

LaTeX 零基础教程

从第一段文字到完整学术论文

面向完全没有接触过 LaTeX 的中文学习者

这份教程不要求你先记术语。你只需要按照“如果想做某件事，就这样写”的方式边看边试。教程默认使用 XeLaTeX，示例已经在 macOS、VS Code 和 LaTeX Workshop 环境中实际编译。

编写日期：2026 年 7 月 6 日

开始之前：你只需要知道三件事

1. 你写的是一个以 `.tex` 结尾的普通文本文件。
2. 保存文件后，XeLaTeX 会把它“翻译”成 PDF。
3. 大多数时候，左边改代码，右边看 PDF；出错就看日志中最先出现的错误。

先记住这一句

不要试图一次背完所有命令。写论文时最常用的命令其实很少：章节、公式、图片、表格、引用和参考文献。忘了就回来查，这完全正常。

目录

开始之前	i
第一章 LaTeX 是什么	1
1.1 LaTeX 和 Word 有什么区别	1
1.2 为什么它适合写论文	1
1.3 LaTeX 的基本思想	2
1.4 什么是 tex、编译和 PDF	2
1.5 新手应该用什么编译方式	2
第二章 最小可运行示例：先成功一次	3
2.1 复制这份完整代码	3
2.2 逐行解释	3
2.3 在 VS Code 中新建和打开项目	4
2.4 编译和预览	4
2.5 LaTeX Workshop 侧栏有什么用	5
2.6 源代码和 PDF 互相定位	7
第三章 LaTeX 文档的基本结构	8
3.1 documentclass 是什么	8
3.2 usepackage 是什么	8
3.3 begin 和 end 是什么	9
3.4 一个带常用设置的完整结构	9
第四章 正文和段落怎么写	11
4.1 普通段落	11
4.2 换段	11
4.3 手动换行	11

4.4	多留一点垂直空白	12
4.5	中文段落注意事项	12
4.6	英文段落注意事项	12
第五章	标题、章节、摘要和目录	13
5.1	文章标题、作者和日期	13
5.2	一级、二级、三级标题	13
5.3	摘要和关键词	14
5.4	自动目录	14
第六章	文字样式、颜色和对齐	16
6.1	加粗、斜体、下划线和删除线	16
6.2	改变字体大小	17
6.3	改变文字颜色	17
6.4	居中、左对齐和右对齐	18
第七章	列表怎么写	20
7.1	无序列表	20
7.2	有序列表	20
7.3	多级列表	21
7.4	在列表中使用样式、公式和引用	21
第八章	数学公式：从最常用写法开始	22
8.1	行内公式	22
8.2	不带编号的单行公式	22
8.3	带编号的公式	23
8.4	引用某个公式	23
8.5	多行并对齐的公式	23
8.6	分数	24
8.7	上标和下标	25
8.8	根号	25
8.9	希腊字母	26
8.10	求和、积分和极限	26
8.11	矩阵	27
8.12	分段函数	27

8.13 公式里的中文说明	28
8.14 括号随内容自动变大	28
第九章 插入图片	29
9.1 图片文件放在哪里	29
9.2 一个完整图片例子	29
9.3 逐行解释	30
9.4 在正文中引用图片	30
9.5 控制图片大小	30
9.6 图片位置为什么会“乱跑”	31
第十章 表格：尤其是论文常用三线表	32
10.1 最简单的表格	32
10.2 每一列如何对齐	32
10.3 加横线和竖线	33
10.4 带标题、编号和引用的完整表格	33
10.5 论文常用三线表	33
10.6 表格太宽怎么办	34
10.7 合并多列和多行	35
第十一章 交叉引用：让编号永远自动更新	37
11.1 label 和 ref 是什么	37
11.2 标签为什么要加前缀	37
11.3 引用章节	38
11.4 引用图片	38
11.5 引用表格	38
11.6 引用公式	39
第十二章 参考文献：把引用集中管理	40
12.1 参考文献到底是什么	40
12.2 BibTeX、Biber 和 bib 文件	40
12.3 新建 references.bib	40
12.4 在 main.tex 中连接 bib 文件	41
12.5 正文引用和文末列表	41
12.6 最小可运行参考文献项目	42

12.7 常见引用格式	43
12.8 从网站复制 BibTeX	44
12.9 中文和英文文献有什么区别	44
第十三章 在论文中插入代码	45
13.1 准备 listings 样式	45
13.2 插入简单 Python 代码	46
13.3 给代码加标题、编号和标签	46
13.4 从外部文件读取代码	46
第十四章 算法伪代码	48
14.1 机器学习训练流程示例	48
14.2 逐项理解算法命令	49
第十五章 页面格式、行距、缩进和字体	51
15.1 纸张大小和页边距	51
15.2 行距	51
15.3 段落缩进和段间距	52
15.4 页眉、页脚和页码	53
15.5 基础字体大小	53
15.6 中文字体	53
15.7 英文和数字字体	54
15.8 适合中文课程报告的常用配置	55
第十六章 封面、中文摘要、英文摘要和目录	56
16.1 一个简单但完整的封面	56
16.2 逐块理解封面	57
16.3 中文摘要	57
16.4 英文摘要	58
16.5 目录	58
16.6 论文前几页的组合顺序	59
第十七章 附录	60
17.1 附录是什么	60
17.2 开始附录	60
17.3 附录中的图片、表格和代码	61

第十八章 一份可以直接使用的通用论文模板	62
18.1 模板包含什么	62
18.2 如何开始使用模板	62
18.3 完整 main.tex	63
18.4 完整 references.bib	69
第十九章 常见报错和解决方法	70
19.1 忘记写 end document	70
19.2 中文无法显示	70
19.3 图片找不到	71
19.4 参考文献不显示	71
19.5 公式报错	72
19.6 表格太宽	72
19.7 编译后 PDF 没变化	72
19.8 VS Code 预览打不开	73
19.9 找不到 xelatex	73
19.10 找不到 latexmk	74
19.11 一个通用排错顺序	74
第二十章 零基础一周学习路线	75
20.1 第一天：只追求成功生成 PDF	75
20.2 第二天：公式、图片和表格	75
20.3 第三天：目录和参考文献	76
20.4 第四到第七天	76
20.5 写论文必须掌握的内容	76
20.6 可以暂时不学的内容	77
20.7 写作时最实用的工作习惯	77
最后的命令速查表	78

第一章 LaTeX 是什么

1.1 LaTeX 和 Word 有什么区别

用大白话说，Word 更像“直接在成品纸上排版”：你选中文字、点加粗、拖图片、调字号，屏幕上看到的基本就是最终效果。

LaTeX 更像“把排版要求写成说明书”：你写“这里是标题”“这里是一条公式”“这里放一张图片”，然后让编译器统一完成排版。

例如，在 Word 里你会选中文字再点“加粗”；在 LaTeX 里你会写：

```
\textbf{这几个字要加粗}
```

这段代码怎么理解：

`\textbf` 是“把文字变粗”的命令；后面大括号里的内容是这个命令要处理的文字。

1.2 为什么它适合写论文

LaTeX 特别适合论文，不是因为它看起来“高深”，而是因为它能替你处理很多容易乱的事情：

- 公式大小、上下标和编号比较自然；
- 章节、图片、表格和公式可以自动编号；
- “见图 3”“见式 (5)”这类引用会自动更新；
- 参考文献可以集中管理，不用手工调整编号；
- 文档很长时，标题、目录、页码和格式仍然能保持一致；
- 源文件是纯文本，方便 Git 管理和比较修改。

1.3 LaTeX 的基本思想

它的基本思想可以浓缩成一句话：你说明“这是什么”，LaTeX 决定“它应该长什么样”。

例如：

```
\section{实验结果}
```

你告诉 LaTeX：“实验结果”是一个章节标题。它会根据文档整体规则决定字号、粗细、编号和上下间距。这样你不用每次手动把标题调成完全一样。

1.4 什么是 tex、编译和 PDF

.tex 文件

你真正编辑的源文件。它同时包含正文和排版命令。

编译

把 .tex 源文件转换成排版结果的过程。你可以把它理解成“生成 PDF”。

PDF

最终阅读、打印或提交的文件。一般不要直接修改 PDF，而是修改 .tex 后重新编译。

1.5 新手应该用什么编译方式

中文写作默认使用 **XeLaTeX**。它能直接处理 UTF-8 中文，也能使用电脑中的现代字体。

实际操作时，我们使用 **latexmk + XeLaTeX**：XeLaTeX 负责排版；latexmk 负责判断要编译几次，以及是否需要处理目录、引用和参考文献。

先记住这一句

你不需要在 XeLaTeX、pdfLaTeX、LuaLaTeX 之间反复纠结。本教程中的中文项目统一选择“latexmk (XeLaTeX)”。

第二章 最小可运行示例：先成功一次

2.1 复制这份完整代码

它的作用：

下面是一个真正完整、可以生成中文 PDF 的文件。请新建 `hello.tex`，完整复制进去并保存。

```
1 \documentclass[UTF8,fontset=fandol,12pt]{ctexart}
2
3 \title{我的第一份 LaTeX 文档}
4 \author{你的名字}
5 \date{\today}
6
7 \begin{document}
8
9 \maketitle
10
11 你好，LaTeX!
12
13 这是我写的第一段正文。
14
15 \end{document}
```

2.2 逐行解释

1. 第 1 行的 `\documentclass` 决定整份文件采用什么基础类型。`ctexart` 表示“适合中文文章的文档类型”。`UTF8` 表示源文件使用 UTF-8；`fontset=fandol` 使用 TeX 自带的中文字体；`12pt` 是正文基础字号。
2. 第 3 行的 `\title` 保存文章标题。此时只是登记标题，还没有打印出来。

3. 第 4 行的 `\author` 保存作者名字。
4. 第 5 行的 `\date{\today}` 使用当天日期。如果不想显示日期，可以写 `\date{}`。
5. 第 7 行的 `\begin{document}` 表示“正文从这里开始”。
6. 第 9 行的 `\maketitle` 把刚才登记的标题、作者和日期真正排到页面上。
7. 第 11、13 行就是普通正文，直接写即可。
8. 第 15 行的 `\end{document}` 表示“整份正文到这里结束”。后面的内容不会进入 PDF。

容易踩坑

LaTeX 命令通常以反斜杠 `\` 开头。大括号 `{ }` 里通常放“这个命令要处理的内容”。少一个大括号、把反斜杠写成普通斜杠，都会导致报错。

2.3 在 VS Code 中新建和打开项目

如果你想创建自己的第一个项目，就这样做：

1. 在“文稿”目录新建一个文件夹，例如 `my-first-paper`。
2. 打开 VS Code，选择“文件 → 打开文件夹”。
3. 选择刚才的文件夹。不要只打开一个孤零零的 `.tex` 文件。
4. 在左侧资源管理器中新建 `main.tex`。
5. 复制最小示例，按 `Command + S` 保存。

图2.1显示了真实的编辑界面。左侧是项目文件，中间是正在编辑的 `test.tex`，右上角的播放形状按钮用于构建，旁边的预览按钮用于打开 PDF。

2.4 编译和预览

如果你想手动编译当前文档，可以使用下面任一种方法：

- 点击编辑器右上角的“构建 LaTeX 项目”按钮；
- 按快捷键 `Command + Option + B`；

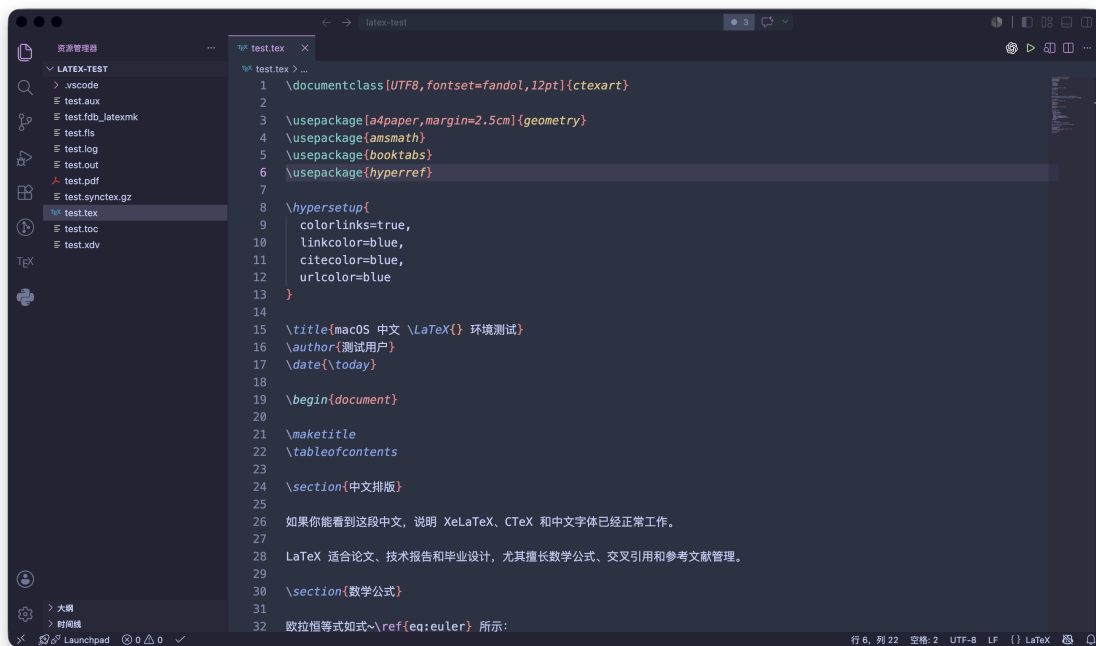


图 2.1: 在 VS Code 中编辑 LaTeX 源文件

- 打开命令面板，搜索 LaTeX Workshop: Build LaTeX project。

我们已经把环境配置为“保存时自动编译”，所以大多数时候你只要按 Command + S。

如果你想打开 PDF 预览，可以：

- 点击右上角的“查看 LaTeX PDF 文件”按钮；
- 按 Command + Option + V。

图2.2是最实用的工作方式：左边写源代码，右边看 PDF。保存后，右边会自动刷新。

2.5 LaTeX Workshop 侧栏有什么用

点击 VS Code 左边活动栏中的 \TeX 图标，会打开图2.3所示的侧栏。这里有：

- 构建项目、查看 PDF、查看日志；
- 当前文档的章节结构；
- 已经写过的 label；
- 可以点击插入的数学符号。



图 2.2: 源代码与 PDF 左右对照



图 2.3: LaTeX Workshop 的命令、结构和数学符号侧栏

2.6 源代码和 PDF 互相定位

如果你想从源代码跳到 PDF 对应位置，先把光标放在那一行，再按快捷键 `Command + Option + J`。

如果你想从 PDF 跳回源代码，在内置 PDF 预览中按住 `Control` 并单击对应文字。

这个功能叫 SyncTeX。名字可以暂时忘掉，只要记住它能帮助你在长论文中快速找位置。

第三章 LaTeX 文档的基本结构

一份 LaTeX 文档可以想成两部分：

1. `\begin{document}` 之前是“准备区”：选择文档类型、加载功能包、设置页面和字体。
2. `\begin{document}` 与 `\end{document}` 之间是“正文区”：标题、摘要、章节、公式、图片、表格等实际内容。

3.1 documentclass 是什么

它的作用：

它决定整份文档的基础类型，通常是第一条命令。

如果你想写普通文章、课程报告或短论文，可以这样写：

```
\documentclass[UTF8,fontset=fandol,12pt]{ctexart}
```

如果你想写比较长、需要“章”的毕业设计，可以这样写：

```
\documentclass[UTF8,fontset=fandol,12pt,openany]{ctexbook}
```

这段代码怎么理解：

`ctexart` 适合“文章”，最高常用层级是 `\section`；`ctexbook` 适合“书或长论文”，可以使用 `\chapter`。学校提供模板时，应优先使用学校指定的文档类，不要擅自替换。

3.2 usepackage 是什么

它的作用：

它相当于“给 LaTeX 加装功能”。基础 LaTeX 不会默认打开所有能力，需要什么就加载什么。

如果你想插入图片，就加载 `graphicx`：

```
\usepackage{graphicx}
```

如果你想写更丰富的数学公式，就加载 `amsmath`：

```
\usepackage{amsmath}
```

如果你想改变文字颜色，就加载 `xcolor`：

```
\usepackage{xcolor}
```

这段代码怎么理解：

宏包名写在大括号里。`\usepackage` 必须放在 `\begin{document}` 之前，不能写到正文中间。

3.3 begin 和 end 是什么

很多功能有一个开始和一个结束：

```
\begin{document}
这里是正文。
\end{document}
```

`document` 是环境名。开始和结束的名字必须完全相同。列表、表格、图片、摘要、公式也常采用这种写法。

3.4 一个带常用设置的完整结构

```
1 \documentclass[UTF8,fontset=fandol,12pt]{ctexart}
2
```

```
3 % 下面这些是正文开始前的准备
4 \usepackage[a4paper,margin=2.5cm]{geometry}
5 \usepackage{amsmath}
6 \usepackage{graphicx}
7 \usepackage{xcolor}
8 \usepackage{hyperref}
9
10 \title{课程报告标题}
11 \author{你的名字}
12 \date{\today}
13
14 \begin{document}
15
16 \maketitle
17
18 \section{引言}
19 这里开始写正文。
20
21 \end{document}
```

百分号 % 后面的内容是注释。注释只给写作者看，不会出现在 PDF 中。

第四章 正文和段落怎么写

4.1 普通段落

如果你想写一个普通段落，直接写文字就行：

这是第一段文字。LaTeX 会根据页面宽度自动换行，所以你不需要在每一行末尾做任何处理。

这段代码怎么理解：

源文件里的一次普通回车，通常只相当于一个空格，不代表换段。LaTeX 会自己决定一行在哪里结束。

4.2 换段

如果你想开始新段落，在两段之间留一个空白行：

这是第一段文字。

这是第二段文字。

这段代码怎么理解：

中间的空白行告诉 LaTeX：“上一段结束了，下面是新段。”这也是最推荐的换段方法。

4.3 手动换行

如果你想不开始新段，只想在当前位置强制换到下一行，可以写：

```
第一行文字。\\  
第二行文字。
```

两个连续反斜杠 `\\` 表示强制换行。它适合地址、短诗、封面信息等固定格式，不建议在普通正文每行都使用。

4.4 多留一点竖直空白

如果你想在两块内容之间额外留出一行左右的空间，可以写：

```
上面的内容。  
  
\vspace{\baselineskip}  
  
下面的内容。
```

`\baselineskip` 大致表示当前一行文字的高度。如果只想正常换段，不要使用 `\vspace`，空一行就够了。

4.5 中文段落注意事项

- 源文件保存成 UTF-8。VS Code 右下角应显示 UTF-8。
- 使用 XeLaTeX 编译，文档类优先选 `ctexart` 或 `ctexbook`。
- 中文之间是否在源文件中换行，一般不影响最终排版。
- 中文段首缩进可以用 `\setlength{\parindent}{2em}` 统一设置，不要手工输入许多空格。

4.6 英文段落注意事项

英文单词之间要保留普通空格。源文件中连续多个普通空格通常会被合并成一个；如果确实需要不可断开的空格，可以使用波浪号，例如 `Figure~1`，这样 `Figure` 和 `1` 不容易被拆到两行。

```
As shown in Figure~1, the proposed method performs well.
```

第五章 标题、章节、摘要和目录

5.1 文章标题、作者和日期

如果你想生成文章标题页头，可以这样写：

```
\title{基于深度学习的图像分类研究}  
\author{张三}  
\date{\today}  
  
\begin{document}  
\maketitle  
...  
\end{document}
```

这段代码怎么理解：

`\title`、`\author`、`\date` 负责保存信息，`\maketitle` 才会把它们排出来。想隐藏日期就写 `\date{}`。

5.2 一级、二级、三级标题

如果你想写一级标题，可以这样写：

```
\section{实验方法}
```

如果你想在一级标题下面再分一层，可以这样写：

```
\subsection{数据预处理}
```

如果你想再细分一层，可以这样写：

```
\subsubsection{图像归一化}
```

这段代码怎么理解：

`section`、`subsection`、`subsubsection` 分别是一级、二级、三级标题。编号由 LaTeX 自动生成，不要自己写“1.1”。使用 `ctexbook` 时，还可以在最上层使用 `\chapter{方法}`。

5.3 摘要和关键词

如果你想写中文摘要，可以这样写：

```
\begin{abstract}
本文研究了……问题，提出了……方法。
实验结果表明……

\noindent\textbf{关键词：} 深度学习；图像分类；卷积神经网络
\end{abstract}
```

这段代码怎么理解：

`abstract` 环境会排出摘要区域。`\noindent` 取消这一行的段首缩进，`\textbf` 把“关键词”加粗。中文关键词常用分号隔开，具体格式听学校或期刊要求。

5.4 自动目录

如果你想自动生成目录，在希望出现目录的位置写：

```
\tableofcontents
```

目录会读取所有章节标题及页码。第一次编译时目录可能为空或页码不对；`latexmk` 会自动多编译几次。手动编译时需要重复运行 `XeLaTeX`，直到目录稳定。

新手提示

目录和交叉引用“第一次不对、第二次才对”并不是故障。第一次编译先收集信息，下一次才能把信息排进 PDF。使用 `latexmk` 就不用自己数编译次数。

第六章 文字样式、颜色和对齐

本章的原则是“局部修改、范围清楚”。你只想改变几个字，就把那几个字放进大括号；你想改变一整段，就使用带开始和结束的环境。

6.1 加粗、斜体、下划线和删除线

如果你想让几个字加粗，就这样写：

```
这是普通文字，\textbf{这几个字是粗体}。
```

`\textbf` 可以理解成 `text bold face`。大括号界定了加粗范围。

如果你想让几个字变成斜体，就这样写：

```
This is normal, and \textit{these words are italic}.
```

`\textit` 对英文斜体效果明显。中文字体未必有真正的斜体字形，因此中文论文通常少用斜体强调。

如果你想给文字加下划线，就这样写：

```
\underline{这句话带有下划线}
```

如果你想给文字加删除线，就这样写：

```
\sout{这句话被删除了}
```

`\underline` 和 `\sout` 来自 `ulem` 宏包，所以准备区需要：

```
\usepackage[normalem]{ulem}
```

6.2 改变字体大小

如果你想只把一小段文字放大，可以这样写：

```
普通大小, {\large 这部分稍大}, 又回到普通大小。
```

外层大括号把 `\large` 的影响限制在这一小段。常见大小从小到大包括：

```
{\tiny 最小}
{\scriptsize 很小}
{\footnotesize 脚注大小}
{\small 稍小}
{\normalsize 正常}
{\large 稍大}
{\Large 更大}
{\LARGE 很大}
{\huge 非常大}
{\Huge 最大}
```

中文文档类还提供字号命令，例如 `\zihao{4}` 表示四号，`\zihao{-4}` 表示小四。

```
{\zihao{4} 这是四号字}
{\zihao{-4} 这是小四号字}
```

容易踩坑

论文正文的字号通常由学校模板统一规定。不要为了“看起来顺眼”在正文里到处手工改字号；局部字号更适合封面、说明文字或特殊展示。

6.3 改变文字颜色

先在准备区加载颜色功能：

```
\usepackage{xcolor}
```

如果你想让几个字变成红色，可以这样写：

```
\textcolor{red}{这几个字是红色的}
```

这段代码怎么理解：

red 是颜色名；第二个大括号里是要变色的文字。常见颜色名还有 blue、green、orange、purple、gray。

如果你想自定义一个更柔和的颜色，可以这样写：

```
\definecolor{myblue}{HTML}{1F5AA6}  
  
\textcolor{myblue}{这是自定义蓝色}
```

HTML 后面是六位十六进制颜色值，写法与网页颜色相同。

如果你想让一整段都变成蓝色，可以把颜色命令限制在一个组里：

```
{\color{blue}  
这一整段文字都是蓝色。  
  
即使中间换段，颜色仍然有效。  
}
```

最后的右大括号结束颜色范围。论文正文一般保持黑色；颜色更适合提示、链接或图表。

6.4 居中、左对齐和右对齐

如果你想让一段内容居中，可以这样写：

```
\begin{center}  
这段文字居中。  
\end{center}
```

如果你想让一段内容靠左，可以这样写：

```
\begin{flushleft}  
这段文字左对齐。  
\end{flushleft}
```

如果你想让一段内容靠右，可以这样写：

```
\begin{flushright}  
这段文字右对齐。  
\end{flushright}
```

这些都是环境，所以必须有匹配的 `\begin` 和 `\end`。

第七章 列表怎么写

7.1 无序列表

它的作用：

无序列表适合列举没有先后顺序的要点，默认使用圆点。

如果你想写一个无序列表，就这样写：

```
\begin{itemize}
  \item 第一项内容
  \item 第二项内容
  \item 第三项内容
\end{itemize}
```

这段代码怎么理解：

`itemize` 表示无序列表；每一个 `\item` 开始一项。缩进不是必须的，但适当缩进能让源文件更好读。

7.2 有序列表

如果你想让列表自动显示 1、2、3，就使用 `enumerate`：

```
\begin{enumerate}
  \item 收集数据
  \item 训练模型
  \item 评估结果
\end{enumerate}
```

不要自己手工输入编号，否则插入新项目时要全部重排。

7.3 多级列表

如果你想在一個列表項下面再放子列表，可以嵌套環境：

```
\begin{enumerate}
  \item 数据准备
    \begin{itemize}
      \item 清洗数据
      \item 划分训练集和测试集
    \end{itemize}
  \item 模型训练
    \begin{itemize}
      \item 设置学习率
      \item 记录损失函数
    \end{itemize}
\end{enumerate}
```

外层是有序列表，内层是无序列表。LaTeX 会自动改变缩进和项目符号。

7.4 在列表中使用样式、公式和引用

```
\begin{itemize}
  \item \textbf{主要目标:} 提高分类准确率。
  \item 损失函数记为  $L(\theta)$ 。
  \item 模型结构见图 ~\ref{fig:model}。
  \item 参考已有工作 ~\cite{lecun1998}。
\end{itemize}
```

列表項內部和普通正文一樣，可以加粗、寫行內公式、引用圖片和引用文獻。

第八章 数学公式：从最常用写法开始

数学公式是 LaTeX 的强项。你不需要先理解所有数学排版规则，只要先分清两类：夹在句子里的行内公式，以及单独占一行的展示公式。

准备区建议加载：

```
\usepackage{amsmath,amssymb,mathtools}
```

8.1 行内公式

如果你想在一句话中插入公式，用一对美元符号包住：

```
勾股定理可以写成  $a^2+b^2=c^2$ 。
```

两个美元符号之间进入“数学模式”。普通中文仍写在美元符号外面。

8.2 不带编号的单行公式

如果你想让公式单独占一行，但不需要编号，可以这样写：

```
\[  
  a^2+b^2=c^2  
\]
```

$\[$ 表示公式开始， $\]$ 表示公式结束。这比使用双美元符号更规范。

8.3 带编号的公式

如果你想让公式自动编号，可以使用 `equation` 环境：

```
\begin{equation}
  a^2+b^2=c^2
\end{equation}
```

LaTeX 会在公式右侧生成编号，不要自己手打“(1)”。

8.4 引用某个公式

如果你想以后在正文中写“见式 (1)”，先给公式贴标签：

```
\begin{equation}
  a^2+b^2=c^2
  \label{eq:pythagorean}
\end{equation}
```

由式 `\eqref{eq:pythagorean}` 可知，直角三角形三边满足上述关系。

`\label` 给公式一个内部名字；`\eqref` 显示带圆括号的公式编号。标签名建议以 `eq:` 开头，后面写有意义的英文短名。

8.5 多行并对齐的公式

如果你想写多行推导，并让等号对齐，可以使用 `align`：

```
\begin{align}
  (a+b)^2
  &= (a+b)(a+b) \\
  &= a^2+2ab+b^2.
\end{align}
```

这段代码怎么理解：

`&` 是对齐标记，通常放在等号前；`\\` 表示这一行结束。`align` 默认给每一行编号。

如果你想多行对齐但完全不要编号，把环境名改成 `align*`：

```
\begin{align*}
  y &= 2x+1, \\
  z &= 3x-4.
\end{align*}
```

环境名后面的星号通常表示“不编号”。

如果你想只让其中某几行不编号，可以这样写：

```
\begin{align}
  x+y &= 10, \notag\\
  x-y &= 2.
\end{align}
```

8.6 分数

如果你想写三分之二，可以这样写：

```
\[
  \frac{2}{3}
\]
```

第一个大括号是分子，第二个大括号是分母。复杂内容也可以放进大括号：

```
\[
  \frac{x+1}{x^2-1}
\]
```

8.7 上标和下标

如果你想写平方，也就是给变量加上标：

```
$x^2$
```

如果你想上标有多个字符，要用大括号包起来：

```
$x^{n+1}$
```

如果你想给变量写下标：

```
$x_1$, $a_{i,j}$, $L_{train}$
```

单个字符可以不加大括号；多个字符必须加。源文件中普通下划线有特殊含义，正文若真要显示下划线，应写 `_` 。

8.8 根号

如果你想写平方根，可以这样写：

```
$$\sqrt{x}$$
```

如果你想写三次根号，把次数写在方括号里：

```
$$\sqrt[3]{x}$$
```

方括号通常表示可选设置，大括号通常表示必须提供的内容。

8.9 希腊字母

如果你想写常见希腊字母，可以使用它们的英文名称：

```
\alpha$  $\beta$  $\gamma$  $\delta$  $\epsilon$
$\theta$  $\lambda$  $\mu$  $\sigma$  $\phi$  $\omega$
```

大写形式通常把首字母大写，例如：

```
\Gamma$  $\Delta$  $\Theta$  $\Lambda$  $\Sigma$  $\Phi$  $\Omega$
```

LaTeX Workshop 侧栏的“数学符号”区域也可以点击插入，不必全靠记忆。

8.10 求和、积分和极限

如果你想写从 1 加到 n 的求和：

```
\[
  \sum_{i=1}^{n} x_i
\]
```

`\sum` 是求和符号，`_{i=1}` 放在下面，`^{n}` 放在上面。

如果你想写定积分：

```
\[
  \int_0^1 x^2 dx
\]
```

`\int` 是积分号；`\,` 在被积函数和 `dx` 之间加入一小段合适的数学间距。

如果你想写趋近于零时的极限：

```
\[
  \lim_{x \to 0} \frac{\sin x}{x} = 1
\]
```

`\to` 生成箭头。很多数学运算符都有专用命令，不建议直接用英文拼出来。

8.11 矩阵

如果你想写带圆括号的矩阵，可以这样写：

```
\[
  A=
  \begin{pmatrix}
    1 & 2 \\
    3 & 4
  \end{pmatrix}
\]
```

每个 & 分隔一列，\\ 结束一行。常见矩阵环境：

- `pmatrix`：圆括号；
- `bmatrix`：方括号；
- `Bmatrix`：大括号；
- `vmatrix`：单竖线，常用于行列式；
- `Vmatrix`：双竖线。

8.12 分段函数

如果你想写带左大括号的分段函数，可以使用 `cases`：

```
\[
  f(x)=
  \begin{cases}
    x^2, & x \geq 0, \\
    -x, & x < 0.
  \end{cases}
\]
```

& 把条件对齐。`\ge` 表示“大于等于”，也可以写成 `\geq`。

8.13 公式里的中文说明

如果你想在公式中写“当 x 大于零时”这样的中文说明：

```
\[
  f(x)=x^2,\quad \text{当 }x>0\text{ 时}.
\]
```

`\text{...}` 让其中内容使用正常文字排版；`\quad` 加入较宽的数学间距。

8.14 括号随内容自动变大

如果你想让括号根据分数高度自动变大，可以写：

```
\[
  \left(\frac{a}{b}\right)^2
\]
```

`\left` 和 `\right` 必须成对出现。它们也适用于方括号、绝对值和大括号。

新手提示

公式报错时，先检查四件事：美元符号是否成对、大括号是否成对、`\begin` 和 `\end` 是否对应、下划线是否误写在普通正文中。大部分新手公式错误都在这里。

第九章 插入图片

准备区先加载：

```
\usepackage{graphicx}
```

9.1 图片文件放在哪里

推荐给项目建一个 `figures` 文件夹：

```
my-paper/  
|-- main.tex  
|-- references.bib  
`-- figures/  
    |-- model.png  
    `-- result.pdf
```

常用图片格式包括 PNG、JPG 和 PDF。流程图、曲线图优先使用 PDF 这类矢量格式；照片适合 JPG；带透明背景的界面图适合 PNG。

9.2 一个完整图片例子

如果你想插入图片、居中、加标题并允许引用，可以这样写：

```
\begin{figure}[htbp]  
  \centering  
  \includegraphics[width=0.8\textwidth]{figures/example.png}  
  \caption{这是一张示例图片}  
  \label{fig:example}  
\end{figure}
```

9.3 逐行解释

`\begin{figure}[htbp]`

开始一个图片浮动环境。h、t、b、p 分别表示尽量放在当前位置、页顶、页底或专门的浮动页。

`\centering`

让图片在页面中居中。

`\includegraphics`

真正读取图片。`width=0.8\textwidth` 表示图片宽度是正文宽度的 80%。

`\caption`

图片标题，LaTeX 会自动生成“图 1”之类的编号。

`\label`

给图片编号起一个内部名字，供正文引用。

`\end{figure}`

结束图片环境。

9.4 在正文中引用图片

如果你想写“如图 1 所示”，可以这样写：

```
如图 ~\ref{fig:example} 所示，模型准确率逐渐提高。
```

波浪号 ~ 是一个不会在此处断行的空格，能防止“图”和编号被拆开。

9.5 控制图片大小

```
% 占正文宽度的一半
\includegraphics[width=0.5\textwidth]{figures/example.png}

% 固定高度，保持宽高比例
\includegraphics[height=6cm,keepaspectratio]{figures/example.png}

% 最大不超过正文宽度
\includegraphics[width=\linewidth]{figures/example.png}
```

通常只设置宽度，不要同时强行设置宽和高，否则图片可能被拉伸变形。

9.6 图片位置为什么会“乱跑”

LaTeX 会尝试把图片放到不中断阅读的位置，因此图片不一定严格停在源代码所在位置。这叫“浮动”，是正常行为。

推荐处理顺序：

1. 先使用 `[htbp]`，让 LaTeX 自己选择；
2. 把图片代码放在正文第一次提到它的位置附近；
3. 适当缩小图片，避免当前页放不下；
4. 只有确实必须固定时，加载 `float` 并使用 `[H]`。

```
\usepackage{float}

\begin{figure}[H]
  ...
\end{figure}
```

容易踩坑

不要给所有图片都使用 `[H]`。强行固定太多图片可能制造大片空白和难看的分页。论文排版中，图片稍微移动到页顶或下一页通常完全可以接受。

第十章 表格：尤其是论文常用三线表

10.1 最简单的表格

如果你想做一个两列三行的小表格，可以这样写：

```
\begin{tabular}{lc}  
  姓名 & 成绩 \\  
  张三 & 90 \\  
  李四 & 85  
\end{tabular}
```

这段代码怎么理解：

{lc} 表示第一列左对齐、第二列居中。每行中的 & 分隔列，\\ 结束这一行。

10.2 每一列如何对齐

列格式写在 tabular 后的大括号中：

- l: 左对齐；
- c: 居中；
- r: 右对齐；
- p{4cm}: 固定宽度为 4 cm，并允许内容自动换行。

例如三列分别左、中、右对齐：

```
\begin{tabular}{lcr}  
  项目 & 数量 & 金额 \\  
  数据集 & 3 & 120.00 \\  
\end{tabular}
```

10.3 加横线和竖线

如果你想做一个带横线和竖线的普通网格表：

```
\begin{tabular}{|l|c|}
\hline
姓名 & 成绩 \\
\hline
张三 & 90 \\
\hline
\end{tabular}
```

列格式中的 | 是竖线，\hline 是横线。但学术论文通常不推荐满是竖线的网格表，而更常用三线表。

10.4 带标题、编号和引用的完整表格

```
\begin{table}[htbp]
\centering
\caption{不同模型的实验结果}
\label{tab:results}
\begin{tabular}{lcc}
\hline
模型 & 准确率 & 训练时间 \\
\hline
模型 A & 91.2\% & 10 min \\
模型 B & 93.5\% & 15 min \\
\hline
\end{tabular}
\end{table}
```

表 `\ref{tab:results}` 比较了两个模型的实验结果。

table 负责浮动、标题、编号和引用；tabular 负责表格内部的行列。

10.5 论文常用三线表

准备区加载：

```
\usepackage{booktabs}
```

如果你想写规范、简洁的三线表，可以这样写：

```
\begin{table}[htbp]
  \centering
  \caption{不同模型的分类结果}
  \label{tab:classification}
  \begin{tabular}{lccc}
    \toprule
    模型 & 准确率 / \% & 精确率 / \% & 召回率 / \% \\
    \midrule
    ResNet-18 & 92.3 & 91.8 & 91.5 \\
    ResNet-50 & 94.1 & 93.7 & 93.2 \\
    本文方法 & \textbf{95.6} & \textbf{95.1} & \textbf{94.9} \\
    \bottomrule
  \end{tabular}
\end{table}
```

这段代码怎么理解：

`\toprule` 是顶线，`\midrule` 是表头下面的中线，`\bottomrule` 是底线。三线表通常不使用竖线，数据较清爽，学术论文中非常常见。

10.6 表格太宽怎么办

先尝试减少不必要文字、缩短列标题、把单位写进标题。还太宽时，可按优先级处理：

如果你想让某一列自动换行，使用固定宽度列：

```
\begin{tabular}{lp{8cm}}
  方法 & 说明 \\
  方法 A & 这是一段比较长的说明，超过 8 cm 后会自动换行。 \\
\end{tabular}
```

如果你想让表格自动占满正文宽度，使用 `tabularx`：

```

\usepackage{tabularx}

\begin{tabularx}{\textwidth}{lXc}
  \toprule
  方法 & 详细说明 & 准确率 \\
  \midrule
  方法 A & X 列会自动使用剩余宽度并换行。 & 92.1\% \\
  \bottomrule
\end{tabularx}

```

X 列会自动伸缩。若仍然太宽，最后才考虑整体缩放：

```

\resizebox{\textwidth}{!}{%
  \begin{tabular}{lcccccc}
    ...
  \end{tabular}
}

```

整体缩放会让字号一起变小，可能影响可读性，因此不是第一选择。

10.7 合并多列和多行

如果你想让一个单元格横跨两列：

```

\begin{tabular}{lcc}
  \toprule
  模型 & \multicolumn{2}{c}{性能指标} \\
  \cmidrule(lr){2-3}
  & 准确率 & 召回率 \\
  \midrule
  方法 A & 92.1 & 90.5 \\
  \bottomrule
\end{tabular}

```

`\multicolumn{2}{c}{性能指标}` 表示跨两列并居中。`\cmidrule(lr){2-3}` 在第 2 到第 3 列下方画一条局部横线。

如果你想让一个单元格跨两行，先加载 `multirow`，再这样写：

```
\usepackage{multirow}

\begin{tabular}{llc}
\toprule
数据集 & 模型 & 准确率 \\
\midrule
\multirow{2}{*}{数据集 A}
& 模型 1 & 91.2 \\
& 模型 2 & 93.4 \\
\bottomrule
\end{tabular}
```

`\multirow{2}{*}{数据集 A}` 表示向下跨两行，宽度由内容自动决定。

第十一章 交叉引用：让编号永远自动更新

11.1 label 和 ref 是什么

你可以把 `\label` 理解成“给某个对象贴一个内部标签”，把 `\ref` 理解成“在别处读取这个对象现在的编号”。

```
\section{实验结果}
\label{sec:results}
```

详细结果见第 `\ref{sec:results}` 节。

如果“实验结果”后来从第 3 节变成第 4 节，`\ref` 会自动显示新的编号。

11.2 标签为什么要加前缀

推荐命名方式：

对象	前缀	示例
章节	<code>sec:</code> 或 <code>chap:</code>	<code>sec:method</code>
图片	<code>fig:</code>	<code>fig:model</code>
表格	<code>tab:</code>	<code>tab:results</code>
公式	<code>eq:</code>	<code>eq:loss</code>
算法	<code>alg:</code>	<code>alg:training</code>
代码	<code>code:</code>	<code>code:python</code>

前缀不是 LaTeX 强制要求，而是帮助你一眼分清标签类型。标签不要有空格，尽量使用短小、稳定、有意义的英文名称。

11.3 引用章节

```
\section{研究方法}
\label{sec:method}
```

第 `\ref{sec:method}` 节介绍了研究方法。

11.4 引用图片

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.7\textwidth]{figures/model.pdf}
  \caption{模型整体结构}
  \label{fig:model}
\end{figure}
```

模型整体结构如图 `\ref{fig:model}` 所示。

图片的 `\label` 最好紧跟在 `\caption` 后面，因为编号由标题生成。

11.5 引用表格

```
\begin{table}[htbp]
  \centering
  \caption{实验结果}
  \label{tab:results}
  \begin{tabular}{lc}
    \toprule
    模型 & 准确率 \\
    \midrule
    方法 A & 92.1\% \\
    \bottomrule
  \end{tabular}
\end{table}
```

实验结果见表 `\ref{tab:results}`。

11.6 引用公式

```
\begin{equation}
L=-\sum_{i=1}^ny_i\log \hat{y}_i
\label{eq:loss}
\end{equation}
```

损失函数定义见式 `\eqref{eq:loss}`。

公式推荐用 `\eqref`，它会自动带圆括号；图片、表格和章节用 `\ref`。

新手提示

PDF 中出现“??”时，不一定是标签写错，也可能只是还没完成第二轮编译。使用 `latexmk` 保存一次通常就会自动解决。如果仍是“??”，再检查 `label` 拼写是否完全一致、是否出现了重复标签。

第十二章 参考文献：把引用集中管理

12.1 参考文献到底是什么

论文中引用别人的观点、数据、方法或结论时，需要告诉读者来源。正文里通常出现 “[1]” 或 “(Smith, 2024)”，文末则给出作者、题目、期刊、年份等完整信息。

手工维护几十条参考文献很容易出错，因此 LaTeX 通常把文献信息单独放在一个 .bib 文件中。

12.2 BibTeX、Biber 和 bib 文件

先用不那么专业的方式理解：

references.bib

像一个“文献通讯录”，每一条记录一篇文章、一本书或一个网页。

\cite{...}

在正文中说“这里要引用通讯录中的某条记录”。

BibTeX / Biber

负责读取通讯录、排序、生成编号和文末列表。它们是处理工具，不是文献格式本身。

biblatex

提供现代、容易配置的引用命令。本教程使用 biblatex + Biber。

12.3 新建 references.bib

在 main.tex 同一文件夹中新建 references.bib：

```
@book{lampport1994,  
  author    = {Leslie Lamport},  
  title     = {LaTeX: A Document Preparation System},
```

```
publisher = {Addison-Wesley},
year      = {1994}
}

@article{lecun1998,
  author   = {Yann LeCun and L{\'}e{on Bottou and Yoshua Bengio
             and Patrick Haffner},
  title    = {Gradient-Based Learning Applied to Document Recognition},
  journal  = {Proceedings of the IEEE},
  volume   = {86},
  number   = {11},
  pages    = {2278--2324},
  year     = {1998}
}
```

这段代码怎么理解：

@book 表示书, @article 表示期刊论文。紧跟在类型后的 lamport1994、lecun1998 是你自定义的引用键，正文通过它们引用文献。字段之间用逗号分隔。

12.4 在 main.tex 中连接 bib 文件

准备区加入：

```
\usepackage[backend=biber,style=numeric,sorting=none]{biblatex}
\addbibresource{references.bib}
```

- backend=biber 表示用 Biber 处理文献；
- style=numeric 表示正文用数字编号，例如 [1]；
- sorting=none 表示按正文首次引用顺序排列；
- \addbibresource 告诉 LaTeX 文献数据库文件在哪里。

12.5 正文引用和文末列表

如果你想在正文中引用 Lampport 的书，可以写：

Lamport 系统介绍了 LaTeX 的文档准备方法\cite{lampport1994}。

如果你想一次引用多篇文献，可以把多个引用键放在同一对大括号里：

已有研究讨论了这一问题\cite{lampport1994,lecun1998}。

如果你想在文末自动输出参考文献列表，写：

```
\printbibliography[title={参考文献}]
```

放置 \printbibliography 的位置，就是文献列表出现的位置。

12.6 最小可运行参考文献项目

下面是本教程随附、已经实际编译的 main.tex：

```
1 \documentclass[UTF8,fontset=fandol,12pt]{ctexart}
2
3 \usepackage[a4paper,margin=2.5cm]{geometry}
4 \usepackage[colorlinks=true,linkcolor=blue,citecolor=blue]{hyperref}
5 \usepackage[backend=biber,style=numeric,sorting=none]{biblatex}
6
7 \addbibresource{references.bib}
8
9 \title{最小参考文献示例}
10 \author{你的名字}
11 \date{\today}
12
13 \begin{document}
14
15 \maketitle
16
17 Lamport 的著作系统介绍了 LaTeX 的文档准备方法\cite{lampport1994}。
18 深度学习的发展也受到卷积神经网络工作的推动\cite{lecun1998}。
19
20 \printbibliography[title={参考文献}]
21
22 \end{document}
```

对应的 references.bib:

```
@book{lampport1994,
  author    = {Leslie Lamport},
  title     = {LaTeX: A Document Preparation System},
  publisher = {Addison-Wesley},
  year      = {1994}
}

@article{lecun1998,
  author = {Yann LeCun and L{\e}on Bottou and Yoshua Bengio and Patrick
↵ Haffner},
  title  = {Gradient-Based Learning Applied to Document Recognition},
  journal = {Proceedings of the IEEE},
  volume = {86},
  number = {11},
  pages  = {2278--2324},
  year   = {1998}
}
```

在 VS Code 中保存 main.tex，latexmk 会自动调用 XeLaTeX 和 Biber。终端中也可以直接运行：

```
latexmk -xelatex main.tex
```

12.7 常见引用格式

- **数字顺序制**：正文显示 [1]、[2]，理工科常见；
- **作者-年份制**：正文显示作者与年份，社会科学常见；
- **IEEE**：电子、电气和计算机领域常见；
- **ACM**：计算机领域部分会议和期刊使用；
- **GB/T 7714**：很多中文学位论文和中文期刊要求。

真正投稿或写毕业论文时，**优先使用学校、学院、期刊或会议提供的模板和样式**。不要仅凭自己喜欢选择格式。

若安装了对应样式，GB/T 7714 可以这样配置：

```
\usepackage[
  backend=biber,
  style=gb7714-2015,
  gpub=false
]{biblatex}
```

不同模板对参数可能有自己的规定，照模板说明调整。

12.8 从网站复制 BibTeX

如果你想从 Google Scholar 获取：

搜索文献后点击“引用”，再选择 BibTeX，复制整个条目到 `references.bib`。复制后重点检查作者、年份、题目大小写、期刊名、卷期和页码。

如果你想从 IEEE Xplore 或 ACM Digital Library 获取：

在文献页面找到“Cite”或“Export Citation”，选择 BibTeX。不同页面按钮名称略有差异，但核心都是导出 `.bib` 内容。

如果你想从知网获取：

知网页面提供的导出格式可能因文献和页面版本不同而变化。如果没有直接可用的 BibTeX，可以导出到 Zotero、EndNote 等工具后再导出 BibTeX，或者依据页面信息手工建立 `@article` 条目。

容易踩坑

网上复制的 BibTeX 不是百分之百正确。常见问题包括作者顺序错、题目大小写被改、缺 DOI、会议名缩写不统一。正式提交前一定要逐条核对 PDF 首页或出版方页面。

12.9 中文和英文文献有什么区别

LaTeX 的引用方式没有本质区别，都是 `\cite{键}`。区别主要在最终格式：中文作者名、标点、期刊名和“等”字如何显示，通常由 GB/T 7714 或学校模板决定。不要在 `.bib` 中为了模仿排版而随意添加多余标点；让样式统一处理。

第十三章 在论文中插入代码

新手推荐使用 `listings`。它不需要额外开启 `shell-escape`，安装稳定，足够应付课程报告和多数论文。

13.1 准备 `listings` 样式

在准备区写：

```
\usepackage{xcolor}
\usepackage{listings}

\lstset{
  basicstyle=\ttfamily\small,
  keywordstyle=\color{blue},
  commentstyle=\color{green!50!black},
  stringstyle=\color{orange!70!black},
  numbers=left,
  numberstyle=\tiny\color{gray},
  frame=single,
  breaklines=true,
  showstringspaces=false,
  columns=fullflexible
}
```

- `basicstyle` 设置代码使用等宽小号字体；
- `keywordstyle`、`commentstyle`、`stringstyle` 设置关键字、注释和字符串颜色；
- `numbers=left` 在左边显示行号；
- `frame=single` 给代码加边框；
- `breaklines=true` 允许太长的代码自动换行；
- `showstringspaces=false` 不把字符串里的普通空格显示成特殊符号。

13.2 插入简单 Python 代码

如果你想插入一段 Python 代码，可以这样写：

```
\begin{lstlisting}[language=Python]
def add(a, b):
    return a + b

print(add(2, 3))
\end{lstlisting}
```

`language=Python` 告诉 listings 按 Python 语法识别关键字。

实际排版出来的效果如下：

```
1 def add(a, b):
2     # 返回两个数的和
3     return a + b
4
5 print(add(2, 3))
```

13.3 给代码加标题、编号和标签

```
\begin{lstlisting}[
  language=Python,
  caption={计算分类准确率},
  label={code:accuracy}
]
def accuracy(correct, total):
    return correct / total
\end{lstlisting}
```

代码 `~\ref{code:accuracy}` 给出了准确率的计算方法。

`caption` 生成代码标题和编号；`label` 允许使用 `\ref` 引用。

13.4 从外部文件读取代码

如果你想项目里已经有 `train.py`，不想复制粘贴，可以写：

```
\lstinputlisting[  
  language=Python,  
  caption={模型训练代码},  
  label={code:training}  
]{src/train.py}
```

这样源代码更新后，论文中的代码也会随下一次编译更新。代码很长时，论文正文通常只展示关键部分，其余放附录或代码仓库。

第十四章 算法伪代码

伪代码不是直接运行的程序，而是用比较清楚的结构描述算法步骤。它适合展示输入、输出、循环、条件判断和返回值。

准备区加载：

```
\usepackage{algorithm}
\usepackage{algpseudocode}

\floatname{algorithm}{算法}
\algnewcommand\algorithmicrequire{\textbf{输入：}}
\algnewcommand\algorithmicensure{\textbf{输出：}}
```

14.1 机器学习训练流程示例

```
\begin{algorithm}[htbp]
  \caption{机器学习模型训练流程}
  \label{alg:training}
  \begin{algorithmic}[1]
    \Require 训练集  $D$ ，学习率  $\eta$ ，最大轮数  $T$ 
    \Ensure 训练后的参数  $\theta$ 
    \State 随机初始化参数  $\theta$ 
    \For{$t=1$ to $T$}
      \State 从  $D$  中读取一个小批量样本
      \State 计算损失函数  $L(\theta)$ 
      \State  $\theta \leftarrow \theta - \eta \nabla_{\theta} L(\theta)$ 
      \If{验证集性能不再提升}
        \State \textbf{break}
      \EndIf
    \EndFor
    \State \Return  $\theta$ 
  \end{algorithmic}
\end{algorithm}
```

```
\end{algorithm}
```

训练过程见算法 `\ref{alg:training}`。

14.2 逐项理解算法命令

\caption

算法标题，同时生成编号。

\label

给算法设置可引用的标签。

\begin{algorithmic}[1]

开始算法内容，[1] 表示显示行号。

\Require

输入。

\Ensure

输出。

\State

一个普通步骤。

\For ... \EndFor

循环开始和结束。

\If ... \EndIf

条件判断开始和结束。

\Return

返回结果。

下面是这段源代码的实际排版效果：

算法 1 机器学习模型训练流程

输入: 训练集 D , 学习率 η , 最大轮数 T

输出: 训练后的参数 θ

- 1: 随机初始化参数 θ
 - 2: **for** $t = 1$ to T **do**
 - 3: 从 D 中读取一个小批量样本
 - 4: 计算损失函数 $L(\theta)$
 - 5: $\theta \leftarrow \theta - \eta \nabla_{\theta} L(\theta)$
 - 6: **if** 验证集性能不再提升 **then**
 - 7: **break**
 - 8: **end if**
 - 9: **end for**
 - 10: **return** θ
-

第十五章 页面格式、行距、缩进和字体

学校提供模板时，优先使用模板，不要自己重做页面格式。本章适合课程报告、个人技术报告，或帮助你理解模板里的设置。

15.1 纸张大小和页边距

如果你想设置 A4 纸，四边页边距都是 2.5 cm：

```
\usepackage[a4paper,margin=2.5cm]{geometry}
```

如果你想分别设置四边：

```
\usepackage[  
  a4paper,  
  top=2.8cm,  
  bottom=2.6cm,  
  left=3cm,  
  right=2.6cm  
{geometry}
```

左边距有时会比右边距稍大，是为了装订。具体数值应以学校格式要求为准。

15.2 行距

加载：

```
\usepackage{setspace}
```

如果你想设置 1.5 倍行距：

```
\onehalfspacing
```

如果你想设置双倍行距：

```
\doublespacing
```

如果你想自定义为 1.25 倍左右：

```
\setstretch{1.25}
```

15.3 段落缩进和段间距

如果你想让每段首行缩进两个中文字符：

```
\setlength{\parindent}{2em}
```

em 大致与当前字体中一个“M”的宽度相关；中文排版里 2em 常被用作两个字符的缩进。

如果你想段落之间不额外留白：

```
\setlength{\parskip}{0pt}
```

如果你想让章节标题后的第一段也缩进，可以加载：

```
\usepackage{indentfirst}
```

15.4 页眉、页脚和页码

加载：

```
\usepackage{fancyhdr}
```

如果你想页眉中间显示论文标题，页脚中间显示页码：

```
\pagestyle{fancy}
\fancyhf{}
\fancyhead[C]{论文标题}
\fancyfoot[C]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0pt}
```

`\fancyhf{}` 先清空默认页眉页脚；L、C、R 分别表示左、中、右。

如果你想封面不显示页码，可以在封面页写：

```
\thispagestyle{empty}
```

15.5 基础字体大小

文档类选项中的 12pt 决定基础字号：

```
\documentclass[UTF8,fontset=fandol,12pt]{ctexart}
```

局部字号参见第 6 章。正式论文通常由模板统一控制，不要全篇混用多种字号。

15.6 中文字体

本教程使用 `fontset=fandol`，优点是跨电脑稳定：

```
\documentclass[UTF8,fontset=fandol]{ctexart}
```

如果你想明确使用 macOS 自带宋体，可以这样设置：

```
\documentclass[UTF8,fontset=none]{ctexart}
\usepackage{fontspec}
\usepackage{xeCJK}

\setCJKmainfont{Songti SC}
\setCJKsansfont{Heiti SC}
\setCJKmonofont{Kaiti SC}
```

如果把项目发给没有这些字体的人，可能编译失败。因此普通学习项目优先用 Fandol；学校要求宋体、黑体时再按模板配置。

15.7 英文和数字字体

在 XeLaTeX 中，可以用 fontspec 设置：

```
\usepackage{fontspec}

\setmainfont{TeX Gyre Termes}
\setsansfont{TeX Gyre Heros}
\setmonofont{Latin Modern Mono}
```

- `\setmainfont`：英文正文和数字的主要字体；
- `\setsansfont`：无衬线字体；
- `\setmonofont`：代码使用的等宽字体。

如果学校明确要求 Times New Roman，可在确认电脑已安装后写：

```
\setmainfont{Times New Roman}
```

15.8 适合中文课程报告的常用配置

```
\documentclass[UTF8,fontset=fandol,12pt]{ctexart}

\usepackage[
  a4paper,
  top=2.5cm,
  bottom=2.5cm,
  left=2.8cm,
  right=2.8cm
]{geometry}
\usepackage{setspace}
\usepackage{indentfirst}
\usepackage{fancyhdr}

\onehalfspacing
\setlength{\parindent}{2em}
\setlength{\parskip}{0pt}

\pagestyle{fancy}
\fancyhf{}
\fancyhead[C]{课程报告标题}
\fancyfoot[C]{\thepage}
```

这是一套通用起点，不代表任何学校的强制标准。正式提交前必须对照要求调整。

第十六章 封面、中文摘要、英文摘要和目录

16.1 一个简单但完整的封面

如果你想自己做一个课程报告封面，可以使用 `titlepage` 环境：

```
\begin{titlepage}
  \centering
  \thispagestyle{empty}
  \vspace*{2cm}

  {\zihao{2}\bfseries 某某大学\par}
  \vspace{1.5cm}
  {\zihao{-2}\bfseries 课程报告\par}
  \vspace{2cm}

  {\zihao{1}\bfseries 在这里填写论文题目\par}
  \vspace{2.5cm}

  \begin{tabular}{rl}
    作者： & 张三 \\
    学号： & 2026000000 \\
    专业： & 计算机科学与技术 \\
    指导教师： & 李老师
  \end{tabular}

  \vfill
  {\zihao{4}\today\par}
\end{titlepage}
```

16.2 逐块理解封面

- `titlepage` 让封面单独占一页；
- `\centering` 让后续内容整体居中；
- `\thispagestyle{empty}` 隐藏封面页眉页脚；
- `\vspace*{2cm}` 在页面顶部留出竖直空间；
- `\zihao` 设置中文字号，`\bfseries` 加粗；
- `tabular{rl}` 把“作者、学号、专业”等信息排成两列；
- `\vfill` 把剩余空间撑开，使日期靠近页面底部。

容易踩坑

毕业论文通常有学校提供的封面模板，可能要求校徽、固定字体、装订线和准确位置。那种情况下应使用学校模板，不要直接照搬这个通用封面。

16.3 中文摘要

短报告可以使用 `abstract` 环境；长论文使用 `ctexbook` 时，常用无编号章节做摘要：

```
\chapter*{摘\quad 要}
\addcontentsline{toc}{chapter}{摘要}

本文针对……问题，提出了……方法。实验结果表明，
该方法在……数据集上取得了……结果。研究说明……

\vspace{1em}
\noindent\textbf{关键词：} 关键词一； 关键词二； 关键词三
```

`\chapter*` 后面的星号表示不自动编号；`\addcontentsline` 手动把“摘要”加入目录。

摘要内容通常回答五个问题：

1. 为什么做这项工作？
2. 具体解决什么问题？

3. 使用了什么方法?
4. 得到了什么主要结果?
5. 最终说明了什么?

16.4 英文摘要

```
\chapter*{Abstract}
\addcontentsline{toc}{chapter}{Abstract}

This study investigates ... and proposes ... .
Experimental results show that ... .

\vspace{1em}
\noindent\textbf{Keywords:} keyword one; keyword two; keyword three
```

英文摘要不是把中文逐字翻译成生硬英语，而是保持相同信息。关键词常使用英文分号分隔，是否首字母大写按学校要求。

16.5 目录

如果你想生成目录，写：

```
\tableofcontents
\clearpage
```

`\clearpage` 从下一页继续正文，并先处理前面等待排版的图片和表格。

如果你想目录显示到二级标题，可以在准备区设置：

```
\setcounter{tocdepth}{2}
```

层级数字通常理解为：章是 0，节是 1，小节是 2。目录太细会很长，论文一般无需列出所有三级标题。

16.6 论文前几页的组合顺序

下面是常见顺序：

```
\begin{document}

% 1. 封面
\begin{titlepage}
...
\end{titlepage}

% 2. 前置部分使用罗马页码 (ctexbook)
\frontmatter

% 3. 中文摘要
\chapter*{摘\quad要}
...

% 4. 英文摘要
\chapter*{Abstract}
...

% 5. 目录
\tableofcontents
\clearpage

% 6. 正文恢复阿拉伯数字页码
\mainmatter

\chapter{引言}
...

\end{document}
```

`\frontmatter` 和 `\mainmatter` 主要用于 book 类长文档；短报告使用 `ctexart` 时可以直接省略。

第十七章 附录

17.1 附录是什么

附录放“对理解或复现有帮助，但放在正文里会打断阅读”的内容，例如：

- 很长的数学推导；
- 额外实验和更多结果表格；
- 完整问卷或访谈提纲；
- 较长的程序代码；
- 补充图片、证明或数据说明。

17.2 开始附录

如果你想让后面的章自动变成“附录 A、附录 B”，写：

```
\appendix

\chapter{补充实验}
这里放补充实验内容。

\chapter{完整代码}
这里放较长的代码。
```

对于 `ctexart`，通常使用：

```
\appendix

\section{补充实验}
这里放补充实验内容。
```

`\appendix` 本身不会立即打印标题，它只是告诉 LaTeX：“从现在开始，章节按附录方式编号。”

17.3 附录中的图片、表格和代码

附录里的写法和正文完全相同：

```
\appendix
\chapter{补充材料}

\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.7\textwidth]{figures/extra-result.pdf}
  \caption{补充实验结果}
  \label{fig:extra-result}
\end{figure}

\begin{table}[htbp]
  \centering
  \caption{额外参数实验}
  \label{tab:extra}
  \begin{tabular}{lc}
    \toprule
    参数 & 结果 \\
    \midrule
    0.1 & 91.2 \\
    0.2 & 92.5 \\
    \bottomrule
  \end{tabular}
\end{table}

\begin{lstlisting}[language=Python,caption={完整训练循环}]
for epoch in range(100):
    train_one_epoch(model, loader)
\end{lstlisting}
```

图片和表格编号通常会自动带附录字母，具体外观由文档类和学校模板决定。

第十八章 一份可以直接使用的通用论文模板

18.1 模板包含什么

随本教程提供的 `paper-template` 文件夹包含：

- 标题页；
- 中文摘要和关键词；
- 英文摘要和关键词；
- 自动目录；
- 引言、相关工作、方法、实验、结果分析、结论；
- 公式、三线表、图片占位符；
- 机器学习训练伪代码；
- Python 代码块；
- Biber 参考文献；
- 附录；
- 已配置好的 VS Code 自动编译设置。

18.2 如何开始使用模板

1. 复制整个 `paper-template` 文件夹，不要只复制 `main.tex`。
2. 用 VS Code 的“打开文件夹”打开复制后的目录。
3. 修改封面上的学校、题目、作者、学号、专业和教师。

4. 按章节替换示例正文。
5. 把文献加入 `references.bib`, 正文用 `\cite` 引用。
6. 保存 `main.tex`, 等待 `latexmk` 完成编译。
7. 按 `Command + Option + V` 查看 PDF。

终端编译命令是:

```
cd paper-template
latexmk -xelatex main.tex
```

18.3 完整 main.tex

下面就是随附模板的完整源文件, 不是零散片段:

```
1 \documentclass[UTF8,fontset=fandol,12pt,openany]{ctexbook}
2
3 % ----- 页面与字体 -----
4 \usepackage[a4paper,top=2.8cm,bottom=2.6cm,left=3cm,right=2.6cm]{geometry}
5 % ----- 常用功能 -----
6 \usepackage{setspace}
7 \usepackage{indentfirst}
8 \usepackage{amsmath,amssymb,mathtools}
9 \usepackage{graphicx}
10 \usepackage{booktabs,tabularx,array,multirow}
11 \usepackage{xcolor}
12 \usepackage{float}
13 \usepackage{listings}
14 \usepackage{algorithm}
15 \usepackage{algpseudocode}
16 \usepackage{fancyhdr}
17 \usepackage{hyperref}
18 \usepackage{bookmark}
19 \usepackage[backend=biber,style=numeric,sorting=none]{biblatex}
20
21 \addbibresource{references.bib}
22
23 % ----- 正文样式 -----
24 \onehalfspacing
25 \setlength{\parindent}{2em}
26 \setlength{\parskip}{0pt}
27 \setlength{\headheight}{15pt}
```

```
28
29 \pagestyle{fancy}
30 \fancyhf{}
31 \fancyhead[C]{通用中文论文模板}
32 \fancyfoot[C]{\thepage}
33 \renewcommand{\headrulewidth}{0.4pt}
34
35 \hypersetup{
36   colorlinks=true,
37   linkcolor=blue,
38   citecolor=blue,
39   urlcolor=blue,
40   pdftitle={通用中文论文模板},
41   pdfauthor={你的名字}
42 }
43
44 \lstset{
45   basicstyle=\ttfamily\small,
46   keywordstyle=\color{blue},
47   commentstyle=\color{green!50!black},
48   stringstyle=\color{orange!70!black},
49   numbers=left,
50   numberstyle=\tiny\color{gray},
51   frame=single,
52   breaklines=true,
53   showstringspaces=false,
54   columns=fullflexible
55 }
56
57 \floatname{algorithm}{算法}
58 \algnewcommand\algorithmicrequire{\textbf{输入: }}
59 \algnewcommand\algorithmicensure{\textbf{输出: }}
60
61 \begin{document}
62
63 % =====
64 % 封面
65 % =====
66 \begin{titlepage}
67   \centering
68   \vspace*{2cm}
69
70   {\zihao{2}\bfseries 某某大学\par}
71   \vspace{1.5cm}
72   {\zihao{-2}\bfseries 课程报告 / 学术论文 / 毕业设计\par}
73   \vspace{2cm}
74
75   {\zihao{1}\bfseries 在这里填写论文题目\par}
```

```

76 \vspace{2.5cm}
77
78 \begin{tabular}{rl}
79   \zihao{4} 作者: & \zihao{4} 你的名字 \\
80   \zihao{4} 学号: & \zihao{4}2026000000 \\
81   \zihao{4} 专业: & \zihao{4} 计算机科学与技术 \\
82   \zihao{4} 指导教师: & \zihao{4} 某某老师 \\
83 \end{tabular}
84
85 \vfill
86 {\zihao{4}\today\par}
87 \end{titlepage}
88
89 \frontmatter
90
91 % =====
92 % 中文摘要
93 % =====
94 \chapter*{摘要}
95 \addcontentsline{toc}{chapter}{摘要}
96
97 这里填写中文摘要。摘要通常说明研究背景、要解决的问题、使用的方法、主要实验结果和结论。
98 ↪ 摘要一般不放图片、表格和很长的公式。
99
100 \vspace{1em}
101 \noindent\textbf{关键词: }LaTeX; 学术写作; 论文模板; XeLaTeX
102
103 % =====
104 % 英文摘要
105 % =====
106 \chapter*{Abstract}
107 \addcontentsline{toc}{chapter}{Abstract}
108
109 Write the English abstract here. Briefly introduce the background, problem, method,
110 ↪ main results, and conclusion.
111
112 \vspace{1em}
113 \noindent\textbf{Keywords:} LaTeX; academic writing; thesis template; XeLaTeX
114
115 \tableofcontents
116 \clearpage
117
118 \mainmatter
119
120 % =====
121 % 第一章 引言
122 % =====
123 \chapter{引言}

```

```

122 \label{chap:introduction}
123
124 这里介绍研究背景、研究问题和本文贡献。LaTeX 的文档准备思想由 Lamport
    ↳ 系统推广\cite{lampport1994}。
125
126 本文的主要贡献可以写成列表：
127
128 \begin{itemize}
129   \item 提出了一种简单、清晰的方法；
130   \item 完成了可重复的实验；
131   \item 对实验结果进行了分析。
132 \end{itemize}
133
134 \section{研究背景}
135
136 这里填写研究领域的背景和实际意义。
137
138 \section{本文结构}
139
140 第\ref{chap:related}章介绍相关工作，第\ref{chap:method}章介绍方法，
    ↳ 第\ref{chap:experiment}章给出实验，第\ref{chap:conclusion}章总结全文。
141
142 % =====
143 % 第二章 相关工作
144 % =====
145 \chapter{相关工作}
146 \label{chap:related}
147
148 这里按照主题介绍已有研究，并指出它们与本文工作的关系。不要只罗列论文，要说明它们做了什么、
    ↳ 有什么不足、与你的方法有什么联系。
149
150 % =====
151 % 第三章 方法
152 % =====
153 \chapter{方法}
154 \label{chap:method}
155
156 设输入为  $x$ ，模型参数为  $\theta$ ，预测结果为
157
158 \begin{equation}
159   \hat{y}=f(x;\theta).
160   \label{eq:model}
161 \end{equation}
162
163 式\eqref{eq:model} 给出了模型的基本形式。
164
165 \section{训练流程}
166

```

```

167 \begin{algorithm}[htbp]
168   \caption{机器学习模型训练流程}
169   \label{alg:training}
170   \begin{algorithmic}[1]
171     \Require 训练集  $D$ , 学习率  $\eta$ , 最大轮数  $T$ 
172     \Ensure 训练后的参数  $\theta$ 
173     \State 随机初始化参数  $\theta$ 
174     \For{ $t=1$  to  $T$ }
175       \State 从  $D$  中读取一个小批量样本
176       \State 计算损失函数  $L(\theta)$ 
177       \State  $\theta \gets \theta - \eta \nabla_{\theta} L(\theta)$ 
178       \If{验证集性能不再提升}
179         \State \textbf{break}
180       \EndIf
181     \EndFor
182     \State \Return  $\theta$ 
183   \end{algorithmic}
184 \end{algorithm}
185
186 算法\ref{alg:training} 展示了一个简化的训练过程。
187
188 % =====
189 % 第四章 实验
190 % =====
191 \chapter{实验}
192 \label{chap:experiment}
193
194 \section{实验设置}
195
196 表\ref{tab:settings} 给出示例实验设置。
197
198 \begin{table}[htbp]
199   \centering
200   \caption{实验设置}
201   \label{tab:settings}
202   \begin{tabular}{lcc}
203     \toprule
204     参数 & 取值 & 说明 \\
205     \midrule
206     学习率 & 0.001 & Adam 优化器 \\
207     批大小 & 32 & 每批样本数 \\
208     训练轮数 & 100 & 最大训练轮数 \\
209     \bottomrule
210   \end{tabular}
211 \end{table}
212
213 \section{示例图片位置}
214

```

```

215 图\ref{fig:placeholder}是一个不依赖外部文件的占位图。替换成真实图片时，把方框部分换成
    ↳ \verb|\includegraphics| 命令。
216
217 \begin{figure}[htbp]
218     \centering
219     \fbox{\rule{0pt}{5cm}\rule{0.75\textwidth}{0pt}}
220     \caption{请在这里替换成实验结果图片}
221     \label{fig:placeholder}
222 \end{figure}
223
224 % =====
225 % 第五章 结果分析
226 % =====
227 \chapter{结果分析}
228
229 这里解释实验结果为什么会这样，而不是只重复表格里的数字。可以讨论最佳结果、异常现象、
    ↳ 误差来源、方法局限和改进方向。
230
231 \begin{lstlisting}[language=Python,caption={简单的 Python 示例},label={code:python}]
232 def accuracy(correct, total):
233     return correct / total
234
235 print(accuracy(92, 100))
236 \end{lstlisting}
237
238 代码\ref{code:python} 展示了一个简单的准确率计算函数。
239
240 % =====
241 % 第六章 结论
242 % =====
243 \chapter{结论}
244 \label{chap:conclusion}
245
246 这里总结本文解决了什么问题、采用了什么方法、得到了什么结果，并简要说明未来工作。
    ↳ 不要在结论中突然加入正文从未出现的新实验。
247
248 \printbibliography[heading=bibintoc,title={参考文献}]
249
250 % =====
251 % 附录
252 % =====
253 \appendix
254
255 \chapter{补充实验}
256
257 这里放不适合塞进正文、但又值得保留的推导、额外实验、完整问卷或大段代码。
258
259 \chapter{补充代码}

```

```
260
261 \begin{lstlisting}[language=Python,caption={附录中的示例代码}]
262 for epoch in range(10):
263     print(f"epoch = {epoch}")
264 \end{lstlisting}
265
266 \end{document}
```

18.4 完整 references.bib

```
@book{lampport1994,
  author    = {Leslie Lamport},
  title     = {LaTeX: A Document Preparation System},
  publisher = {Addison-Wesley},
  year      = {1994}
}

@article{lecun1998,
  author = {Yann LeCun and L{\'}eon Bottou and Yoshua Bengio and Patrick
    ↪ Haffner},
  title  = {Gradient-Based Learning Applied to Document Recognition},
  journal = {Proceedings of the IEEE},
  volume = {86},
  number = {11},
  pages  = {2278--2324},
  year   = {1998}
}
```

新手提示

这是一份“通用起点”，适合学习、课程报告和个人技术论文。真正的毕业论文或投稿论文，应把学校、期刊或会议模板放在第一优先级，因为它们可能对封面、标题层级、页边距、参考文献样式和字体有明确要求。

第十九章 常见报错和解决方法

遇到报错时，不要从日志最后一行开始猜。LaTeX 错误经常像多米诺骨牌：前面一个小错误，会引起后面很多错误。优先找到日志中最早出现的错误，修复后重新编译。

19.1 忘记写 end document

报错现象：

编译出现 Emergency stop、File ended while scanning，或者没有生成完整 PDF。

可能原因：

写了正文开始命令，但文件末尾没有对应的正文结束命令；也可能是某个更早的大括号或环境没有结束。

解决方法：

在正文最后补上 `\end{document}`。同时检查每个环境是否都有同名的结束命令。

推荐检查步骤：

从文件末尾往上看；使用 VS Code 的括号高亮；折叠环境，寻找没有闭合的部分；修复最早的日志错误后重编译。

19.2 中文无法显示

报错现象：

中文变成方框、乱码、空白，或日志提示 Missing character、font not found。

可能原因：

使用了 pdfLaTeX；源文件不是 UTF-8；没有使用 ctex；指定了电脑不存在的字体。

解决方法：

使用 ctexart 或 ctexbook，用 XeLaTeX 编译，并先选择 fontset=fandol。

推荐检查步骤：

看 VS Code 右下角是不是 UTF-8；确认构建 recipe 是 XeLaTeX；在终端运行 `kpsewhich ctexart.cls`；暂时移除自定义中文字体设置。

19.3 图片找不到

报错现象：

日志出现 `File `...` not found`，PDF 中图片位置为空或编译停止。

可能原因：

路径写错；文件名大小写不一致；图片不在项目目录；扩展名错误；路径相对于错误的主文件理解。

解决方法：

把图片放入项目的 `figures` 文件夹，并使用类似 `{figures/model.png}` 的相对路径。

推荐检查步骤：

在 VS Code 左侧确认图片确实存在；复制文件名避免拼错；检查 `Model.png` 和 `model.png` 的大小写；确认打开的是包含主文件的整个文件夹。

19.4 参考文献不显示

报错现象：

正文出现 `[?]`，日志提示 `Citation undefined`，或者文末列表为空。

可能原因：

`\cite` 的引用键和 `.bib` 不一致；没有 `\printbibliography`；Biber 没运行；`.bib` 语法中缺逗号或大括号。

解决方法：

检查引用键，确保加载 `biblatex`、调用 `\addbibresource` 和 `\printbibliography`，然后使用 `latexmk` 重新编译。

推荐检查步骤：

查看 `main.blg` 和 LaTeX Workshop 日志；运行 `latexmk -xelatex main.tex`；若缓存损坏，执行 `latexmk -C main.tex` 后重新编译。

19.5 公式报错

报错现象：

日志提示 `Missing $ inserted, Extra }, Bad math environment delimiter`。

可能原因：

美元符号没有成对；大括号缺失；在普通正文中直接写下划线；把 `\[...\]` 放进另一个数学环境。

解决方法：

确认每个数学环境只有一种开始和结束方式；多字符上下标使用大括号；正文下划线写成 `_`。

推荐检查步骤：

暂时注释掉最近添加的公式；逐行恢复；数一数 `$` 和大括号；检查报错行前一两行，因为真正错误可能发生在上一行。

19.6 表格太宽

报错现象：

表格伸出右边页边距，日志出现 `Overfull \hbox`。

可能原因：

列太多、文字太长、列标题冗长，或者所有列都用不会换行的 `l/c/r`。

解决方法：

缩短文字；用 `p{宽度}` 或 `tabularx` 自动换行；横向拆成两个表格；最后才使用 `\resizebox`。

推荐检查步骤：

先看是哪一列最长；把单位移到表头；使用 `X` 列；确认缩小后字号仍然可读。

19.7 编译后 PDF 没变化

报错现象：

保存源文件后右侧 PDF 仍是旧内容。

可能原因：

文件没有保存；本次编译失败，所以旧 PDF 被保留；编辑的是子文件但 LaTeX Workshop 找错主文件；预览没有刷新。

解决方法：

先看底部状态栏是否显示构建成功；打开 LaTeX Workshop 日志；手动执行构建和刷新预览。

推荐检查步骤：

按 `Command + S`；点击“构建 LaTeX 项目”；确认日志最后有 `All targets are up-to-date`；多文件项目在子文件顶部加入 `% !TEX root = main.tex`。

19.8 VS Code 预览打不开

报错现象：

按 `Command + Option + V` 没反应，或提示找不到 PDF。

可能原因：

文档还没有成功编译；LaTeX Workshop 没加载；PDF 输出目录与配置不一致；扩展刚安装但窗口没重载。

解决方法：

先成功构建一次，再执行“查看 LaTeX PDF”；必要时运行 `Developer: Reload Window`。

推荐检查步骤：

在资源管理器里确认同名 PDF 存在；点击 LaTeX 侧栏的“查看日志消息”；确认 `latex-workshop.view.pdf.viewer` 为 `tab`。

19.9 找不到 xelatex

报错现象：

终端显示 `command not found: xelatex`，VS Code 显示 `spawn xelatex ENOENT`。

可能原因：

TeX Live 没安装完成；终端和 VS Code 没读取新 PATH；配置使用了错误路径。

解决方法：

新开终端并刷新系统路径：

```
eval "$(/usr/libexec/path_helper)"
```

然后确认 `/Library/TeX/texbin/xelatex` 存在。

推荐检查步骤：

运行 `command -v xelatex` 和 `xelatex --version`；完全退出并重开 VS Code；检查项目设置中的命令路径。

19.10 找不到 latexmk

报错现象：

显示 `command not found: latexmk` 或 LaTeX Workshop 构建失败。

可能原因：

TeX 发行版不完整、PATH 没刷新，或使用 BasicTeX 但没有安装 latexmk。

解决方法：

完整 MacTeX 通常自带 latexmk；BasicTeX 用户运行：

```
sudo tlmgr install latexmk
```

推荐检查步骤：

分别运行 `command -v latexmk` 和 `latexmk -v`；确认返回了 TeX 路径，然后重开 VS Code。

19.11 一个通用排错顺序

1. 保存所有文件。
2. 找日志中最早的错误，不要先看最后一条。
3. 检查最近添加的那几行。
4. 检查大括号、美元符号、环境开始和结束。
5. 暂时注释掉可疑代码，再逐步恢复。
6. 终端运行 `latexmk -xelatex main.tex`，观察完整错误。
7. 必要时运行 `latexmk -C main.tex` 清理缓存，再重新编译。

第二十章 零基础一周学习路线

20.1 第一天：只追求成功生成 PDF

第一天不要碰复杂表格和参考文献。完成以下任务：

1. 建立一个项目文件夹和 `main.tex`；
2. 复制最小中文示例；
3. 保存并看到 PDF；
4. 写三段正文；
5. 加入 `section` 和 `subsection`；
6. 学会加粗和行内公式。

第一天的成功标准不是“背命令”，而是你能独立创建文件、编译、预览和找到日志。

20.2 第二天：公式、图片和表格

完成一个两页小报告：

1. 写一个带编号公式并引用它；
2. 插入一张图片，添加标题和标签；
3. 写一张三线表；
4. 在正文中引用图片和表格；
5. 故意改错一个图片路径，学会从日志找到问题。

20.3 第三天：目录和参考文献

完成以下练习：

1. 写 3 个一级标题和若干二级标题；
2. 使用 `\tableofcontents` 生成目录；
3. 新建 `references.bib`；
4. 添加两篇文献；
5. 在正文中引用它们；
6. 使用 `\printbibliography` 生成文末列表。

20.4 第四到第七天

第四天：

学习代码块和算法伪代码；把一段真实 Python 代码放进报告。

第五天：

练习封面、中文摘要、英文摘要、关键词和目录。

第六天：

使用本教程的 `paper-template` 写一篇 4-6 页的小论文，主题可以是你熟悉的课程项目。

第七天：

专门做排错练习：改错标签、图片路径和引用键，再把它们修好；最后整理项目目录并备份。

20.5 写论文必须掌握的内容

- 文档结构：`documentclass`、`usepackage`、`begin/end document`；
- 段落和章节；
- 行内公式、带编号公式和对齐公式；
- 图片、三线表、标题和标签；
- `label/ref/eqref` 交叉引用；

- .bib、cite 和文末参考文献；
- 在 VS Code 中构建、预览和查看日志；
- 知道怎样使用学校或期刊模板。

20.6 可以暂时不学的内容

零基础阶段可以先跳过：

- 自己编写文档类 .cls；
- 自己开发宏包 .sty；
- TeX 底层盒子、胶水和复杂宏编程；
- TikZ 的复杂绘图；
- 自定义输出例程；
- 为每个标点做微观排版调整。

这些内容并不是没用，而是不会阻止你写出一篇正常的课程报告或论文。遇到确切需求再学，效率更高。

20.7 写作时最实用的工作习惯

1. 每次只改一小块，保存并确认能编译，再继续写。
2. 图片文件使用英文、数字、短横线命名，避免空格和奇怪符号。
3. 标签名有规律，例如 fig:model、tab:results、eq:loss。
4. 参考文献随读随加，不要在提交前一天集中补。
5. 定期备份，最好使用 Git；至少保留按日期命名的副本。
6. 提交前在另一台电脑或干净文件夹中重新编译一次，检查是否遗漏图片、字体或文献文件。

最后的命令速查表

想做什么	最常用写法
新段落	两段之间空一行
加粗	<code>\textbf{文字}</code>
斜体	<code>\textit{words}</code>
红色文字	<code>\textcolor{red}{文字}</code>
一级标题	<code>\section{标题}</code>
二级标题	<code>\subsection{标题}</code>
无序列表	<code>\begin{itemize} ... \end{itemize}</code>
行内公式	$a^2+b^2=c^2$
无编号公式	<code>\[... \]</code>
带编号公式	<code>\begin{equation} ... \end{equation}</code>
分数	<code>\frac{分子}{分母}</code>
图片	<code>\includegraphics[width=...]{路径}</code>
图片标题	<code>\caption{标题}</code>
标签	<code>\label{fig:name}</code>
引用编号	<code>\ref{fig:name}</code>
引用公式	<code>\eqref{eq:name}</code>
引用文献	<code>\cite{引用键}</code>
目录	<code>\tableofcontents</code>
附录	<code>\appendix</code>
终端编译	<code>latexmk -xelatex main.tex</code>
清理缓存	<code>latexmk -C main.tex</code>

你已经可以开始写了

你不需要等到“完全学会 LaTeX”才开始论文。复制通用模板，先把标题和章节换成自己的内容，然后在需要公式、图片、表格或引用时回来查对应章节。真正的熟练，来自在自己的文档里反复使用。