

This table shows all results in the report. Use the column headers to sort the results in this report. Double-click a result to see detailed metrics. Double-click on demangled names to rename it.

ID	Estimated Speedup [%]	Function Name	Demangled Name	Duration [us] (2119.42 us)	Runtime Improvement [us] (1300.80 us)	Compute Throughput [%]	Memory Throughput [%]	# Reg
2	62.41	noiseReduction_ker...	noiseReduction_ker...	220.21	141.19	40.50	40.50	
3	50.50	gradient_kernel	gradient_kernel(flo...	447.14	225.81	43.86	49.50	
4	47.01	edge_kernel	edge_kernel..char..	178.05	83.69	68.12	68.12	

The following performance optimization opportunities were discovered for this result. Follow the rule links to see more context on the Details page.
Note: Speedup estimates provide upper bounds for the optimization potential of a kernel assuming its overall algorithmic structure is kept unchanged.

- [L1TEX Global Store Access Pattern](#)
Est. Speedup: 47.01%

The memory access pattern for global stores to L1TEX might not be optimal. On average, only 15.9 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the [Source Counters](#) section for uncoalesced global stores.
- [L1TEX Global Load Access Pattern](#)
Est. Speedup: 42.47%

The memory access pattern for global loads from L1TEX might not be optimal. On average, only 17.5 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the [Source Counters](#) section for uncoalesced global loads.
- [Thread Divergence](#)
Est. Speedup: 25.92%

Instructions are executed in warps, which are groups of 32 threads. Optimal instruction throughput is achieved if all 32 threads of a warp execute the same instruction. The chosen launch configuration, early thread completion, and divergent flow control can significantly lower the number of active threads in a warp per cycle. This workload achieves an average of 27.0 threads being active per cycle. This is further reduced to 19.8 threads per warp due to predication. The compiler may use predication to avoid an actual branch. Instead, all instructions are scheduled, but a per-thread condition code or predicate controls which threads execute the instructions. Try to avoid different execution paths within a warp when possible.