Recent publications indicate that traditional measures such as improving memory channels, clock speed, and bus width are no longer sufficient to meet growing data demands. I believe it is essential to explore novel techniques such as value prediction, prefetchers, and memory-centric computing, as well as HW/SW codesign. Additionally, I want to express that while these topics reflect my current interests, computer architecture is a broad field, and many other topics such as security, hardware accelerators, and interconnect technologies, are also exciting to me.

As we transition towards wider cores, data dependency stalls pose a more significant bottleneck. Value prediction (vp) can break true data dependencies and extract more performance from these wide cores with high ILP. Although vp's real-world impact has been limited due to accuracy issues and the heavy hardware and performance overhead required for prediction and recovery, I believe it holds potential for future applications. Current vp techniques draw inspiration from other hardware predictors, such as TAGE; I think this can be explored further, with approaches utilizing neural networks (perceptron), compiler-hints, etc.

Prefetchers are also vital for efficient memory management. As cores become wider and faster, the gap between computation and memory access continues to grow. Accurate prefetching and prefetcher filtering are essential as we can no longer afford to wait for cache misses. I believe there is still work to be done on this topic, and new innovations are always being published, such as a recent paper on a two level prefetcher enabled by perceptrons.

Memory-centric computing (mcc) is another emerging concept that seeks to shift the paradigm; by establishing systems which offload processing to near or in memory devices, or devices where data is being generated (sensors). This new paradigm encompasses many novel concepts such as in memory computing (imc) or near memory computing (nmc). Several publications have demonstrated various imc techniques such as data copying, bitwise operations, and even matrix multiplication operations, by exploiting the intrinsic properties of existing memory technologies (SRAM, DRAM, NAND Flash, etc.). Similarly, recent publications have also explored and highlighted the benefits of nmc; that is, by including processors and or accelerators onboard memory devices, data movement is significantly reduced, thereby improving both performance and efficiency. Mcc is still a nascent field, and I believe there is much exciting progress to be made on this topic. Integration remains a key component of this topic, and requires overcoming major challenges within the current memory technology space. Equally important are the target applications, which heavily impact the predicted performance gains netted from mcc techniques.

Lastly, I believe HW/SW codesign, more specifically compiler/uarch interactions, are particularly crucial for more performant systems. The layer of abstraction that is traditionally kept between hardware and software limits the abilities of both sides, leaving performance gains on the table. Bridging this gap can provide hardware with information it can not find otherwise, enabling more robust and optimized hardware techniques, such as better prediction mechanisms. The reverse is also true: recent publications have indicated promising performance gains for hardware assisted compilation in specific domains, such as quantum computing.