Fine-Tuning Lightweight Large Language Models for a Bilingual DSP Teaching Assistant

Irfan Uruchi

Computer Engineering
Introduction to Data Science – Course Project

Professor: Nuhi Besimi

December 2025

Abstract

The purpose of this project is to test whether a small modern Large Language Model (LLM) can be transformed into a practical teaching assistant for Digital Signal Processing (DSP) without relying on massive server-grade GPU clusters. The main goal is to fine-tune a lightweight model capable of solving numeric DSP problems—such as FFT bin calculations and aliasing—and explaining concepts clearly in both English and Albanian.

The project evolved through six iterations, starting from manually written datasets and gradually expanding into an automated synthetic data generation pipeline. This pipeline uses deterministic Python scripts to generate mathematically correct DSP problems, verifies all numeric steps and packages them in an instruction format suitable for fine-tuning. By ensuring correctness at the dataset level, the system avoids common LLM issues such as hallucinated formulas, inconsistent arithmetic and unstable reasoning.

Ten foundation models were evaluated under the same conditions, including LLaMA 3.2 1B and 3B variants, Qwen 2.5 and Qwen 3, Alpaca 7B, Mistral 7B, LLaMA 3.1 8B, Gemma 2 9B, GPT-4o mini and Phi-3 mini. Fine-tuning was performed using parameter-efficient methods (LoRA and QLoRA). The fine-tuned LLaMA 3.2 1B model delivered strong performance for its size, achieving around 82% accuracy on numerical DSP tasks in English and showing usable bilingual transfer to Albanian.

These results suggest that high-quality, verified synthetic data paired with lightweight modern architectures can outperform older and even larger models on narrow technical domains. The project concludes that data quality, verification and format consistency matter more than raw parameter count in technical fields like DSP, where mathematical correctness is essential.

I. Introduction

Large Language Models (LLMs) are rapidly becoming useful tools in technical education, especially in fields where both mathematical reasoning and clear explanations are required. Digital Signal Processing (DSP) is one of these subjects, combining strict numerical rules with concepts that students often find difficult to connect. Topics such as FFT bin spacing, sampling theory and filter behaviour require both conceptual understanding and step-by-step calculations.

This project investigates whether a small but modern LLM can be fine-tuned to act as a reliable bilingual DSP teaching assistant using only consumer-grade hardware. Rather than relying on multi-GPU servers or 13B–70B models, the focus is on a 1B-parameter model running within roughly 5–6 GB of VRAM on a desktop GPU, and on comparing it to other models up to 9B parameters that remain accessible for students and small research groups.

The core idea is that careful dataset design, numerical validation and modern fine-tuning techniques can compensate for smaller model sizes. To explore this, the project progressed through six iterations. Early stages focused on manual dataset creation, environment setup and defining a bilingual schema. Later, an automated DSP dataset generator capable of producing mathematically verified problems in both English and Albanian was developed, addressing weaknesses such as incorrect formulas, hallucinated constants and unstable arithmetic.

In the final stages, ten different models were evaluated under identical conditions. By comparing LLaMA 3.2, Qwen 2.5 and Qwen 3, Alpaca, Gemma 2, GPT-4o mini and others, it was possible to measure how much improvement comes from domain-specific fine-tuning compared to raw model size. The results show that with the right dataset and training approach, even very small models can reach meaningful domain-specific performance.

II. Literature Review / Related Work

II.1. Parameter-Efficient Fine-Tuning Techniques

Fine-tuning large language models traditionally requires updating all model weights, which is expensive for very large architectures. Recent research has focused on parameter-efficient fine-tuning techniques. Low-Rank Adaptation (LoRA), introduced by Hu et al. (2021), reduces the number of trainable parameters by injecting small trainable low-rank matrices into each transformer layer while keeping the original weights frozen. This considerably lowers the number of trainable parameters and the GPU memory needed, while achieving performance comparable to full-model fine-tuning on many tasks.

Building on LoRA, QLoRA (Dettmers et al., 2023) further reduces memory usage by quantizing model weights to 4-bit precision during fine-tuning. Gradients are backpropagated through a 4-bit quantized model into LoRA adapters, enabling even 65B-parameter models to be fine-tuned on a single 48 GB GPU without losing accuracy. Dettmers et al. (2023) also report that a QLoRA-based model ("Guanaco") reached 99% of ChatGPT's performance on an open benchmark after only 24 hours of training. These advances show that low-VRAM fine-tuning of LLMs is possible and, importantly for this project, that even small 1–3B models can be fine-tuned in realistic conditions on consumer GPUs.

II.2. Multilingual and Low-Resource LLM Performance

Most LLMs to date have been English-centric, with weaker performance in low-resource languages. On the large-model side, projects such as BLOOM (BigScience Workshop, 2023) trained on dozens of languages to provide broad multilingual coverage, but still favour high-resource languages. Another approach is cross-lingual transfer: Muennighoff et al. (2023) show that fine-tuning a multilingual model only on English tasks can still yield zero-shot performance in other languages present in pre-training.

Instead of relying exclusively on extremely large models, recent work has explored bilingual fine-tuning of moderate-sized LLMs for underrepresented languages. Bao et al. (2023) created a Galician instruction dataset (around 52k translated instruction–response pairs) and used it to fine-tune LLaMA 7B, producing a model that can follow instructions in Galician, a language not explicitly supported by the base model. The authors also leveraged Portuguese—a related high-resource language—to improve Galician coherence. Similar efforts exist for Italian, for example LLaMAntino models (Basile et al., 2023). These results motivate the strategy in this project: fine-tuning a 1–3B model to act as a bilingual DSP teaching assistant in English and Albanian.

II.3. Synthetic Domain Data and Numeric Verification

A key challenge in training domain-specific LLMs, especially for fields like DSP, is the lack of high-quality labelled data in underrepresented languages. One way to address this is synthetic dataset generation, combining programmatic problem generation with automated solution verification.

For instance, the SYNTHETIC-1 initiative (Prime Intellect, 2025) compiles over a million verifiable math problems and reasoning traces, using deterministic logic and evaluation scripts to guarantee correctness. This strategy is particularly effective in technical domains where correctness can be validated numerically or symbolically. The Minerva project (Lewkowycz et al., 2022) takes a similar approach, training on scientific texts and math problems and using chain-of-thought prompting to improve step-by-step reasoning. Its models perform well on advanced STEM questions without external tools.

Inspired by these methods, this project uses custom Python scripts to generate bilingual DSP problems (e.g., FFT, aliasing, sampling) and to verify the solutions numerically. Synthetic questions and answers are only accepted into the dataset if they pass verification checks, so the model learns from correct and domain-aligned data.

II.4. Domain-Specific Fine-Tuning for DSP

Domain-specific fine-tuning has been shown to improve performance significantly in specialised fields. SciBERT and BioBERT (Beltagy et al., 2019; Lee et al., 2020) are examples of models tuned on scientific and biomedical texts, outperforming general-purpose LLMs in their respective domains. More recently, Meta AI's Galactica (Galactica Team, 2022), trained on scientific corpora, outperformed much larger general-purpose models (such as GPT-3 and PaLM) on math and scientific reasoning benchmarks.

These findings suggest that smaller models can outperform larger ones on specialised tasks when trained on targeted, high-quality data. This project applies the same principle to DSP: by fine-tuning on verified DSP questions (covering transforms, sampling, filters and related concepts) the model learns both the content and the typical reasoning patterns of the field. Using LoRA helps keep the fine-tuning efficient while preserving the base model's general linguistic abilities.

III. Methodology

The development process was structured into six stages, from setting up the environment to evaluating the final model. Each stage corresponds to an iteration of the project.

III.1. Step 1: Setting Up the Environment

In the first stage, the development and training environment was set up. The system used Windows 11 with an Ubuntu 22.04 WSL2 backend, a desktop GPU with a VRAM budget of around 6 GB and sufficient system RAM. The objective was to demonstrate that LLMs can be fine-tuned with modest hardware.

Python 3.12 was installed, and a virtual environment was created inside WSL to manage dependencies. Core libraries included PyTorch with CUDA support, Hugging Face Transformers and Datasets, the PEFT library for parameter-efficient fine-tuning and `bitsandbytes` for 4-bit quantisation. Several candidate base models (such as Qwen 1.5B) were loaded to confirm that the environment worked reliably within the imposed memory limits.

III.2. Step 2: Manual Dataset Creation

After the environment was ready, the focus shifted to defining the dataset structure and creating initial content. A strict JSON schema was designed for each Q&A record, including a unique ID, the DSP topic, an English prompt (`prompt_en`) and its Albanian translation (`prompt_sq`), the corresponding answers in each language and a verification flag. Each Q&A pair was kept atomic, meaning a single precise question and answer, which simplifies evaluation and verification.

A bilingual English–Albanian glossary was also created for DSP terminology to keep translations consistent. Terms such as "sampling rate" and "quantization noise" were standardised as "frekuenca e mostrimit" and "zhurmë e kuantizimit". The glossary also ensured the correct use of characters such as "ë" and "ç" in Albanian. By the end of this step, the project had a small, clean, manually created bilingual DSP dataset and a clear schema for future expansion.

III.3. Step 3: Automated Synthetic Dataset Preparation

The third stage introduced quality control and automation tools to prepare the dataset for larger-scale generation. A dataset card was written to document the dataset's purpose, composition and generation pipeline, as well as any potential risks and licensing considerations. A simple Q&A quality rubric was defined to guide the creation of clear questions and correct, pedagogically useful answers in both languages.

Several utility scripts were implemented. A cleaning script removed HTML fragments and URLs and normalised whitespace. A deduplication tool filtered out near-duplicate entries to preserve diversity. A leak-proof splitting script partitioned data into training, validation and test sets using hash-based grouping to avoid any overlap. At this stage, numerical answers were checked either via regular expressions or by re-computing the expected value with small Python functions. This provided an automated pipeline for basic preprocessing and quality assurance.

III.4. Step 4: Automated Synthetic Data Generation with a Teacher–Student Pipeline

Once the foundations were in place, the fourth stage moved to automated data generation using a teacher–student setup. The goal was to scale up the dataset with high-quality DSP Q&A pairs while preserving bilingual alignment and numerical correctness.

The repository structure was reorganised, and new modules under `src/` were created for generation and validation. Generator modules produced parameterised problems for different DSP topics (for example, FFT bin index, sampling aliasing, frequency response), while verifier modules recomputed numeric answers and enforced constraints. A linguistic module used the glossary to enforce consistent terminology.

Large LLMs were then used as teachers in this pipeline. LLaMA 3.3 70B and Qwen 3 235B were prompted to refine draft problems and explanations. The generator first created a rough English question and answer; then the teacher model rewrote the question more clearly and produced an Albanian version of both the question and the answer. These outputs were passed through numeric and linguistic verifiers before being accepted into the dataset.

In this way, the project built a teacher–student synthetic pipeline: stronger LLMs helped produce clearer and more fluent training data, while deterministic scripts verified the correctness of the mathematical content.

III.5. Step 5: Pilot Fine-Tuning and Pipeline Refinement

In the fifth stage, the new dataset was used for a pilot fine-tuning run on a 1.5B-parameter model. The goal was to validate the whole pipeline end-to-end under low-resource constraints.

LoRA-based fine-tuning was used to fit within the 6 GB VRAM limit by adding only small trainable adapter matrices. The pilot experiment used around 100 curated FFT-related examples and ran for two epochs. Training hyperparameters were tuned for stability and memory usage; for example, a small batch size with gradient accumulation was used to emulate a larger effective batch size.

A custom inference script was built to evaluate the model. It applied a fixed prompt template to each test question and used post-processing to remove trailing artefacts. A simple math-checker parsed the model's answer, extracted key values and recomputed the expected result, flagging mismatches. This pilot run confirmed that the training code, dataset pipeline and verification logic worked correctly and highlighted small issues that were fixed before the final training.

III.6. Step 6: Final Fine-Tuning and Multilingual Evaluation

In the final stage, the complete bilingual DSP dataset was used to fine-tune the target model LLaMA 3.2 1B. The model was loaded in 4-bit precision and fine-tuned with LoRA to remain within approximately 5–6 GB of VRAM. The dataset covered diverse DSP topics with aligned English–Albanian pairs, and all numerical answers in the dataset had been verified.

Three evaluation pipelines were defined: an English-only DSP set, an Albanian-only set and a bilingual set mixing both languages. For each pipeline, the model's outputs were evaluated using automatic numeric checks and manual scoring of explanations. Terminology consistency was also inspected, particularly in Albanian and bilingual responses.

The fine-tuned LLaMA 1B model was then compared against several other models from 1.5B to 9B parameters using the same question sets and evaluation scripts. While larger models produced more fluent and stylistically polished answers, the fine-tuned 1B model achieved strong numerical accuracy and consistent task behaviour, confirming that the teacher–student pipeline and fine-tuning strategy were effective.

IV. Results

This section presents the evaluation outcomes obtained after the final fine-tuning stage and the cross-model comparison. All models were tested under the same conditions using the three pipelines described previously. Numerical DSP results were verified with an automated Python checker, while conceptual responses were manually graded for correctness, clarity and terminology consistency.

IV.1. Baseline vs Fine-Tuned LLaMA 3.2–1B

Before comparing to other models, the fine-tuned LLaMA 3.2 1B was evaluated against its own base version. The base model performed poorly on DSP tasks, especially numerical computations, with accuracy between roughly 30% and 35% across all three pipelines.

After fine-tuning on the verified DSP dataset, accuracy increased substantially to a range of about 70% to 80%, depending on the language. On English numeric tasks, the fine-tuned model

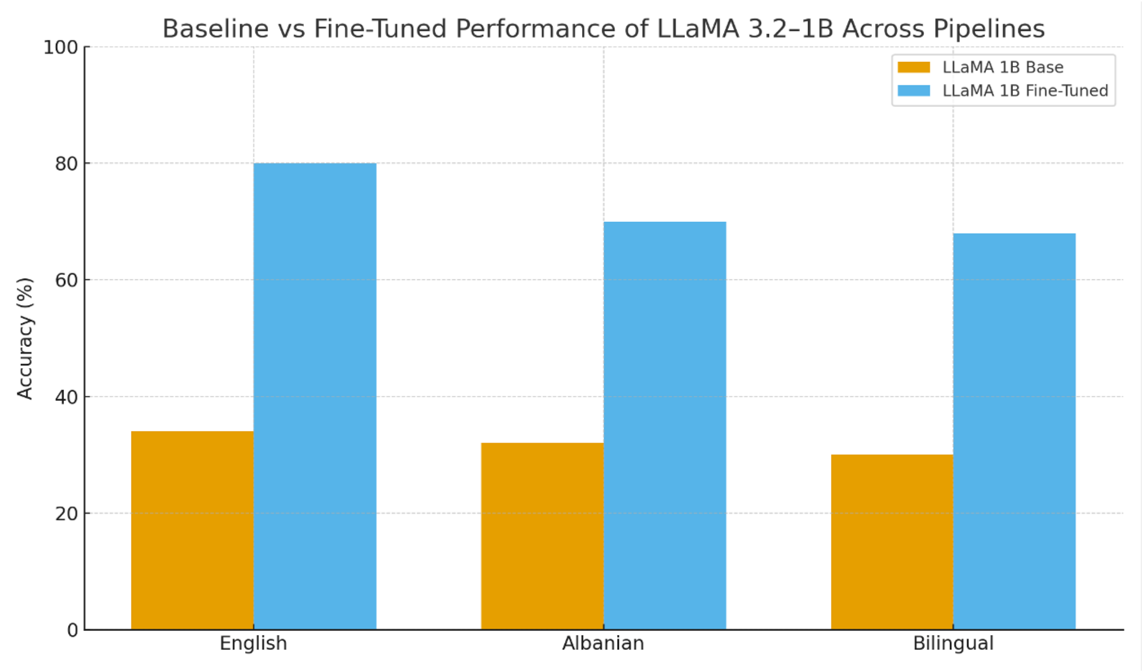

Figure 1: Baseline vs fine-tuned performance of LLaMA 3.2–1B across English, Albanian and bilingual DSP pipelines.

reached roughly 80% accuracy, while Albanian accuracy was around 70%. The bilingual pipeline, where prompts and answers could mix languages, improved from around 30% to about 68%.

Figure 1 summarises these results. The main observations are:

- English accuracy more than doubled after fine-tuning.
- Albanian accuracy increased from barely usable to a reliable level for a 1B model.
- Bilingual accuracy also improved significantly, although it remained the most challenging setting.

IV.2. Cross-Model Comparison Across Pipelines

To compare the fine-tuned 1B model with other architectures, nine additional models from 1.5B to 9B parameters, plus GPT-4o mini, were evaluated on the same question sets and with the same scoring protocol. Figure 2 shows the resulting accuracies for the English, Albanian and bilingual pipelines.

Several patterns emerge. First, English performance is highest for every model, reflecting the greater availability of English data during pre-training. Second, Albanian accuracy drops across all models, confirming its status as a low-resource language. Third, the bilingual pipeline scores are generally the lowest, as mixing languages tends to trigger grammar drift and code-switching.

Despite its small size, the fine-tuned LLaMA 1B model reaches performance close to mid-range models in the 3B–7B class on English DSP tasks. It even outperforms Alpaca 7B numerically, where Alpaca often produces longer explanations but less reliable calculations due to its older architecture and lack of DSP specialisation.

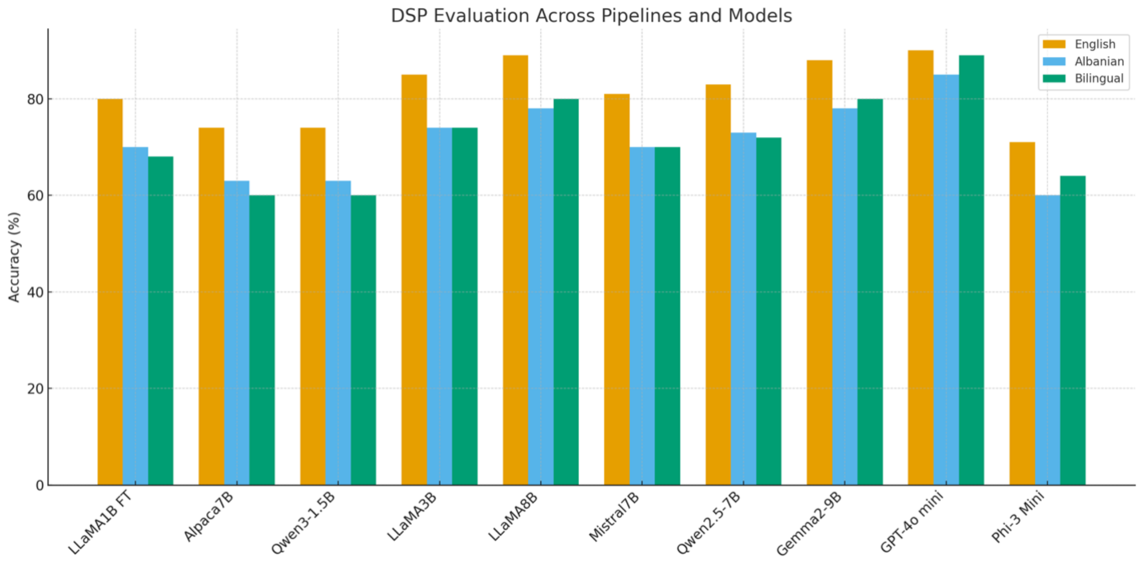

Figure 2: DSP evaluation across English, Albanian and bilingual pipelines for all evaluated models.

IV.3. Multilingual Behaviour and Error Trends

All models exhibit the same general ranking across pipelines: English performs best, followed by Albanian, then the bilingual setup. This is consistent with expectations for low-resource languages and code-mixed prompts.

Typical linguistic issues include missing diacritics (“ë” and “ç”), literal translations, unnatural word order and mid-sentence code-switching. Stronger Albanian grammar appears in GPT-4o mini, Gemma 2 9B and LLaMA 3.1 8B, while Phi-3 Mini, Alpaca 7B and Qwen 3 1.5B show weaker Albanian performance.

Common numerical errors involve FFT bin indexing, rounding and incomplete step-by-step reasoning. However, the fine-tuned LLaMA 1B model benefits from the verified dataset and exhibits fewer hallucinated constants or completely incorrect formulas.

IV.4. Resource Efficiency

A major goal of the project is to test whether DSP-specialised LLMs can be trained under low-resource conditions using parameter-efficient fine-tuning. The LLaMA 3.2 1B model was fine-tuned within approximately 6 GB of VRAM. In contrast, some larger models such as Gemma 2 9B (without quantisation) required around 16–24 GB of VRAM for inference alone.

These observations support the idea that, with a carefully designed dataset and LoRA/QLoRA, it is feasible to train effective domain-specific LLMs on consumer hardware. This is important for students and smaller institutions that do not have access to large GPU clusters.

V. Discussion

The results show that fine-tuning a small 1B-parameter model can significantly improve DSP performance, even under strict hardware limitations. The base LLaMA 3.2 1B model was not reliable for numerical reasoning or multilingual explanations, but after fine-tuning it reached accuracy levels that make it usable as a teaching assistant.

Comparisons with larger models suggest that data quality and specialisation can compensate for small model size. On English numeric tasks, the fine-tuned LLaMA 1B approaches or matches models in the 3B–7B range and surpasses Alpaca 7B on several DSP benchmarks. This supports a central conclusion of the project: in a narrow technical domain, a small model with high-quality, verified training data can compete with or outperform larger, more generic models.

At the same time, the multilingual analysis highlights remaining limitations. All models perform worse in Albanian and in bilingual settings compared to English. The fine-tuned 1B model is usable in Albanian but still suffers from occasional grammar issues and terminology inconsistencies. Larger models such as Gemma 2 9B demonstrate more stable multilingual grammar, indicating that pre-training diversity is still an important factor.

Another important observation is the effect of verification. The strict numeric verification pipeline prevents the fine-tuned model from learning incorrect formulas or constants, which is a common problem when models are trained on noisy or unverified data. As a result, the fine-tuned model rarely produces hallucinated numerical facts in the tested tasks. This suggests that synthetic but verifiable training data is especially valuable for technical disciplines.

Overall, the project supports the idea that domain-specific models for university courses can be trained with modest hardware and open-source tools, as long as careful attention is paid to dataset design, verification and the choice of fine-tuning methods.

VI. Conclusion and Future Work

This project shows that a lightweight LLM can be transformed into a bilingual DSP teaching assistant through targeted dataset design, numeric verification and parameter-efficient fine-tuning. By progressing through six development stages, from manual dataset creation to a fully automated teacher–student pipeline, it was possible to construct a clean, scalable dataset aligned with DSP concepts in both English and Albanian.

The final fine-tuned LLaMA 3.2 1B model achieved around 80% accuracy on English numeric DSP tasks and about 70% on comparable Albanian tasks. These results indicate that even small models can provide meaningful domain-specific assistance when trained on verified and well-structured data. The comparison with other models from 1.5B to 9B parameters suggests that correctness, alignment and instruction formatting influence DSP performance at least as much as parameter count.

All code, training scripts and sample datasets are published in a public GitHub repository, and the two fine-tuned models (LLaMA 3.2–1B DSP and Qwen 2.5–1.5B DSP) are released on HuggingFace together with model cards and basic usage examples.

There are several directions for future work. First, the dataset can be expanded beyond FFT, sampling and basic filter topics to include convolution, z-transforms, filter design and time–frequency analysis. Second, Albanian performance could be improved further by incorporating a larger Albanian corpus and adding grammar-focused augmentation. Third, a simple user interface or web API could be built so students can interact with the model by typing questions and receiving step-by-step answers.

Finally, it would be interesting to explore reinforcement learning from human feedback (RLHF) or direct preference optimisation (DPO) on top of the current model, with the goal of improving the clarity and structure of explanations rather than focusing only on numerical accuracy.

References

[1] Bao, E., Pérez, A., & Parapar, J. (2023). Conversations in Galician: A large language model for an underrepresented language. arXiv. `https://doi.org/10.48550/arXiv.2311.03812`

[2] Basile, P., Cassotti, P., de Gemmis, M., Gentile, A. L., Lops, P., & Semeraro, G. (2023). LLaMAntino: LLaMA 2 models for effective text generation in Italian language. arXiv. `https://arxiv.org/abs/2312.09993`

[3] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (pp. 3615–3620).

[4] BigScience Workshop. (2023). BLOOM: A 176B-parameter open-access multilingual language model. arXiv. `https://arxiv.org/abs/2211.05100`

[5] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. arXiv. `https://arxiv.org/abs/2305.14314`

[6] Galactica Team (Taylor, R., Kardas, M., Cucurull, G., Scialom, T., et al.). (2022). Galactica: A large language model for science. arXiv. `https://arxiv.org/abs/2211.09085`

[7] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., Wang, S., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv. `https://arxiv.org/abs/2106.09685`

[8] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234–1240.

[9] Lewkowycz, A., Sellam, T., Menick, J., et al. (2022). Minerva: Solving quantitative reasoning problems with language models. arXiv. `https://arxiv.org/abs/2206.14858`

[10] Muennighoff, N., Tazi, N., Wenzek, G., et al. (2023). Crosslingual generalization through multitask finetuning. arXiv. `https://arxiv.org/abs/2211.01786`

[11] Prime Intellect. (2025). SYNTHETIC-1 release: Two million collaboratively generated reasoning traces from DeepSeek-R1. Blog post. `https://www.primeintellect.ai/blog/synthetic-1-release`

Appendix

A. Sample Dataset (Before Fine-Tuning)

Below are two examples from the early bilingual DSP dataset used before the full synthetic generator was built. They illustrate the JSON structure and the alignment between English and Albanian prompts.

```
{
  "id": "fft_0031",
  "topic": "FFT",
  "prompt_en": "A signal is sampled at 8000 Hz and contains a
  tone at 1000 Hz. What is the FFT bin index for N = 256?",
  "answer_en": "k = (1000 / 8000) * 256 = 32.",
  "prompt_sq": "Një sinjal mostrohet me 8000 Hz dhe përmban një
  frekuencë prej 1000 Hz. Cili është indeksi FFT për N = 256?",
  "answer_sq": "k = (1000 / 8000) * 256 = 32.",
  "verified": true
}

{
  "id": "alias_0015",
  "topic": "Sampling & Aliasing",
  "prompt_en": "Does a 13 kHz tone alias when sampled at 20 kHz?
  If yes, find the aliased frequency.",
  "answer_en": "Yes. f_alias = |13 - 20| = 7 kHz.",
  "prompt_sq": "A pëson aliasing një ton prej 13 kHz kur
  mostrohet me 20 kHz? Nëse po, gjej frekuencën e aliasuar.",
  "answer_sq": "Po. f_alias = |13 - 20| = 7 kHz.",
  "verified": true
}
```

B. Evaluation Criteria

Each model (1B–9B) was evaluated using the same scoring template. For every question, four aspects were considered:

1. **Numeric correctness**: whether the numeric result matches the expected value.
2. **Reasoning quality**: whether the model shows the essential steps (e.g., FFT index formula, aliasing rules).
3. **Terminology accuracy**: correct and consistent use of DSP terms in the relevant language.
4. **Albanian or bilingual consistency**: grammar, diacritics (“ë”, “ç”) and avoidance of unnecessary code-switching.

The final accuracy score for a model and pipeline is the average across these dimensions.

C. System and Environment Setup

Hardware

- GPU: NVIDIA RTX 3050 (desktop) – 8 GB VRAM
- CPU: Intel Core i7-12700K
- RAM: 64 GB
- OS: Windows 11 with Ubuntu 22.04 via WSL2

Software

- Python 3.12
- PyTorch with CUDA support

- Hugging Face Transformers and Datasets
- PEFT (LoRA / QLoRA)
- `bitsandbytes`
- `accelerate`, `datasets` and supporting libraries

This configuration demonstrates that fine-tuning is possible on hardware that many students and small labs can access, not only on specialised servers.

D. Before and After Fine-Tuning

Before fine-tuning, the base LLaMA 1B often produced incorrect FFT indices, skipped steps and gave vague or speculative answers (for example, suggesting a bin "around 20 or 30" without a clear calculation).

After fine-tuning, the LLaMA 1B model produced precise and numerically correct answers for many tasks. On the same FFT example, a typical output became:

"The FFT bin index is $k = (1000/8000) \times 256 = 32$*, so the tone appears at bin 32."*

This illustrates the effect of training on verified data with explicit step-by-step reasoning.

E. External Links

HuggingFace Models

- Fine-tuned LLaMA 3.2–1B DSP model:
  `https://huggingface.co/Irfanuruchi/llama_3_2_1B_dsp-llm-bilingual`
- Fine-tuned Qwen 2.5–1.5B DSP model:
  `https://huggingface.co/Irfanuruchi/qwen2.5-1.5b-dsp-finetuned`

GitHub Repository

- Full project (code, dataset samples and evaluation scripts):
  `https://github.com/IrfanUruchi/dsp-llm-bilingual-finetuning`