浙江大学
ZHEJIANG UNIVERSITY

实验二：Typst 模板实现

Typst 短学期 (Typst101) 实验报告

2025 年 8 月 26 日

324010XXXX 犬戎
浙江大学 计算机科学与技术
课程综合实践 I (CS1145M), 2025

I. Introduction

This is a simple template that can be used for a report.

这是一个简单的模板，你可以用它来写报告。

II. Feature 特性

II.1. Colorful items

The main color can be set with the `main-color` property, which affects inline code, lists, links and important items. For example, the words highlight and important are highlighted !

- These bullet
- points
- are colored

1. It also
2. works with
3. numbered lists!

II.2. Customized items

Codeblock

This is a codeblock.

```
use rand::Rng;
use std::cmp::Ordering;
use std::io;

fn main() {
    println!("Guess the number!");

    let secret_number = rand::thread_rng().gen_range(1..101);

    loop {
        println!("Please input your guess.");

        let mut guess = String::new();

        io::stdin()
            .read_line(&mut guess)
            .expect("Failed to read line");

        let guess: u32 = match guess.trim().parse() {
            Ok(num) => num,
            Err(_) => continue,
        };
```

```
        println!("You guessed: {}", guess);

        match guess.cmp(&secret_number) {
            Ordering::Less => println!("Too small!"),
            Ordering::Greater => println!("Too big!"),
            Ordering::Equal => {
                println!("You win!");
                break;
            }
        }
    }
}
```

代码块中英文使用「JetBrainsMonoNL NF」，中文使用「霞鹜文楷屏幕阅读版」。

```
text = "未甚拔行间，犬戎大充斥"
print(text.encode())
```

Figures

Figures are customized. You can of course reference them: Figure 1.

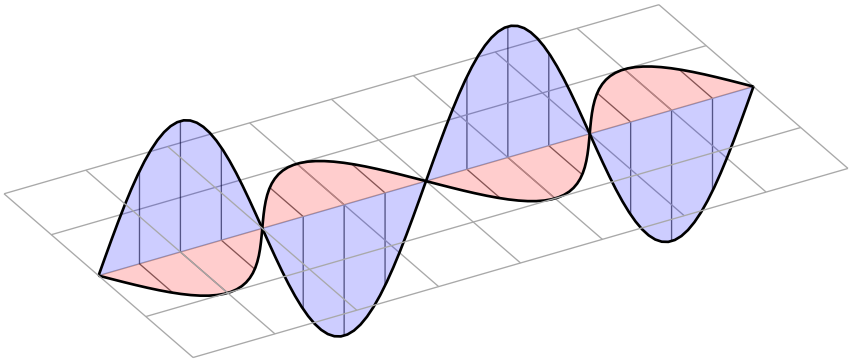

Figure 1 — Waves. (An example from cetz)

II.3. Formula

假设 $\sum_{n=1}^{\infty} a_n$ 是一个条件收敛的无穷级数。对任意的一个实数 C，都存在一种从自然数集合到自然数集合的排列 $\sigma : n \mapsto \sigma(n)$，使得

$$\sum_{n=1}^{\infty} a_{\sigma(n)} = C.$$

此外，也存在另一种排列 $\sigma' : n \mapsto \sigma'(n)$，使得

$$\sum_{n=1}^{\infty} a_{\sigma'(n)} = \infty.$$

类似地，也可以有办法使它的部分和趋于 $-\infty$，或没有任何极限。

反之，如果级数是绝对收敛的，那么无论怎样重排，它仍然会收敛到同一个值，也就是级数的和。

III. 这是一个使用例

以下提供一个真实的使用例，节选自某次课程实验报告 _(:3」∠)_

III.1. Qwen3 Decoder Layer

Qwen3 Decoder Layer 是一个标准的 Transformer 的 Decoder 架构，在此基础上 Layer Norm 部分使用了 RMS Norm。

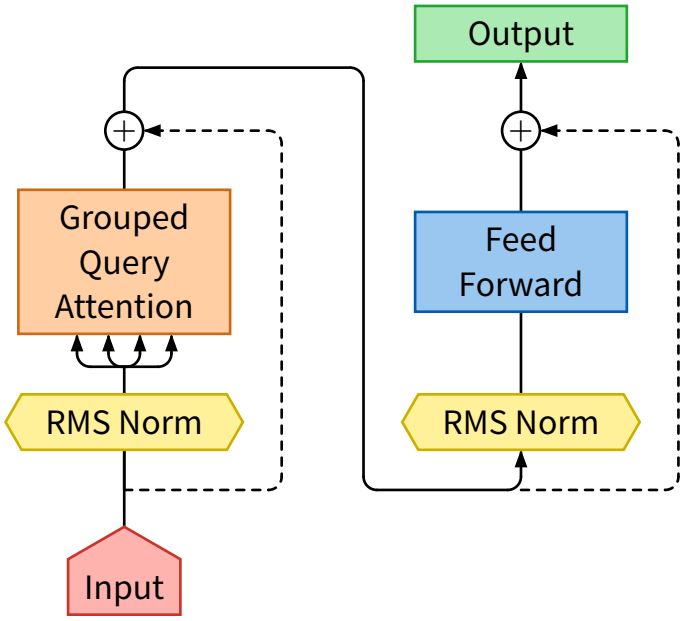

Figure 2 — Overview of Qwen3 Decoder Layer.

III.2. LayerNorm 与 RMSNorm

LayerNorm 主要对每个 token 的特征向量进行归一化计算，其公式为

$$\mathrm{LayerNorm}(\boldsymbol{x}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{x} - \hat{\mu}}{\hat{\sigma}} + \boldsymbol{\beta}.$$

其中

$$\hat{\mu} = \frac{1}{d}\sum_{i=1}^{d} x_i,$$

$$\hat{\sigma}^2 = \frac{1}{d}\sum_{i=1}^{d} (x_i - \mu)^2 + \varepsilon,$$

ε 是防止除零的小常数。$\beta, \gamma \in \mathbb{R}^d$ 是可学习的偏移参数与缩放参数，代表着把第 i 个特征的 batch 分布的均值和方差移动到 β_i, γ_i.

RMSNorm 由论文 Root Mean Square Layer Normalization (arXiv:1910.07467) 提出，其提出动机是传统的 LayerNorm 运算量比较大；而相比 LayerNorm，RMSNorm 不需要同时计算均值和方差两个统计量，而只需要计算均方根这一个统计量，性能和 LayerNorm 相当的同时节省了运算。RMSNorm 的公式为

$$\mathrm{RMSNorm}(\boldsymbol{x}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{x}}{\mathrm{RMS}(\boldsymbol{x})},$$

其中 $\mathrm{RMS}(\boldsymbol{x})$ 是求均方根操作，公式为

$$\mathrm{RMS}(\boldsymbol{x}) = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \varepsilon},$$

ε 是防止除零的小常数。$\boldsymbol{\gamma} \in \mathbb{R}^d$ 是可学习的缩放参数。

III.3. Grouped Query Attention

首先需要了解 Multi-head Attention 与其变体 Multi-query Attention。MQA 在 MHA 的基础上，让所有的头之间共享同一份 K, V，每个头只单独保留了一份 Q，节省了大量 K, V。

而 GQA 实则是 MHA 与 MQA 的一个中间态，它选择的是使用 n 份 Q 对应一份 K, V。也就是说 GQA-1 即为 MHA，GQA-n 即为 MQA。GQA 在节省 K, V 的同时，且在实践中性能仍与经典的 MHA 相近。

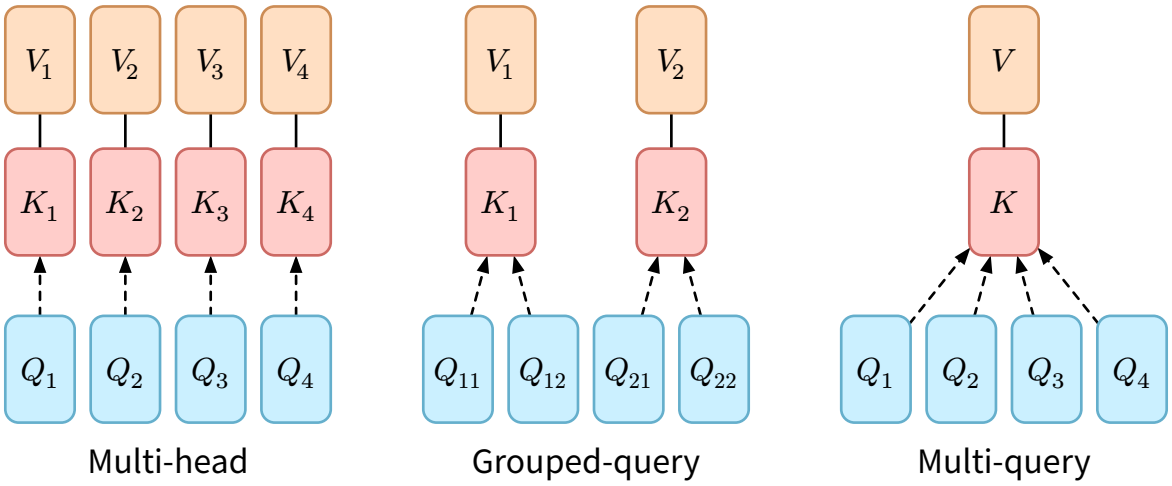

Figure 3 — Overview of grouped-query method.

III.4. Feed Forward Network with Gated Linear Unit

Transformer 中经典的 FFN 通常由一个含偏置的线性层、一个激活函数 σ 再一个含偏置的线性层组成。FFN 的公式可以写作

$$\mathrm{FFN}(x) = W_d \cdot \sigma(W_u \cdot x + b_u) + b_d.$$

> **Note**
>
> 所以也可以见到将 FFN 这一组件叫作 MLP 的称呼，例如实验框架中 FFN 类的类名就叫作 `Qwen3MLP`。不过有一点小出入是，在 Transformer 之外一般所说的经典双层 MLP 每一个神经元都会有一个 σ，于是最后还会有一个 σ。）

而 Gated Linear Unit 可以理解为第一层线性层的一个替代，其核心思想是使用一个带参数的门控层 $\sigma(W_g \cdot x)$ 代替简单激活函数 σ，从而更精确地控制信息的流动；另外使用 GLU 的 FFN 实现通常会去掉偏置。

GLU 的公式可以写作

$$\mathrm{GLU}(x) = \sigma(W_g \cdot x) \odot (W_u \cdot x).$$

> **Tip**
>
> 对比之下，前面提到的经典 FFN 中的第一层线性层可以写作
>
> $$\mathrm{FFN}_1(x) = \sigma(W_u \cdot x + b_u).$$

则 FFN with GLU 可以写作

$$\mathrm{FFN}'(x) = W_d\ \mathrm{GLU}(x) = W_d \cdot \big(\sigma(W_g \cdot x) \odot (W_u \cdot x)\big).$$

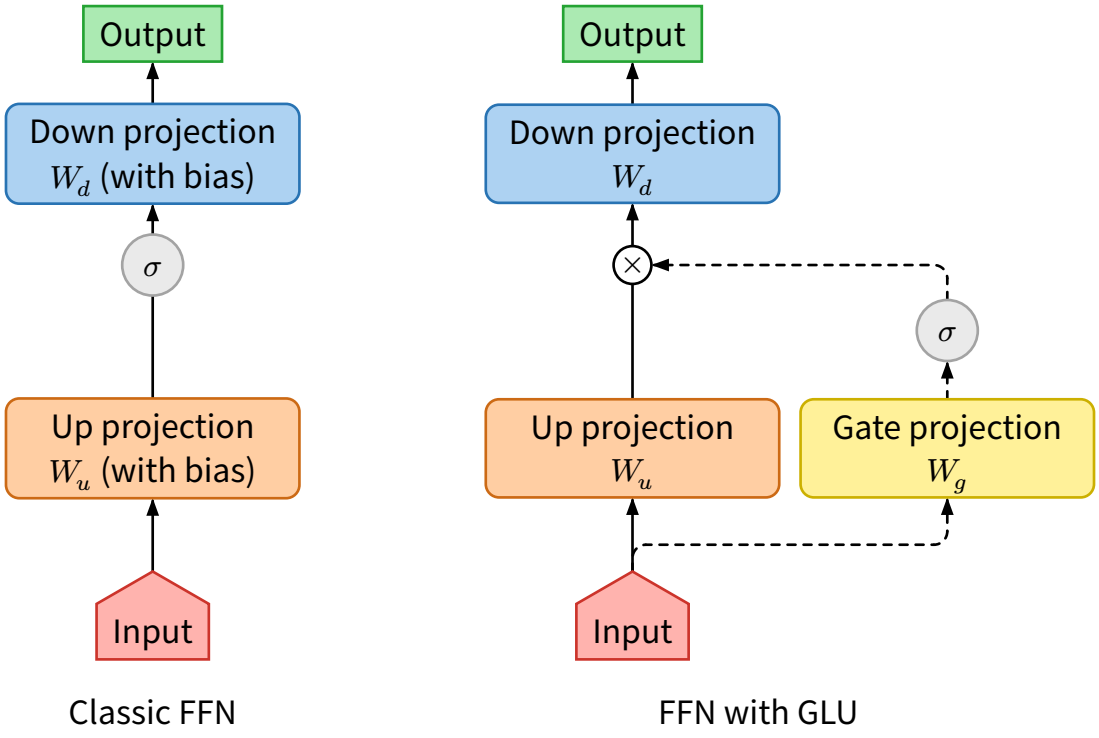

Figure 4 — Comparison of classic FFN and FFN with GLU