

|title|

CONTENTS

I	Introduction	2
II	Related Work	2
III	Implementation of optimization with OpenCL	2
IV	Implementation of optimization with CUDA	2
V	Benchmarks on Workload	2
VI	Experimental Results	2
VII	Conclusion & Outlook	2
VIII	References	2

I. INTRODUCTION

This paper compares the implementation of various optimization algorthims with high-performance GPGPU processing frameworks like OpenCL [?] and CUDA [?].

The utilization of GPGPU has become more important over the past years due the interest in training deep neural networks [?]. Thus the interest in software frameworks that utilize GPGPU efficiently also has become more important in the past years [?]. One large provider of GPGPU Nvidia propsed their software framework CUDA in 2007 and is now widely utilized by many common deep learning libaries as of Pytorch, Keras and Tensorflow [?]. CUDA provides a software framework for C/C++ and Python to utilize and execute workloads on the GPU. Here the user is only limited to Nvidia GPU which leaves outer other large vendors of GPU chips as of AMD [?]. Thus it is in interest to compare the performance of OpenCL and CUDA to don't miss the GAP.

As mentioned GPGPU are widely used for training and running neural networks or other machine learning models. An impression gives for example the planned data center from german schwarz group which plans to install 100'000 GPUs to train large model with AI inference. [<https://schwarz-digits.de/en/presse/archive/2025/ground-breaking-ceremony-in-luebbenauschwarz-digits-invests-11-billion-euros-in-europe-s-digital-sovereignty>]

II. RELATED WORK

III. IMPLEMENTATION OF OPTIMIZATION WITH OPENCL

...

IV. IMPLEMENTATION OF OPTIMIZATION WITH CUDA

V. BENCHMARKS ON WORKLOAD

VI. EXPERIMENTAL RESULTS

VII. CONCULION & OUTLOOK

VIII. REFERENCES