Table 1: **Model sizes, architectures, and optimization hyper-parameters.**

| # of Params | Hidden size | Heads | Layers | Learning rate | Batch size | Training tokens |
|---|---|---|---|---|---|---|
| 1.8B | 2048 | 16 | 24 | $3.0 \times 10^{-4}$ | 4M | 2.2T |
| 7B | 4096 | 32 | 32 | $3.0 \times 10^{-4}$ | 4M | 2.4T |
| 14B | 5120 | 40 | 40 | $3.0 \times 10^{-4}$ | 4M | 3.0T |

- **Positional embedding**. We have chosen RoPE (Rotary Positional Embedding) (Su et al., 2021) as our preferred option for incorporating positional information into our model. RoPE has been widely adopted and has demonstrated success in contemporary large language models, notably PaLM (Chowdhery et al., 2022; Anil et al., 2023) and LLaMA (Touvron et al., 2023a;b). In particular, we have opted to use FP32 precision for the inverse frequency matrix, rather than BF16 or FP16, in order to prioritize model performance and achieve higher accuracy.
- **Bias**. For most layers, we remove biases following Chowdhery et al. (2022), but we add biases in the QKV layer of attention to enhance the extrapolation ability of the model (Su, 2023b).
- **Pre-Norm & RMSNorm**. In modern Transformer models, pre-normalization is the most widely used approach, which has been shown to improve training stability compared to post-normalization. Recent research has suggested alternative methods for better training stability, which we plan to explore in future versions of our model. Additionally, we have replaced the traditional layer normalization technique described in (Ba et al., 2016) with RMSNorm (Jiang et al., 2023). This change has resulted in equivalent performance while also improving efficiency.
- **Activation function**. We have selected SwiGLU (Shazeer, 2020) as our activation function, a combination of Swish (Ramachandran et al., 2017) and Gated Linear Unit (Dauphin et al., 2017). Our initial experiments have shown that activation functions based on GLU generally outperform other baseline options, such as GeLU (Hendrycks & Gimpel, 2016). As is common practice in previous research, we have reduced the dimension of the feed-forward network (FFN) from $4$ times the hidden size to $\frac{8}{3}$ of the hidden size.

### 2.3.2 Context Length Extension

Transformer models have a significant limitation in terms of the context length for their attention mechanism. As the context length increases, the quadratic-complexity computation leads to a drastic increase in both computation and memory costs. In this work, we have implemented simple training-free techniques that are solely applied during inference to extend the context length of the model. One of the key techniques we have used is NTK-aware interpolation (bloc97, 2023), which adjusts the scale to prevent the loss of high-frequency information in a training-free manner. To further improve performance, we have also implemented a trivial extension called dynamic NTK-aware interpolation, which is later formally discussed in (Peng et al., 2023a). It dynamically changes the scale by chunks, avoiding severe performance degradation. These techniques allow us to effectively extend the context length of Transformer models without compromising their computational efficiency or accuracy.

Qwen additionally incorporates two attention mechanisms: LogN-Scaling (Chiang & Cholak, 2022; Su, 2023a) and window attention (Beltagy et al., 2020). LogN-Scaling rescales the dot product of the query and value by a factor that depends on the ratio of the context length to the training length, ensuring that the entropy of the attention value remains stable as the context length grows. Window attention restricts the attention to a limited context window, preventing the model from attending to tokens that are too far away.

We also observed that the long-context modeling ability of our model varies across layers, with lower layers being more sensitive in context length extension compared to the higher layers. To leverage this observation, we assign different window sizes to each layer, using shorter windows for lower layers and longer windows for higher layers.