

# Overview Document

**“Bringing auto-tuning to HIP: Analysis of tuning impact and difficulty on AMD and Nvidia GPUs” by Milo Lurati, Stijn Heldens, Alessio Sclocco and Ben van Werkhoven**

## Getting started Guide

To run all search spaces of all kernels on all devices is a task that takes days. Hence, this artifact generates only plots and data used in the paper, rather than running all search spaces from scratch. It achieves this by utilizing cache files included in the artifact. However, we still provide all kernel scripts for reference and will explain how to execute them if necessary.

---

Ensure that Python pip is installed on your system. The code makes use of the BlooPy Python package. Please ensure the latest version is installed:

```
pip install bloopy
```

To re-create the plots requires the seaborn package.

```
pip install seaborn
```

lastly we need to install networkx.

```
pip install networkx
```

---

If you want to reproduce the cache files and run all the kernels you will be needing access to the following devices: AMD Radeon PRO W6600, AMD Instinct MI250X, NVIDIA RTX A4000, NVIDIA A100-PCIE-40GB.

Next you will need to install HIP and Kernel Tuner. To install HIP, please consult the corresponding AMD resources. Once HIP is install, proceed to install Kernel Tuner:

```
pip install kernel_tuner[hip]
```

## Step-by-Step Instructions

If you want to reproduce all experiments described in the paper with one command, you can do so with the script `run_experiments.py`:

```
python run_experiments.py
```

In `run_experiments.py`, by default, there are two lines of code commented out. These lines process the cache files and compute and analyze the FFGs. As these results are included by default in the repository, `run_experiments.py` will only generate the plots and data. All plots are saved in the directory `plots`, and all data is printed to the terminal.

If you prefer running the different experiments separately, you can follow these steps.

### Plot pagerank centralities

To plot FFGs proportion of PageRank centralities run:

```
python plot_centralities.py
```

## Creating and plotting FFGs

To create new FFGs, run:

```
python compute_and_analyze_FFGs.py
```

By default, the script creates the FFG and computes the PageRank centralities (and saves them). By uncommenting line 198, the script will also draw the graph using networkX and save it as PDF. NOTE: Plotting FFGs is very expensive and may take a lot of RAM and time to plot.

## Plot violins and calculate search space statistical values

To plot the violins and calculate the statistical values run:

```
python violins.py <kernel name>
```

Give kernel name as argument (convolution, hotspot, dedisp, gemm).

## Plot Performance Portability

```
python performance_portability.py
```

## Calculate Top Configurations

```
python top_configurations.py <kernel name>
```

Give kernel name as argument (convolution, hotspot, dedisp, gemm).

---

If needed, you can run each python script of the corresponding kernel. You can find these in the directory `cache_scripts`. For example you can run:

```
python cache_scripts/convolution/convolution.py
```

NOTE: Remember that this process will take DAYS to execute!