Qwen2.5 Technical Report

Qwen Team

https://huggingface.co/Qwen
https://modelscope.cn/organization/qwen
https://github.com/QwenLM/Qwen2.5

Abstract

In this report, we introduce Qwen2.5, a comprehensive series of large language models (LLMs) designed to meet diverse needs. Compared to previous iterations, Qwen 2.5 has been significantly improved during both the pre-training and post-training stages. In terms of pre-training, we have scaled the high-quality pre-training datasets from the previous 7 trillion tokens to 18 trillion tokens. This provides a strong foundation for common sense, expert knowledge, and reasoning capabilities. In terms of post-training, we implement intricate supervised finetuning with over 1 million samples, as well as multistage reinforcement learning, including offline learning DPO and online learning GRPO. Post-training techniques significantly enhance human preference, and notably improve long text generation, structural data analysis, and instruction following.

To handle diverse and varied use cases effectively, we present Qwen2.5 LLM series in rich configurations. The open-weight offerings include base models and instruction-tuned models in sizes of 0.5B, 1.5B, 3B, 7B, 14B, 32B, and 72B parameters. Quantized versions of the instruction-tuned models are also provided. Over 100 models can be accessed from Hugging Face Hub, ModelScope, and Kaggle. In addition, for hosted solutions, the proprietary models currently include two mixture-of-experts (MoE) variants: Qwen2.5-Turbo and Qwen2.5-Plus, both available from Alibaba Cloud Model Studio.

Qwen2.5 has demonstrated top-tier performance on a wide range of benchmarks evaluating language understanding, reasoning, mathematics, coding, human preference alignment, etc. Specifically, the open-weight flagship Qwen2.5-72B-Instruct outperforms a number of open and proprietary models and demonstrates competitive performance to the state-of-the-art open-weight model, Llama-3-405B-Instruct, which is around 5 times larger. Qwen2.5-Turbo and Qwen2.5-Plus offer superior cost-effectiveness while performing competitively against GPT-4o-mini and GPT-4o respectively. Additionally, as the foundation, Qwen2.5 models have been instrumental in training specialized models such as Qwen2.5-Math (Yang et al., 2024b), Qwen2.5-Coder (Hui et al., 2024), QwQ (Qwen Team, 2024d), and multimodal models.

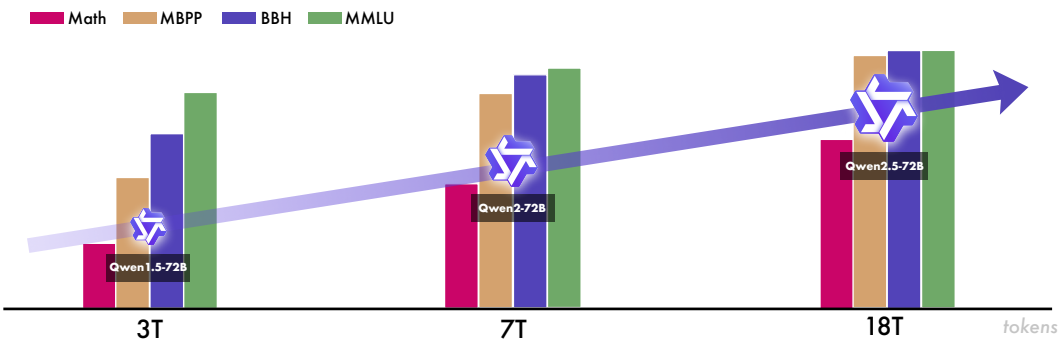

Figure 1: In the iterative development of the Qwen series, data scaling has played a crucial role. Qwen 2.5, which leverages 18 trillion tokens for pre-training, has demonstrated the most advanced capabilities within the Qwen series, especially in terms of domain expertise, underscoring the importance of scale together with mixture in enhancing the model's capabilities.

1 Introduction

The sparks of artificial general intelligence (AGI) are increasingly visible through the fast development of large foundation models, notably large language models (LLMs) (Brown et al., 2020; OpenAI, 2023; 2024a; Gemini Team, 2024; Anthropic, 2023a;b; 2024; Bai et al., 2023; Yang et al., 2024a; Touvron et al., 2023a;b; Dubey et al., 2024). The continuous advancement in model and data scaling, combined with the paradigm of large-scale pre-training followed by high-quality supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), has enabled large language models (LLMs) to develop emergent capabilities in language understanding, generation, and reasoning. Building on this foundation, recent breakthroughs in inference time scaling, particularly demonstrated by o1 (OpenAI, 2024b), have enhanced LLMs' capacity for deep thinking through step-by-step reasoning and reflection. These developments have elevated the potential of language models, suggesting they may achieve significant breakthroughs in scientific exploration as they continue to demonstrate emergent capabilities indicative of more general artificial intelligence.

Besides the fast development of model capabilities, the recent two years have witnessed a burst of open (open-weight) large language models in the LLM community, for example, the Llama series (Touvron et al., 2023a;b; Dubey et al., 2024), Mistral series (Jiang et al., 2023a; 2024a), and our Qwen series (Bai et al., 2023; Yang et al., 2024a; Qwen Team, 2024a; Hui et al., 2024; Qwen Team, 2024c; Yang et al., 2024b). The open-weight models have democratized the access of large language models to common users and developers, enabling broader research participation, fostering innovation through community collaboration, and accelerating the development of AI applications across diverse domains.

Recently, we release the details of our latest version of the Qwen series, Qwen2.5. In terms of the open-weight part, we release pre-trained and instruction-tuned models of 7 sizes, including 0.5B, 1.5B, 3B, 7B, 14B, 32B, and 72B, and we provide not only the original models in bfloat16 precision but also the quantized models in different precisions. Specifically, the flagship model Qwen2.5-72B-Instruct demonstrates competitive performance against the state-of-the-art open-weight model, Llama-3-405B-Instruct, which is around 5 times larger. Additionally, we also release the proprietary models of Mixture-of-Experts (MoE, Lepikhin et al., 2020; Fedus et al., 2022; Zoph et al., 2022), namely Qwen2.5-Turbo and Qwen2.5-Plus[1], which performs competitively against GPT-4o-mini and GPT-4o respectively.

In this technical report, we introduce Qwen2.5, the result of our continuous endeavor to create better LLMs. Below, we show the key features of the latest version of Qwen:

- **Better in Size**: Compared with Qwen2, in addition to 0.5B, 1.5B, 7B, and 72B models, Qwen2.5 brings back the 3B, 14B, and 32B models, which are more cost-effective for resource-limited scenarios and are under-represented in the current field of open foundation models. Qwen2.5-Turbo and Qwen2.5-Plus offer a great balance among accuracy, latency, and cost.
- **Better in Data**: The pre-training and post-training data have been improved significantly. The pre-training data increased from 7 trillion tokens to 18 trillion tokens, with focus on knowledge, coding, and mathematics. The pre-training is staged to allow transitions among different mixtures. The post-training data amounts to 1 million examples, across the stage of supervised finetuning (SFT, Ouyang et al., 2022), direct preference optimization (DPO, Rafailov et al., 2023), and group relative policy optimization (GRPO, Shao et al., 2024).
- **Better in Use**: Several key limitations of Qwen2 in use have been eliminated, including larger generation length (from 2K tokens to 8K tokens), better support for structured input and output, (e.g., tables and JSON), and easier tool use. In addition, Qwen2.5-Turbo supports a context length of up to 1 million tokens.

2 Architecture & Tokenizer

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 (Vaswani et al., 2017; Radford et al., 2018) as Qwen2 (Yang et al., 2024a). The architecture incorporates several key components: Grouped Query Attention (GQA, Ainslie et al., 2023) for efficient KV cache utilization, SwiGLU activation function (Dauphin et al., 2017) for non-linear activation, Rotary Positional Embeddings (RoPE, Su

Kiến trúc kết hợp 1 số các thành phần chính:
+ GQA sử dụng KV cache hiệu quả
+ SwiGLU activation function
+ RoPE để encode positional infos
+ QKVbias trong attention và RMSNorm nhằm chuẩn hóa để đảm bảo quá trình training

[1] Qwen2.5-Turbo is identified as `qwen-turbo-2024-11-01` and Qwen2.5-Plus is identified as `qwen-plus-2024-xx-xx` (to be released) in the API.

Table 1: Model architecture and license of Qwen2.5 open-weight models.

Models	Layers	Heads (Q / KV)	Tie Embedding	Context / Generation Length	License
0.5B	24	14 / 2	Yes	32K / 8K	Apache 2.0
1.5B	28	12 / 2	Yes	32K / 8K	Apache 2.0
3B	36	16 / 2	Yes	32K / 8K	Qwen Research
7B	28	28 / 4	No	128K / 8K	Apache 2.0
14B	48	40 / 8	No	128K / 8K	Apache 2.0
32B	64	40 / 8	No	128K / 8K	Apache 2.0
72B	80	64 / 8	No	128K / 8K	Qwen

et al., 2024) for encoding position information, QKV bias (Su, 2023) in the attention mechanism and RMSNorm (Jiang et al., 2023b) with pre-normalization to ensure stable training.

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 (Yang et al., 2024a), we implement fine-grained expert segmentation (Dai et al., 2024) and shared experts routing (Rajbhandari et al., 2022; Dai et al., 2024). These architectural innovations have yielded substantial improvements in model performance across downstream tasks.

For tokenization, we utilize Qwen's tokenizer (Bai et al., 2023), which implements byte-level byte-pair encoding (BBPE, Brown et al., 2020; Wang et al., 2020; Sennrich et al., 2016) with a vocabulary of 151,643 regular tokens. We 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.

3 Pre-training

Our language model pre-training process consists of several key components. First, we carefully curate high-quality training data through sophisticated filtering and scoring mechanisms, combined with strategic data mixture. Second, we conduct extensive research on hyperparameter optimization to effectively train models at various scales. Finally, we incorporate specialized long-context pre-training to enhance the model's ability to process and understand extended sequences. Below, we detail our approaches to data preparation, hyperparameter selection, and long-context training.

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 crucial for model performance, making data quality assessment and filtering a critical component of our pipeline. We leverage Qwen2-Instruct models as data quality filters that perform comprehensive, multi-dimensional analysis to evaluate and score training samples. The filtering method represents a significant advancement over our previous approach used for Qwen2, as it benefits from Qwen2's expanded pre-training on a larger multilingual corpus. The enhanced capabilities enable more nuanced quality assessment, resulting in both improved retention of high-quality training data and more effective filtering of low-quality samples across multiple languages.

(2) **Better math and code data**. During the pre-training phase of Qwen2.5, we incorporate training data from Qwen2.5-Math (Yang et al., 2024b) and Qwen2.5-Coder (Hui et al., 2024). This data integration strategy proves highly effective, as these specialized datasets are instrumental in achieving state-of-the-art performance on mathematical and coding tasks. By leveraging these high-quality domain-specific datasets during pre-training, Qwen2.5 inherits strong capabilities in both mathematical reasoning and code generation.

(3) **Better synthetic data**. To generate high-quality synthetic data, particularly in mathematics, code, and knowledge domains, we leverage both Qwen2-72B-Instruct (Yang et al., 2024a) and Qwen2-Math-72B-Instruct (Qwen Team, 2024c). The quality of this synthesized data is further enhanced through rigorous filtering using our proprietary general reward model and the specialized Qwen2-Math-RM-72B (Qwen Team, 2024c) model.

(4) **Better data mixture**. To optimize the pre-training data distribution, we employ Qwen2-Instruct models to classify and balance content across different domains. Our analysis revealed that domains like e-commerce, social media, and entertainment are significantly overrepresented in web-scale data, often containing repetitive, template-based, or machine-generated content. Conversely, domains such as technology, science, and academic research, while containing higher-quality information, are traditionally underrepresented. Through strategic down-sampling of overrepresented domains and up-sampling of high-value domains, we ensure a more balanced and information-rich training dataset that better serves our model's learning objectives.

Building on these techniques, we have developed a larger and higher-quality pre-training dataset, expanding from the 7 trillion tokens used in Qwen2 (Yang et al., 2024a) to **18 trillion** tokens.

3.2 Scaling Law for Hyper-parameters

We develop scaling laws for hyper-parameter based on the pre-training data of Qwen2.5 (Hoffmann et al., 2022; Kaplan et al., 2020). While previous studies (Dubey et al., 2024; Almazrouei et al., 2023; Hoffmann et al., 2022) primarily used scaling laws to determine optimal model sizes given compute budgets, we leverage them to identify optimal hyperparameters across model architectures. Specifically, our scaling laws help determine key training parameters like batch size B and learning rate μ for both dense models and MoE models of varying sizes.

Through extensive experimentation, we systematically study the relationship between model architecture and optimal training hyper-parameters. Specifically, we analyze how the optimal learning rate μ_{opt} and batch size B_{opt} vary with model size N and pre-training data size D. Our experiments cover a comprehensive range of architectures, including dense models with 44M to 14B parameters and MoE models with 44M to 1B activated parameters, trained on datasets ranging from 0.8B to 600B tokens. Using these optimal hyper-parameter predictions, we then model the final loss as a function of model architecture and training data scale.

Additionally, we leverage scaling laws to predict and compare the performance of MoE models with varying parameter counts against their dense counterparts. This analysis guides our hyper-parameter configuration for MoE models, enabling us to achieve performance parity with specific dense model variants (such as Qwen2.5-72B and Qwen2.5-14B) through careful tuning of both activated and total parameters.

3.3 Long-context Pre-training

For optimal training efficiency, Qwen2.5 employs a two-phase pre-training approach: an initial phase with a 4,096-token context length, followed by an extension phase for longer sequences. Following the strategy used in Qwen2, we extend the context length from 4,096 to 32,768 tokens during the final pre-training stage for all model variants except Qwen2.5-Turbo. Concurrently, we increase the base frequency of RoPEfrom 10,000 to 1,000,000 using the ABF technique (Xiong et al., 2023).

For Qwen2.5-Turbo, we implement a progressive context length expansion strategy during training, advancing through four stages: 32,768 tokens, 65,536 tokens, 131,072 tokens, and ultimately 262,144 tokens, with a RoPE base frequency of 10,000,000. At each stage, we carefully curate the training data to include 40% sequences at the current maximum length and 60% shorter sequences. This progressive training methodology enables smooth adaptation to increasing context lengths while maintaining the model's ability to effectively process and generalize across sequences of varying lengths.

To enhance our models' ability to process longer sequences during inference, we implement two key strategies: YARN (Peng et al., 2023) and Dual Chunk Attention (DCA, An et al., 2024). Through these innovations, we achieve a four-fold increase in sequence length capacity, enabling Qwen2.5-Turbo to handle up to **1 million** tokens and other models to process up to 131,072 tokens. Notably, these approaches not only improve the modeling of long sequences by reducing perplexity but also maintain the models' strong performance on shorter sequences, ensuring consistent quality across varying input lengths.

4 Post-training

Qwen 2.5 introduces two significant advancements in its post-training design compared to Qwen 2:

(1) **Expanded Supervised Fine-tuning Data Coverage:** The supervised fine-tuning process leverages a massive dataset comprising millions of high-quality examples. This expansion specifically addresses key areas where the previous model showed limitations, such as long-sequence

generation, mathematical problem-solving, coding, instruction-following, structured data understanding, logical reasoning, cross-lingual transfer, and robust system instruction.

(2) **Two-stage Reinforcement Learning:** The reinforcement learning (RL) process in Qwen 2.5 is divided into two distinct stages: Offline RL and Online RL.

- *Offline RL:* This stage focuses on developing capabilities that are challenging for the reward model to evaluate, such as reasoning, factuality, and instruction-following. Through meticulous construction and validation of training data, we ensure that the Offline RL signals are both learnable and reliable (Xiang et al., 2024), enabling the model to acquire those complex skills effectively.
- *Online RL:* The Online RL phase leverages the reward model's ability to detect nuances in output quality, including truthfulness, helpfulness, conciseness, relevance, harmlessness and debiasing. It enables the model to generate responses that are precise, coherent, and well-structured while maintaining safety and readability. As a result, the model's outputs consistently meet human quality standards and expectations.

4.1 Supervised Fine-tuning

In this section, we detail the key enhancements made during the SFT phase of Qwen2.5, focusing on several critical areas:

(1) **Long-sequence Generation:** Qwen2.5 is capable of generating high-quality content with an output context length of up to 8,192 tokens, a significant advancement over the typical post-training response length, which often remains under 2,000 tokens. To address this gap, we develop long-response datasets (Quan et al., 2024). We employ back-translation techniques to generate queries for long-text data from pre-training corpora, impose output length constraints, and use Qwen2 to filter out low-quality paired data.

(2) **Mathematics:** We introduce the chain-of-thought data of Qwen2.5-Math (Yang et al., 2024b), which encompasses a diverse range of query sources, including public datasets, K-12 problem collections, and synthetic problems. To ensure high-quality reasoning, we employ rejection sampling (Yuan et al., 2023) along with reward modeling and annotated answers for guidance, producing step-by-step reasoning process.

(3) **Coding:** To enhance coding capabilities, we incorporate the instruction tuning data of Qwen2.5-Coder (Hui et al., 2024). We use multiple language-specific agents into a collaborative framework, generating diverse and high-quality instruction pairs across nearly 40 programming languages. We expand our instruction dataset by synthesizing new examples from code-related Q&A websites and gathering algorithmic code snippets from GitHub. A comprehensive multilingual sandbox is used to perform static code checking and validate code snippets through automated unit testing, ensuring code quality and correctness (Dou et al., 2024; Yang et al., 2024c).

(4) **Instruction-following:** To ensure high-quality instruction-following data, we implement a rigorous code-based validation framework. In this approach, LLMs generate both instructions and corresponding verification code, along with comprehensive unit tests for cross-validation. Through execution feedback-based rejection sampling, we carefully curate the training data used for Supervised Fine-Tuning, thereby guaranteeing the model's faithful adherence to intended instructions (Dong et al., 2024).

(5) **Structured Data Understanding:** We develop a comprehensive structured understanding dataset that encompasses both traditional tasks, such as tabular question-answering, fact verification, error correction, and structural understanding, as well as complex tasks involving structured and semi-structured data. By incorporating reasoning chains into the model's responses, we significantly enhance its ability to infer information from structured data, thereby improving its performance across these diverse tasks. This approach not only broadens the scope of the dataset but also deepens the model's capacity to reason and derive meaningful insights from complex data structures.

(6) **Logical Reasoning:** To enhance the model's logical reasoning capabilities, we introduce a diverse set of 70,000 new queries spanning various domains. These queries encompass multiple-choice questions, true / false questions, and open-ended questions. The model is trained to approach problems systematically, employing a range of reasoning methods such as deductive reasoning, inductive generalization, analogical reasoning, causal reasoning, and statistical reasoning. Through iterative refinement, we systematically filter out data containing incorrect answers or flawed reasoning processes. This process progressively strengthens the model's ability to reason logically and accurately, ensuring robust performance across different types of reasoning tasks.

(7) **Cross-Lingual Transfer:** To facilitate the transfer of the model's general capabilities across languages, we employ a translation model to convert instructions from high-resource languages into various low-resource languages, thereby generating corresponding response candidates. To ensure the accuracy and consistency of these responses, we evaluate the semantic alignment between each multilingual response and its original counterpart. This process preserves the logical structure and stylistic nuances of the original responses, thereby maintaining their integrity and coherence across different languages.

(8) **Robust System Instruction:** We construct hundreds of general system prompts to improve the diversity of system prompts in post-training, ensuring consistency between system prompts and conversations. Evaluations with different system prompts show that the model maintains good performance (Lu et al., 2024b) and reduced variance, indicating improved robustness.

(9) **Response Filtering:** To evaluate the quality of responses, we employ multiple automatic annotation methods, including a dedicated critic model and a multi-agent collaborative scoring system. Responses are subjected to rigorous assessment, and only those deem flawless by all scoring systems are retained. This comprehensive approach ensures that our outputs maintain the highest quality standards.

Ultimately, we construct a dataset of over 1 million SFT examples. The model is fine-tuned for two epochs with a sequence length of 32,768 tokens. To optimize learning, the learning rate is gradually decreased from 7×10^{-6} to 7×10^{-7}. To address overfitting, we apply a weight decay of 0.1, and gradient norms are clipped at a maximum value of 1.0.

4.2 Offline Reinforcement Learning

Compared to Online Reinforcement Learning (RL), Offline RL enables the pre-preparation of training signals, which is particularly advantageous for tasks where standard answers exist but are challenging to evaluate using reward models. In this study, we focus on objective query domains such as mathematics, coding, instruction following, and logical reasoning, where obtaining accurate evaluations can be complex. In the previous phase, we extensively employ strategies like execution feedback and answer matching to ensure the quality of responses. For the current phase, we reuse that pipeline, employing the SFT model to resample responses for a new set of queries. Responses that pass our quality checks are used as positive examples, while those that fail are treated as negative examples for Direct Preference Optimization (DPO) training (Rafailov et al., 2023). To further enhance the reliability and accuracy of the training signals, we make use of both human and automated review processes (Cao et al., 2024). This dual approach ensures that the training data is not only learnable but also aligned with human expectations. Ultimately, we construct a dataset consisting of approximately 150,000 training pairs. The model is then trained for one epoch using the Online Merging Optimizer (Lu et al., 2024a), with a learning rate of 7×10^{-7}.

4.3 Online Reinforcement Learning

To develop a robust reward model for online RL, we adhere to a set of carefully defined labeling criteria. Those criteria ensure that the responses generated by the model are not only high-quality but also aligned with ethical and user-centric standards (Wang et al., 2024a). The specific guidelines for data labeling are as follows:

- **Truthfulness:** Responses must be grounded in factual accuracy, faithfully reflecting the provided context and instructions. The model should avoid generating information that is false or unsupported by the given data.
- **Helpfulness:** The model's output should be genuinely useful, addressing the user's query effectively while providing content that is positive, engaging, educational, and relevant. It should follow the given instructions precisely and offer value to the user.
- **Conciseness:** Responses should be succinct and to the point, avoiding unnecessary verbosity. The goal is to convey information clearly and efficiently without overwhelming the user with excessive detail.
- **Relevance:** All parts of the response should be directly related to the user's query, dialogue history, and the assistant's context. The model should tailor its output to ensure it is perfectly aligned with the user's needs and expectations.
- **Harmlessness:** The model must prioritize user safety by avoiding any content that could lead to illegal, immoral, or harmful behavior. It should promote ethical conduct and responsible communication at all times.

- **Debiasing:** The model should produce responses that are free from bias, including but not limited to gender, race, nationality, and politics. It should treat all topics equally and fairly, adhering to widely accepted moral and ethical standards.

The queries utilized to train the reward model are drawn from two distinct datasets: publicly available open-source data and a proprietary query set characterized by higher complexity. Responses are generated from checkpoints of the Qwen models, which have been fine-tuned using different methods—SFT, DPO, and RL—at various stages of training. To introduce diversity, those responses are sampled at different temperature settings. Preference pairs are created through both human and automated labeling processes, and the training data for DPO is also integrated into this dataset.

In our online reinforcement learning (RL) framework, we employ Group Relative Policy Optimization (GRPO, Shao et al., 2024). The query set utilized for training the reward model is identical to the one used in the RL training phase. The sequence in which queries are processed during training is determined by the variance of their response scores, as evaluated by the reward model. Specifically, queries with higher variance in response scores are prioritized to ensure more effective learning. We sample 8 responses for each query. All models are trained with a 2048 global batch size and 2048 samples in each episode, considering a pair of queries and responses as a sample.

4.4 Long Context Fine-tuning

To further extend the context length of Qwen2.5-Turbo, we introduce longer SFT examples during post-training, enabling it to better align with human preference in long queries.

In the SFT phase, we employ a two-stage approach. In the first stage, the model is fine-tuned exclusively using short instructions, each containing up to 32,768 tokens. This stage uses the same data and training steps as those employed for the other Qwen2.5 models, ensuring strong performance on short tasks. In the second stage, the fine-tuning process combines both short instructions (up to 32,768 tokens) and long instructions (up to 262,144 tokens). This hybrid approach effectively enhances the model's instruction-following ability in long context tasks while maintaining its performance on short tasks.

During the RL stage, we use a training strategy similar to that used for the other Qwen2.5 models, focusing solely on short instructions. This design choice is driven by two primary considerations: first, RL training is computationally expensive for long context tasks; second, there is currently a scarcity of reward models that provide suitable reward signals for long context tasks. Additionally, we find that adopting RL on short instructions alone can still significantly enhance the model's alignment with human preferences in long context tasks.

5 Evaluation

The base models produced by pre-training and the instruction-tuned models produced by post-training are evaluated accordingly with a comprehensive evaluation suite, including both commonly-used open benchmarks and skill-oriented in-house datasets. The evaluation suite is designed to be primarily automatic with minimal human interaction.

To prevent test data leakage, we exclude potentially contaminated data using n-gram matching when constructing the pre-training and post-training datasets. Following the criteria used in Qwen2, a training sequence $\mathbf{s}_t$ is removed from the training data if there exists a test sequence $\mathbf{s}_e$ such that the length of the longest common subsequence (LCS) between tokenized $\mathbf{s}_t$ and $\mathbf{s}_e$ satisfies both $|\text{LCS}(\mathbf{s}_t, \mathbf{s}_e)| \geq 13$ and $|\text{LCS}(\mathbf{s}_t, \mathbf{s}_e)| \geq 0.6 \times \min(|\mathbf{s}_t|, |\mathbf{s}_e|)$.

5.1 Base Models

We conduct comprehensive evaluations of the base language models of the Qwen2.5 series. The evaluation of base models primarily emphasizes their performance in natural language understanding, general question answering, coding, mathematics, scientific knowledge, reasoning, and multilingual capabilities.

The evaluation datasets include:

General Tasks MMLU (Hendrycks et al., 2021a) (5-shot), MMLU-Pro (Wang et al., 2024b) (5-shot), MMLU-redux (Gema et al., 2024) (5-shot), BBH (Suzgun et al., 2023) (3-shot), ARC-C (Clark et al., 2018) (25-shot), TruthfulQA (Lin et al., 2022a) (0-shot), Winogrande (Sakaguchi et al., 2021) (5-shot), HellaSwag (Zellers et al., 2019) (10-shot).

Table 2: **Performance of the 70B+ base models and Qwen2.5-Plus.**

Datasets	Llama-3-70B	Mixtral-8x22B	Llama-3-405B	Qwen2-72B	Qwen2.5-72B	Qwen2.5-Plus
General Tasks						
MMLU	79.5	77.8	85.2	84.2	**86.1**	85.4
MMLU-Pro	52.8	51.6	61.6	55.7	58.1	**64.0**
MMLU-redux	75.0	72.9	-	80.5	**83.9**	82.8
BBH	81.0	78.9	85.9	82.4	**86.3**	85.8
ARC-C	68.8	70.7	-	68.9	**72.4**	70.9
TruthfulQA	45.6	51.0	-	54.8	**60.4**	55.3
WindoGrande	85.3	85.0	**86.7**	85.1	83.9	85.5
HellaSwag	88.0	88.7	-	87.3	87.6	**89.2**
Mathematics & Science Tasks						
GPQA	36.3	34.3	-	37.4	**45.9**	43.9
TheoremQA	32.3	35.9	-	42.8	42.4	**48.5**
MATH	42.5	41.7	53.8	50.9	62.1	**64.4**
MMLU-stem	73.7	71.7	-	79.6	**82.7**	81.2
GSM8K	77.6	83.7	89.0	89.0	91.5	**93.0**
Coding Tasks						
HumanEval	48.2	46.3	**61.0**	64.6	59.1	59.1
HumanEval+	42.1	40.2	-	**56.1**	51.2	52.4
MBPP	70.4	71.7	73.0	76.9	**84.7**	79.7
MBPP+	58.4	58.1	-	63.9	**69.2**	66.9
MultiPL-E	46.3	46.7	-	59.6	60.5	**61.0**
Multilingual Tasks						
Multi-Exam	70.0	63.5	-	76.6	**78.7**	78.5
Multi-Understanding	79.9	77.7	-	80.7	**89.6**	89.2
Multi-Mathematics	67.1	62.9	-	76.0	76.7	**82.4**
Multi-Translation	38.0	23.3	-	37.8	39.0	**40.4**

Mathematics & Science Tasks GPQA (Rein et al., 2023) (5-shot), Theorem QA (Chen et al., 2023a) (5-shot), GSM8K (Cobbe et al., 2021) (4-shot), MATH (Hendrycks et al., 2021b) (4-shot).

Coding Tasks HumanEval (Chen et al., 2021) (0-shot), HumanEval+ (Liu et al., 2023)(0-shot), MBPP (Austin et al., 2021) (0-shot), MBPP+ (Liu et al., 2023) (0-shot), MultiPL-E (Cassano et al., 2023) (0-shot) (Python, C++, JAVA, PHP, TypeScript, C#, Bash, JavaScript).

Multilingual Tasks We group them into four categories: (a) Exam: M3Exam (5-shot, we only choose examples that require no image), IndoMMLU (Koto et al., 2023) (3-shot), ruMMLU (Fenogenova et al., 2024) (5-shot), and translated MMLU (Chen et al., 2023b) (5-shot on Arabic, Spanish, French, Portuguese, German, Italian, Japanese, and Korean); (b) Understanding: BELEBELE (Bandarkar et al., 2023) (5-shot), XCOPA (Ponti et al., 2020) (5-shot), XWinograd (Muennighoff et al., 2023) (5-shot), XStoryCloze (Lin et al., 2022b) (0-shot) and PAWS-X (Yang et al., 2019) (5-shot); (c) Mathematics: MGSM (Goyal et al., 2022) (8-shot CoT); and (d) Translation: Flores-101 (Goyal et al., 2022) (5-shot).

For base models, we compare Qwen2.5 models with Qwen2 models and other leading open-weight models in terms of scales of parameters.

Qwen2.5-72B & Qwen2.5-Plus We compare the base models of Qwen2.5-72B and Qwen2.5-Plus to other leading open-weight base models: Llama3-70B (Dubey et al., 2024), Llama3-405B (Dubey et al., 2024), Mixtrail-8x22B (Jiang et al., 2024a), and our previous 72B version, the Qwen2-72B (Yang et al., 2024a). The Qwen2.5-72B base model significantly outperforms its peers in the same category across a wide range of tasks. It achieves results comparable to Llama-3-405B while utilizing only one-fifth of the parameters. Furthermore, when compared to its predecessor, Qwen2-72B, the Qwen2.5-72B shows marked improvements in nearly all benchmark evaluations, particularly excelling in general tasks, mathematics, and coding challenges. With significantly lower training and inference costs, Qwen2.5-Plus achieves very competitive performance results compared to Qwen2.5-72B and Llama3-405B, outperforming other baseline models on the Hellaswag, TheoremQA, MATH, GSM8K, MultiPL-E, Multi-Mathematics, and Multi-Translation. Moreover, Qwen2.5-Plus achieves 64.0 on MMLU-Pro, which is 5.9 points higher than Qwen2.5-72B.

Qwen2.5-14B/32B & Qwen2.5-Turbo The evaluation of the Qwen2.5-Turbo, Qwen2.5-14B, and 32B models is compared against baselines of similar sizes. These baselines include Yi-1.5-34B (Young et al.,

Table 3: **Performance of the 14B-30B+ base models and Qwen2.5-Turbo.**

Datasets	Qwen1.5-32B	Gemma2-27B	Yi-1.5-34B	Qwen2.5-Turbo	Qwen2.5-14B	Qwen2.5-32B
General Tasks						
MMLU	74.3	75.2	77.2	79.5	79.7	**83.3**
MMLU-pro	44.1	49.1	48.3	**55.6**	51.2	55.1
MMLU-redux	69.0	-	74.1	77.1	76.6	**82.0**
BBH	66.8	74.9	76.4	76.1	78.2	**84.5**
ARC-C	63.6	**71.4**	65.6	67.8	67.3	70.4
TruthfulQA	57.4	40.1	53.9	56.3	**58.4**	57.8
Winogrande	81.5	59.7	**84.9**	81.1	81.0	82.0
Hellaswag	85.0	**86.4**	85.9	85.0	84.3	85.2
Mathematics & Science Tasks						
GPQA	30.8	34.9	37.4	41.4	32.8	**48.0**
Theoremqa	28.8	35.8	40.0	42.1	43.0	**44.1**
MATH	36.1	42.7	41.7	55.6	55.6	**57.7**
MMLU-stem	66.5	71.0	72.6	77.0	76.4	**80.9**
GSM8K	78.5	81.1	81.7	88.3	90.2	**92.9**
Coding Tasks						
HumanEval	43.3	54.9	46.3	57.3	56.7	**58.5**
HumanEval+	40.2	46.3	40.2	51.2	51.2	**52.4**
MBPP	64.2	75.7	65.5	76.2	76.7	**84.5**
MBPP+	53.9	60.2	55.4	63.0	63.2	**67.2**
MultiPL-E	38.5	48.0	39.5	53.9	53.5	**59.4**
Multilingual Tasks						
Multi-Exam	61.6	65.8	58.3	70.3	70.6	**75.4**
Multi-Understanding	76.5	82.2	73.9	85.3	85.9	**88.4**
Multi-Mathematics	56.1	61.6	49.3	71.3	68.5	**73.7**
Multi-Translation	33.5	**38.7**	30.0	36.8	36.2	37.3

Table 4: **Performance of the 7B+ base models.**

Datasets	Mistral-7B	Llama3-8B	Gemma2-9B	Qwen2-7B	Qwen2.5-7B
General Tasks					
MMLU	64.2	66.6	71.3	70.3	**74.2**
MMLU-pro	30.9	35.4	44.7	40.1	**45.0**
MMLU-redux	58.1	61.6	67.9	68.1	**71.1**
BBH	56.1	57.7	68.2	62.3	**70.4**
ARC-C	60.0	59.3	**68.2**	60.6	63.7
TruthfulQA	42.2	44.0	45.3	54.2	**56.4**
Winogrande	78.4	77.4	**79.5**	77.0	75.9
HellaSwag	**83.3**	82.1	81.9	80.7	80.2
Mathematics & Science Tasks					
GPQA	24.7	25.8	32.8	30.8	**36.4**
TheoremQA	19.2	22.1	28.9	29.6	**36.0**
MATH	10.2	20.5	37.7	43.5	**49.8**
MMLU-stem	50.1	55.3	65.1	64.2	**72.3**
GSM8K	36.2	55.3	70.7	80.2	**85.4**
Coding Tasks					
HumanEval	29.3	33.5	37.8	51.2	**57.9**
HumanEval+	24.4	29.3	30.5	43.3	**50.6**
MBPP	51.1	53.9	62.2	64.2	**74.9**
MBPP+	40.9	44.4	50.6	51.9	**62.9**
MultiPL-E	29.4	22.6	34.9	41.0	**50.3**
Multilingual Tasks					
Multi-Exam	47.1	52.3	**61.2**	59.2	59.4
Multi-Understanding	63.3	68.6	78.3	72.0	**79.3**
Multi-Mathematics	26.3	36.3	53.0	57.5	**57.8**
Multi-Translation	23.3	31.9	**36.5**	31.5	32.4

Table 5: **Performance of the smaller base models.**

Datasets	Qwen2-0.5B	Qwen2.5-0.5B	Qwen2-1.5B	Qwen2.5-1.5B	Gemma2-2.6B	Qwen2.5-3B
General Tasks						
MMLU	44.3	47.5	55.9	60.9	52.2	**65.6**
MMLU-pro	14.7	15.7	21.6	28.5	23.0	**34.6**
MMLU-redux	40.7	45.1	51.8	58.5	50.9	**63.7**
BBH	18.2	20.3	36.5	45.1	41.9	**56.3**
ARC-C	31.0	35.6	43.7	54.7	55.7	**56.5**
TruthfulQA	39.7	40.2	45.9	46.6	36.2	**48.9**
Winogrande	56.9	56.3	65.0	65.0	**71.5**	71.1
Hellaswag	49.1	52.1	67.0	67.9	**74.6**	**74.6**
Mathematics & Science Tasks						
GPQA	**29.8**	24.8	20.7	24.2	25.3	26.3
TheoremQA	9.6	16.0	14.8	22.1	15.9	**27.4**
MATH	11.2	19.5	21.6	35.0	18.3	**42.6**
MMLU-STEM	27.5	39.8	42.7	54.8	45.8	**62.5**
GSM8K	36.4	41.6	46.9	68.5	30.3	**79.1**
Coding Tasks						
HumanEval	22.6	30.5	34.8	37.2	19.5	**42.1**
HumanEval+	18.9	26.8	29.9	32.9	15.9	**36.0**
MBPP	33.1	39.3	46.9	**60.2**	42.1	57.1
MBPP+	27.6	33.8	37.6	**49.6**	33.6	49.4
MultiPL-E	16.3	18.9	27.9	33.1	17.6	**41.2**
Multilingual Tasks						
Multi-Exam	29.4	30.8	43.1	47.9	38.1	**54.6**
Multi-Understanding	40.4	41.0	50.7	65.1	46.8	**76.6**
Multi-Mathematics	7.8	13.5	21.3	37.5	18.2	**48.9**
Multi-Translation	14.1	15.3	23.8	25.0	26.9	**29.3**

2024), Gemma2-27B (Gemma Team et al., 2024), and Qwen1.5-32B (Qwen Team, 2024b). The results are shown in Table 3. The Qwen2.5-14B model demonstrates a solid performance across various tasks, particularly excelling in general tasks like MMLU and BBH, where it achieves scores of 79.7 and 78.2, outcompeting competitors of larger sizes. Meanwhile, Qwen2.5-32B, in particular, showcases exceptional capabilities, often surpassing larger models of similar model sizes. Notably, it outperforms its predecessor Qwen1.5-32B significantly, especially in challenging areas such as mathematics and coding, with notable scores of 57.7 in MATH and 84.5 in MBPP. For Qwen2.5-Turbo, although its training cost and inference cost are significantly smaller than those of Qwen2.5-14B, it achieves comparable results, where its MMLU-Pro score is even better than that of Qwen2.5-32B.

Qwen2.5-7B For 7B-level models, we focus on comparing Qwen2.5-7B with other leading 7B+ models, including Mistral-7B (Jiang et al., 2023a), Llama3-8B (Dubey et al., 2024), Gemma2-9B (Gemma Team et al., 2024), and our predecessor, Qwen2-7B (Yang et al., 2024a). The results can be found in Table 4. Note that the non-embedding parameters of Qwen2-7B and Qwen2.5-7B are only 6.5B, while that of Gemma2-9B is 8.2B. The Qwen2.5-7B model surpasses its predecessors and counterparts in numerous benchmarks, despite having fewer non-embedding parameters. It demonstrates significant improvements across various tasks, achieving 74.2 on general benchmarks like MMLU (Hendrycks et al., 2021a), 49.8 on math challenges such as MATH (Hendrycks et al., 2021b), and 57.9 on coding tasks like HumanEval (Chen et al., 2021).

Qwen2.5-0.5B/1.5B/3B For edge-side models, we compare Qwen2.5-0.5B, 1.5B, and 3B against established baselines: Qwen2-0.5B/1.5B (Yang et al., 2024a) and Gemma2-2.6B (Gemma Team et al., 2024). The results are given in Table 5. Qwen2.5-0.5B, 1.5B, and 3B continue to maintain strong performance across nearly all benchmarks. Notably, the Qwen2.5-0.5B model outperforms the Gemma2-2.6B on various math and coding tasks.

5.2 Instruction-tuned Model

To critically evaluate instruction-tuned models, we adopt a multifaceted approach. Foundational skills and human preferences are assessed using open datasets and benchmarks. Additionally, our detailed in-house evaluations delve deeper into the models'competencies in key areas and multilingualism. A particular focus is placed on assessing long-context capability. The subsequent sections outline the evaluation methods and present the results.

Table 6: **Performance of the 70B+ Instruct models and Qwen2.5-Plus.**

Datasets	Llama-3.1-70B	Llama-3.1-405B	Qwen2-72B	Qwen2.5-72B	Qwen2.5-Plus
General Tasks					
MMLU-Pro	66.4	**73.3**	64.4	71.1	72.5
MMLU-redux	83.0	86.2	81.6	**86.8**	86.3
LiveBench 0831	46.6	53.2	41.5	52.3	**54.6**
Mathematics & Science Tasks					
GPQA	46.7	**51.1**	42.4	49.0	49.7
MATH	68.0	73.8	69.0	83.1	**84.7**
GSM8K	95.1	**96.8**	93.2	95.8	96.0
Coding Tasks					
HumanEval	80.5	**89.0**	86.0	86.6	87.8
MBPP	84.2	84.5	80.2	**88.2**	85.5
MultiPL-E	68.2	73.5	69.2	75.1	**77.0**
LiveCodeBench	32.1	41.6	32.2	**55.5**	51.4
Alignment Tasks					
IFEval	83.6	86.0	77.6	84.1	**86.3**
Arena-Hard	55.7	69.3	48.1	81.2	**81.4**
MTbench	8.79	9.08	9.12	**9.35**	9.30

Table 7: **Performance of the 14B-30B+ instruction-tuned models and Qwen2.5-Turbo.**

Datasets	Qwen2-57BA14B	Gemma2-27B	GPT4o-mini	Qwen2.5-Turbo	Qwen2.5-14B	Qwen2.5-32B
General Tasks						
MMLU-Pro	52.8	55.5	63.1	64.5	63.7	**69.0**
MMLU-redux	72.6	75.7	81.5	81.7	80.0	**83.9**
LiveBench 0831	31.1	39.6	43.3	42.3	44.4	**50.7**
Mathematics & Science Tasks						
GPQA	34.3	38.4	40.2	42.3	45.5	**49.5**
MATH	49.1	54.4	70.2	81.1	80.0	**83.1**
GSM8K	85.3	90.4	93.2	93.8	94.8	**95.9**
Coding Tasks						
HumanEval	79.9	78.7	**88.4**	86.6	83.5	**88.4**
MBPP	70.9	81.0	**85.7**	82.8	82.0	84.0
MultiPL-E	66.4	67.4	75.0	73.7	72.8	**75.4**
LiveCodeBench	22.5	-	40.7	37.8	42.6	**51.2**
Alignment Tasks						
IFEval	59.9	77.1	80.4	76.3	**81.0**	79.5
Arena-Hard	17.8	57.5	**74.9**	67.1	68.3	74.5
MTbench	8.55	9.10	-	8.81	8.88	**9.20**

5.2.1 Open Benchmark Evaluation

To comprehensively evaluate the quality of instruction-tuned models, we compile automatic and human evaluation to assess the capabilities and human preference. For the evaluation of basic capabilities, we apply similar datasets in the pre-trained model evaluation, which target on natural language understanding, coding, mathematics, and reasoning. Specifically, we evaluate on MMLU-Pro, MMLU-redux and LiveBench 0831 (White et al., 2024) for general evaluation, GPQA, GSM8K and MATH for science and mathematics, HumanEval, MBPP, MultiPL-E and LiveCodeBench 2305-2409 (Jain et al., 2024) for coding, IFEval (Zhou et al., 2023)[2] for instruction following. Additionally, we assess the performance of human preference alignment and instruction following by evaluating on benchmarks including MT-Bench (Zheng et al., 2023) and Arena-Hard (Li et al., 2024).

Qwen2.5-72B-Instruct & Qwen2.5-Plus As shown in Table 6, we compare Qwen2.5-72B-Instruct and Qwen2.5-Plus to other leading open-weight instrution-tuned models: Llama3.1-70B-Instruct (Dubey

[2]For simplicity, we report the results of the subset *strict-prompt*.

Table 8: **Performance of the 7B+ instruction-tuned models.**

Datasets	Gemma2-9B	Llama3.1-8B	Qwen2-7B	Qwen2.5-7B
General Tasks				
MMLU-Pro	52.1	48.3	44.1	**56.3**
MMLU-redux	72.8	67.2	67.3	**75.4**
LiveBench 0831	30.6	26.7	29.2	**35.9**
Mathematics & Science Tasks				
GPQA	32.8	32.8	34.3	**36.4**
MATH	44.3	51.9	52.9	**75.5**
GSM8K	76.7	84.5	85.7	**91.6**
Coding Tasks				
HumanEval	68.9	72.6	79.9	**84.8**
MBPP	74.9	69.6	67.2	**79.2**
MultiPL-E	53.4	50.7	59.1	**70.4**
LiveCodeBench	18.9	8.3	23.9	**28.7**
Alignment Tasks				
IFEval	70.1	**75.9**	54.7	71.2
Arena-Hard	41.6	27.8	25.0	**52.0**
MTbench	8.49	8.23	8.26	**8.75**

Table 9: **Performance comparison of 2B-4B instruction-tuned models.**

Datasets	Gemma2-2B	Phi3.5-Mini	MiniCPM3-4B	Qwen2.5-3B
Non-Emb Params	2.0B	3.6B	4.0B	2.8B
General Tasks				
MMLU-Pro	26.7	**47.5**	43.0	43.7
MMLU-redux	51.9	**67.7**	59.9	64.4
LiveBench 0831	20.1	27.4	**27.6**	26.8
Mathematics & Science Tasks				
GPQA	29.3	27.2	**31.3**	30.3
MATH	26.6	48.5	46.6	**65.9**
GSM8K	63.2	86.2	81.1	**86.7**
Coding Tasks				
HumanEval	68.9	72.6	**74.4**	**74.4**
MBPP	**74.9**	63.2	72.5	72.7
MultiPL-E	30.5	47.2	49.1	**60.2**
LiveCodeBench	5.8	15.8	**23.8**	19.9
Alignment Tasks				
IFEval	51.0	52.1	**68.4**	58.2

et al., 2024), Llama3.1-405B-Instruct (Dubey et al., 2024), and our previous 72B version, Qwen2-72B-Instruct (Yang et al., 2024a). The Qwen2.5-72B-Instruct model delivers exceptional performance, even surpassing the larger Llama-3.1-405B-Instruct in several critical benchmarks including MMLU-redux, MATH, MBPP, MultiPL-E, LiveCodeBench, Arena-Hard and MTBench. Moreover, Qwen2.5-Plus outperforms Qwen2.5-72B-Instruct on 9 out of 13 benchmarks.

Qwen2.5-14B/32B-Instruct & Qwen2.5-Turbo The performance of the Qwen2.5-Turbo, Qwen2.5-14B-Instruct, and Qwen2.5-32B-Instruct models is evaluated and compared against baselines of similar sizes. The baselines include GPT4o-mini, Gemma2-27B-IT (Gemma Team et al., 2024), and Qwen2-57BA14B-Instruct (Yang et al., 2024a). The results are summarized in Table 7. The Qwen2.5-32B-Instruct model exhibits superior performance across most tasks when compared to other models of similar size. Notably, our open-weight Qwen2.5-14B-Instruct model delivers competitive results across all benchmarks, rivaling those of GPT-4o-mini. Despite its significantly lower training and inference costs, the Qwen2.5-Turbo model outperforms Qwen2.5-14B-Instruct on eight out of ten benchmarks. This demonstrates that Qwen2.5-Turbo achieves remarkable efficiency and effectiveness, making it a compelling choice for resource-constrained environments.

Table 10: **Performance comparison of 0.5B-1.5B instruction-tuned models.**

Datasets	Qwen2-0.5B	Qwen2.5-0.5B	Qwen2-1.5B	Qwen2.5-1.5B
General Tasks				
MMLU-Pro	14.4	**15.0**	22.9	**32.4**
MMLU-redux	12.9	**24.1**	41.2	**50.7**
LiveBench	7.4	**12.6**	12.4	**18.8**
Mathematics & Science Tasks				
GPQA	23.7	**29.8**	21.2	**29.8**
MATH	13.9	**34.4**	25.3	**55.2**
GSM8K	40.1	**49.6**	61.6	**73.2**
Coding Tasks				
HumanEval	31.1	**35.4**	42.1	**61.6**
MBPP	39.7	**49.6**	44.2	**63.2**
MultiPL-E	20.8	**28.5**	38.5	**50.4**
LiveCodeBench	1.6	**5.1**	4.5	**14.8**
Alignment Tasks				
IFEval	14.6	**27.9**	29.0	**42.5**

Table 11: **Performance Comparison on our in-house English automatic evaluation benchmark.**

Models	IF	Knowledge	Comprehension	Coding	Math	Reasoning
Proprietary LLMs						
GPT-4o-2024-08-06	83.28	68.08	76.51	58.05	52.36	66.45
GPT-4o-2024-11-20	80.06	65.25	79.07	60.19	49.74	67.07
Claude3.5-sonnet-2024-10-22	84.22	74.61	79.02	67.17	48.67	70.20
Qwen2 Series						
Qwen2-0.5B-Instruct	18.33	18.59	30.64	5.42	13.16	32.03
Qwen2-1.5B-Instruct	29.42	29.23	45.81	17.02	20.34	38.86
Qwen2-7B-Instruct	50.47	44.79	58.04	43.04	38.31	50.25
Qwen2-72B-Instruct	76.08	59.49	72.19	48.95	48.07	60.33
Llama-3.1 Series						
Llama-3.1-70B-Instruct	81.33	63.42	69.29	55.96	48.00	63.18
Llama-3.1-405B-Instruct	83.33	67.10	75.55	58.14	47.09	64.74
Qwen2.5 Series						
Qwen2.5-0.5B-Instruct	33.35	30.29	29.78	15.41	26.29	36.13
Qwen2.5-1.5B-Instruct	40.25	41.19	47.69	26.19	40.99	42.23
Qwen2.5-3B-Instruct	60.60	46.11	57.98	41.43	49.38	49.80
Qwen2.5-7B-Instruct	70.01	52.74	62.69	48.41	56.93	54.69
Qwen2.5-14B-Instruct	74.17	59.78	69.11	52.68	59.68	62.51
Qwen2.5-Turbo	72.76	58.56	68.70	54.48	57.77	61.06
Qwen2.5-32B-Instruct	76.79	64.08	71.28	58.90	60.97	65.49
Qwen2.5-72B-Instruct	82.65	66.09	74.43	60.41	59.73	65.90
Qwen2.5-Plus	83.18	68.41	79.35	59.58	62.52	66.92

Other Instruction-tuned Models As illustrated in Table 8, the Qwen2.5-7B-Instruct model significantly outperforms its competitors, Gemma2-9B-IT and Llama3.1-8B-Instruct, across all tasks except IFEval. Notably, Qwen2.5-7B-Instruct exhibits clear advantages in mathematics (MATH: 75.5) and coding (HumanEval: 84.8). For the edge-side instruction models, the Qwen2.5-3B-Instruct model, despite having fewer parameters than both the Phi3.5-mini-instruct (Abdin et al., 2024) and MiniCPM3-4B-Instruct (Hu et al., 2024) models, surpasses them in mathematics and coding tasks, as shown in Table 9. Additionally, it delivers competitive results in language understanding. The Qwen2.5-1.5B-Instruct and Qwen2.5-0.5B-Instruct models have also seen substantial performance improvements over their previous versions, as detailed in Table 10. These enhancements make them particularly well-suited for edge-side applications in highly resource-constrained environments.

Table 12: **Performance Comparison on our in-house Chinese automatic evaluation benchmark.**

Models	IF	Knowledge	Comprehension	Coding	Math	Reasoning
Proprietary LLMs						
GPT-4o-2024-08-06	42.50	68.55	80.11	61.53	61.74	56.88
GPT-4o-2024-11-20	42.71	71.29	83.04	62.39	66.04	62.04
Claude3.5-sonnet-2024-10-22	49.25	72.09	82.16	66.00	63.71	66.60
Qwen2 Series						
Qwen2-0.5B-Instruct	4.69	40.43	39.13	9.85	14.07	32.73
Qwen2-1.5B-Instruct	6.81	51.54	46.89	14.14	24.57	35.19
Qwen2-7B-Instruct	16.83	65.95	60.30	37.05	50.52	44.96
Qwen2-72B-Instruct	31.98	74.96	75.49	41.57	65.55	58.19
Llama-3.1 Series						
Llama-3.1-70B-Instruct	28.96	57.41	67.24	54.82	41.18	52.42
Llama-3.1-405B-Instruct	30.39	63.79	72.27	60.73	46.05	55.88
Qwen2.5 Series						
Qwen2.5-0.5B-Instruct	6.12	39.13	42.97	9.60	24.03	33.72
Qwen2.5-1.5B-Instruct	7.38	48.68	49.69	22.96	37.30	39.17
Qwen2.5-3B-Instruct	16.50	57.18	62.55	29.88	51.64	39.57
Qwen2.5-7B-Instruct	26.64	65.77	67.55	39.56	61.06	49.70
Qwen2.5-14B-Instruct	26.87	70.28	76.96	49.78	67.01	56.41
Qwen2.5-Turbo	32.94	72.93	74.37	51.92	66.08	53.30
Qwen2.5-32B-Instruct	32.64	74.70	79.46	54.45	67.86	60.19
Qwen2.5-72B-Instruct	37.22	75.86	78.85	56.71	68.39	63.02
Qwen2.5-Plus	46.15	72.07	82.64	58.48	69.96	62.98

5.2.2 In-house Automatic Evaluation

Despite the availability of several open benchmark datasets for evaluation, we believe that these are insufficient to fully capture the capabilities of LLMs. To address this, we have developed a series of in-house datasets designed to assess various aspects of model performance, including knowledge understanding, text generation, coding, and more. These evaluations are conducted in both Chinese and English. In addition, we have specifically evaluated the multilingual performance of instruction-tuned models. The results are summarized in Table 11 for English, Table 12 for Chinese, Table 13 for multilingualism of 70B+ Instruct models, and Table 14 for 7B-14B models, respectively.

English & Chinese Evaluation We compare the performance of Qwen2.5-Instruct models against several leading language models, including GPT-4, Claude3.5-sonnet, Qwen2, and Llama-3.1, across both English and Chinese languages. Our analysis focuses on model size and its impact on performance, as well as how our latest Qwen2.5 series compares to previous iterations and competing models. For smaller models, we observe that the Qwen2.5-0.5B model achieves performance that is on par with or even surpasses the Qwen2-1.5B model. This indicates that the Qwen2.5 series has optimized parameter usage, enabling mid-sized models to achieve similar performance levels to larger models from the previous generation. The Qwen2.5-3B model demonstrates performance that is comparable to the Qwen2-7B model. Notably, the Qwen2.5-32B model exhibits a remarkable improvement over the Qwen2-72B model. Our flagship model, Qwen2.5-72B, further narrows the gap between Qwen and state-of-the-art models like GPT-4 and Claude3.5-sonnet. In particular, Qwen2.5-72B matches or exceeds the performance of Llama-3.1-405B in all metrics except for instruction following. This achievement underscores the competitiveness of Qwen2.5-72B in a wide range of language processing tasks, while also identifying areas for future improvement. Qwen2.5-Plus addresses the previous shortcomings in Chinese instruction following and further enhances its advantages in other areas.

Multilingual Evaluation To comprehensively evaluate the multilingual capabilities of instruction-tuned models, we followed P-MMEval (Zhang et al., 2024) and extended several benchmarks as follows: (1) IFEval (Multilingual): We expanded the IFEval benchmark, originally in English, to include multilingual examples. To ensure language neutrality, we removed instances that contained language-specific content (e.g., "start with letter A"). (2) Knowledge Utilization: to assess the knowledge utilization abilities of the Qwen2.5 series models across multiple languages, we employed five MMLU-like benchmarks (multiple-choice format). These benchmarks include: AMMLU (Arabic), JMMLU (Japanese), KMMLU (Korean), IndoMMLU (Indonesian), and TurkishMMLU (Turkish). Additionally, we evaluated the models' performance on the translated version of the MMLU benchmark (okapi_MMLU), which has been adapted

Table 13: **Performance of the 70B+ Instruct models on Multilingual Tasks.**

Datasets	Qwen2-72B	Llama3.1-70B	Qwen2.5-32B	Mistral-Large	GPT4o-mini	Qwen2.5-72B
Instruction Following						
IFEval (multilingual)	79.69	80.47	82.68	82.69	85.03	**86.98**
Knowledge						
AMMLU (Arabic)	68.85	70.08	70.44	69.24	69.73	**72.44**
JMMLU (Japanese)	77.37	73.89	76.55	75.77	73.74	**80.56**
KMMLU (Korean)	57.04	53.23	60.75	56.42	56.77	**61.96**
IndoMMLU (Indonesian)	66.31	67.50	66.42	63.21	67.75	**69.25**
TurkishMMLU (Turkish)	69.22	66.89	72.41	64.78	71.19	**76.12**
okapi MMLU (translated)	77.84	76.49	77.16	78.37	73.44	**79.97**
Math Reasoning						
MGSM8K (extended)	82.72	73.31	87.15	**89.01**	87.36	88.16
Cultural Nuances						
BLEnD	25.90	30.49	27.88	33.47	**35.91**	32.48

Table 14: **Performance of the 7B-14B Instruct models on Multilingual Tasks.**

Datasets	Qwen2-7B	Llama3.1-8B	Qwen2.5-7B	Gemma2-9B	Qwen2.5-14B
Instruction Following					
IFEval (multilingual)	51.43	60.68	74.87	**77.47**	77.08
Knowledge					
AMMLU (Arabic)	54.87	54.28	59.78	60.26	**66.81**
JMMLU (Japanese)	57.71	53.26	61.88	64.59	**72.78**
KMMLU (Korean)	43.96	42.28	46.59	46.24	**59.71**
IndoMMLU (Indonesian)	54.05	53.92	56.42	61.73	**65.09**
TurkishMMLU (Turkish)	49.27	45.61	54.28	55.44	**66.85**
okapi MMLU (translated)	60.47	55.18	66.98	46.72	**72.12**
Math Reasoning					
MGSM8K (extended)	56.13	66.05	66.11	78.37	**82.27**
Cultural Nuances					
BLEnD	22.49	19.47	23.66	**28.31**	26.99

into multiple languages from its original English form. (3) MGSM8K (Extended): Building upon the original MGSM8K benchmark, we extended the language support to include Arabic (ar), Korean (ko), Portuguese (pt), and Vietnamese (vi). (4) Cultural Nuances: To evaluate the models' ability to capture cultural nuances, we utilized the BLEnD benchmark (Myung et al., 2024). This benchmark is specifically designed to test LLMs on their understanding of cultural subtleties.

Qwen2.5 exhibits competitive performance in instruction following, multilingual knowledge, and mathematical reasoning, aligning well with models of comparable size. Although it shows notable improvements in capturing cultural nuances relative to its predecessor, Qwen2, there remains potential for further refinement in this domain.

5.2.3 Reward Model

The reward model serves as the cornerstone for guiding RL processes, and thus we conduct a separate evaluation of the reward model used in the Qwen2.5 series. Our assessment benchmarks encompass Reward Bench (Lambert et al., 2024), RMB (Zhou et al., 2024), PPE (Frick et al., 2024b), and an internally collected out-of-domain Chinese human preference benchmark (Human-Preference-Chinese) to provide a comprehensive analysis. For comparison, we included baseline models such as Nemotron-4-340B-Reward (Adler et al., 2024), Llama-3.1-Nemotron-70B-Reward (Wang et al., 2024c), and Athene-RM-70B (Frick et al., 2024a). The results are shown in Table 15. Overall, our findings indicate that Llama-3.1-Nemotron-70B-Reward excels on the Reward Bench, while Athene-RM-70B performs best on the RMB benchmark. The Qwen2.5-RM-72B, leads in both the PPE and Human-Preference-Chinese evaluations, ranking second only to Athene-RM-70B on the RMB and achieving a performance level comparable to

Table 15: **Performance comparison across multiple RM benchmarks.**

Metric	Nemotron-4-340B-Reward	Llama-3.1-Nemotron-70B-Reward	Athene-RM -70B	Qwen2.5-RM -72B
		Reward Bench		
Chat	95.80	97.50	**98.32**	97.21
Chat Hard	**87.10**	85.70	70.61	78.73
Safety	91.50	**95.10**	92.10	92.71
Reasoning	93.60	**98.10**	92.19	97.65
Score	92.00	**94.10**	88.32	91.59
		RMB		
Helpfulness (BoN)	48.85	61.02	**67.24**	65.72
Helpfulness (Pairwise)	68.70	75.28	**80.82**	78.83
Harmlessness (BoN)	50.92	52.00	**67.02**	56.35
Harmlessness (Pairwise)	70.84	69.96	**80.83**	73.94
Overall	59.83	64.57	**73.98**	68.71
		PPE		
Human Preference	59.28	64.32	**66.48**	64.80
IFEval	62.66	63.40	62.15	**67.97**
GPQA	56.56	59.14	59.26	**59.80**
MATH	65.12	69.73	79.14	**81.48**
MBPP-Plus	49.15	55.62	**67.97**	64.34
MMLU-Pro	69.69	70.20	**76.95**	75.66
Objective-Avg	60.64	63.62	69.09	**69.85**
		Human-Preference-Chinese		
Accuracy	50.46	59.95	61.11	**61.27**

Nemotron-4-340B-Reward on the Reward Bench, albeit slightly behind Llama-3.1-Nemotron-70B-Reward.

Due to the lack of evaluation methods for reward models, current reward models are typically evaluated using Reward Bench. However, our evaluation results from multiple RM benchmarks suggest that over-optimization on a specific benchmark may trigger Goodhart's law (Hoskin, 1996), resulting in degraded performance on other benchmarks and potentially impacting downstream alignment performance. This highlights the need for comprehensive evaluation of reward models across diverse benchmarks rather than relying solely on a single benchmark.

More importantly, through iterative experimentation, we have also come to recognize a critical limitation: current reward model evaluation benchmarks do not accurately predict the performance of the RL models trained under their guidance. In other words, a higher score on RM benchmarks does not necessarily correlate with superior performance of the resulting RL model. This insight underscores the need for further research into more predictive evaluation methods for reward models.

5.2.4 Long Context Capabilities

We utilize three benchmarks to evaluate long context capabilities of Qwen2.5 models: RULER (Hsieh et al., 2024), LV-Eval (Yuan et al., 2024), and Longbench-Chat (Bai et al., 2024). In LV-Eval, we adopt keyword recall as the reported score to mitigate the high rate of false negatives present in the original metrics.

The results are shown in Table 16 and Table 17. We can observe that the Qwen2.5 models, after equipping length extrapolation techniques (i.e., DCA + YARN), have demonstrated strong long context processing capabilities on the three datasets. Among them, Qwen2.5-72B-Instruct has shown the strongest performance across all context lengths, significantly outperforming existing open-weight long-context models as well as the proprietary models like GPT-4o-mini and GPT-4.

Furthermore, as shown in Figure 2, Qwen2.5-Turbo achieves 100% accuracy in the 1M-token passkey retrieval task, demonstrating its exceptional ability to capture detailed information from ultra-long contexts. We develop a sparse attention mechanism based on Minference (Jiang et al., 2024b) to significantly enhance inference speed, which is critical for user experience when processing long contexts. For sequences of 1M tokens, this approach reduces the computational load of the attention mechanism by 12.5 times. Figure 3 illustrates the time to first token (TTFT) of Qwen2.5-Turbo across various hardware configurations, where our method achieves a 3.2 to 4.3 times speedup.

Table 16: **Performance of Qwen2.5 Models on RULER.** *YARN+DCA* does not change the model behavior within 32K tokens.

Model	Claimed Length	RULER						
		Avg.	4K	8K	16K	32K	64K	128K
GLM4-9b-Chat-1M	1M	89.9	94.7	92.8	92.1	89.9	86.7	83.1
Llama-3-8B-Instruct-Gradient-1048k	1M	88.3	95.5	93.8	91.6	87.4	84.7	77.0
Llama-3.1-70B-Instruct	128K	89.6	96.5	95.8	95.4	94.8	88.4	66.6
GPT-4o-mini	128K	87.3	95.0	92.9	92.7	90.2	87.6	65.8
GPT-4	128K	91.6	96.6	96.3	95.2	93.2	87.0	81.2
Qwen2.5-7B-Instruct	128K	85.4	96.7	95.1	93.7	89.4	82.3	55.1
w/o DCA + YARN		80.1	96.7	95.1	93.7	89.4	74.5	31.4
Qwen2.5-14B-Instruct	128K	91.4	97.7	96.8	95.9	93.4	86.7	78.1
w/o DCA + YARN		86.5	97.7	96.8	95.9	93.4	82.3	53.0
Qwen2.5-32B-Instruct	128K	92.9	96.9	97.1	95.5	95.5	90.3	82.0
w/o DCA + YARN		88.0	96.9	97.1	95.5	95.5	85.3	57.7
Qwen2.5-72B-Instruct	128K	**95.1**	**97.7**	**97.2**	**97.7**	**96.5**	**93.0**	**88.4**
w/o DCA + YARN		90.8	97.7	97.2	97.7	96.5	88.5	67.0
Qwen2.5-Turbo	1M	93.1	97.5	95.7	95.5	94.8	90.8	84.5

Table 17: **Performance of Qwen2.5 Models on LV-Eval and LongBench-Chat.** *YARN+DCA* does not change the model behavior within 32k tokens.

Model	Claimed Length	LV-Eval					LongBench-Chat
		16k	32k	64k	128k	256k	
GLM4-9B-Chat-1M	1M	46.4	43.2	42.9	40.4	37.0	7.82
Llama-3-8B-Instruct-Gradient-1048k	1M	31.7	31.8	28.8	26.3	21.1	6.20
Llama-3.1-70B-Instruct	128k	48.6	47.4	42.9	26.2	N/A	6.80
GPT-4o-mini	128k	52.9	48.1	46.0	40.7	N/A	8.48
Qwen2.5-7B-Instruct	128k	55.9	49.7	48.0	41.1	36.9	7.42
w/o DCA + YARN		55.9	49.7	33.1	13.6	0.5	-
Qwen2.5-14B-Instruct	128k	53.0	50.8	46.8	43.6	39.4	8.04
w/o DCA + YARN		53.0	50.8	37.0	18.4	0.8	-
Qwen2.5-32B-Instruct	128k	56.0	53.6	48.8	45.3	41.0	8.70
w/o DCA + YARN		56.0	53.6	40.1	20.5	0.7	-
Qwen2.5-72B-Instruct	128k	**60.4**	**57.5**	**53.9**	**50.9**	**45.2**	**8.72**
w/o DCA + YARN		60.4	57.5	47.4	27.0	2.4	-
Qwen2.5-Turbo	1M	53.4	50.0	45.4	43.9	38.0	8.34

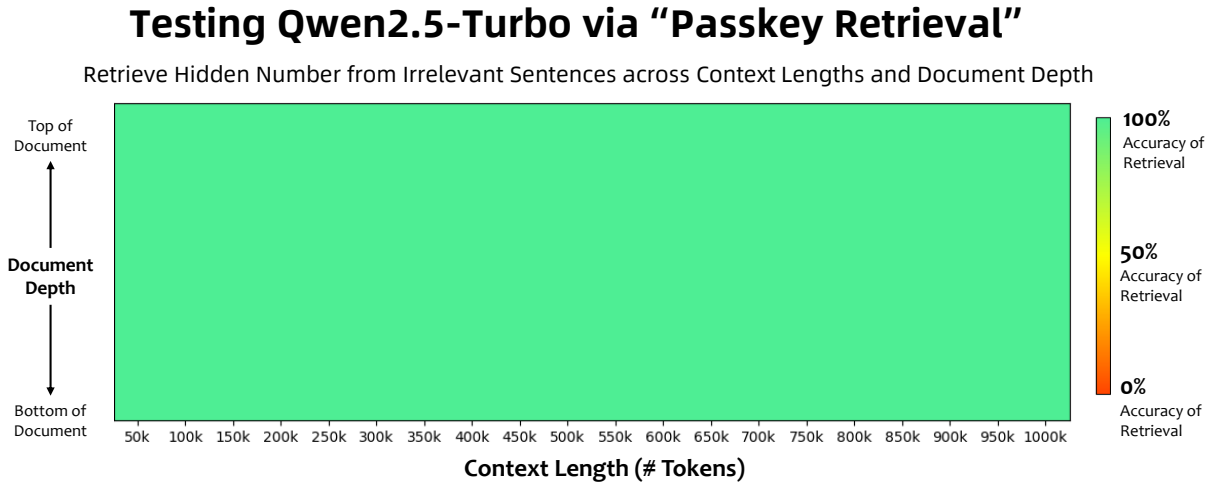

Figure 2: **Performance of Qwen2.5-Turbo on Passkey Retrieval Task with 1M Token Lengths.**

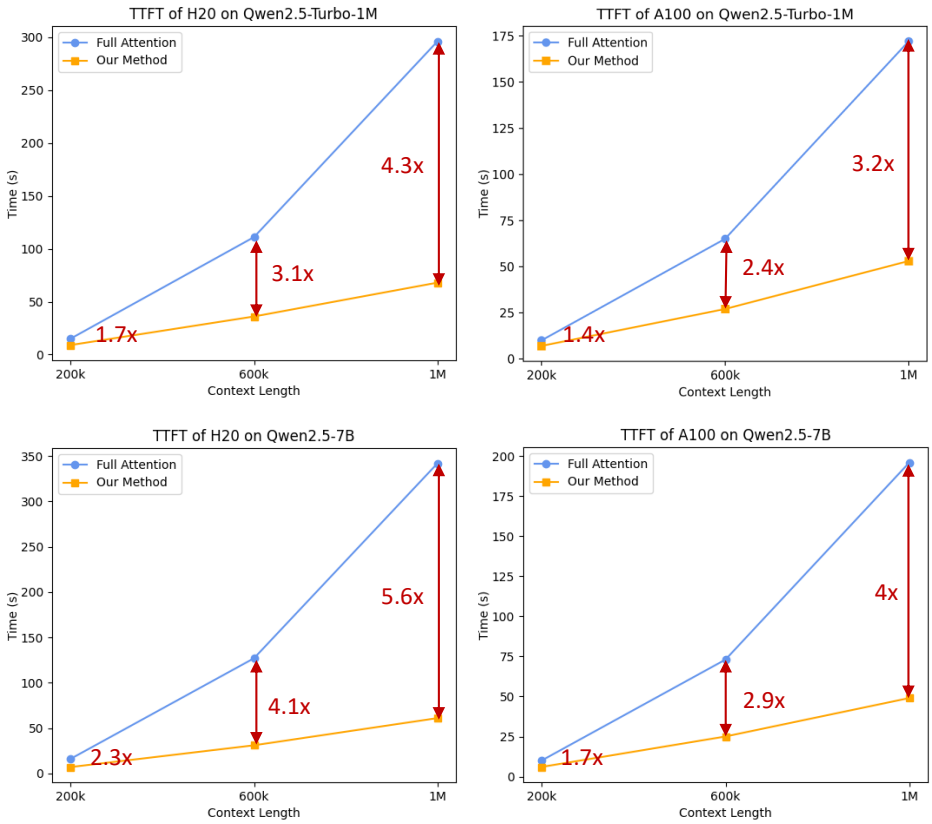

Figure 3: **TTFT (Time To First Token) of Qwen2.5-Turbo and Qwen2.5-7B with Full Attention and Our Method.**

6 Conclusion

Qwen2.5 represents a significant advancement in large language models (LLMs), with enhanced pre-training on 18 trillion tokens and sophisticated post-training techniques, including supervised fine-tuning and multi-stage reinforcement learning. These improvements boost human preference alignment, long text generation, and structural data analysis, making Qwen2.5 highly effective for instruction-following tasks. Available in various configurations, Qwen2.5 offers both open-weight from 0.5B to 72B parameters and proprietary models including cost-effective MoE variants like Qwen2.5-Turbo and Qwen2.5-Plus. Empirical evaluations show that Qwen2.5-72B-Instruct matches the performance of the state-of-the-art Llama-3-405B-Instruct, despite being six times smaller. Qwen2.5 also serves as a foundation for specialized models, demonstrating its versatility for domain-specific applications. We believe that Qwen2.5's robust performance, flexible architecture, and broad availability make it a valuable resource for both academic research and industrial applications, positioning it as a key player of future innovations.

In the future, we will focus on advancing robust foundational models. First, we will iteratively refine both base and instruction-tuned large language models (LLMs) by incorporating broader, more diverse, higher-quality data. Second, we will also continue to develop multimodal models. Our goal is to integrate various modalities into a unified framework. This will facilitate seamless, end-to-end information processing across textual, visual, and auditory domains. Third, we are committed to enhancing the reasoning capabilities of our models. This will be achieved through strategic scaling of inference compute resources. These efforts aim to push the boundaries of current technological limitations and contribute to the broader field of artificial intelligence.

7 Authors

Core Contributors: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia,

Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu

Contributors: Biao Sun, Bin Luo, Bin Zhang, Binghai Wang, Chaojie Yang, Chang Si, Cheng Chen, Chengpeng Li, Chujie Zheng, Fan Hong, Guanting Dong, Guobin Zhao, Hangrui Hu, Hanyu Zhao, Hao Lin, Hao Xiang, Haoyan Huang, Humen Zhong, Jialin Wang, Jialong Tang, Jiandong Jiang, Jianqiang Wan, Jianxin Ma, Jianyuan Zeng, Jie Zhang, Jin Xu, Jinkai Wang, Jinzheng He, Jun Tang, Ke Yi, Keqin Chen, Langshi Chen, Le Jiang, Lei Zhang, Liang Chen, Man Yuan, Mingkun Yang, Minmin Sun, Na Ni, Nuo Chen, Peng Wang, Peng Zhu, Pengcheng Zhang, Pengfei Wang, Qiaoyu Tang, Qing Fu, Rong Zhang, Ru Peng, Ruize Gao, Shanghaoran Quan, Shen Huang, Shuai Bai, Shuang Luo, Sibo Song, Song Chen, Tao He, Ting He, Wei Ding, Wei Liao, Weijia Xu, Wenbin Ge, Wenbiao Yin, Wenyuan Yu, Xianyan Jia, Xianzhong Shi, Xiaodong Deng, Xiaoming Huang, Ximing Zhou, Xinyu Wang, Xipin Wei, Xuejing Liu, Yang Liu, Yang Yao, Yang Zhang, Yibo Miao, Yidan Zhang, Yikai Zhu, Yinger Zhang, Yong Jiang, Yong Li, Yongan Yue, Yuanzhi Zhu, Yunfei Chu, Zekun Wang, Zhaohai Li, Zheren Fu, Zhi Li, Zhibo Yang, Zhifang Guo, Zhipeng Zhang, Zhiying Xu, Zile Qiao, Ziye Meng

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Peters Long, Ameya Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason D. Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340B technical report. *CoRR*, abs/2406.11704, 2024.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query Transformer models from multi-head checkpoints. In *EMNLP*, pp. 4895–4901. Association for Computational Linguistics, 2023.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The Falcon series of open language models. *CoRR*, abs/2311.16867, 2023.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models. *CoRR*, abs/2402.17463, 2024.

Anthropic. Introducing Claude, 2023a. URL https://www.anthropic.com/index/introducing-claude.

Anthropic. Claude 2. Technical report, Anthropic, 2023b. URL https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf.

Anthropic. The Claude 3 model family: Opus, Sonnet, Haiku. Technical report, Anthropic, AI, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *CoRR*, abs/2309.16609, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A recipe for long context alignment of large language models. In *EMNLP (Findings)*, pp. 1376–1395. Association for Computational Linguistics, 2024.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. The Belebele benchmark: A parallel reading comprehension dataset in 122 language variants. *CoRR*, abs/2308.16884, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

Boxi Cao, Keming Lu, Xinyu Lu, Jiawei Chen, Mengjie Ren, Hao Xiang, Peilin Liu, Yaojie Lu, Ben He, Xianpei Han, Le Sun, Hongyu Lin, and Bowen Yu. Towards scalable automated alignment of LLMs: A survey. *CoRR*, abs/2406.01252, 2024.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. MultiPL-E: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Trans. Software Eng.*, 49(7):3675–3691, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *EMNLP*, pp. 7889–7901. Association for Computational Linguistics, 2023a.

Zhihong Chen, Shuo Yan, Juhao Liang, Feng Jiang, Xiangbo Wu, Fei Yu, Guiming Hardy Chen, Junying Chen, Hongbo Zhang, Li Jianquan, Wan Xiang, and Benyou Wang. MultilingualSIFT: Multilingual supervised instruction fine-tuning, 2023b. URL https://github.com/FreedomIntelligence/MultilingualSIFT.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *CoRR*, abs/2401.06066, 2024.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 2017.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. Self-play with execution feedback: Improving instruction-following capabilities of large language models. *CoRR*, abs/2406.13542, 2024.

Shihan Dou, Jiazheng Zhang, Jianxiang Zang, Yunbo Tao, Haoxiang Jia, Shichun Liu, Yuming Yang, Shenxi Wu, Shaoqing Zhang, Muling Wu, et al. Multi-programming language sandbox for llms. *CoRR*, abs/2410.23074, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

Alena Fenogenova, Artem Chervyakov, Nikita Martynov, Anastasia Kozlova, Maria Tikhonova, Albina Akhmetgareeva, Anton A. Emelyanov, Denis Shevelev, Pavel Lebedev, Leonid Sinev, Ulyana Isaeva, Katerina Kolomeytseva, Daniil Moskovskiy, Elizaveta Goncharova, Nikita Savushkin, Polina Mikhailova, Denis Dimitrov, Alexander Panchenko, and Sergey Markov. MERA: A comprehensive LLM evaluation in russian. *CoRR*, abs/2401.04531, 2024.

Evan Frick, Peter Jin, Tianle Li, Karthik Ganesan, Jian Zhang, Jiantao Jiao, and Banghua Zhu. Athene-70b: Redefining the boundaries of post-training for open models, July 2024a. URL https://nexusflow.ai/blogs/athene.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. *CoRR*, abs/2410.14872, 2024b.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? *CoRR*, abs/2406.04127, 2024.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Technical report, Google, 2024. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10: 522–538, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*, 2021b.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022.

Keith Hoskin. The "awful idea of accountability": Inscribing people into the measurement of objects. *Accountability: Power, ethos and the technologies of managing*, 1996.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *CoRR*, abs/2404.06654, 2024.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhen Leng Thai, Kai Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *CoRR*, abs/2404.06395, 2024.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-Coder technical report. *CoRR*, abs/2409.12186, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *CoRR*, abs/2403.07974, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *CoRR*, abs/2310.06825, 2023a.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024a.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *arXiv preprint arXiv:2407.02490*, 2024b.

Zixuan Jiang, Jiaqi Gu, Hanqing Zhu, and David Z. Pan. Pre-RMSNorm and Pre-CRMSNorm Transformers: Equivalent and efficient pre-LN Transformers. *CoRR*, abs/2305.14858, 2023b.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020.

Fajri Koto, Nurul Aisyah, Haonan Li, and Timothy Baldwin. Large language models only pass primary school exams in Indonesia: A comprehensive test on IndoMMLU. In *EMNLP*, pp. 12359–12374. Association for Computational Linguistics, 2023.

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. RewardBench: Evaluating reward models for language modeling. *CoRR*, abs/2403.13787, 2024.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *CoRR*, abs/2006.16668, 2020.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-Hard and BenchBuilder pipeline. *CoRR*, abs/2406.11939, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL (1)*, pp. 3214–3252. Association for Computational Linguistics, 2022a.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual generative language models. In *EMNLP*, pp. 9019–9052. Association for Computational Linguistics, 2022b.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *NeurIPS*, 2023.

Keming Lu, Bowen Yu, Fei Huang, Yang Fan, Runji Lin, and Chang Zhou. Online merging optimizers for boosting rewards and mitigating tax in alignment. *CoRR*, abs/2405.17931, 2024a.

Keming Lu, Bowen Yu, Chang Zhou, and Jingren Zhou. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. *CoRR*, abs/2401.12474, 2024b.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In *ACL (1)*, pp. 15991–16111. Association for Computational Linguistics, 2023.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Pérez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Ki-Woong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, Nedjma Ousidhoum, José Camacho-Collados, and Alice Oh. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. *CoRR*, abs/2406.09948, 2024.

OpenAI. GPT4 technical report. *CoRR*, abs/2303.08774, 2023.

OpenAI. Hello GPT-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Learning to reason with LLMs, 2024b. URL https://openai.com/index/learning-to-reason-with-llms/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. *CoRR*, abs/2309.00071, 2023.

Edoardo Maria Ponti, Goran Glavas, Olga Majewska, Qianchu Liu, Ivan Vulic, and Anna Korhonen. XCOPA: A multilingual dataset for causal commonsense reasoning. In *EMNLP (1)*, pp. 2362–2376. Association for Computational Linguistics, 2020.

Shanghaoran Quan, Tianyi Tang, Bowen Yu, An Yang, Dayiheng Liu, Bofei Gao, Jianhong Tu, Yichang Zhang, Jingren Zhou, and Junyang Lin. Language models can self-lengthen to generate long texts. *CoRR*, abs/2410.23933, 2024.

Qwen Team. Code with CodeQwen1.5, 2024a. URL https://qwenlm.github.io/blog/codeqwen1.5/.

Qwen Team. Introducing Qwen1.5, 2024b. URL https://qwenlm.github.io/blog/qwen1.5/.

Qwen Team. Introducing Qwen2-Math, 2024c. URL https://qwenlm.github.io/blog/qwen2-math/.

Qwen Team. QwQ: Reflect deeply on the boundaries of the unknown, 2024d. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18332–18346. PMLR, 2022.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *CoRR*, abs/2311.12022, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL (1)*. The Association for Computer Linguistics, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

Jianlin Su. The magical effect of the Bias term: RoPE + Bias = better length extrapolation, 2023. URL https://spaces.ac.cn/archives/9577.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced Transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *ACL (Findings)*, pp. 13003–13051. Association for Computational Linguistics, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. Secrets of RLHF in large language models part II: Reward modeling. *CoRR*, abs/2401.06080, 2024a.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. In *AAAI*, pp. 9154–9160. AAAI Press, 2020.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574, 2024b.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. HelpSteer2-Preference: Complementing ratings with preferences. *CoRR*, abs/2410.01257, 2024c.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-free LLM benchmark. *CoRR*, abs/2406.19314, 2024.

Hao Xiang, Bowen Yu, Hongyu Lin, Keming Lu, Yaojie Lu, Xianpei Han, Le Sun, Jingren Zhou, and Junyang Lin. Aligning large language models via self-steering optimization. *CoRR*, abs/2410.17131, 2024.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models. *CoRR*, abs/2309.16039, 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122, 2024b.

Jian Yang, Jiaxi Yang, Ke Jin, Yibo Miao, Lei Zhang, Liqun Yang, Zeyu Cui, Yichang Zhang, Binyuan Hui, and Junyang Lin. Evaluating and aligning codellms on human preference. *CoRR*, abs/2412.05210, 2024c.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *EMNLP/IJCNLP (1)*, pp. 3685–3690. Association for Computational Linguistics, 2019.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.AI. *CoRR*, abs/2403.04652, 2024.

Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuyu Yao, Dahua Lin, Boxun Li, Guohao Dai, Shengen Yan, and Yu Wang. LV-Eval: A balanced long-context benchmark with 5 length levels up to 256K. *CoRR*, abs/2402.05136, 2024.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL (1)*, pp. 4791–4800. Association for Computational Linguistics, 2019.

Yidan Zhang, Boyi Deng, Yu Wan, Baosong Yang, Haoran Wei, Fei Huang, Bowen Yu, Junyang Lin, and Jingren Zhou. P-MMEval: A parallel multilingual multitask benchmark for consistent evaluation of LLMs. *CoRR*, abs/2411.09116, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *NeurIPS*, 2023.

Enyu Zhou, Guodong Zheng, Bing Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. RMB: Comprehensively benchmarking reward models in LLM alignment. *CoRR*, abs/2410.09893, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911, 2023.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *CoRR*, abs/2202.08906, 2022.