VR-Pipe
Streamlining Hardware Graphics Pipeline for Volume Rendering

Junseo Lee Jaisung Kim Junyong Park Jaewoong Sim

Seoul National University

HPCA'25 | March 2025

Department of ECE
Seoul National University

Advent of Graphics Rendering

3D Gaussian Splatting (3DGS)



Advent of Graphics Rendering

3D Gaussian Splatting (3DGS)

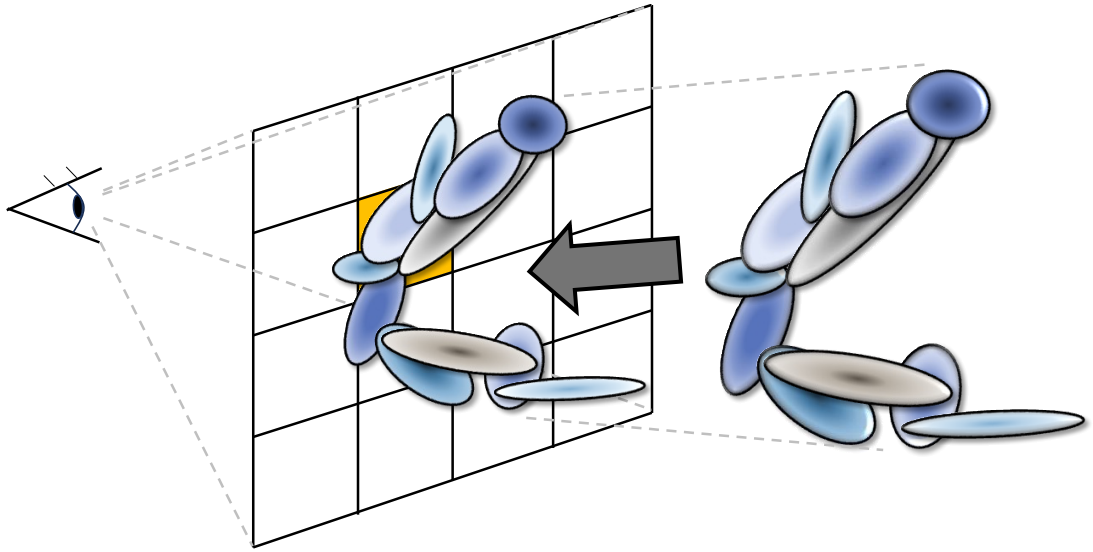
Captured Images





Explicit Representation:
3D Gaussians

+



Splatting + Volume Rendering

Advent of Graphics Rendering

3D Gaussian Splatting (3DGS)

Captured Images





Explicit Representation:

3D Gaussians

Advent of Graphics Rendering

3D Gaussian Splatting (3DGS)

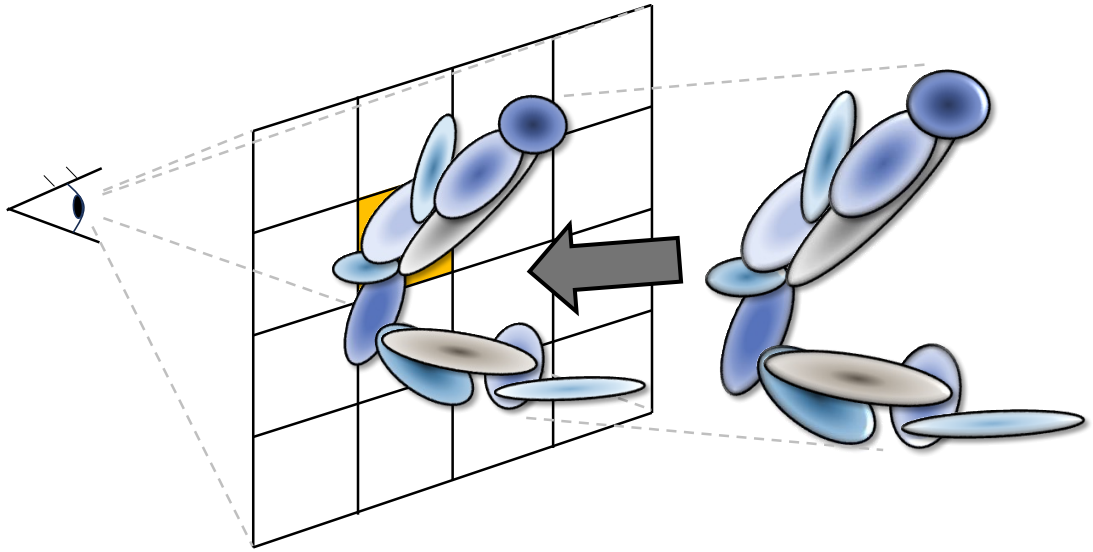
Captured Images



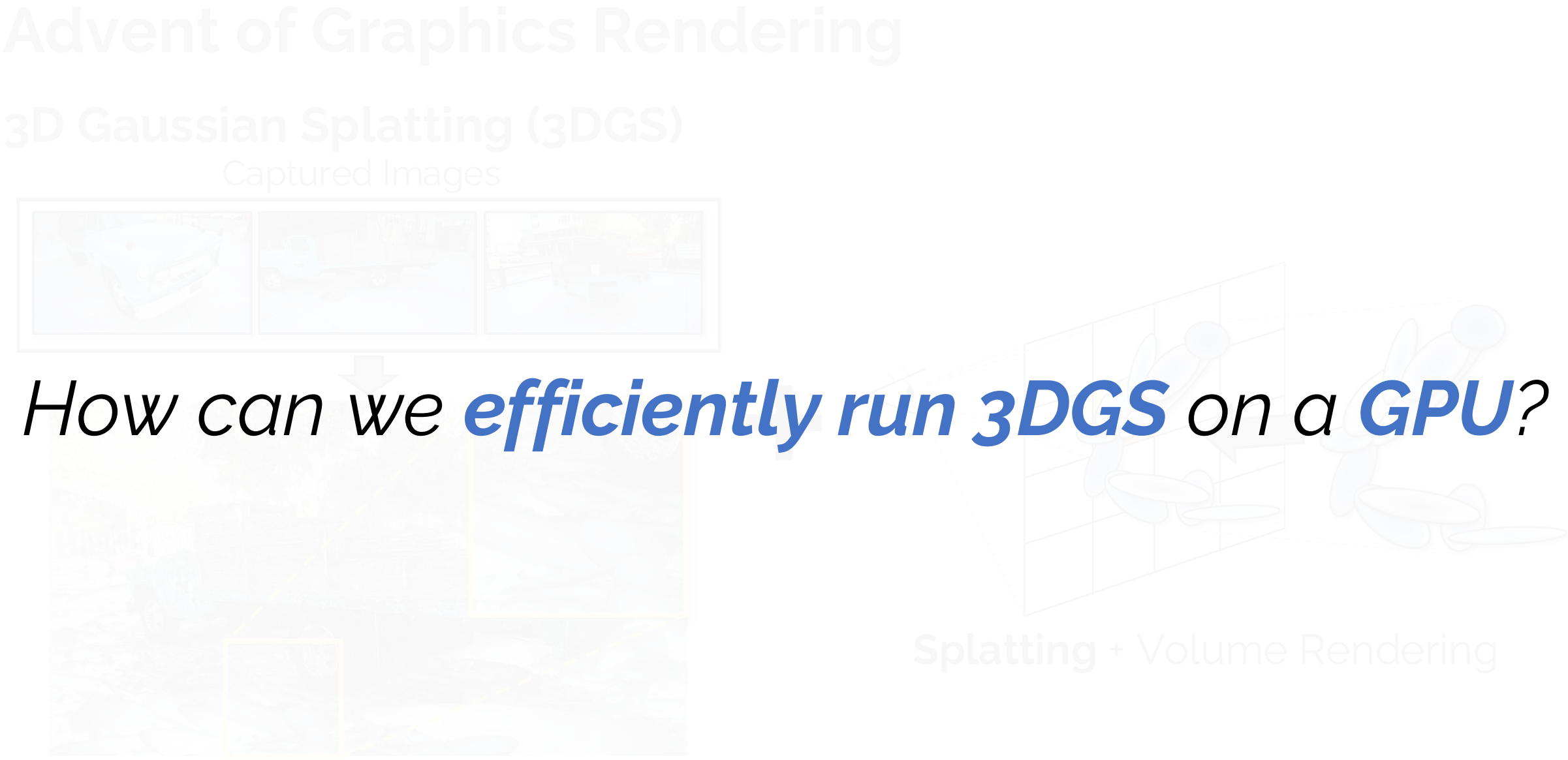


Explicit Representation:
3D Gaussians

+

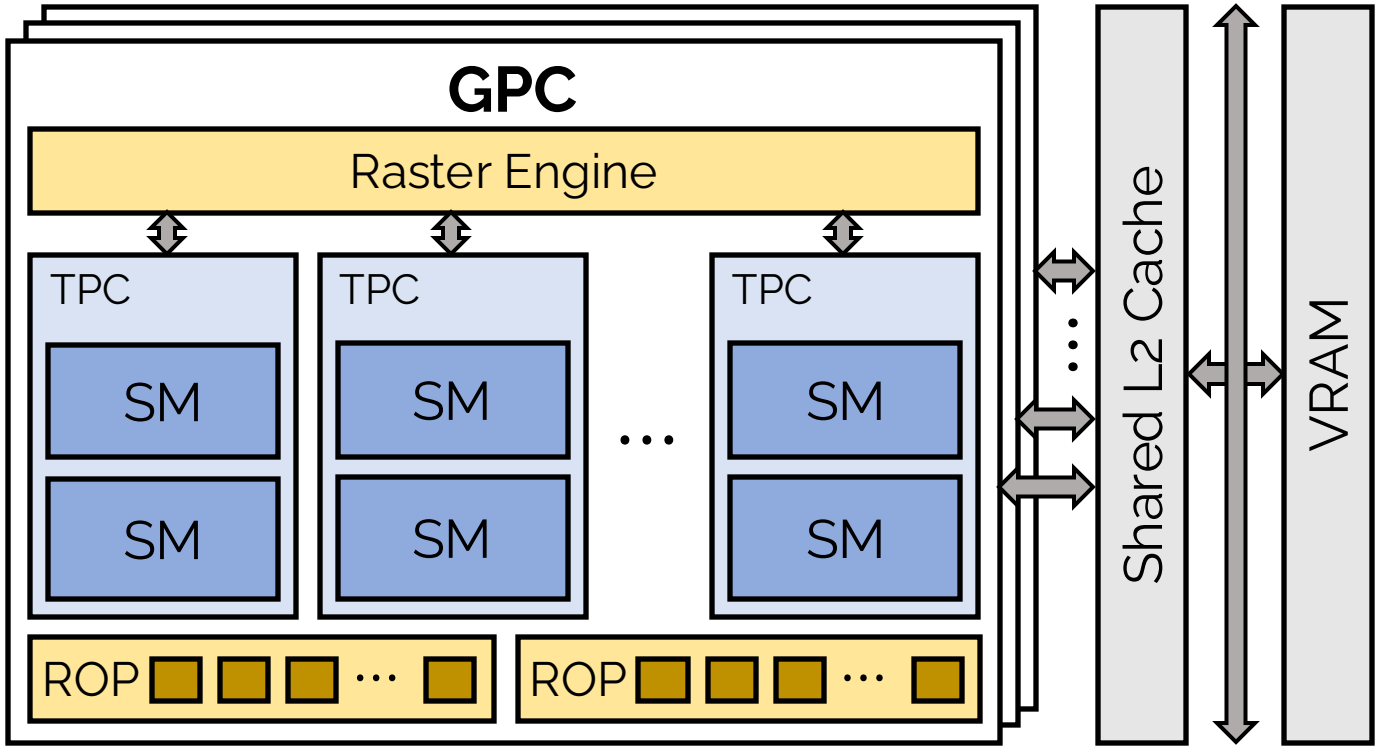


Splatting + Volume Rendering

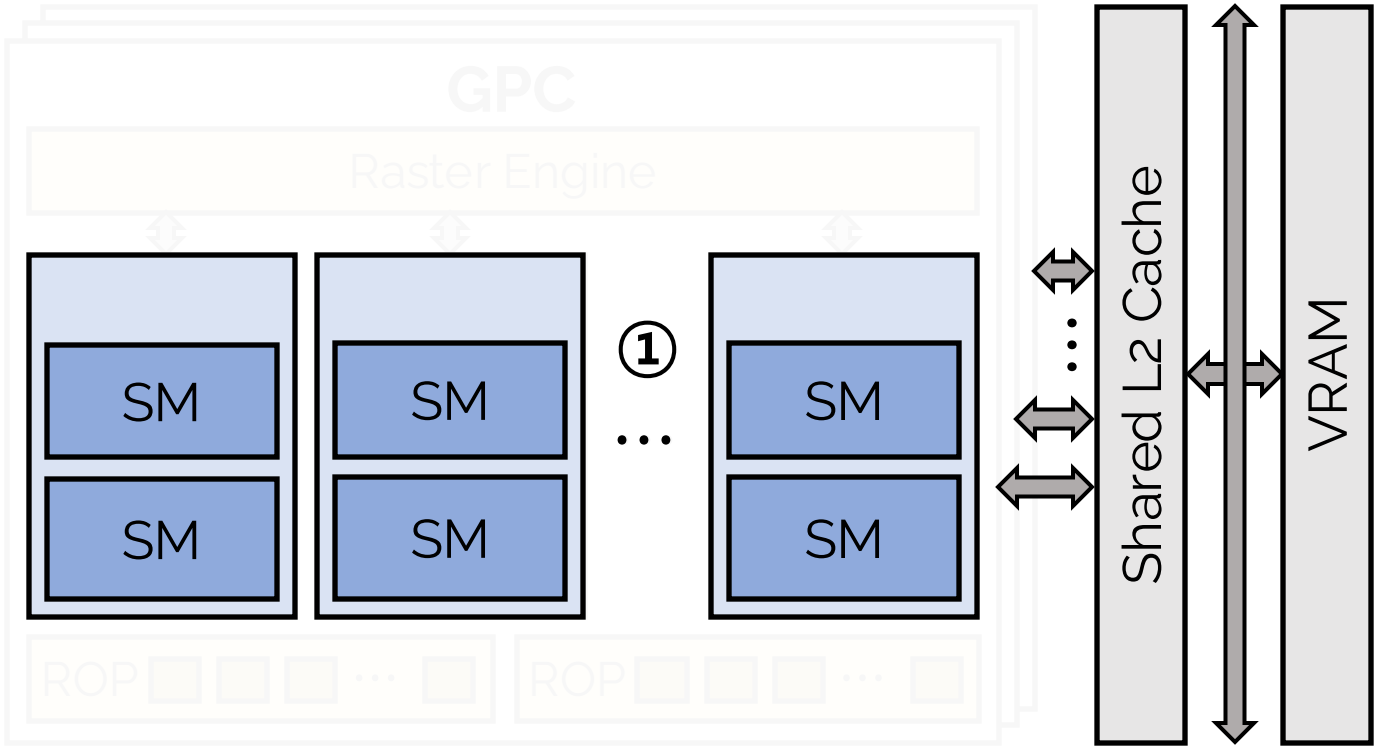


*How can we **efficiently run 3DGS** on a **GPU**?*

3D Gaussian Splatting on a GPU



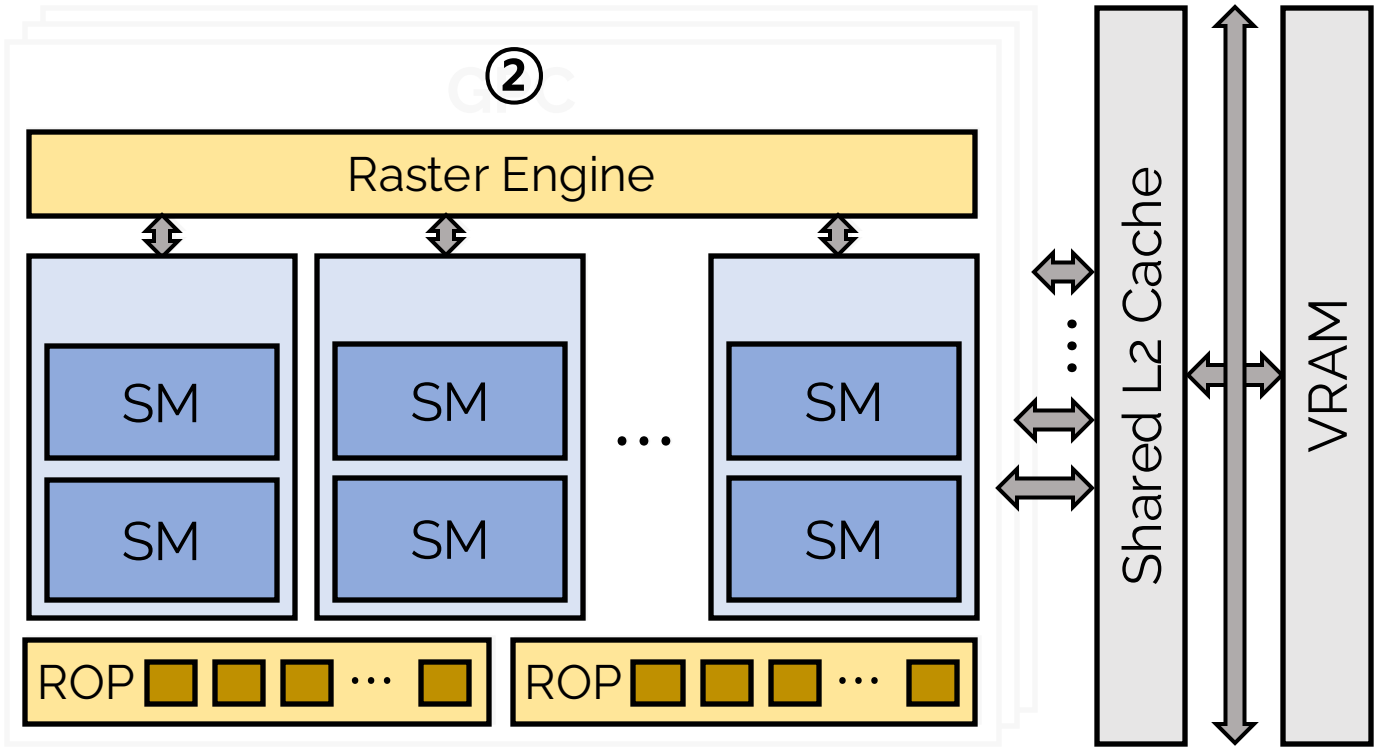
3D Gaussian Splatting on a GPU



① SW-based rendering

- Use only **SMs**
- General-purpose computing frameworks (e.g., CUDA, OpenCL)

3D Gaussian Splatting on a GPU



① SW-based rendering

- Use only **SMs**
- General-purpose computing frameworks (e.g., CUDA, OpenCL)

② HW-based rendering

- Use **graphics-specific fixed-function units** w/ **SMs**
  = hardware graphics pipeline
- Graphics APIs
  (e.g., OpenGL, Vulkan)

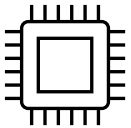
Goal of Our Work



CUDA Optimizations

StopThePop [SIGGRAPH'24]

FlashGS [arXiv'24]



Specialized Accelerators

GSCore [ASPLOS'24]

MetaSapiens [ASPLOS'25]

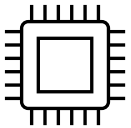
Goal of Our Work



CUDA Optimizations

StopThePop [SIGGRAPH'24]

FlashGS [arXiv'24]



Specialized Accelerators

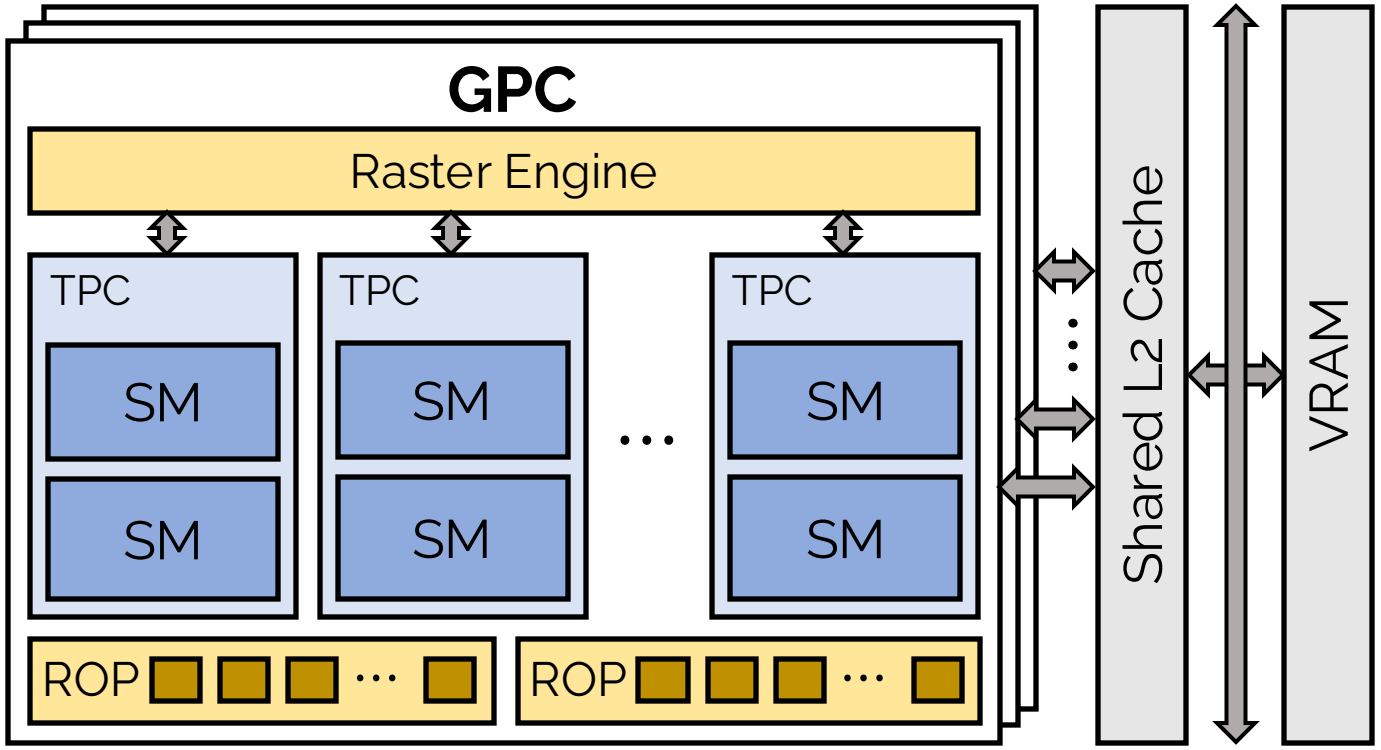
GSCore [ASPLOS'24]

MetaSapiens [ASPLOS'25]

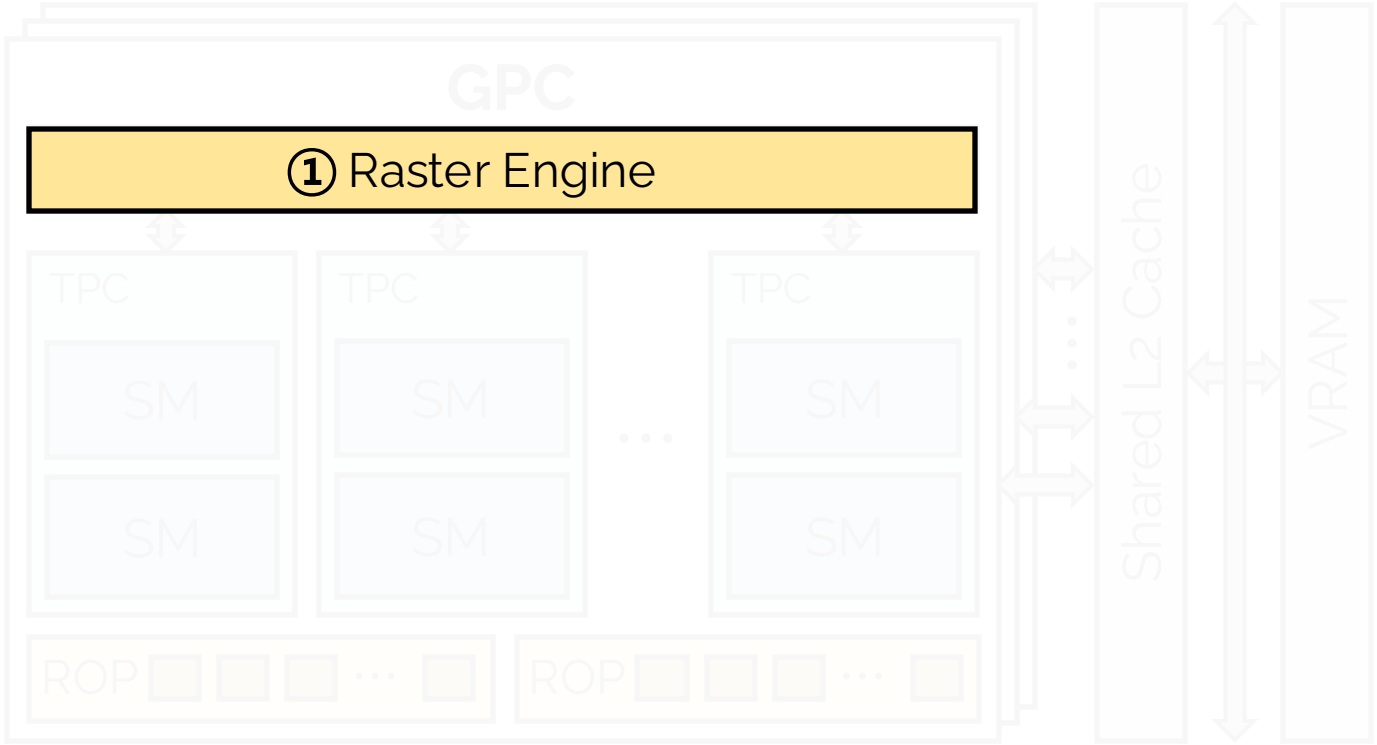
Our Work

Extend the existing **hardware graphics pipeline** for **volume rendering (e.g., 3DGS)**

Hardware Graphics Pipeline



Hardware Graphics Pipeline



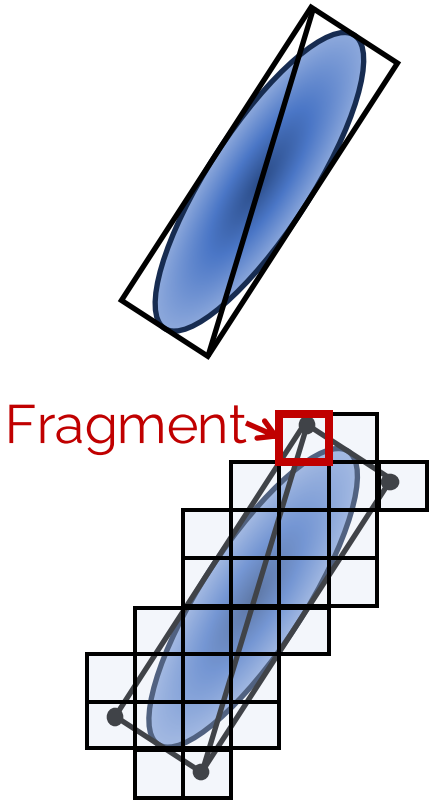
Triangle



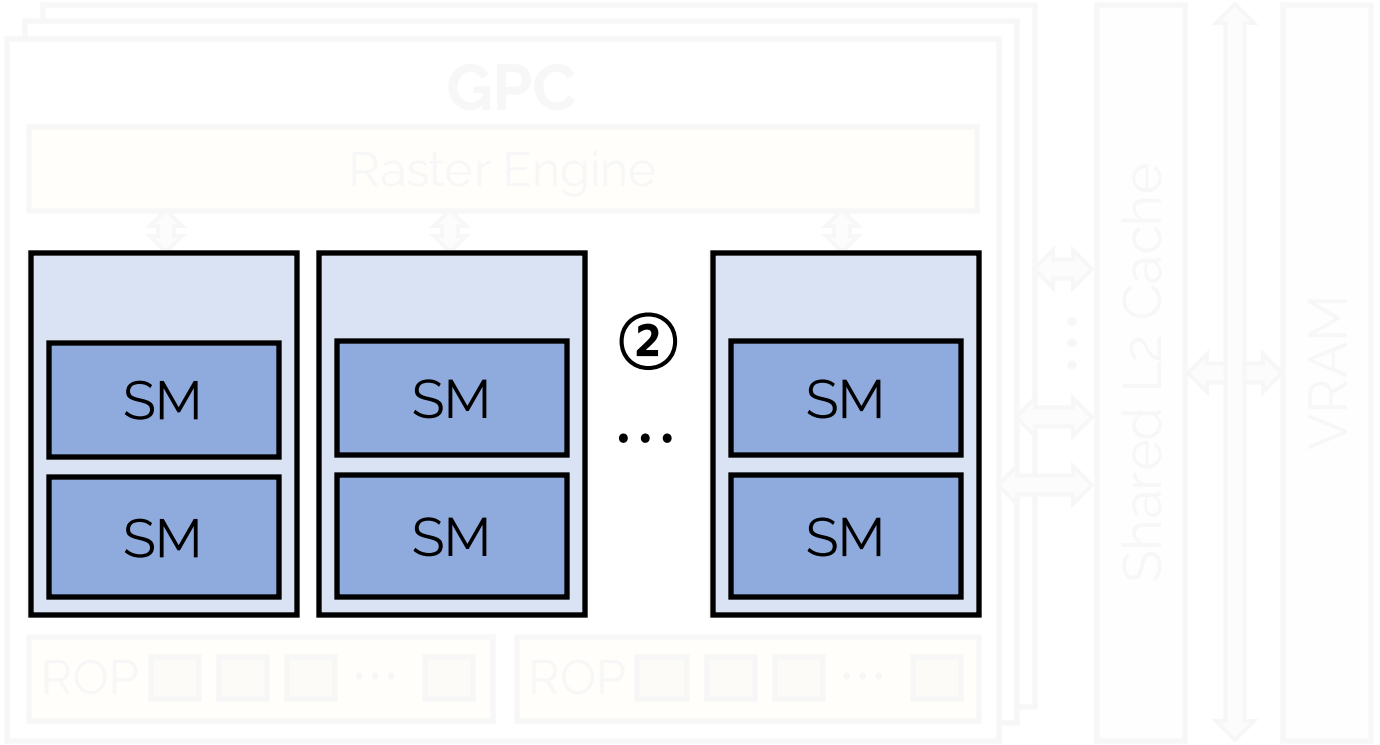
① Rasterization



Fragments



Hardware Graphics Pipeline



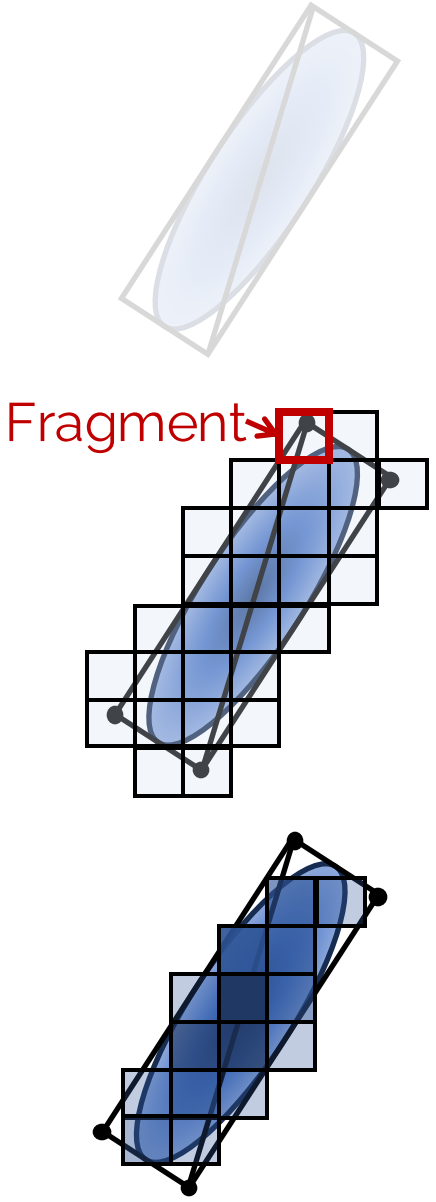
Triangle

① Rasterization

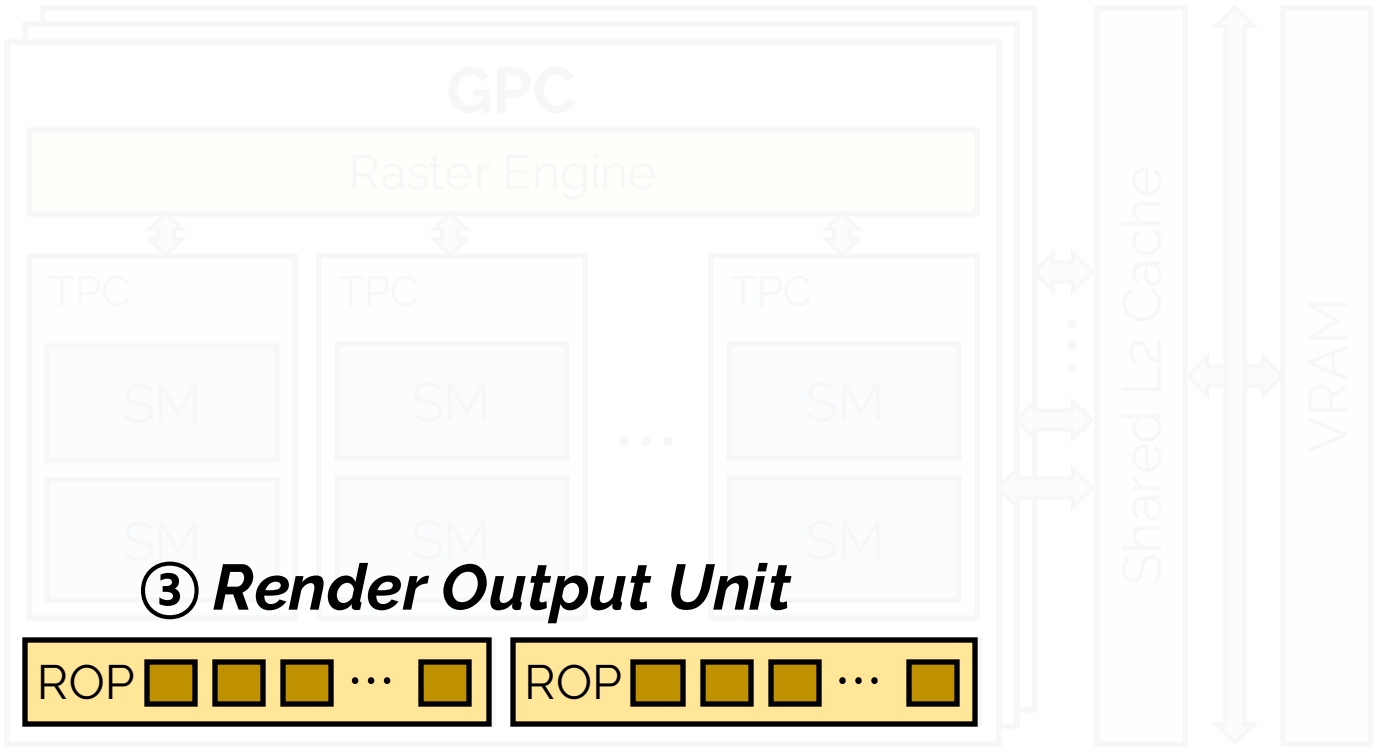
Fragments

② Fragment Shading

Shaded fragments



Hardware Graphics Pipeline



③ ***Render Output Unit***

ROP ROP

Triangle

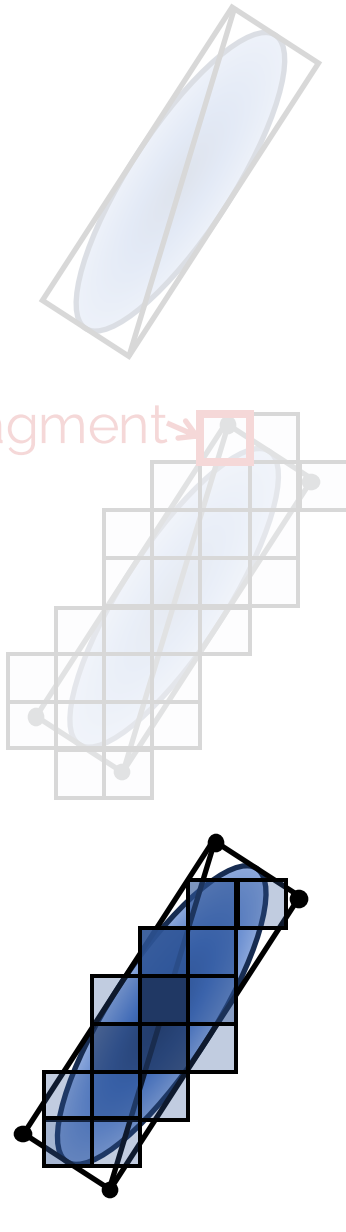
① **Rasterization**

Fragments

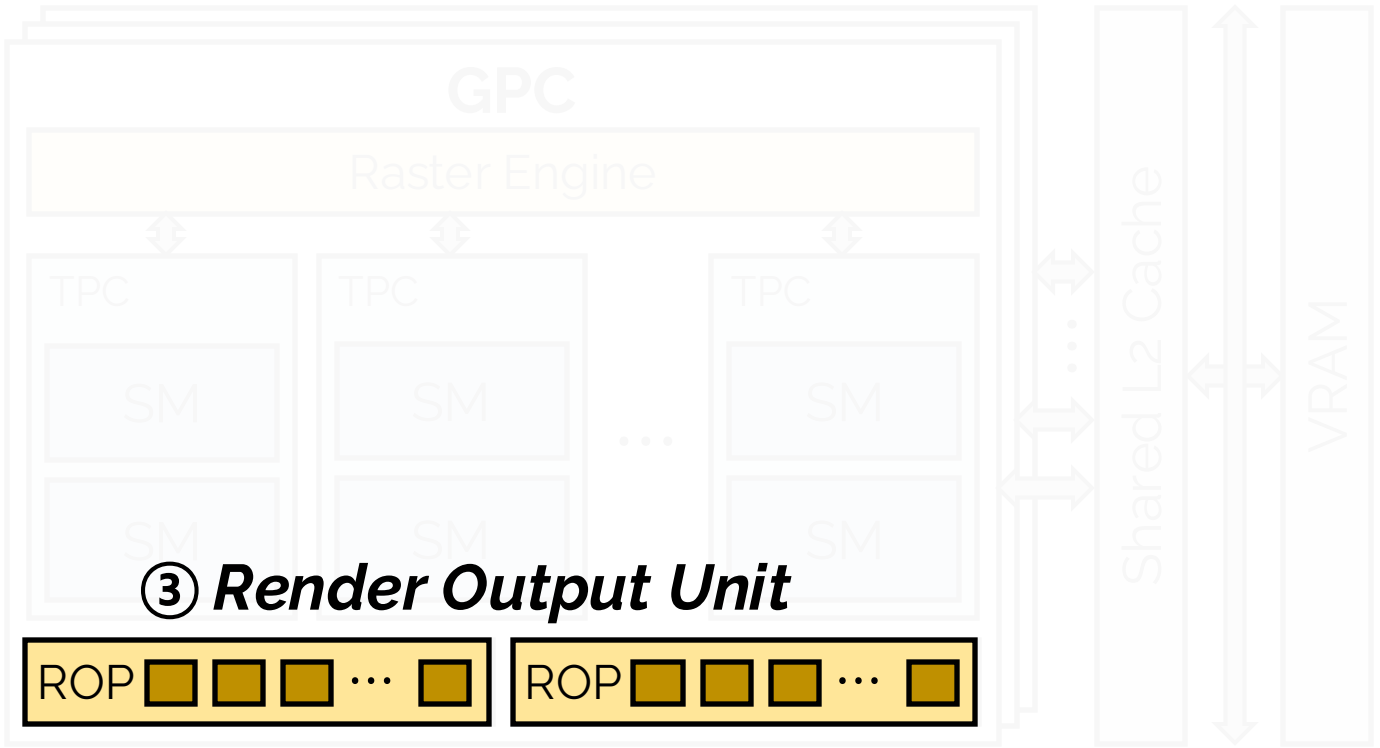
② **Fragment Shading**

Shaded fragments

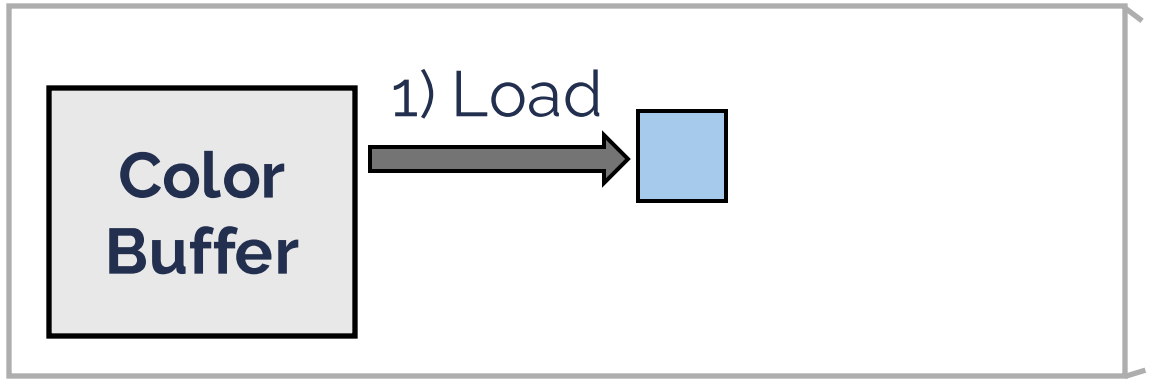
③ **Pixel Blending**



Hardware Graphics Pipeline



③ *Render Output Unit*



Triangle

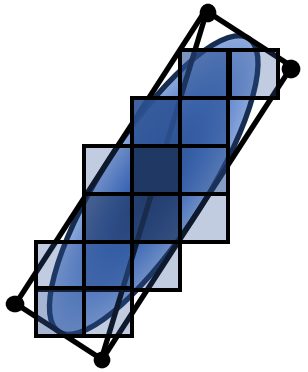
① Rasterization

Fragments

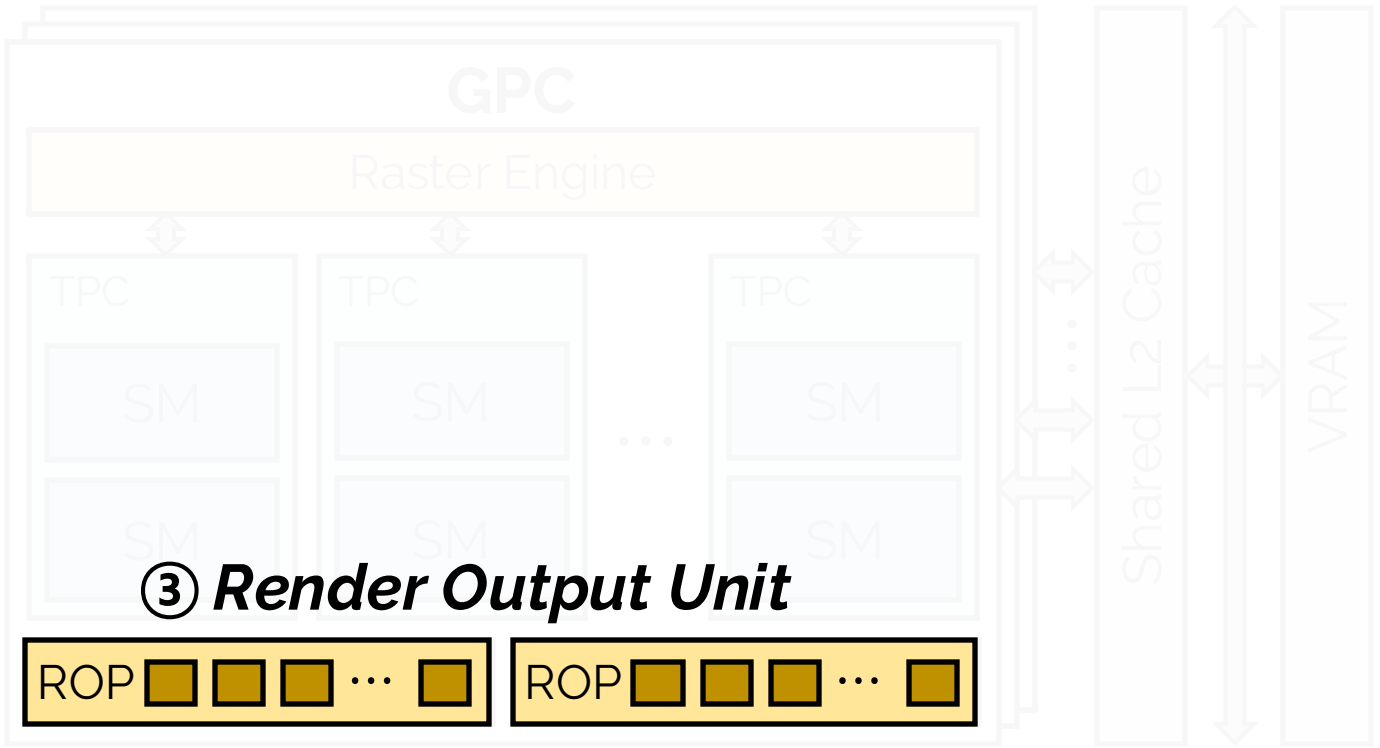
② Fragment Shading

Shaded fragments

③ Pixel Blending



Hardware Graphics Pipeline



Triangle

① Rasterization

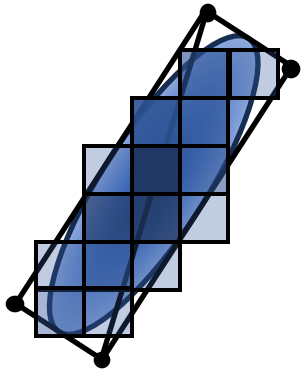
Fragments

② Fragment Shading

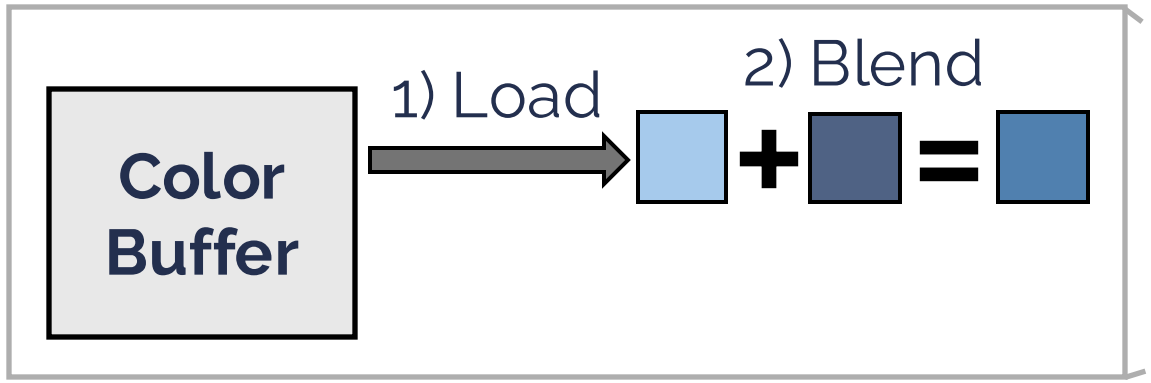
Shaded fragments

③ Pixel Blending

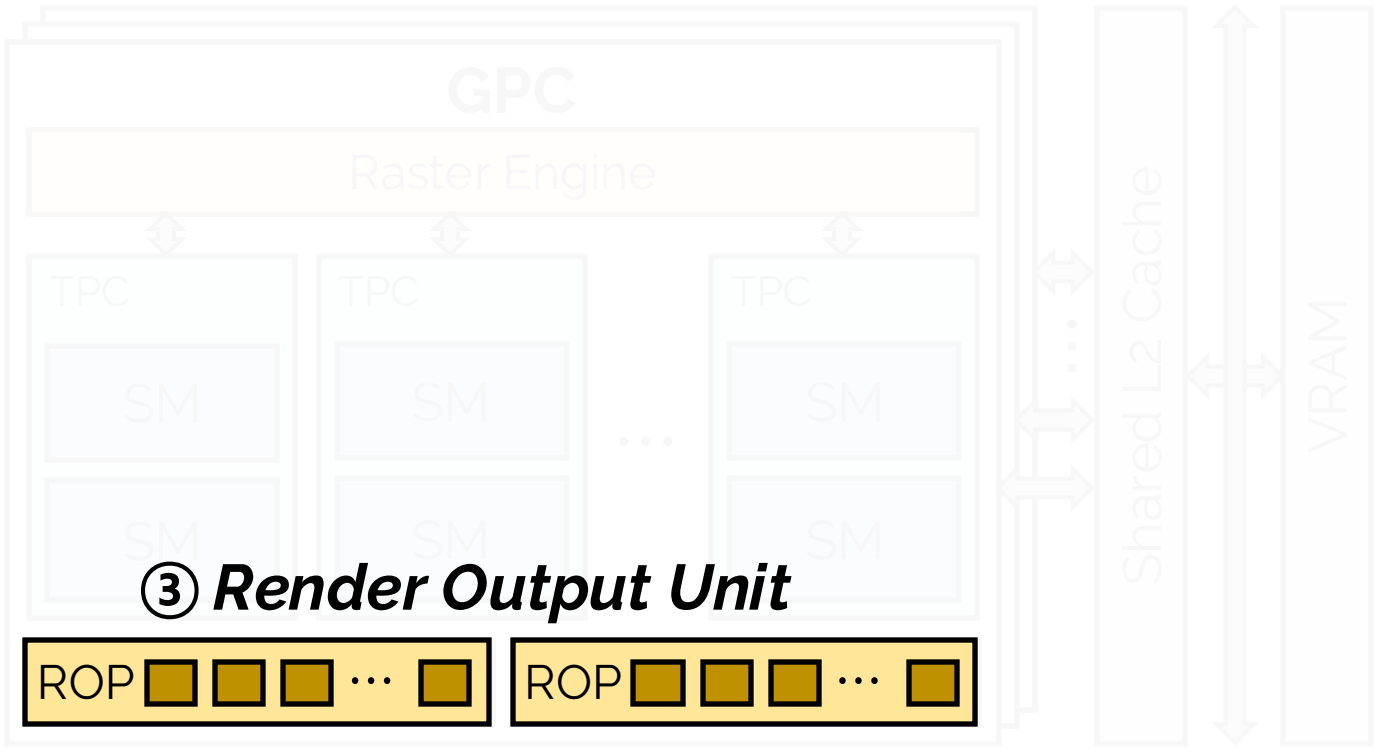
Fragment



③ ***Render Output Unit***



Hardware Graphics Pipeline



Triangle

① **Rasterization**

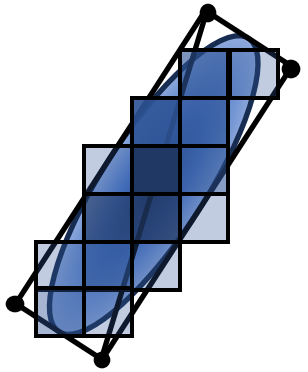
Fragments

② **Fragment Shading**

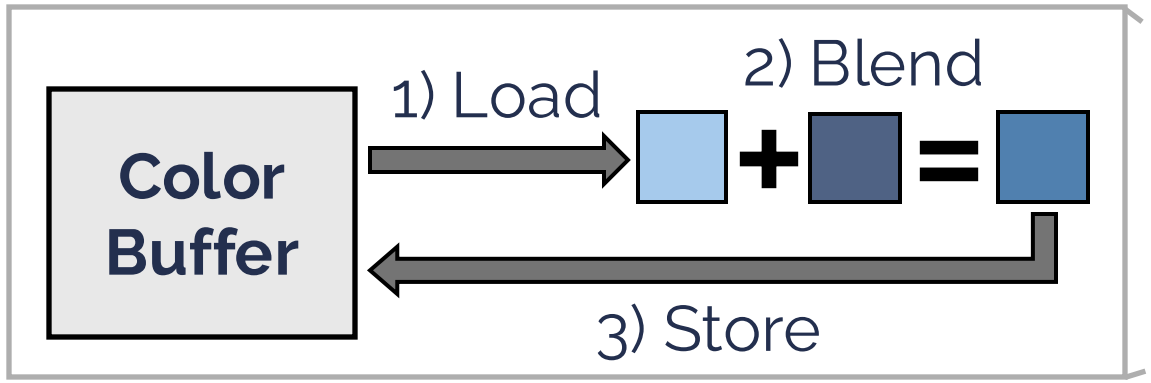
Shaded fragments

③ **Pixel Blending**

Fragment



③ ***Render Output Unit***

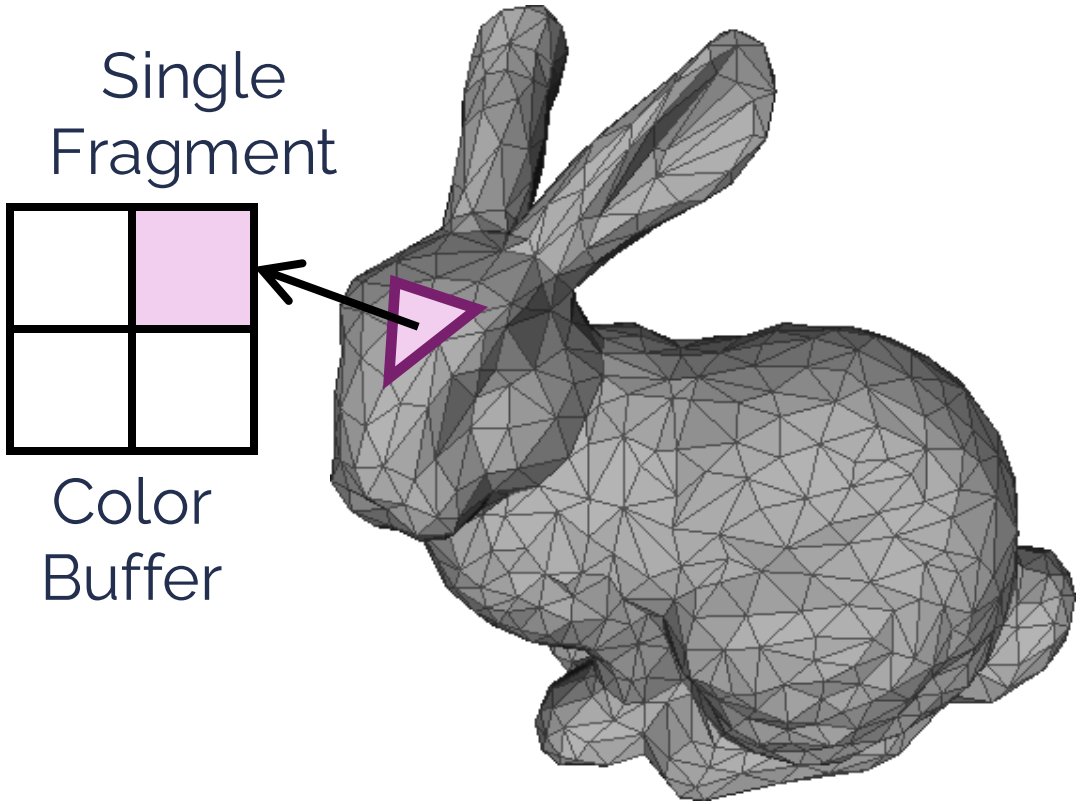


Outline

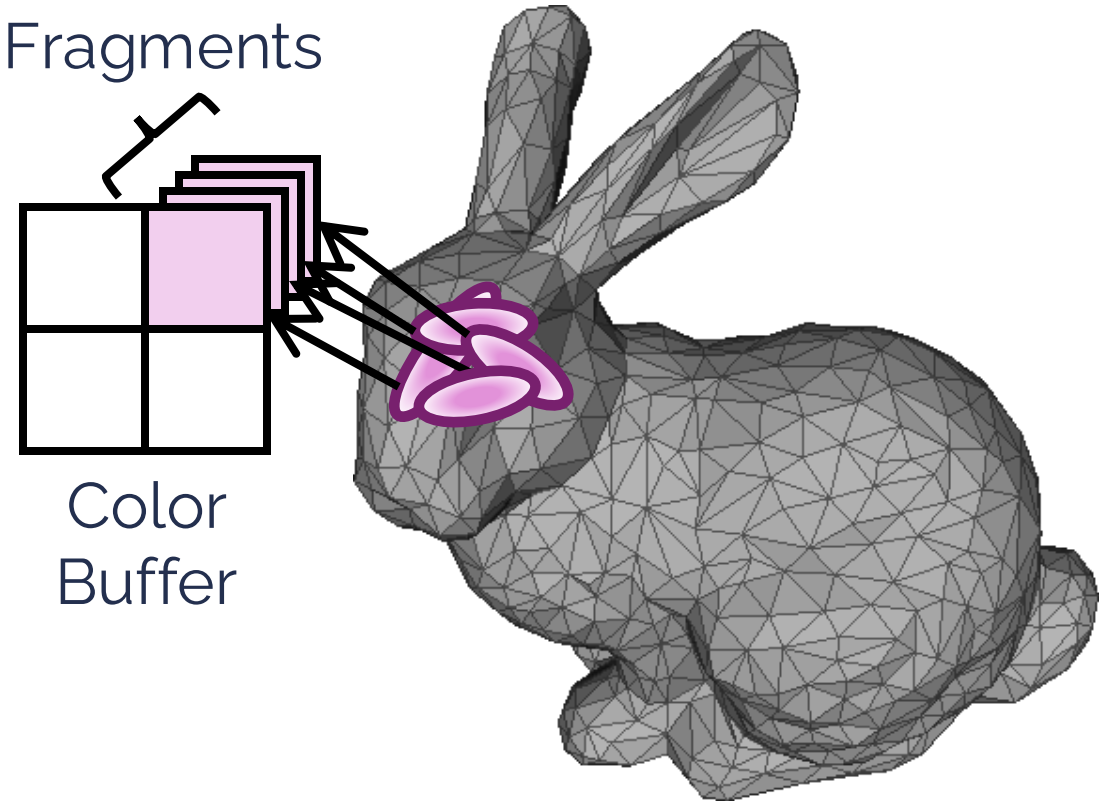
- **Background**
  - 3D Gaussian Splatting (3DGS)
  - Hardware Graphics Pipeline
- **Limitations of Graphics Hardware**
- **VR-Pipe: Graphics Hardware Extension for Volume Rendering**
  - Quad Merging with Multi-Granular Tile Binning
  - Hardware Support for Early Termination
- **Evaluation**
- **Conclusion**

Limitations of Graphics Hardware

Mesh-based Rendering

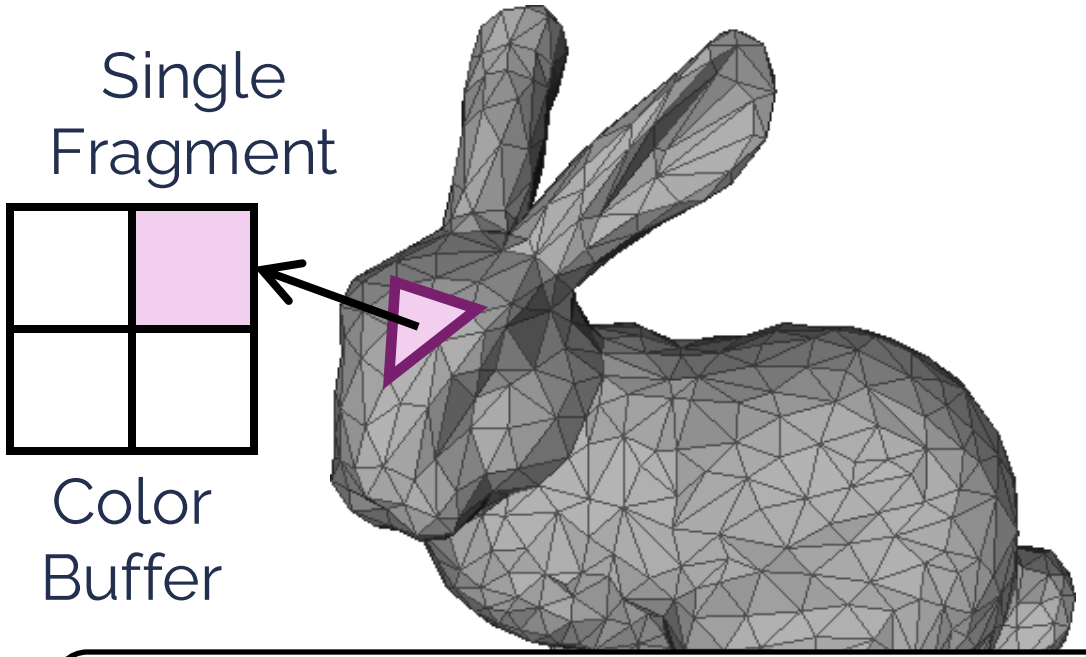


Volume Rendering (e.g., 3DGS)



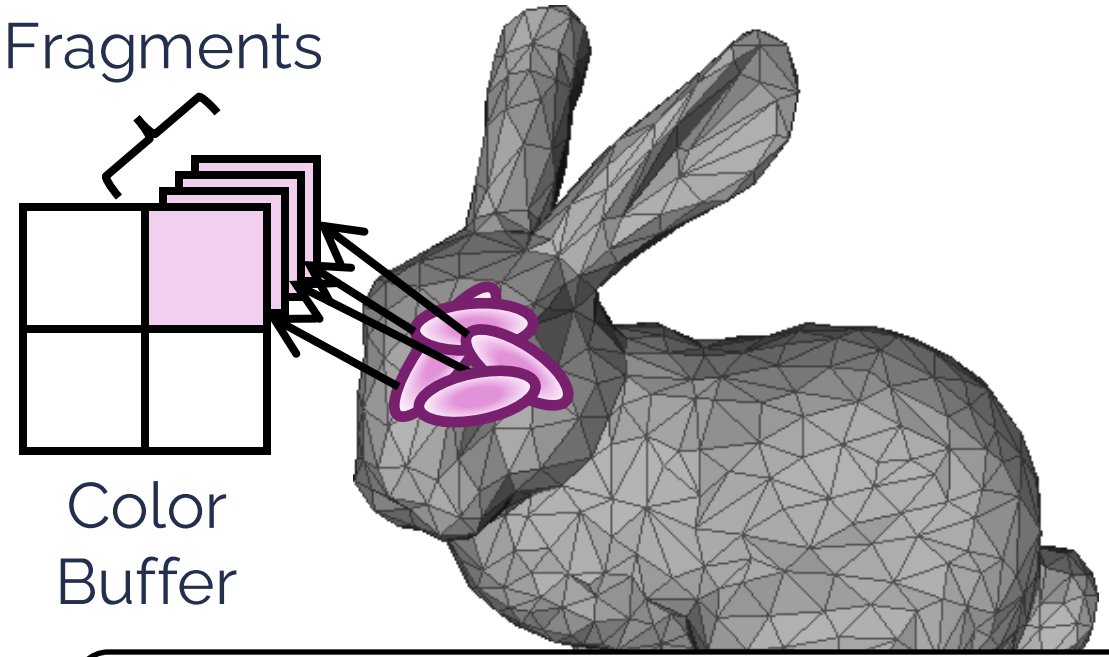
Limitations of Graphics Hardware

Mesh-based Rendering



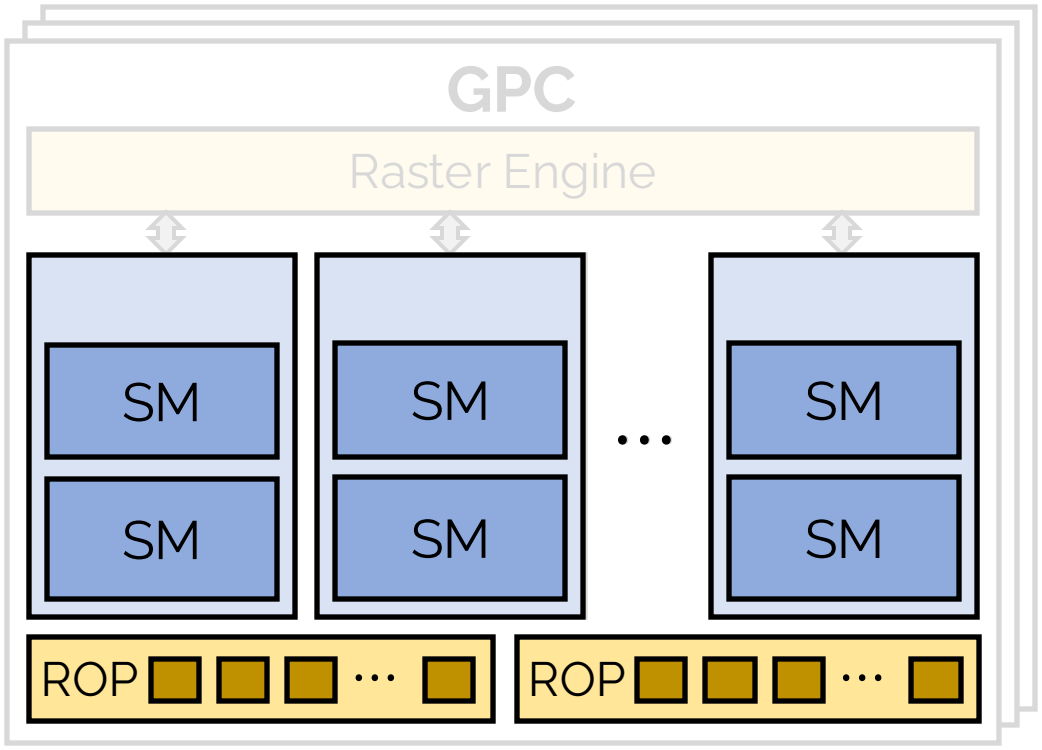
One or few *opaque* fragments per pixel

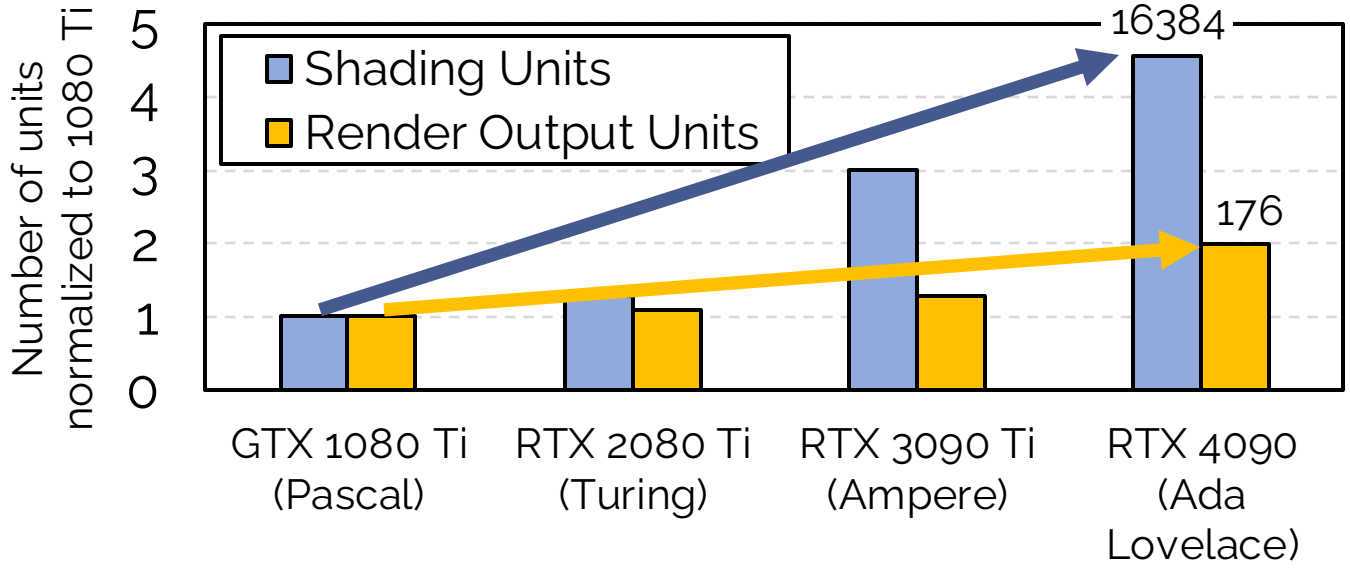
Volume Rendering (e.g., 3DGS)



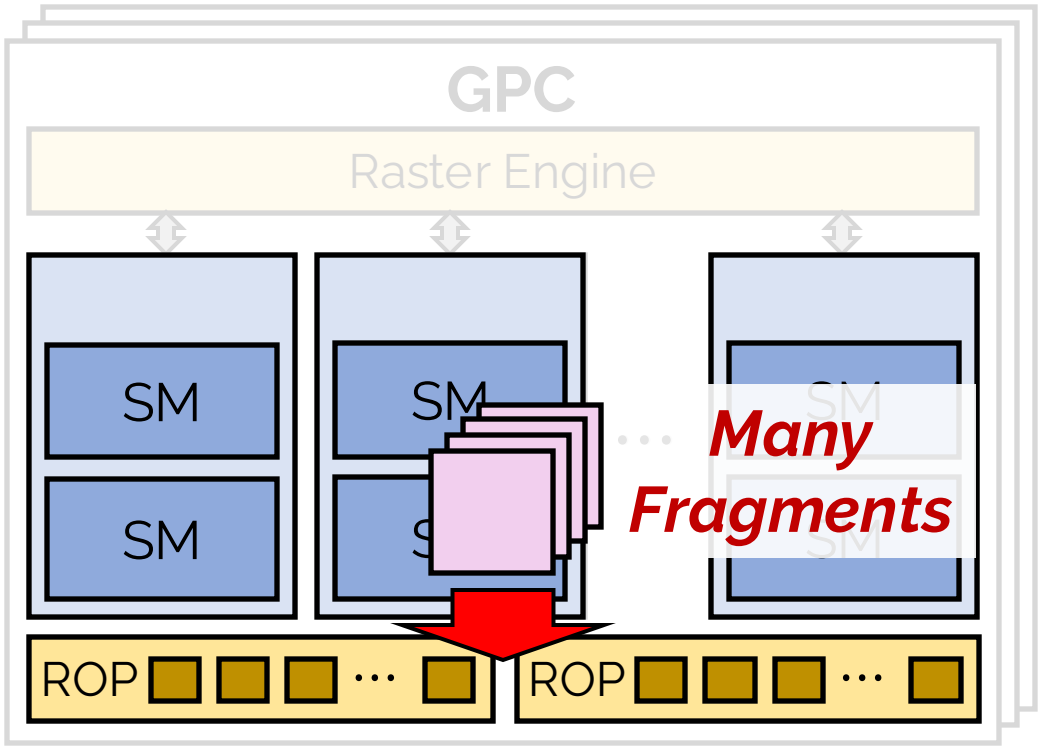
Many *transparent* fragments (i.e., 10-100) per pixel

Limitations of Graphics Hardware

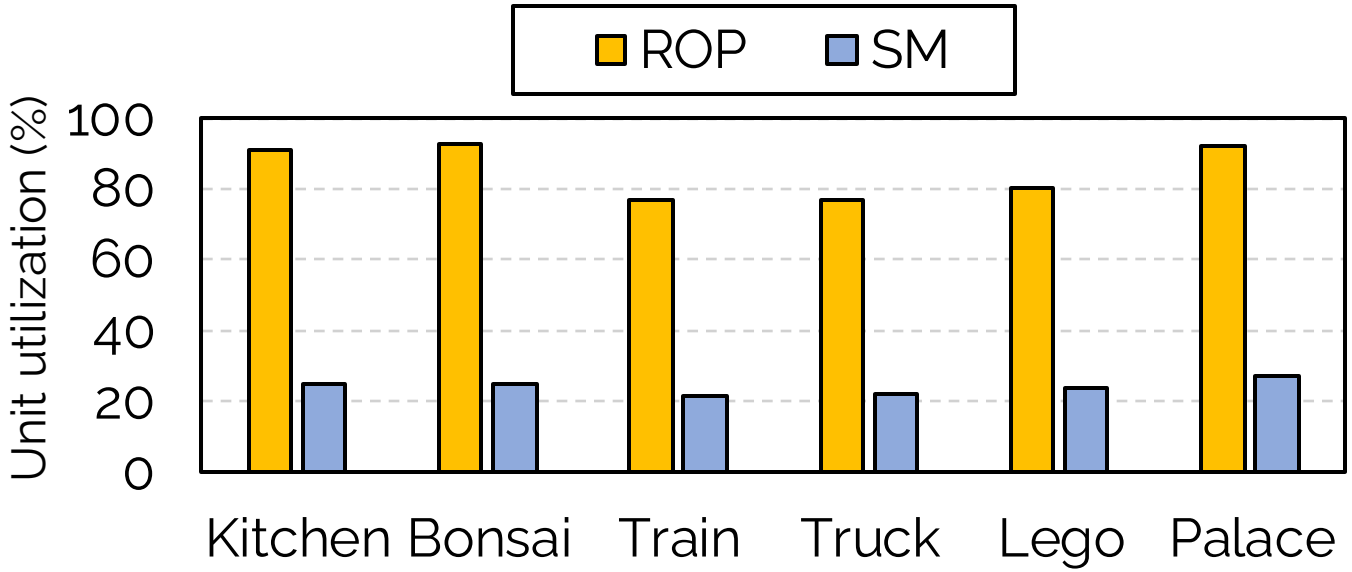




Limitations of Graphics Hardware

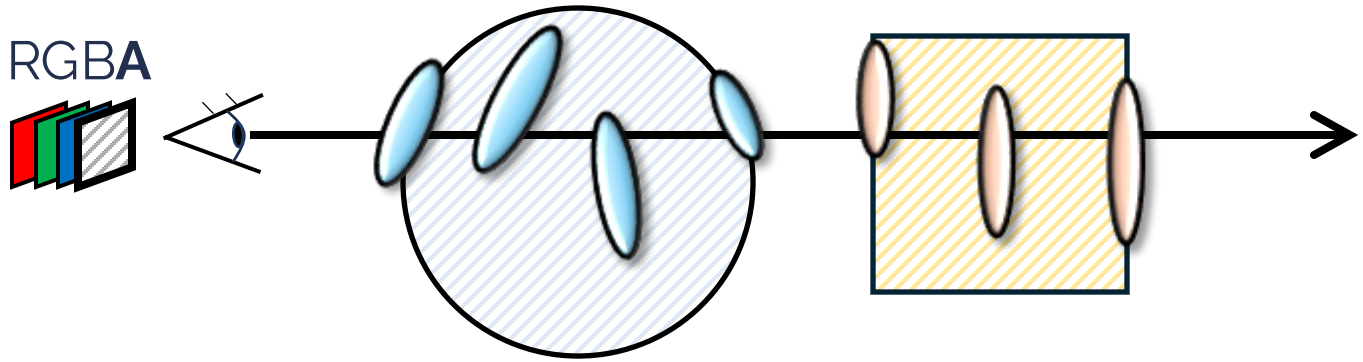


Rendering performance is **dictated by ROPs**



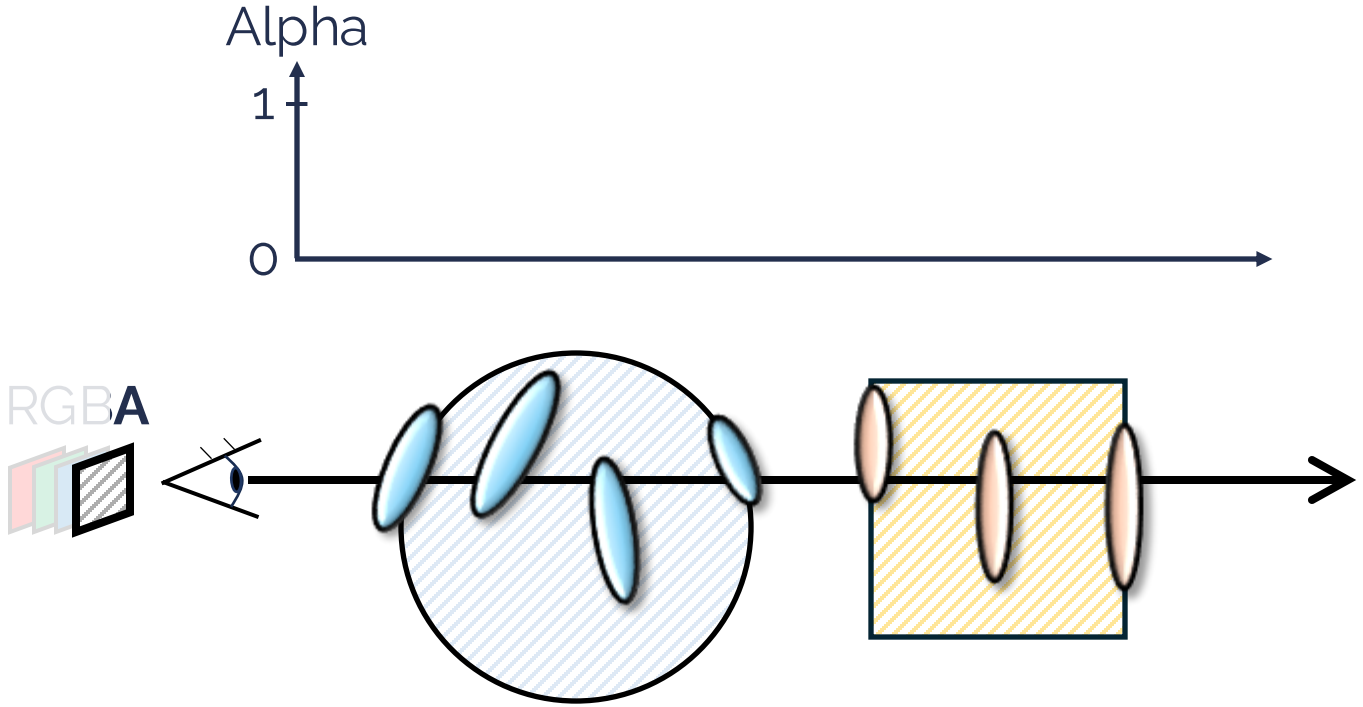
Limitations of Graphics Hardware

Early Termination



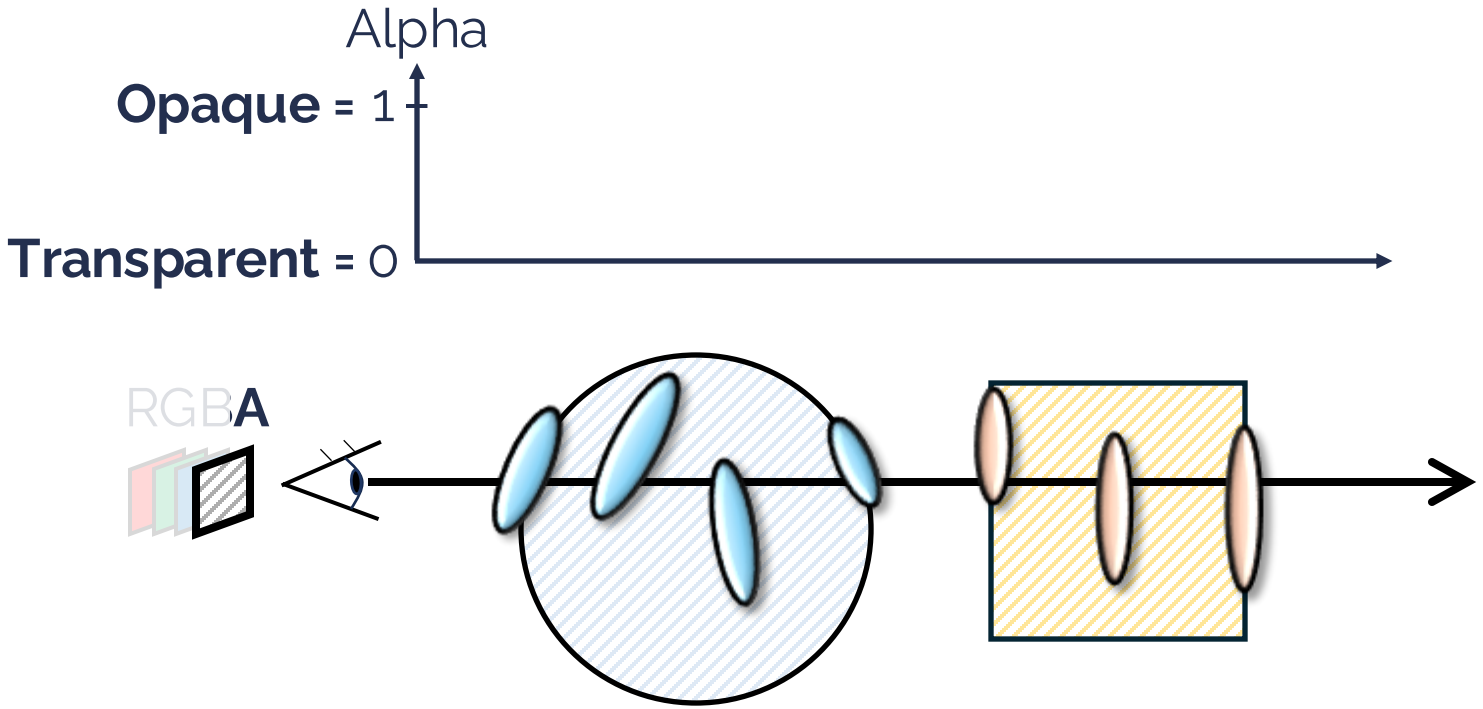
Limitations of Graphics Hardware

Early Termination



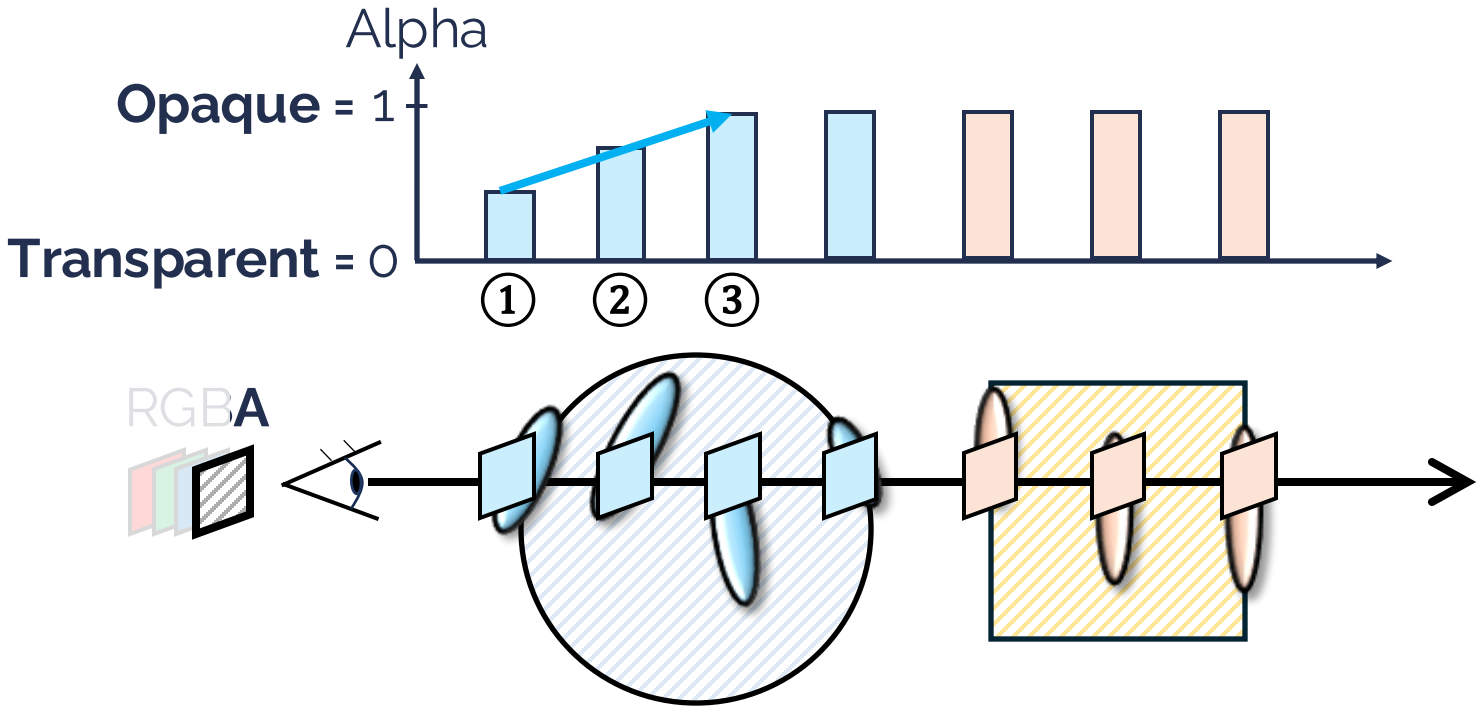
Limitations of Graphics Hardware

Early Termination



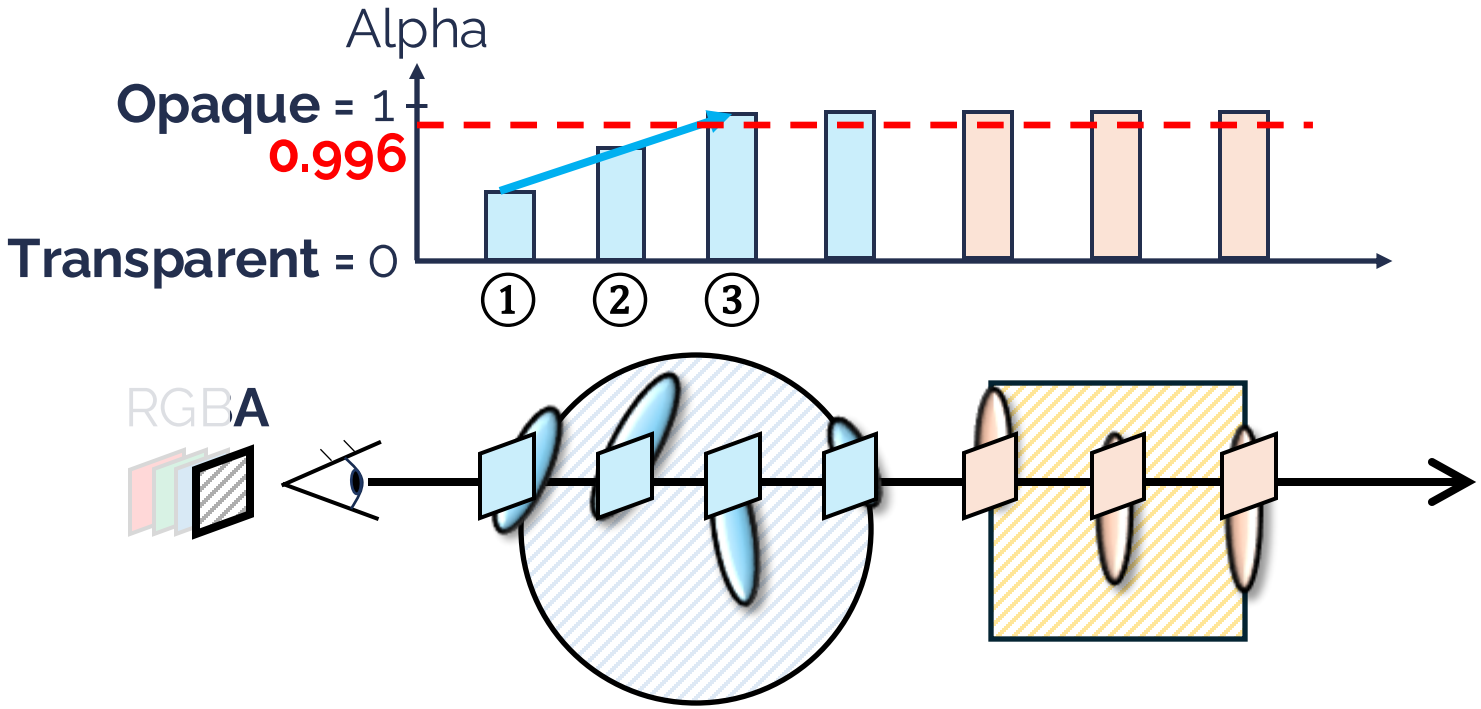
Limitations of Graphics Hardware

Early Termination



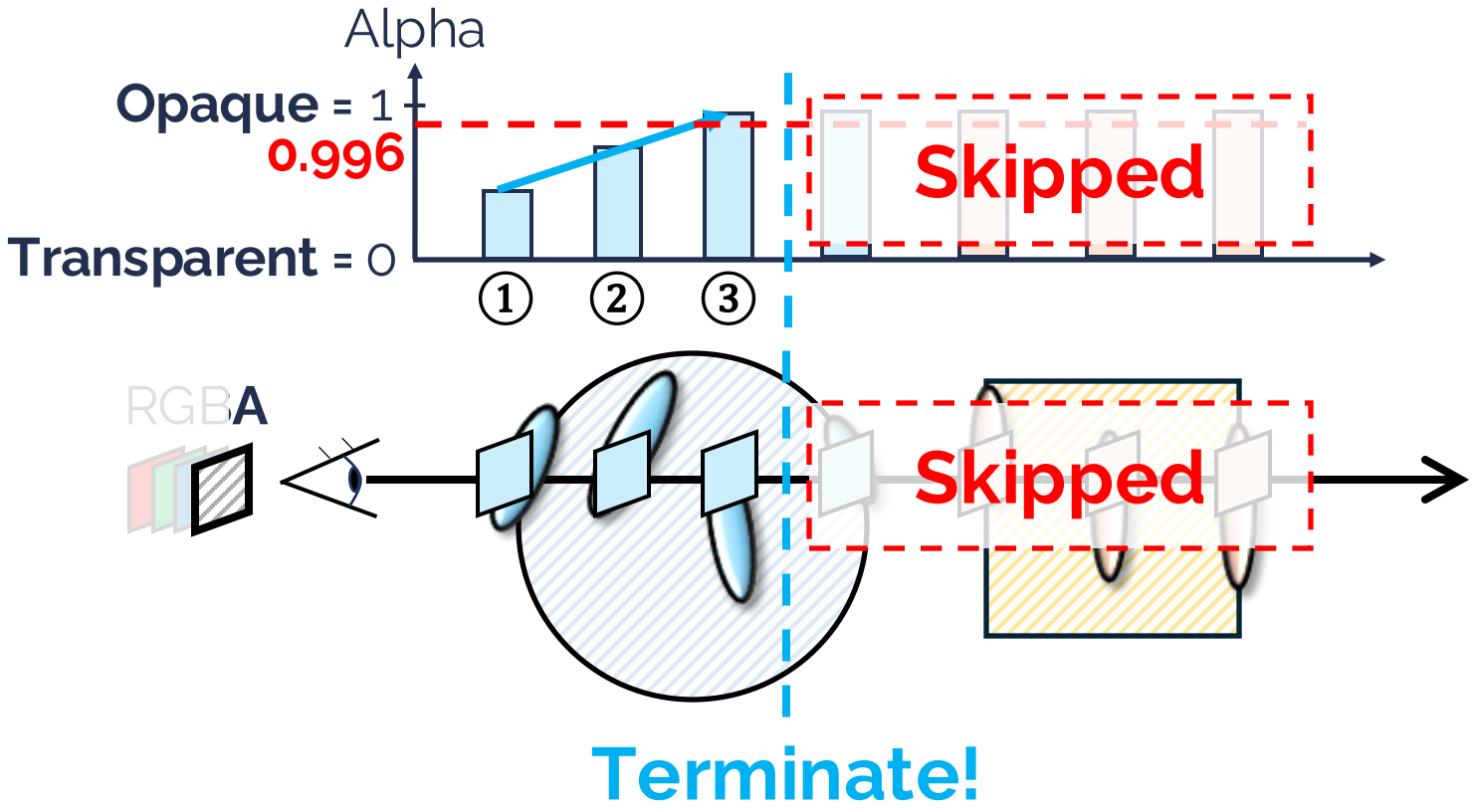
Limitations of Graphics Hardware

Early Termination



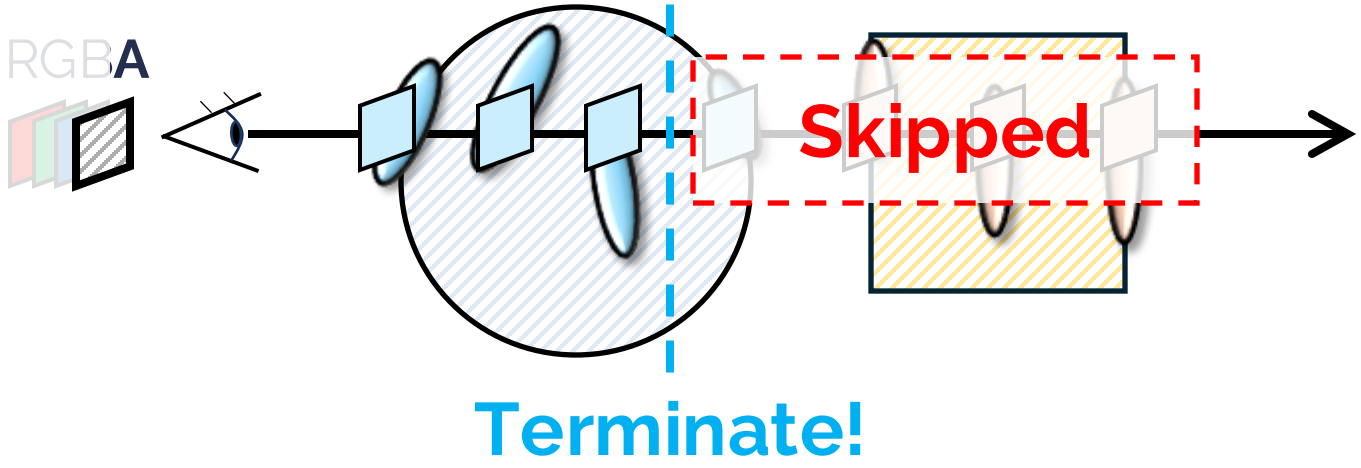
Limitations of Graphics Hardware

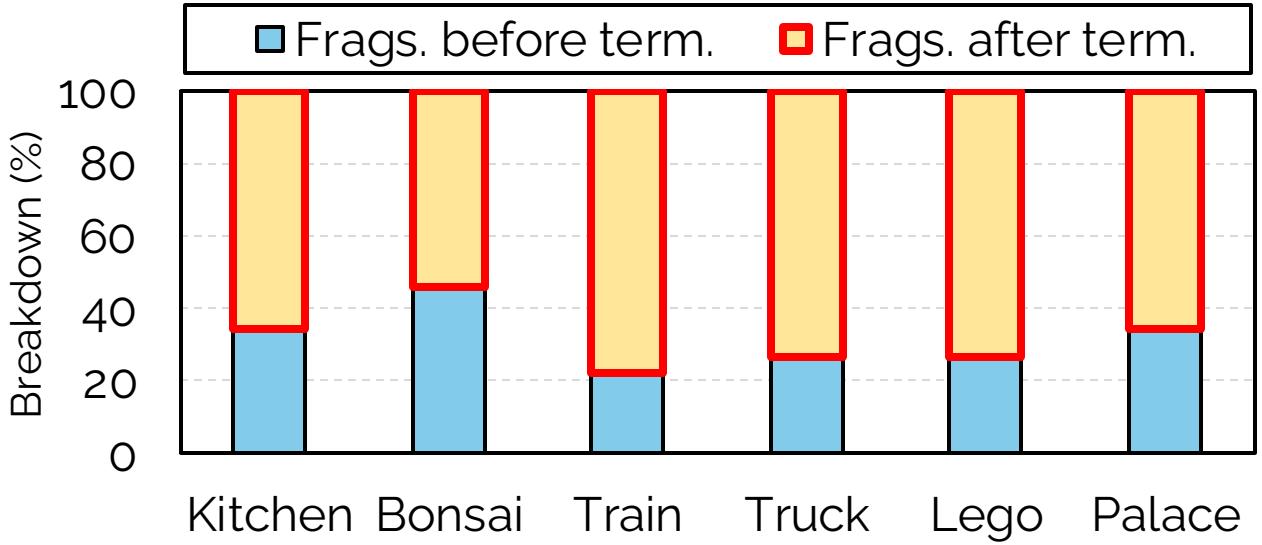
Early Termination



Limitations of Graphics Hardware

Early Termination



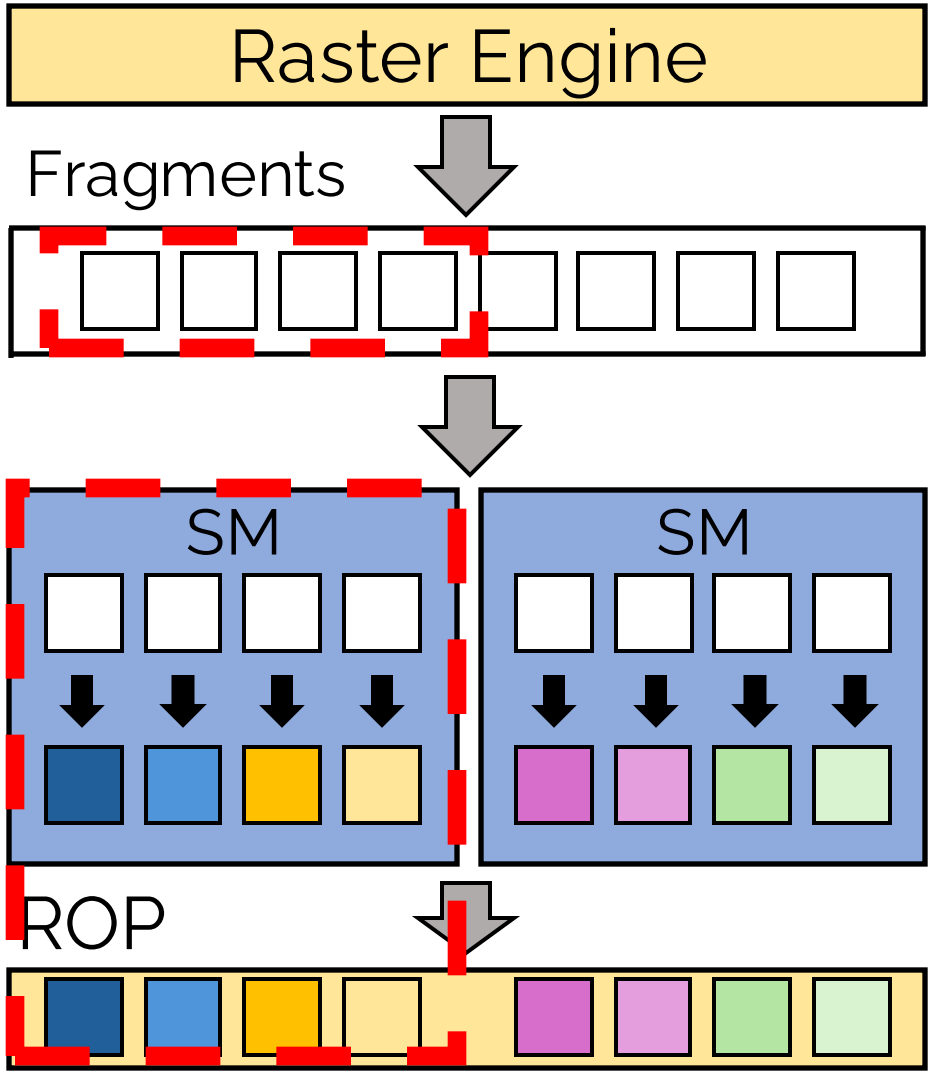


Limitations of Graphics Hardware

Limitations of Graphics Hardware

Observation 1

Many fragments are
unnecessarily
shaded and blended



Limitations of Graphics Hardware

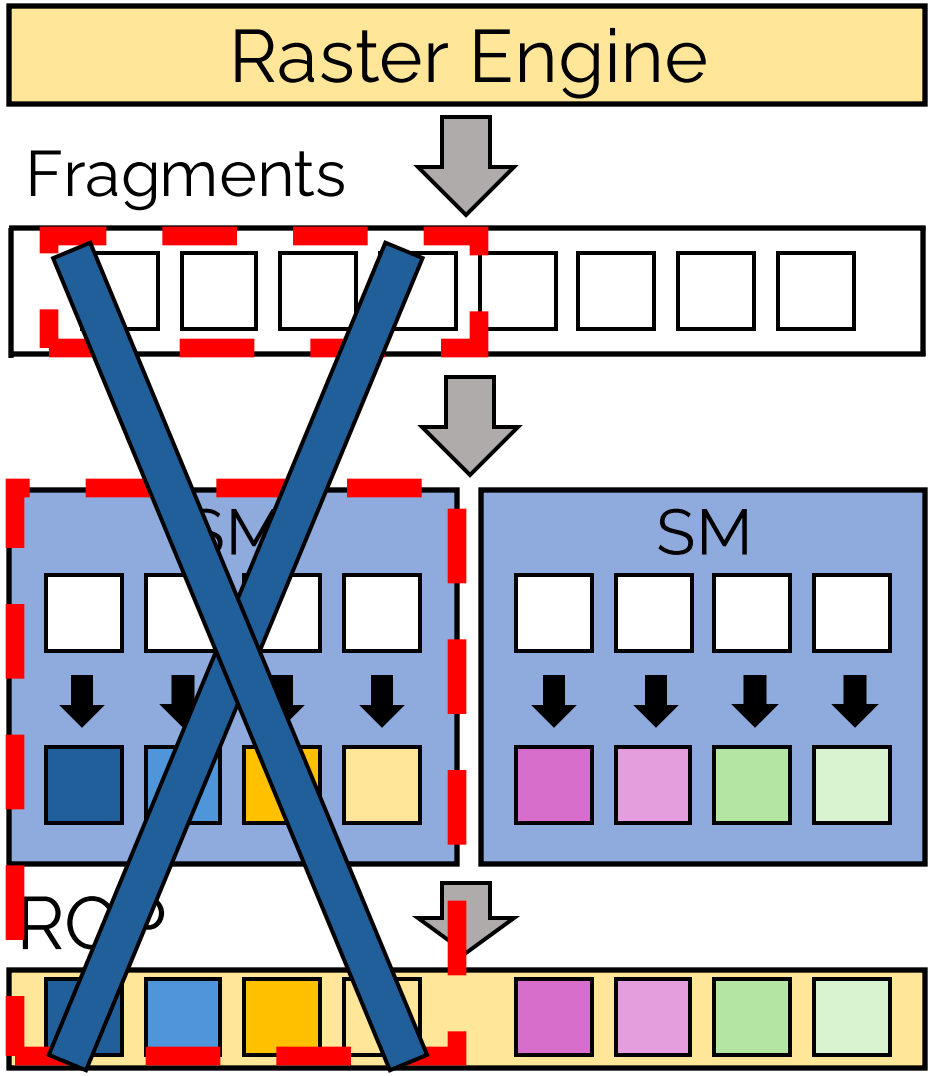
Observation 1

Many fragments are **unnecessarily shaded and blended**

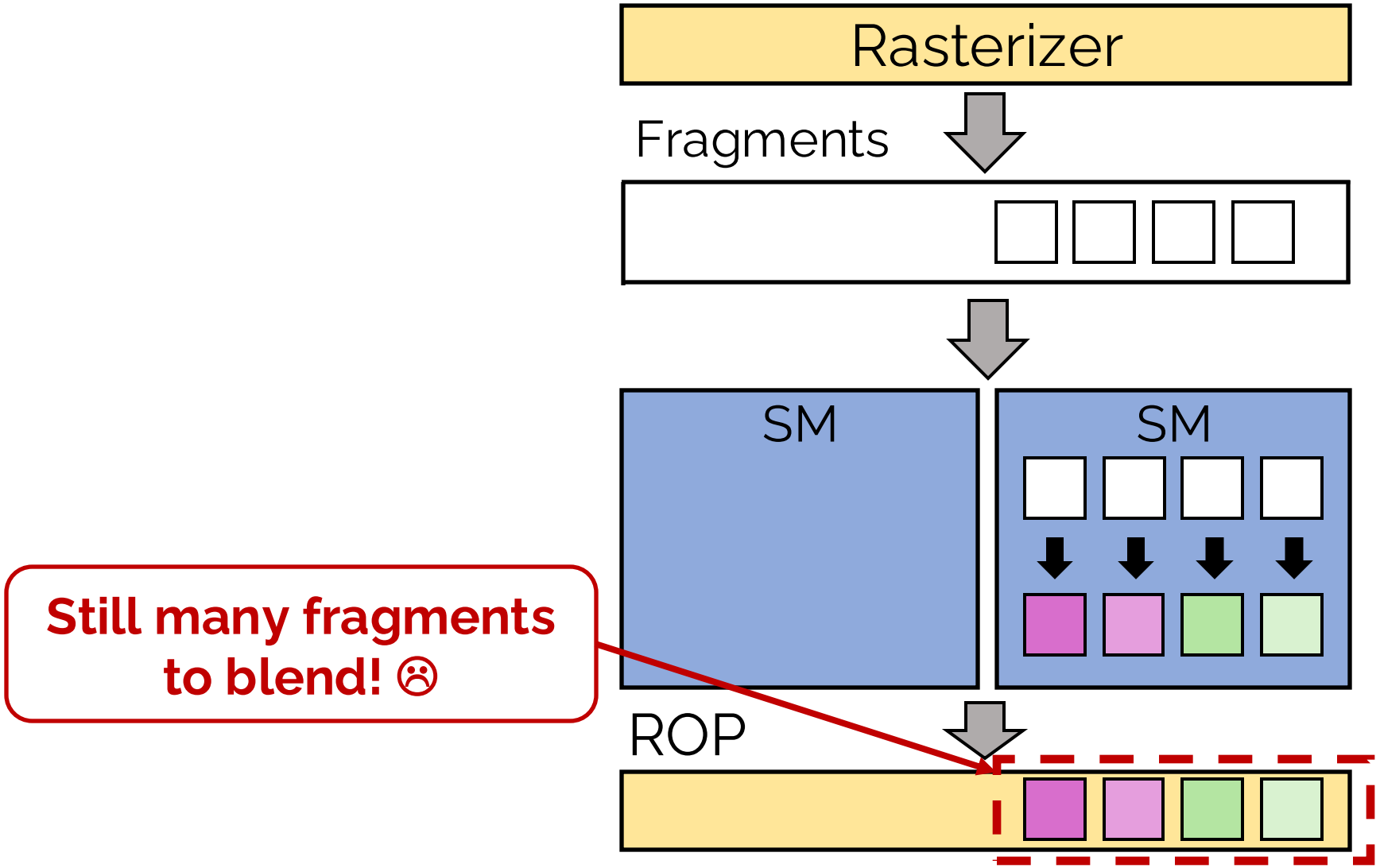
Proposal 1

Add **hardware support for early termination**

= Hardware-Based Early Termination (HET)



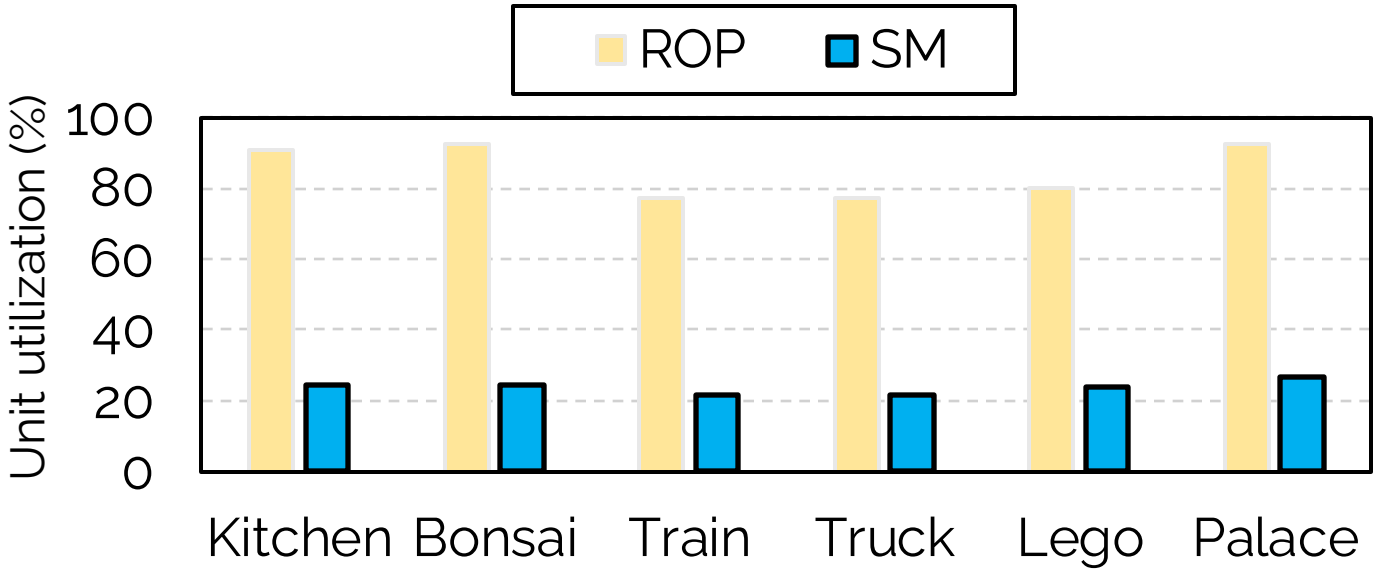
Limitations of Graphics Hardware



Limitations of Graphics Hardware

Observation 2

SMs are underutilized
due to back pressure



Limitations of Graphics Hardware

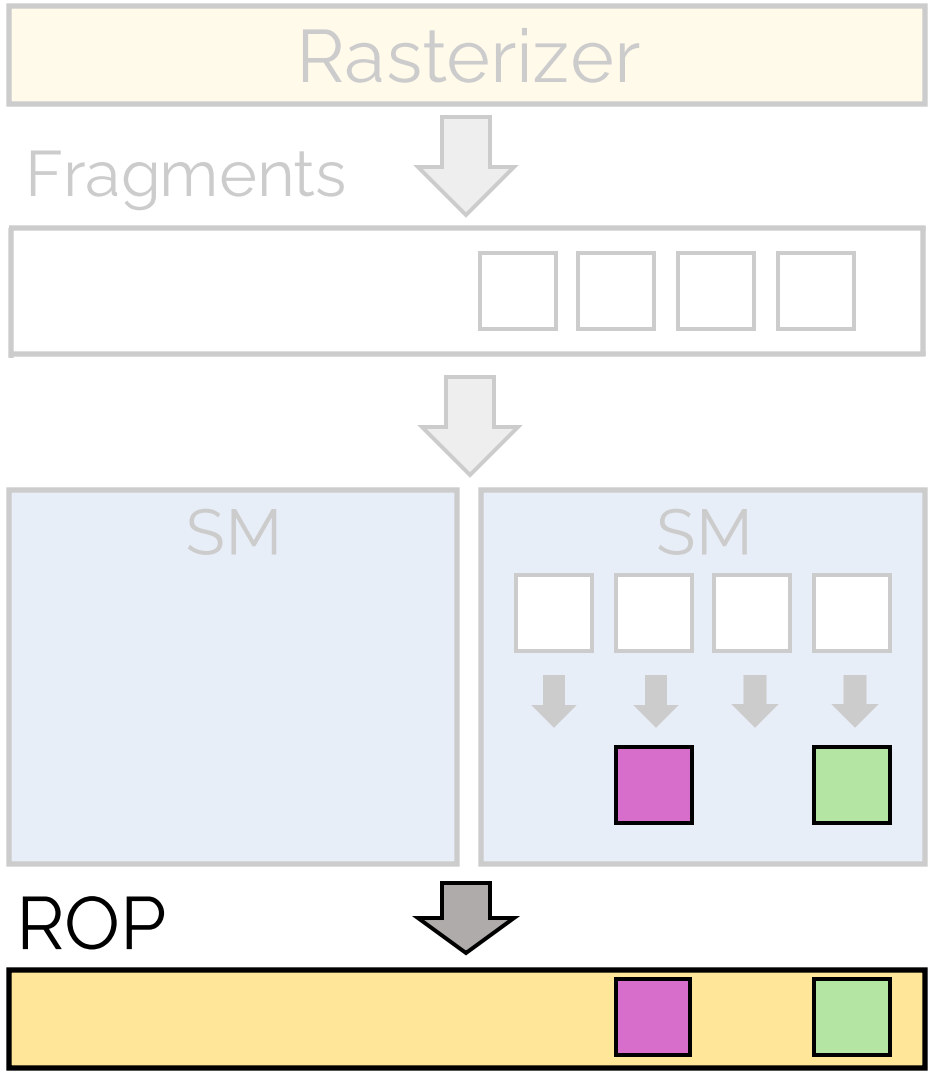
Observation 2

SMs are underutilized
due to back pressure

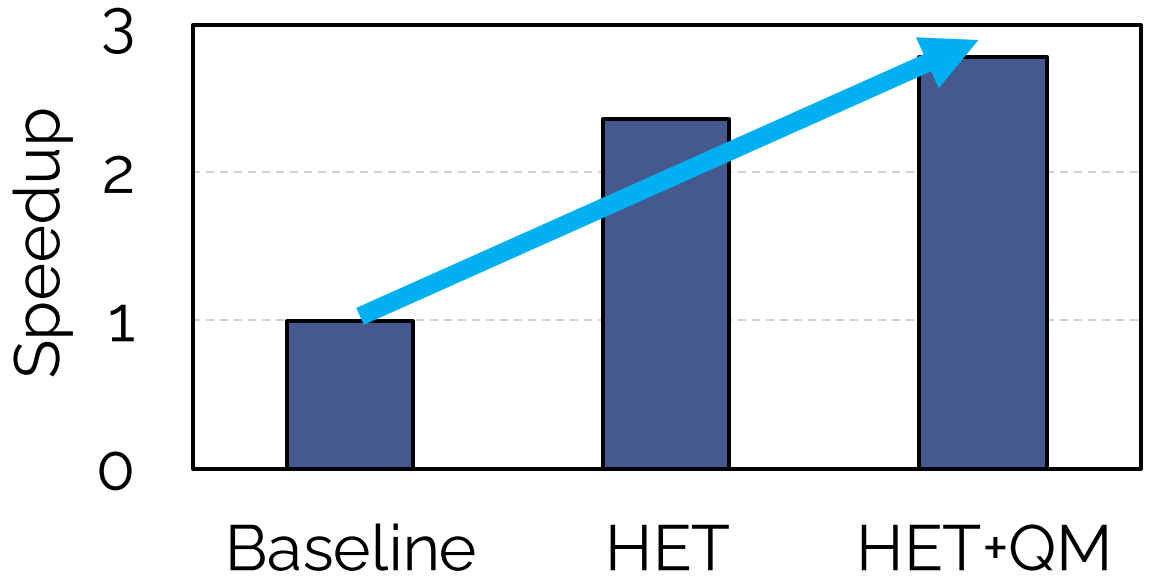
Proposal 2

Partially blend fragments
in SMs

= Quad Merging (QM)



Limitations of Graphics Hardware

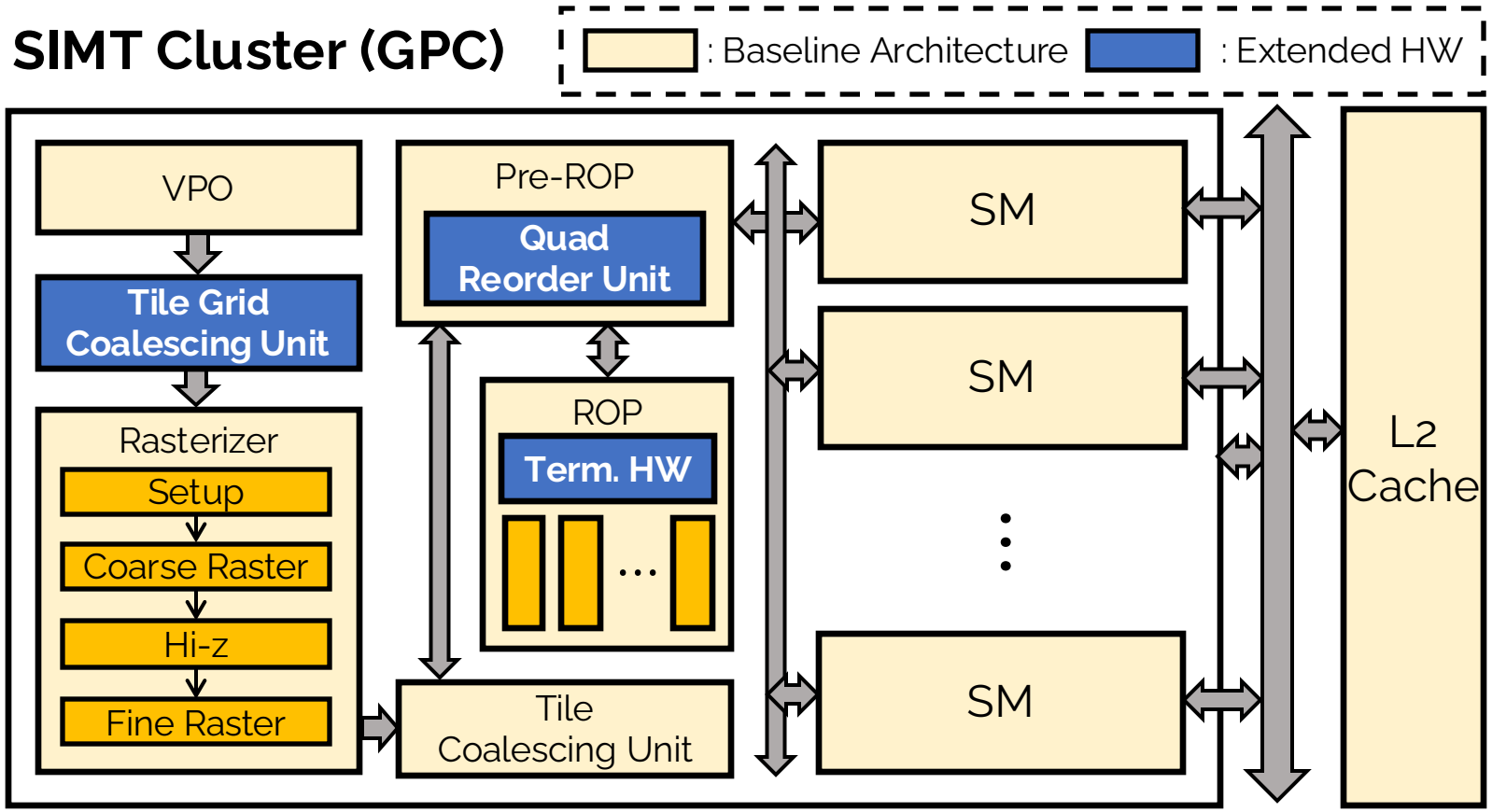


VR-Pipe improves rendering performance
by reducing ROP pressure! ☺

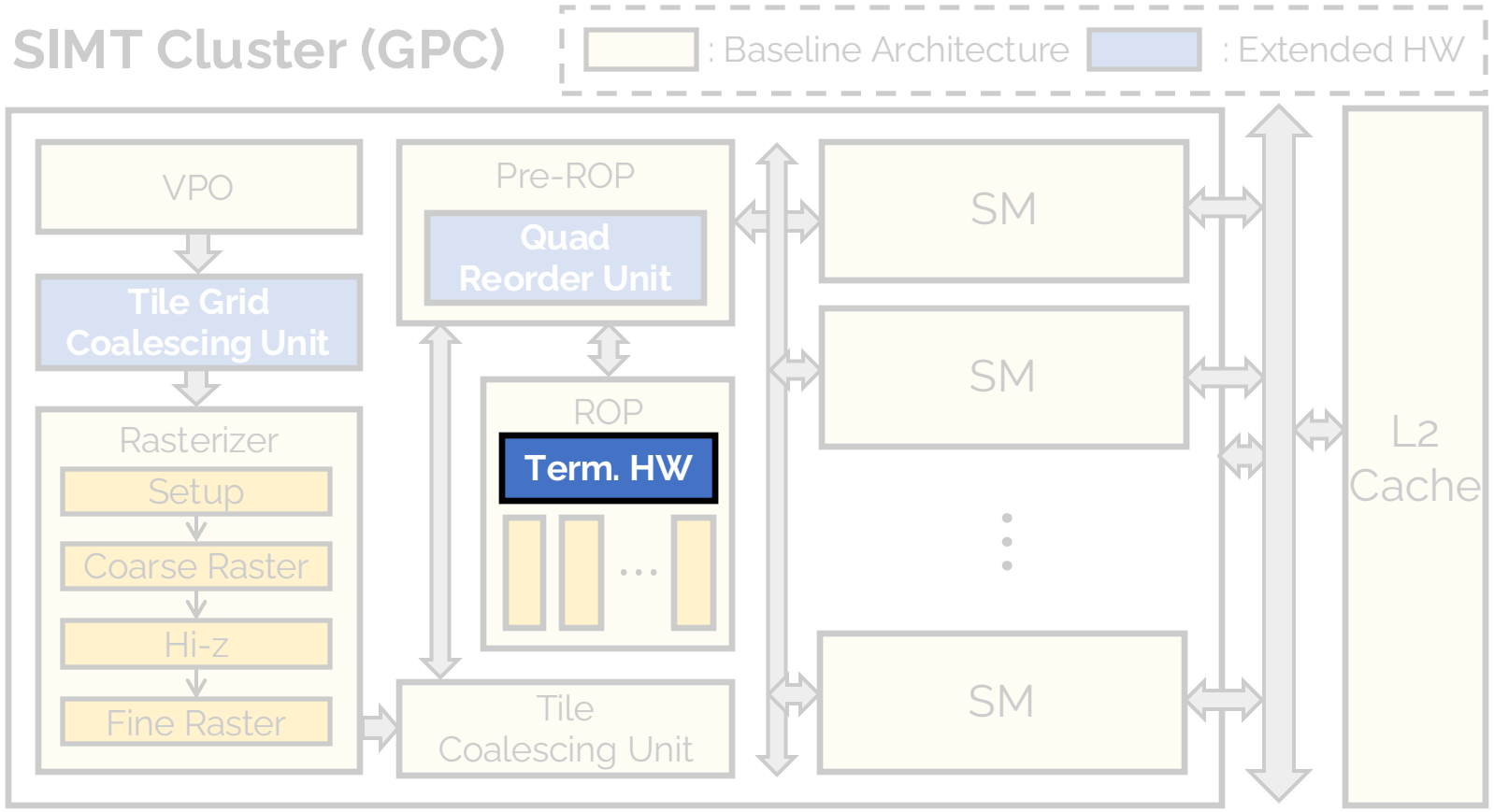
Outline

- **Background**
  - 3D Gaussian Splatting (3DGS)
  - Hardware Graphics Pipeline
- **Limitations of Graphics Hardware**
- **VR-Pipe: Graphics Hardware Extension for Volume Rendering**
  - Quad Merging with Multi-Granular Tile Binning
  - Hardware Support for Early Termination
- **Evaluation**
- **Conclusion**

VR-Pipe: GPU Extension for Volume Rendering

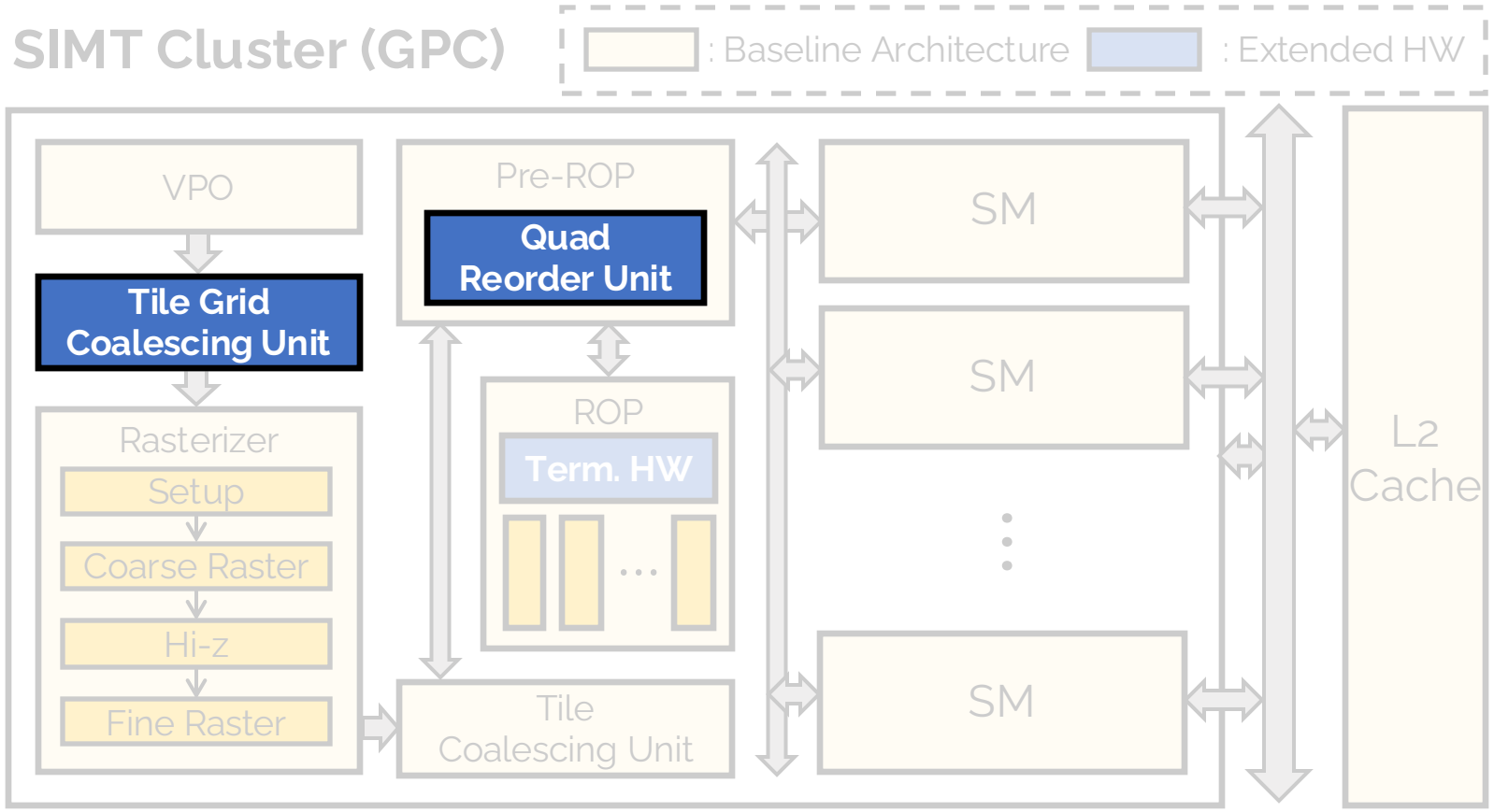


VR-Pipe: GPU Extension for Volume Rendering



Hardware-Based Early Termination

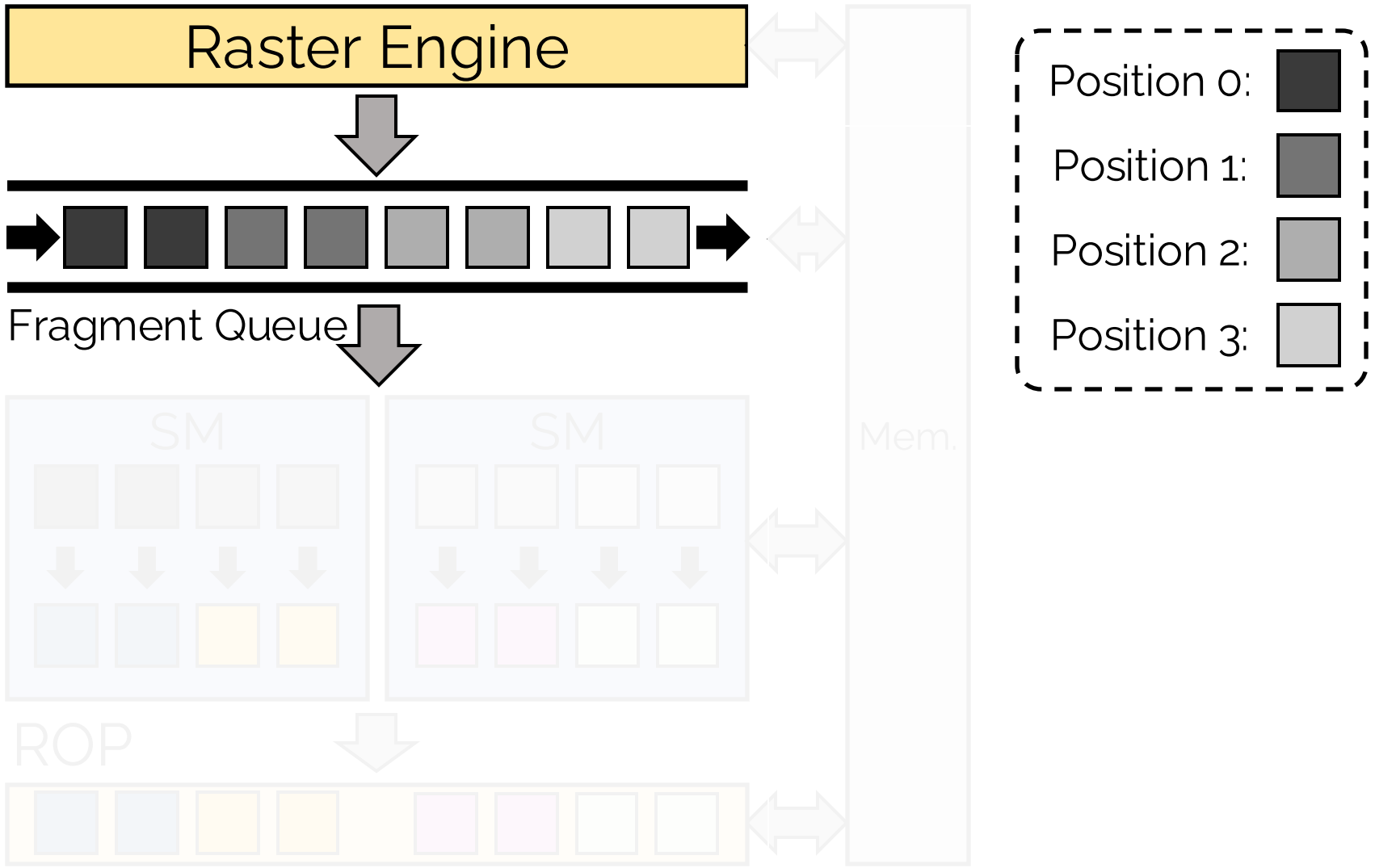
VR-Pipe: GPU Extension for Volume Rendering



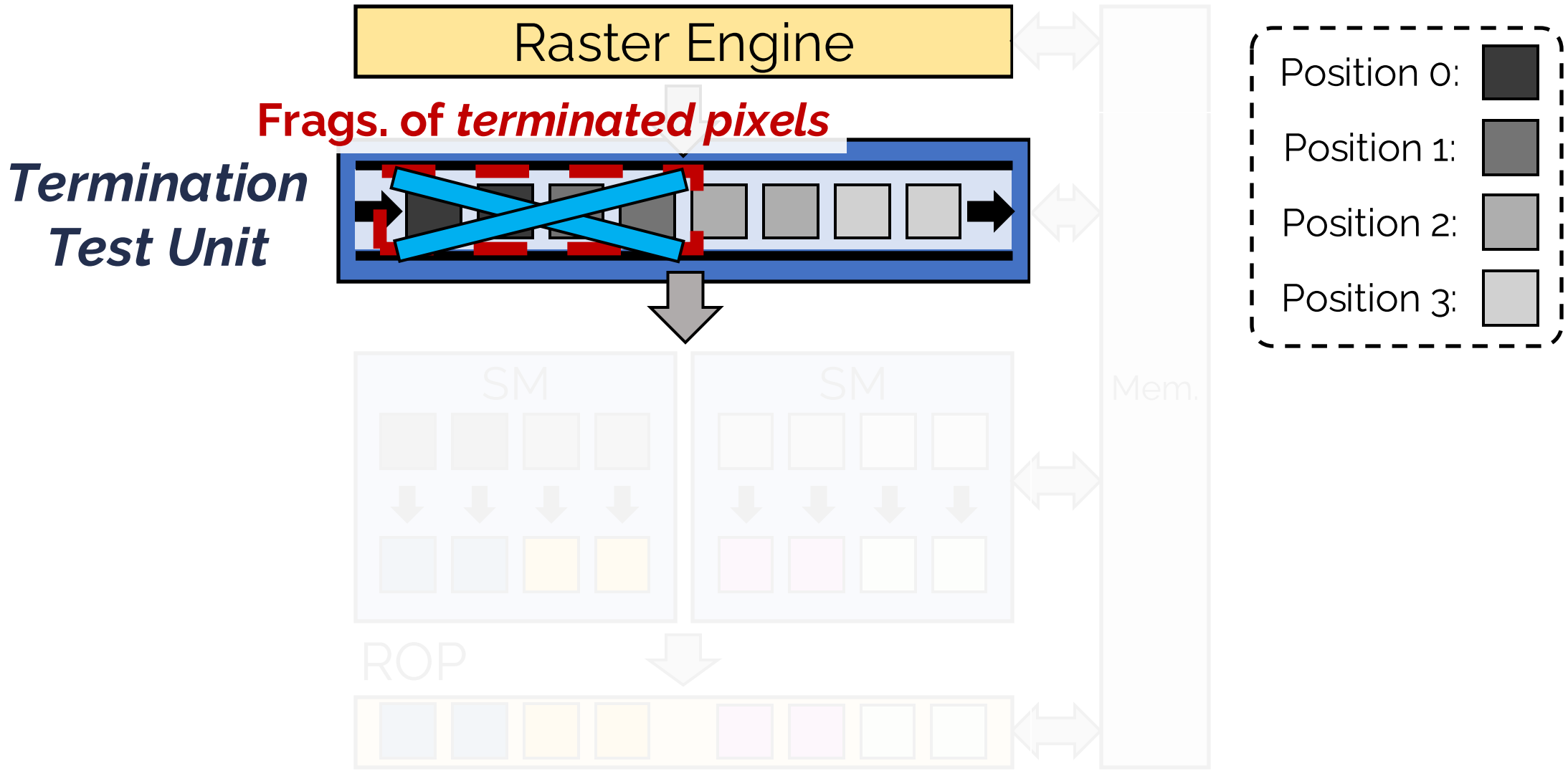
Hardware-Based Early Termination

Quad Merging w/ Multi-Granular Tile Binning

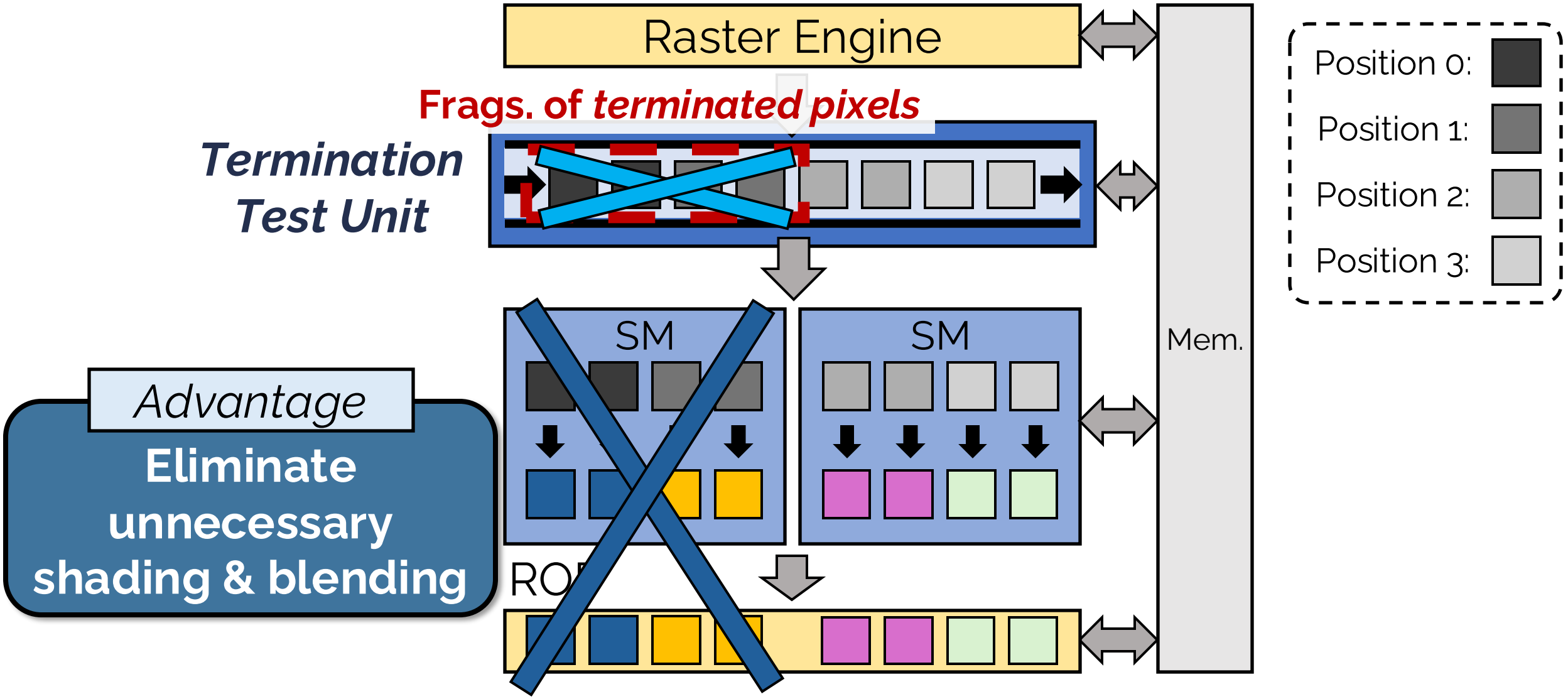
Hardware-Based Early Termination



Hardware-Based Early Termination



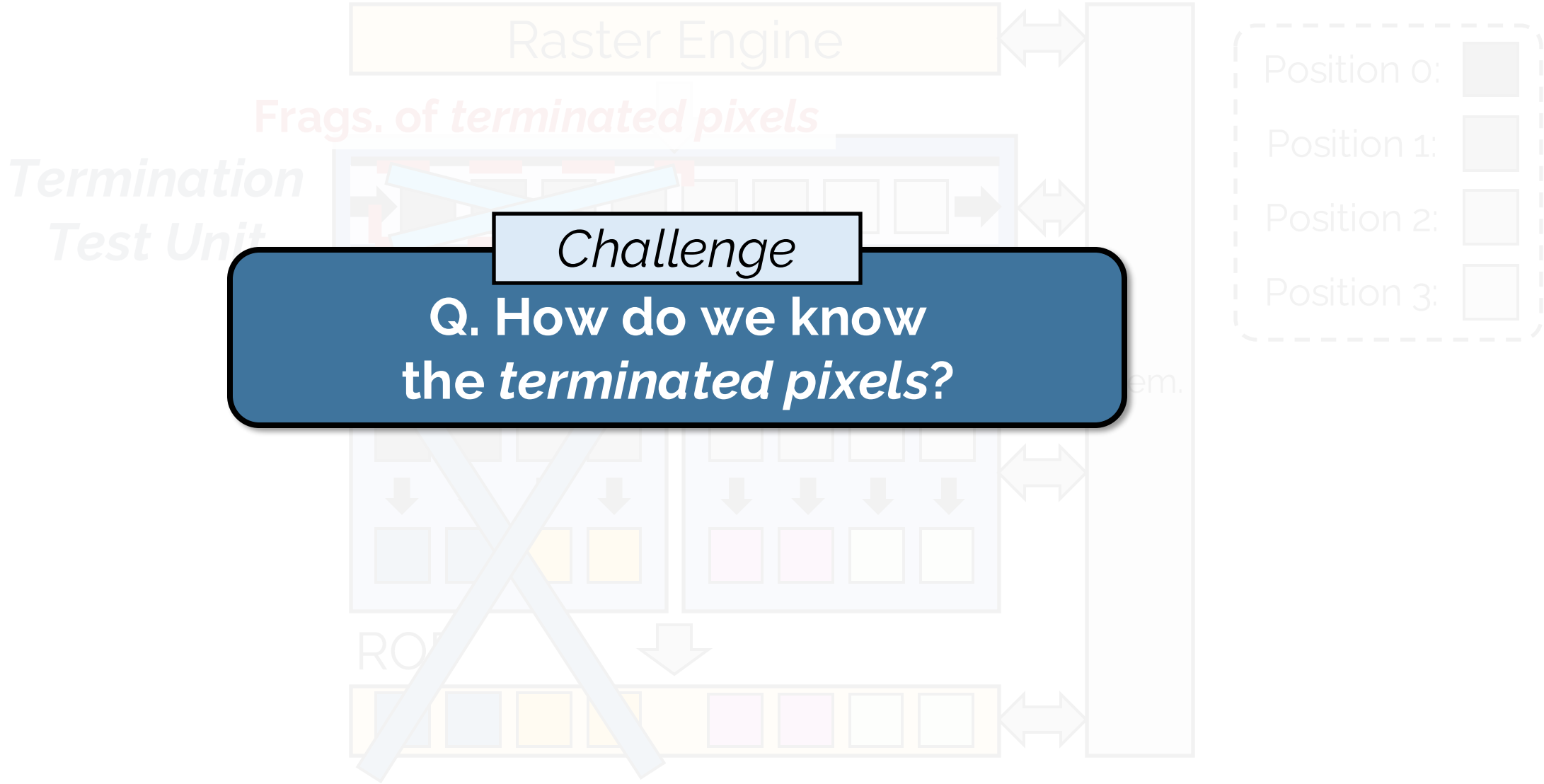
Hardware-Based Early Termination



Advantage

Eliminate unnecessary shading & blending

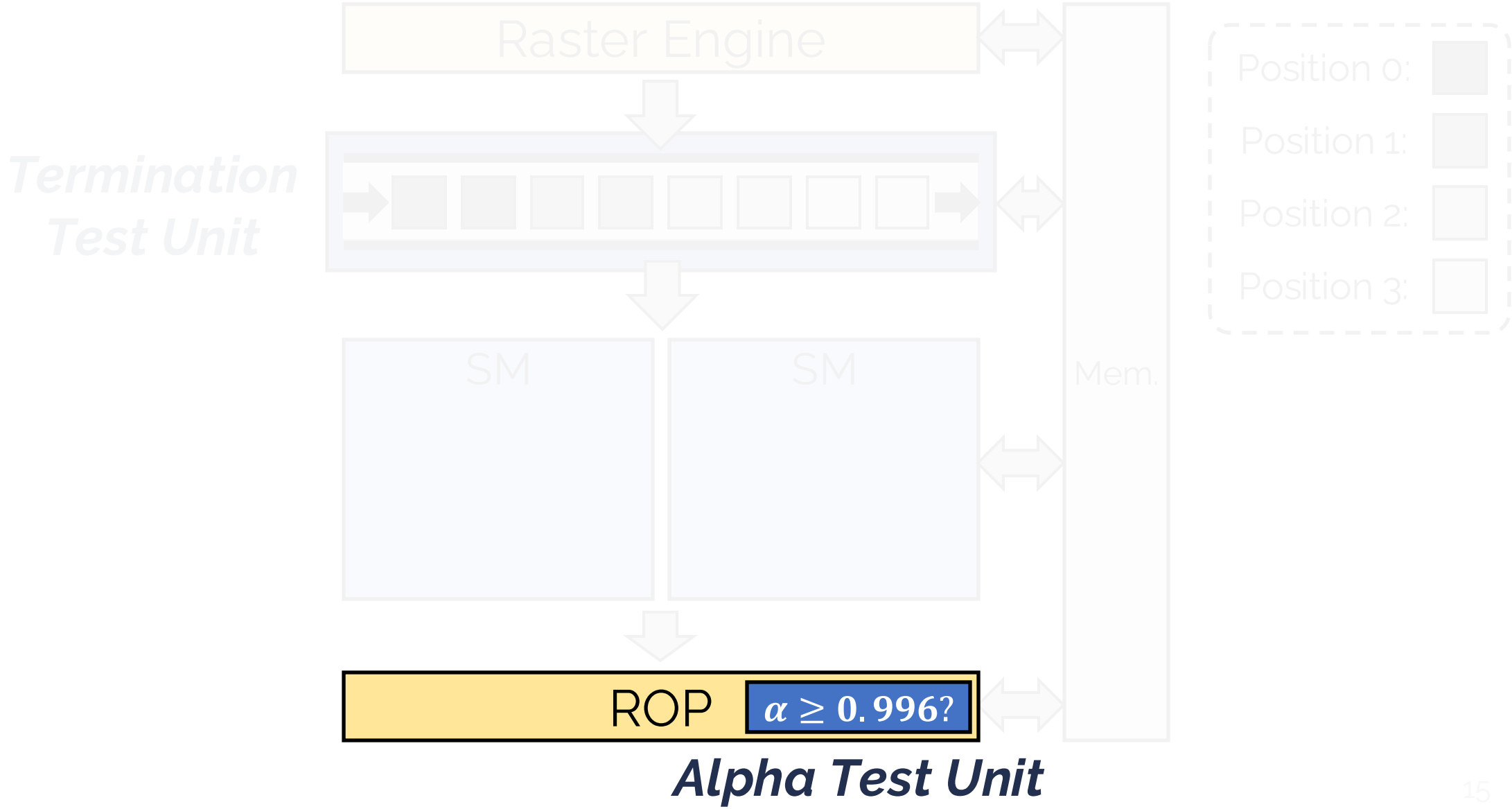
Hardware-Based Early Termination



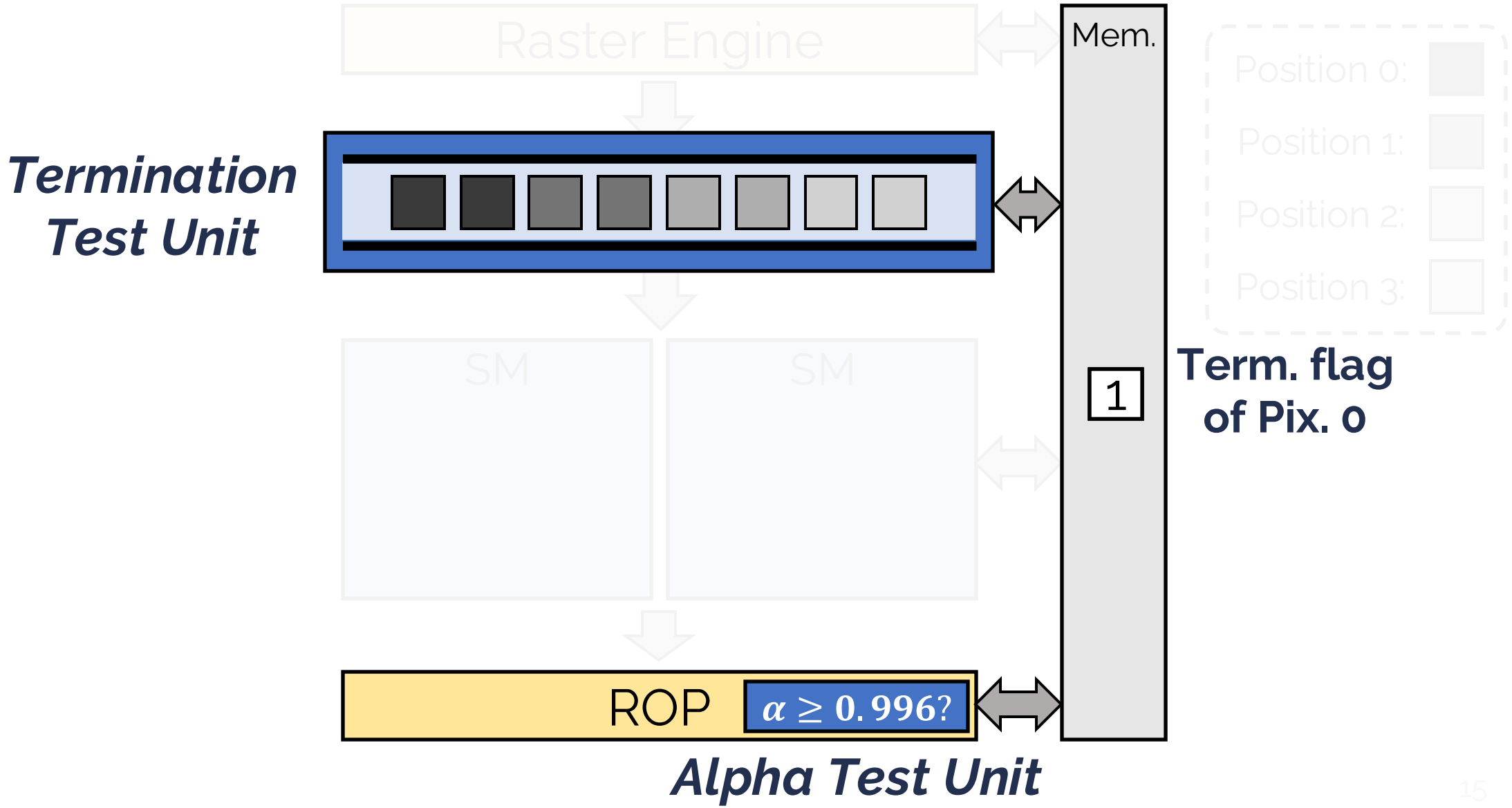
Challenge

Q. How do we know the *terminated pixels*?

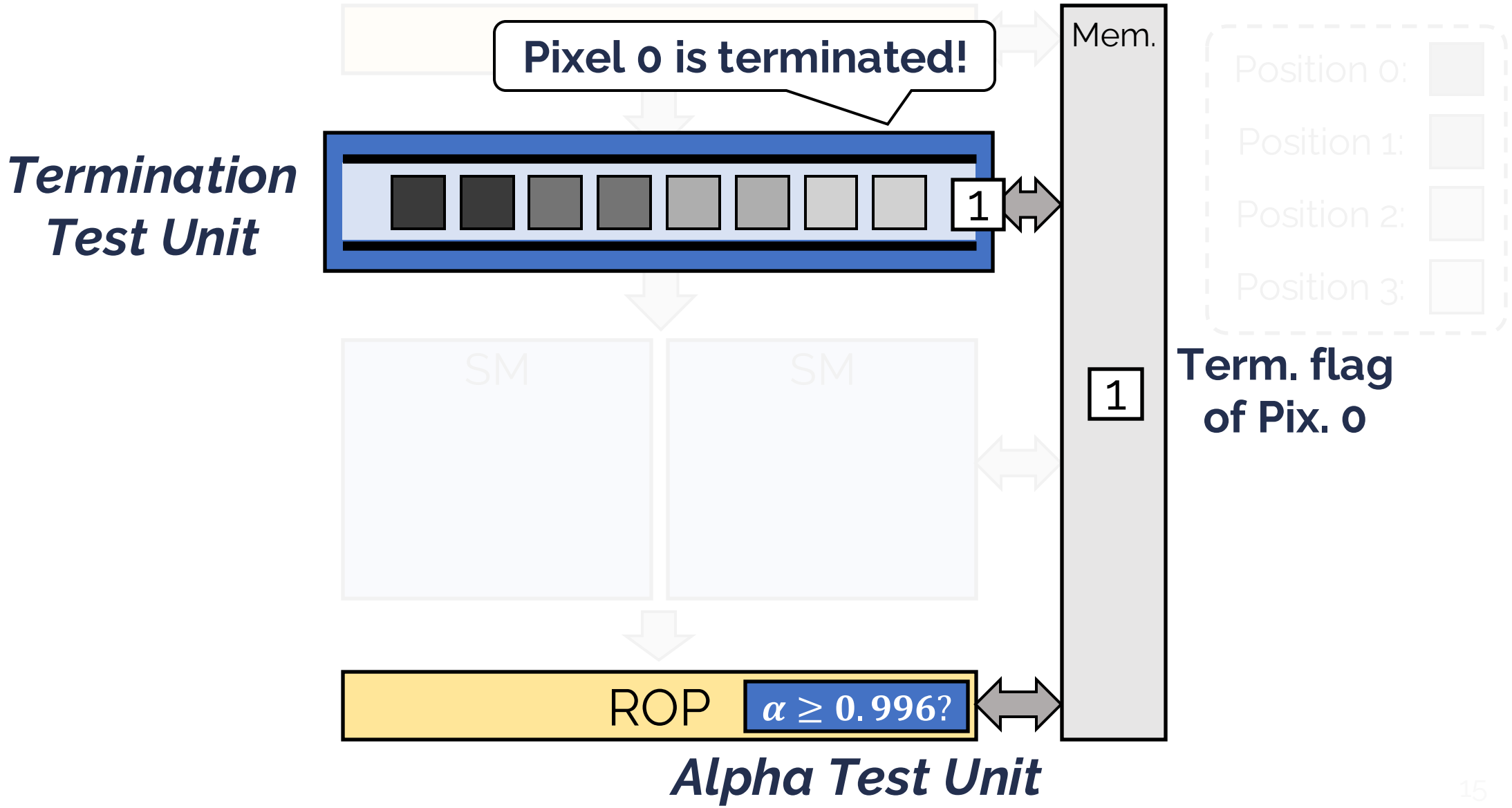
Hardware-Based Early Termination



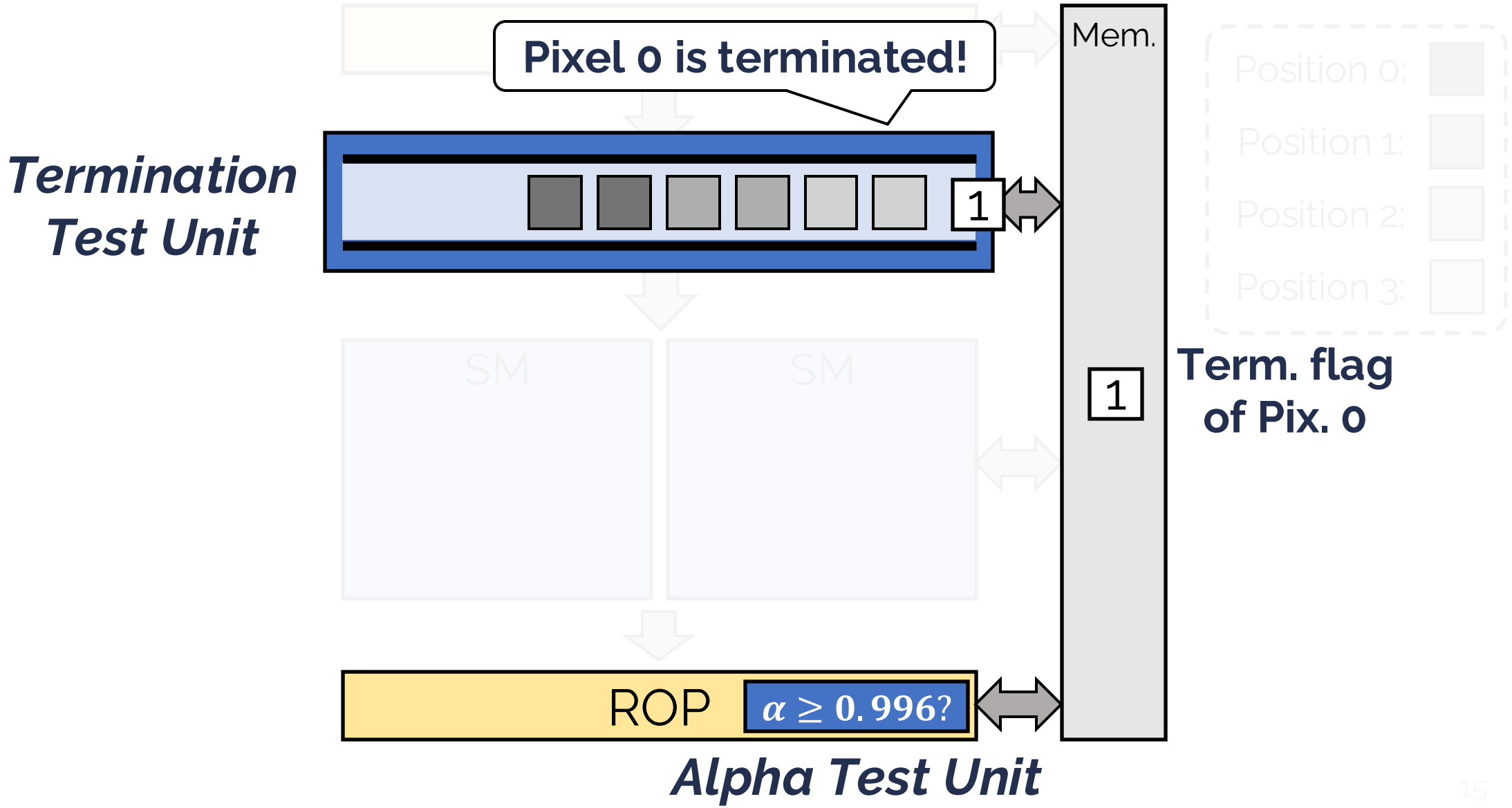
Hardware-Based Early Termination



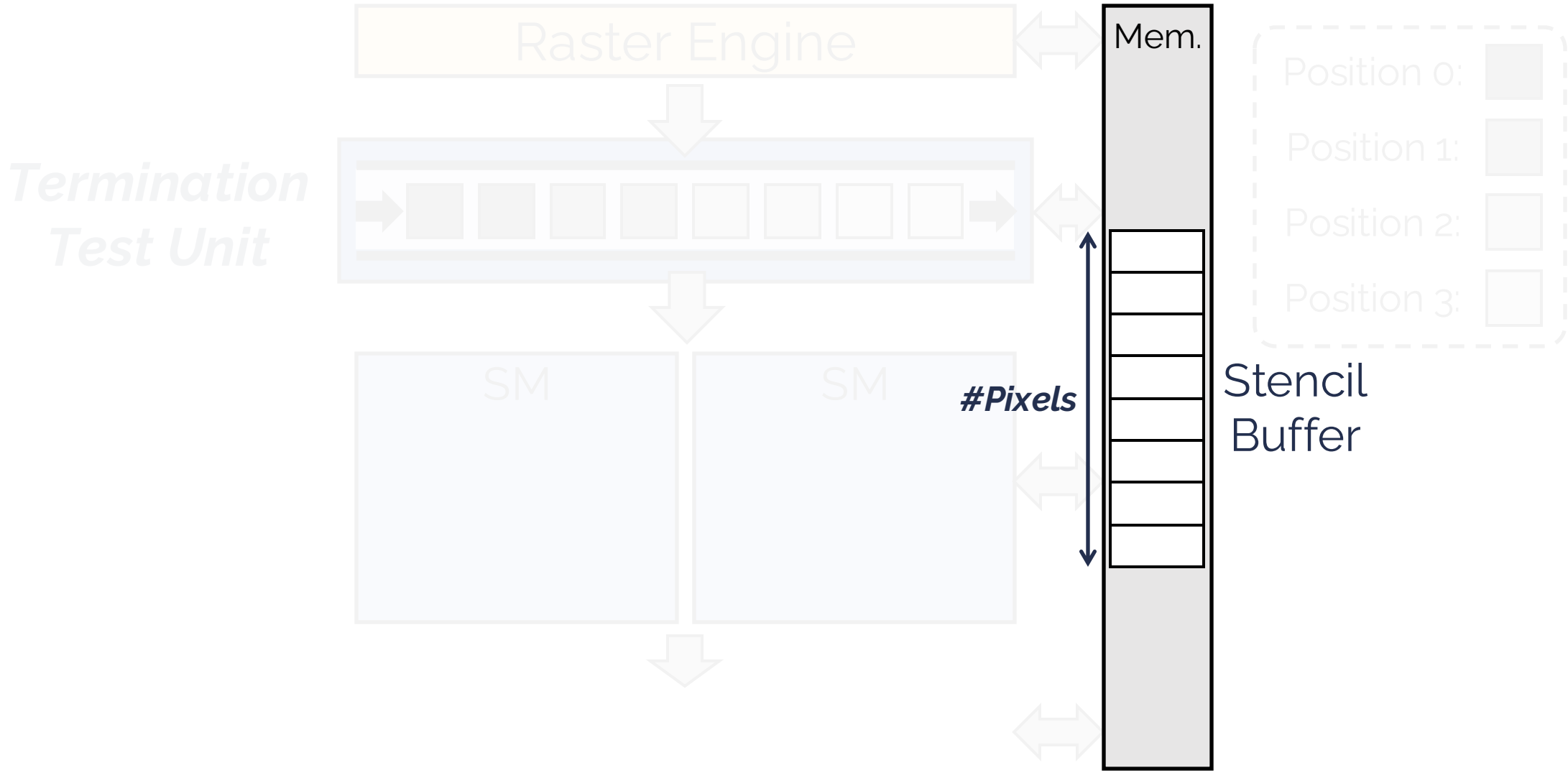
Hardware-Based Early Termination



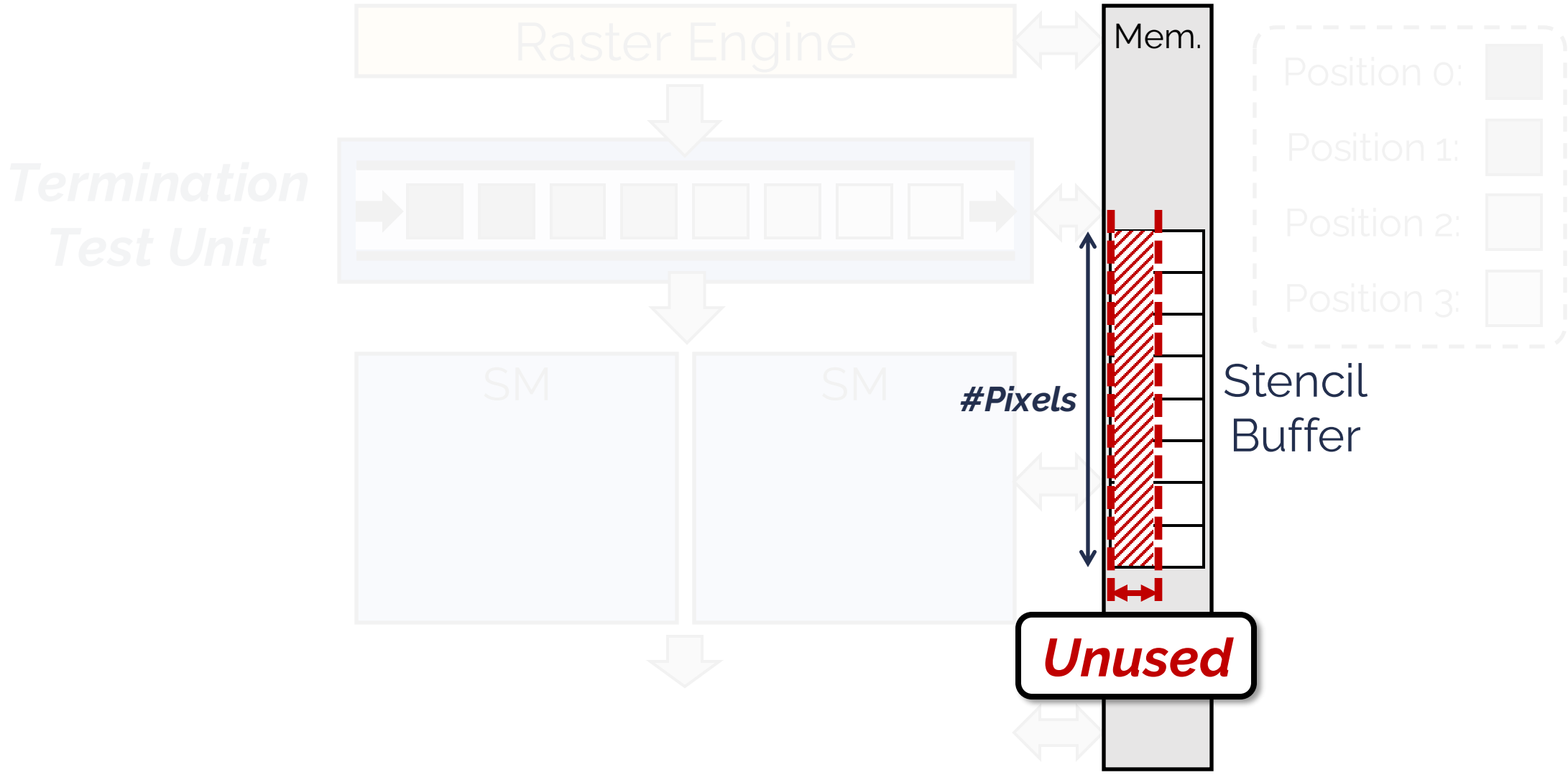
Hardware-Based Early Termination



Hardware-Based Early Termination



Hardware-Based Early Termination

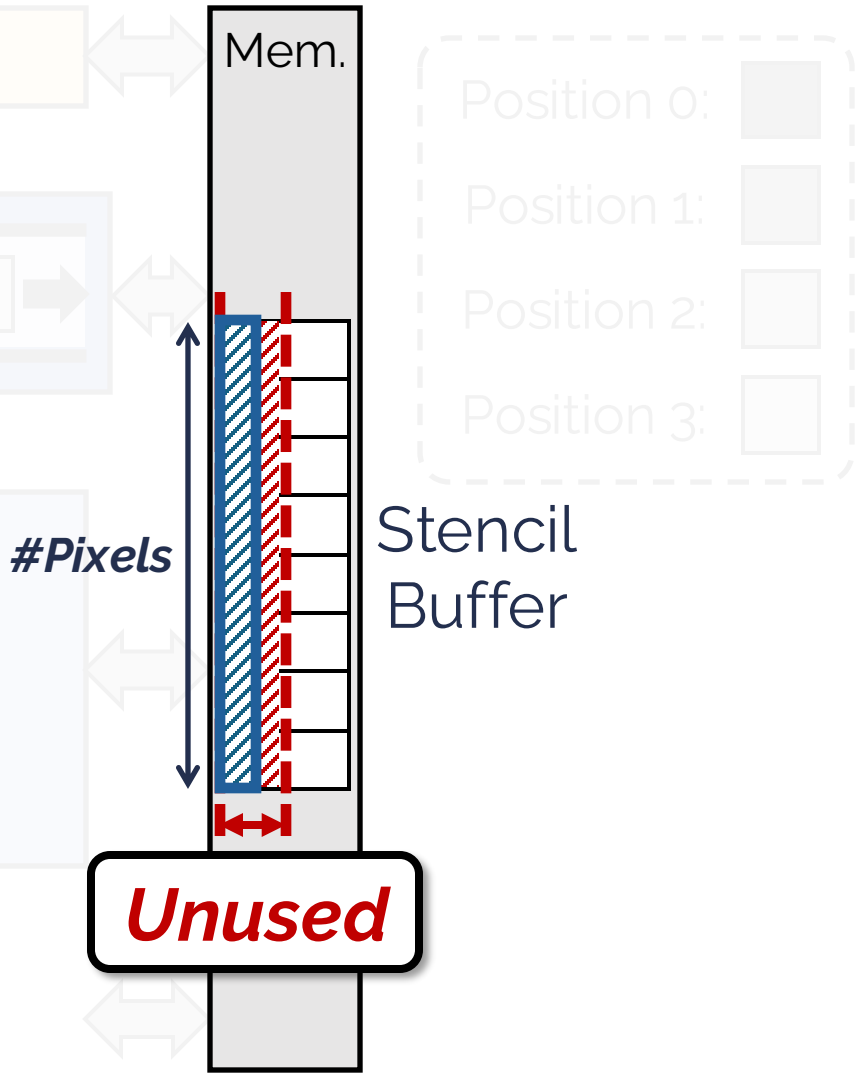


Hardware-Based Early Termination

Raster Engine

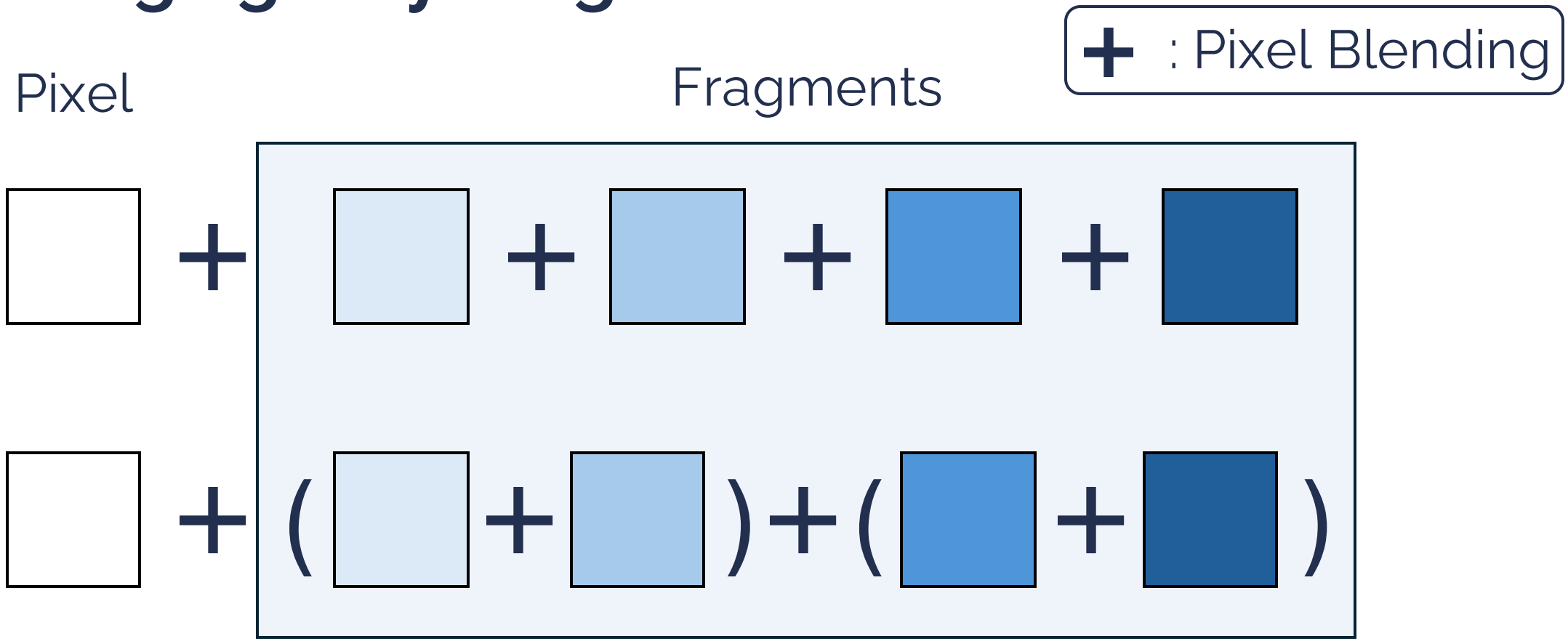
Termination Test Unit

Repurpose one bit of a stencil value as a **termination flag**

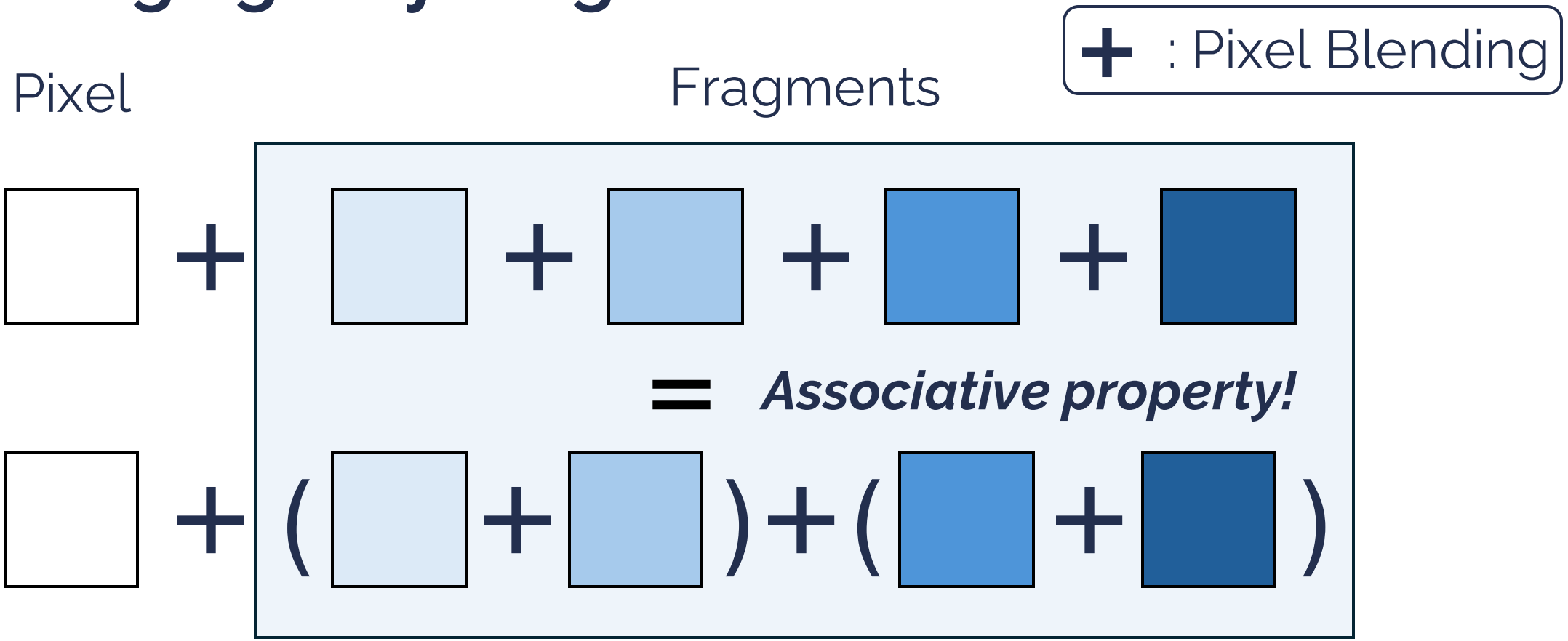


Quad Merging: Key Insight

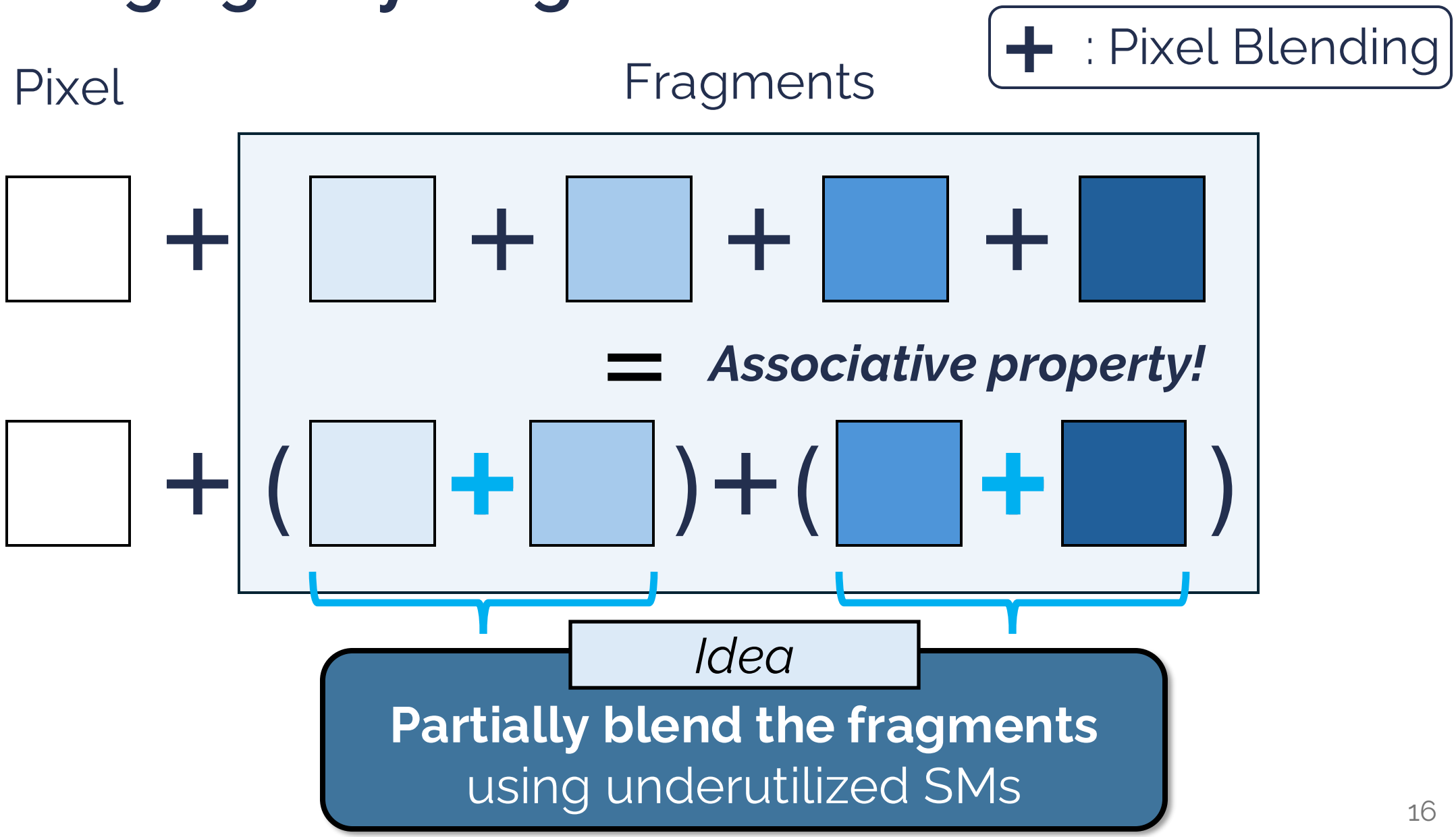
Quad Merging: Key Insight



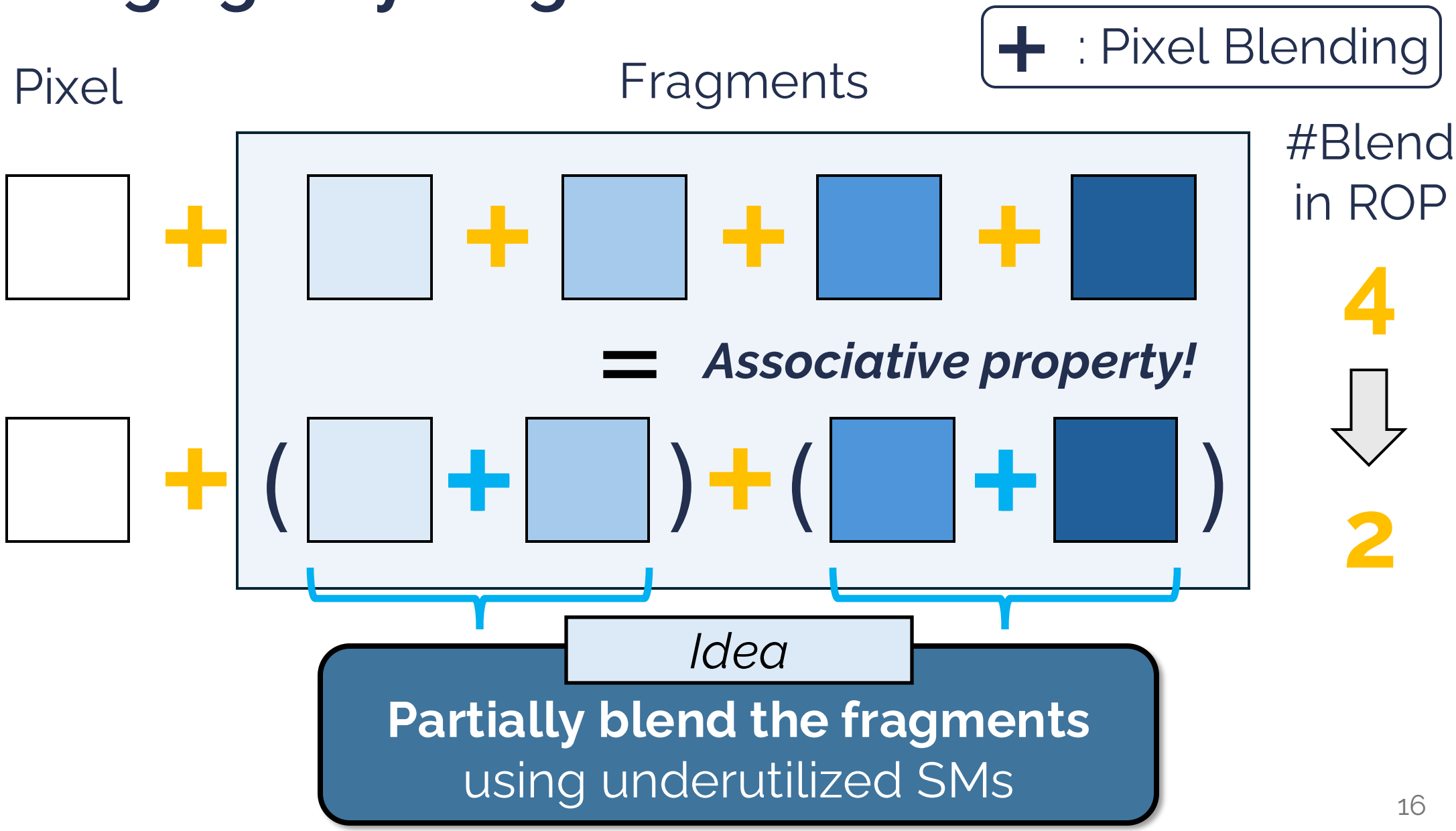
Quad Merging: Key Insight



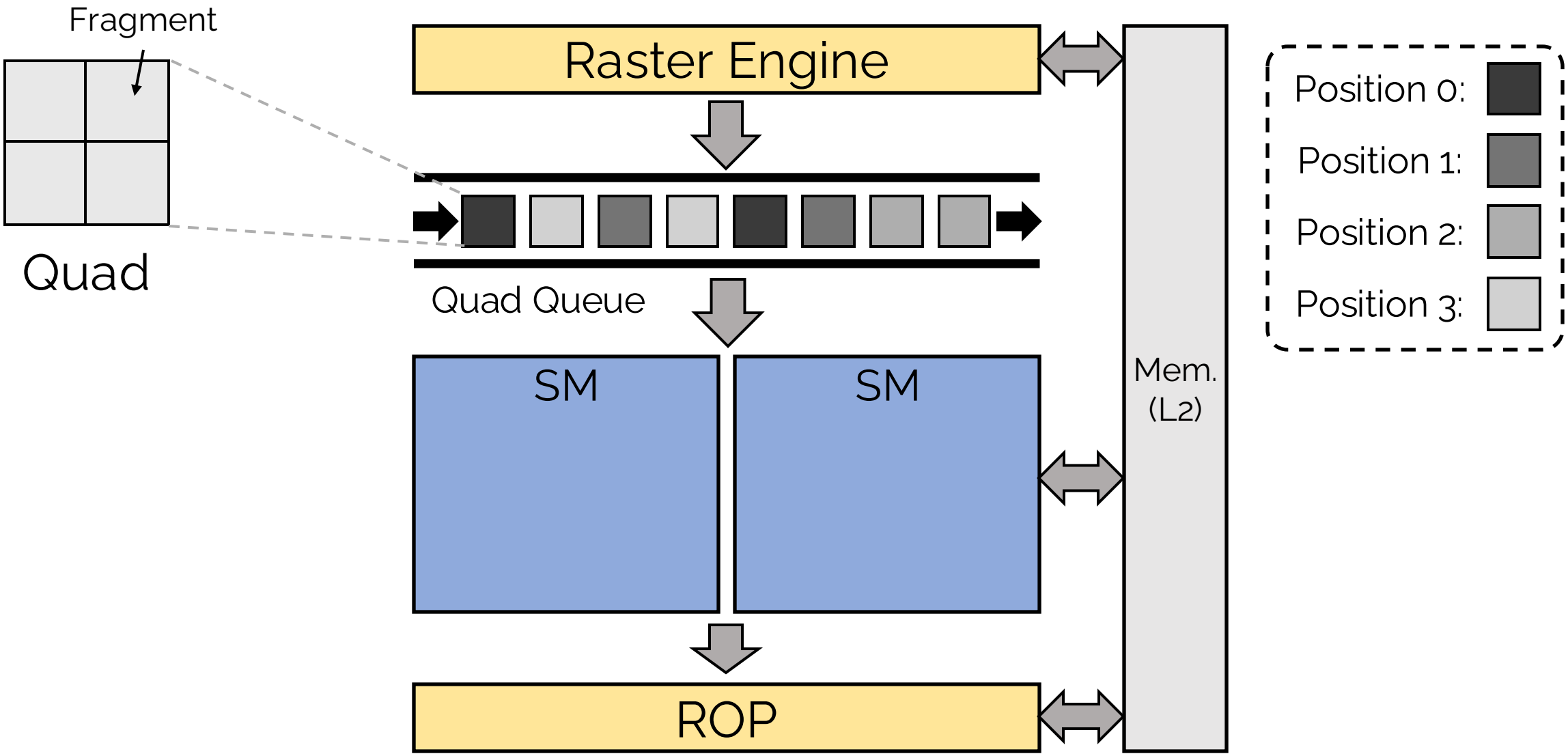
Quad Merging: Key Insight



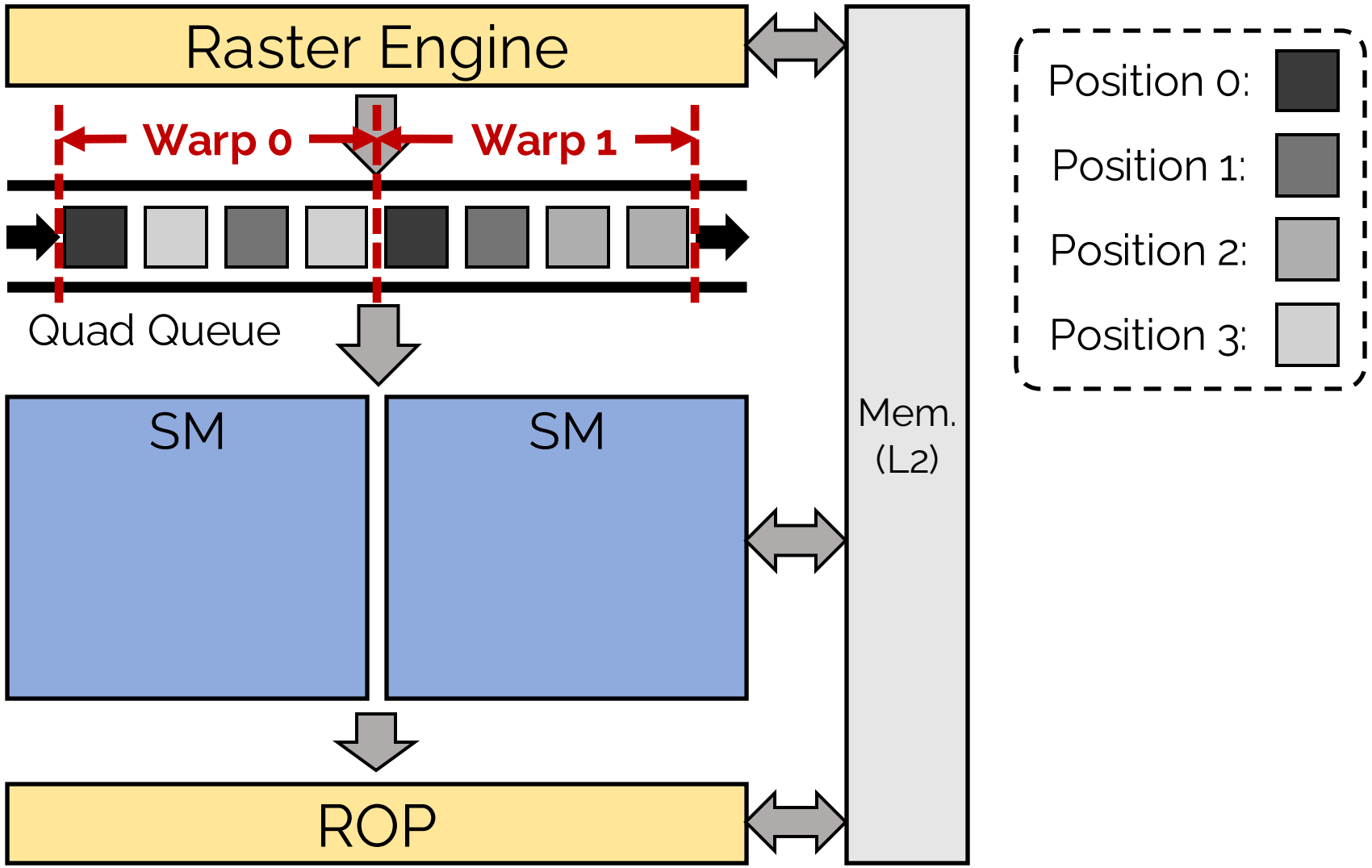
Quad Merging: Key Insight



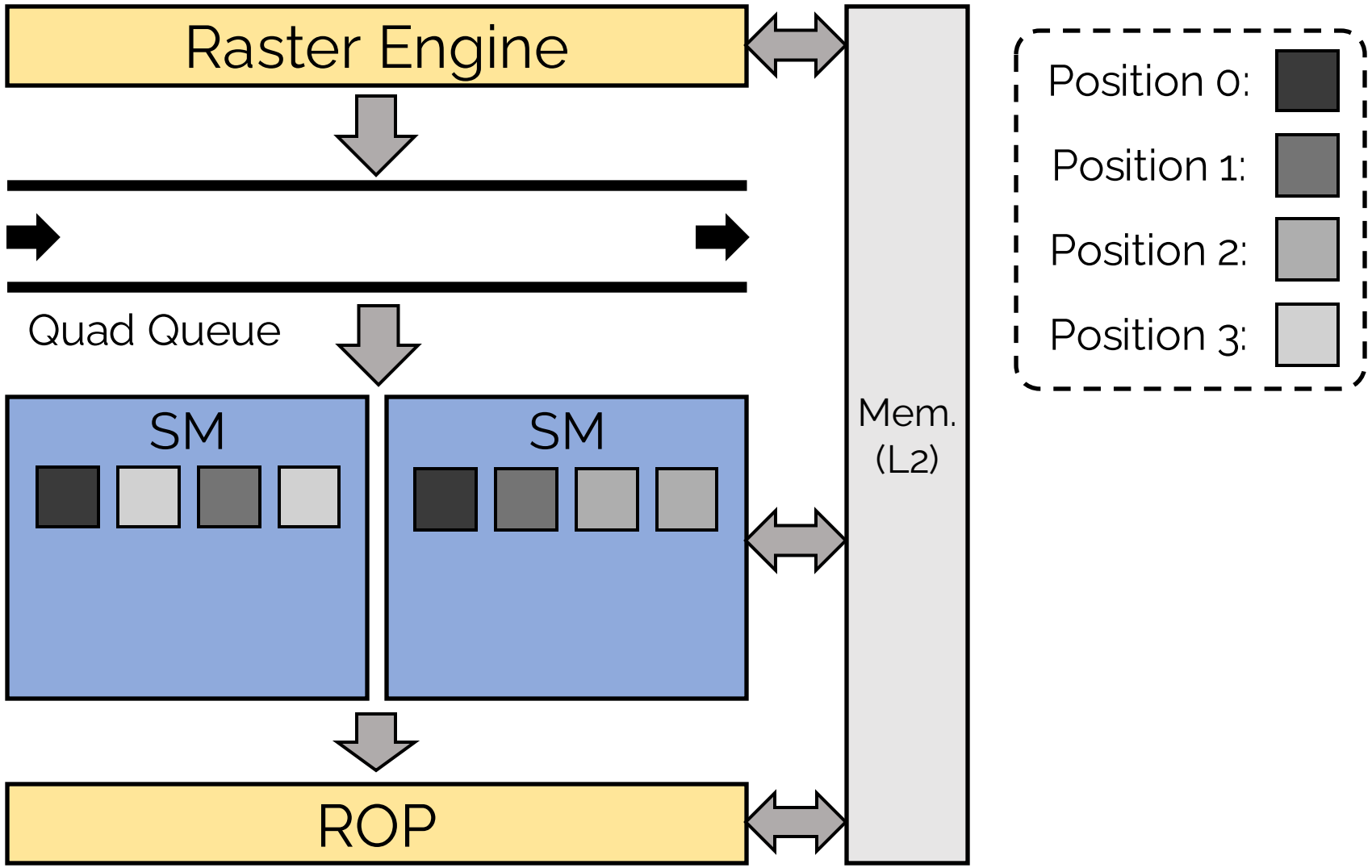
Quad Merging: Challenge



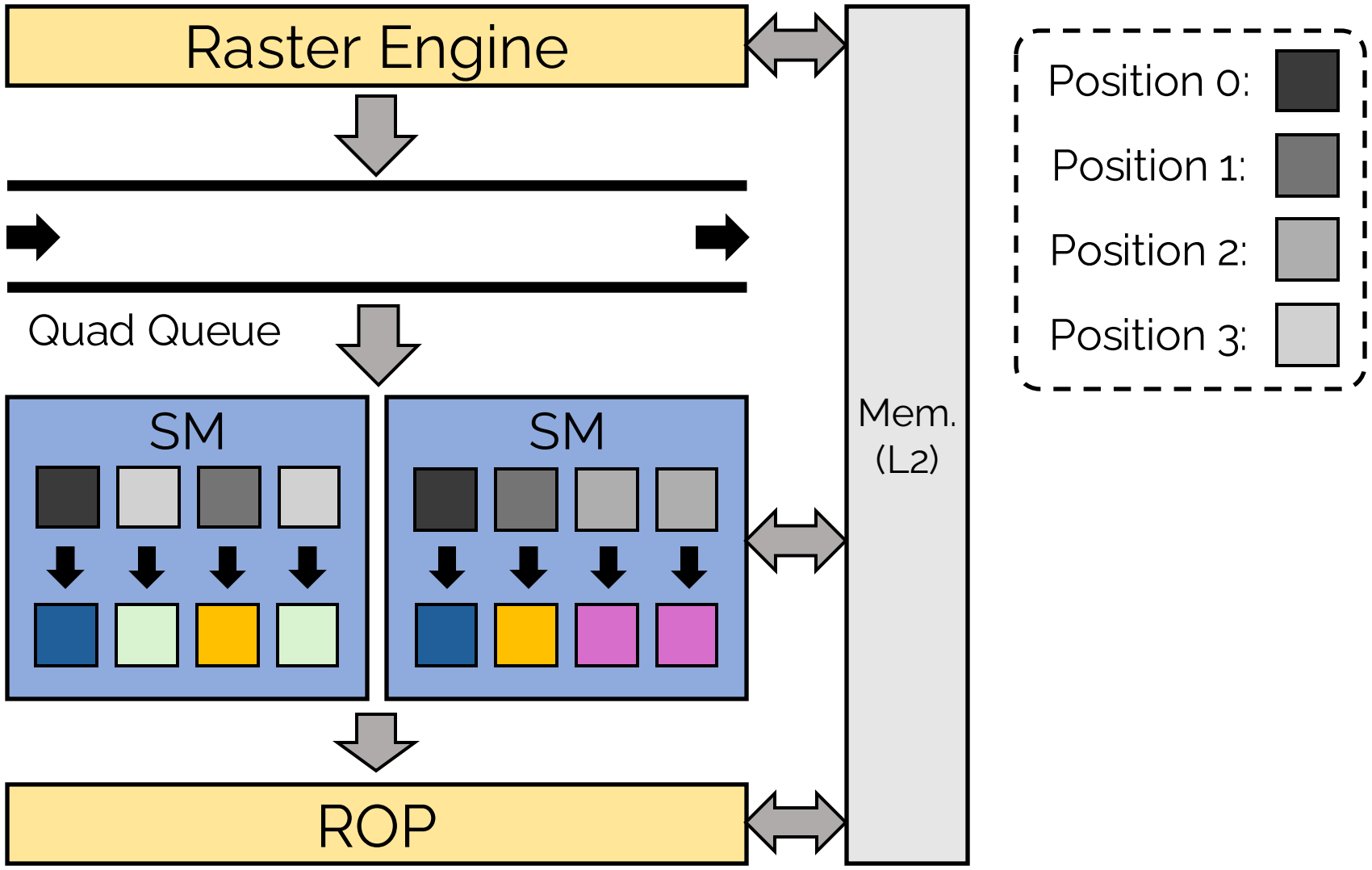
Quad Merging: Challenge



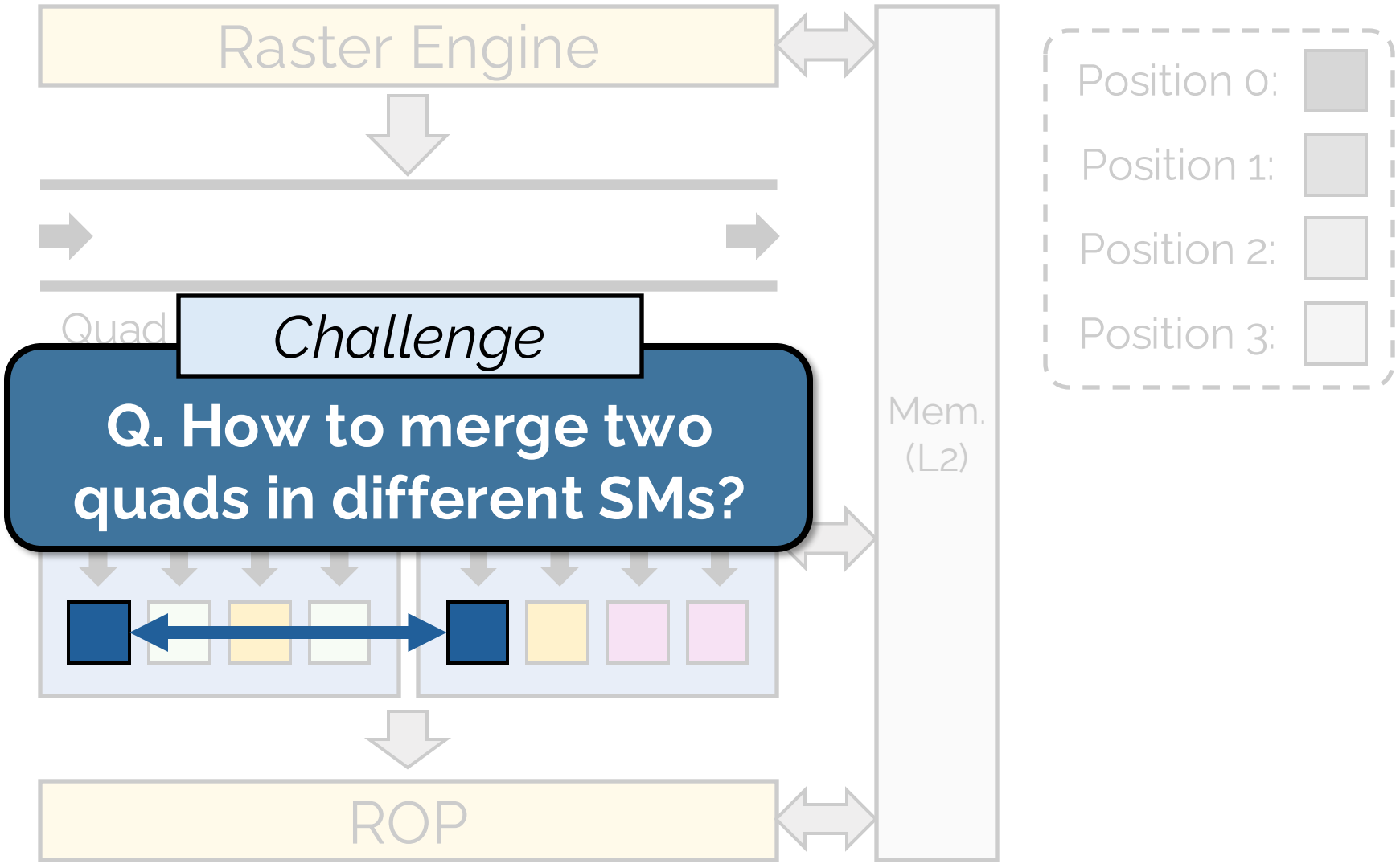
Quad Merging: Challenge



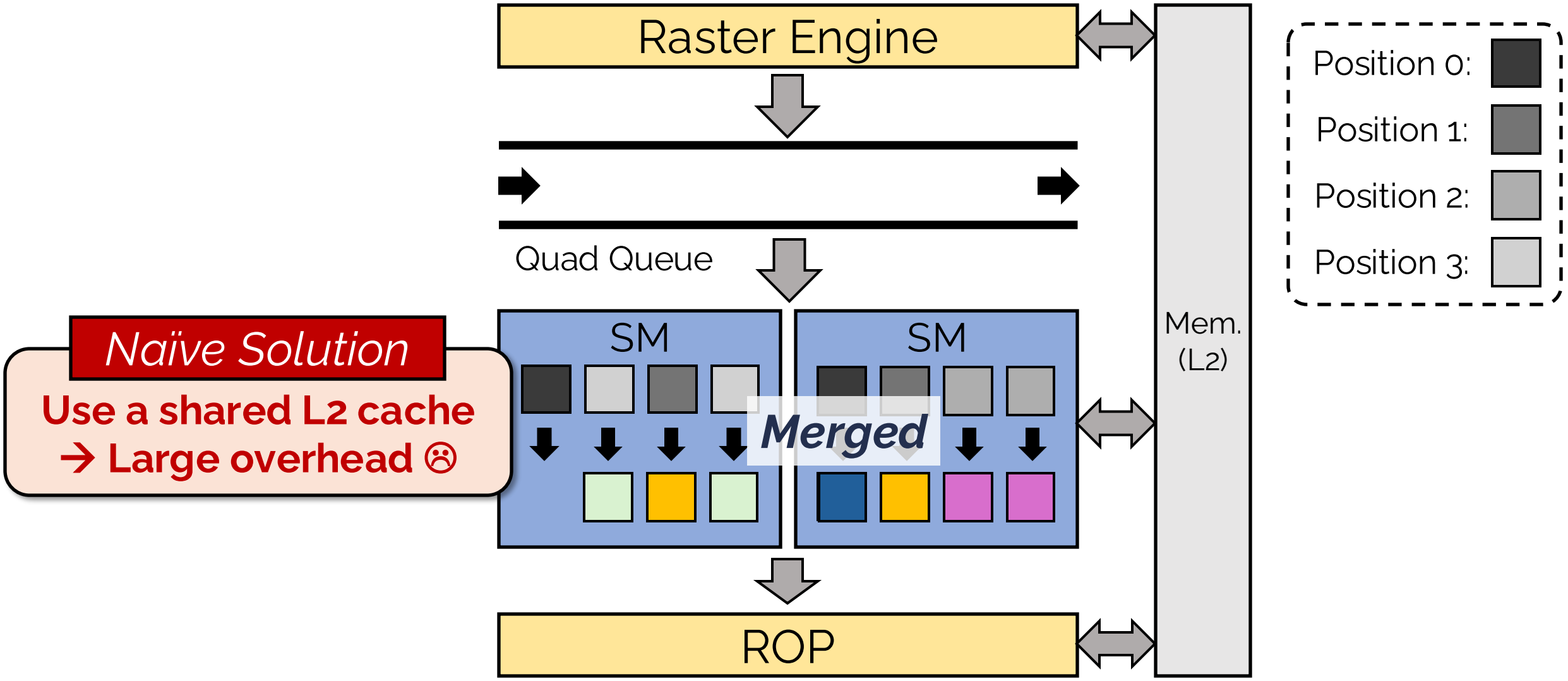
Quad Merging: Challenge



Quad Merging: Challenge



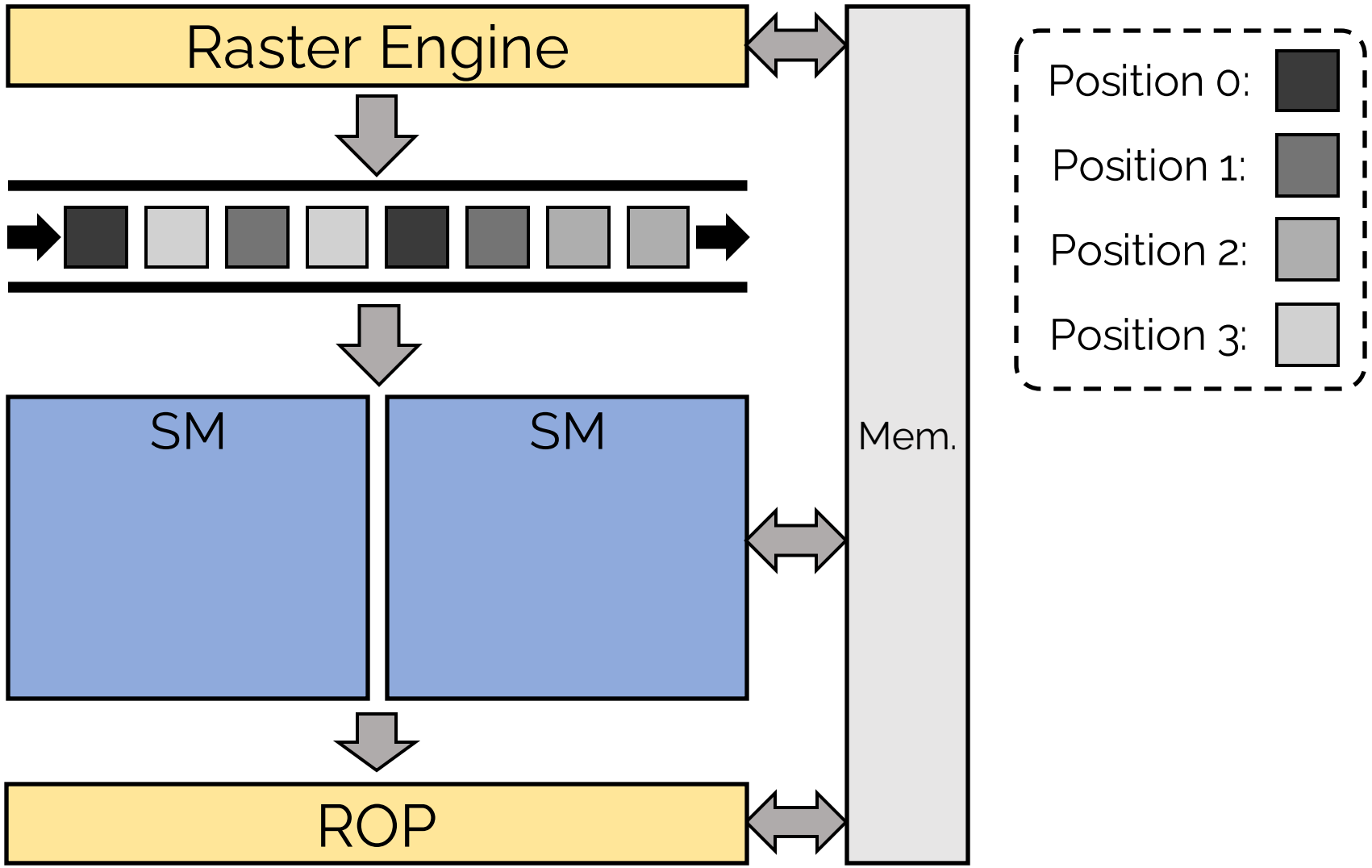
Quad Merging: Challenge



Naïve Solution

Use a shared L2 cache
→ Large overhead ☹

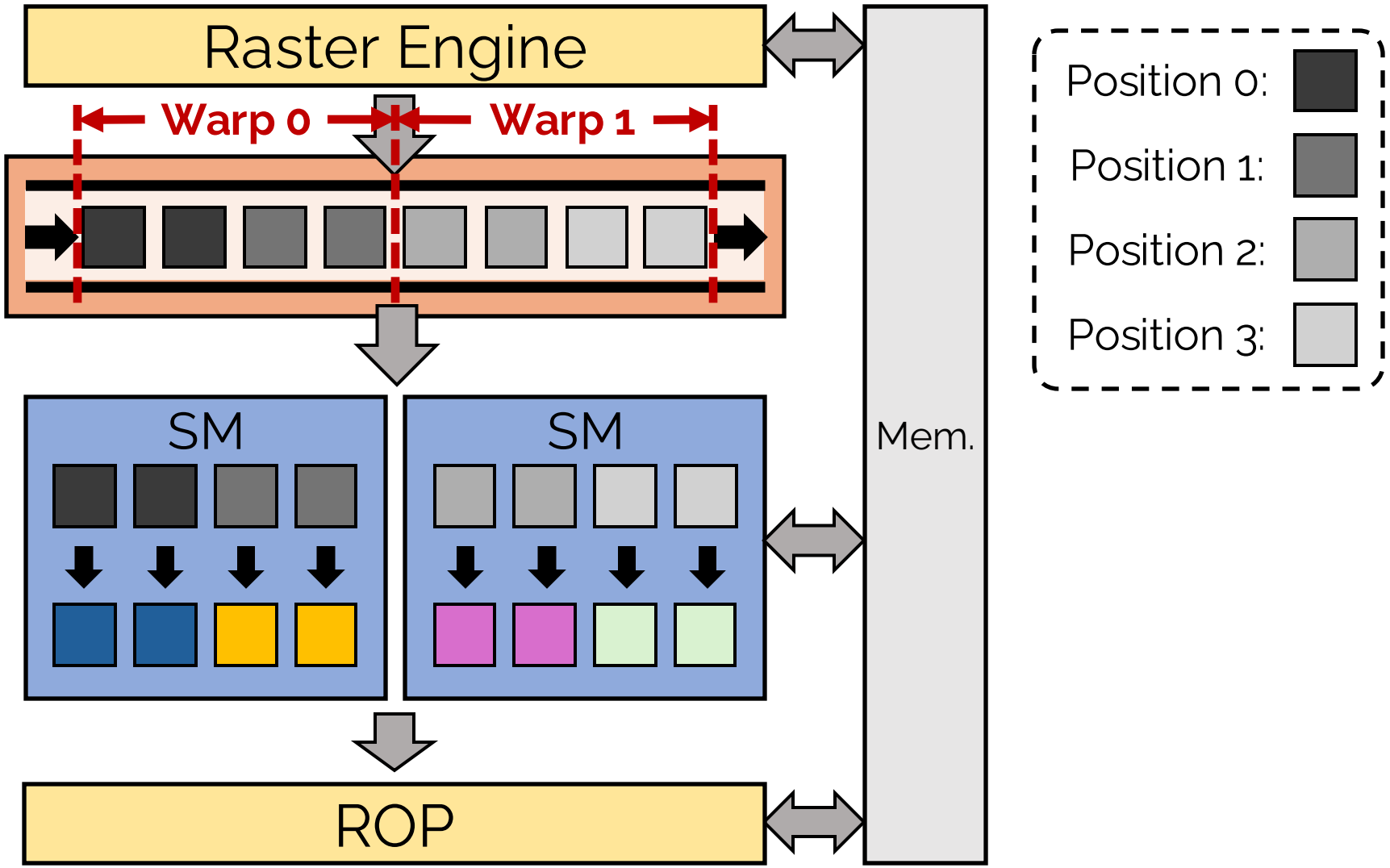
Quad Merging



Quad Merging

Quad Reorder Unit

1) Reorder the quads

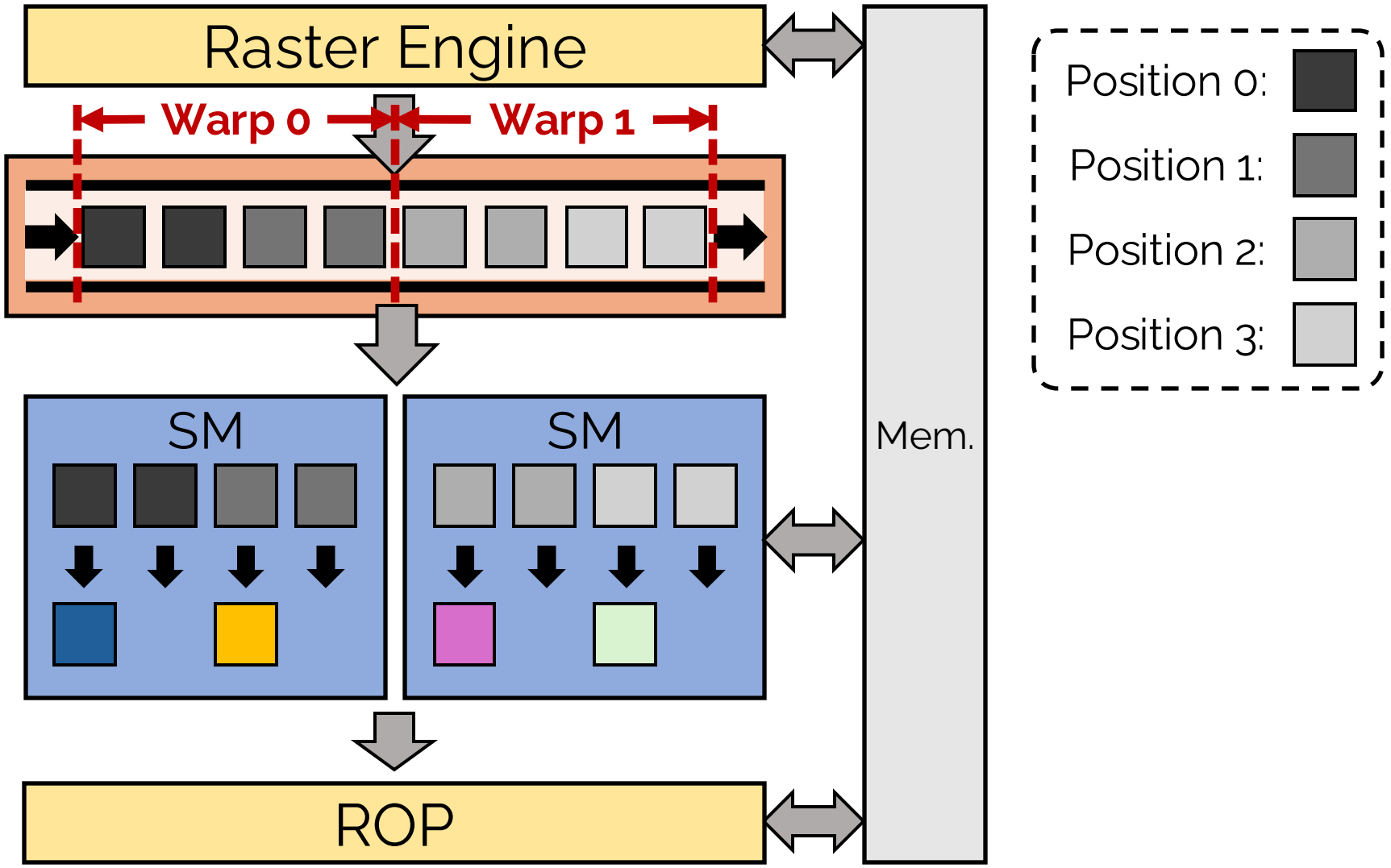


Quad Merging

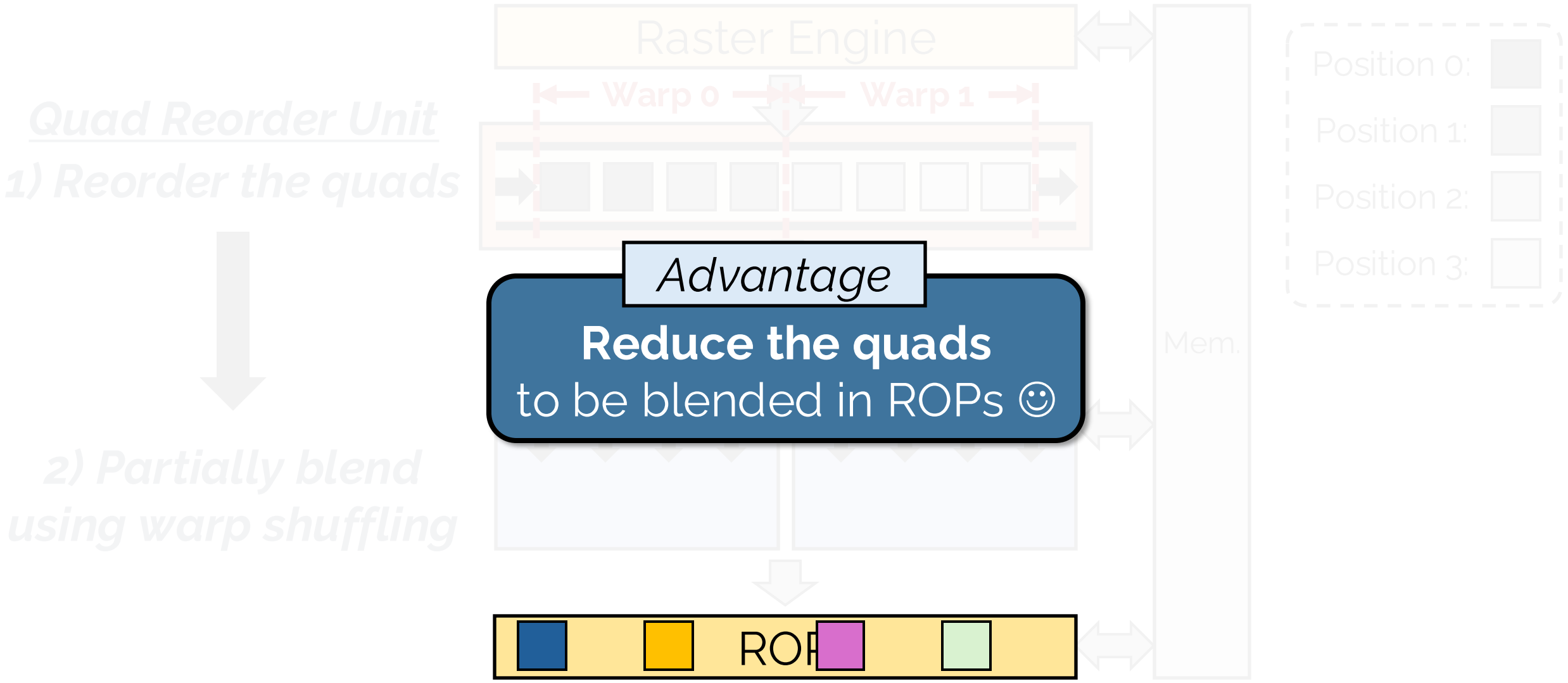
Quad Reorder Unit

1) Reorder the quads

2) Partially blend using warp shuffling



Quad Merging



Outline

- **Background**
  - 3D Gaussian Splatting (3DGS)
  - Hardware Graphics Pipeline
- **Limitations of Graphics Hardware**
- **VR-Pipe: Graphics Hardware Extension for Volume Rendering**
  - Quad Merging with Multi-Granular Tile Binning
  - Hardware Support for Early Termination
- **Evaluation**
- **Conclusion**

Experimental Setup

Performance Evaluation

- Emerald (ISCA' 19)
  - Cycle-level simulator w/ graphics hardware modeling based on GPGPU-sim and gem5
  - With extensive modifications based on our analysis

Workloads

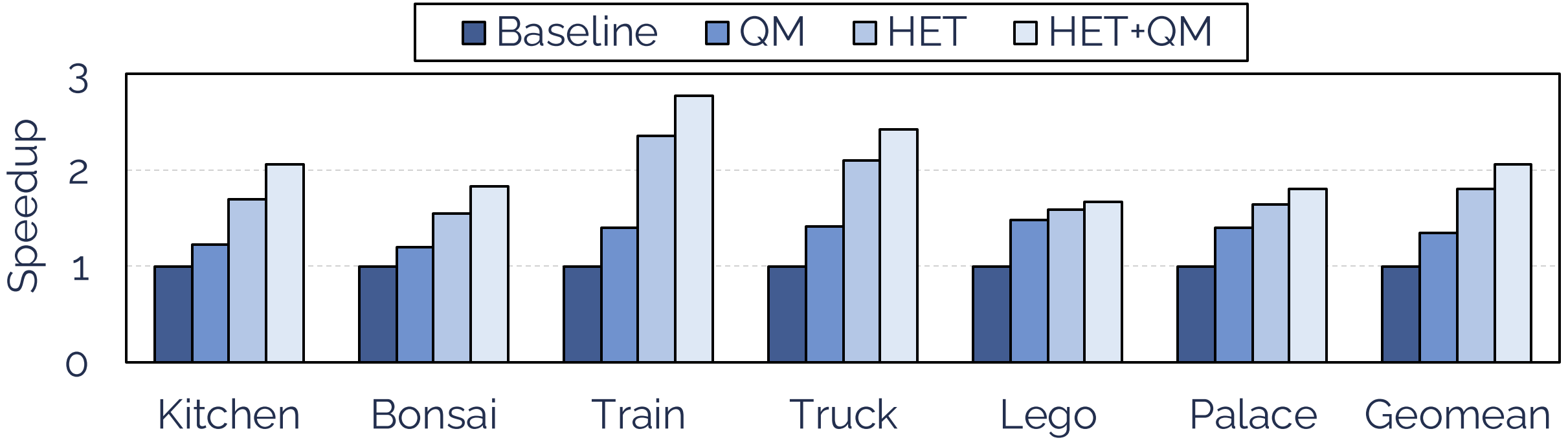
- Mip-NeRF 360: Kitchen, Bonsai
- Tanks & Temples: Train, Truck
- Synthetic-NeRF: Lego
- Synthetic-NSVF: Palace

Baseline GPU Configuration

# GPC	1
# SMs	16 (1024 CUDA Cores)
Core Frequency	612 MHz
L1D/T	48KB, 128B line
Shared L2	4MB, 128B line (sectored)
ROP Cache	16KB, 128B line (sectored)
ROP Throughput	2 quads/cycle (RGBA16F)
DRAM	LPDDR3-1600 (16-channel)

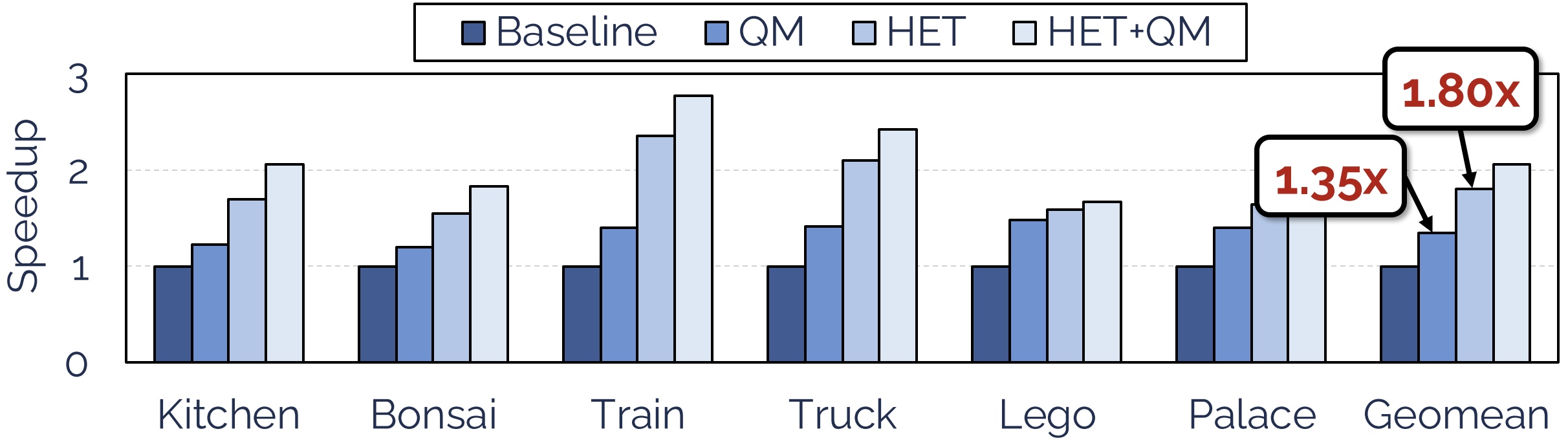
Performance

QM: Quad Merging
HET: Hardware-based Early Termination



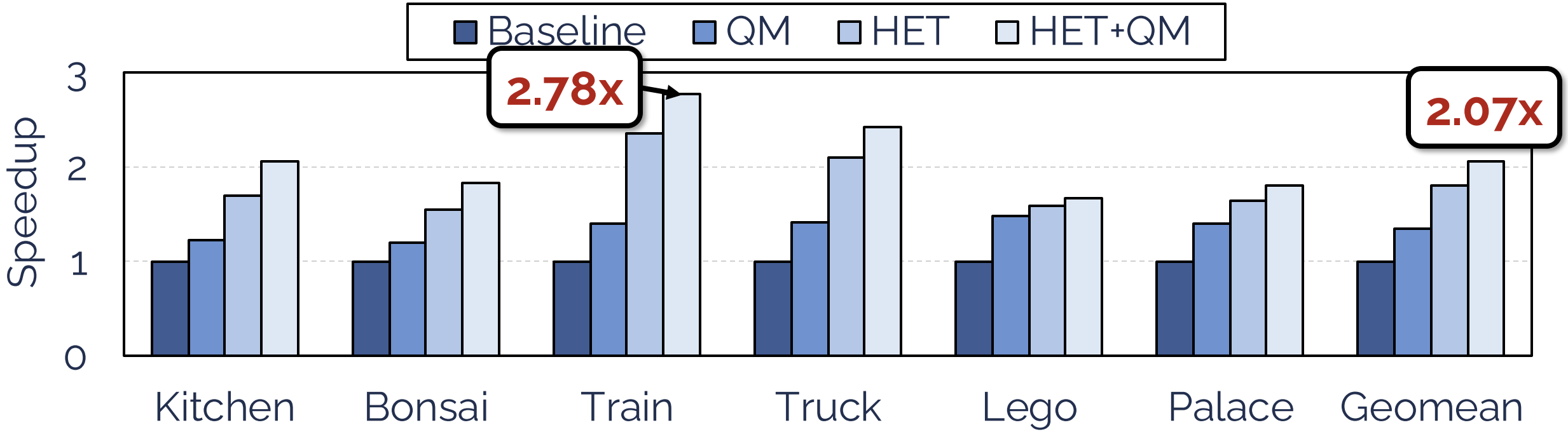
Performance

QM: Quad Merging
HET: Hardware-based Early Termination



Performance

QM: Quad Merging
HET: Hardware-based Early Termination

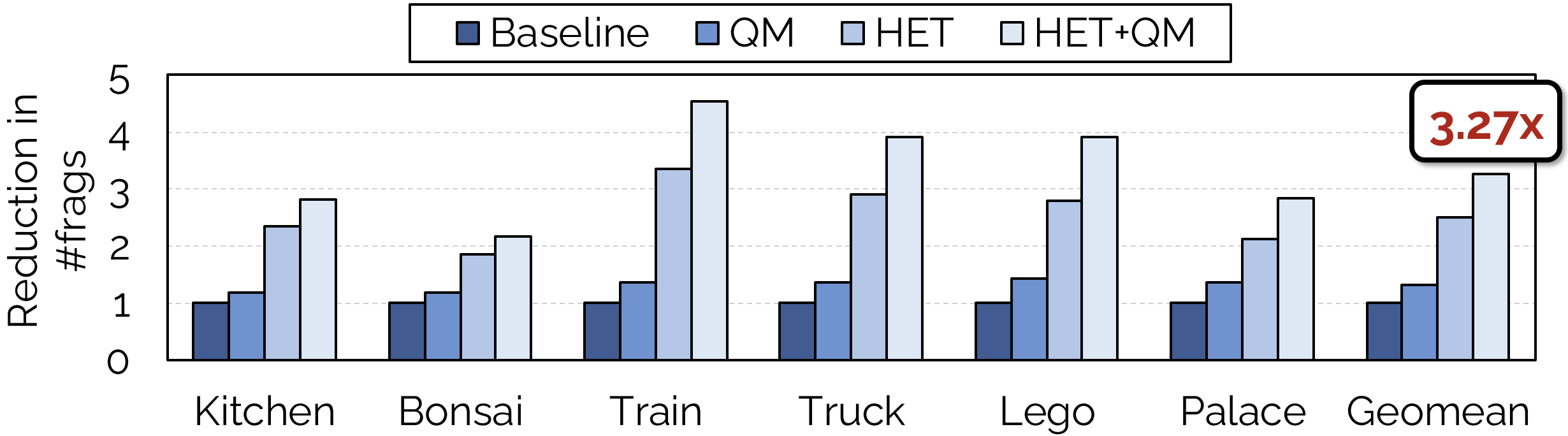


VR-Pipe greatly improves rendering performance

with minimal hardware overhead in a GPU

Source of Performance Gain

Reduction in the Number of Fragments

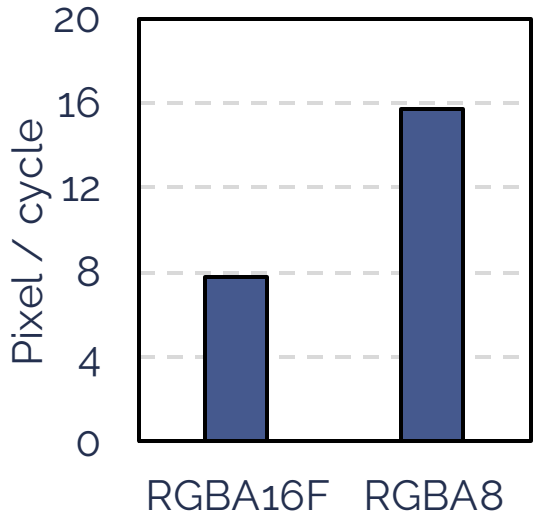


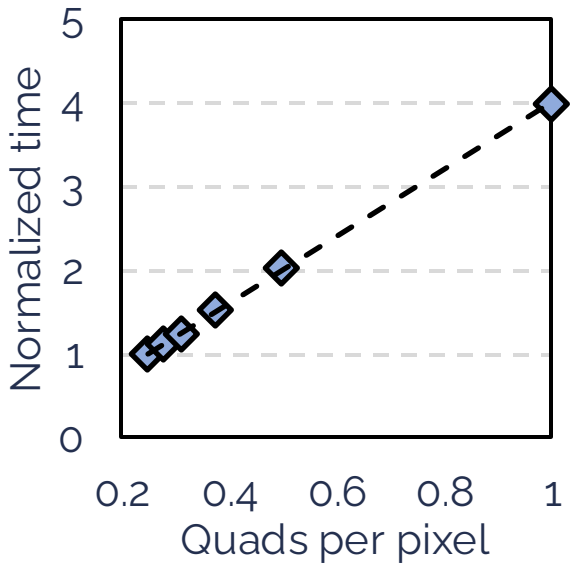
VR-Pipe significantly reduces the number of fragments

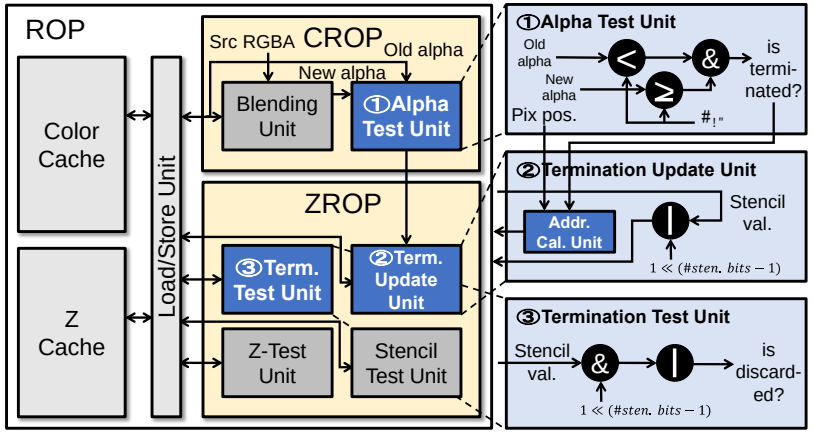
blended by ROP

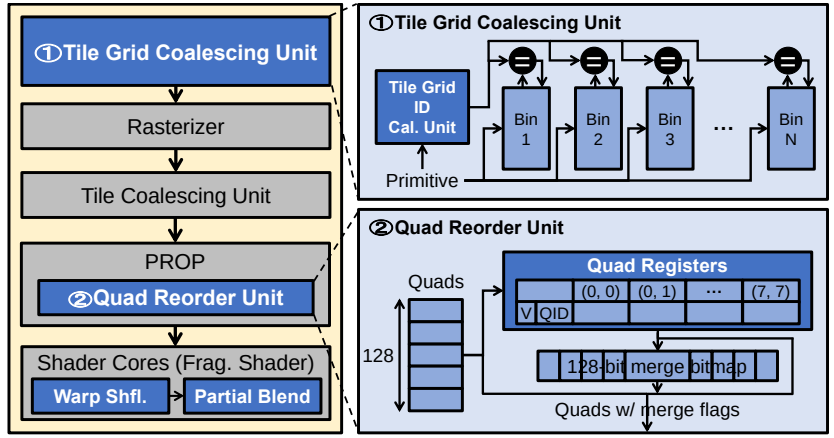
More Details in Our Paper

- Analysis on Real Graphics Hardware
- Limitations of SW-based Optimizations
- Hardware Implementation Cost
- Details of Proposed Microarchitecture
- Others…









Conclusion

Problem

- **High ROP pressure** for blending a number of fragments per pixel
- Lack of native hardware support for early termination

Solution: **VR-Pipe**, a GPU hardware extension for volume rendering

- Hardware-based early termination to early-discard the fragments
- Quad merging with multi-granular tile binning to exploit underutilized SMs

Result

- VR-Pipe achieves up to a **2.78x speedup** over the conventional graphics pipeline with mininal hardware overhead! ☺

Thank You!

VR-Pipe

Streamlining Hardware Graphics Pipeline for Volume Rendering

Junseo Lee (junseo.lee@snu.ac.kr)

