袭他人的作品（抄袭比例达 20%-50%）；抄袭自己之前发表过的作品（抄袭比例达 20%-50%）；
- c. 等级 3：在未引用的情况下，抄袭的段落、句子或插图等独立元素构成文章的明显

篇幅（抄袭比例不超过 20%）；

d. 等级 4：在未引用的情况下，成段或成篇改写出自其他来源的资料，包括但不限于：修改了部分短语，或对句子顺序进行了修改；

e. 等级 5：虽引用了但在没有标注明确标识符（如加引号或缩进）的情况下，逐字逐句抄袭其他文章。

2. 论文写作中若干需特别关切（容易认识不清或出现疏漏）的禁止事项

a. 在任何情况下（包括在撰写引言、综述或类似内容时），决不允许以拷贝或在拷贝基础上略加文字编辑的方式，将他人论文中的文字放入自己的论文中。本人论文中的每一个文字必须经过自己思考斟酌后独立写出。否则属于科研规范之严禁情形 1 中的“抄袭、剽窃”行为。

b. 在任何情况下，如果直接引用某个相关研究工作的部分原文文字，在明确给出对应参考文献的前提下，还必须在论文中予以明确阐明并以引号将全部直接引用文字明确标识出来（即使在此种条件下，被直接引用文字的长度也应尽量短）。否则属于科研规范之严禁情形 1 中的“抄袭、剽窃”行为。

c. 上述禁止事项（a）和（b）存在一个类似的情形：即作者引用自己以前正式发表过的论文。此时需要特别小心谨慎：必须按照同样的原则处理。否则会被视作“自我抄袭、剽窃”行为。

d. 未经出版机构或版权所有人版权许可，禁止复用他人或自己的已发表论文和书籍中包括图、表在内的内容。

e. 在学术会议政策允许的前提下，可将学术会议上已经发表的论文工作在显著扩展、完善后投稿给学术期刊，但该论文必须有足够的新增实质性研究内容以及足够的新增篇幅。按照 [ACM 有关条例](#)，这通常意味着至少有 25% 的论文内容是以前没有发表过的材料，具体比例约定可能因出版物而异。并且在投稿前必须将已在学术会议上发表的情况向期刊编辑部明确说明，在得到编辑部书面允许及相关具体出版指引的条件下，方可投稿。此时仍需满足禁止事项（d）。

开源规范

1. 参考样例：可参考 <https://github.com/thunlp/PTR>

2. 基础代码组织：

a. `src`：放置源码；

b. `scripts`：放置运行脚本；

c. `data`：放置数据下载脚本；

d. `figs`：放置 README 中的图片；

- e. `requirements.txt` : python 环境, 需确保重新创建 python 环境后, 通过 `pip install -r requirements.txt` 可以正常运行源码 ;
 - f. `LICENSE` : 一般选 MIT 协议 ;
 - g. `README.md` : 具体内容见下文 ;
 - h. 在此基础上, 可视具体情况增加其他文件夹放置其他内容, 例如增加 `tools` 文件夹放置数据预处理相关代码、`demo` 文件夹放置 `demo` 等。
3. README 规范, 应该至少包含以下内容 :
- a. `论文 Title`, 提供该仓库对应的论文链接 ;
 - b. `Citation`, 给出引用格式 ;
 - c. `Quick Links` : 提供到下面各个部分的跳转链接 ;
 - d. `Overview` : 简单介绍该工作, 并附上模型/方法示意图 ;
 - e. `Requirements / Environment Setup` : 阐明环境搭建的流程。开源前确保按照该章节的说明能够正常搭建代码运行环境 ;
 - f. `Data Preparation` : 给出下载数据的脚本运行方式 ;
 - g. `Experiments` : 提供复现论文中结果的脚本运行方式。确保脚本使用的超参数能够确实跑出论文中的结果。如论文中有多个 `Table`, 可逐个 `Table` 列举复现方式 ;
 - h. 在此基础上, 可视具体情况增加内容。例如在 `Experiments` 中, 除了提供脚本运行方式, 同时解释其中部分重要参数的含义等。

代码规范

1. 文件注释 : 文件注释应当写在每个文件的开头, 并至少包含如下部分。
 - a. 文件内容简述 ;
 - b. 作者 ;
 - c. 时间 ;
 - d. `copyright` ;

例子 :

```
Python
# This file is .....
# Author: XXX, YYY, ZZZ, ...
# Date: 20xx-yy
```

```
# Copyright (c) THUNLP, Tsinghua University. All rights reserved.  
# See LICENSE file in the project root for license information.
```

1. 使用其他开源项目时，复制其他开源项目代码，应遵照对应项目开源协议要求；如有 copyright，应一并复制。

例子：

```
Python  
# This file comes from .....  
# Copyright (c) XXXX （将原开源代码的 copyright 复制过来）
```

1. 注释规范

- a. 如无特殊情况，所有注释应使用英文书写，且通过 `grammarly` 检查。
- b. 建议对核心的类、函数等内容辅以注释，位于类或函数的代码前后，阐明功能及使用方法。
- c. 建议对核心的代码实现辅以注释，辅助读者在通读论文后能够理解代码的核心运行逻辑。

例子：

```
Python  
  
# 此处填写功能解释  
class Example:  
    """  
        此处填写长段的功能解释  
    """  
  
    # 此处填写功能解释  
    def func1(...):  
        ...  
  
    def func2(...):  
        """ 此处填写长段的功能解释  
        Args:  
            此处填写参数注释  
        Return:  
            此处填写输出注释  
        """  
        ...
```