Additional results, *Machine learning based auto-tuning for enhanced performance portability of OpenCL applications*

Thomas L. Falch
Department of computer and information science, Norwegian University of Science and Technolog
email: thomafal@idi.ntnu.no

This repport contains additional results for the paper:

Machine learning based auto-tuning for enhanced performance portability of OpenCL applications
Thomas L. Falch and Anne C. Elster
Concurrency and Computation, Practice and Experience, 2016

which were not included in that paper due to space considerations.

To evaluate our auto-tuners ability to find good configurations, we measured the actual execution times for all possible configurations, for all combinations of devices and benchmarks. While this process is too time consuming to perform as a part of a normal auto-tuning procedure, it allows us to assess the quality of the output of our auto-tuner by comparing it to the ground truth, that is, to the known globally optimal configurations. As explained in the previous section, the model, and therefore the best configuration found, depends upon the configurations used for training, and the random initial weights of the neural network. All the results presented in this section is therefore the average of 30 runs. The quality of the output depends on the number of samples used for training, as well as the threshold used in the second stage, which controls the size of the subspace of interesting configurations which are searched exhaustively. The figures in this report shows the results for the median, raycast, stereo and bilateral benchmarks, for the different devices. The figure shows the slowdown of the best configuration found by the auto-tuner compared to the known globally best configuration, as the number of training samples, and second stage thresholds are varied.

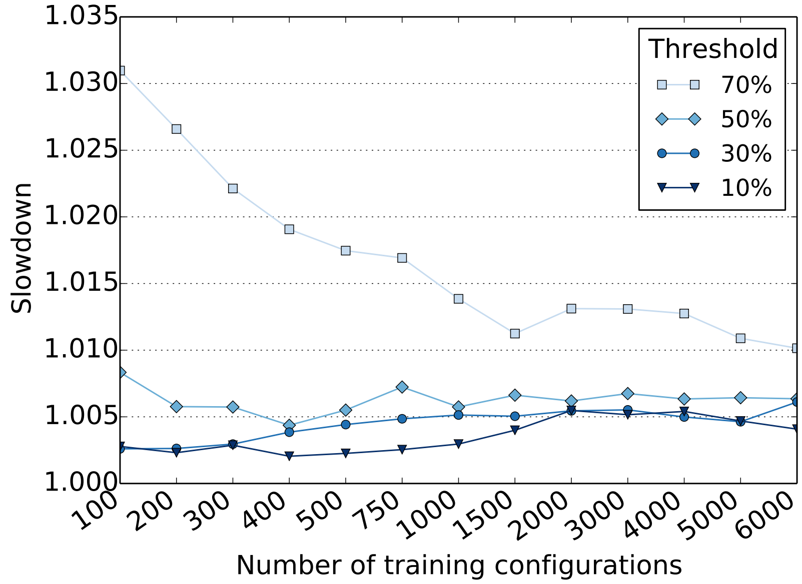

Figure 1: `bilateral`, AMD 7970

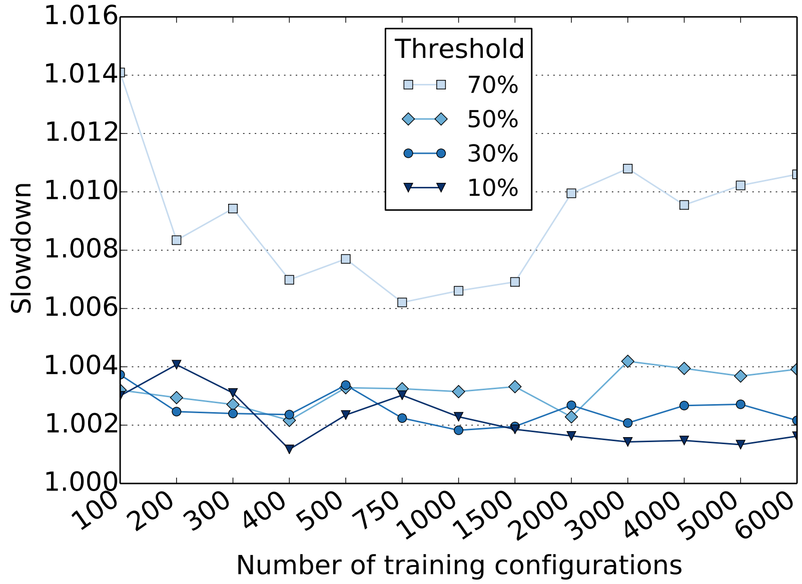

Figure 2: `bilateral`, Nvidia GTX980

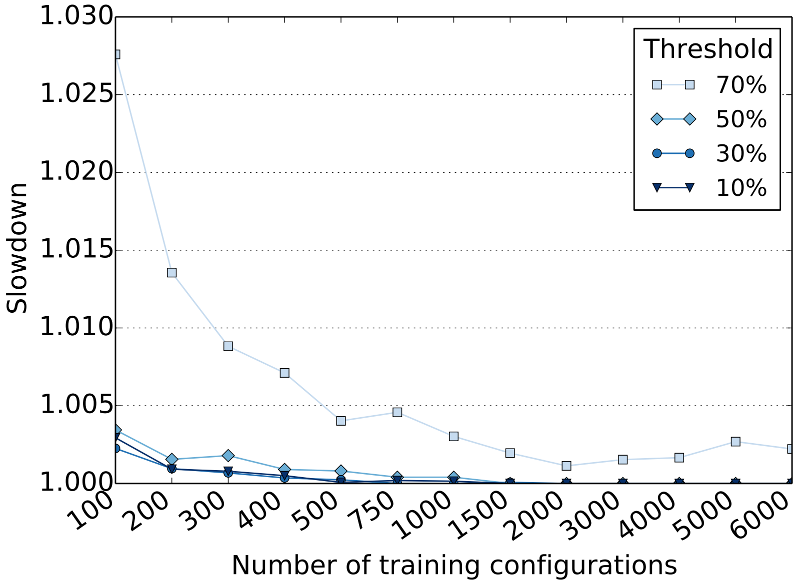

Figure 3: `bilateral`, Nvidia K40

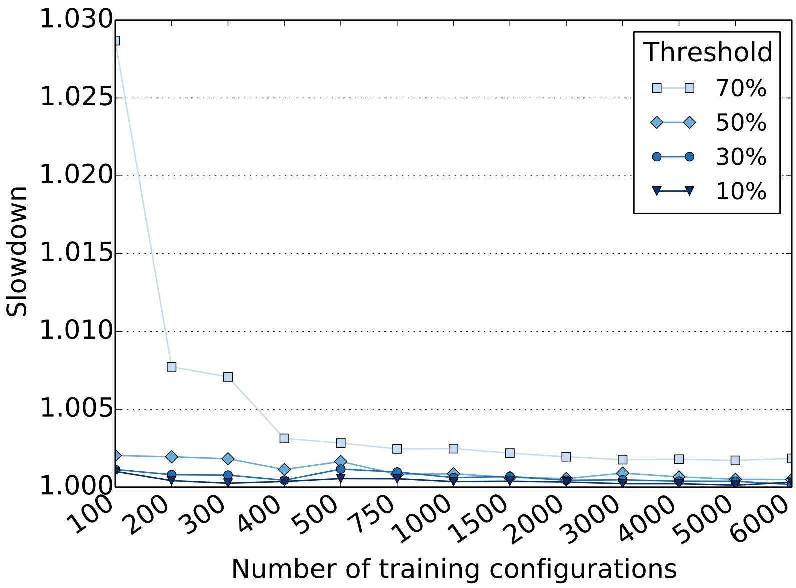

Figure 4: `bilateral`, Intel i7 3770

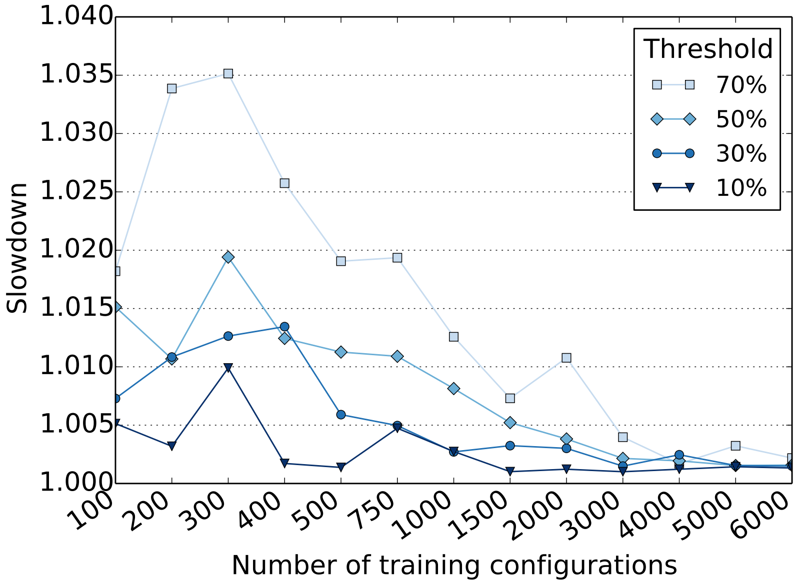

Figure 5: `median`, AMD 7970

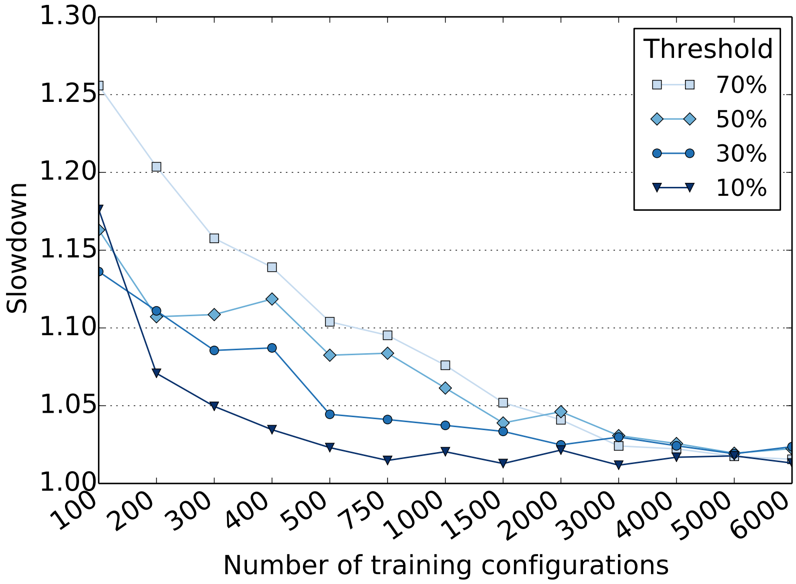

Figure 6: `median`, Nvidia K40

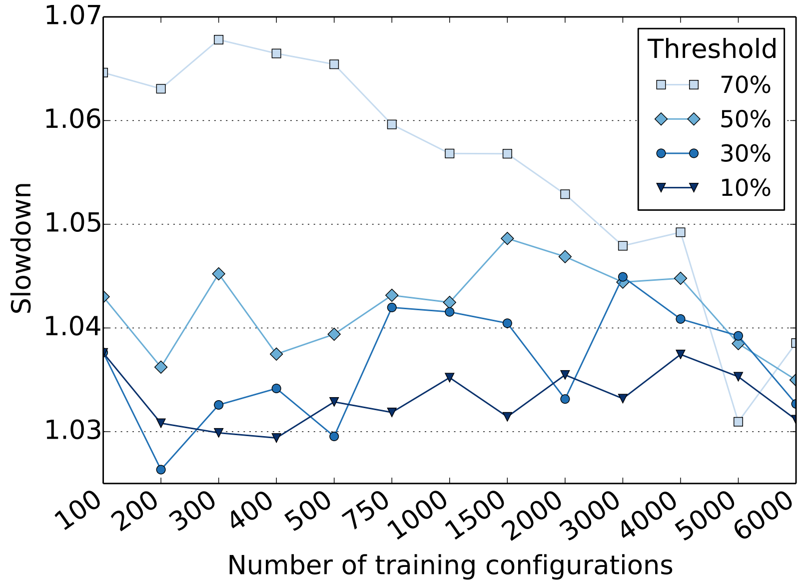

Figure 7: `median`, Intel i7 3770

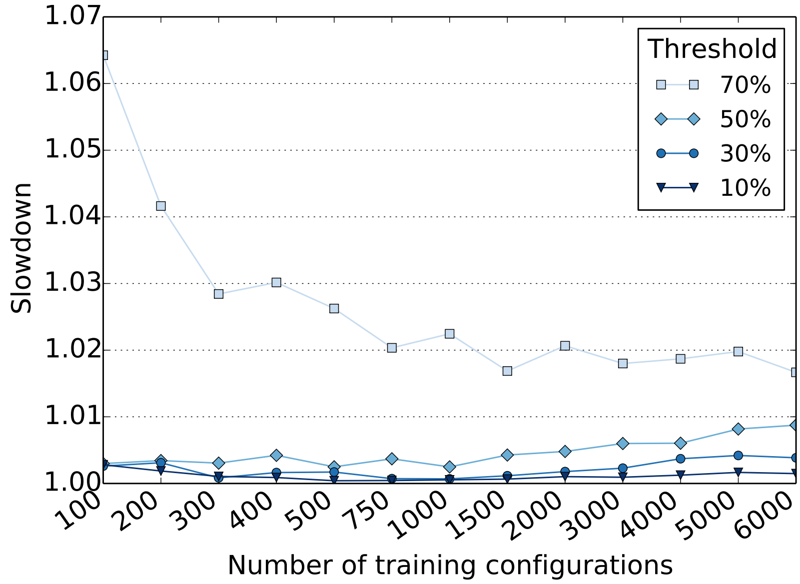

Figure 8: `stereo`, AMD 7970

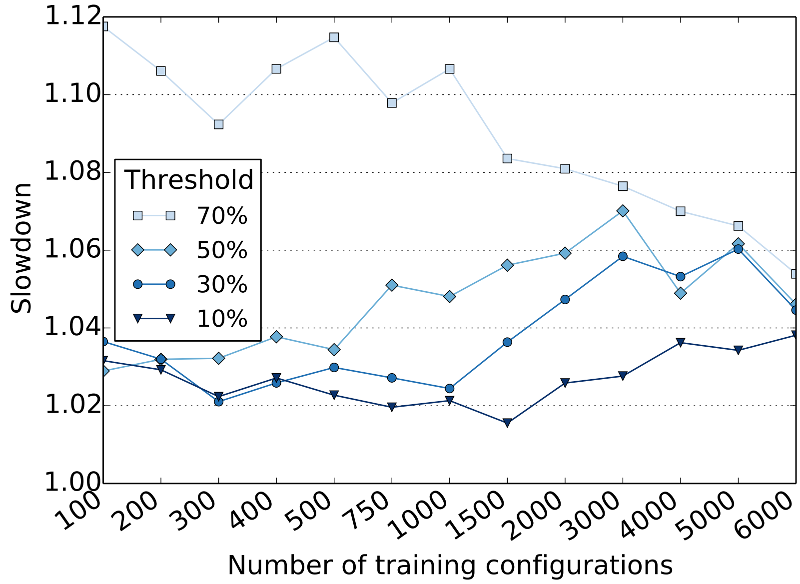

Figure 9: `stereo`, Nvidia K40

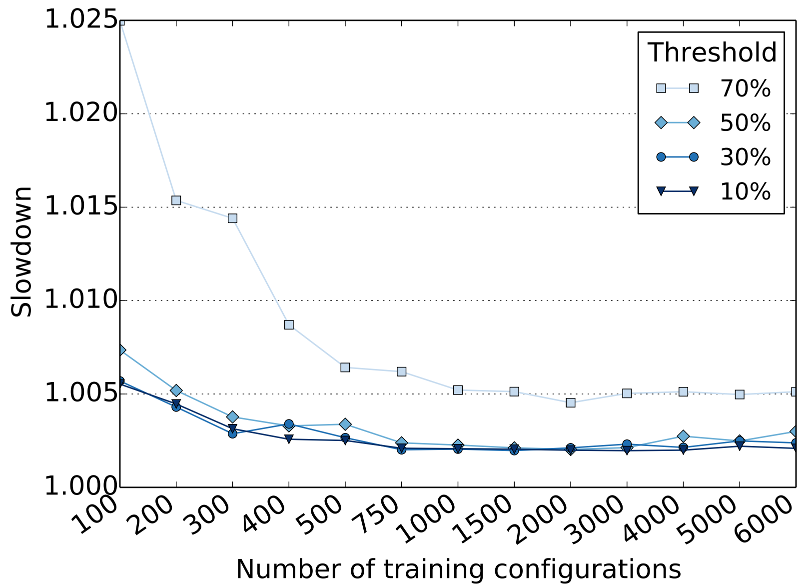

Figure 10: `stereo`, Intel i7 3770

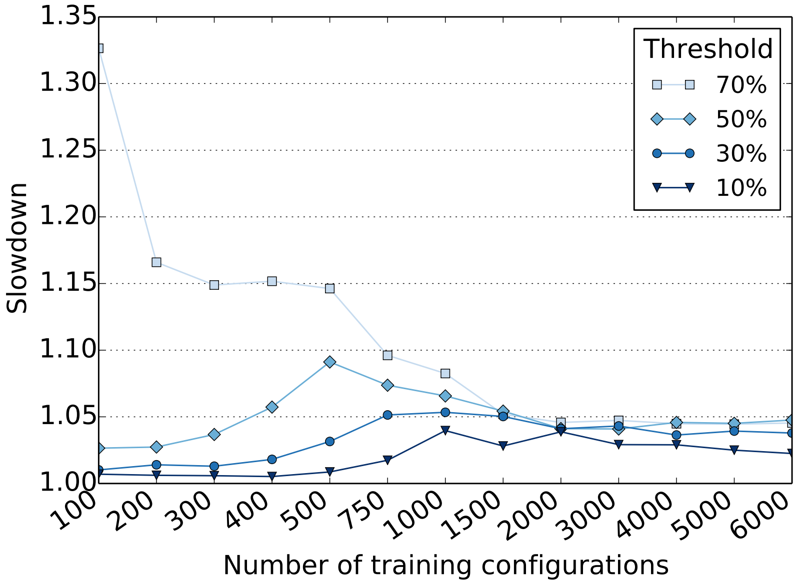

Figure 11: `raycast`, AMD 7970

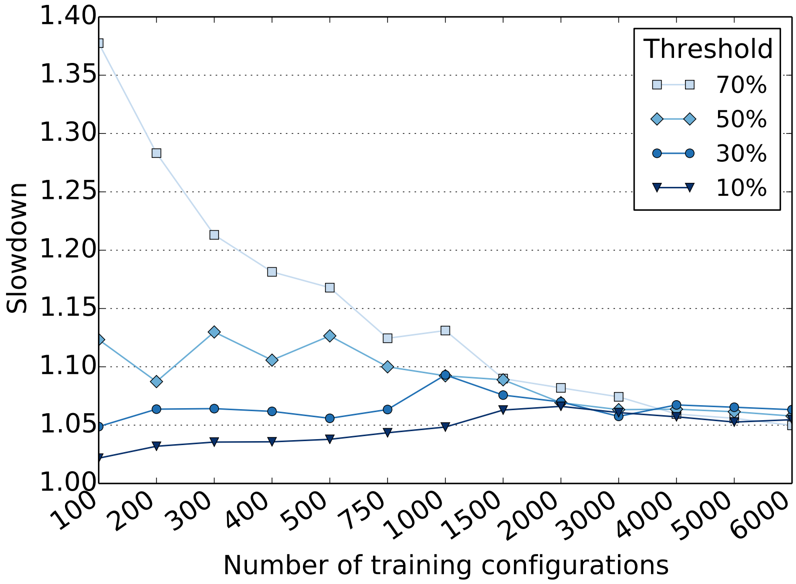

Figure 12: `raycast`, Nvidia K40

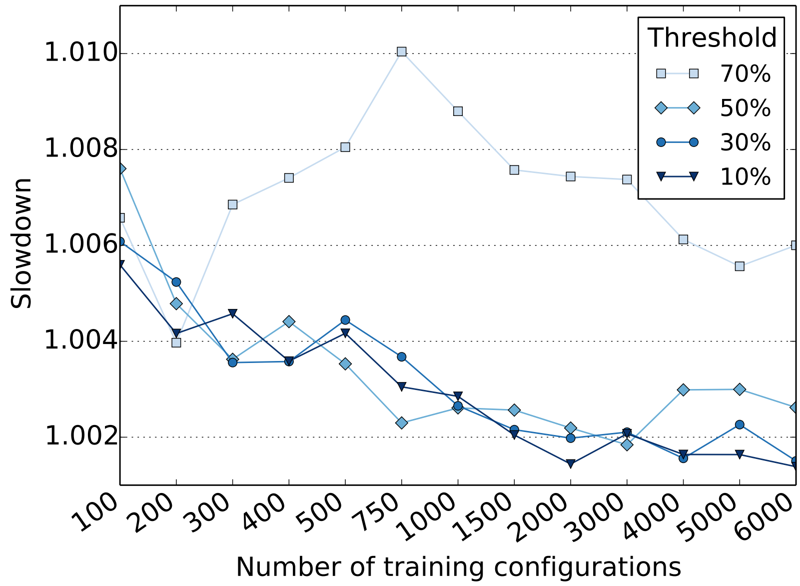

Figure 13: `raycast`, Intel i7 3770