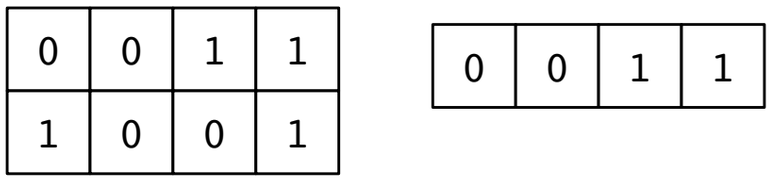
# Riptide: Techniques for Fast End-to-End Binarized Networks

Josh Fromm

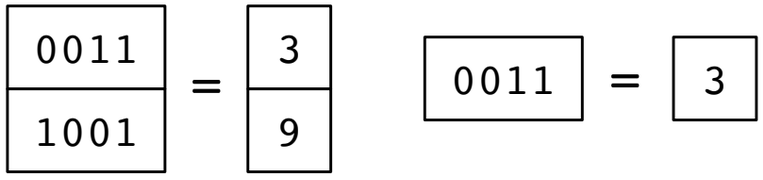
# Binary Networks

Original Data
uint32


Bitplanes
uint1




Bitpacked Data
uint4




Bitserial Dot Product

$$1\times \text{popcount}(3\&3) + 2\times \text{popcount}(3\&9) = 4$$

# Binary Networks

Original Data
uint32

| 2 | 0 | 1 | 3 |
|---|---|---|---|

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitplanes
uint1

| 0 | 0 | 1 | 1 |
|---|---|---|---|
| 1 | 0 | 0 | 1 |

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitpacked Data
uint4

| 0011 |
|---|
| 1001 |

=

| 3 |
|---|
| 9 |

| 0011 |
|---|

=

| 3 |
|---|

Bitserial Dot Product

$$1\times \text{popcount}(3\&3) + 2\times \text{popcount}(3\&9) = 4$$

- **Replaces 32-bit values with 1 or 2 bits**

# Binary Networks

Original Data
uint32

| 2 | 0 | 1 | 3 |
|---|---|---|---|

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitplanes
uint1

| 0 | 0 | 1 | 1 |
|---|---|---|---|
| 1 | 0 | 0 | 1 |

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitpacked Data
uint4

| 0011 |
|---|
| 1001 |

=

| 3 |
|---|
| 9 |

| 0011 |
|---|

=

| 3 |
|---|

Bitserial Dot Product

$$1\times \text{popcount}(3\&3) + 2\times \text{popcount}(3\&9) = 4$$

- **Replaces 32-bit values with 1 or 2 bits**
- **Up to 32X compression**

# Binary Networks

Original Data
uint32

| 2 | 0 | 1 | 3 |
|---|---|---|---|

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitplanes
uint1

| 0 | 0 | 1 | 1 |
|---|---|---|---|
| 1 | 0 | 0 | 1 |

| 0 | 0 | 1 | 1 |
|---|---|---|---|

Bitpacked Data
uint4

| 0011 |
|---|
| 1001 |

=

| 3 |
|---|
| 9 |

| 0011 |
|---|

=

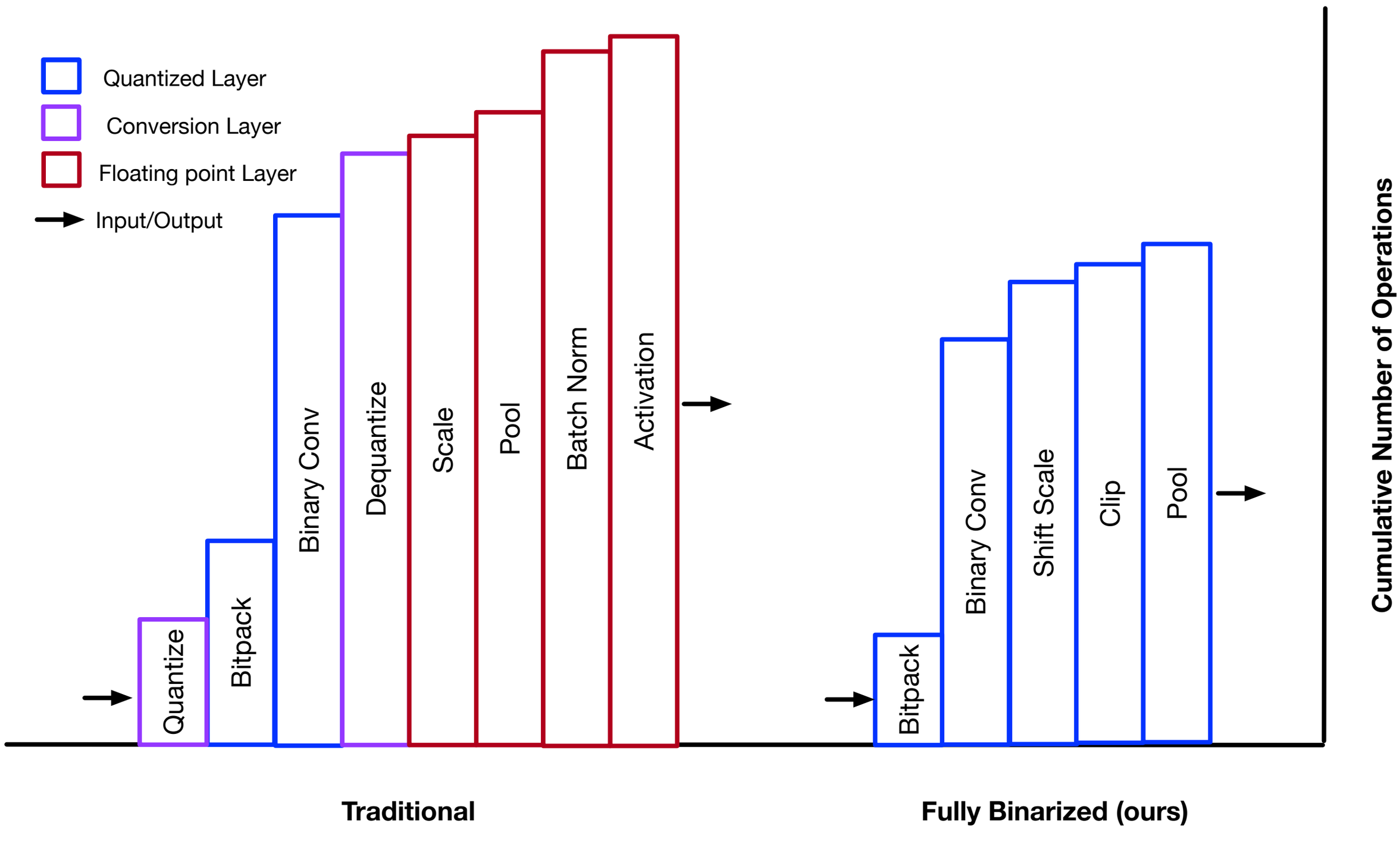
| 3 |
|---|

Bitserial Dot Product

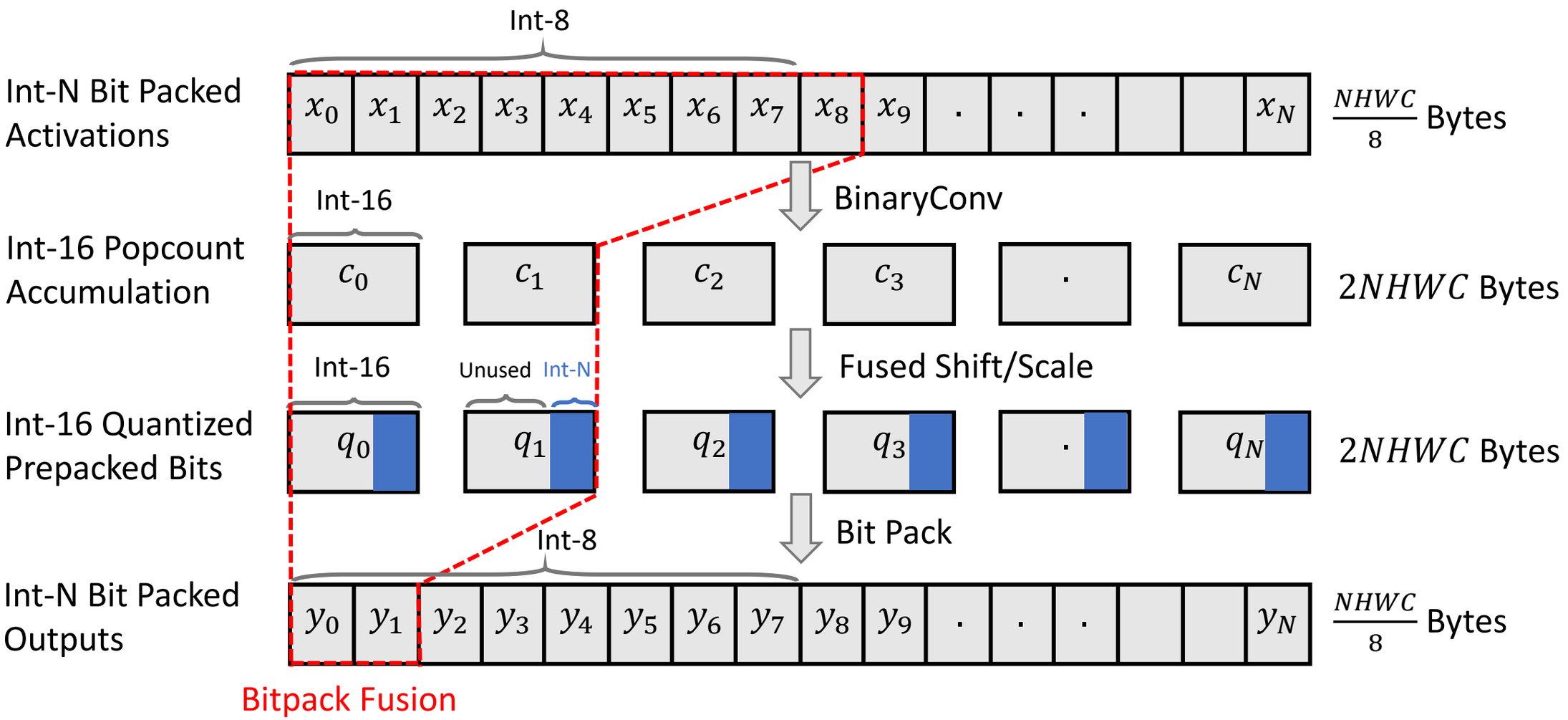
$$1\times \text{popcount}(3\&3) + 2\times \text{popcount}(3\&9) = 4$$

- **Replaces 32-bit values with 1 or 2 bits**
- **Up to 32X compression**
- **Up to 48X (hypothetical) speedup**

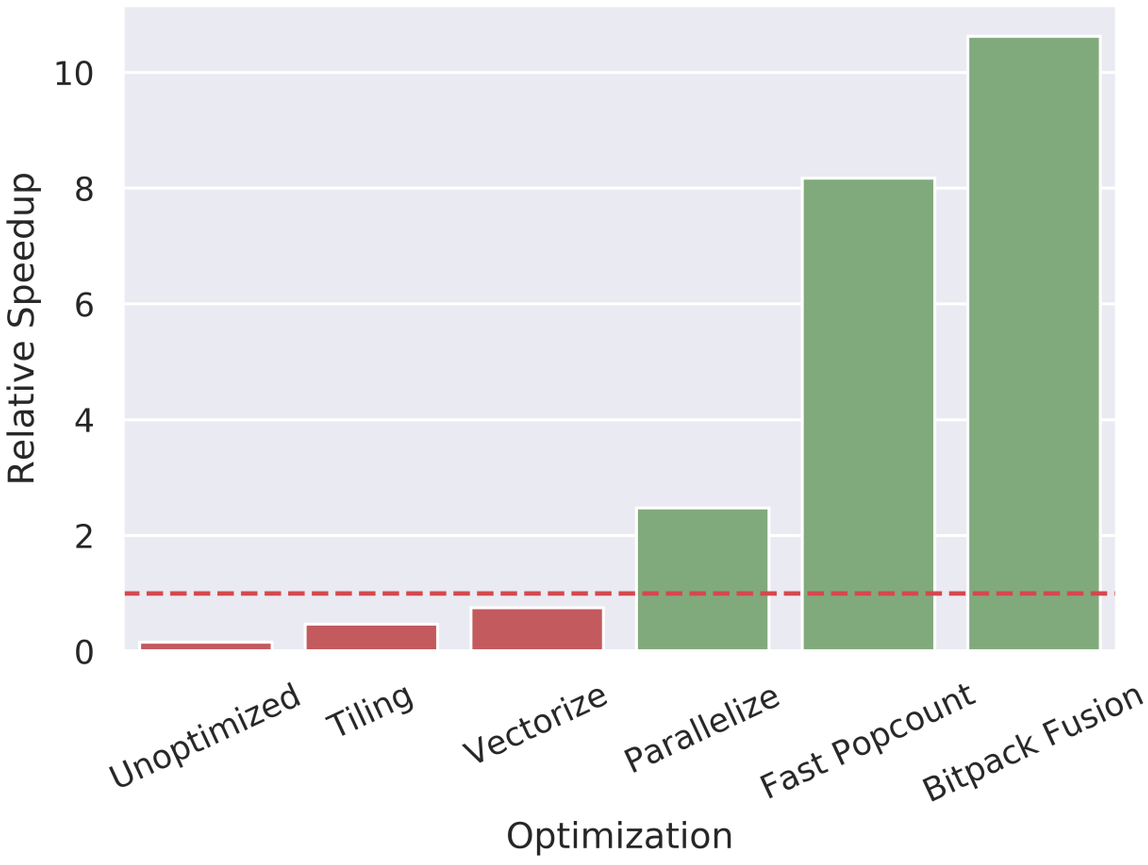
# Fused Glue

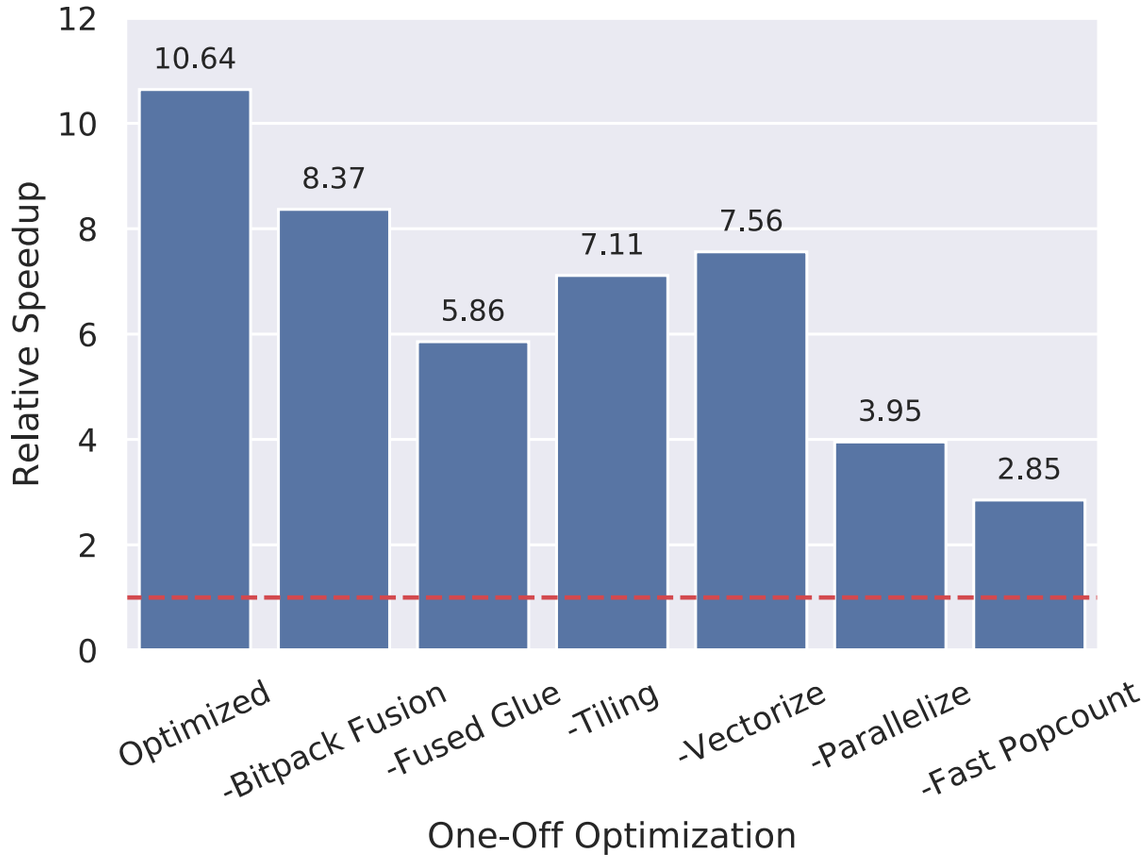

Quantized Layer
Conversion Layer
Floating point Layer
Input/Output
Quantize
Bitpack
Binary Conv
Dequantize
Scale
Pool
Batch Norm
Activation
Traditional
Bitpack
Binary Conv
Shift Scale
Clip
Pool
Fully Binarized (ours)
Cumulative Number of Operations

# Bitpack Fusion

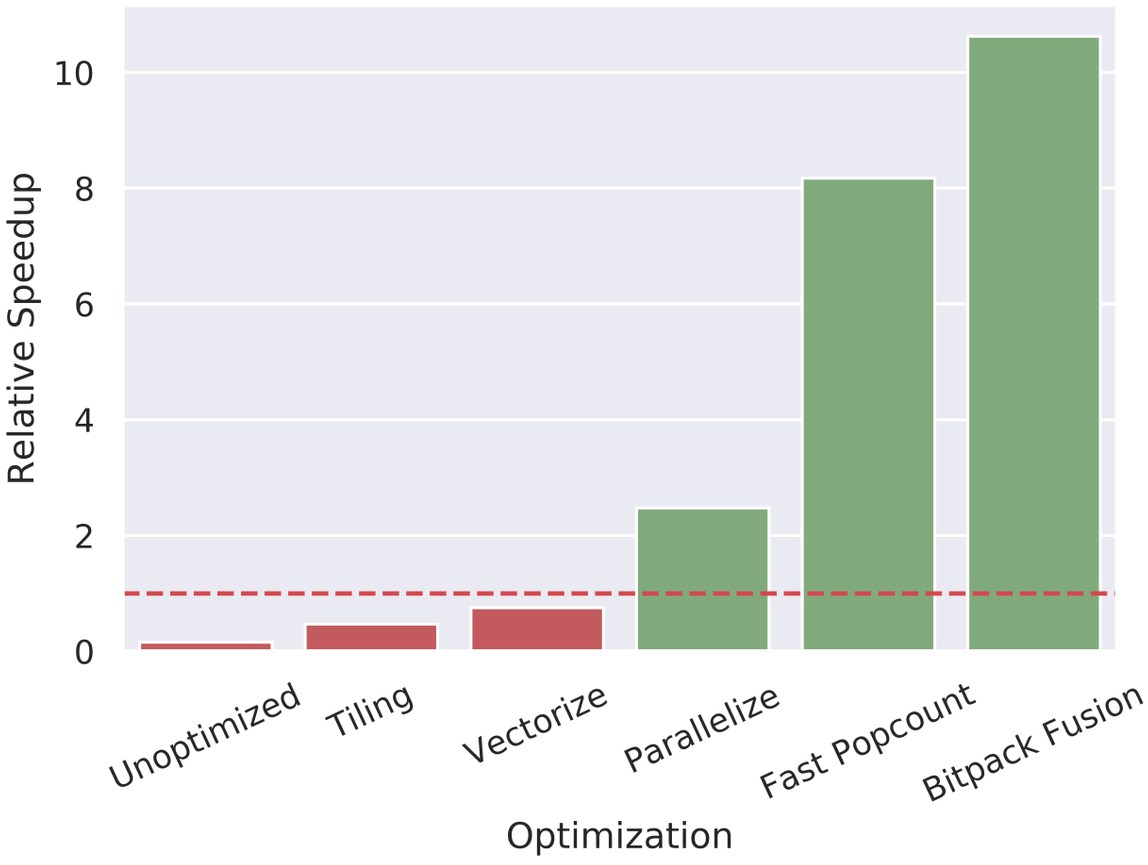


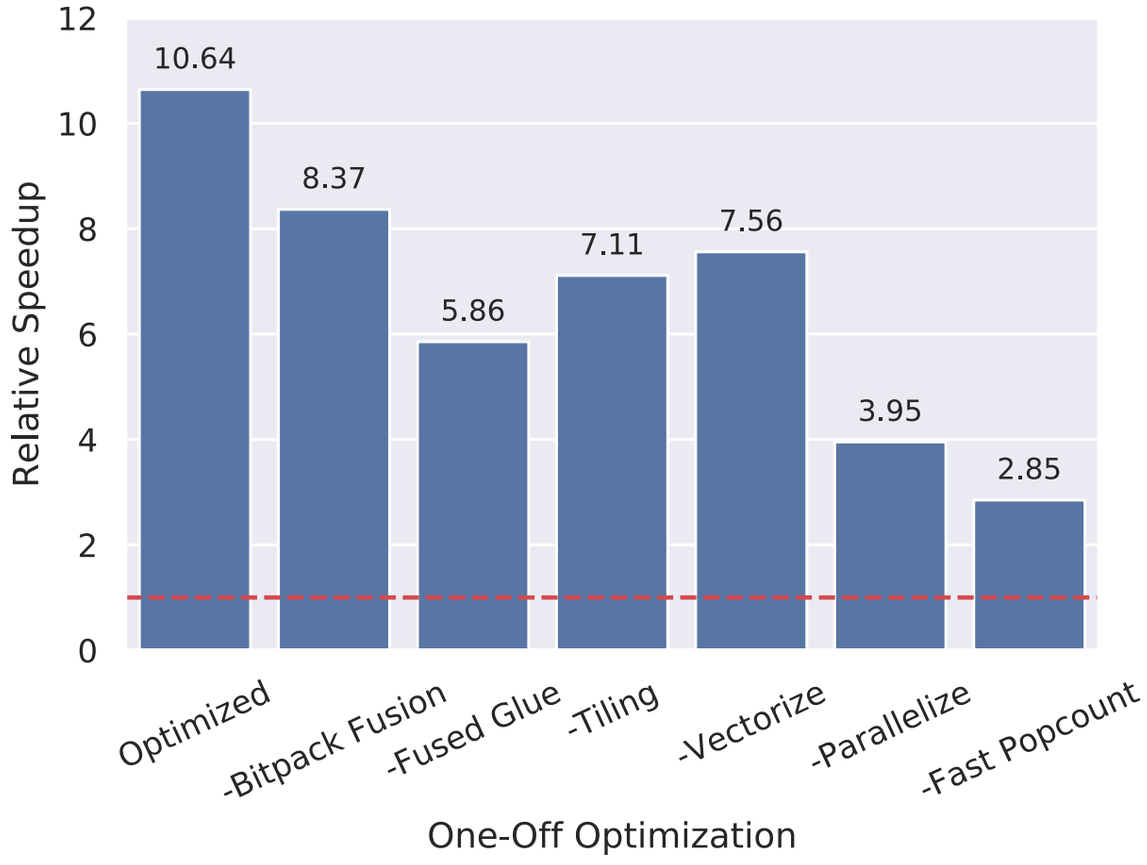
# SqueezeNet Speedups



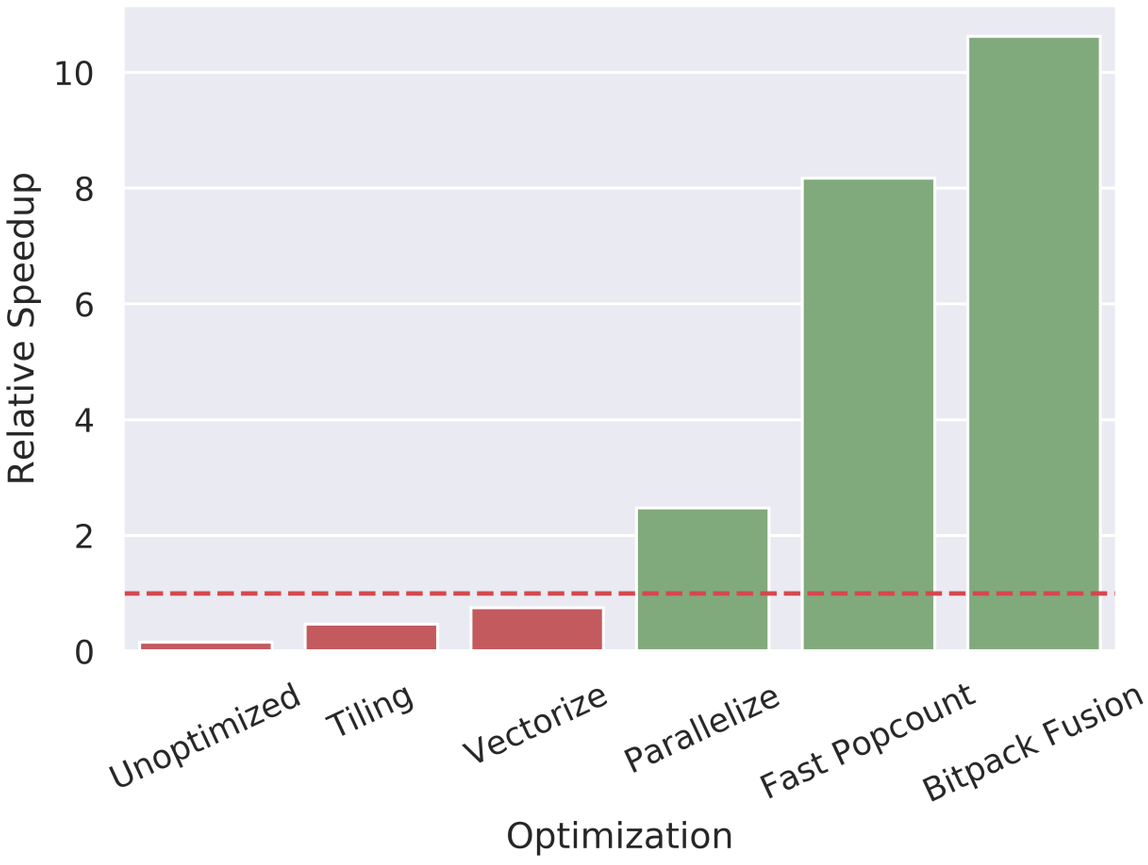
# SqueezeNet Speedups

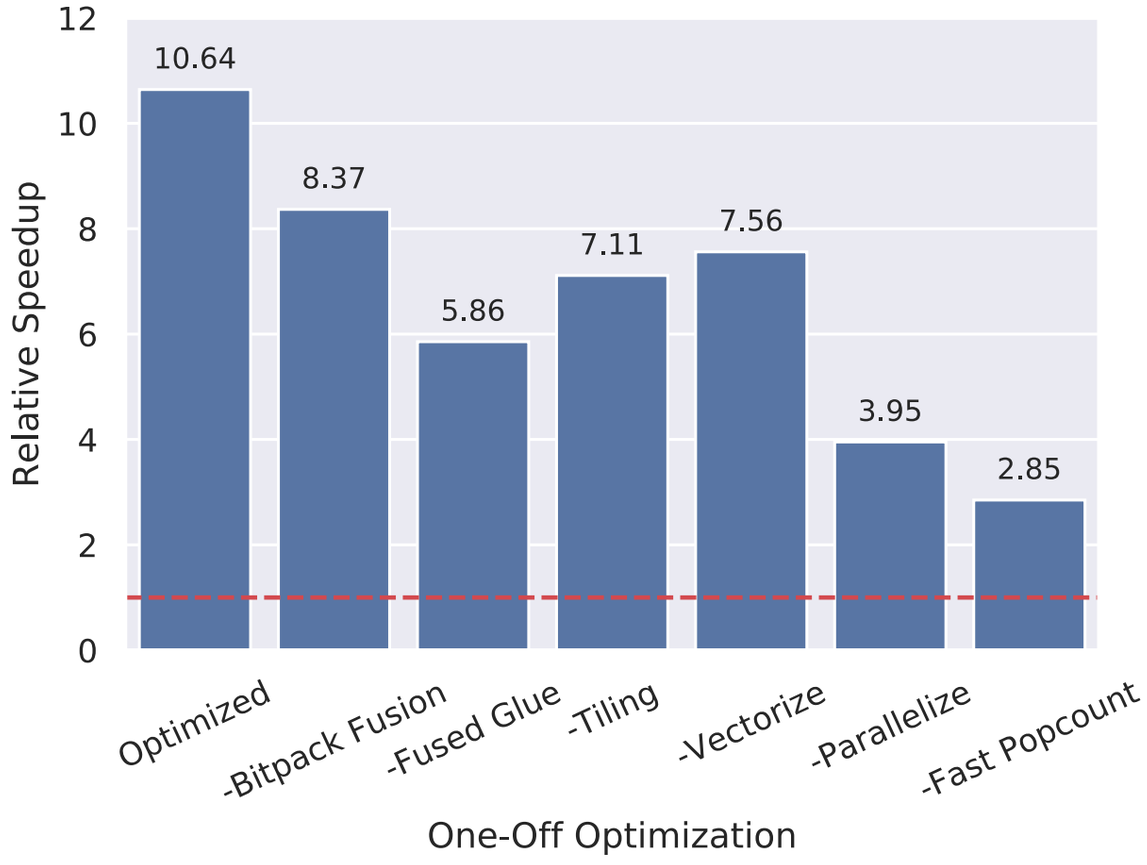



- **Up to 10X speedup**

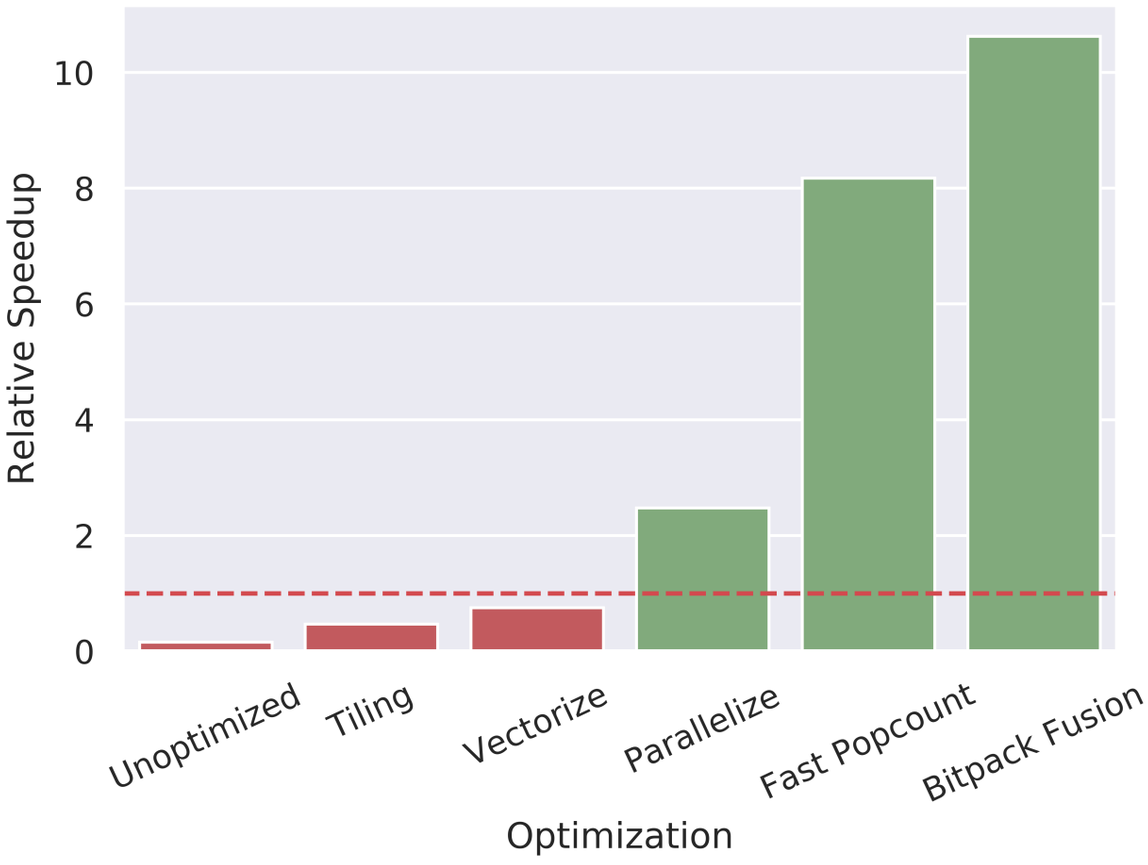
# SqueezeNet Speedups



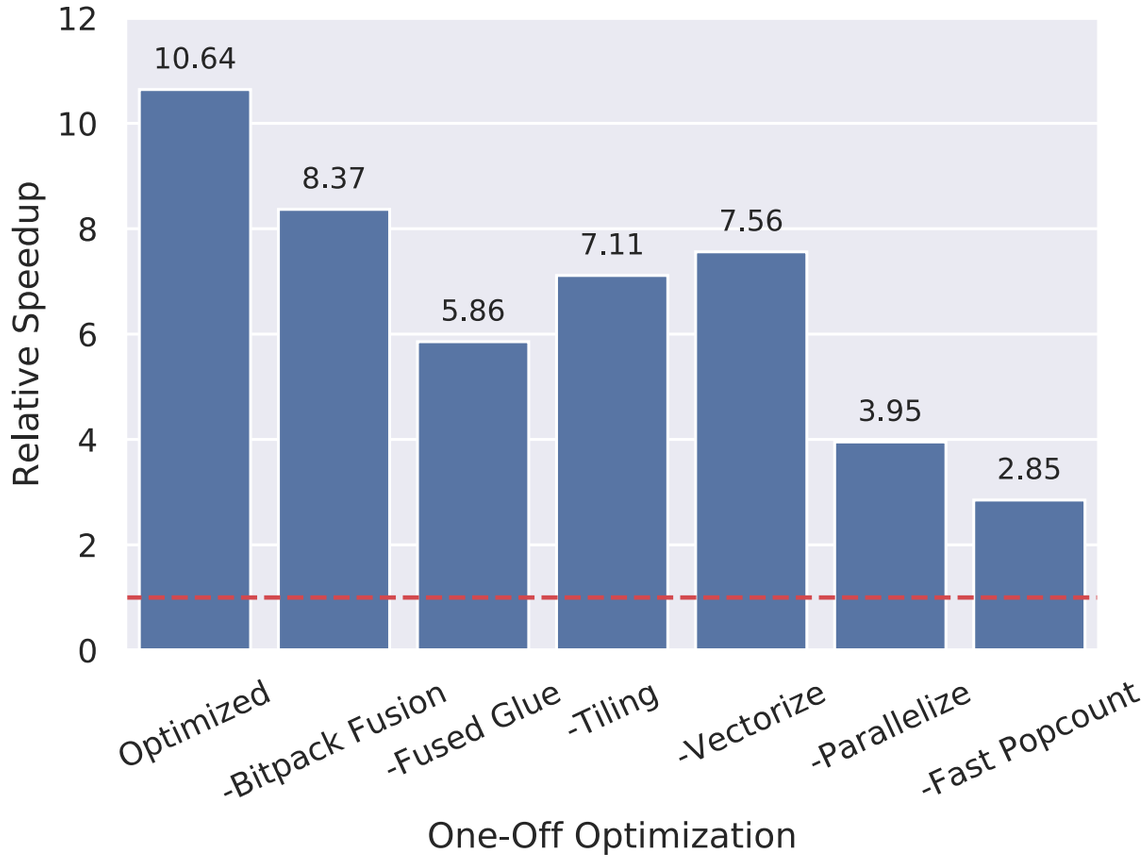




- **Up to 10X speedup**
- **Memory savings from fused schedule yield 1.3X speedup**

# SqueezeNet Speedups

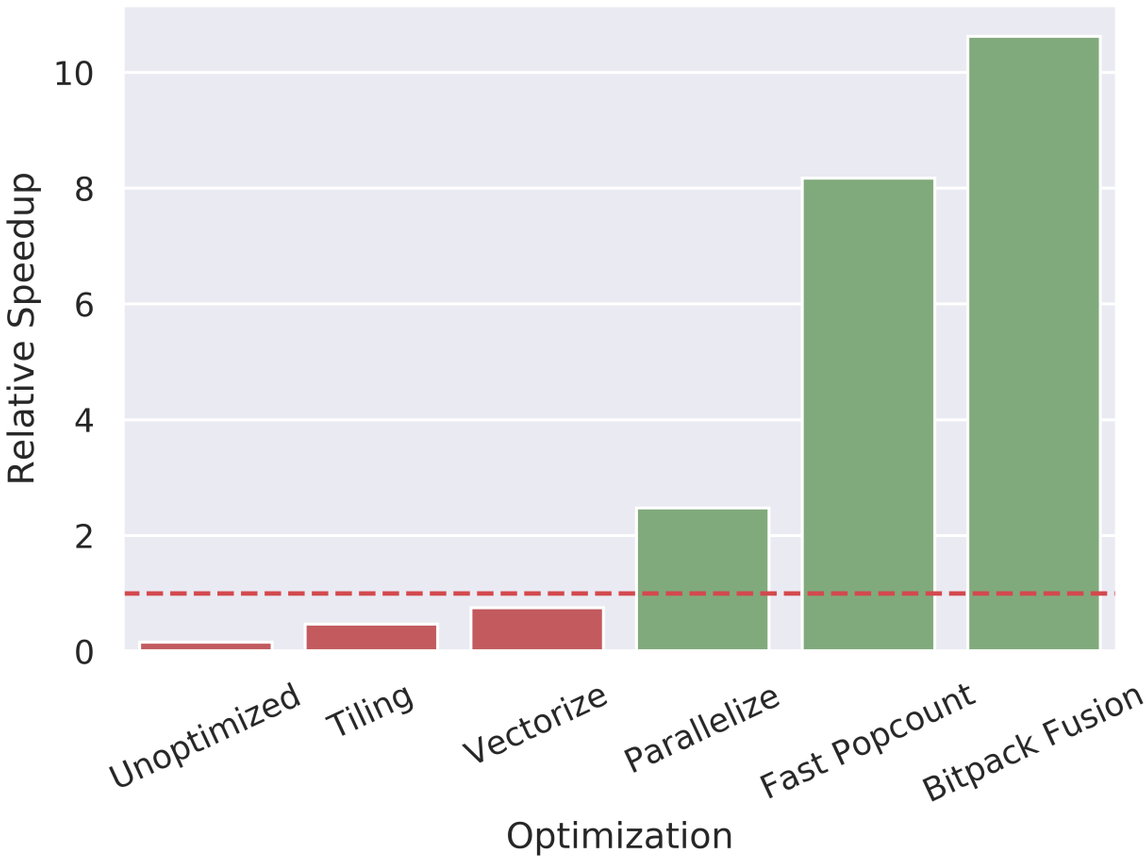



- **Up to 10X speedup**
- **Memory savings from fused schedule yield 1.3X speedup**

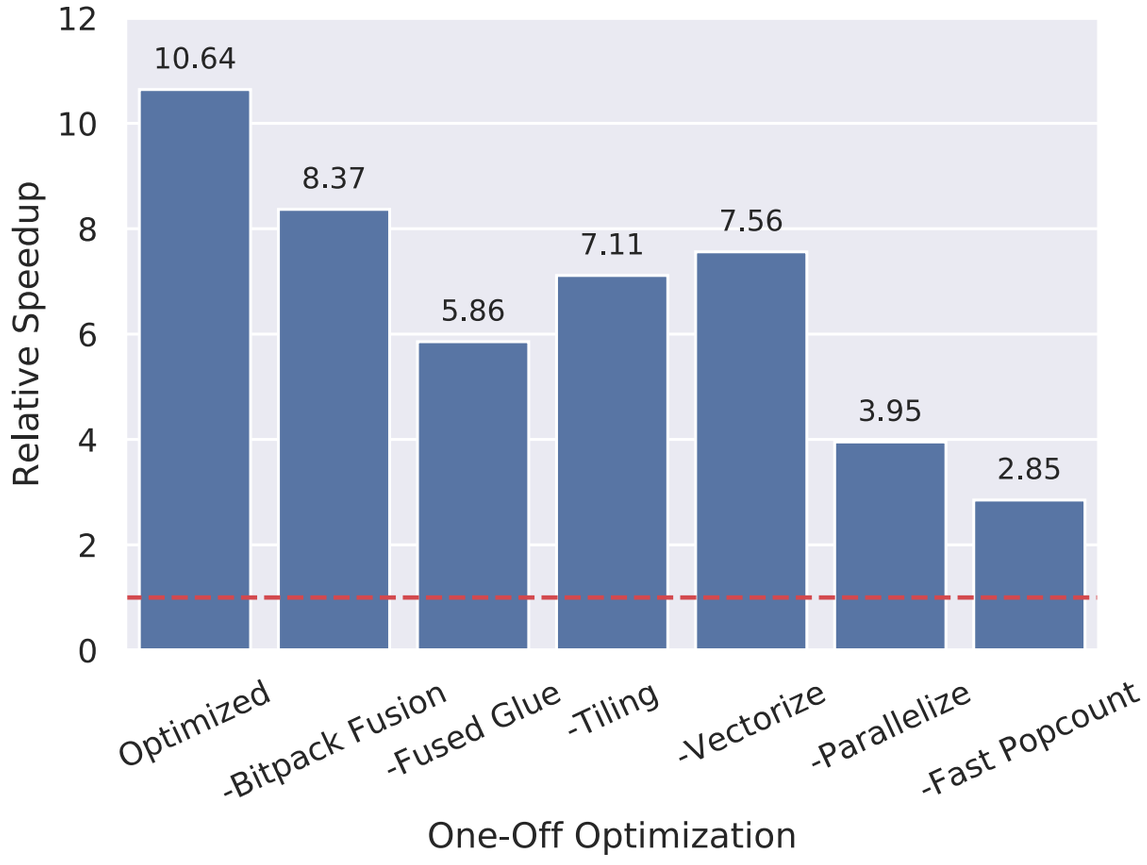



- **All optimizations have significant impact**

# SqueezeNet Speedups

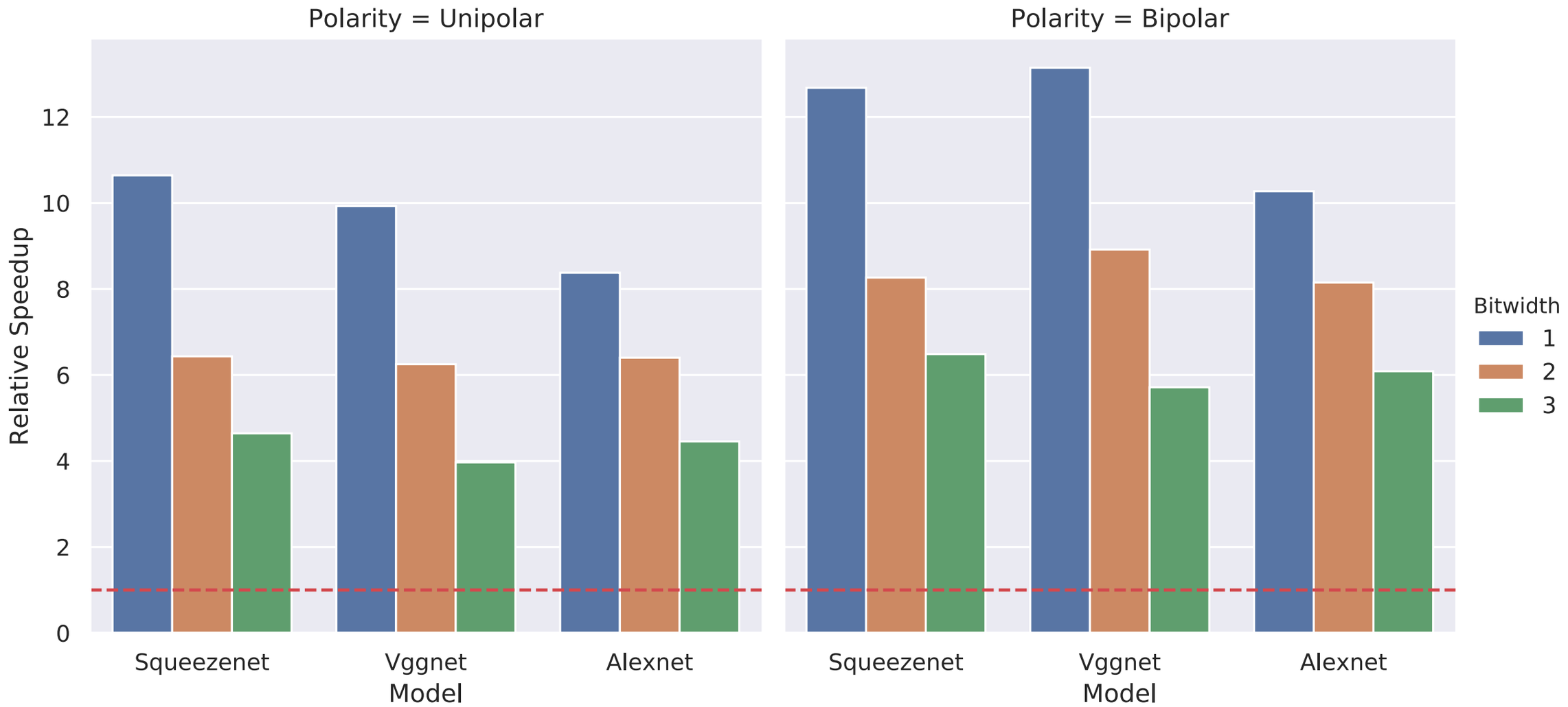



- **Up to 10X speedup**
- **Memory savings from fused schedule yield 1.3X speedup**




- **All optimizations have significant impact**
- **Fused glue offers a 2X speedup**

# All Speedups



# Accuracy / Runtime

| | Model | Name | 1-bit | 2-bit | 3-bit | full precision |
|---|---|---|---|---|---|---|
| | | | ImageNet top-1 accuracy / Runtime (ms) | | | |
| 1 | AlexNet | Xnor-Net [48] | 44.2% / — | — / — | — / — | 56.6% / — |
| 2 | AlexNet | BNN [12] | 27.9% / — | — / — | — / — | — / — |
| 3 | AlexNet | DoReFaNet [63] | 43.6% / — | 49.8% / — | 48.4% / — | 55.9% / — |
| 4 | AlexNet | QNN [27] | 43.3% / — | 51.0% / — | — / — | 56.6% / — |
| 5 | AlexNet | HWGQ [4] | — / — | 52.7% / — | — / — | 58.5% / — |
| 6 | VGGNet | HWGQ [4] | — / — | 64.1% / — | — / — | 69.8% / — |
| 7 | AlexNet | Riptide-unipolar (ours) | 44.5% / 150.4 | 52.5% / 196.8 | 53.6% / 282.8 | 56.5% / 1260.0 |
| 8 | AlexNet | Riptide-bipolar (ours) | 42.8% / 122.7 | 50.4% / 154.6 | 52.4% / 207.0 | 56.5% / 1260.0 |
| 9 | VGGNet | Riptide-unipolar (ours) | 56.8% / 243.8 | 64.2% / 387.2 | 67.1% / 610.0 | 72.7% / 2420.0 |
| 10 | VGGNet | Riptide-bipolar (ours) | 54.4% / 184.1 | 61.5% / 271.4 | 65.2% / 423.5 | 72.7% / 2420.0 |
| 11 | ResNet18 | Riptide-unipolar (ours) | 47.9% / 76.2 | 58.4% / 112.0 | 61.8% / 152.3 | 70.9% / 380.8 |

# Accuracy / Runtime

| | Model | Name | 1-bit | 2-bit | 3-bit | full precision |
|---|---|---|---|---|---|---|
| | | | ImageNet top-1 accuracy / Runtime (ms) | | | |
| 1 | AlexNet | Xnor-Net [48] | 44.2% / — | — / — | — / — | 56.6% / — |
| 2 | AlexNet | BNN [12] | 27.9% / — | — / — | — / — | — / — |
| 3 | AlexNet | DoReFaNet [63] | 43.6% / — | 49.8% / — | 48.4% / — | 55.9% / — |
| 4 | AlexNet | QNN [27] | 43.3% / — | 51.0% / — | — / — | 56.6% / — |
| 5 | AlexNet | HWGQ [4] | — / — | 52.7% / — | — / — | 58.5% / — |
| 6 | VGGNet | HWGQ [4] | — / — | 64.1% / — | — / — | 69.8% / — |
| 7 | AlexNet | Riptide-unipolar (ours) | 44.5% / 150.4 | 52.5% / 196.8 | 53.6% / 282.8 | 56.5% / 1260.0 |
| 8 | AlexNet | Riptide-bipolar (ours) | 42.8% / 122.7 | 50.4% / 154.6 | 52.4% / 207.0 | 56.5% / 1260.0 |
| 9 | VGGNet | Riptide-unipolar (ours) | 56.8% / 243.8 | 64.2% / 387.2 | 67.1% / 610.0 | 72.7% / 2420.0 |
| 10 | VGGNet | Riptide-bipolar (ours) | 54.4% / 184.1 | 61.5% / 271.4 | 65.2% / 423.5 | 72.7% / 2420.0 |
| 11 | ResNet18 | Riptide-unipolar (ours) | 47.9% / 76.2 | 58.4% / 112.0 | 61.8% / 152.3 | 70.9% / 380.8 |

## Available Open Source Soon!