# Winning Amazon KDD Cup'24

Chris Deotte*
cdeotte@nvidia.com
NVIDIA
USA

Ivan Sorokin*
isorokin@nvidia.com
NVIDIA
Finland

Ahmet Erdem*
aerdem@nvidia.com
NVIDIA
Türkiye

Benedikt Schifferer*
bschifferer@nvidia.com
NVIDIA
Germany

Gilberto Titericz Jr*
gtitericz@nvidia.com
NVIDIA
Brazil

Simon Jegou*
sjegou@nvidia.com
NVIDIA
France

## Abstract

This paper describes the winning solution of all 5 tasks for the Amazon KDD Cup 2024 *Multi Task Online Shopping Challenge for LLMs*. The challenge was to build a useful assistant, answering questions in the domain of online shopping. The competition contained 57 diverse tasks, covering 5 different task types (e.g. multiple choice) and across 4 different tracks (e.g. multi-lingual).

Our solution is a single model per track. We fine-tune Qwen2-72B-Instruct on our own training dataset. As the competition released only 96 example questions, we developed our own training dataset by processing multiple public datasets or using Large Language Models for data augmentation and synthetic data generation. We apply *wise-ft* to account for distribution shifts and ensemble multiple LoRA adapters in one model. We employed *Logits Processors* to constrain the model output on relevant tokens for the tasks. *AWQ 4-bit Quantization* and *vLLM* are used during inference to predict the test dataset in the time constraints of 20 to 140 minutes depending on the track.

Our solution achieved the first place in each individual track and is the first place overall of Amazon's KDD Cup 2024.

## CCS Concepts

• **Computing methodologies → Natural language generation**; *Machine translation*; *Information extraction.*

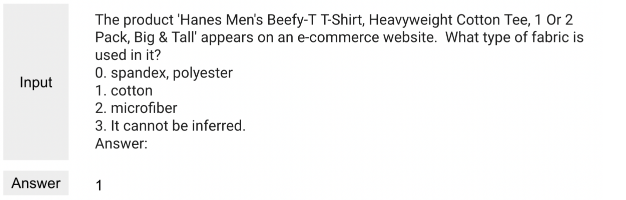
## Keywords

Large Language Models, LLM, Shopping Assistant, KDD Cup, Multi Task Learning, Multi-Lingual

**ACM Reference Format:**
Chris Deotte, Ivan Sorokin, Ahmet Erdem, Benedikt Schifferer, Gilberto Titericz Jr, and Simon Jegou. 2024. Winning Amazon KDD Cup'24. In *Proceedings of* . ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

The capabilities of Large Language Models (LLMs) have significantly improved in the last years and they have become popular due to their easiness to use. Users can interact with the systems in natural language. The LLMs excel on a variety of tasks, such as general reasoning, math questions, coding, etc. Many systems are getting updated by adding a LLMs to make them easier to use and/or providing more functionality. (Online) shopping is a large domain with billions of users and high economic output. The Amazon KDD Cup 2024 [2] is designed to evaluate LLMs to be a useful shopping assistant.



Input

The product 'Hanes Men's Beefy-T T-Shirt, Heavyweight Cotton Tee, 1 Or 2 Pack, Big & Tall' appears on an e-commerce website. What type of fabric is used in it?
0. spandex, polyester
1. cotton
2. microfiber
3. It cannot be inferred.
Answer:

Answer

1

**Figure 1: One example of the development dataset. It is a multiple choice question answering tasks for understanding shopping concepts.**

Amazon developed an evaluation dataset *ShopBench*, containing approx. 20,000 questions across 57 different tasks covering 5 task types (e.g. retrieval), to test LLMs capabilites in the online shopping domain (see an example in Figure 1). The competition had 5 different tracks, which evaluates different aspects such as shopping knowledge understanding or user behavior alignment. The 5th track was the overall track containing all 20,000 questions. The competition was organized as a code competition in which participants have no access to the ShopBench dataset and instead they have to submit their model.

Our team from NVIDIA won all 5 tracks (see Table 3). This paper describes our final solution and an ablation study on our experiments. Our solution is based on a single model per track, which shares following techniques:

(1) **Developing a Training Dataset**: As the hosts did not provide a training dataset, we processed many multiple public datasets and enriched it by prompting Large Language Models to generate a training dataset

*All authors contributed equally to this research.

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

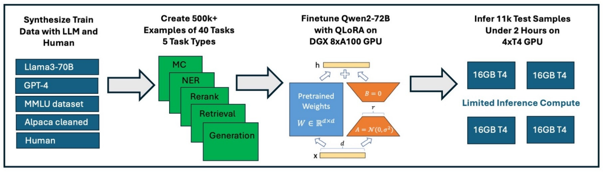

Synthesize Train Data with LLM and Human
Llama3-70B
GPT-4
MMLU dataset
Alpaca cleaned
Human
Create 500k+ Examples of 40 Tasks 5 Task Types
MC
NER
Rerank
Retrieval
Generation
Finetune Qwen2-72B with QLoRA on DGX 8xA100 GPU
h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x
Infer 11k Test Samples Under 2 Hours on 4xT4 GPU
16GB T4
16GB T4
Limited Inference Compute
16GB T4
16GB T4

**Figure 2: High-Level Oerview of our pipeline for the KDD Cup 2024**

**Table 1: Runtime limit in minutes per track in phase 2**

| Phase | Track 1 | Track 2 | Track 3 | Track 4 | Track 5 |
|---|---|---|---|---|---|
| Phase 2 | 70 | 20 | 30 | 20 | 140 |

(2) **Fine-Tuning a Base Model**: We fine-tuned Qwen2-72B-Instruct [18]
(3) **Additional Optimization**: We applied multiple techniques to maximize our solution by prompt engineering, ensembling multiple adapters, addressing distribution shift with *wise-ft* and constraining the model with a *Logits Processors*
(4) **Optimizing Inference**: As the competition had compute and time constraints, we optimized our inference code with 4-bit quantization and vLLM.

We describe our methods in details in the following sections 3 and 4. Section 5 will compare our solutions and share more experiments we ran during the competition as a small ablation study.

## 2 Amazon KDDCup 2024: Multi Task Online Shopping Challenge for LLMs

Amazon hosted the KDD Cup 2024 for *Multi Task Online Shopping Challenge for LLMs* [2]. They developed a test dataset, called *ShopBench*, containing 20,000 questions across 57 tasks. A development dataset of 96 question of only 18 different tasks were shared with the participants for the competition. The KDD Cup 2024 is designed as a code competition. Participants submit model weights with code which will be evaluated on infrastructure provided by Amazon. Participants have no access to the test dataset and can build solutions based on the 96 development questions. They receive only the scores on the full *ShopBench* dataset via the leaderboard. A submission is evaluated on 4x NVIDIA T4 GPUs with each 16 GB GPU memory within a runtime limit (see Table 1).

Participants have to address following challenges:

(1) **No training dataset**: Participants have access to only 96 examples.
(2) **Hidden tasks**: The development dataset contains only 18 out of 57 tasks. Therefore, the solution has to generalize to the unknown tasks.
(3) **Time and compute constrains**: Solutions have to run within a runtime limit on 4x NVIDIA T4 GPUs with each 16 GB memory (see Table 1)

The competition contains 5 tracks:

- **Shopping Concept Understanding**: Understanding shopping concepts (e.g. brands, product lines, attributes, etc.)
- **Shopping Knowledge Reasoning**: Reasoning ability about products or product attributes (e.g. total amount in a product pack, are two products compliments or substitutes, etc.)
- **User Behavior Alignment**: Understanding user behavior in online shopping (e.g. implicit information by user click stream
- **Multi-lingual Abilities**: Shopping concept understanding and user behavior alignment across different languages
- **Overall**: A final track which combines all 4 tracks

The evaluation dataset is based on 5 task types:

- **Multiple Choice**: Only one correct answer. Evaluation metric is accuracy.
- **Ranking**: Input contains multiple candidates and the model should provide an ordered list. Evaluation metric is nDCG.
- **Named Entity Recognition (NER)**: Extract pieces of text given an entity type. Evaluation metric is Micro-F1.
- **Retrieval**: Select candidates from a list which satisfy the requirements. Evaluation metric is Hit@3
- **Generation**: There are a diverse set of generation tasks depending on the task (e.g. translation). Evaluation metrics are ROUGE-L, BLEU or cosine similarity of sentence embedding.

A track can contain one or multiple task types. The final score of a track is calculated by averaging across all questions because each evaluation metric is between 0-1. The overall challenge score is determined by the sum of position per track.

For every question, the requirement is to generate text, which is parsed by Amazon's evaluation script. The solution has to follow the prompt instructions (e.g. *return 3 candidates IDs separated by a comma*). If the evaluation script is not able to parse the generated text, then the score will be 0 for this question.

The competition was organized in 2 phases. The organizer shared that Phase 2 contains harder samples and tasks than Phase 1. They increased the compute resources from 2x NVIDIA T4s to 4x NVIDIA T4s for phase 2.

## 3 Training Dataset

Amazon shared multiple eCommerce datasets with participants, which are related to the ShopBench dataset, but do not have the same structure. We created our training dataset by processing multiple datasets to have a similar structure as the 18 tasks from ShopBench development dataset. In addition, we developed new tasks. Finally, we augmented the dataset by prompting LLaMa3-70B-Instruct [3] and GPT-4 [11] for more diversity or infer missing information (e.g. product type, category). A detailed overview can be found in Appendix A.

### 3.1 Real Datasets

We utilized multiple data sources, including non e-commerce datasets such as MMLU and Alpaca-Cleaned. Samples from these datasets were transformed into the instruction prompts.

**Amazon-M2** [9] - A multi-lingual Amazon session dataset with rich meta-data used for KDD Cup 2023.

**Amazon Reviews 2023** [8] - A large scale Amazon Review Dataset with rich features and over 500M reviews across 33 categories.

**NingLab/ECInstruct** [14] - instruction dataset covers 116,528 samples from 10 real and widely performed e-commerce tasks of 4 categories.

**ESCI-data** [15] - Shopping Queries dataset provides a list of up to 40 potentially relevant results, together with ESCI relevance judgements (Exact, Substitute, Complement, Irrelevant) indicating the relevance of the product to the query.

**MMLU** [7] [6] - massive multitask test consisting of 16k multiple-choice questions and auxiliary 100k multiple-choice training questions from ARC, MC_TEST, OBQA, RACE, etc.

**Alpaca-Cleaned** [16] - a cleaned version of the original Alpaca Dataset released by Stanford.

## 3.2 Synthetic Datasets

To further improve diversity of dataset we utilized the synthetic data generation (SDG) pipelines. In general, we used three different methods.

We prompt LLM to construct the tasks specific prompts from the seed data. For example, we rephrase the original tasks from *NingLab ECInstruct* dataset. These tasks include various information about the product (title, description, attributes) and we combine all of them into one prompt. (see Table 7 Dataset No 11-19).

Before constructing a task specific prompt, we extract the correct labels from the seed data using LLM. For example, we extract the product type, categories or attributes first and then construct the question. (see Table 7 Dataset No 1,20).

We used GPT-4 to generate the instructions with different wordings, and then used it to construct MC tasks from *ESCI-data* dataset. The correct answer was randomly selected from the E entries, the remaining options were selected from the entries with S/C/I labels (see Table 7 Dataset No 21-26).

# 4 Model

## 4.1 Prompt Template

We explored both zero shot LLM models and fine-tuned LLM models. Our final winning solution achieving our best model accuracy is fine-tuned. See Table 4 for a comparison.

When using zero shot with an instruction tuned LLM, we found it helpful to use both the system role and user role when formatting prompts. Designing better prompts improved the zero shot model's performance.

When fine-tuning, we found that the prompt was not as important because the model is fine-tuned to exhibit a certain behavior given whatever prompt we choose to train with.

One technique of our fine-tuned models used is to include an instruction to the model identifying which of the 5 task types the model is solving. Then during inference, we used a heuristic rule classifier which determined question task type and included this is the system role's instruction prompt. Specifically, we used the following template.

**Listing 1: System prompt template with task type**

```
system_prompt = "You␣are␣a␣helpful␣online␣shopping␣
    ↪ assistant.␣Your␣task␣is␣{task_type}."
```

**Table 2: Model hyperparameters**

| Hyperparameter | Value |
|---|---:|
| Optimizer | AdamW |
| LR Scheduler | cosine |
| Learning Rate (LR) | 0.0002 |
| Weight Decay | 0.01 |
| Warm Up Steps | 10 |
| Micro Batch Size | 1 |
| Gradient Accumulation Steps | 4 |
| QLoRA R | 64 |
| QLoRA Alpha | 32 |
| QLoRA Dropout | 0.05 |
| QLoRA Linear | TRUE |
| Quantization | 4-bit |

## 4.2 Fine-Tuning Qwen2

We fine-tuned *Qwen/Qwen2-72B-Instruct* [18] on our developed training dataset using 8x NVIDIA A100 with each 80GB GPU memory. Training on 500k examples takes around 24 hours. We used the library axolotl [1] and bitsandbytes [2] with QLoRA [5] with 4-bit quantization and bfloat16 [4]. The library applies *Multipack (Sample Packing)* [3], concatenating multiple sequences into one batch to increase training throughput. Table 2 provides an overview of the hyperparameters.

We train the LLM in a supervised fine-tuning strategy. The loss is calculated only on the answer tokens (see Figure 1). A common technique in large language model training is to apply *Reinforcement Learning from Human Feedback (RLHF)* [12]. Our hypothesis is that supervised fine-tuning is sufficient for the competition. Many answers are a single number or a list of numbers, which have an exact solutions and does not require human preferences between multiple possible answers.

## 4.3 Ensemble Adapters

Our five track solutions are created from 4 fine-tuned LoRA adapters. We call them v7, v8, v7b, and v9b. First for tracks 1,3,5 we merged v8 to base model Qwen2-72B with 56% weight explained in Section 4.4. For tracks 2,4 we merged v7 to base with 100%. Version 7 adapter was trained with 417k samples whereas v8 was trained with 462k. See Table 7 for details about train data.

Next we trained two more LoRA adapters named v7b and v9b using two different new subsets of 152k and 40k samples respectively. Our final solutions with ensemble weights and leaderboard scores to tracks 1-5 are shown in Table 6.

[1] https://github.com/axolotl-ai-cloud/axolotl
[2] https://huggingface.co/docs/transformers/main/en/quantization/bitsandbytes
[3] https://github.com/axolotl-ai-cloud/axolotl/blob/main/docs/multipack.qmd

### 4.4 Wise-ft

To take into account the distribution shift between the evaluation data from the ShopBench dataset and the collected training data (see Section 3), we used wise-ft [17].

Wise-ft interpolates between the weights $W_{base}$ of a base model and the weights $W_{ft}$ of a fine-tuned model using the following formula:

$$W_{wise} = (1 - \alpha) * W_{base} + \alpha * W_{ft}$$

where $\alpha \in [0, 1]$. This approach effectively balances the trade-off between the zero-shot capabilities of the base model ($\alpha = 0$) and the task-specific performance of the fine-tuned model ($\alpha = 1$).

For LoRA, as $W_{ft} = W_{base} + W_A \cdot W_B$, we can rewrite the formula as:

$$W_{wise} = W_{base} + \alpha * W_A \cdot W_B$$

We implemented wise-ft by rescaling the ensembled adapter weights $W_A$ and $W_B$ by a factor $\sqrt{\alpha}$, so that $(\sqrt{\alpha} * W_A) \cdot (\sqrt{\alpha} * W_B) = \alpha * W_A \cdot W_B$. For each track, we optimized $\alpha$ based on leaderboard results and obtained significant improvements for Track 1, Track 3 and Track 5 (see Figure 3).

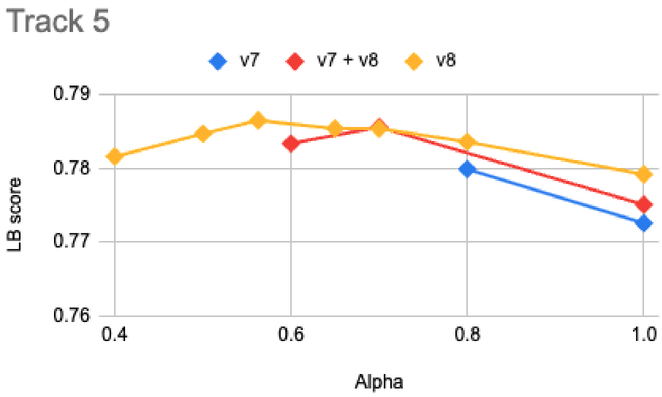


**Figure 3: Accuracy gain on Track 5 using various $\alpha$ for wise-ft for 3 different versions of LoRA adapters (v7, v8 and the ensemble v7 + v8)**

### 4.5 Logits Processors

We employed a variety of logits processors to generate outputs in specific formats. For multiple choice, ranking, and retrieval questions, we constrained our models to produce only digits and commas. For NER tasks, we enhanced the logits of the prompt tokens, encouraging the model to cite directly from the prompt. These logits processors were particularly useful in Phase 1 when we utilized less powerful models. These constrains also were useful in case when the training dataset includes only few task types. For example, you can finetune a model for MC tasks only and successfully apply it for Retrival or Reranking tasks. However, their importance diminished in Phase 2 as we transitioned to larger models that more effectively followed instructions.

**Table 3: Leaderboard Results: Final is the sum of the ranks per track (lower is better). T1 - T5 are the scores per track (higher is better)**

| Team | Final | T1 | T2 | T3 | T4 | T5 |
|---|---|---|---|---|---|---|
| Team_NVIDIA | 5 | 0.833 | 0.791 | 0.746 | 0.761 | 0.788 |
| AML_LabCityU | 13 | 0.825 | 0.781 | 0.728 | 0.715 | 0.782 |
| shimmering_as_... | 18 | 0.824 | 0.747 | 0.713 | 0.735 | 0.763 |
| CM_RLLM | 29 | 0.823 | 0.728 | 0.722 | 0.690 | 0.773 |
| ZJU_AI4H | 33 | 0.791 | 0.784 | 0.694 | 0.706 | 0.746 |
| BMI_DLUT | | | | 0.733 | | |

### 4.6 Quantization / vLLM

KDD Cup 2024 was a code competition meaning that we must submit code plus model weights to be run on the host's pre-defined compute resources. Each participant could submit (to each track) a GitLab repository of maximize size 100GB to be executed on 4x NVIDIA T4 GPU each with 16 GB GPU memory within a time constraint.

The Qwen2-72B model is about 150GB at fp16. Therefore in order to fit this into disk and memory size constraints, we used 4bit quantization which reduced its size to 40GB.

Quantization plus using the library vLLM [10] accelerated our inference which allowed our model to answer all the questions within the time limit. Tracks 1-5 had 6102, 1896, 2373, 1349, and 11720 questions to be answered in 70, 20, 30, 20, 140 minutes respectively.

We improved AWQ quantization accuracy by calibrating with the 96 development questions. We compared AWQ versus GPTQ quantization and found both to be about equal in speed and accuracy.

AWQ quantization for Qwen2-72B takes about 1.5 hours on 1xA100 GPU to process. In order for the AWQ quantized Qwen2-72B to work with vLLM, we needed to pad the unquantized model with zeros to change the shape of the weights before quantization.

## 5 Results

Our quantized, fine-tuned Qwen2-72B model achieves the highest score on each individual track (T1 - T4) and overall track T5 with a significant lead of 0.007 to 0.026 to the 2nd place (Table 3). As we placed 1st in each individual track, our final score is 5, the sum of our positions, which is the highest possible score.

Each submission for the individual track is based on the key concepts of fine-tuning a Qwen2-72B model on our developed training dataset and optionally, ensemble multiple versions and/or apply *wise-tf.* The submission might differ slightly in the fine-tuning time, exact amount of training dataset and ensemble combination.

We provide an ablation study in Table 4, 5 and 6. Some values are missing in the tables due to failed submissions and the successful submissions were sufficient to decide the next experiments. First, table 4 compares different base models without being fine-tuned. We observe that Qwen2-72B has the highest score except of for Track 2, followed by LLaMa3-70B is 2nd place except of Track 4. The performance of the models are equivalent to public LLMs benchmarks.

Next, we fine-tuned Smaug-72B and Qwen2-72B on our training dataset. We compare the zero-shot version (SZ) with the fine-tuned

**Table 4: Comparison of different base model without fine-tuning on Track 1 - Track 5. We report only scores we could run during the competition.**

| Model | T 1 | T 2 | T 3 | T 4 | T 5 |
|---|---|---|---|---|---|
| Bagel-34B-v0.5 [1] | 0.7007 | 0.6609 | 0.6339 | 0.5871 | 0.6834 |
| LLaMa3-70B [3] | 0.7806 | 0.6532 | 0.6658 | 0.6237 | 0.7183 |
| Smaug-72B [13] | 0.7178 |  | 0.6564 | 0.6484 | 0.6975 |
| Qwen2-72B [18] | 0.7982 | 0.6407 | 0.7193 | 0.6918 | 0.7486 |

**Table 5: Comparison foundation model as zero-shot (SZ) with fine-tuned version (FT) on Track 1 - Track 5. We report only scores we could run during the competition.**

| Model | T 1 | T 2 | T 3 | T 4 | T 5 |
|---|---|---|---|---|---|
| Smaug 72B ZS | 0.7178 |  | 0.6564 | 0.6484 | 0.6975 |
| Smaug 72B FT | 0.7800 | 0.7389 |  |  |  |
| Qwen2 72B ZS | 0.7982 | 0.6407 | 0.7193 | 0.6918 | 0.7486 |
| Qwen2 72B FT | 0.8334 | 0.7909 | 0.7461 | 0.7609 | 0.7883 |

**Table 6: Configuration and results of ensembling multiple LoRA configurations. B is the base model weights. We add the weights M1 from the LoRA Adapter multiplied with the weight W1 (equivalent for the ensemble B+W1xM1+W2xM2). LB demonstrates the scores on the leaderboard.**

| Model | T 1 | T 2 | T 3 | T 4 | T 5 |
|---|---|---|---|---|---|
| LoRA 1 name (M1) | v8 | v7 | v8 | v7 | v8 |
| LoRA 1 weight (W1) | 0.56 | 1.0 | 0.56 | 1.0 | 0.56 |
| LB B+W1xM1 | 0.831 | 0.787 | 0.742 | 0.758 | 0.787 |
| LoRA 2 name (M2) | v9b | v7b | v9b | v7b | v9b |
| LoRA 2 weight (W2) | 0.75 | 0.5 | 0.25 | 0.5 | 0.25 |
| **LB B+W1xM1+W2xM2** | **0.833** | **0.791** | **0.746** | **0.761** | **0.788** |

version (FT) as seen in Table 5. Qwen2-72B SZ would scored 4th place on Track 5, demonstrating that base model provide great capabilities without fine-tuning. We achieve significant gains of 0.0035 to 0.15 by additional fine-tuning.

Table 6 summarize the effect of ensembling multiple models. The first solution is to submit the base model merged with the first LoRA adapter, scaled by weight W1. If we ensemble two adapters (*LB B+W1xM1+W2xM2*), we observe that the LB score improves between 0.001 to 0.004.

## 6 Conclusion

The KDD Cup 2024 was a great competition with a diverse set of tasks to evaluate Large Language Models capabilities in the domain of online shopping. The code competition design ensured a fair comparison of solutions. Our team solution is a single models with multiple optimization methods, which scored the 1st place on each track. It was essential to fine-tune a base model with an additional training dataset. The lack of an official training dataset was compensated by processing multiple public datasets and prompting Large Language Models. We ensembled multiple LoRA adapater, applied *wise-ft* for distribution shift and constrained the model output with a *Logits Processors*. We optimized inference with 4-bit quantization and vLLM to run a 72 billion parameters model on 4x NVIDIA T4 with each 16 GB GPU memory in the time constrain. In addition, we share multiple experiments as an ablation study.

## References

[1] 2024. A bagel, with everything (except DPO). (2024). https://huggingface.co/jondurbin/bagel-34b-v0.5

[2] 2024. A Multi-task Online Shopping Challenge for Large Language Models. (2024). https://amazon-kddcup24.github.io/

[3] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[4] Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. 2019. Bfloat16 Processing for Neural Networks. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*. 88–91. https://doi.org/10.1109/ARITH.2019.00022

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs.LG] https://arxiv.org/abs/2305.14314

[6] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning AI With Shared Human Values. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[7] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[8] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[9] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. 2023. Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation. arXiv:2307.09688 [cs.IR] https://arxiv.org/abs/2307.09688

[10] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

[11] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang,

Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[12] Long Ouyang, Jeff 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL] https://arxiv.org/abs/2203.02155

[13] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing Failure Modes of Preference Optimisation with DPO-Positive. *arXiv preprint arXiv:2402.13228* (2024).

[14] Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. 2024. eCeLLM: Generalizing Large Language Models for E-commerce from Large-scale, High-quality Instruction Data. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=LWRI4uPG2X

[15] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. (2022). arXiv:2206.06588

[16] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[17] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. Robust fine-tuning of zero-shot models. arXiv:arXiv:2109.01903

[18] 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. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

Table 7: Overview of the different training datasets we developed for fine-tuning Qwen2-72B. The task column describes if our processed dataset is close to an existing dataset task. The column LLM indicates if a LLM was used for generating the dataset.

| No | Source Dataset | Task | Task Type | Adapter | Size | LLM | Additional Explanation |
|---|---|---|---|---|---|---|---|
| 1 | Amazon-M2 | KDD Cup2024 Task 2 | multiple-choice | v7, v8 | 2350 | Yes | Select product categories given product attributes |
| 2 | Amazon Reviews 2023 | KDD Cup2024 Task 3 | retrieval | v7, v7b, v8 | 7373 | Yes | Given a product type and sentiment, select 3 most likely snippet a customer would write about the product |
| 3 | Amazon Reviews 2023 | KDD Cup2024 Task 7 | retrieval | v7, v7b, v8 | 3608 | Yes | Given a product type and a review, select 3 aspects covered by the review |
| 4 | Amazon Reviews 2023 | KDD Cup2024 Task 10 | multiple-choice | v7, v7b, v8 | 10000 | Yes | Given a product type, which of the following categories complement the product type best? |
| 5 | ESCI-data | KDD Cup2024 Task 12 | ranking | v7, v8 | 16728 | No | |
| 6 | Amazon Reviews 2023 | KDD Cup2024 Task 14 | ranking | v7, v7b, v8 | 5815 | No | Given a product title a customer will buy, which other product titles will he like |
| 7 | Amazon Reviews 2023 | KDD Cup2024 Task 15 | multiple-choice | v7, v7b, v8 | 10000 | No | Given a product review, estimate the rating of the review |
| 8 | ESCI-data | KDD Cup2024 Task 16 | multiple-choice | v7, v8 | 10000 | No | |
| 9 | Amazon-M2 | KDD Cup2024 Task 17 | generation | v7, v8 | 10000 | No | |
| 10 | Amazon-M2 | KDD Cup2024 Task 18 | multiple-choice | v7, v8 | 10000 | No | |
| 11 | NingLab/ECInstruct | Attribute Value Extraction | named entity recognition | v7, v8 | 19622 | Yes | |
| 12 | NingLab/ECInstruct | Multiclass Product Classification | multiple-choice | v7, v8 | 10000 | Yes | |
| 13 | NingLab/ECInstruct | Product Relation Prediction | multiple-choice | v7, v8 | 10000 | Yes | |
| 14 | NingLab/ECInstruct | Query Product Rank | retrieval | v7, v8 | 10000 | Yes | |
| 15 | NingLab/ECInstruct | Sequential Recommendation | multiple-choice | v7, v8 | 10000 | Yes | |
| 16 | NingLab/ECInstruct | Answerability Prediction | multiple-choice | v7, v8 | 10000 | No | |
| 17 | NingLab/ECInstruct | Product Matching | multiple-choice | v7, v8 | 4044 | No | |
| 18 | NingLab/ECInstruct | Product Substitute Identification | multiple-choice | v7, v8 | 10000 | No | |
| 19 | NingLab/ECInstruct | Sentiment Analysis | multiple-choice | v7, v8 | 10000 | No | |
| 20 | Amazon-M2 | New Idea | generation | v7, v8 | 10000 | Yes | Explain product type given title, description, and product type |
| 21 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the user query that matches the product description |
| 22 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the user query that matches the product features |
| 23 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the user query that matches the product title |
| 24 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the title for the product description |
| 25 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the title for the product features |
| 26 | ESCI-data | New Idea | multiple-choice | v7, v8 | 10000 | Yes | Select the product title for the user query |
| 27 | ESCI-data | New Idea | retrieval | v8 | 10000 | No | Pick 3 bullet points to match product |
| 28 | NingLab/ECInstruct | New Idea | ranking | v8 | 5000 | No | Rank product reviews - positive to negative |
| 29 | ESCI-data | New Idea | multiple-choice | v8 | 5435 | No | Pick product to match query |
| 30 | ESCI-data | New Idea | ranking | v8 | 5000 | No | Task12 backwards. Given product, rank queries |
| 31 | KDD Cup 2023 | New Idea | retrieval | v8 | 10000 | No | Given purchase pick previous clicks (similar to task 14) |
| 32 | ESCI-data | New Idea | multiple-choice | v8 | 10000 | No | Given title pick brand |
| 33 | ESCI-data | New Idea | multiple-choice | v9b | 10000 | No | Given query product pair, what is relationship? E S C I |
| 34 | ESCI-data | New Idea | ranking | v9b | 10000 | No | Given list of query product pairs, rank which are most related to least related |
| 35 | ESCI-data | New Idea | retrieval | v9b | 10000 | No | Given query, select products which are exact match not substitute, complement, or irrelevent |
| 36 | Amazon Reviews 2023 | New Idea | ranking | v9b | 10000 | No | Given a product title and multiple reviews, rank the reviews based on the helpfulness |
| 37 | Alpaca Cleaned | No Changes | generation | v7, v8 | 51760 | No | |
| 38 | MMLU | No Changes | multiple-choice | v7, v7b, v8 | 115700 | No | |
| | Total | Total | Total | Total | 502435 | | |

## A Details on Training Dataset

In Table 7, we provide an overview of the different datasets we generated. We share which dataset was used as a source input. We describe which task the resulting dataset is most similar to (column *Task*), the size and if a LLM was used. Finally, we provide additional explanation for our own ideas.