Qwen2.5 Technical Report Analysis

Author: Darshil Patel
Date: 12 April 2025

Abstract

This report introduces **Qwen2.5**, the latest and most advanced release in the Qwen large language model (LLM) series by Alibaba Group. Qwen2.5 features significant improvements in both pre-training and post-training stages. The pre-training data has been expanded to **18 trillion high-quality tokens**, providing a solid foundation for enhanced common sense reasoning, expert knowledge, and multilingual understanding. Post-training incorporates over **1 million supervised fine-tuning samples** and leverages cutting-edge reinforcement learning techniques, including **Direct Preference Optimization (DPO)** and **Group Relative Policy Optimization (GRPO)**, to enhance instruction-following, human alignment, and long-context generation.

Qwen2.5 models are offered in a wide range of sizes from **0.5B to 72B parameters**, including both dense and **Mixture-of-Experts (MoE)** variants like **Qwen2.5-Turbo** and **Qwen2.5-Plus**. These models are available as open-weight and quantized versions through platforms such as Hugging Face, ModelScope, and Alibaba Cloud. Qwen2.5 has demonstrated **state-of-the-art performance across key benchmarks**, including **MMLU (86.1)**, **MATH (62.1)**, and **HumanEval (59.1)**—performing competitively against larger proprietary models like **LLaMA-3-405B** and **GPT-4o**.

Additionally, Qwen2.5 serves as the foundation for specialized models such as **Qwen2.5-Math**, **Qwen2.5-Coder**, and **QwQ**, extending its capabilities across domains. Its modular, scalable, and open-access design positions Qwen2.5 as a leading choice for advanced natural language processing and AI research applications.

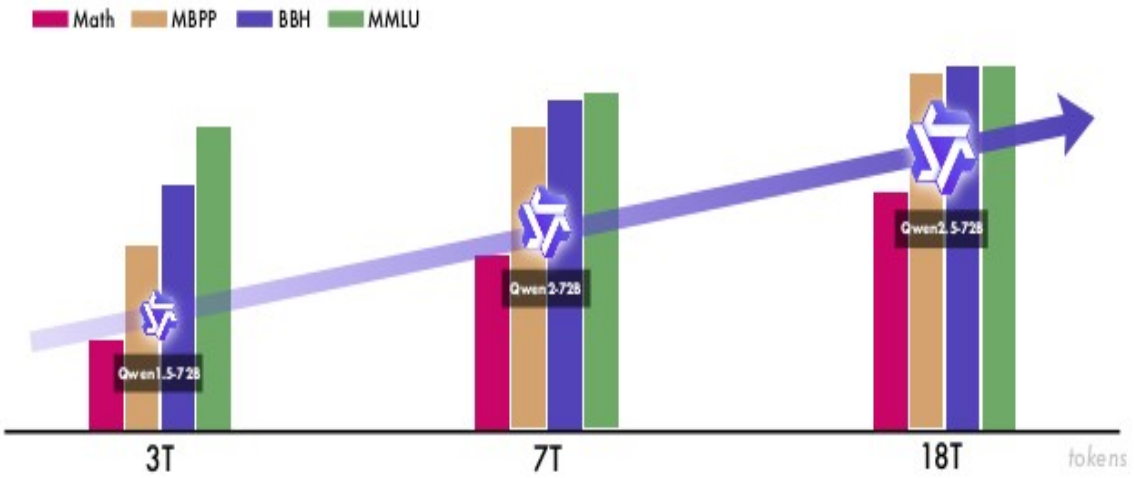


Figure 1

Table Of Content

1. Introduction

The Qwen2.5 technical report highlights significant advancements in large language models (LLMs), underscoring their growing alignment with the goals of artificial general intelligence (AGI). Fueled by advancements in pre-training scale, supervised fine-tuning, and reinforcement learning from human feedback (RLHF), LLMs now exhibit emergent capabilities such as step-by-step reasoning and reflection.
Within this rapidly evolving open-weight LLM ecosystem—alongside models like LLaMA and Mistral—the Qwen series has emerged as a major contributor. **Qwen2.5** builds on this progress with:

- **Model Availability**: Released in seven sizes (0.5B to 72B) including dense and quantized versions, plus two proprietary Mixture-of-Experts (MoE) variants: **Qwen2.5-Turbo** and **Qwen2.5-Plus**.
- **Performance**: The flagship **Qwen2.5-72B-Instruct** matches or exceeds performance of much larger models like **LLaMA-3-405B-Instruct** and is competitive with **GPT-4o**.
- **Key Improvements**:
    - **Better in Size**: Reintroduces under-represented sizes (3B, 14B, 32B) for cost-efficiency and deployment flexibility.
    - **Better in Data**: Enlarges pre-training data from **7T to 18T tokens**, with targeted focus on knowledge, coding, and math; incorporates staged data mixing.
    - **Better in Use**: Expands generation length (2K → 8K tokens), enhances structured data handling (e.g., JSON, tables), and supports **1M-token context** via Turbo models.

Qwen2.5 thus represents a well-rounded improvement in **scalability**, **efficiency**, and **practical usability**, tailored to both high-performance computing and edge applications.

2. Model Architecture & Tokenizer

2.1 Tokenizer

- For tokenization, they utilize Qwen’s tokenizer , which implements **byte-level byte-pair encoding (BBPE)** with a vocabulary of **151,643** regular tokens. They have expanded the set of control tokens from 3 to 22 compared to previous Qwen versions, adding two new tokens for tool functionality and allocating the remainder for other model capabilities. This

expansion establishes a unified vocabulary across all Qwen2.5 models, enhancing consistency and reducing potential compatibility issues.

2.2 Architecture

- Basically, the Qwen2.5 series include dense models for opensource, namely Qwen2.5-0.5B / 1.5B / 3B / 7B / 14B / 32B / 72B, and MoE models for API service, namely Qwen2.5-Turbo and Qwen2.5-Plus.Below, we provide details about the architecture of models.
- For dense models, we maintain the Transformer-based decoder architecture as Qwen2. The architecture incorporates several key components:
  - Grouped Query Attention(GQA) : - for efficient KV cache utilization
  - SwiGLU activation function : - for non-linear activation
  - Rotary Positional Embeddings(RoPE) : - for encoding position information
  - QKV bias in the attention mechanism
  - RMSNorm : - with pre-normalization to ensure stable training.
  - MoE : - Building upon the dense model architectures, we extend it to MoE model architectures. This is achieved by replacing standard feed-forward network (FFN) layers with specialized MoE layers, where each layer comprises multiple FFN experts and a routing mechanism that dispatches tokens to the top-K experts.
  - Following the approaches demonstrated in Qwen1.5-MoE : -
    They implement fine-grained expert segmentation , shared experts routing

3. Pre-Training

Qwen language model pre-training process includes the following key components:

1. **High-Quality Data Curation:**
   - We use sophisticated filtering and scoring mechanisms to curate high-quality training data.
   - A strategic data mixture approach ensures a balanced and effective training dataset.
2. **Hyperparameter Optimization:**
   - Extensive research is conducted to optimize hyperparameters.
   - This enables effective training across various model scales.
3. **Long-Context Pre-Training:**
   - Specialized techniques are incorporated to support long-context understanding.
   - This enhances the model’s ability to process and comprehend extended sequences.

3.1 Pre-training Data

- Qwen2.5 demonstrates significant enhancements in pre-training data quality compared to its predecessor Qwen2. These improvements stem from several key aspects:

1. **Better Data Filtering:**
    - High-quality pre-training data is essential for model performance.
    - Qwen2-Instruct models are used as advanced data quality filters.
    - These models perform **multi-dimensional analysis** to evaluate and score training samples.
    - The filtering process is an upgrade from Qwen2, benefiting from a **larger multilingual corpus**.
    - Enables **more nuanced assessments** and better retention of high-quality data across languages.
2. **Better Math and Code Data:**
    - Pre-training includes data from **Qwen2.5-Math** and **Qwen2.5-Coder**.
    - These datasets help the model excel in **mathematical reasoning** and **code generation**.
    - Domain-specific data boosts **state-of-the-art performance** in math and coding tasks.
3. **Better Synthetic Data:**
    - High-quality synthetic data is generated in math, code, and knowledge domains.
    - Uses models like **Qwen2-72B-Instruct** and **Qwen2-Math-72B-Instruct**.
    - Synthetic data is **filtered with a general reward model** and **Qwen2-Math-RM-72B** for enhanced quality.
4. **Better Data Mixture:**
    - Qwen2-Instruct models classify and balance domain-specific content.
    - Overrepresented domains (e.g., **e-commerce, social media, entertainment**) are **down-sampled** due to repetitiveness and low information density.
    - Underrepresented, high-value domains (e.g., **technology, science, academia**) are **up-sampled** to ensure richness and diversity.
    - Results in a **balanced, information-rich dataset** aligned with the model's learning goals.

3.2 Scaling Law for Hyper-parameters

Hyperparameter Scaling Laws in Qwen2.5

1. **Purpose of Scaling Laws:**
    - Based on prior work (Hoffmann et al., 2022; Kaplan et al., 2020).
    - Unlike previous studies (e.g., Dubey et al., 2024; Almazrouei et al., 2023) that used scaling laws to **determine optimal model sizes**, we use them to **identify optimal hyperparameters** across architectures.
2. **Key Objectives:**
    - Determine optimal **batch size (B)** and **learning rate (μ)**.
    - Apply to both **dense** and **Mixture-of-Experts (MoE)** models across different scales.
3. **Experimental Scope:**
    - Models tested range from:
        - **Dense:** 44M to 14B parameters.
        - **MoE:** 44M to 1B *activated* parameters.
    - Training data size varied from **0.8B to 600B tokens**.
4. **Findings:**
    - Analyzed how optimal **learning rate (μ_{opt})** and **batch size (B_{opt})** vary with:
        - **Model size (N)**
        - **Pre-training data size (D)**
    - Derived predictive relationships between architecture, data scale, and hyperparameters.
5. **Practical Application:**
    - Used optimal hyperparameters to **model final training loss** as a function of:
        - Model architecture
        - Data scale
6. **MoE vs Dense Model Performance:**
    - Scaling laws helped **predict and compare MoE and dense model performance**.
    - Enabled **MoE models to match dense counterparts (e.g., Qwen2.5-72B and Qwen2.5-14B)** through:
        - Careful tuning of **activated** and **total** parameters.

3.3 Long-context Pre-training

1. Two-Phase Pre-Training Strategy:

- **Phase 1:**
    - Initial training with a **4,096-token context length**.
- **Phase 2 (Extension Phase):**
    - Context length is extended to **32,768 tokens** during the final pre-training stage.
    - Applies to **all models except Qwen2.5-Turbo**.
- **RoPE Enhancement:**
    - **Rotary Position Embedding (RoPE)** base frequency increased from **10,000 → 1,000,000** using the **Attention-Before-Frequency (ABF)** technique.

2. Qwen2.5-Turbo: Progressive Context Expansion

- Employs **progressive context length training** through **4 stages**:
    - **32,768 → 65,536 → 131,072 → 262,144 tokens**
- RoPE base frequency set to **10,000,000**
- At each stage:
    - **40%** of training data consists of sequences at the **current max length**
    - **60%** consists of **shorter sequences**
- Purpose: Smooth adaptation to **increasing context lengths** while maintaining generalization to various sequence sizes.

3. Enhancing Long-Sequence Inference:

- Two key techniques used:
    - **YARN** (Peng et al., 2023)
    - **Dual Chunk Attention (DCA)** (An et al., 2024)
- **Impact:**
    - Enables **Qwen2.5-Turbo** to handle **up to 1 million tokens**
    - Other models can process **up to 131,072 tokens**
    - **Reduces perplexity** and maintains **high performance on shorter sequences** as well.

NOTE : -

- **Attention-Before-Frequency (ABF) : -** ABF is a method used in audio and speech processing models — especially those that deal with sounds or voice.

  - **In simple Words :-** Imagine you're listening to someone talk in a noisy room.
    Before trying to analyze every little detail (like pitch or frequency),
    you first focus your attention on the parts of the sound that are most important — maybe their voice, not the background noise.
    So , "Attention-Before-Frequency" means : - First, focus on the important parts of the sound (Attention), then look at the sound's detailed features (Frequency).

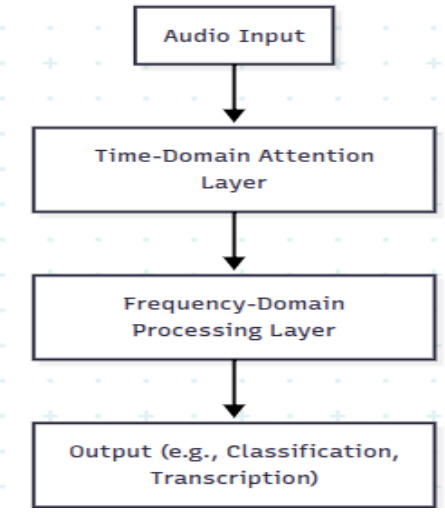
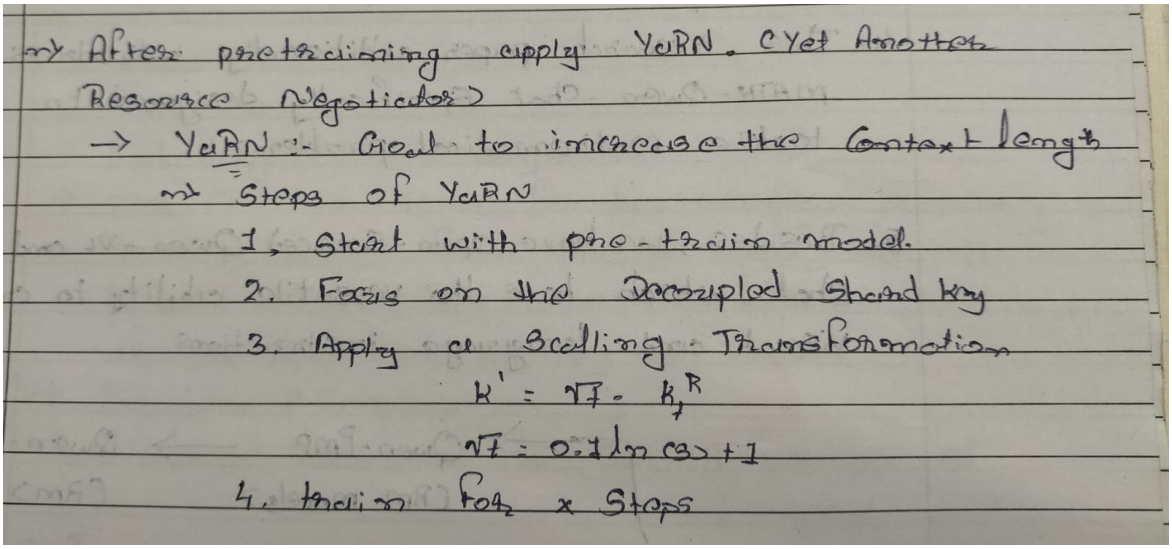


Figure 3.3.1 :- Flow of Attention-Before-Frequency

- **YARN :-**

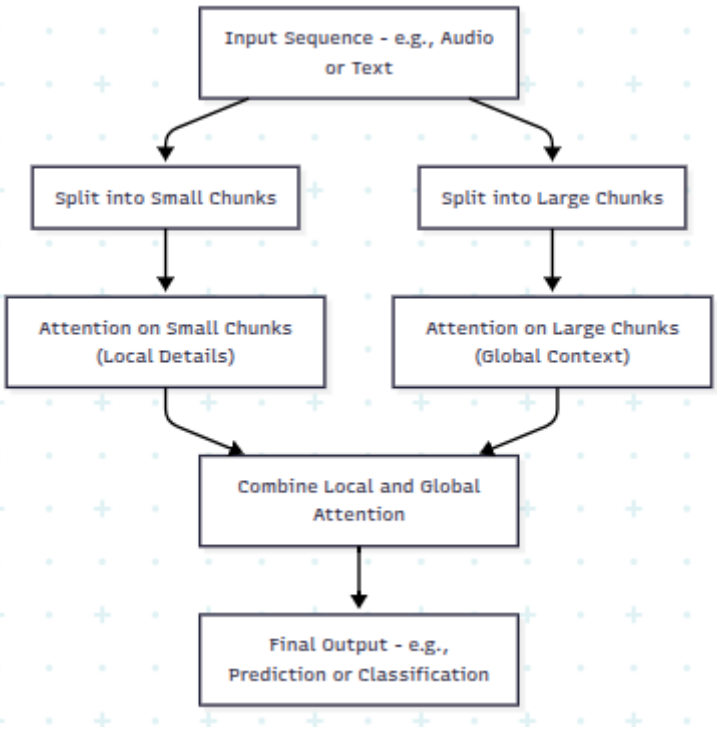


- **Dual Chunk Attention (DCA) : -** DCA is a method used in AI models that work with long audio or text.It helps the model understand both small details and the big picture at the same time.

  - **In simple Words : -** You're listening to a long speech.
    You don’t just focus on each sentence — you also try to understand the entire paragraph or topic.
    DCA does something similar. It looks at:

    1. **Small chunks** – to catch fine details (like a single word or sound).
    2. **Big chunks** – to get the overall meaning or context.

    Then it **combines both attentions** to make better decisions.



4. Training

- The Qwen models are trained using the standard autoregressive language modeling approach, where the model learns to predict the next token based on the sequence of previous tokens. This method is widely used and was initially introduced in the GPT paper by Radford et al. (2018).

- **Key Training Components:**

1. **Context Length:**
   All Qwen models are trained with a **context length of 2048 tokens**.
2. **Data Preparation:**

   Documents are **shuffled and merged**.

   These are then **split into chunks** of the target context length to form training batches.
3. **Efficiency Improvements:**

   FlashAttention (Dao et al., 2022) is used in attention modules to **reduce memory usage and increase speed**.
4. **Optimizer:**

   AdamW optimizer is used for model training.

   Hyperparameters:

   $\beta_1 = 0.9$

   $\beta_2 = 0.95$

   $\varepsilon = 1e\text{-}8$
5. **Learning Rate Strategy:**

   A **cosine learning rate schedule** is applied.

   Each model size has a **custom peak learning rate**.

   The learning rate **decays to 10%** of the peak value over time.
6. **Precision:**

   BFloat16 mixed precision is used during training to improve **training stability** and reduce memory usage.

5. Post-training

5.1 Post-training Data

Qwen 2.5 brings substantial improvements over Qwen 2 by focusing on post-training design, especially via high-quality datasets and a two-stage reinforcement learning process.

5.1.1 Collaborative Data Annotation

- **Human & Automated Review:** Human reviewers and AI critics jointly assess responses using predefined guidelines to ensure quality, relevance, truthfulness, and harmlessness.
- **Quality Scoring:** Responses undergo multi-agent scoring, where only top-tier outputs are retained.
- **Preference Pair Construction:** Feedback from both humans and models is used to construct comparison pairs for reward model training.

5.1.2 Automated Data Synthesis

- **Back-translation for Long Sequences:** Long-text queries are auto-generated and filtered using Qwen2.
- **Code & Math Generation:** Instruction tuning data, synthetic problems, and collaborative coding agents generate high-quality, diverse examples.
- **Multilingual Transfer:** Translation models convert high-resource language instructions to low-resource ones, evaluated via semantic similarity checks.

5.2 Supervised Fine-tuning

- The supervised fine-tuning (SFT) phase involves over 1 million curated examples across critical skill areas, with training conducted over 2 epochs using a 32,768 token sequence length.
- **Key Enhancements in SFT:**
  1. **Long-sequence Generation:**
     - Supports up to **8,192 tokens** in responses.
     - Uses synthetic datasets with output-length constraints and quality filtering.
  2. **Mathematics:**
     - Introduces *Qwen2.5-Math* using public and synthetic datasets.
     - Uses **chain-of-thought reasoning** and **rejection sampling** for accuracy.
  3. **Coding:**
     - Uses *Qwen2.5-Coder*, collaborative agents across **~40 languages**.
     - Includes static code checking and automated unit testing.
  4. **Instruction-following:**
     - Verifies instructions using **code-based validation** and **unit tests**.
     - Curates examples via **execution feedback rejection sampling**.
  5. **Structured Data Understanding:**
     - Covers tabular QA, fact verification, and structural reasoning.
     - Improves inference with reasoning chains.
  6. **Logical Reasoning:**
     - Trains on **70,000+ queries** with varied reasoning styles.
     - Applies filtering and refinement to eliminate flawed logic.
  7. **Cross-Lingual Transfer:**
     - Translates instructions to low-resource languages and evaluates consistency.
  8. **Robust System Instruction:**

        - Trains on **hundreds of system prompts** to increase prompt diversity and reduce performance variance.

    9. **Response Filtering:**
        - Uses **critic models and collaborative scoring** to retain only flawless responses.

- **Training Hyperparameters:**
    - Learning rate: gradually decayed from 7e-6 to 7e-7
    - Weight decay: 0.1
    - Gradient clipping: 1.0

5.3 Offline Reinforcement Learning

Offline RL improves the model's understanding in tasks where reward models struggle to judge quality, such as:

- **Math, Coding, Logical Reasoning, Instruction-following**
- **High-quality response filtering** via execution feedback and answer matching
- **Training Examples:**
    - Good responses → positive examples
    - Rejected responses → negative examples for **DPO (Direct Preference Optimization)**
- **Training Pipeline:**
    - 150,000 response pairs
    - 1 epoch using **Online Merging Optimizer**
- Learning rate: 7e-7
- **Direct Preference Optimization (DPO) :-** DPO is a training method for language models that directly learns from **human preferences** without using a separate reward model or reinforcement learning loop. Instead of assigning numeric scores to responses, DPO simply uses **pairs of outputs**: one that a human prefers, and one they don't. The model is then trained to increase the probability of generating the **preferred response** and decrease it for the **rejected one**. Architecturally, it modifies the **loss function** of the model based on **the difference in log-probabilities** between the two outputs. This makes DPO **simpler, more stable, and easier to implement** than traditional RL methods like PPO, while still aligning the model with human values and preferences.

5.4 Online Reinforcement Learning

Online RL focuses on refining model behavior based on human preferences and reward model feedback.

Reward Model Labeling Guidelines:

- **Truthfulness:** Responses must be factually correct.
- **Helpfulness:** Should be useful and engaging.
- **Conciseness:** Clear and to the point.

- **Relevance:** Aligned with the user's query.
- **Harmlessness:** Avoid harmful or unsafe content.
- **Debiasing:** Avoid political, gender, or racial bias.

Training Details:

- Uses **Group Relative Policy Optimization (GRPO)**
- Queries sorted by **response score variance**
- **8 responses** sampled per query
- Trained on:
  - **2048 global batch size**
  - **2048 samples per episode** (1 = query + 2 responses)
- **Group Relative Policy Optimization (GRPO)** :- GRPO is a reinforcement learning algorithm designed to fine-tune language models by optimizing their behavior based on **relative feedback within a group of outputs**. Unlike traditional methods like PPO (Proximal Policy Optimization) that train on single prompt-response pairs with scalar rewards, GRPO works by generating a **batch (group) of responses** to the same prompt, **ranking them** using a reward model, and then training the policy to favor higher-ranked responses **relative to others in the group**. Architecturally, it modifies the loss function to be **group-wise**, focusing on **pairwise or list-wise comparisons** among outputs, which leads to more stable and efficient learning. It blends ideas from **listwise ranking** in information retrieval with **policy gradient methods** in RL, allowing the model to improve its responses by learning from its own variations rather than relying only on absolute correctness.

5.5 Long Context Fine-tuning

To enable better long-context handling, especially for Qwen2.5-Turbo, fine-tuning is done in two stages:

Phase 1: Short Instruction Tuning

- Instructions up to **32,768 tokens**
- Same SFT data and steps as the base models

Phase 2: Hybrid Instruction Tuning

- Combines short (≤ **32k tokens**) and long instructions (≤ **262,144 tokens**)
- Improves instruction-following in **long-context tasks**

Note: RL is applied only on short instructions due to:

- High computational cost for long sequences

- Lack of reward models suited for long-context evaluation
  Yet, even this improves alignment for long tasks.

6. Evaluation

The Qwen model underwent a comprehensive evaluation process structured along the following key components:

1. **Base Model Evaluation:**
   Assessment of the core capabilities of the pre-trained base models prior to any instruction tuning or reinforcement learning.
2. **Instruction-Tuned Model Evaluation:**
   Evaluation of the model's performance after supervised fine-tuning and reinforcement learning, covering multiple dimensions:
   - **Open Benchmark Evaluation:**
     Comparison against standard public benchmarks to assess general-purpose language understanding and task performance.
   - **In-house Automatic Evaluation:**
     Internal metrics and automated systems were used to evaluate consistency, accuracy, and robustness across a wide range of tasks.
   - **Reward Model Evaluation:**
     Assessment of the reward model used for reinforcement learning, ensuring alignment with human preferences such as truthfulness, helpfulness, and safety.
   - **Long Context Capability Evaluation:**
     Testing of the model’s ability to handle extended input and output sequences, measuring performance on long-context instruction-following and reasoning tasks.

Testing Qwen2.5-Turbo via “Passkey Retrieval” : -

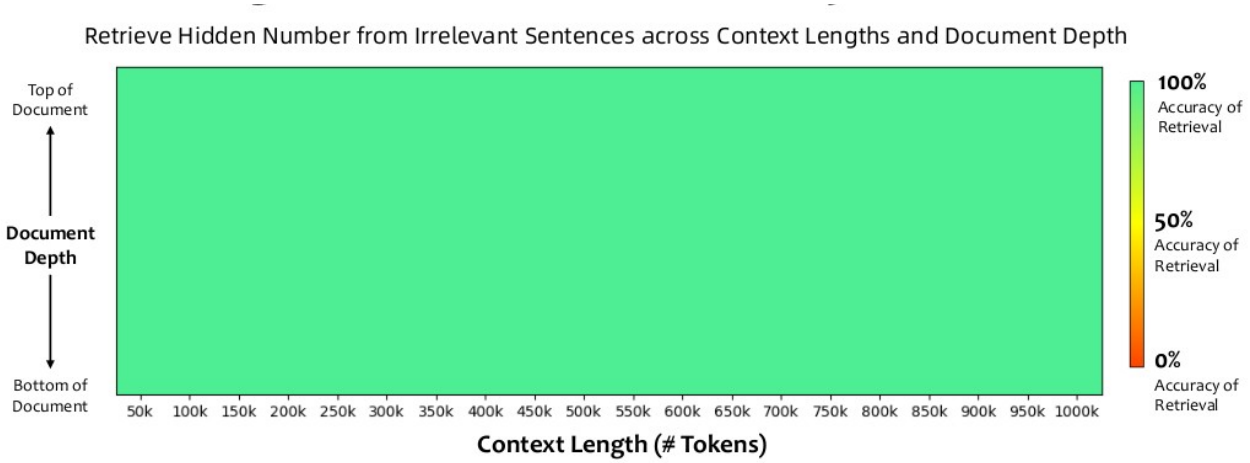


Figure 1 :- Performance of Qwen2.5-Turbo on Passkey Retrieval Task with 1M Token Lengths.

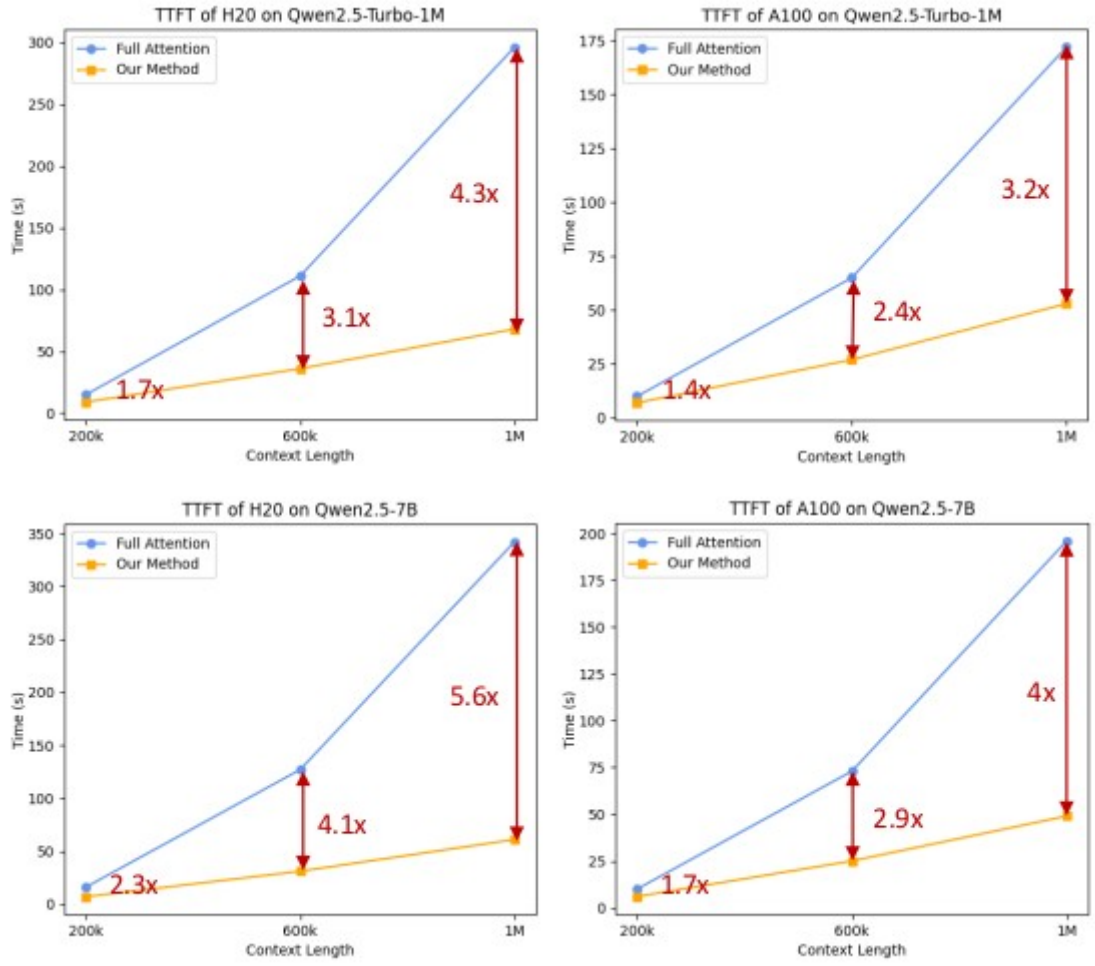


Figure 2 :- TTFT (Time To First Token) of Qwen2.5-Turbo and Qwen2.5-7B with Full Attention and Our Method.

7. Conclusion

- Qwen2.5 marks a significant leap forward in the evolution of large language models (LLMs). With its enhanced pretraining on 18 trillion tokens and the adoption of advanced post-training methods such as supervised fine-tuning and multi-stage reinforcement learning, Qwen2.5 achieves substantial improvements in:
    - Alignment with **human preferences**,
    - **Long-context generation** capabilities,
    - And **structured data understanding**.
- It is available in a wide range of configurations—from open-weight models (ranging from **0.5B to 72B parameters**) to proprietary and cost-efficient **MoE (Mixture-of-Experts)** variants like **Qwen2.5-Turbo** and **Qwen2.5-Plus**.
- Empirical results reveal that **Qwen2.5-72B-Instruct** delivers performance **on par with LLaMA-3-405B-Instruct**, despite being **six times smaller**, highlighting its **efficiency and competitiveness**.
- Moreover, Qwen2.5 serves as a robust **foundation for domain-specific and specialized models**, underlining its versatility for both **academic research** and **industrial use cases**.

8. References

- Qwen-2 Technical Paper
- Qwen-2.5 Technical Paper
- Take Reference of **My DeepSeek-Analysis Paper** for understandin **MoE (Mixture-of-Experts)** (more clearly) , **GQA (Grouped Query Attention)** , **RoPE (Rotary Positional Embeddings)**

Analysis Summary

This document presents a detailed analysis of the Qwen2.5 technical report released by Alibaba Group. The analysis is based on official sources and explores core aspects such as the model's Transformer-based architecture with Grouped Query Attention (GQA), advanced training on 18 trillion high-quality tokens, and the use of Mixture-of-Experts (MoE) in Turbo variants. Post-training strategies including Direct Preference Optimization (DPO) and Group Relative Policy Optimization (GRPO) are also covered, emphasizing Qwen2.5’s reinforcement learning alignment pipeline. Additionally, the report discusses long-context handling techniques like RoPE scaling, Dual Chunk Attention, and YARN, which enable token processing up to 1 million in length. This summary is created for educational and research purposes, with all technical insights and references drawn directly from the official Qwen2.5 technical paper and related documentation. Credit is given to the original authors and research teams behind Qwen2.5.