Comparative Study of LoRA and Full Fine-Tuning in Large Language Models*

E.V. Surikova, E.A. Sabidaeva

National Research University Higher School of Economics

In recent years, large language model (LLM) fine-tuning has become a key area of research in the field of natural language processing (NLP). However, taking into account constraints and high costs of training, researchers are increasingly turning to PEFT [1] methods such as LoRA, which demonstrate good results. In this study, we conducted an experimental comparison of full tuning and LoRA-based tuning methods using the QWEN2-7b [2] model as the base one. We compared performance across three distinct tasks within the domain of economics and information technology: classification task, short question answering and long question answering. The results substantiate that LoRA facilitates significant reductions in computational and storage resources, thereby simplifying model deployment processes. However, full fine-tuning provides higher accuracy for all types of ML tasks considered. Additionally, we investigated the hypothesis that integrating various task types within the same domain could enhance generation quality. However, our findings indicate that combining training data from classification tasks and QA resulted in a decline in performance metrics, aligning them with those achieved by the LoRA approach.

Keywords: fine-tuning, LoRA, full-parameter, LLM.

1. Introduction

Low-rank adaptation (LoRA) [3] is a widely used method for fine-tuning large language models (LLMs). This approach utilizes low-rank matrices, enabling the preservation of a significant portion of valuable information while optimizing memory usage. In this study, we aim to assess the performance of LoRA in comparison to full fine-tuning of LLMs across various tasks, such as classification and text generation in the domains of economics and digital technologies.

The LoRA method is based on freezing the pre-trained weighting matrix $W_{pretrained} \in R^{d \times k}$, and training solely on the low-rank matrix Δ, as follows:

$$W_{finetuned} = W_{pretrained} + \Delta, \quad (1)$$

$$\Delta = AB, \ \ A \in R^{d \times r}, \ \ B \in R^{r \times k}, \quad (2)$$

where $A_0 \sim N(0,1), B_0 = 0$.

$W_{pretrained}$ needs to be adapted and the rank should be r << d, k, so that instead of d × k parameters, only d × r + r × k parameters are trained, which reduces the memory and FLOPS needed to compute the gradient. For example, applying r = 16 LoRA to a 7B weight matrix with d = k = 4096 requires < 1% of the original number of parameters.

The article is structured as follows. The first section provides an overview of the data used for both classification and QA tasks. Next, we delve into the experimental phase, detailing the hyperparameter settings and the metrics obtained from the training process. In the related works section, we examine existing datasets within the relevant domains and analyze papers that compare full fine-tuning and LoRa methods. Finally, we summarize the key findings and highlight the main results of the study.

2. Data overview

2.1 Dataset for classification task

The dataset was generated using gpt4o-mini [4]. It was given a prompt which instructed the model to attribute each fragment with a class:

* This research was supported in part through computational resources of HPC facilities at HSE University.

Objective:

You are tasked with determining whether a given text contains a specific theme that is classified as "Class A". Your output should be either 1 (indicating the presence of the theme) or 0 (indicating its absence). Return 1 if the theme is present in the text, or 0 if it is not.

Example:

Input Text: "The advancements in renewable energy technology are crucial for combating climate change and fostering sustainable development."

Class: "EnergyTech" ,

Output: 1.

Input Text: "The latest trends in fashion highlight the importance of personal style and self-expression.",

Class: "AI",

Output: 0.

The fragments inputted to the model represent excerpts from media articles published over the past three years in the domain of economics and information technology. As a result, we formed the dataset in which each text fragment is assigned one of the nine classes: "Science, Technology and Innovation", "Telecom", "AI", "FinTech", "Digital Transformation", "EdTech", "Business", "EnergyTech", and "MedTech". The training part of the dataset comprises 9000 rows, while the test dataset contains 1800 rows. The total size of the dataset in tokens is 2M. A sample from the dataset is presented in Table 1.

Table 1. An example of data input from the dataset for the classification task

Texts	**Class**
Financial Services 2030 About 43% of the surveyed experts assessed ESG at the moment as a "hype", especially promoted by marketing departments of many companies, and found it difficult to estimate the preservation of ESG agenda in the next 10 years.	FinTech

2.2 Dataset for QA task

For the QA task, we created another dataset within the same domain using textual data from StackExchange [5]. It was also enriched with the data generated with gpt4o-mini. To achieve this, we created a prompt that requested the model to generate both short and long answers based on the information from the original data:

SFQA prompt

Task:

Based on the provided Stack Exchange answer, create a short answer that captures the main point or solution presented in the answer. Your answer should be clear and to the point, ideally no longer than 2-3 sentences.

Example:

Input:

Question: "How can you improve your coding skills?"

Stack Exchange Answer: "To improve your coding skills, practice regularly by working on small projects. Collaborate with others and seek feedback on your code. Additionally, consider contributing to open-source projects to gain real-world experience."

Output:

Short answer: "To enhance your coding skills, engage in regular practice through small projects, collaborate with peers for feedback, and contribute to open-source initiatives for practical experience."

LFQA prompt

Task:

Using the provided Stack Exchange answer, create a detailed answer that expands on the ideas presented. Include examples, explanations, and any relevant context to provide a comprehensive understanding of the topic.

Example:

Input:

Question: "What is Git for?"

Stack Exchange Answer: "Using version control systems like Git allows developers to track changes in their code, collaborate with others, and manage different versions of their projects effectively."

Output:

Long answer: "Version control systems, particularly Git, are essential tools for modern software development. They enable developers to keep a detailed history of changes made to their codebase, allowing for easy tracking and rollback if necessary. For instance, when working in teams, Git facilitates collaboration by allowing multiple developers to work on different features simultaneously without overwriting each other's work. Furthermore, Git's branching capabilities allow developers to experiment with new ideas in isolated environments before merging them into the main project. Overall, mastering version control is crucial for maintaining code integrity and enhancing team productivity."

As a result, the dataset consisting of context, query, and examples of both short and long answers was generated. The training set includes 7340 rows, while the test set contains 1200 rows. The total size of the dataset in tokens for Short-Form Question Answering (SFQA) is 6.8M, and for Long-Form Question Answering (LFQA), it is 8.2M. A sample from the dataset is presented in Table 2.

Table 2. An example of data input from the dataset for the QA task

Context	Query	Short answer	Long answer
The Buenos Aires AI Plan is an overarching strategy for the generation of a positive impact through the use and development of artificial intelligence. The plan breaks down potential risks and opportunities, specifies main objectives, provides a structure for its implementation and describes the main challenges to achieve this…	What are the main objectives of the AI strategy being implemented in Buenos Aires?	The main goal is to use AI to develop the city for the benefit of the citizens and ensure the sustainability of the city.	The Future City AI Strategy, launched in August 2021, has three main objectives: 1. to use artificial intelligence to develop the city; …

These datasets were combined in order to investigate the impact of different types of tasks on each other, as well as the amount of data. To compare full parameter fine-tuning and LoRA, the dataset from Stack Exchange was used.

The number of tokens of system prompt length for each type of task is given in Table 3.

Table 3. The number of tokens in the instruction, depending on the type of task

Task	System prompt length (tokens)
Classification	146
SFQA	65
LFQA	78

Figure 1 shows the average answer length for each type of task.

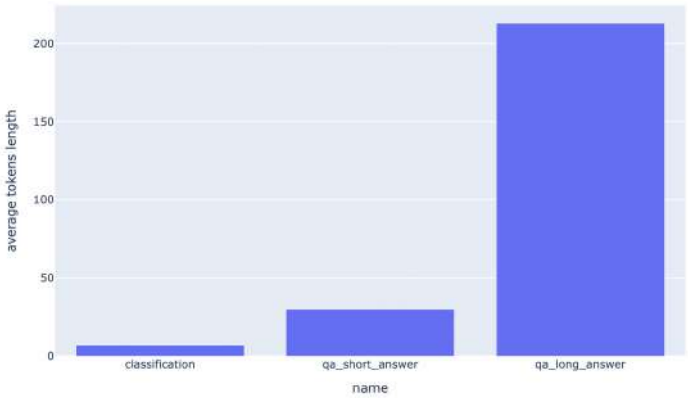

Figure 1. Golden average tokens

3. Experimental part

Qwen2-7B [6] was chosen as the base model for fine-tuning. Qwen2 language models, including pre-trained and instruction-tuned models in 5 sizes, including 0.5B, 1.5B, 7B, 57B-A14B, and 72B. Qwen2-7B supports context length of up to 131,072 tokens. Hyperparameter settings for full fine-tuning and LoRA are presented in Tables 4–5.

Table 4. Hyperparameter settings: full parameter fine-tuning

Hyperparameter	Value
sequence_len	1024
gradient_accumulation_steps	8
micro_batch_size	1
num_epochs	2
bf16	true
optimizer	paged_adamw_8bit
lr_scheduler	cosine
learning_rate	2e-5
warmup_steps	10

Table 5. Hyperparameter settings: LoRA fine-tuning

Hyperparameter	Value
adapter	lora
lora_r	32
lora_alpha	16
lora_dropout	0.05
Precision	fp16
gradient_accumulation_steps	4
micro_batch_size	2
num_epochs	2
learning_rate	3e-4
optimizer	paged_adamw_8bit

Table 6 demonstrates the structure of the LoRA matrix. The use of LoRA with a rank of 8 significantly reduces the number of trainable parameters compared to fine-tuning the entire original matrix. Instead of updating all $152\,064 \times 3\,584 = 543\,360\,256$ parameters, only $(152\,064 \times 8) + (8 \times 3\,584) = 1\,216\,512 + 28\,672 = 1\,245\,184$ parameters are updated. This reduction makes training faster and less resource-intensive.

Table 6. Structure of the LoRA matrix for the classification task

Matrix parameters	Sizes
Original matrix	152064 x 3584
LoRA rank	8
Additional matrix	4 x 3584
LoRA matrix A	152064 x 8
LoRA matrix B	8 x 3584

The total number of trained parameters is 1 245 184.

The experiment was conducted on 2 NVIDIA A100 80 GB GPUs (HPC Cluster "cHARISMa").

3.1 Classification task metrics

Table 7 presents performance metrics for various classes in a classification task. The metrics include precision, recall, F1-score, and support for each class, along with overall accuracy and average scores. Qwen2:7b-instruct was chosen as the baseline model. Its metrics are given in Table 8.

It can be concluded that the models fine-tuned with LoRA and baseline model demonstrate lower quality with the class "Digital Transformation" and "Science, Technology and Innovation".

Table 7. Metrics for fine-tuning LoRA Qwen2: 7b-instruct-clf-lora

class	precision	recall	f1-score	support
AI	0.62	0.73	0.67	200
Business	0.84	0.56	0.67	200
Digital Transformation	0.45	0.56	0.50	200
EdTech	0.93	0.9	0.91	200
EnergyTech	0.83	0.76	0.79	200
FinTech	0.76	0.8	0.78	200
MedTech	0.83	0.91	0.87	200
Science, Technology and Innovation	0.44	0.52	0.48	200
Telecom	0.8	0.58	0.68	200
accuracy			0.70	1800
macro avg	0.72	0.70	0.70	1800
weighted avg	0.72	0.70	0.70	1800

Table 8. Metrics for Qwen2: 7b-instruct

class	precision	recall	f1-score	support
AI	0.46	0.77	0.57	200
Business	0.24	0.2	0.22	200
Digital Transformation	0.28	0.2	0.23	200
EdTech	0.75	0.9	0.82	200
EnergyTech	0.74	0.8	0.76	200
FinTech	0.78	0.68	0.72	200
MedTech	0.83	0.8	0.82	200
OTHER	0	0	0	0
Science, Technology and Innovation	0.26	0.17	0.21	200
Telecom	0.57	0.28	0.37	200
accuracy			0.53	1800
macro avg	0.49	0.48	0.47	1800
weighted avg	0.54	0.53	0.52	1800

Table 9. Full parameter fine-tuning Qwen2: 7b-instruct-clf

class	precision	recall	f1-score	support
AI	0.72	0.9	0.8	200
Business	0.89	0.66	0.76	200
Digital Transformation	0.79	0.96	0.87	200
EdTech	0.93	0.99	0.96	200
EnergyTech	0.89	0.98	0.93	200
FinTech	0.95	0.95	0.95	200
MedTech	0.93	0.9	0.91	200
Science, Technology and Innovation	0.7	0.56	0.62	200
Telecom	0.91	0.78	0.84	200
accuracy			0.85	1800
macro avg	0.86	0.85	0.85	1800
weighted avg	0.86	0.85	0.85	1800

Full parameter fine-tuning shows better results on the classification task compared to LoRA and baseline model. On average F1 metrics have 0.85 (full parameter fine-tuning), 0.7 for the LoRA and 0.52 for the baseline model.

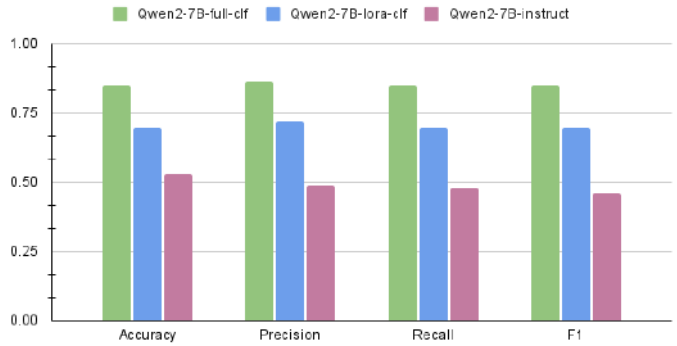

Figure 2. Metrics for classification tasks

Both full-parameter fine-tuning and LoRA exhibit unique characteristics in terms of loss function convergence. While full-parameter fine-tuning offers flexibility and potentially higher performance on complex tasks, LoRA provides efficiency and speed. Comparative analysis of the loss functions for LoRA and full-parameter fine-tuning, demonstrating that both methods exhibit comparable convergence rates (Figure 3).

Figure 3. Training loss for classification task

3.2 QA task metrics

The models were evaluated on the SFQA and LFQA tasks using offline metrics: BERTScore [7], BLEU [8], ROUGE [9] and semantic similarity [10]. The results are presented in Table 10. Average response length of the model with full parameter fine-tuning is most similar to golden responses. There is a slight difference between the models according to the BERTScore metric.

Table 10. Metrics for the QA task on short and long answers

model_name	bleu	ref_len	resp_len	rouge_1	bert_score_f1	sem_sim
qwen2_7b_qa_short_lora	0.119	9.583	9.121	0.312	0.732	0.81
qwen2_7b_qa_long_lora	0.013	336.991	160.425	0.163	0.682	0.732
qwen2_7b_qa_short_full	0.227	9.583	10.053	0.395	0.819	0.893
qwen2_7b_qa_long_full	0.02	336.991	289.002	0.151	0.679	0.89
baseline model short	0.023	9.583	11.123	0.327	0.759	0.813
baseline model long	0.015	336.991	228.266	0.136	0.662	0.854

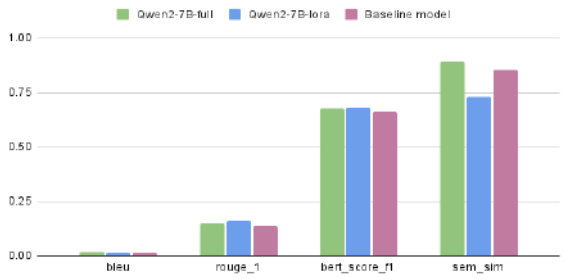

Figure 4. Metrics for QA long answer task

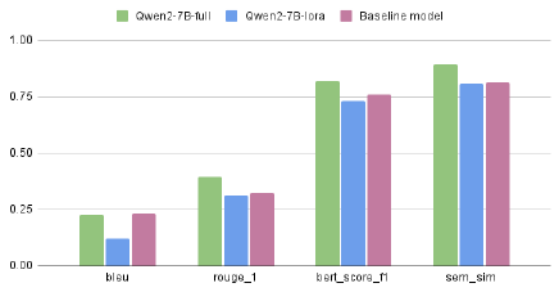

Figure 5. Metrics for QA long short task

4. Related work

4.1 Data

Several studies have been conducted in which authors have developed datasets for similar tasks. The paper [11] introduces the EconQA dataset, designed to assess the performance of LLMs in economics through multiple-choice questions. The authors present the results of ten experiments that varied both the prompts and model variants. Findings from these experiments indicate that the choice of prompt significantly influences the quality of the responses. The dataset was compiled using questions from the test bank of "McConnell and Brue Economics, 16th edition." Initially, this test bank contained approximately 6,000 questions, but after preprocessing, the number was reduced to 3,623.

The paper [12] presents the EconLogicQA benchmark, which is designed to assess the sequential reasoning capabilities of LLMs in the fields of economics, business, and supply chain management. EconLogicQA poses a more challenging problem: it requires models to distinguish and organize multiple interrelated events, reflecting the complexity of economic logic. The dataset includes a variety of scenarios extracted from economic articles that necessitate a deep understanding of both temporal and logical relationships between events. EconLogicQA effectively evaluates undergraduate students' ability to navigate the sequential complexities inherent in economic contexts. To streamline the question generation process and reduce subjectivity, labor intensity, and randomness associated with manual generation, the authors utilize GPT-4 to automatically generate questions by extracting key points from news articles.

Additionally, there are platforms such as Economiga [13], which allow users to engage in dialogue in a question-and-answer format concerning economic issues. Another example of economic data is “Explanations for CommonsenseQA: Dataset” [14].

4.2 Research on LoRA and full fine-tuning

In the study [15], researchers compared full parameter fine-tuning and LoRA-based fine-tuning that shows significant advantages in terms of training costs. The authors conducted an experimental comparison of full fine-tuning and LoRA-based fine-tuning methods using LLaMA as the base model. The results of the experiment showed that the choice of the base model, the scale of the training dataset, the number of trainable parameters, and the cost of model training are important factors.

Due to their learning ability, LLMs can generate a large number of diverse instructions. The authors of the paper translated the original data into Chinese and modified certain elements to better align with Chinese culture and context. Then, using these original data as contextual examples, prompts were created for ChatGPT to generate more examples.

To compare the quality of the models, the authors trained two models using full fine-tuning on training data of 0.6M and 2M instructions. The results demonstrate that full fine-tuning gives better experimental results than LoRA

To assess the quality of the models, the authors trained two models using full fine-tuning on datasets containing 0.6M and 2M instructions. The findings indicate that full fine-tuning yields superior experimental results compared to LoRA. In their study [16], the authors drew the following conclusions regarding the comparison between full fine-tuning and LoRA:

1) Full fine-tuning outperforms LoRA in terms of accuracy and effectiveness, particularly in programming and mathematics domains.

2) LoRA exhibits less forgetting of the original domain, providing a form of regularization.

3) The regularization effect of LoRA is stronger than that of typical regularization methods and contributes to maintaining diversity.

In another work [17] the authors claim that full fine-tuning (FPFT) has become the main choice for adapting language models to problems due to its excellent performance. As the LM size grows, fine-tuning all the LM parameters requires prohibitive amounts of GPU memory. Existing approaches use zero-order optimizers to save GPU memory, which potentially reduces LM performance since non-zero-order optimizers tend to converge faster on most subsequent problems.

The researchers propose a new memory-efficient, optimizer-independent approach, HiFT, which updates only a subset of parameters at each training step. HiFT significantly reduces the number of gradients and optimizer state parameters that reside in GPU memory at the same time, thereby reducing GPU memory consumption.

5. Conclusion

In conclusion, a comparison of the full parameter fine-tuning and LoRA approaches was conducted. Full parameter fine-tuning achieves superior results, as it allows all model parameters to be adapted to specific tasks, making the model versatile enough to address a wide range of problems. However, this approach comes with significant drawbacks, including high computational costs, the need for powerful GPUs, and extended training times, particularly for large models. Additionally, it is prone to overfitting on small datasets, which can diminish its generalization capabilities. After further training, the model can occupy considerable space, complicating deployment and usage.

In contrast, LoRA [16] requires fewer computational resources and less memory which allows using less powerful GPUs. The process of training is faster due to the reduced number of parameters which need updating. LoRA checkpoints are significantly smaller in size which makes storage and deployment easier.

In our study the loss of accuracy is observed compared to full parameter fine-tuning. The effectiveness of LoRA highly depends on training data quality and quantity. The research showed that full parameter fine-tuning leads to higher accuracy rates compared to LoRA on our classification and QA short and detailed answers tasks. For the classification task, the average accuracy rate for full fine-tuning was 0.85 and LoRA achieved 0.7. However, LoRA helps to reduce the time of additional training by 2 to 5 times compared to full parameter fine-tuning.

We also tested the hypothesis that different types of tasks from the same domain increase the quality of generation; in our case, combining training data for the classification task and QA led to a decrease to the level of LoRA metrics.

The choice between full fine-tuning and LoRA depends on the specific requirements of the task, the available computing resources, and the desired level of accuracy. Full fine-tuning is ideal for tasks that require maximum accuracy, while LoRA is an excellent choice for resource and time-constrained scenarios.

References

1. Han Z., Gao C., Liu J. et al. Parameter-efficient fine-tuning for large models: A comprehensive survey // CoRR. 2024. Vol. abs/2403.14608. arXiv: 2403.14608 [cs.LG]. DOI: 10.48550/arXiv.2403.14608.
2. Yang A., Yang B., Zhang B. et al. Qwen2.5 Technical Report // CoRR. 2024. Vol. abs/2412.15115. arXiv: 2412.15115 [cs.CL]. DOI: 10.48550/arXiv.2412.15115.

3. Hu E.J., Shen Y., Wallis P. et al. LoRA: Low-rank adaptation of large language models // CoRR. 2021. Vol. abs/2106.09685. arXiv: 2106.09685 [cs.CL]. DOI: 10.48550/arXiv.2106.09685.
4. Achiam J., Adler S., Agarwal S. et al. GPT-4 technical report // CoRR. 2023. Vol. abs/2303.08774. arXiv: 2303.08774 [cs.CL]. DOI: 10.48550/arXiv.2303.08774.
5. Stack Exchange. URL: https://stackexchange.com/ (accessed: 16.01.2025)
6. Qwen2-7B. URL: https://huggingface.co/Qwen/Qwen2-7B (accessed: 16.01.2025)
7. Zhang T., Kishore V., Wu F.et al. BERTScore: Evaluating Text Generation with BERT // CoRR. 2019. Vol. abs/1904.09675. arXiv: 1904.09675 [cs.CL]. DOI: 10.48550/arXiv.1904.09675.
8. Papineni K., Roukos S., Ward T., Zhu W.-J. BlEU: a Method for Automatic Evaluation of Machine Translation // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002. P. 311–318. DOI: 10.3115/1073083.1073135.
9. Lin C.Y. Rouge: A Package for Automatic Evaluation of Summaries // Text Summarization Branches Out. 2004. P. 74–81.
10. Chandrasekaran D., Mago V. Evolution of Semantic Similarity - a Survey //ACM Computing Surveys (CSUR). 2021. Vol. 54, no. 2. P. 1–37. DOI: 10.1145/3440755.
11. van Patten T. Evaluating Domain Specific LLM Performance Within Economics Using the Novel EconQA Dataset // WWU Honors College Senior Projects. 2023.
12. Quan Y., Liu Z. EconLogicQA: A Question-Answering Benchmark for Evaluating Large Language Models in Economic Sequential Reasoning // CoRR. 2024. Vol. abs/2405.07938. arXiv: 2405.07938 [cs.CL]. DOI: 10.18653/v1/2024.findings-emnlp.125.
13. Economiga Question Bank. URL: https://www.economiga.org/qbank (accessed: 16.01.2025)
14. Aggarwal S., Mandowara D., Agrawal V. et al. Explanations for CommonsenseQA: New Dataset and Models // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021. P. 3050–3065. DOI: 10.18653/v1/2021.acl-long.238.
15. Sun X., Ji Yu., Ma B., Li X. A Comparative Study Between Full-Parameter and LoRA-based Fine-Tuning on Chinese Instruction Data for Instruction Following Large Language Model // CoRR. 2023. Vol. abs/2304.08109. arXiv: 2304.08109 [cs.CL]. DOI: 10.48550/arXiv.2304.08109.
16. Biderman D., Portes J., Ortiz J.J.G. et al. LoRA Learns Less and Forgets Less // CoRR. 2024. Vol. abs/2405.09673. arXiv: 2405.09673 [cs.LG]. DOI: 10.48550/arXiv.2405.09673.
17. Liu Y., Zhang Yi., Li Q. et al. Hift: A Hierarchical Full Parameter Fine-Tuning Strategy // CoRR. 2024. Vol. abs.2401.15207. arXiv: 2401.15207 [cs.LG]. DOI: 10.18653/v1/2024.emnlp-main.1015.