

# Cache Compression in Two Dimensions

Amin Ghasemazar

Mohammad Ewais

Prashant Nair

Mieszko Lis

University of British Columbia

## INTRODUCTION

### PROBLEM

cache densities and sizes have not kept up with increasing cache demands of applications

### EXISTING SOLUTIONS

transparent, in-hardware cache compression

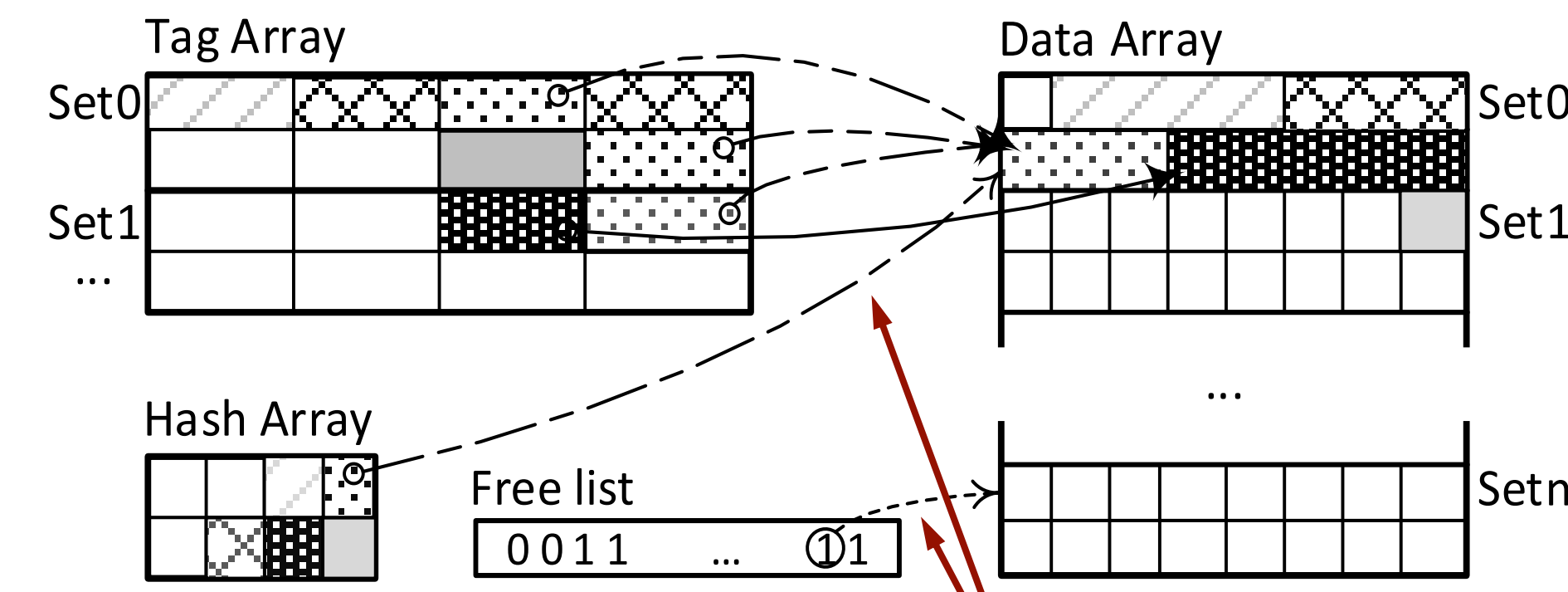
- *inter-block*: deduplicating identical blocks across the cache (e.g., [1])
- *intra-block*: compressing common patterns within each block (e.g., [2])

### LIMITATIONS

compression in only one dimension (either inter-block or intra-block)

[1] Pekhimenko et al. "Base-delta-immediate compression." PACT 2012.  
[2] Tian et al. "Last-level Cache Deduplication." 2014.

## ARCHITECTURE: STORAGE STRUCTURES



### data array

- variable-sized blocks to allow compression
- evicts blocks with fewest duplicates

### tag array

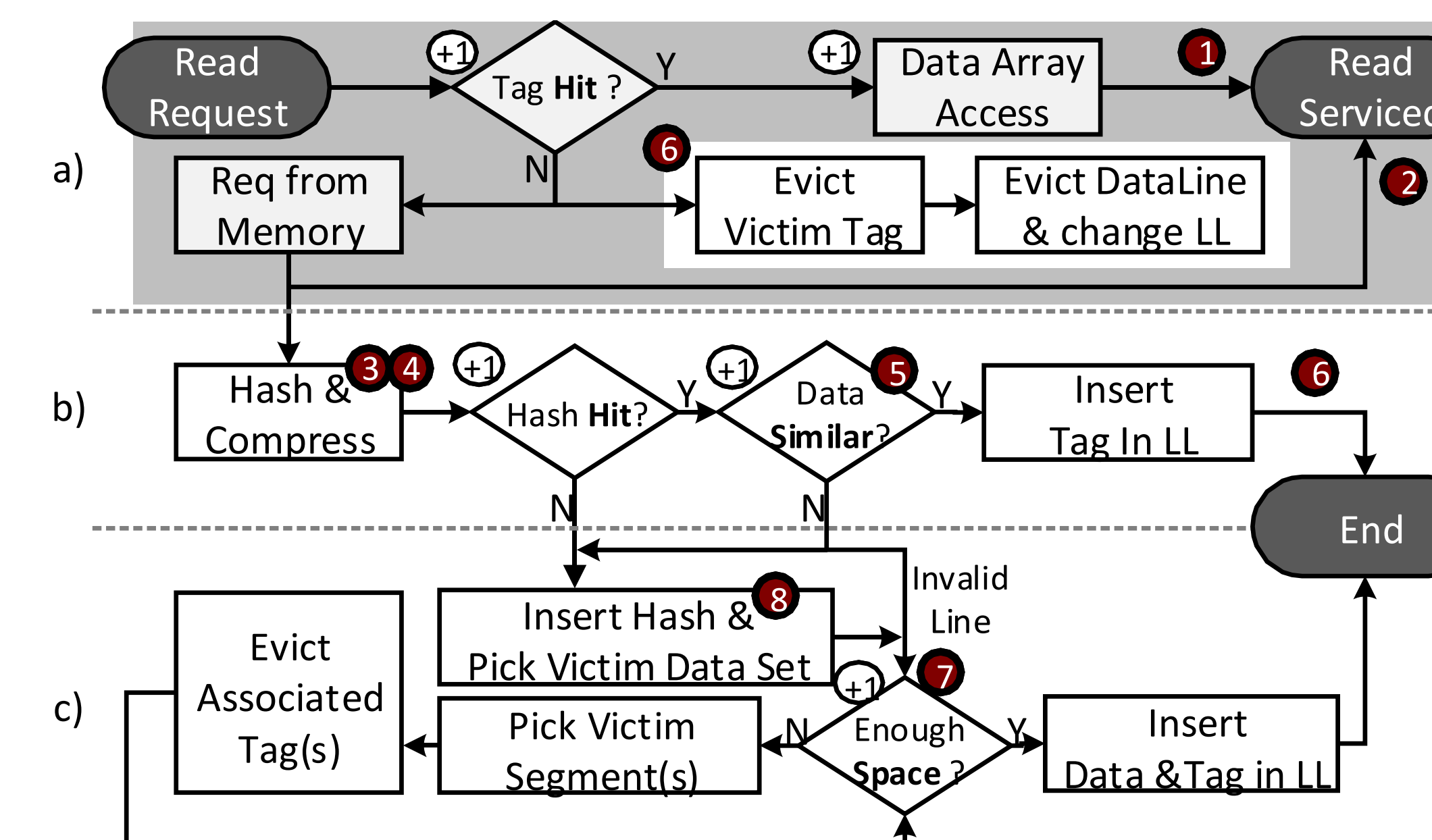
- 4x capacity vs standard to allow compression
- evicts least recently used tags

### hash array

- helps detect duplicate blocks
- evicts least recently seen block hashes

## ARCHITECTURE: OPERATION MECHANISM

- **reads, evictions, and insertions**: the critical-path similar to a conventional cache
- **writes**: execute off the critical path, may change the compression factor (steps 3–8)
- deduplication / compression overheads are incurred off the critical path



on the critical path (shaded):

- (1) hit response
- (2) miss response

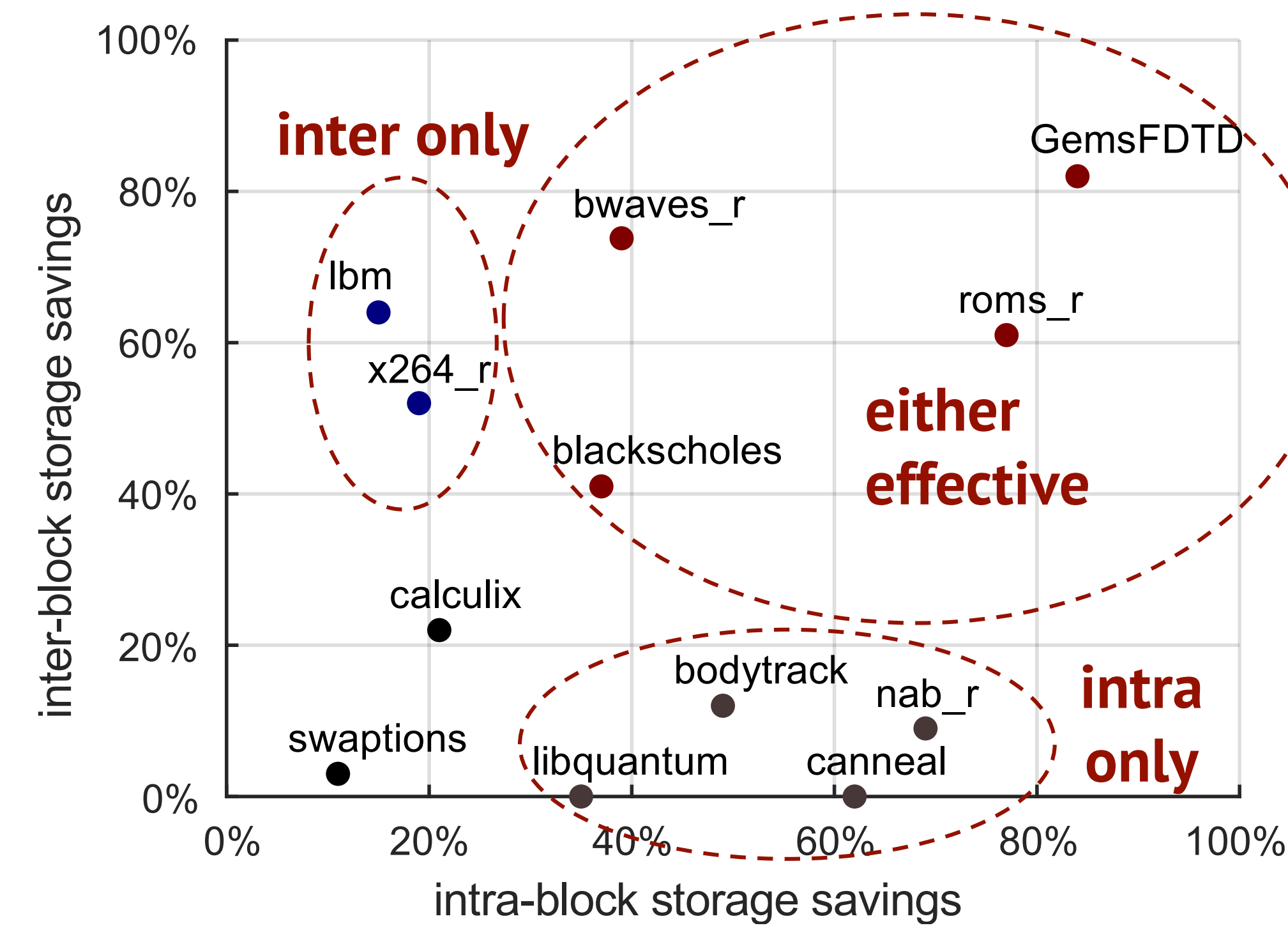
off the critical path:

- (3) hash calculation
- (4) block compression
- (5) duplicate block detection
- (6) tag entry insertion/replacement
- (7) data entry insertion/replacement
- (8) hash entry insertion (unique blocks)

## OPPORTUNITIES

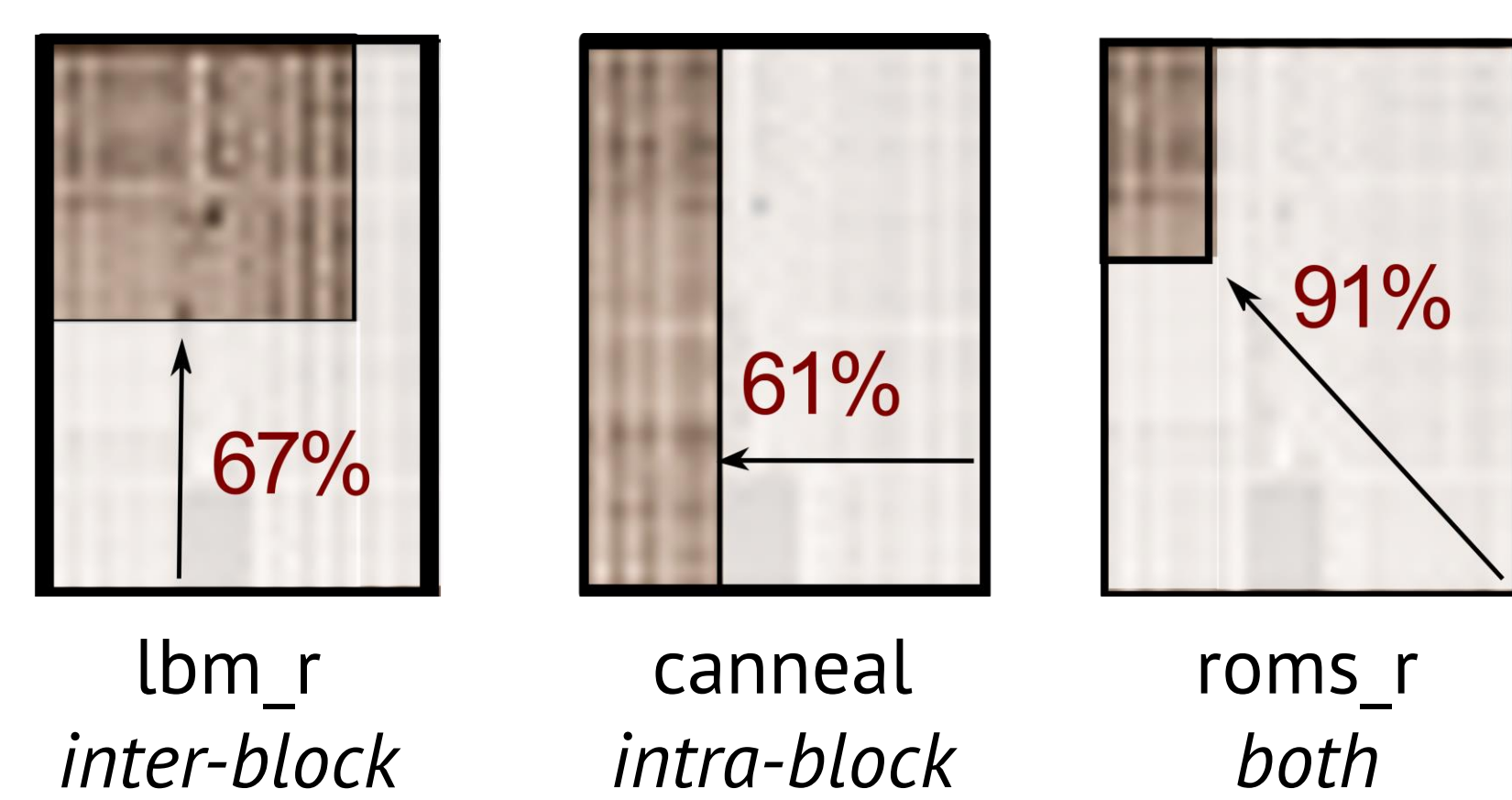
### ACROSS WORKLOADS: CHOOSE INTER OR INTRA

many workloads are compressible using *only* inter-block or *only* intra-block methods:



### WITHIN WORKLOADS: NEED BOTH METHODS

many workloads compress best with *both*:



## RESULTS

