Conventional solution and proof

Here A has shape 1024×65536, B has shape 65536×32, and C has shape 32×65536. Both products have shape 1024×32 and use 2^{31} classical scalar multiplications. If B and C are independently generated, the outputs differ, although the operation counts agree. Performance can differ because of memory order and packing. If $C = B^{\mathsf{T}}$ is a view, then $C^{\mathsf{T}} = B$ as an indexed array and both expressions use the same logical right factor. Double transposition restores the original strides. A view shares storage and needs only metadata; cloning creates another 2^{21}-entry array, or 16 MiB in double precision. A modification through the view also changes the shared base array.

Structural or analytic derivation

The index permutation viewpoint separates algebraic equality from physical layout. The contraction indices are identical in the two expressions; no spectral or complex transformation changes their mathematical operation count. A compiler or BLAS library may exploit transpose flags without materializing a copy.

Comparative cost and preferred method

The two methods are tied at 2^{31} multiplications. View construction is $O(1)$ metadata, compared with $\Theta(2^{21})$ elements copied when cloning. Wall-clock differences are implementation-dependent and are measured separately on a scaled instance. No universal faster product can be proved from the shapes alone.