In the pretraining process, we initialize the model by corresponding base language models, CODE-QWEN-14B-CHAT and CODE-QWEN-7B-CHAT. Most applications depend on specialized models for coding may lead to long contextual scenarios, such as tool use and code interpreter in Section 3.4. In that case, we train models with context lengths of 8192. Similar to base model training in Section 2.4, we employ Flash Attention (Dao et al., 2022) in the attention modules, and adopt the standard optimizer AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017), setting $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. We set the learning rate as 3.0×10^{-5} for CODE-QWEN-7B and 6.0×10^{-5} CODE-QWEN-14B, with 3% warm up iterations and no learning rate decays.

4.2 CODE SUPERVISED FINE-TUNING

After conducting a series of empirical experiments, we have determined that the multi-stage SFT strategy yields the best performance compared to other methods. In the supervised fine-tuning stage, the models CODE-QWEN-7B-CHAT and CODE-QWEN-14B-CHAT initialized by the code foundation model CODE-QWEN-7B-CHAT and CODE-QWEN-14B-CHAT are optimized by the AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$) with a learning rate of 1.0×10^{-5} and 2.0×10^{-6}, respectively. The learning rate increases to the peaking value with the cosine learning rate schedule (3% warm-up steps) and remains constant.

4.3 EVALUATION

Our CODE-QWEN models have been compared with both proprietary and open-source language models, as shown in Tables 11 and 10. These tables present the results of our evaluation on the test sets of Humaneval (Chen et al., 2021b), MBPP (Austin et al., 2021), and the multi-lingual code generation benchmark HUMANEVALPACK (Muennighoff et al., 2023). The comparison is based on the pass@1 performance of the models on these benchmark datasets. The results of this comparison are clearly demonstrated in Tables 10 and 11.

Our analysis reveals that specialized models, specifically CODE-QWEN and CODE-QWEN-CHAT, significantly outperform previous baselines with similar parameter counts, such as OCTOGEEX (Muennighoff et al., 2023), InstructCodeT5+ (Wang et al., 2023d), and CodeGeeX2 (Zheng et al., 2023). In fact, these models even rival the performance of larger models like Starcoder (Li et al., 2023d) and WizardCoder-15B (Luo et al., 2023b).

When compared to some of the extremely large-scale closed-source models, CODE-QWEN and CODE-QWEN-CHAT demonstrate clear advantages in terms of pass@1. However, it is important to note that these models fall behind the state-of-the-art methods, such as GPT-4, in general. Nonetheless, with the continued scaling of both model size and data size, we believe that this gap can be narrowed in the near future.

It is crucial to emphasize that the evaluations mentioned previously are insufficient for grasping the full extent of the strengths and weaknesses of the models. In our opinion, it is crucial to develop more rigorous tests to enable us to accurately assess our relative performance in comparison to GPT-4.

5 MATH-QWEN: SPECIALIZED MODEL FOR MATHEMATICS REASONING

We have created a mathematics-specialized model series called MATH-QWEN-CHAT, which is built on top of the QWEN pretrained language models. Specifically, we have developed assistant models that are specifically designed to excel in arithmetic and mathematics and are aligned with human behavior. These models are referred to as MATH-QWEN-CHAT. We are releasing two versions of this model series, MATH-QWEN-14B-CHAT and MATH-QWEN-7B-CHAT, which have 14 billion and 7 billion parameters, respectively.

5.1 TRAINING

We carry out math SFT on our augmented math instructional dataset for mathematics reasoning, and therefore we obtain the chat model, MATH-QWEN-CHAT, directly. Owing to shorter average lengths of the math SFT data, we use a sequence length of 1024 for faster training. Most user inputs