

國立臺北科技大學
電子工程系碩士班
碩士學位論文

集成自動化工具的 L^AT_EX 臺北科技大學學位
論文模板與使用指引

A User Guide for the L^AT_EXThesis Template of NTUT
with Integrated Automation Tools

研究生：佚名

指導教授：佚名 博士

中華民國一百一十四年六月



國立臺北科技大學

電子工程系碩士班

碩士學位論文

集成自動化工具的 L^AT_EX 臺北科技大學學位
論文模板與使用指引

A User Guide for the L^AT_EX Thesis Template of NTUT
with Integrated Automation Tools

研究生：佚名

指導教授：佚名 博士

中華民國一百一十四年六月

在此插入審定書



摘要

論文名稱：集成自動化工具的 LATEX 臺北科技大學學位論文模板與使用指引

頁數：30 頁

校所別：國立臺北科技大學 電子工程系碩士班

畢業時間：一百一十四學年度 第二學期

學位：碩士

研究生：佚名

指導教授：佚名 博士

關鍵詞：關鍵詞 1、關鍵詞 2、關鍵詞 3

這是一份台北科技大學的 LATEX 學位論文模板，同時也是「如何用 LATEX 優雅地折磨自己」的入門指南。本文不僅教你如何使用模板的核心功能、調整排版，還貼心地為你準備了初學者的「踩坑大全」和「細節魔鬼」。當然，別忘了參考學校那些藏在角落裡的神祕 PDF 文件，以及 IEEE 的參考文獻格式——畢竟，誰不想在寫論文時順便成為格式大師呢？總之，這是一份讓你從 LATEX 小白變成「勉強能用」的必備寶典，祝你好運！—由 AI 總結

ABSTRACT

Title : A User Guide for the L^AT_EXThesis Template of NTUT with Integrated Automation Tools

Pages : 30 pages

School : National Taipei University of Technology

Department : Department of Electronic Engineering

Time : June, 2025

Degree : Master

Researcher : Anonymous

Advisor : Guidonymous, Ph.D.

Keyword : keyword1, keyword2, keyword3,

This is a L^AT_EX thesis template for Taipei Tech, and it ' s also a beginner-friendly guide to using L^AT_EX. It explains the main features of the template, how to change the layout, and gives tips for common problems beginners face. You ' ll also find links to helpful files from the school and learn how to format references like a pro (thanks to IEEE rules). Think of it as your cheat sheet to go from "What is L^AT_EX?" to "I can do this!" Good luck, and have fun! — Summarized by AI

誌 謝

首先，感謝我的指導教授，您的耐心與智慧讓我明白，研究不僅是一場與知識的搏鬥，更是一場與自己的耐力賽。感謝實驗室的同學們，你們的陪伴讓我在無數個熬夜的日子裡，零食和冷笑話一樣不可或缺。感謝我的家人和朋友，你們的支持讓我明白，即使我經常把“論文快寫完了”變成“快了快了”。

感謝所有在這段旅程中給予我幫助的人，無論是精神上的支持還是默默忍受我的抱怨。這段經歷讓我學會了如何在壓力下保持微笑（即使是在崩潰邊緣）。

終於，這篇論文不僅是學術上的成就，更是我人生中的一次重要成長與蛻變。願所有研究生都能順利畢業，迎接更美好的明天——因為我們都知道，這只是開始，而不是結束。—由 AI 產生



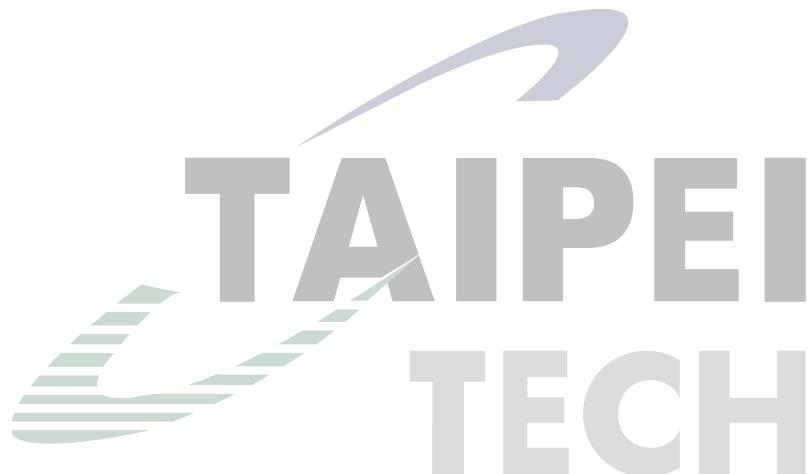
目 錄

摘要.....	i
ABSTRACT.....	ii
誌 謝.....	iii
目 錄.....	v
表目錄.....	vi
圖目錄.....	vii
第一章 Introduction.....	1
1.1 模板使用引導	1
1.1.1 必要參數	3
1.1.2 自動化指令	4
1.1.3 工作區檔案結構	6
1.1.4 排版細節微調	6
1.2 L ^A T _E X 初步使用引導	7
1.2.1 基於自指的指令引導	7
1.2.2 使用 B _I B ^A T _E X 參考文獻.....	10

第二章 常見問題.....	14
2.1 文章結構.....	14
2.1.1 入門寫作指引	14
2.1.2 其他常見文本排版問題	16
2.2 符號系統設計	17
2.3 美化排版.....	18
2.3.1 多圖與跨欄排版	18
2.3.2 簡化引用	18
2.3.3 標題斷行	20
2.3.4 表格	20
2.3.5 美化公式	21
2.3.6 程式代碼	23
2.3.7 函數定義與演算法	23
2.4 參考文獻格式	23
2.4.1 引用網頁	24
2.4.2 引用 arXiv	25
第三章 Conclusion	27
參考文獻.....	28

表目錄

表 1.1 表格也是一種浮動環境，一般而言標題在上。 8



圖目錄

圖 1.1 圖片是其中一種浮動環境，一般而言標題會在圖片下方。	7
圖 1.2 不同平台 BiBTEx 引用文獻格式取得步驟	11
圖 1.3 使用 Google Scholar 生成 BiBTEx 以及推薦的引用名稱	11
圖 2.1 使用表格的多圖排版演示	19
圖 2.2 使用\subfloat 的多圖排版跨欄演示	19



第一章 Introduction

本文是一份臺北科技大學學位論文的 L^AT_EX 排版範本，同時也是本範本的引導文件與簡易的 L^AT_EX 教程。在這份教程中會 1) 簡易的介紹本範本中所提供的核心自動化功能、2) 本範本中的檔案結構、3) 如何根據自身需求調整部分排版、4) 初學以 L^AT_EX 撰寫學位論文中常遇到的問題以及 5) 一些可能需要額外注意的細節考量。本範本的編纂過程中參考了多項文件，其中包含：

- 本校 (國立臺北科技大學) 法規一覽表頁面 [1] 中編號為 **J1** 的論文撰寫規範書與編號為 **J2** 的論文撰寫範例。
- 本校圖書館所提供的學位論文相關說明下載區 [2]，該網頁中提供了多項與學位論文有關的檔案下載。其中“學位論文編輯”表格中的**本校學位論文格式範本**。該範本中包含兩份可供參考的 PDF 文件於 PDF 與說明。此外，該表格中也提供了另外的 Latex 範本由學生分享於 Github。
- 由於學校並無特別規定參考文獻格式，參考文獻格式應遵循各科系主流格式即可。因此，我們在參考文獻格式方面的相關說明則參考於 IEEE 相關規範，其中主要包含了 IEEE Reference Guide [3] 與 How to Use the IEEEtran BibTeX Style [4] 這兩份文件。同時，我們也會藉由 BibTeX 官方的說明 [5]，討論如何在不使用 **IEEEtran.bst** (IEEE 提供的參考文獻風格文件) 的情況下實現 IEEE 的 Reference 格式¹。

在撰寫論文的同時記得在各自系上網頁查看必要要求，並可於 [6] 查詢相關文獻以及在 [7] 確認畢業流程。另一方面，務必注意，您的論文頁數與引用數量都應比這篇模板/介紹之內容要多得多。如果內容量不及本篇文章，請務必再擴充文本與內容。

1.1 模板使用引導

相較於更為人所知且普遍被使用的 Microsoft (MS) Word 而言，L^AT_EX 最大的優勢莫過於是 1) 更好看且方便的數學字體、2) 更整齊且美觀的整體文字排版以及關鍵的 3) 排

¹現在的 Overleaf 已經默認裝載了 **IEEEtran.sty** 與 **IEEEtran.bst** 這兩個風格化文件，可以在不依賴外部檔案的情況下直接撰寫 IEEE 格式的論文。

版功能的自動化和 4) 內容與排版解耦²。這其中“排版功能的自動化”尤為重要，這是一部分人始終使用 LATEX 作為文章撰寫工具的核心原因，特別是相當依賴於使用交叉引用的學術文章尤其如此。

當然，其寫作使用純文字（代碼）使（文章）內容與排版解耦合的特性使得 LATEX 更容易被主流的版本控制工具（Git/Github）檢索與比對、在資料壓縮過程中可以獲得更好的壓縮率，同時也更容易與其他常見格式（如：Markdown 與 HTML）之間相互轉換。這些也是重要的優勢，但是並不會直接的帶來體驗升級，也很難作為無可替代的原因。

對於大多數初學 LATEX 的使用者而言，通常最先會被 `\section` 指令與數學環境所吸引。這兩者也確實對應了 LATEX 最直觀的兩項功能 1) 自動編號與 2) 可編程的複雜數學排版。接下來，真正開始作業後，`figure` 環境與 `table` 環境加入為手邊常用的工具。而 `\ref` 和 `\cite` 這倆引用工具的開始被使用後，“自動化”在 LATEX 中的涵義才真正得到體現。然而 LATEX 雖然是圖靈完備的程式語言，然而其語法晦澀且抽象方式相當小眾且古老，一般使用者並不會為了用其編寫自動化的排版代碼而來。反之，用戶則是為了那些已經準備好的、開箱即用的工具而來。

如今大多數可見的學位論文模板廣泛的使用 `\input` 指令並將其視為“便於模組化”的開發工具。這可以讓主文件看起來簡潔，但是反而容易喪失了整合開發環境（integrated development environment, IDE）提供自動補全的便利性，也反而更難構造各個文章段落和交叉引用之間的關係。同時，在“正文之外”很多因為各人之間複用性高而真正需要模板化的部分卻沒有真正的模板化，反而需要各人手動調整。這不僅帶來了不便，而且也喪失了 LATEX 自動化排版的巨大優勢。

在這個模板中我們不僅提供了已經經過光學比對誤差足夠小的排版模板，同時我們也提供了自動化工具以在給定了各個研究生必須的個人訊息後可以藉由足夠直覺的指令自動生成下面頁面而無須手動修改文本與計算頁數：

- 1) 論文封面
- 2) 空白頁
- 3) 書名頁³
- 4) 學位論文口試委員會審定書（碩士、博士）

²文字內容本身不影響排版，排版由代碼指令產生，這杜絕了一般集成 GUI 介面的字排版軟體（如 MS Word）在一部分文字內容產生變化時可能會影響其他內容排版結果的問題。

³電子論文由此開始

- 5) 中文摘要
- 6) 英文摘要
- 7) 謝
- 8) 目錄
- 9) 表目錄
- 10) 圖目錄

除此之外，針對謝、目錄、表目錄、圖目錄、參考文獻與章節標題格式這些中文與英文論文有區別的部分也可以藉由簡單的指令直接切換論文中所有相關的設定。接下來我們會簡單的描述如何使用這些自動化工具，然後再進一步的說明如果不喜歡這些格式，或是需要進一步調整這些格式應該於何處用什麼代碼進行處理。

1.1.1 必要參數

根據最終整理的結果，在學位論文中可重複使用的必要資訊包含了九組（中英文）合計總共十六個必要參數。這些參數會用於那謝自動生成特定頁面的自動化指令之中。具體而言，這些參數在本模板中表示如下：

- 中文標題為\titlezh，英文姓名為\titleen。
- 中文姓名為\authorzh，英文姓名為\authoren。
- 中文指導教授姓名為\advisorzh，英文指導教授姓名為\advisoren。
- 中文學位名稱為\degreezh，英文學位名稱為\degreeen。
- 中文系所名稱為\depzh，英文系所名稱為\depen。
- 中文學校名稱為\schoolzh，英文學校名稱為\schoolen。
- 中文畢業民國年為\yearzh，英文畢業西元年為\yearen。
- 中文畢業月份為\monzh，英文畢業月份為\monen。
- 中文畢業學年度為\semesterzh，英文畢業學年度為\semesteren。

在 L^AT_EX 中我們可以使用\renewcommand 指令更新上面這些參數的名稱，具體而言可以參考下面代碼。如果我們要將中文標題更換為「這是一個標題」，我們可以如下變更參數名稱：

```
1 \renewcommand{\titlezh}{這是一個標題}
```

注意，為了避免初學使用者不知道在何處更新參數才能有效影響文檔內容，本

模板已經在導言區⁴定義好了所有參數的變更指令，使用者可以直接在其中變更所需要變更的參數即可。

1.1.2 自動化指令

在這一小節中會簡要的介紹那些自動化生成特定頁面的指令的應用方式與具體行為。首先生成封面頁的指令為 `\makecoverpage`，這個指令會生成一個特定格式的封面頁，以及規定上緊隨其後的空白頁。接著，作為上傳“電子論文”第一頁的書名頁則是使用 `\maketitlepage` 執行。上述這兩個指令都會根據“指令下達前所定義”的 1) 研究生姓名、2) 指導教授姓名 3) 中英文標題和 4) 畢業年月去生成具體頁面排版。

接下來的頁面需要插入“學位論文口試委員會審定書”，這份審定書對於碩士班與博士班學生而言略有不同。在學校表單頁面 [6] 中可以分別在 J3 找到碩士班的審定書，並在 J9 找到博士班的審定書。在此模板中定義了 `\checkresult` 指令執行填入審定書的工作，這個指令會在“填入審定書路徑時”將審定書插入於指定的頁面位置中，如：

```
1 \checkresult{path/ 審定書.pdf}
```

而如果沒有填入路徑

```
1 \checkresult{}
```

則會如現在的狀態一般生成一個佔用頁提示尚未完成審定書的插入並指示審定書最終的放置位置。這個範本同樣也提供了以 LATEX 生成的自動化審定書排版可供使用，其具體指令為：

```
1 \makeverifmaster{系所}{姓名}
```

如果填入系所與姓名，可以獲得自動排版的審定書提供列印⁵，如果欄位留空則可以獲得僅填入畢業民國年為`\yearzh` 的排版結果。如果想要盡可能最大化利用先前填入的必要參數則可以參考下面指令生成排版結果：

```
1 \makeverifmaster{\depzh\degreezh{}班}{\authorzh}
```

由於碩士班與博士班的模板有微妙區別，這裡也分為兩種指令分別為碩士班模板生

⁴document 環境 (`\begin{document}`) 開始之前的區域，反之在`\begin{document}... \end{document}`之間的則是正文，而`\end{document}`後的代碼不會被處理。

⁵由於需要口試委員簽名，最終都需要以外部檔案的方式才能插入於文檔之中。

成\makeverifmaster 和博士班模板生成\makeverifphd。這兩種指令的使用方式皆如同上面所述。

注意！ 雖然學校在 [6] 提供了 MS Word (.doc) 與 OpenDocument Text (.odt) 兩種格式，但是在不同辦公室軟體 (MS Word/Apple Pages/OpenOffice/LibreOffice) 上讀取檔案時可能會由於演算法不同、預設參數不同甚至字體庫不同而產生的排版結果差異，甚至於使排版結果產生錯亂。這時，由於 L^AT_EX 生成的檔案是直接的排版結果 (PDF)，因此可以嘗試直接嘗試使用上述本模板提供之指令的生成結果。

而對於中文與英文摘要，本模板中提供了 **abszh** 與 **absen** 環境取代傳統上使用 **\input** 拆分的方法。中英文摘要的環境都需要一個“關鍵詞 (Keywords)”參數，具體而言其使用方法如下⁶：

1 \begin{abszh}{關鍵詞1、關鍵詞2、關鍵詞3}

2 這是一份台北科技大學的\LaTeX{}學位論文模板，同時也是「如何用\LaTeX{}優雅地折磨自己」的入門指南。本文不僅教你如何使用模板的核心功能、調整排版，還貼心地為你準備了初學者的「踩坑大全」和「細節魔鬼」。當然，別忘了參考學校那些藏在角落裡的神祕PDF文件，以及IEEE的參考文獻格式[]畢竟，誰不想在寫論文時順便成為格式大師呢？總之，這是一份讓你從\LaTeX{}小白變成「勉強能用」的必備寶典，祝你好運！ --- 由 AI 總結

3 \end{abszh}

填入環境內的內容會成為摘要的內容，而關鍵詞以及先前提供的必要參數則會自動排版生成常用的摘要模板，而文章的總頁數也是自動生成的結果。上面演示的最終排版結果也可以在[模板演示的頁面](#)中查看。

在本模板中使用了 **\makecontentlists** 指令一次性地生成所有的需要的目錄，包含 1) 章節目錄、2) 表目錄和 3) 圖目錄。除此之外，在規定中這些非正文內容使用小寫羅馬字母作為頁碼，而正文使用阿拉伯數字作為頁碼。在此模板中使用 **init** 環境區分非正文內容與正文內容。在這個環境範圍內的內容會使用小寫羅馬數字標記頁碼，而之後的內容將重置使用阿拉伯數字作為頁碼。

除此之外，本模板中提供了 **\setlang** 指令，這個指令可以接收 **zh** 或是 **en** 的文字參數修改中英文的預定義排版規則。其中中文與英文的差異包含 1) 誌謝頁面到

⁶此處以中文摘要演示，英文摘要僅須將環境中的參數名稱修改為前面所述之英文摘要的環境名稱即可。

圖目錄頁面的標題英文化，以及 2) 章節標題的英文化。

1.1.3 工作區檔案結構

這個模板中以 `main.tex` 作為主文件而 `ref.bib` 是 BibTeX 格式文獻資料的儲存檔案。除此諸外，模板中 `style.tex` 定義了模板中具體的排版攝鏡，`macros_clean.tex` 定義了論文中常用的指令，`env.tex` 定義了本模板中提供的自動化指令以及撰寫本模板時使用到的文章格式指令，最後 `makelist.tex` 定義了目錄生成相關的指令。

1.1.4 排版細節微調

如果需要調整具體的排版細節有兩種方式，一種是直接進入 `style.tex` 或者 `env.tex` 中修改排版細節，另一種是在這兩個檔案的 `\input` 之後修改排版格式。常見的排版格式修改包含標題的格式、段落間距和字體的修改。這方面的修改具體而言可以參考並搜尋下面幾條指令，由於文章篇幅關係這裡將不介紹其使用方式：

```
1 % 字體設定
2 \setmainfont{Times New Roman} % 設定英文字型 Times New Roman
3 % \setCJKmainfont{cwTeXKai} % 另一個Overleaf預安裝的楷體字體
4 % \setCJKmainfont{標楷體.ttf} % 工作區中上傳的標楷體檔案
5 \setCJKmainfont{TW-Kai} % 目前使用的楷體字體
6
7 % 段落與行間距設定
8 \setlength{\parindent}{2em}
9 \setlength\baselineskip{1em}
10 \setlength{\parskip}{1em}
11 \linespread{1.5}
12
13 % 標題間距設定
14 \titlespacing*\{chapter\}{0pt}{-1em}{2em}
15 \titlespacing*\{section\}{0pt}{0em}{-1em}
16 \titlespacing*\{subsection\}{0pt}{0em}{-1em}
17
18 % 標題格式設定
19 \titleformat{\chapter}[hang]{\centering\fontsize{20}{30}\bfseries}{Chapter\hspace{1ex}\thechapter}{0.5em}{}  
TAIPEI  
TECH
```

1.2.1 基於自指的指令引導

自指 (Self-reference) 是哲學與數學上的重要組成結構，其可以簡單的理解為「以自身定義自身」。在 L^AT_EX 中，其執行過程可以理解為是以文字 (代碼) 編輯自身 (文字) 本身在版面中的存在形式 (排版) 的過程，我們將依賴這一思路去建構對於 L^AT_EX 語法的引導。首先，我們以列表形式描述接下來所要介紹的內容

- 一項列表元素
- 作為浮動元素的圖與表
- 基於 Package Algorithm2e 的演算法撰寫
- 引用自動編號的元素

圖 1.1: 圖片是其中一種浮動環境，一般而言標題會在圖片下方。

1.2 L^AT_EX 初步使用引導

在這一節中會使用一段簡易的仿論文描述展示論文中常用的功能 (敘事方式) 要如何在 L^AT_EX 中使用⁷。這些功能具體使用方式不會直接地一一在文本中進行介紹而是展示於原始代碼之中，因此務必留意最終文檔內容與原始代碼的異同之處，而範例中也會盡可能地以完全可以區分的內容展示不一樣的代碼功能。

1.2.1 基於自指的指令引導

自指 (Self-reference) 是哲學與數學上的重要組成結構，其可以簡單的理解為「以自身定義自身」。在 L^AT_EX 中，其執行過程可以理解為是以文字 (代碼) 編輯自身 (文字) 本身在版面中的存在形式 (排版) 的過程，我們將依賴這一思路去建構對於 L^AT_EX 語法的引導。首先，我們以列表形式描述接下來所要介紹的內容

- 一項列表元素
- 作為浮動元素的圖與表
- 基於 Package Algorithm2e 的演算法撰寫
- 引用自動編號的元素

這些內容我們將會以實例自指的形式呈現。

浮動元素是 L^AT_EX 中的重要環境，常見的浮動環境包含圖片與表格其環境名稱分別為 `figure` 和 `table`。這些浮動環境會自動選擇合適的位置排版，或者我們也可以使用參數指定這個浮動環境可以在哪些位置排版。這些參數包含 `h` (當前位置)、`t` (頂部)、`b` (底部)、`H` (強制當前位置)，並且除了 `H` 之外可以自由組合。通常而言會建議使用 `[htb]` 或 `[tb]` 作為浮動環境的排版參數。但是由於學位論文本身

⁷注意，接下來的描述中包含著錯誤與發散的內容，除了排版語法以外的訊息僅供娛樂而不具備實際指導性。

的性質，因此使用 `[h]` 或 `[H]` 強制鎖定圖片位置的需求也並非少見。需要特別注意的是，通常而言浮動表格的標題 `\caption` 通常會置於“頂部”，而浮動圖片的標題會置於“底部”。像是圖 1.1 是不使用參數的自動排版，而表 1.1 則是使用參數強制將圖片排版於畫面底部。

演算法也是論文中的重要組成，所有可以被應用於現實中的方法策略通常而言都可以以文字/偽代碼形式描述。而合理的演算法應具備下面的特徵：

- 1) 明確輸入
- 2) 明確輸出
- 3) 有限的邏輯狀態(命令)

Algorithm 1: 一個清楚的演算法標題

Data: 一些輸入

Result: 一些輸出

```
1 初始化代碼;  
2 while 迭代條件 do  
3   迭代過程;  
4   for 遍歷規則 do  
5     遍歷流程;  
6     if 分支條件 then  
7       若條件相符;  
8     else  
9       若條件不符;  
10    if 觸發條件 then  
11      條件執行;
```



一般而言，演算法中常用的文字訊息包含兩種迴圈與條件判斷。而實際上，這些邏輯構造也確實便足以組成並解決所有的可計算問題。除了演算法之外，代數、符號與方

表 1.1: 表格也是一種浮動環境，一般而言標題在上。

此行置中	自訂寬度	此行靠右	此行靠左
表格	會隨著文章	長度變化	但是使用 <code>p</code> 作為參數時可以鎖定長度並提供自動換行，這裡設定為 10em(10 個“M”的寬度)

程式也是描述問題的重要工具。尤其是當問題本身存在遞迴特性時，如

$$f(x) = \begin{cases} f(x-1) + 1, & x = \text{cond}, \\ 0, & x \neq \text{cond}. \end{cases}$$

除了遞迴之外，我們也可以用 L^AT_EX 的數學模式表示一些更加具有物理意義的方程式

$$C = B \log_2 \left(1 + \frac{S}{I + n_0} \right) \quad (1.1)$$

或是一些常用的數學工具

$$\begin{pmatrix} 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2x_1 + 3x_2 \quad (1.2)$$

如上所示，很多概念在研究的過程中遭遇到的模型與概念都需要使用數學語言進行描述。此外，對於指數 x^2 甚至是連續數列 $[a_i, a_{i+1}, \dots]$ 的描述也需要這樣的數學工具的輔助才能順利進行。當然也不能忘記使用連加 \sum 、連乘 \prod 、連續聯集 \bigcup 和連續交集 \bigcap 符號以簡化演算法中的表達形式。藉由這些符號也才能如下表達極限等抽象概念

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

此外，對於一篇學術論文而言引用也是重中之重。引用文獻可以為自己的文章做知識與事實的補充，也可以作為自身研究工作新穎性與必要性的舉證。在常見的理工論文中通常使用方括號的形式引用“文獻”[3]。而對於連續引用而言則沒有詳實的統一標準，我們可以直接使用 [8–10] 的形式引用一系列文章，也可以像如今更多期刊論文也逐漸回歸這種逐一引用的形式 [8], [9], [10]。除此之外，交叉引用也是論文撰寫中重要的功能。藉由已編號的圖片、表格與公式，可以解決僅使用文字描述時的歧義，也可以簡化描述時所需的語句數量。這也是 L^AT_EX 最為標誌性的自動化功能之一，L^AT_EX 中可以藉由\label 指令錨定當前環境中的編碼，並使用\ref{label}的形式引用並生成對應編碼。這個 label 與編碼的關聯可以用於章節1.2.1、方程式(1.1)以及浮動環境的圖1.1、表1.1和演算法1。

至此已經完成了 L^AT_EX 中常用的排版功能介紹，並且提供了具像化的範例。更複雜的功能可以在有需求時參考更多 L^AT_EX 的相關的指令功能，尤其是 L^AT_EX 的數學排版功能有許多常用但是無法詳細說明的部分可以參考 [11]。

1.2.2 使用 BibTeX 參考文獻

BibTeX 是 L^AT_EX 常用的參考文獻管理格式，其格式與 JSON (JavaScript Object Notation) 有異曲同工之妙。一般而言，BibTeX 的形式會類似於下面格式

```
1 @entry{entryname  
2     author={...},  
3     title={...},  
4     args...  
5 }
```

不同的 **entry** (條目) 是不同類型的文獻，常見、常用的包含 **article** (journal)、**inproceedings** (conference) 和 **misc** (網頁與其他)。此外，其中需要特別注意的是上面代碼中的 **entryname** 是在使用\cite 指令引用所使用的“變數名稱”。以本模板而言文獻 [4] 的 BibTeX 代碼為

```
1 @misc{ieeebib,  
2     title      = {How to Use the IEEEtran BibTeX Style},  
3     author     = {Michael Shell},  
4     year       = {Feb. 2025},  
5     note       = {\url{https://tw.mirrors.cicku.me/ctan/macros/latex/  
6     contrib/IEEEtran/bibtex/IEEEtran_bst_HOWTO.pdf}},  
7     howpublished = {IEEE}
```

這裡的 **ieeebib** 便是我們引用時所用的名稱，因此我們需要以\cite{ieeebib} 在 L^AT_EX 中實現參考文獻的引用。其餘如 **author** 與 **title** 等等參數都是可以根據實際參考文獻格式需求調整甚至於刪除，而一般而言 **author** 與 **title** (有時包含 **year**) 是參考文獻的必要參數。一個 BibTeX 格式的條目代表著一篇參考文獻，這些條目一般會被彙整在一個以 **.bib** 作為副檔名的檔案中並使用下面代碼自動生成指定風格的參考文獻

```
1 \bibliographystyle{參考文獻風格}  
2 \bibliography{無副檔名的檔名}
```

具體而言，在本模板中的代碼為

```
1 \bibliographystyle{IEEEtran}  
2 \bibliography{ref}
```



圖 1.2: 不同平台 BibTeX 引用文獻格式取得步驟



圖 1.3: 使用 Google Scholar 生成 BibTeX 以及推薦的引用名稱

一般上，工程類的論文常以 IEEE 格式作為規範，因此風格方面本論文中使用 IEEE 提供的 **IEEEtran** 風格。除此之外，其他預定義的參考文獻風格可以參考 [12]⁸。

BibTeX 格式更偏向於代碼而非人類直接使用的文本，這樣的格式可以再 L^AT_EX 中自動化的轉換為各種所需要的參考文獻格式。雖然這樣的格式不容易直接被人類直接使用與編輯，但所幸主流的學術文獻查詢平台都可以直接得到學術文獻的 BibTeX 引用格式。在這裡我們會介紹三個理工學院常用且有自動提供 BibTeX 格式的出版平台如何在其文章發表網頁得到對應的 BibTeX 引用格式，這三個平台分別為 1) IEEE、2) Sciedencedirect 與 3) arXiv。而沒有特別介紹或者不直接提供 BibTeX 引用格式的平台則可以直接使用 Google Scholar 生成可用的 BibTeX 引用格式，這一點在後面我們也會提供具體的生成方式介紹，並且我們也推薦在實際應用中使用 Google Scholar 生成的“文獻引用名稱”取代一些平台所生成的 BibTeX 上的流水號式的引用名稱。

在圖 1.2 中使用紅色箭頭與對應的文字說明了 1) IEEE Xplore⁹、2) Sciedencedirect 網站與 3) arXiv 網站產生 BibTeX 引用文獻格式的具體流程。其中 IEEE Xplore 與 arXiv 會在

⁸ **IEEEtran** 是 Overleaf 預先安裝的風格之一，但是並非 BibTeX 的標準風格之一。

⁹ 即 IEEE 的線上數位資料庫平台，也就是平常 IEEE 論文的發布網站。

點擊 Cite This 或 Export BibTeX Citation 後彈出一個網頁內的子窗口。IEEE Xplore 需要再點擊 BibTeX 分頁才能找到結果，而 arXiv 的窗口內則直接是生成結果。Sciencedirect 網站則不太一樣，其點按 Cite 後可以生成多種格式的結果其中便有 BibTeX 選項，不一樣的是他會直將生成的結果作為指定檔案格式(如 .bib)下載而非在網頁上開啟。其他平台雖然沒有提及，但是多半也能在“Cite”或“Citation”等關鍵字上找到對應的引用格式生成功能。如果無法方便的找到官方的生成結果也可以使用 Google Scholar 自動生成的 BiBT_EX，其具體過程可以參考圖1.3。簡而言之，Google Scholar 上可以直接在搜尋條目下方找到“引用”，點擊後一樣會產生一個網頁內的子窗口其中有個“BiBT_EX”連結，點按之後可以得到類似於圖1.3(b)的 BiBT_EX 結果。

值得注意的是，由於我們希望在 L_AT_EX 撰寫過程中還是希望能足夠方便且簡短的區分文獻，因此時常會建議使用“年分”、“作者姓”與“文章標題單詞”取代圖1.2(a)中那樣流水編號或者1.2(c)那樣過長的名稱。這時可以注意到圖1.3(b)中 Google Scholar 自動生成的 BiBT_EX 其引用名稱剛好符合合適的命名需求。因此，個人建議 BiBT_EX 文獻命名可以直接使用 Google Scholar 所生成的結果¹⁰。另一方面，通常而言 L_AT_EX 排版時會傾向於將“title”參數的內容以僅首字母大寫的方式排版，這時標題有些必須大寫的部分會被錯誤排版。同樣以 [4] 舉例，在上面給出的代碼中“title”部分在最終排版結果中會變得非常難看(因為本文中也沒有為其進行特別調整)，實際上在要求參考文獻排版時我們應該把那些需要維持大小寫的部分以 {...} 的方式包起來，以做出類似於如下的改進

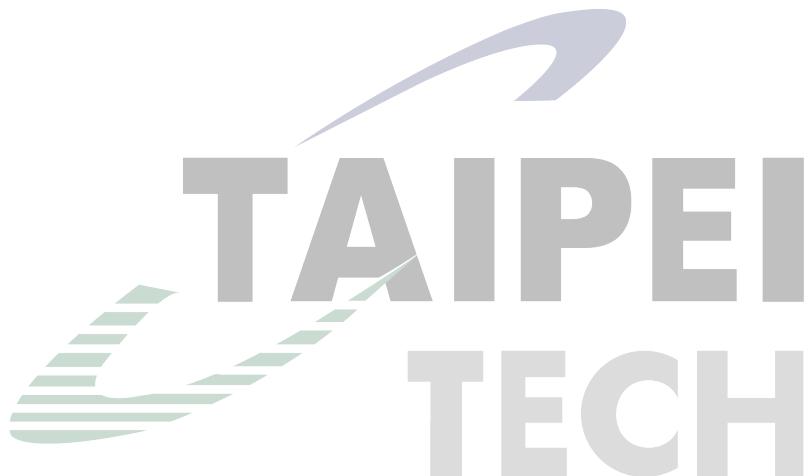
```
1 @misc{ieeebibAlt,
2   title      = {How to Use the {IEEETran} {BibTeX} Style},
3   author     = {Michael Shell},
4   year       = {Feb. 2025},
5   note       = {\url{https://tw.mirrors.cicku.me/ctan/macros/latex/
6                 contrib/IEEETran/bibtex/IEEETran bst_HOWTO.pdf}},
7   howpublished = {IEEE}
```

這時如 [13] 的結果便會是符合我們預期、較為美觀且正確的排版方式。

至此，BiBT_EX 文獻格式的主要使用方式已經大致上介紹完畢，而整體上本模板與 L_AT_EX 的基本使用方式也已經完成說明。然而 L_AT_EX 是一個圖靈完備的排版語言，在實際使用過程中可能仍會遭遇到一些排版的問題與需求需要使用更多的 Package、經

¹⁰具體的 BiBT_EX 內容還是以官方提供的訊息為準進行調整更不容易出錯。

驗¹¹甚至複雜的代碼解決，甚至有些問題對於不熟悉排版術語的使用者來說可能難以準確地描述足以搜尋的提問。好在，僅考慮學術與學位論文寫作中所會遭遇的問題都較為單純而且提問的人較多，在接下來的章節中我們會進一步說明一些常見的學位論文排版問題。此外，這裡也會為初次撰寫論文的圖者提供(學位)論文的基本撰寫思路。



¹¹由於 LATEX 作為沒有命名空間，甚至於語法邏輯都是基於巨集 (Macro) 的古老程式語言，指令的順序、Package 順序等等都有可能影響代碼執行的具體行為。

第二章 常見問題

許多初入 L^AT_EX 的使用者也是第一次作為研究生至撰寫具備學術性質或者說論文性質的文章。這時會有很多人不知所措，不知道從何起頭，也不知道如何收尾，更可能會疑惑「做好」為何如此困難。簡而言之，正是因為「第一次」的身份，“我們”沒有一個適合的評斷依據與參考標準去判斷自己的決策與自己所撰寫的文章是否是導師們想要的亦或者是否是自己所想要的。這些問題在大語言模型當道的現在很多都已經不再是問題，但是缺乏經驗的“我們”也仍需要依據與思路去判斷 AI 所描述的是否正確，而我們又應當如何使之更加符合我們的需求。這一章節會簡要地解釋與描述一些常見的問題，以使得這份模板可以不僅僅作為代碼意義上的“模板”，也能為有需要的使用者提供一些必要的思路。

2.1 文章結構

2.1.1 入門寫作指引

萬事起頭難，對於大部分的研究生而言「如何為自己的論文開頭？」又「如何安排自己的文章架構」通常是最艱難而苦澀的時候。尤其是當我們並不知道「為什麼」要這麼做的時候。而這一切「為什麼」實際上問題在閱讀其他已發表的學術期刊論文時就已經開始醞釀。

如何讀懂一篇論文是需要大量練習與訓練。這裡所練習與訓練的並不是如何在語言的層面看懂一篇論文，而是在需求與目標的層面看懂一篇論文。論文這個格式作為當前學術研究的主流載體，雖然在日新月異的數位時代其版面與格式或許已經有些過時與落伍，但是其文章結構仍有其值得參考的意義。雖然期刊論文與學位論文常用的文章結構略有不同，但是參考期刊論文常見的論述結構我們能更加直觀地明白論文這種載體想要的是什麼效果，而學位論文又是因為什麼原因而有了微妙的不同。

以個人認為中文的「論文」一詞是相當優雅且準確的稱呼，比起 paper, article, report, essay, ... 等等英語上的細分規格都更能體現這種文字體裁所應具備的含義。依我之見，論文的意義在於論述觀點，而如何系統性地展示「這個觀點」的價值與意義正是「論文」的文章結構上應該具備的能力。我們可以注意到，常見的論文體裁通常

文章結構都不會缺少下面幾個部分

- 1) 緒論/介紹 (Introduction)
- 2) 相關文獻 (Related Works)
- 3) 系統模型 (System Model)
- 4) 問題定義 (Problem Definition)
- 5) 提出方法 (Proposed Method)
- 6) 模擬/實驗結果 (Simulation/Numerical/Experiment Results)
- 7) 總結 (Conclusion)

這樣的文章結構只是「模板」亦或者這有其原理，很多人都有自己的看法。而我希望回歸到我一開始的表述「論文是一種用以表達觀點，並說明觀點意義的文章體裁」。這個表述其實本身伴隨著一個問題，即「我們要如何說明和證明這個觀點的意義呢？」這個問題的答案便是其結果。

「很多時候，問題遠比答案更重要。」如果一個觀點有意義，就意味著這個觀點經的起各種方面的質詢(提問)。首先，如果要能說服一個人某個觀點是具有意義的，那麼這個觀點必須對於現實世界或是邏輯思維上具備指導性。我們無法說明一個「憑空出現」且難以定義的事物或問題是否具有意義，因此我們需要讓我們的觀點與現實世界或是與某種現今已知的邏輯概念發生交集(即便只是數學意義上)。為了說明一個觀點與現有事物/概念的連結，並指出這個觀點並非憑空而生而是有著清晰的邏輯脈絡(動機)，因此我們會需要一個章節(緒論/介紹, Introduction)去說明這一切。

接著，也同樣由於一切並不會憑空發生，我們需要有現今已知、已被認可的事物為我們的觀點背書。此外，沒有人喜歡重複造輪胎。一個已經被人所研究過、論述過的觀點，不需要再沒有理由、沒有觀察的重新研究一次。如果我們的觀點是新的，就意味著過往的觀點必須是舊的；如果我們有新觀察是對的，那就意味著過往的觀察是錯的或是需要補充的。而我們要如何證明這點，便只能舉證足夠數量或者足夠有代表性的過往觀點。這不僅是一種演辯工具，讓過去的研究，也成為支持你觀點的武器；也是在巨人的肩膀上，讓過去的堆砌，推砌成足以見證全新風景的台階。這便是需要過往“相關文獻(Related Works)”的理由。

我並沒有打算每一章節的概念都以上述的形式說下去，我們必須真正直指那問題的核心——「很多時候，問題遠比答案更重要」。觀點與問題往往相映而生，「如果多了

這個考量 …？」這時考量便成了某些「變化」形成的觀點，而在提出「為什麼這個現象在 … 會發生？」爾後問題本身孕育著即將孵化的答案(觀點)。我們需要足夠清晰的去描述我們的問題從而引導我們的觀點得以在他人的思想中孵化。系統模型與問題定義常常出現在同一個章節之中，正因為他們本來就為了這樣相同的目的而生。

「你說的很對，那我問你，你怎麼能證明你的方法的正確性呢？」—證明，就是論述我們觀點意義的最後一步。我們可以有很多種方法，而實驗與模擬往往是最直白的。基於一個簡單的邏輯，在環境本身不變的情況下，基於我們觀點產生的方法去與過往不包含我們觀點的方法比較，觀察是否在我們所設想的具體指標上存在如我們所述的影響，這便是提出方法與模擬實現的意義。而在這之後理所當然的，重新描述自己的觀點與驗證流程，為我們的觀點論述作出總結。

很多人說，寫論文就像是寫故事。我想這並沒有錯。論文是給人閱讀的，我們需要讓人能讀下去、能理解最終才能明白我們所希望他們知道的。「摘要」是一個給自己論述的這個故事一個貼合文章結構而且吸引人的介紹，讓人可以一眼就看出來你的觀點哪裡吸引人而又哪裡與眾不同。

最後，我們回到一個問題「學位論文」到底微妙的差在哪裡呢？依我之見，學位論文不僅僅只是一個闡述觀點的「論文」，它還具備了兩項功能 1) 作為教育單位學術研究的總結性結果，其還可以是對後輩新人的指引，以及 2) 作為畢業條件其同時也是對於自身專業理解、專業能力的側寫。因此許多的學位論文不僅僅是觀點的呈現，同時也兼具著許多科普的責任，需要明確的展示自身對於這些專業領域的概念有著足夠的理解，也需要能證明自己的理解足以引領他人入門。

2.1.2 其他常見文本排版問題

上面介紹了論文寫作的大致要點，接下來對於論文文本本身排版有些要求，特別是在意中英文混合排版的讀者可以特別注意一些問題。特別是理工類的中文論文無法避免中英文混合排版的問題，這裡對於中英文混合排版有幾個簡單的建議。

- 1) 專有名詞與專業術語至少在文章中首次出現時應列出其英文名稱，而這個詞彙如果會在文章中大量出現通常會順便舉出其縮寫。舉例而言，低地球軌道 (Low Earth orbit, LEO) 衛星、基地台 (Base station, BS)、積體電路 (Integrated circuit, IC)。如果不確定一個名詞是否應該舉出英文名稱，就應該列出英文名稱，因為這意味著該詞彙本身就並不足夠大眾 (為人所知)。

- 2) 在全文中風格與用詞統一。由於一個名詞概念同時有英文、中文和縮寫三種表達形式，在文章中除了名詞第一次出現以外其餘時候盡可能只使用唯一一種表達形式以保證文章閱讀順暢。而英文詞彙上是首字母是否大寫主要也是以閱讀舒服美觀為主。如果是常見的名詞且不在句首則全文統一風格即可，如果是專有名詞或有特別意義的詞彙則建議首字母大寫，而如果是標題則建議除了介系詞之外各個單詞首字母都大寫。
- 3) 確保中英文用詞無歧義。不論是在中文或者英文中，有許多詞彙需要特定上下文才能明確其無歧義的含義。因此，有些詞彙在使用前應該在搜尋階段確保含義正確以避免誤用，或者藉由中英文對照明確其所表達的含義。其中最常出現的誤用是標準化、正規化這類習慣用詞，不僅計算過程應在文章中將其明確定義，也需要在資料搜集階段就確定應該是 Standardize, Normalize 或者甚至於 Maximum absolute scaling。合理且明確的用詞可以增加自己的專業度，也能顯著的提升文章的易讀性。
- 4) 中英混合排版時有時候英文與中文之間應有的間距(空白)會遺失。如果要避免這項狀況發生 English Text 和中文文字之間可以在撰寫文章時主動留白。除此之外，英文字詞與括號在 LATEX 中不會主動產生空白(間距)，因此當使用當英文單字與括號相鄰時需要主動的留出空白以避免極其難看的排版。這種狀況常見於縮寫的使用上，舉例而言我們更希望看到 “Table of Contents (ToC)” 而不是 “Table of Contents(ToC)”。

2.2 符號系統設計

很多人在撰寫學位論文時才第一次接觸到使用數學語言的形式描述問題，這時不要懷疑，和寫程式不一樣的在論文的形式上符號精簡而優美而易於理解才是正道。我們已經在文章裡解釋過一次符號的意義了，我們就不需要再為符號取冗長的名稱了。為了方便理解與，通常這符號會參考下英文翻譯，像是容量的 “capacity” 可以用 c 表示，如果太多不同的 capacity 我們還有上下標、大小寫、 \hat{c} 、 \bar{c} 甚至 c' 可以使用。

也要記得，數學上本來就有一些常用的符號，像是代表函數的 f/F , g/G , h/H ，或是常用來作為索引的 i,j,k ，以及常常作為標變數的 a,b,x,y,z ，以及常見的那些 α, β, γ ，這些符號都時常用作抽象概念的代表符號使用。更重要的是參考已經看過的論文，對於不同領域時常有一些常見的、大家熟悉的符號系統。有些甚至會取代原有的符號

慣例，像是機器學習裡面的 Q, π, \dots ，還有訊號處理裡常常直接使用 y, x 作為訊號，有些已經成為符號設計的「慣例」都可以記錄下來在論文中加以利用。

2.3 美化排版

研究生群體裡從來不缺少對於排版有美觀追求的人，但往常使用 MS Word 時由於排版與文字內容耦合度高，對於論文「排版」本身的美感追求反而會使文字內容編輯工作變得十分彆扭（特別是雙欄排版格式下圖片與表格的位置調整）。但是在 L^AT_EX 上內容與排版的解耦，為一些排版上的美感追求挪出了空間。然而由於 L^AT_EX 本身語法的複雜性以及臃腫且過時的庫管理，實際上要做出美觀的排版對於新使用者而言並不友善也相當耗費時間與心力。因此，在這一節中介紹了一些較為常用且實用排版語法，為有一些美觀追求的使用者減少心神的消耗。

2.3.1 多圖與跨欄排版

對於多圖排版常見的有兩種方式，一種是如圖 2.1 使用表格進行多圖排版，另一種則是如圖 2.2 使用樣式包進行多圖排版。

有些論文排版格式會要求雙欄，這時多圖排版容易侵犯到另一欄導致排版出現問題。在 L^AT_EX 中對於浮動環境提供了跨欄的排版，只需要將原先的環境名稱後加上 * 即可。具體而言，一般的浮動圖片排版如下

```
1 \begin{figure}  
2 ....  
3 \end{figure}
```

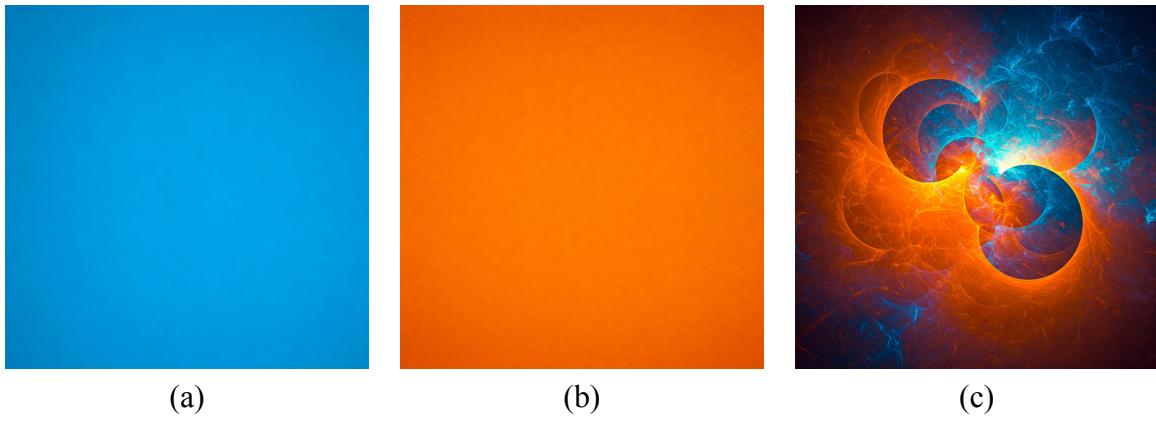
而跨欄的浮動圖片排版則如下

```
1 \begin{figure*}  
2 ....  
3 \end{figure*}
```

除了跨欄的圖片排版之外，跨欄的表格排版也常被使用，這裡也直接提供具體的環境名稱 `table*`。

2.3.2 簡化引用

至此應該有人注意到 `\ref` 指令只會生成標號，如果需要“圖. n”這樣的常用結構就需要一直輸入 `圖.\ref` 這樣的代碼結構。針對這樣的問題，我們可以自行定義

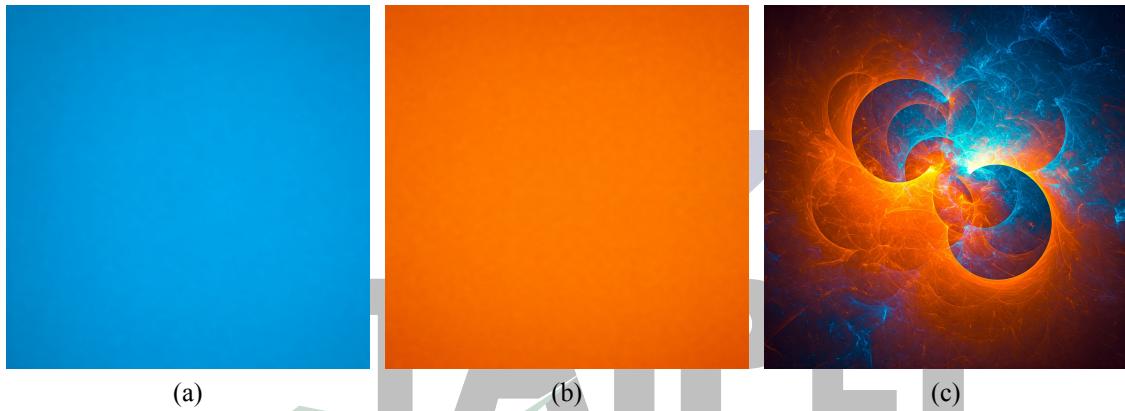


(a)

(b)

(c)

圖 2.1: 由 Gemini 產生的圖片。(a) 藍色的純色圖片 (b) 橘色的純色圖片 (c) 適合與藍色和橘色搭配的顏色。



(a)

(b)

(c)

圖 2.2: 由 Gemini 產生的圖片。(a) 藍色的純色圖片 (b) 橘色的純色圖片 (c) 適合與藍色和橘色搭配的顏色。

指令如下

```
1 \newcommand{\Fig}[1]{Fig.\ref{#1}}
```

這樣我們便可以使用 `\Fig{fig:multifig2}` 這樣的代碼形式生成 Fig. 2.2 這樣的結果。在 `macros_clean.tex` 中定義了一系列這樣的指令，如果需要可以自行於檔案中修改生成結果的格式，其中比較常用的指令包含如下

```
1 \newcommand{\Eq}[1]{Eq.\ref{#1}}
2 \newcommand{\Fig}[1]{Fig.\ref{#1}}
3 \newcommand{\Tab}[1]{Table\ref{#1}}
4 \newcommand{\Algo}[1]{Algorithm\ref{#1}}
```

其中我們額外建議，如果不習慣預設的 `\Algo`，可以將其修改為 `\Alg` 或者使用類似於下面的代碼在不修改原始定義的情況下新增指令

```
1 \newcommand{\Alg}[1]{\Algo{#1}}
```

2.3.3 標題斷行

這是專屬於本模板的特有問題，自動化處理固然方便但是卻較難調整一些客製化的需求。有些人可能會注意到，有些標題在標題頁與封面頁斷行/換行之處並不好看。一般而言 L^AT_EX 上可以自行在標題中使用 `\backslash` 換行，但是在此模板中由於使用 `\settitles` 指令或者藉由 `\renewcommand` 修改 `\titlezh` 或 `\titleen` 的標題會被自動填入到多個地方，因此不能直接在這些指令中調整標題。

為針對這個問題，本文中額外定義了 `\setbigtitles`、`\setbigtitleszh` 和 `\setbigtitlesen` 指令可以單獨修改封面頁與標題頁的標題，具體而言我們可以如下使用這些指令

```
1 \setbigtitleszh{中文\\標題}  
2 \setbigtitlesen{English\\Title}  
3 \setbigtitles{中文\\標題}{English\\Title}
```

需要特別注意的是，這三個指令都需要在原先標題指令使用之後、封面頁生成指令使用之前使用才能正常生效。

2.3.4 表格

有注意到的人應該注意到了，在上面的演示中 L^AT_EX 中的表格使用 `c`, `l`, `r` 表示表格直欄的文字對齊方式為置中、靠左或靠右。而這三個指令都會「自動改變表格寬度」，因此如果當文字過長時表格排版就會超出合理的版面範圍。在演示中同樣也呈現了使用 `p` 指令限制表格大小的方法，但是這個方法默認情況下只會靠左對齊。如果需要在 `p` 模式下使用其他對齊方式可以參考 [14] 並可以使用 [15] 的方式簡化指令。

在表格中另一個常用功能是合併行或列，在 L^AT_EX 中這是一項複雜的功能，具體語法可以參考官方文檔 [16] 中“**Combining rows and columns**”小節。而複雜的複合合併則可以參考 [17]，主要細節是行列的合併嵌套有一定順序，否則有可能會出現編譯錯誤，一般而言嵌套方式如下

```
1 \multicolumn{2}{c|}{\multirow{2}{*}{Multi-Row and Col}}
```

2.3.5 美化公式

先前演示過一些常用的數學公式語法，但是卻沒有介紹過數學公式的排版。數學公式的排版也是由「環境」構成並且與表格十分相像但又更加複雜，首先目前最為推薦使用的是 `split` 環境。這個環境只做簡單的「單錨點」排版。具體而言可以參考下面方程

$$\begin{aligned} y &= \frac{1}{3} \times \frac{4}{9} \times \frac{36}{16} \\ &= \frac{1}{3} \times \frac{4}{9} \times \frac{9}{4} \\ &= \frac{1}{3} \\ \text{Ans: } &\frac{1}{3} \end{aligned} \tag{2.1}$$

在這個環境中 `&` 是錨點，所有公式的排版會以 `&` 對齊，並以換行符號換行。這種排版簡單、實用、常用，而且可以較為易於理解的與其他環境一同使用，如

$$f(x) = \begin{cases} f(x - 1) + 1, & x = \text{cond}, \\ 0, & x \neq \text{cond} \end{cases} \tag{2.2}$$

另一種更為強大但是複雜的環境是 `aligned`，這個環境可以定義排版的「欄位」和「錨點」。具體而言，我們可以參考下面代碼

$$\begin{aligned} x &= 1.1 & y &= 1.2 & z &= 1.3 \\ x1 &= 2.1 & y1 &= 2.2 & z1 &= 2.3 \\ \text{\left} && \text{\center} && \text{\right} & \end{aligned} \tag{2.3}$$

在這個排版環境中「奇數」的 `&` 是該直欄的「錨點」，而「偶數」的 `&` 則是和表格一樣的「分欄」符號。除 `aligned` 之外，`LATEX` 中還有 `align` 環境與 `align*`，只是這兩個環境是用來分別取代 `equation` 和 `\[\]`，而不像 `aligned` 是「數學環境」內的環境。而 `align` 與 `align*` 具體的區別也如上面所述，`align` 有以「行」為單位的編號而 `align*` 沒有編號。

除此之外，這裡還必須介紹 `\left` 與 `\right` 指令。在上面的演示中應該有

人注意到如果撰寫如下代碼

$$\left(\frac{\log_m a}{\log_m b} \right) = \log_a b$$

小括號 (...) 無法匹配其所應包裹的內容，這時的排版沒有錯誤但是難以說得上美觀。而 `\left` 與 `\right` 指令便是用來解決這樣的問題，這兩個指令總是成對出現、讀取該指令後一個用於“包圍”其他數學內容的字符¹，並且可以將這些字符延展到足以包圍其內容的數學元素。使用這兩條指令我們可以重寫代碼如下

$$\left(\frac{\log_m a}{\log_m b} \right) = \log_a b$$

這兩個指令對於符號的匹配實際上相當自由，我們甚至可以寫出下面的符號排版

$$\left. \begin{array}{lll} x=1.1 & y=1.2 & z=1.3 \\ x1=2.1 & y1=2.2 & z1=2.3 \end{array} \right\} \quad (2.4)$$

如果我們只需要單邊的括號，我們可以用下面讓 `\left` 或 `\right` 讀取 .，藉此我們甚至可以使用 `aligned` 實現如同 `cases` 環境效果的代碼如

$$Eq(x,y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \quad (2.5)$$

至此，如何美化與自定義數學模式下的“排版”的方法大致上都已經介紹完畢。實際上美觀的數學定義仍有許多排版以外的要素存在，包含字體、符號使用、上下標等等。但是這些美觀的需求往往都需要更多的經驗取得而非三言兩語所能知之。如果不知道如何展示美觀的數學表示，至少需要謹記 1) 盡可能的簡潔而且清晰、2) 上下標邏輯意義明確且統一、3) 一個公式僅表達一個概念(或觀點)。此外，若無必要勿使用長單詞，如有必要則使用`\mathrm` 調整常單詞字體增加可讀性。一般而言讀者更希望看到 `SINR` 而不是 `SINR` 或 `sinr`，以 `SINR` 而言排版上會更像多個各自具備意義的符號，而 `sinr` 則本身並不易於閱讀。因此我們更希望使用`\mathrm{}` 將應該被視為單詞排版的詞彙在數學模式下依然作為單詞(通常是函數)進行排版，從而提升可讀性並更加美觀。

¹通常而言這個字符包含 (,),[],{},|,|| 等等。

2.3.6 程式代碼

代碼環境就不過多贅述，上面的文本中已經使用過非常多次了，在這裡僅提供一些合適的文獻參考，如 Overleaf 官方文檔 [18] 和 wikibooks [19]。如果要說特別需要注意的地方，我建議使用 `\lstset` 設定全局的樣式 (本範中該設定於 `style.tex`)，並在局部使用 `[...]` 定義樣式，如：

```
1 def main() :  
2     print('Hello, world!')  
3 if __name__ == '__main__' :  
4     main()
```

若是需要 C 語言語法樣式則能改為 `[language=C]`

```
1 #include<stdio.h>  
2 int main (void){  
3     printf("Hello, world!\n");  
4     return 0;  
5 }
```

2.3.7 函數定義與演算法

有些人可能會需要在演算法裡面定義函數，在具體方法上面實際上有相當多種。這裡僅舉例一個最最簡單的方式，即「實際上演算法環境也只是一種特殊的浮動表格」所以我們可以單純的「描述」函數，在很多的字排版軟體中是用這種方法操作。其中一種方法參考至 [20]，其具體範例可以參考如下

Algorithm 2: 一個清楚的演算法標題

```
1 Function FunctionName(args):  
2     something;
```

2.4 參考文獻格式

對於理工系學生來說，IEEE 格式幾乎可以被視為標準的參考文獻格式，這時我們可以直接使用 `\bibliographystyle{IEEEtran}` 定義我們的參考文獻格式是符合 IEEE 的格式。但是 `IEEEtran` 格式多定義了一些格式上的描述，因此如果我們需要一些非其規定的特殊格式時可能無法簡單的修改而成。為滿足遇到特殊格式時的需求，這裡我們稍微討論一下如何「排出」參考文獻格式。

2.4.1 引用網頁

舉例而言，如果我們當前只有 `ieeetr` 格式，想要排出 `IEEEtran` 中網頁的排版效果。我們可以先參考 BibTeX 官網 [5] 中對於 `misc` 或這其他類型格式的介紹，再比較其與 IEEE 規範 [3,4] 之間的區別，直接用文字的方式「拼出」需要的格式。舉例而言，本模板 `ref.bib` 中 BibTeX 官網 [5] 的 BibTeX 格式文獻條目如下

```
1 @misc{bibtex,
2   title      = {The 14 BibTeX entry types},
3   author     = {Paperpile LLC},
4   year       = {Feb. 2025},
5   url        = {https://www.bibtex.com/e/entry-types/},
6   howpublished = {}
7 }
```

這個目錄在 `IEEEtran` 格式下可以直接編譯出符合 IEEE 規範的網頁參考文獻格式

```
P. LLC, "The 14 bibtex entry types," Feb. 2025. [Online]. Available: https://www.bibtex.com/e/entry-types/
```

可是當使用 `ieeetr` 時會編譯出如下明顯不符合規範的格式

```
P. LLC, "The 14 bibtex entry types," Feb. 2025.
```

如果我們真的需要用 `ieeetr` 或是其他風格生成符合格式的參考文獻可以參考類似於 [21] 的所定義的 BibTeX 如下

```
1 @misc{bibtexAlt,
2   title      = {The 14 BibTeX entry types},
3   author     = {Paperpile LLC},
4   year       = {Accessed: Feb. 2025},
5   note       = {[Online.] Available: \url{https://www.bibtex.com/e/entry-
6   types/}},
7   howpublished = {}
```

具體而言，我們手動加入了“`Accessed:`”和“`[Online.] Available:`”等格式中所規範的描述方式。使之經由 BibTeX 編譯排列之後可以生成合理的參考文獻格式。其他不同類型的

參考文獻格式在沒有可用的預定義風格(格式)時，也可以用相似的手法生成正確的參考文獻格式。

2.4.2 引用 arXiv

另一個常見的引用來源是 arXiv 網站 [22]，這是許多重要的開源文獻以及論文預印本所使用的發表平台，其主要內容包含數學、物理、工程與人工智能。特別是對於人工智能領域，該領域中許多重要的研究結果包括但不限於 YOLO [23] 和 Transformer [24] 都發表於該平台。然而作為純電子出版的開源平台，其合理的引用格式便成為了一項有待議論的問題。

如上面兩篇文章 [23], [24] 在 arXiv 平台上所提供的 BibTeX 是以網頁的形式引用，但是若參考一些主流期刊的引用格式 [25] 則並非如此。在此，我們也基於 Google Scholar 所提供的 BibTeX 模板提供 arXiv 文獻引用格式的“拼湊”方式。首先，先注意到 [25] 中對於一篇 arXiv 文章 [26] 的參考排版結果為

```
Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOx: Exceeding YOLO series in 2021,”  
2021, arXiv:2107.08430.
```

這個格式顯然不是網頁格式(由 arXiv 平台自動生成)的引用。而現在此引用 [26] 使用的是 Google Scholar 所生成的 BibTeX

```
1 @article{ge2021yolox,  
2   title={Yolox: Exceeding yolo series in 2021},  
3   author={Ge, Zheng and Liu, Songtao and Wang, Feng and Li, Zeming and Sun,  
4         Jian},  
5   journal={arXiv preprint arXiv:2107.08430},  
6   year={2021}  
}
```

其排版的生成結果為

```
Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” arXiv  
preprint arXiv:2107.08430, 2021.
```

我們可以注意到，這兩者之間的排版差異並不是多一些字，而是排版順序也不一樣。此時可以參考 IEEE 文檔 [3] 中對引用 arXiv 文獻的格式說明，再參考 [4], [5], [27] 調整

輸入參數，於是我們可以得到下面的 BibTeX

```
1 @unpublished{ge2021yoloxAlt,  
2   title      = {{YOLO}x: Exceeding {YOLO} series in 2021},  
3   author     = {Ge, Zheng and Liu, Songtao and Wang, Feng and Li, Zeming  
and Sun, Jian},  
4   year       = {2021},  
5   note        = {\itshape arXiv:2107.08430}  
6 }
```

其中由於原先的 `note` 沒有斜體效果，因此這裡使用了 `\itshape` 指令用於生成斜體字體，這個 BibTeX 的排版結果可以參考 [28]

```
Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOx: Exceeding YOLO series in 2021,”  
2021, arXiv:2107.08430.
```

此時的排版格式便已經與 IEEE 期刊 [25] 與 IEEE 規範 [3], [4] 完全一致。其他需要的參考文獻格式也可以用相似的方式藉由這些參考資料 [3], [4], [5], [27] 產生。



第三章 Conclusion

這個範本本身是國立臺北科技大學的學位論文格式範本，同時也是對於 L^AT_EX 初學者與初次撰寫論文的研究者/研究生的引導文章。在本文中，一開始我們先介紹了如何使用這個範本，並且進一步說明了這個範本中所定義的一些自動化指令與環境。接著，我們藉由一篇短文介紹了 L^AT_EX 的簡要使用說明，以自指形式排版並說明了 L^AT_EX 撰寫論文時常用的指令。針對需要進一步瞭解如何使用 L^AT_EX 美化排版，以及對於不知道學位論文應該如何起頭的讀者，這個範本中也使用了一個章節說明其中的細節並提供具體的引導。這篇總結也是一樣，期許為不知道如何文章撰寫讀者提供足以作為參考的指引。在本文中，除正文之外也存在一些為了「有趣」而使用的文案。雖然在實際的論文中這樣的文案並不合適，但是對於新生的使用者而言希望能理解—研究是嚴謹但是並非嚴肅的事情。



參考文獻

- [1] 教務處, “法規一覽表,”臺北科技大學, Feb. 2025, <https://oaa.ntut.edu.tw/p/412-1008-12780.php?Lang=zh-tw>.
- [2] 圖書館, “國立臺北科技大學博碩士論文系統-下載區,”臺北科技大學, Feb. 2025, <https://cloud.ncl.edu.tw/ntut/download.php>.
- [3] IEEE, “Reference guide,” Feb. 2025, https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf.
- [4] M. Shell, “How to use the ieeetran bibtex style,” IEEE, Feb. 2025, <https://tw.mirrors.cicku.me/ctan/macros/latex/contrib/IEEEtran/bibtex/IEEEtran bst HOWTO.pdf>.
- [5] P. LLC, “The 14 bibtex entry types,” Feb. 2025. [Online]. Available: <https://www.bibTEX.com/e/entry-types/>
- [6] 教務處, “表單一覽表,”臺北科技大學, Feb. 2025, <https://oaa.ntut.edu.tw/p/412-1008-12839.php?Lang=zh-tw>.
- [7] 圖書館, “臺北科技大學圖書館－研究生畢業專區,”臺北科技大學, Feb. 2025, <https://lib.ntut.edu.tw/sp.asp?xdurl=mp100/SLP.asp&ctNode=1604&CtUnit=460&BaseDSD=7&mp=100>.
- [8] C.-W. Cho and M.-S. Pan, “Resource scheduling in MU-MIMO and NOMA enabled integrated access and backhaul networks,” *IEEE Open Journal of the Communications Society*, vol. 6, pp. 551–559, 2025.
- [9] ——, “Efficient downstream data dissemination in IAB networks with DAG-based topology,” *Wireless Networks*, vol. 31, no. 2, pp. 1735–1746, 2025.
- [10] ——, “Downlink radio resource scheduling for OFDMA systems with hybrid beamforming,” *Wireless Networks*, vol. 28, no. 1, pp. 273–286, 2022.

- [11] 中文維基教科書, “Latex/數學公式 - 維基教科書, 自由的教學讀本,” Feb. 2025, <https://zh.wikibooks.org/zh-tw/LaTeX/□□□□>.
- [12] Overleaf, “Bibtex bibliography styles - overleaf, online latex editor,” Feb. 2025. [Online]. Available: https://www.overleaf.com/learn/latex/Bibtex_bibliography_styles
- [13] M. Shell, “How to use the IEEEtran BibTeX style,” IEEE, Feb. 2025, <https://tw.mirrors.cicku.me/ctan/macros/latex/contrib/IEEEtran/bibtex/IEEEtran bst HOWTO.pdf>.
- [14] gernot, “Right alignment for p{length} box in tabular [duplicate],” Stack Exchange Inc, Feb. 2025, <https://tex.stackexchange.com/questions/338009/right-alignment-for-plength-box-in-tabular>.
- [15] G. Medina, “How does the \newcolumntype command work?” Stack Exchange Inc, Feb. 2025, <https://tex.stackexchange.com/questions/257128/how-does-the-newcolumntype-command-work>.
- [16] Overleaf, “Tables - overleaf, online latex editor,” Feb. 2025, <https://www.overleaf.com/learn/latex/Tables>.
- [17] MEMO123, “使用 latex，跨行、跨列複雜表格的撰寫方法，使用\multicolumn 和\multirow,” Blogger, Feb. 2025, <https://jhengjyun.blogspot.com/2011/01/latexmulticolumnmultirow.html>.
- [18] Overleaf, “Code listing - overleaf, online latex editor,” Feb. 2025, https://www.overleaf.com/learn/latex/Code_listing.
- [19] Wikibooks, “Latex/source code listings - wikibooks, open books for an open world,” Feb. 2025, https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings.
- [20] S. Khan, “algorithm2e - how to use function in latex algorithm? - tex - latex stack exchange,” Stack Exchange Inc, Feb. 2025, https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings.
- [21] P. LLC, “The 14 bibtex entry types,” Accessed: Feb. 2025, [Online.] Available: <https://www.bibtex.com/e/entry-types/>.
- [22] C. University, “arxiv.org e-print archive,” Feb. 2025, <https://arxiv.org/>.

- [23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2016. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [25] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, “Yolov8-qsd: An improved small object detection algorithm for autonomous vehicles based on yolov8,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–16, 2024.
- [26] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [27] P. LLC, “Bibtex bibliography style: Ieeetran,” Feb. 2025. [Online]. Available: <https://www.bibtex.com/s/bibliography-style-ieeetran-ieeetran/>
- [28] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOx: Exceeding YOLO series in 2021,” 2021, *arXiv:2107.08430*.