

# 대학원생을 위한 실전 L<sup>A</sup>T<sub>E</sub>X 튜토리얼

## 기초부터 논문 작성, 그리고 고급 활용법까지

Seonghoon Jeong

System and Network Security Lab. (SNSec Lab.)  
Division of Artificial Intelligence Engineering, Sookmyung Women's University

March 19, 2026







# Why L<sup>A</sup>T<sub>E</sub>X? (Word vs L<sup>A</sup>T<sub>E</sub>X 비교)

## MS Word / 한글

**WYSIWYG** (What You See Is What You Get)

- 내용과 서식을 동시에 편집
- 진입 장벽이 낮음
- **단점:** 수식 작성이 번거롭고 일관된 다중 문서 서식 유지가 어려움, 큰 문서에서 버벅임

VS

## LaTeX

**WYSIWYM** (What You See Is What You Mean)

- 내용 작성과 레이아웃 렌더링의 완전한 분리
- 압도적으로 아름다운 수식 품질
- 상호 참조 및 참고문헌 자동 관리
- 학회/저널 템플릿 완벽 지원

# Hello World: 협업 도구 Overleaf 소개

가장 빠르고 쉬운 시작: Overleaf (<https://www.overleaf.com/>) 

- 로컬 PC에 무거운 프로그램을 설치할 필요 없음 (웹 브라우저에서 실행)
- 구글 문서(Google Docs)처럼 여러 명과 실시간 동시 편집 및 협업 가능
- 프로젝트 전체의 히스토리 관리 및 자동 저장 지원
- 수많은 템플릿(IEEE, ACM 등)을 클릭 한 번으로 불러올 수 있음

“지금 바로 *Overleaf*에 접속해서 회원가입 후 *Blank Project*를 만들어보세요!”

# 기본적인 영문 문서 구조 (Article)

## 코드 (Code)

```
1 \documentclass{article}
2 \title{Introduction to \LaTeX}
3 \author{Seonghoon Jeong}
4 \date{March 2026}
5
6 \begin{document}
7 \maketitle
8
9
10 \section{First Section}
11 Hello world. This is a basic \LaTeX{}
12 document. It cleanly separates content
13 formatting.
14
15 \subsection{A Subsection}
16 Here is another paragraph with a short
17 equation  $E=mc^2$ .
18 \end{document}
```

## 출력 (Rendered PDF)

# Introduction to $\LaTeX$

Seonghoon Jeong

March 2026

## 1 First Section

Hello world. This is a basic  $\LaTeX$  document. It cleanly separates content formatting.

### 1.1 A Subsection

Here is another paragraph with a simple equation  $E = mc^2$ .

# 간단한 텍스트 입력과 한글 팩(kotex) 적용하기

기본적으로  $\LaTeX$ 는 영문을 기준으로 동작. 한국어 출력을 위해 kotex 패키지 로드 필요  
Preamble(문서 설정부)에 `\usepackage{kotex}` 추가

## 코드 (Code)

```
1 \documentclass{article}
2 \usepackage{kotex}
3
4 \begin{document}
5 Hello World!
6
7 안녕하세요,  $\LaTeX$ 의 세계에 오신 것을 환영합니
  다.
8 \end{document}
```

## 출력 (Rendered PDF)

Hello World!  
안녕하세요,  $\LaTeX$ 의 세계에 오신 것을 환영합니다.

# Outline

- 1 Intro (동기 부여와 첫걸음)
- 2 문서의 뼈대 잡기 (역공학 어프로치)**
- 3 실질적인 콘텐츠 채우기 (핵심 문법)
- 4 논문의 완성
- 5 Pro-level 실전 팁 & 고급 세팅

# LaTeX 학습법: 템플릿 뜯어보기

## 일반적인 언어 학습과 다른 “역공학(Reverse Engineering)” 어프로치

- C나 파이썬처럼 처음부터 문법을 하나씩 공부하면 지루하고 포기하기 쉬움.
- 이미 잘 만들어진 템플릿 파일을 다운받아, 글씨만 바꿔보며 배우는 것이 가장 효율적.

### 우리의 목표

“맨땅에서 코딩하지 않고, 이미 완성된 구조 위에서 내 논문의 내용을 채워 넣자!”

# 논문 템플릿(IEEE, ACM 등) 다운로드 및 활용

## 어디서 템플릿을 구하나요?

- 자신이 제출할 타겟 학회, 저널의 공식 홈페이지 (Information for Authors)
- Overleaf의 템플릿 갤러리 검색
- 선배님과 지도교수님

## Overleaf에서 템플릿 열기

- ① **.zip** 템플릿 파일 다운로드
- ② Overleaf 메인에서 **New Project** → **Upload Project**
- ③ **.zip** 업로드 후 메인 컴파일 파일 (**.tex**) 선택

## 대표적인 템플릿

- **IEEEtran**: 전기/전자/컴퓨터 공학 표준
- **acmart**: ACM 주관 학회 표준
- **CVPR/ICCV/ECCV**: 컴퓨터 비전 템플릿
- **NeurIPS/ICLR**: AI 관련 학회 템플릿

# 논문 템플릿 다운로드 받는 곳

## 주요 출판사 (Journals)

- IEEE Template Selector
- ACM Primary Templates
- Elsevier  $\LaTeX$  Instructions
- Springer Nature Support

## 보안 주요 학회 (Big 4)

- IEEE S&P / IEEE (Templates)
- USENIX Security (Templates)
- ACM CCS / ACM (Templates)
- NDSS Author Templates

## AI / Vision 주요 학회

- CVPR / ICCV / ECCV Guidelines
- NeurIPS / ICLR / ICML Info
- AAAI (Overleaf  $\LaTeX$ )

### 꿀팁: arXiv 활용

이미 출판된 논문의 **LaTeX** 소스 전체를 arXiv에서 다운로드하여 형식을 참고할 수 있습니다!

# 실습: IEEE Transactions on Industrial Informatics 저널 템플릿 다운로드

## IEEE Template Selector ← 클릭

### IEEE TEMPLATE SELECTOR



Start typing the title of your target journal, transaction, or letter publication and select the correct option from the dropdown menu.

Select name of the publication ▼

🔍

IEEE Transactions on Industrial Informatics

# 역공학 실습: 코드-PDF 비교 분석

- 1 `\subsection{Other Recommendations}`
- 2 Use one space after periods and colons. Hyphenate complex modifiers: ```zero-field-cooled magnetization.``` Avoid dangling participles, such as, ```Using \eqref{eq}, the potential was calculated.``` [It is not clear who or what used `\eqref{eq}`.] Write instead, ```The potential was calculated by using \eqref{eq},``` or ```Using \eqref{eq}, we calculated the potential.```
- 3
- 4 Use a zero before decimal points: ```0.25,``` not ```.25.``` Use ```cm3,``` not ```cc.``` Indicate sample dimensions as ```0.1 cm  $\times$  0.2 cm,``` not ```0.1  $\times$  0.2 cm2.``` The abbreviation for ```seconds``` is ```s,``` not ```sec.``` Use ```Wb/m2,``` or ```webers per square meter,``` not ```webers/m2.``` When expressing a range of values, write ```7 to 9``` or ```7--9,``` not ```7 $\sim$ 9.```
- 5
- 6 A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American


## B. Other Recommendations

Use one space after periods and colons. Hyphenate complex modifiers: “zero-field-cooled magnetization.” Avoid dangling participles, such as, “Using (1), the potential was calculated.” [It is not clear who or what used (1).] Write instead, “The potential was calculated by using (1),” or “Using (1), we calculated the potential.”

Use a zero before decimal points: “0.25,” not “.25.” Use “cm<sup>3</sup>,” not “cc.” Indicate sample dimensions as “0.1 cm × 0.2 cm,” not “0.1 × 0.2 cm<sup>2</sup>.” The abbreviation for “seconds” is “s,” not “sec.” Use “Wb/m<sup>2</sup>” or “webers per square meter,” not “webers/m<sup>2</sup>.” When expressing a range of values, write “7 to 9” or “7–9,” not “7~9.”

A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American English, periods and commas are within quotation marks, like “this period.” Other punctuation is “outside!” Avoid contractions; for example, write “do not” instead of “don’t.” The serial comma is preferred: “A, B, and C” instead of “A, B and C.”

# 실습: arxiv 논문의 원본 소스코드 다운로드


Cornell University

We gratefully acknowledge support from the Simons Foundation, [member institutions](#), and all contributors.

Donate

---

arXiv

> cs > arXiv:1706.03762

Search...

All fields

Search

**Computer Science > Computation and Language**

*[Submitted on 12 Jun 2017 (v1), last revised 2 Aug 2023 (this version, v7)]*

## Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder–decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English–to–German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English–to–French translation task, our model establishes a new single–model state–of–the–art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

**Access Paper:**

[View PDF](#)  
[HTML \(experimental\)](#)  
[TeX Source](#)

view license

---

Current browse context:

**cs.CL**  
[< prev](#) | [next >](#)  
[new](#) | [recent](#) | [2017-06](#)

Change to browse by:

[cs](#)  
[cs.LG](#)

---

**References & Citations**

[NASA ADS](#)  
[Google Scholar](#)  
[Semantic Scholar](#)

---

[123 blog links](#) (what is this?)

---

[DBLP – CS Bibliography](#)

# Outline

- 1 Intro (동기 부여와 첫걸음)
- 2 문서의 뼈대 잡기 (역공학 어프로치)
- 3 실질적인 콘텐츠 채우기 (핵심 문법)**
- 4 논문의 완성
- 5 Pro-level 실전 팁 & 고급 세팅

# 텍스트 서식과 리스트 (Formatting & Lists)

## 코드 (Code)

```

1 Some of the \textbf{greatest}
2 discoveries in \underline{
  science}
3 were made by \textbf{\textit{
  accident}}}.
4
5 \begin{itemize}
6   \item First item
7   \item Second item
8 \end{itemize}
9
10 \begin{enumerate}
11   \item First item
12   \item Second item
13 \end{enumerate}

```

## 출력 (Rendered PDF)

Some of the **greatest** discoveries in science were made by *accident*.

- First item
- Second item

1. First item
2. Second item

## 단락과 줄바꿈 포인트

엔터 한 번은 아무 의미 없는 공백으로 처리되며, 엔터를 두 번 연속으로 쳐야 빈 줄이 생기며 새로운 단락이 시작됩니다.

# 문서 구조와 목차 생성 (Sections & TOC) (1/2)

## 1. 섹션(Section) 나누기

- `\section{...}` : 장 (1, 2, ...)
- `\subsection{...}` : 절 (1.1, ...)
- `\subsubsection{...}` : 소절
- \* : 번호 생략 (`\section*`)

## 2. 목차 자동 생성

- `\tableofcontents` 입력
- 목차 최신화를 위해 2회 컴파일 (Overleaf는 자동 처리)

## 연습용 코드 (Code)

```

1 \begin{abstract}
2 This research explores the principles
3 of \LaTeX{} typesetting in academia.
4 \end{abstract}
5
6 \section{Introduction}
7 Starting point of our document.
8 \subsection{Background}
9 Brief overview of the field.
10 \subsubsection{Related Work}
11 Existing methodologies reviewed.

```

[다음 페이지에서 결과 확인 →]

# 문서 구조와 목차 생성 (Sections & TOC) (2/2)

## 1. 섹션(Section) 나누기

- `\section{...}` : 장 (1, 2, ...)
- `\subsection{...}` : 절 (1.1, ...)
- `\subsubsection{...}` : 소절
- `*` : 번호 생략 (`\section*`)

## 2. 목차 자동 생성

- `\tableofcontents` 입력
- 목차 최신화를 위해 2회 컴파일 (Overleaf는 자동 처리)

## 출력 결과 (Rendered PDF)

### Abstract

This research explores the fundamental principles of  $\text{\LaTeX}$  typesetting and its practical applications in academic writing.

## 1 Introduction

This is the starting point of our document.

### 1.1 Background

We provide a brief overview of the research field.

#### 1.1.1 Related Work

Existing methodologies are reviewed.

# 패키지 활용: 기능의 무한 확장 (`\usepackage`)

LaTeX의 강력함은 ‘패키지’에서 나옵니다.

## 1. 사용법 및 옵션 설정

```
1 \usepackage[margin=1in]{geometry}
2 \usepackage{amsmath, graphicx}
```

## 2. 도움말 찾기: CTAN(크탄)

- <https://www.ctan.org/>
- 전 세계 6,000개 이상의 패키지 저장소
- 패키지 이름 검색 → **Package documentation** 클릭 (가장 정확한 설명서!)
- 예) `bytefield` — Create illustrations for network protocol specifications

## 대표 패키지들

- **amsmath**: 논문 수식의 필수품
- **graphicx**: 그림 삽입 (`includegraphics`)
- **geometry**: 페이지 여백 조절
- **hyperref**: 클릭 가능한 링크
- **kotex**: 한글 지원 필수

# 논문에 꼭 들어가야 할 것들

### 그림 (Figure)

- `figure` 환경
- `includegraphics` 옵션
- 이미지 삽입 예제

### 캡션 & 레이블

- `caption`, `label`
- `ref`, `pageref`
- 자동 번호 매기기

### 표 (Table)

- `tabular` 환경
- 테두리(borders) 추가
- 캡션 및 참조

### 수식 (Math)

- 인라인/디스플레이 모드
- `equation` 환경
- 주요 수식 문법

# 그림(Figure) 넣기

## 코드 (Code)

```

1 \usepackage{graphicx}
2
3 \begin{figure}[h]
4   \centering
5   \includegraphics[width=0.8\textwidth]
6     {rsrc/03_smu_winter26.jpg}
7   \caption{A photo taken at SMU.}
8   \label{fig:smu}
9 \end{figure}
10
11 As shown in Figure \ref{fig:smu}...
```

위치 옵션 **[h, t, b, p]** :

here · top · bottom · page

! 추가 시 강제 적용 (e.g. **[h!]** )

## 출력 결과



Figure 1: A photo taken at SMU.

# 캡션, 레이블, 참조 (Captions, Labels & References)

## 코드 (Code)

```

1 \begin{figure}[h]
2   \centering
3   \includegraphics[width=0.75\textwidth]
4     {rsrc/03_smu_winter26.jpg}
5   \caption{A nice plot.}
6   \label{fig:smu2}
7 \end{figure}
8
9 As you can see in Figure~\ref{fig:smu2},
10 the function grows near the origin.
11 This example is on page~\pageref{fig:smu2}.

```

## 실전 Prefix 관례

```

fig:architecture
tab:results
eq:loss
sec:method

```

## 명령어 설명

- `\caption{...}`: 그림 아래 설명 텍스트
- `\label{fig:...}`: 고유 이름 부여
- `\ref{... }`: 번호(1, 2, ...) 호출
- `\pageref{...}`: 페이지 번호 호출

꿀팁: `~` (Tilde) 의 역할

`Figure~\ref{...}`

물결표(`~`)는 줄바꿈 방지 공백입니다. 문장 끝에서 "Figure"만 남고 숫자가 다음 줄로 넘어가는 참사를 막아줍니다.

표(Table) 기초: **tabular** 환경

## 가장 기본적인 표

```

1 \begin{center}
2   \begin{tabular}{c c c}
3     cell1 & cell2 & cell3 \\
4     cell4 & cell5 & cell6 \\
5     cell7 & cell8 & cell9
6   \end{tabular}
7 \end{center}

```

출력 결과:

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

열 서식 옵션 **{...}**

- **c**: 가운데 정렬 (center)
- **l**: 왼쪽 정렬 (left)
- **r**: 오른쪽 정렬 (right)
- **&**: 셀 구분자
- **\\**: 행 끝 (줄 바꿈)

## 실전 꿀팁

<https://www.tablesgenerator.com/>  
엑셀처럼 그리고 코드 복사!

# 표(Table) 테두리 추가 (Adding Borders)

## 전체 테두리

```

1 \begin{center}
2   \begin{tabular}{||c c c c||}
3     \hline
4     Col1 & Col2 & Col3 & Col4 \\ \ [0.5ex]
5     \hline\hline
6     1 & 6 & 87837 & 787 \\ \ \ \ \hline
7     2 & 7 & 78 & 5415 \\ \ \ \ \hline
8     3 & 545 & 778 & 7507 \\ \ \ \ \hline
9     4 & 545 & 18744 & 7560 \\ \ \ \ \hline
10    5 & 88 & 788 & 6344 \\ \ [1ex] \hline
11  \end{tabular}
12 \end{center}

```

## 출력 결과:

Col1	Col2	Col3	Col4
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

| : 세로 선 (vertical rule)

\hline : 가로 선 (horizontal rule)

\hline\hline : 이중 선

# 표 캡션 & 레이블 (Table Caption & Label)

## 코드 (Code)

```

1 Table~\ref{table:data} shows the
  experimental results.
2
3 \begin{table}[h!]
4   \centering
5   \caption{The table caption goes here.}
6   \label{table:data}
7   \begin{tabular}{||c c c c||}
8     \hline
9     Col1 & Col2 & Col3 & Col4 \\ \ [0.5ex]
10    \hline\hline
11    1 & 6 & 87837 & 787 \\ \
12    2 & 7 & 78 & 5415 \\ \
13    5 & 88 & 788 & 6344 \\ \ [1ex]
14    \hline
15  \end{tabular}
16 \end{table}

```

## 출력 결과:

[본문 영역] Table 1 shows the experimental results.

Table 1: The table caption goes here.

Col1	Col2	Col3	Col4
1	6	87837	787
2	7	78	5415
5	88	788	6344

`\caption` 의 위치:

- 그림: 아래 ( `\end{figure}` 전)
- 표: 위 ( `\begin{tabular}` 전)
- \* 표준 관례를 따르는 것이 전문적임!

# 수식(Math) 작성의 기초

## 1. 인라인(Inline) 모드 $\$ \dots \$$

- 1 The mass-energy equivalence
- 2  $E=mc^2$  was discovered in 1905.

출력: The mass-energy equivalence  $E = mc^2$  was discovered in 1905.

## 2. 디스플레이(Display) 모드

```

1 \[ E = mc^2 \]
2
3 \begin{equation}
4   E = mc^2
5   \label{eq:einstein}
6 \end{equation}

```

출력 (번호 있음):  $E = mc^2$  (1)

## 주요 수식 문법

- 위첨자:  $x^2 \rightarrow x^2$
- 아래첨자:  $x_i \rightarrow x_i$
- 분수:  $\frac{a}{b} \rightarrow \frac{a}{b}$
- 시그마:  $\sum_{i=1}^n \rightarrow \sum_{i=1}^n$
- 적분:  $\int_a^b \rightarrow \int_a^b$
- 그리스문자:  $\alpha, \beta \rightarrow \alpha, \beta$
- 벡터:  $\mathbf{w} \rightarrow \mathbf{w}$
- 노름:  $\|w\|^2 \rightarrow \|w\|^2$

# 실전 딥러닝 수식 렌더링 예시

## Scaled Dot-Product Attention (Transformer)

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

**Softmax**

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

## Cross-Entropy Loss

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log \hat{y}_{ic}$$

## Batch Normalization

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad y^{(k)} = \gamma \hat{x}^{(k)} + \beta$$

## Adam Optimizer

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

사용 패키지: `amsmath` (필수), `amssymb` (기호),  
`mathtools` (확장)

# 실전 딥러닝 수식 코드 (L<sup>A</sup>T<sub>E</sub>X Source)

## Attention & Softmax

```

1 \begin{equation*}
2   \mathrm{Attention}(Q,K,V) = \mathrm{softmax}\!\left(
3     \frac{QK^{\top}}{\sqrt{d_k}}\right)V
4 \end{equation*}
5 \begin{equation*}
6   \sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}
7 \end{equation*}

```

## Cross-Entropy Loss

```

1 \begin{equation*}
2   \mathcal{L} = -\frac{1}{N}\sum_{i=1}^N\sum_{c=1}^C y_{ic}\log\hat{y}_{ic}
3 \end{equation*}

```

## Batch Normalization

```

1 \begin{equation*}
2   \hat{x}^{(k)} = \frac{x^{(k)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad
3   y^{(k)} = \gamma \hat{x}^{(k)} + \beta
4 \end{equation*}

```

## Adam Optimizer

```

1 \begin{align*}
2   m_t &= \beta_1 m_{t-1} + (1-\beta_1)g_t \quad \backslash\backslash
3   v_t &= \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \quad \backslash\backslash
4   \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \quad \backslash\backslash \hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad \backslash\backslash
5   \theta_t &= \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t
6 \end{align*}

```

# Outline

- 1 Intro (동기 부여와 첫걸음)
- 2 문서의 뼈대 잡기 (역공학 어프로치)
- 3 실질적인 콘텐츠 채우기 (핵심 문법)
- 4 논문의 완성**
- 5 Pro-level 실전 팁 & 고급 세팅

# BibTeX과 References(참고문헌) 관리

논문에서 가장 귀찮은 “참고문헌 양식 맞추기”를 완벽하게 자동화할 수 있습니다.

## 1. **references.bib** 파일 생성

```
1 @article{vaswani2017attention,  
2   title={Attention is all you need},  
3   author={Vaswani, Ashish et al.},  
4   journal={Advances in neural  
5     information  
6     processing systems},  
7   volume={30},  
8   year={2017}  
9 }
```

## 2. 구글 스칼라(Google Scholar) 활용

- 논문 검색 후 “인용(Cite)” 버튼 클릭
- 하단의 “BibTeX” 클릭
- 텍스트 전체를 복사해서 **.bib** 파일에 붙여넣기

## 3. 본문에서 인용하기

- `\cite{vaswani2017attention}`
- 문서 마지막에 양식만 지정하면 알아서 번호와 포맷이 정렬됩니다.

# BibTeX 실습 (문서에 통합하기)

## 메인 파일 (main.tex)

```

1 \documentclass{article}
2
3 \begin{document}
4 \section{First Section}
5 Hello world. This is a basic \LaTeX{}
6 document. We cite the Transformer
7 model~\cite{vaswani2017attention}.
8
9 \bibliographystyle{plain}
10 \bibliography{references}
11 \end{document}

```

## 참고문헌 (references.bib)

```

1 @article{vaswani2017attention,
2   title={Attention is all you need},
3   author={Vaswani, Ashish et al.},
4   journal={NeurIPS},
5   volume={30},
6   year={2017}

```

## 출력 (Rendered PDF)

### 1 First Section

Hello world. This is a basic  $\LaTeX$  document. We cite the Transformer model [1].

### References

[1] Ashish Vaswani et al. Attention is all you need. *NeurIPS*, 30, 2017.

### 컴파일 순서 주의!

로컬 환경에서는

`latex` → `bibtex` → `latex` → `latex` 순서로  
실행해야 [?]가 숫자로 바뀝니다. (Overleaf는 자동 처리)

# Outline

- 1 Intro (동기 부여와 첫걸음)
- 2 문서의 뼈대 잡기 (역공학 어프로치)
- 3 실질적인 콘텐츠 채우기 (핵심 문법)
- 4 논문의 완성
- 5 Pro-level 실전 팁 & 고급 세팅**

# 로컬 환경 세팅: TeXLive + VSCode + L<sup>A</sup>T<sub>E</sub>X Workshop

## 왜 Overleaf 대신 로컬(내 PC) 환경이 필요한가요?

- 논문 마감 직전, 전 세계 연구자들이 몰릴 때 Overleaf 서버가 마비되는 경우가 빈번함.
- 대형 그림 파일이나 수십 개의 챕터가 있는 무거운 문서(학위논문 등) 컴파일 속도.
- 인터넷이 없는 오프라인 환경에서도 작업 가능.

## 추천하는 세팅 조합

- ① **LaTeX 컴파일러 엔진:** TeX Live (Windows/Linux) 또는 MacTeX (macOS). 용량이 매우 큽니다(약 4-8GB).
- ② **텍스트 에디터:** Visual Studio Code (VSCode)
- ③ **VSCode 확장 프로그램:** L<sup>A</sup>T<sub>E</sub>X Workshop (저장 시 자동 컴파일 및 PDF 뷰어 제공)

# 운영체제별 L<sup>A</sup>T<sub>E</sub>X 컴파일러 설치

## Linux (Ubuntu/Debian 환경)

- 1 `sudo apt update`
- 2 `sudo apt install texlive-full`

## Windows (TeX Live)



### TeX Live on Windows

#### Contents

- [Easy install](#)
- [Other options](#)
- [64- and 32-bit binaries](#)
- [Perl and TeX/Tk on Windows](#)
- [Multiple TeX distributions](#)
- [Security](#)
- [Administrator privileges](#)
- [Network installations and the TeX Live launcher](#)
- [Fixing permission problems, especially with texmf-local](#)
- [Non-ascii characters in TEXDIR](#)
- [Troubleshooting other problems](#)

#### Easy install

When installing from the internet, we recommend downloading and running [install-tl-windows.exe](#).

## macOS (MacTeX)



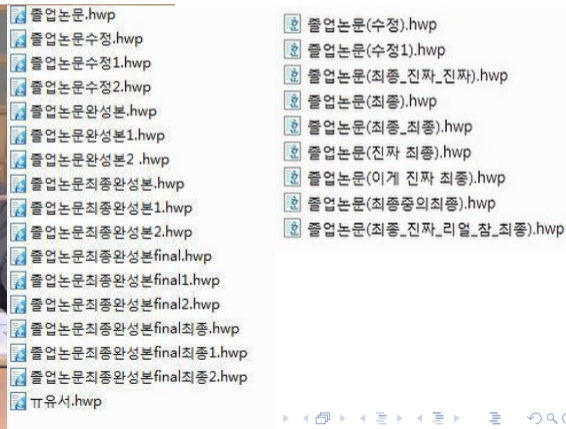
### Downloading

The MacTeX-2026 package can be obtained over the internet for free by clicking on the following link:

[MacTeX.pkg](#) [ approximately 6.4G - 02 March 2026 ]

# 버전 관리 및 `latexdiff` (1/3)

LaTeX으로 작성된 manuscript는 일반적인 코드 프로젝트와 다를 바 없다.  
→ 체계적인 버전 관리(Version Control)의 중요성



# 버전 관리 및 `latexdiff` (2/3)

## 1. 불필요한 빌드 파일 제외하기

$\text{\LaTeX}$ 을 컴파일하면 `*.aux`, `*.log` 등 수많은 임시 파일이 생성됩니다. 이를 Git 추적에서 제외하기 위해 공식 설정 파일을 활용합니다.

공식 `TeX.gitignore`

GitHub 공식 `TeX.gitignore` 다운로드 (클릭)  
프로젝트 폴더에 `.gitignore` 로 저장.

## 2. 저장소 초기화 및 첫 커밋

```
1 $ git init
2 $ vi main.tex
3 $ git add .gitignore main.tex
4 $ git commit -m "initial commit"
```

## 3. 논문 리비전

```
1 $ vi main.tex
2 $ git commit -am "revised commit"
```

버전 관리 및 **latexdiff** (3/3)

## 4. 커밋 히스토리(해시값) 확인

```

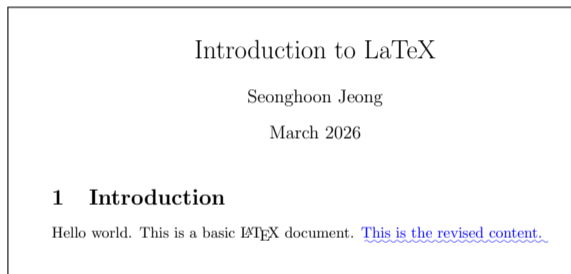
1 $ git log
2
3 commit b3369de4502b85d1ab4d5b356c3a5dc2e88b393e (HEAD
   -> main)
4 Author: Seonghoon Jeong <seonghoon@shjeong.net>
5 Date: Thu Mar 19 02:11:40 2026 +0900
6
7     revised commit
8
9 commit 9f4534f068f46d3e91a3c2e0295924c16dd11d06
10 Author: Seonghoon Jeong <seonghoon@shjeong.net>
11 Date: Thu Mar 19 02:11:25 2026 +0900
12
13     initial commit

```

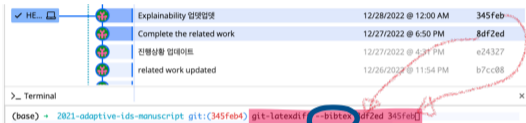
## 5. 변경 사항만 한눈에 보이는 PDF 자동 생성

```
1 $ git-latexdiff 9f4534f b3369de --bibtex
```

## Reviewer를 위한 완벽한 Diff PDF 완성



# git-latexdiff 실제 예시



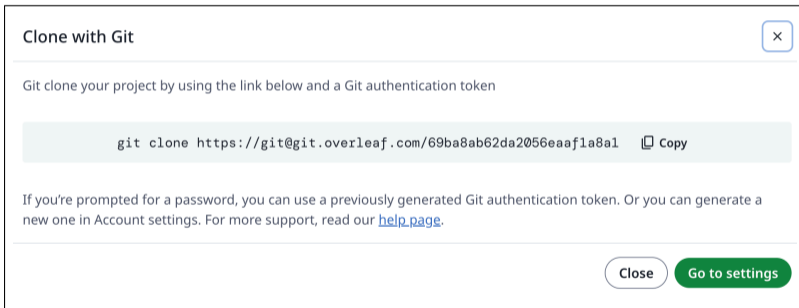
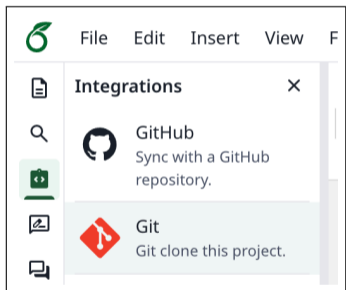
커맨드 입력 후 곧바로 diff 열람됨

Bibtex 내 변경사항 역시 출력됨 (--bibtex 스위치)

[38] T.-N. Hoang and D. Kim, "Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders," *Vehicular Communications*, vol. 38, p. 10520, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214209622000675> "OBD-II PIDs(Wikipedia)," , accessed Jan. 1, 2023.



# Overleaf 프로젝트와 Git을 통한 연동



# texcount — 분량 제한이 있는 경우. 혹은 생산성 측정을 위하여

```

(base) + manuscript git:(master) | texcount TXT_TII-XX-XXXX.tex
File: TXT_TII-XX-XXXX.tex
Words in text: 4490
words-in-headers: 80
Words outside text (captions, etc.): 380
Number of headers: 36
Number of floats/tables/figures: 7
Number of math inlines: 144
Number of math displayed: 5
Subcounts:
text+headers+captions (#headers/#floats/#inlines/#displayed)
197+11+0 (2/0/0/0) _top_
788+127 (11/1/2/0) Section: Introduction
0+1+0 (1/0/0/0) Section: Background
264+12+0 (6/0/0/0) Subsection: Adversary Model
345+4+0 (1/0/0/0) Subsection: Feature Design for In-Vehicle Intrusion Detection
389+12+0 (3/0/2/0) Subsection: Takeaways from Existing Methods
38+3+77 (1/1/0/0) Section: Proposed Method: X-CANIDS+
228+2+0 (1/0/6/0) Subsection: Problem Statement
664+7+0 (5/0/29/2) Subsection: Feature Extractor
273+0+0 (4/0/35/3) Subsection: Explainer
159+3+0 (1/0/16/0) Sub_
20+2+0 (1/0/0/0) Section:
253+2+6 (1/1/12/0) Sub_
241+3+30 (1/2/3/0) Sub_
152+2+0 (1/0/1/0) Subsections: On-device Inference
347+5+105 (3/1/10/0) Subsection: Explanation
174+3+55 (1/1/3/0) Section: Discussion and Conclusion
  
```

총 워드 수: 4,490

Subsection "Feature Extractor" 내 워드 수: 464

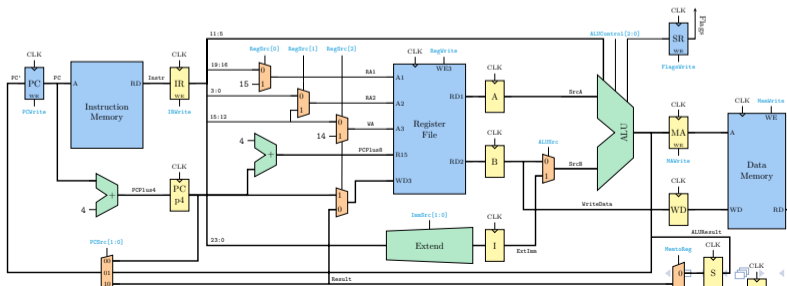
※ 매일 자신이 새로 작성한 단어 수(Word count)를 추적하면, 장기적인 논문 집필 페이스 유지와 생산성 관리에 큰 도움이 됩니다.

# 그래픽 라이브러리: TikZ

## TikZ (티크지)란?

- $\text{\LaTeX}$  환경 내에서 프로그래밍 코드를 이용해 고품질의 기하학적/공학적인 벡터 그래픽을 직접 그리는 라이브러리입니다.
- 방대한 갤러리를 제공하는 [tikz.net](http://tikz.net) 에서 전 세계 유저들이 공유하는 놀라운 데모 소스 코드를 확인하고 가져다 쓸 수 있습니다.

예시: ARM CPU Datapath 직접 렌더링 (출처: [tikz.net/arm-cpu/](http://tikz.net/arm-cpu/))



# Matplotlib → L<sup>A</sup>T<sub>E</sub>X → PDF (그래프 렌더링 맛보기)

## 1. Python 스크립트 작성

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 plt.figure(figsize=(4, 3))
5 x = np.linspace(0, 10, 100)
6 plt.plot(x, np.sin(x), label='sin(x)')
7 plt.legend()
8
9 plt.savefig('plot.pgf')
```

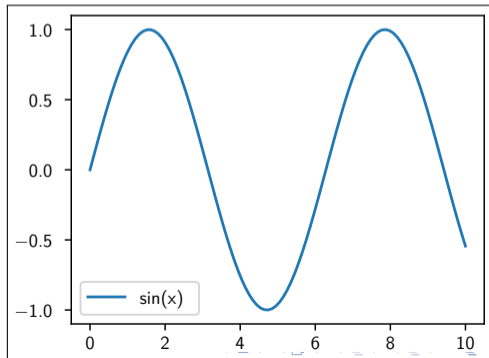
## 2. 생성된 plot.pgf (부분 발췌)

```

1 \begin{pgfpicture}
2 \pgfpathrectangle{\pgfpointorigin}
3 ...
4 \pgftext[x=5.2in, y=3.9in, left, base]{sin(x)}
5 \end{pgfpicture}
```

## 3. 문서 내부에서 불러오기 (최종 결과)

`\input{plot.pgf}` 한 줄로 화질 저하 없이  
논문의 글꼴(Font)까지 그대로 맞춘 채  
렌더링됩니다.



# Matplotlib + SciencePlots (IEEE 포맷) (1/2)

## SciencePlots을 위한 시스템 요구사항

- Python의 **SciencePlots** 패키지를 사용하면 IEEE, Science 등 유명 저널의 깐깐한 논문 스타일(글꼴, 선 굵기, 캡션 등)을 똑같이 재현하여 그래프를 자동 생성할 수 있습니다.
- 단, 필수 전제조건이 있습니다.
- **SciencePlots** 기능을 적용한 상태에서 깔끔한 벡터 그래픽 PDF로 렌더링하려면 자신의 PC에 로컬 **LaTeX** (TeX Live, MacTeX 등) 및 관련 컴파일러가 완전히 설치되어 있어야 합니다.
- Python 프로그램이 뒤에서 백엔드로 우리가 직접 설치한 LaTeX 코어 엔진을 불러와서 PDF 처리에 사용하기 때문입니다.

# Matplotlib + SciencePlots (IEEE 포맷) (2/2)

## 1. Python 스크립트 작성

```

1 import matplotlib.pyplot as plt
2 import scienceplots
3 import numpy as np
4
5 plt.style.use(['science', 'ieee'])
6 plt.figure(figsize=(3.2, 2.4))
7
8 x = np.linspace(0, 10, 100)
9 plt.plot(x, np.sin(x), label=r'\sin(x)')
10 plt.grid(True, linestyle='--')
11
12 t = np.arange(0, 10, np.pi/2)
13 l = ['$0$', r'\frac{\pi}{2}$', r'\pi$', r'\frac{3\pi}{2}$',
14      r'$2\pi$', r'\frac{5\pi}{2}$', r'$3\pi$']
15 plt.xticks(t, l[:len(t)])
16 plt.legend()
17
18 plt.savefig('science_plot.pdf')

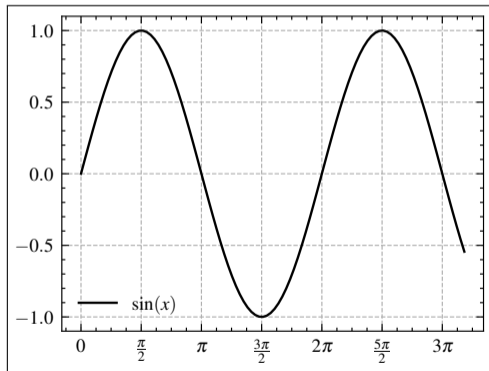
```

**Tip:** 로컬에 TeX Live가 없다면 위 코드로

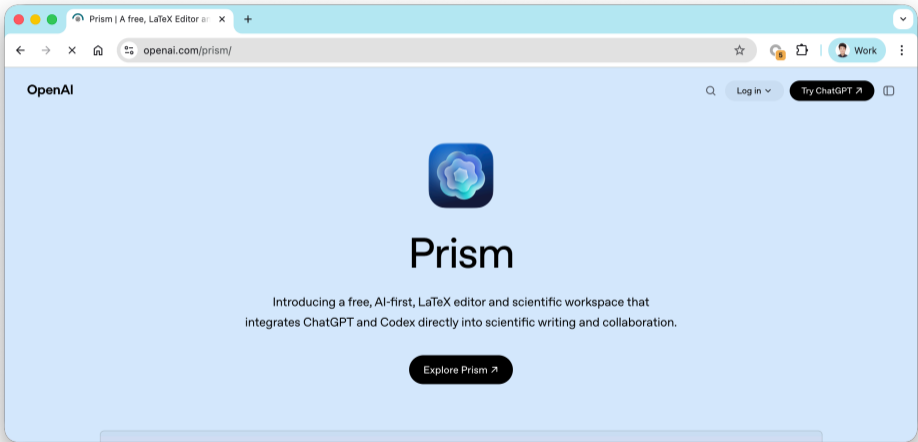
패턴과 나인브 포크그래프 등 추가 옵션 사용

## 2. 생성된 IEEE 포맷 결과물

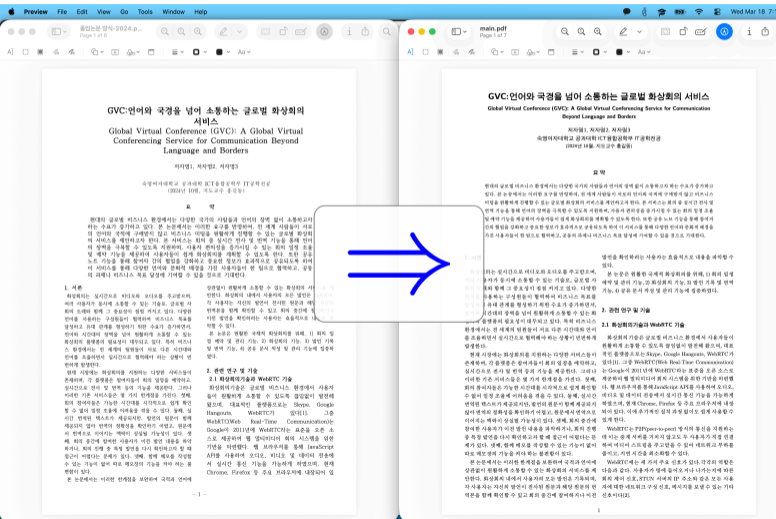
완벽한 논문용 figure 생성



# OpenAI's Prism—an AI-first $\text{\LaTeX}$ editor



# 데모 1: Antigravity를 통한 HWPX → L<sup>A</sup>T<sub>E</sub>X 변환 (IT공학전공 졸업논문 양식)





# Thank you!

Q & A