visualization tasks, we differentiate between two levels of difficulty. The easier level can be achieved by simply writing and executing a single code snippet without the need for advanced planning skills. However, the more challenging level requires strategic planning and executing multiple code snippets in a sequential manner. This is because the subsequent code must be written based on the output of the previous code. For example, an agent may need to examine the structure of a CSV file using one code snippet before proceeding to write and execute additional code to create a plot.

Regarding evaluation metrics, we consider both the executability and correctness of the generated code. To elaborate on the correctness metrics, for math problems, we measure accuracy by verifying if the ground truth numerical answer is present in both the code execution result and the final response. When it comes to data visualization, we assess accuracy by utilizing QWEN-VL (Bai et al., 2023), a powerful multimodal language model. QWEN-VL is capable of answering text questions paired with images, and we rely on it to confirm whether the image generated by the code fulfills the user's request.

The results regarding executability and correctness are presented in Table 7 and Table 8, respectively. It is evident that CODE LLAMA generally outperforms LLAMA 2, its generalist counterpart, which is not surprising since this benchmark specifically requires coding skills. However, it is worth noting that specialist models that are optimized for code synthesis do not necessarily outperform generalist models. This is due to the fact that this benchmark encompasses various skills beyond coding, such as abstracting math problems into equations, understanding language-specified constraints, and responding in the specified format such as ReAct. Notably, QWEN-7B-CHAT and QWEN-14B-CHAT surpass all other open-source alternatives of similar scale significantly, despite being generalist models.

Serving as a Hugging Face Agent Hugging Face provides a framework called the Hugging Face Agent or Transformers Agent (Hugging Face, 2023), which empowers LLM agents with a curated set of multimodal tools, including speech recognition and image synthesis. This framework allows an LLM agent to interact with humans, interpret natural language commands, and employ the provided tools as needed.

To evaluate QWEN's effectiveness as a Hugging Face agent, we utilized the evaluation benchmarks offered by Hugging Face. The results are presented in Table 9. The evaluation results reveal that QWEN performs quite well in comparison to other open-source alternatives, only slightly behind the proprietary GPT-4, demonstrating QWEN's competitive capabilities.

4 CODE-QWEN: SPECIALIZED MODEL FOR CODING

Training on domain-specific data has been shown to be highly effective, particularly in the case of code pretraining and finetuning. A language model that has been reinforced with training on code data can serve as a valuable tool for coding, debugging, and interpretation, among other tasks. In this work, we have developed a series of generalist models using pretraining and alignment techniques. Building on this foundation, we have created domain-specific models for coding by leveraging the base language models of QWEN. We have implemented continued pretraining on code data for CODE-QWEN and then applied supervised finetuning to create CODE-QWEN-CHAT. The code models, CODE-QWEN-14B and CODE-QWEN-7B, are based on base language models with 14 billion and 7 billion parameters.

4.1 CODE PRETRAINING

Unlike previous approaches that focused solely on pretraining on code data (Li et al., 2022; 2023d), we take a different approach (Rozière et al., 2023) by starting with our base language models, CODE-QWEN-14B-CHAT and CODE-QWEN-7B-CHAT, and then continuing to pretrain on a combination of text and code data. We believe that relying solely on code data for pretraining can result in a significant loss of the ability to function as a versatile assistant. Additionally, incorporating data from a diverse range of domains can help enhance the models' ability to understand and generate code, as well as bridge the gap between language and coding. We continue to pretrain the models on a total of around 90 billion tokens.