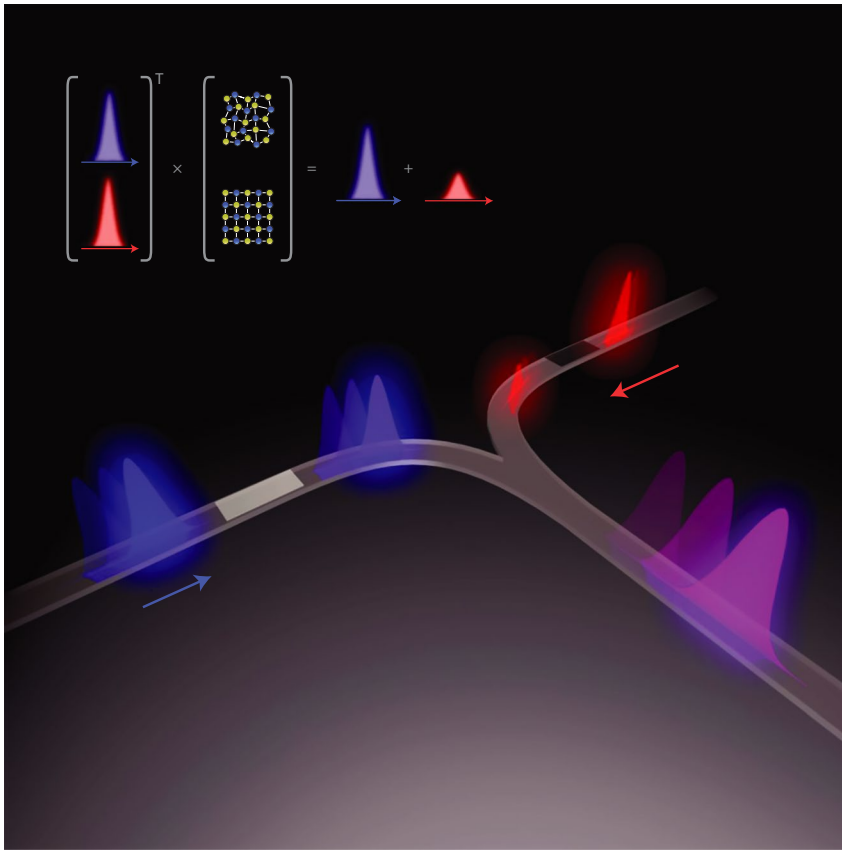


## PHASE-CHANGE MATERIALS

## Photonic devices compute in memory

*Sci. Adv.* **5**, eaau5759 (2019)



Credit: Carlos Ríos, Nathan Youngblood, Oxford University

In-memory computing — an approach that uses memory devices for both data processing and storage — could overcome the limitations of traditional von Neumann computing architectures. The approach has been shown to be capable of a range of tasks including matrix operations and the detection of temporal correlation from data sets. All-photonic devices that can function as nonvolatile memories also exist, but their implementation within a processing architecture has so far proved challenging. Abu Sebastian, Harish Bhaskaran and colleagues have now shown that photonics-based memory devices can also be used for computational tasks.

The researchers — who are based at the University of Oxford, IBM Research–Zurich, the University of Muenster and the University of Exeter — created memory cells from a phase-change material ( $\text{Ge}_2\text{Sb}_2\text{Te}_3$ ) and photonic waveguides, and used them to carry out the direct scalar multiplication of two numbers. For this operation, the

first number is mapped onto a laser pulse and the second number is mapped onto the transmittance of the device, which can be set by another laser pulse. The input pulse propagates through the device and its amplitude is modulated according to the set transmittance. The energy of the input pulse, which is equivalent to the multiplication result, can then be probed optically at the output. By using memory cells in parallel, and mapping the input numbers sequentially, the researchers also show the approach can be used for matrix–vector multiplication. In contrast to electronics-based implementations, the approach of Bhaskaran and colleagues does not require any post-processing due to the fact that the relationship between the input and the output pulses is linear.

Christiana Varnava

Published online: 15 March 2019  
<https://doi.org/10.1038/s41928-019-0226-1>