SUMMARY 1:

Here is the summary of the research paper "DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence":

**Title:** DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence

**Authors:** Qihao Zhu\*, Daya Guo\*, Zhihong Shao\*, Dejian Yang\*, Peiyi Wang, Runxin Xu, Y. Wu, Yukun Li, Huazuo Gao, Shirong Ma, Wangding Zeng, Xiao Bi, Zihui Gu, Hanwei Xu, Damai Dai, Kai Dong, Liyue Zhang, Yishi Piao, Zhibin Gou, Zhenda Xie, Zhewen Hao, Bingxuan Wang, Junxiao Song, Deli Chen, Xin Xie, Kang Guan, Yuxiang You, Aixin Liu, Qiushi Du, Wenjun Gao, Xuan Lu, Qinyu Chen, Yaohui Wang, Chengqi Deng, Jiashi Li, Chenggang Zhao, Chong Ruan, Fuli Luo, Wenfeng Liang

**Abstract:** DeepSeek-Coder-V2 is an open-source Mixture-of-Experts (MoE) code language model that achieves performance comparable to GPT-4 Turbo in code-specific tasks. It is pre-trained from an intermediate checkpoint of DeepSeek-V2 with an additional 6 trillion tokens. This pre-training enhances the coding and mathematical reasoning capabilities while maintaining performance in general language tasks. DeepSeek-Coder-V2 supports 338 programming languages and extends the context length from 16K to 128K tokens, outperforming closed-source models like GPT-4 Turbo, Claude 3 Opus, and Gemini 1.5 Pro in standard benchmarks.

**Key Contributions:**

1. Introduction of DeepSeek-Coder-V2 with significant improvements over its predecessor.
2. Expansion of programming language support from 86 to 338 languages.
3. Increased context length from 16K to 128K tokens.
4. Superior performance in code and mathematical benchmarks compared to leading closed-source models.
5. Public release under a permissive license for both research and commercial use.

**Evaluation Results:**

* **HumanEval Benchmark:** DeepSeek-Coder-V2 achieved a score of 90.2%.
* **MBPP Benchmark:** Scored 76.2%, establishing a new state-of-the-art result with the EvalPlus evaluation pipeline.
* **LiveCodeBench:** Scored 43.4%, making it the first open-source model to surpass a score of 10% on SWEBench Lite.
* **Mathematical Reasoning:** Achieved 75.7% on the MATH benchmark, comparable to state-of-the-art accuracy by GPT-4o.

**Data Collection:** The pre-training data consists of 60% source code, 10% math corpus, and 30% natural language corpus. The code-related tokens are sourced from GitHub and CommonCrawl, while the math-related tokens are sourced from CommonCrawl.

**Training Strategy:**

* **Next-Token Prediction and Fill-In-Middle (FIM):** Utilized for pre-training.
* **Reinforcement Learning (RL):** Employed Group Relative Policy Optimization (GRPO) algorithm to align the model's behavior with human preferences.

**Future Work:** Focus on improving instruction-following capabilities to handle real-world complex programming scenarios and enhance the productivity of the development process.

SUMMARY 2:

**Abstract**

DeepSeek-Coder-V2 is an open-source Mixture-of-Experts (MoE) code language model that matches the performance of GPT4-Turbo in code-specific tasks. It is further pre-trained from an intermediate checkpoint of DeepSeek-V2 with an additional 6 trillion tokens. This enhances its coding and mathematical reasoning capabilities while maintaining general language task performance. DeepSeek-Coder-V2 supports 338 programming languages and extends the context length from 16K to 128K tokens. It surpasses closed-source models like GPT4-Turbo, Claude 3 Opus, and Gemini 1.5 Pro in standard coding and math benchmarks.

**Introduction**

Open-source models like StarCoder, CodeLlama, and DeepSeek-Coder have advanced code intelligence but still lag behind closed-source models. DeepSeek-Coder-V2, built on DeepSeek-V2, is pre-trained with a large corpus of code, math, and natural language data. It significantly improves coding and mathematical reasoning capabilities while supporting more programming languages and longer context lengths.

**Data Collection**

The pre-training dataset for DeepSeek-Coder-V2 includes 60% source code, 10% math corpus, and 30% natural language corpus. Source code data is sourced from GitHub and CommonCrawl, covering 338 programming languages. Math data is collected from CommonCrawl using a pipeline similar to DeepSeekMath.

**Training Strategy**

DeepSeek-Coder-V2 employs Next-Token-Prediction and Fill-In-Middle (FIM) training objectives. The model architecture and hyperparameters align with DeepSeek-V2, utilizing the AdamW optimizer with a cosine decay learning rate schedule. The model is exposed to a total of 10.2 trillion tokens during pre-training.

**Long Context Extension**

DeepSeek-Coder-V2 extends context length to 128K tokens using Yarn, enhancing its capability for handling long and complex coding tasks.

**Evaluation and Metrics**

DeepSeek-Coder-V2 is evaluated on various benchmarks:

* **Coding:** Achieves top-tier results on HumanEval, MBPP, LiveCodeBench, and SWE-Bench.
* **Mathematics:** Shows strong performance on GSM8K, MATH, AIME 2024, and Math Odyssey benchmarks.
* **Natural Language:** Maintains strong general language performance comparable to DeepSeek-V2.

**Key Figures**

1. **Performance on Code and Math Benchmarks**
2. **Needle In A Haystack (NIAH) Tests**
3. **Performances of Different Methods**

**Conclusion**

DeepSeek-Coder-V2 significantly enhances code and mathematical reasoning capabilities, supporting more languages and longer contexts. It achieves comparable performance to state-of-the-art closed-source models in code-specific tasks. Future work will focus on improving instruction-following capabilities for better handling of real-world complex programming scenarios.

SUMMARY 3:

**Professional Summary of "DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence"**

**Introduction and Contributions** DeepSeek-Coder-V2 is an advanced open-source Mixture-of-Experts (MoE) code language model designed to rival the performance of closed-source models like GPT-4 Turbo in code intelligence tasks. Developed from an intermediate checkpoint of DeepSeek-V2 and further pre-trained with 6 trillion tokens, DeepSeek-Coder-V2 enhances coding and mathematical reasoning capabilities while supporting a broader range of programming languages and extended context lengths.

**Key Features and Enhancements**

1. **Pre-training and Data Collection**: The model's pre-training data comprises 60% source code, 10% math corpus, and 30% natural language corpus, sourced from GitHub and CommonCrawl. This comprehensive dataset covers 338 programming languages, a significant increase from the previous 86, and supports a context length of up to 128K tokens, allowing for more complex coding tasks.
2. **Performance Metrics**: DeepSeek-Coder-V2 achieves superior performance across various coding and math benchmarks, including HumanEval, MBPP+, and MATH, often surpassing closed-source counterparts. For instance, it scores 90.2% on HumanEval and 75.7% on MATH, showcasing its robust capabilities.
3. **Model Architecture and Training**: The model utilizes a combination of Next-Token-Prediction and Fill-In-Middle (FIM) training objectives. It incorporates advanced reinforcement learning techniques, such as Group Relative Policy Optimization (GRPO), to align its behavior with human preferences, ensuring accuracy and relevance in coding tasks.
4. **Evaluation and Comparisons**: In extensive evaluations, DeepSeek-Coder-V2 outperforms other open-source models and competes closely with leading closed-source models in code generation, code completion, and mathematical reasoning tasks. It demonstrates exceptional performance in handling various programming languages and coding challenges, making it a versatile tool for developers.

**Figures and Visual Data**

* **Figure 1**: Performance comparison of DeepSeek-Coder-V2 with other models on math and code benchmarks, indicating superior accuracy and robustness.
* **Figure 2**: Evaluation results on the "Needle In A Haystack" tests, highlighting the model's effectiveness across different context lengths up to 128K.
* **Figure 3**: Performance of reinforcement learning signals on LeetCode, demonstrating the advantages of using a reward model signal over raw compiler signal.