Due to the significant number of changes with respect to the previous two submissions, ASPLOS'18 and ASPLOS'19, we decided to summarize the most significant modifications in text. We introduced the aforesaid changes to address the main concerns of the reviewers. Therefore, we also thank all the past reviewers for helping us making the current submission significantly stronger.

ASPLOS 2018. The main concerns of all three reviewers is the lack of a prototype in an existing OS that would demonstrate the feasibility of our proposal. In contrast, in our current submission, we demonstrate that our technique is readily implementable and compatible with commodity OSes by prototyping DTRIM in stock Linux. The OS implementation details appear in Section 5 (OS Support), whereas the experiments on a real system appear in Section 7 (Evaluation).

ASPLOS 2019. Each reviewer has a different concern. The first reviewer is not clear on why the accelerators in our baseline do not translate through an IOMMU. We included an IOMMU in the system overview presented in Figure 1, along with an explanation in Sections 1 and 4 as to how our design relates to IOMMU and why an IOMMU is always a suboptimal design with respect to our baseline. The second reviewer questions whether DTRIM increases the latency of TLB hits and whether thread pollution in the memory-side TLBs dwarfs the benefits of higher TLB miss rates. For the former concern, we included the TLB-hit scenario for both baseline and DTRIM in Figure 3, along with its associated explanation in Section 4, demonstrating that both techniques deliver the same latency. For the latter concern, we added Section 7.4 (Thread Contention) along with Figure 8, to demonstrate that thread contention is minimal, and the benefit of memory partitioning greatly offsets any thread-induced TLB contention. The last and third reviewer is has questions regarding our virtual-to-physical addressing methods and raises concerns about our implementation of remapping virtual addresses. Our Section 5 (OS Support) explains in more detail our addressing methods, following the reviewer's suggestion, and further explains how remap is correctly supported in detail.