# A. Artifact Appendix

## A.1 Abstract

This artifact provides a complete workflow to reproduce the performance evaluation of EdgeLoRA, a multi-tenant LLM serving system optimized for edge devices. The source code is publicly available at `https://github.com/shenzheyu/EdgeLoRA.git`, with precompiled binaries hosted online for convenience. The artifact supports deployment on Jetson AGX Orin, Jetson Orin Nano, and Raspberry Pi 5 devices, and includes detailed installation and execution instructions.

To reproduce the results in the paper, users can launch the EdgeLoRA server with two arguments specifying the model and the number of LoRA adapters, followed by an experiment script that simulates synthetic workloads. Metrics such as throughput, average request latency, first-token latency, and SLO attainment are automatically reported. The default experiment replicates the results presented in the paper, while the setup can be easily customized via parameters such as request rate, adapter count, and input/output lengths. The full experiment completes within minutes and requires approximately 20GB of disk space.

By following the provided steps, users can replicate the benchmark results or conduct customized experiments to evaluate EdgeLoRA's scalability and efficiency across a wide range of configurations.

## A.2 Artifact check-list (meta-information)

- **Compilation:** `gcc/g++`, `nvcc`
- **Binary:** Precompiled EdgeLoRA binary available at `https://github.com/shenzheyu/EdgeLoRA/releases/tag/v1.0.0`
- **Hardware:** Jetson Agx Orin Developer Kit, Jetson Orin Nano, and Rasperry Pi 5
- **Metrics:** Throughput, average request latency, average first-token latency, and SLO attainment.
- **Output:** Printed summary of performance metrics to terminal
- **Experiments:** Described below
- **How much disk space required (approximately)?:** 20GB
- **How much time is needed to prepare workflow (approximately)?:** 1 hour
- **How much time is needed to complete experiments (approximately)?:** 10 minutes per configuration
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** MIT License
- **Archived (provide DOI)?:** 10.6084/m9.figshare.28675676

## A.3 Description

### A.3.1 How to access

- Source code: `https://github.com/shenzheyu/EdgeLoRA.git`
- Binary release: `https://github.com/shenzheyu/EdgeLoRA/releases/tag/v1.0.0`

### A.3.2 Hardware dependencies

To match the experiment setup described in the paper, EdgeLoRA is evaluated on the following devices:

- Jetson AGX Orin Developer Kit
- Jetson Orin Nano
- Raspberry Pi 5

### A.3.3 Software dependencies

- Ubuntu 22.04
- L4T Driver Package Version: 36.6.3
- JetPack Version: 6.2
- g++ Compiler: 11.4.0
- Node.js: 20.18.3

## A.4 Installation

The following steps describe how to install EdgeLoRA from source:

```
# clone the EdgeLoRA repository
git clone https://github.com/shenzheyu/EdgeLoRA.git

# compile the source code
cd EdgeLoRA/edgelora
export GGML_CUDA=1 # enable CUDA if device has a GPU
make llama-server

# download pre-trained models and adapters
pip install gdown
gdown https://drive.google.com/uc\?id\=1
     cyU2MUe8V4bo4IuKZG7cEZ2wpxJzD0nn
tar -xzvf models.tar.gz

# install the dependencies of experiment script
cd llama-client
npm install gamma progress
```

## A.5 Experiment workflow

To reproduce the default experiment for EdgeLoRA using the Llama3.1-8B model and 20 LoRA adapters:

```
# launch the EdgeLoRA server
bash server.sh Llama3.1-8B 20

# run the default experiment script
cd llama-client
node edge_lora.js
```

The script prints the resulting throughput, average request latency, first-token latency, and SLO attainment directly to the terminal.

## A.6 Evaluation and expected results

The server should launch successfully, and the initial terminal output should indicate that all slots are idle. After running the experiment, performance metrics should be displayed. These results are expected to be consistent with the values reported in the paper, validating the correctness of the artifact setup.

## A.7 Experiment customization

The server can be launched using the following command with two arguments: `bash server.sh <model> <lora_count>`.

- **model**: Specifies the name of the base language model to be served. Supported options include `OpenELM-1.1B`, `Llama3.2-3B`, and `Llama3.1-8B`.
- **lora_count**: Indicates the total number of LoRA adapters to be managed by the server. This value can range from a few dozen to several thousand.

The above experiment script could also be customized with multiple arguments in the 'llama-client/edge_lora.js' file:

- **n**: Number of LoRA adapters available in the system. Controls adapter diversity.
- **alpha**: Power-law exponent that defines the skewness of request distribution across adapters.

- **R**: Total request rate, i.e., how many requests per second are sent across all adapters.
- **cv**: Coefficient of variance for arrival intervals in the Gamma process, defining burstiness of the workload.
- **traceDuration**: Duration of the synthetic trace (in milliseconds), default representing 5 minutes.
- **Il, Iu**: Lower and upper bounds for input token lengths sampled from a uniform distribution.
- **Ol, Ou**: Lower and upper bounds for output token lengths, also sampled uniformly.

### A.8 Notes

- The `edgelora_wo_aas` folder contains the implementation of EdgeLoRA without adaptive adapter selection. Its usage is similar to the standard EdgeLoRA workflow.
- The `adapter-router` folder provides the implementation for fine-tuning and evaluating the adapter router. This component requires a custom version of the HuggingFace Transformers library, which can be installed using:

```
pip install git+https://github.com/shenzheyu/
     transformers.git@edgelora#egg=transformers
```