Bio:

Kuljit is a Sr. Distinguished Engineer at Nvidia working on memory sub-systems. Prior to joining Nvidia, he was a Fellow at Intel. Over the past couple of decades, he has overseen the design and specifications of many types of DRAMs including DDR3, DDR4, DDR5, LPDDR4, LPDDR5, LPDDR6, HBM3, and HBM4. His deep knowledge of DRAM architecture and memory sub-systems has made Kuljit influence DRAM standards across server, client, accelerator, and graphics market segments. Over his career, Kuljit has authored over 200 patents, some of which put forward the earliest ideas on how to improve DRAM security going as far back as 2012.

"Is PRAC a good solution to DRAM read disturbance? Are we missing anything? Can we (and should we) do much better (and hopefully not worse)?"

PRAC is major step forward to address bit flips due as a result of row hammer attack on DRAM. It is going to be an industry standard going forward with LPDDR6 and DDR6.

Some of the early work included adding a counter per wordline back in 2013 and recall filing a patent on it. Idea was rejected by memory vendors and now we have come a full circle by adding a counter per wordline with PRAC.

DRAM vendors added different options to address row hammer that were easily overcome. Some of the latest implementations such as DDR5 are more robust though but still not row hammer free.

We believe PRAC is a good solution but will create other attacks such as 'denial-of-service' or 'side channels' that need to be addressed

PRAC is essentially a framework for vendors to implement row hammer mitigation. A lot of papers assume a certain implementation and create attack vectors.

Some of the proposals also underestimate the die size impact for alternate implementations to PRAC. DRAM industry is very sensitive to cost overhead.

PRAC comes with minimal die size overhead. I believe that performance overhead of say 10% is a bit overstated in a lot of publications. Write row cycle times are not increased. Read row cycle goes up by ~20ns and there can be controller implementation to optimize around it. Future JEDEC spec will also try to address performance concerns.

If we have 2-3% perf overhead, then we can position it as a one-time tax of addressing row hammer. This is a tradeoff that industry is willing to make for security.

We can definitely improve on PRAC so all new ideas are welcome to make it more robust.