EuroGraphics Symposium on Parallel Graphics and Visualization 2020

Fast Multi-View Rendering for Real-Time Applications

Johannes Unterguggenberger, Bernhard Kerbl, Markus Steinberger,
Dieter Schmalstieg, and Michael Wimmer

TU Wien, Institute of Visual Computing &
Human-Centered Technology, Austria

Rendering Multiple Views

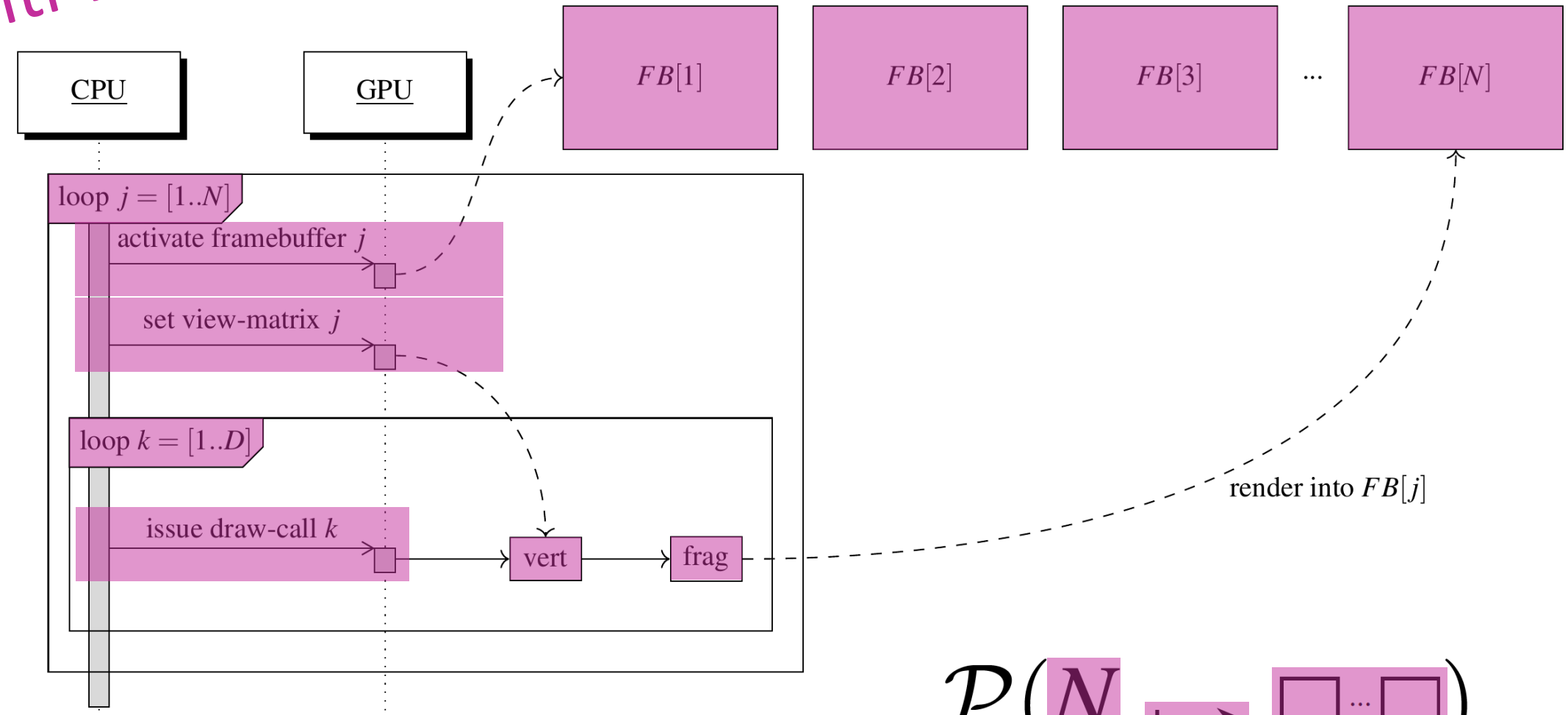
1/5

Different Graphics Pipeline Configurations

How about using brute-force?

How About Just Using Multiple Passes?

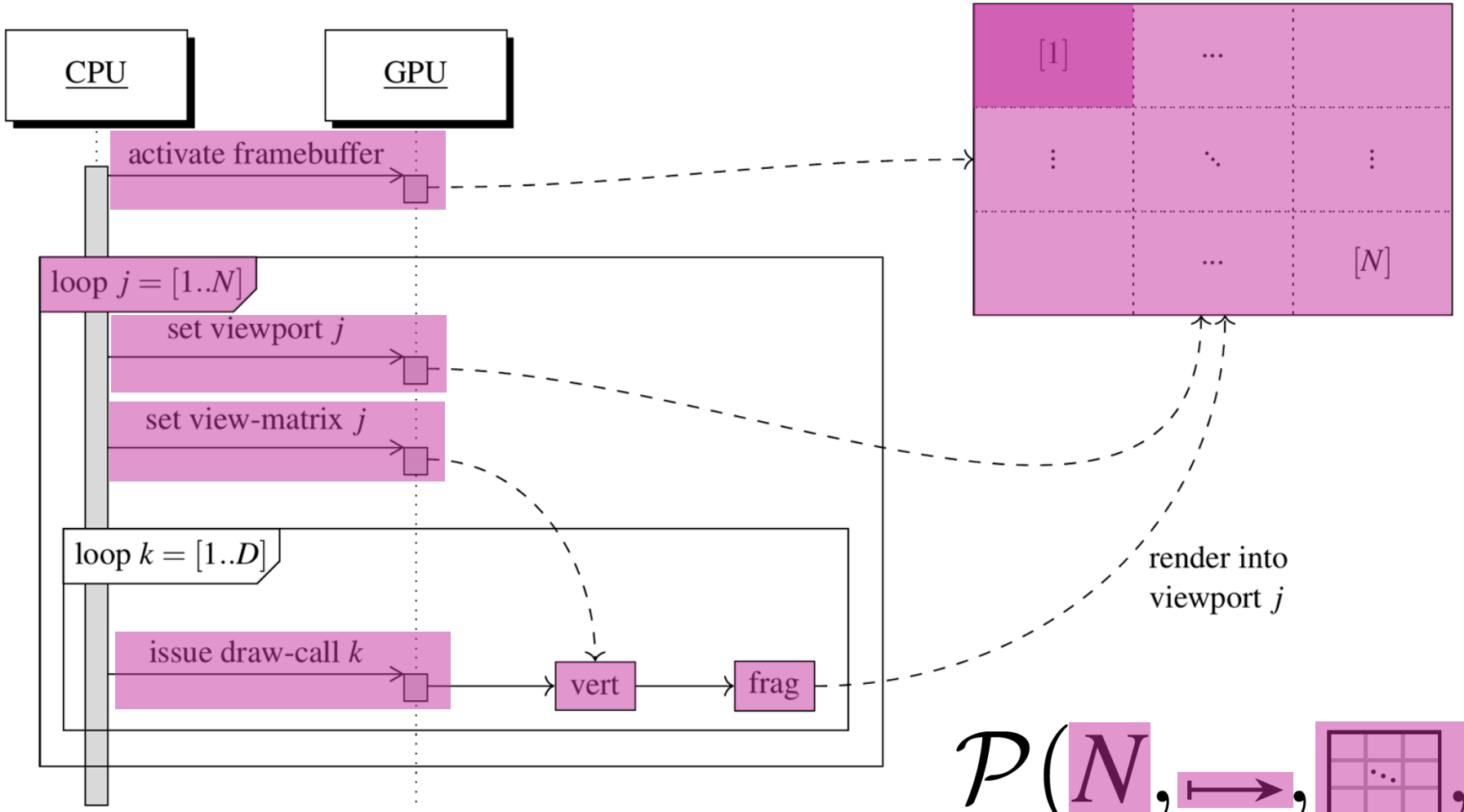
"Multi-Pass"



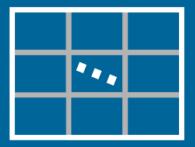
$$\mathcal{P}\left(N, \longmapsto, \square \cdots \square\right)$$

Changed Framebuffer Layout

“Multi-Pass with a large, partitioned framebuffer”



$$\mathcal{P}\left(N, \longmapsto, \boxed{\cdots}, \mathrm{CLIP}_{\mathrm{VP}}\right)$$

Performance difference between  and 

Relative frame times (in milliseconds):

$$\mathcal{P}\left(N, \longmapsto, \boxed{\ddots}, \texttt{CLIP}_{\texttt{VP}}\right) - \mathcal{P}\left(N, \longmapsto, \square \cdots \square\right)$$

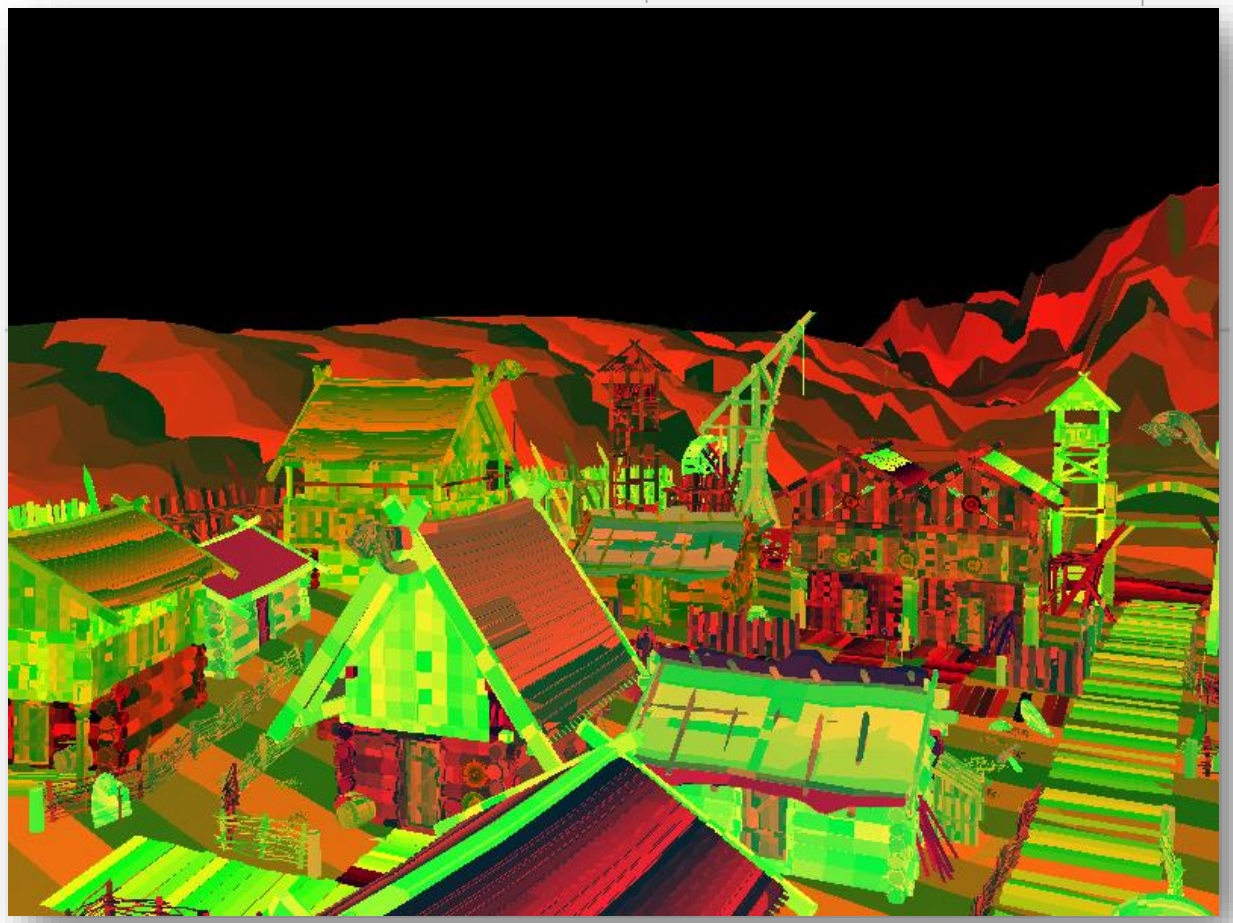
Performance Difference Between ▭ ··· ▭ and ▦

$\mathcal{P}(N, \rightarrow, \blacksquare, \text{CLIP}_{\text{VP}}) - \mathcal{P}(N, \rightarrow, \square \cdots \square)$

▭ ··· ▭ faster by x ms

▦ faster by -x ms

1.0 ms
0.5 ms
0.0 ms
-0.5 ms
-1.0 ms
-1.5 ms



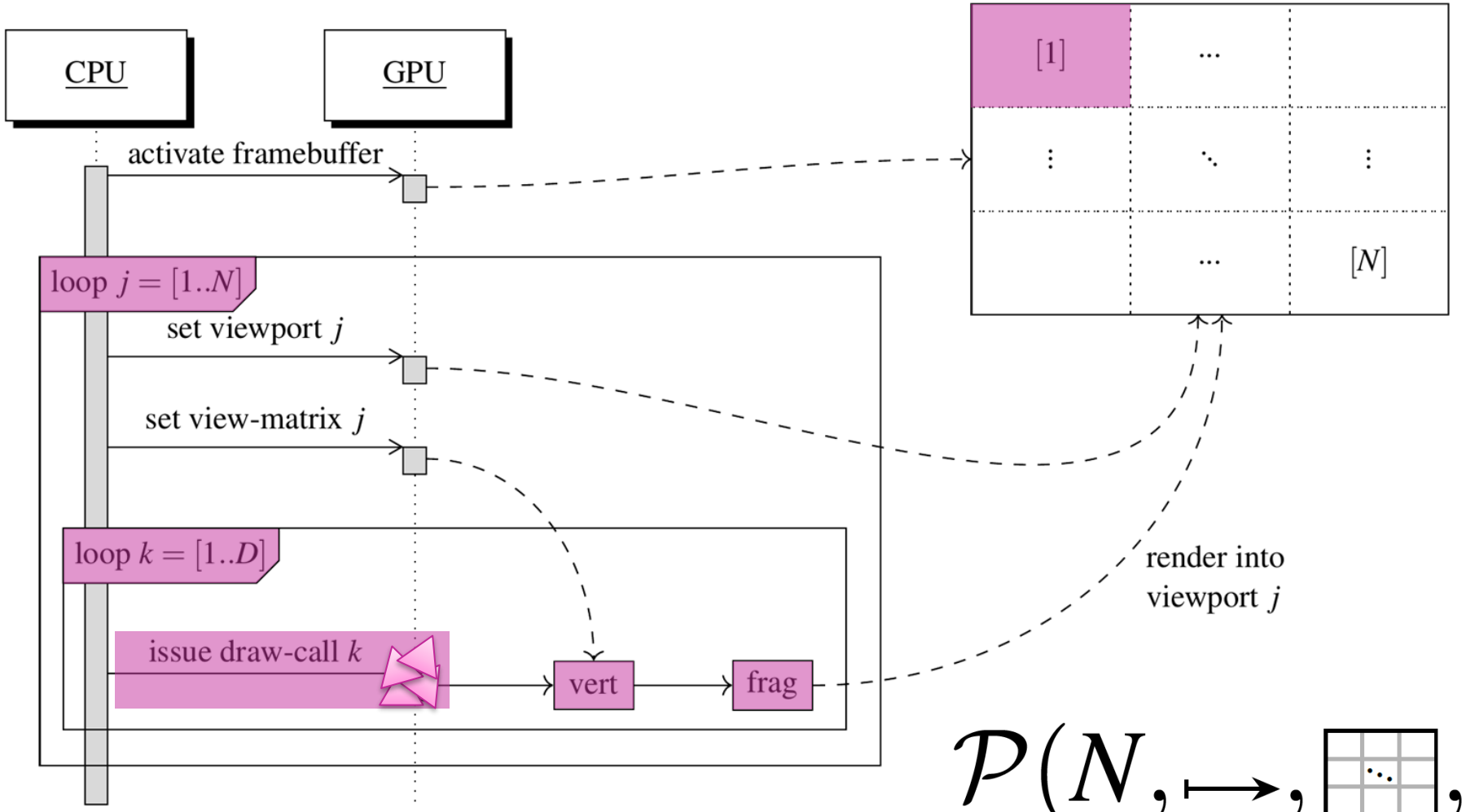
AMD R9 380
AMD RX 580
NVIDIA GTX 780
NVIDIA GTX 980
NVIDIA GTX 1060
NVIDIA GTX 1650 SUPER
NVIDIA RTX 2080
NVIDIA RTX 2080 Ti

2/5

The Journey of the Triangles

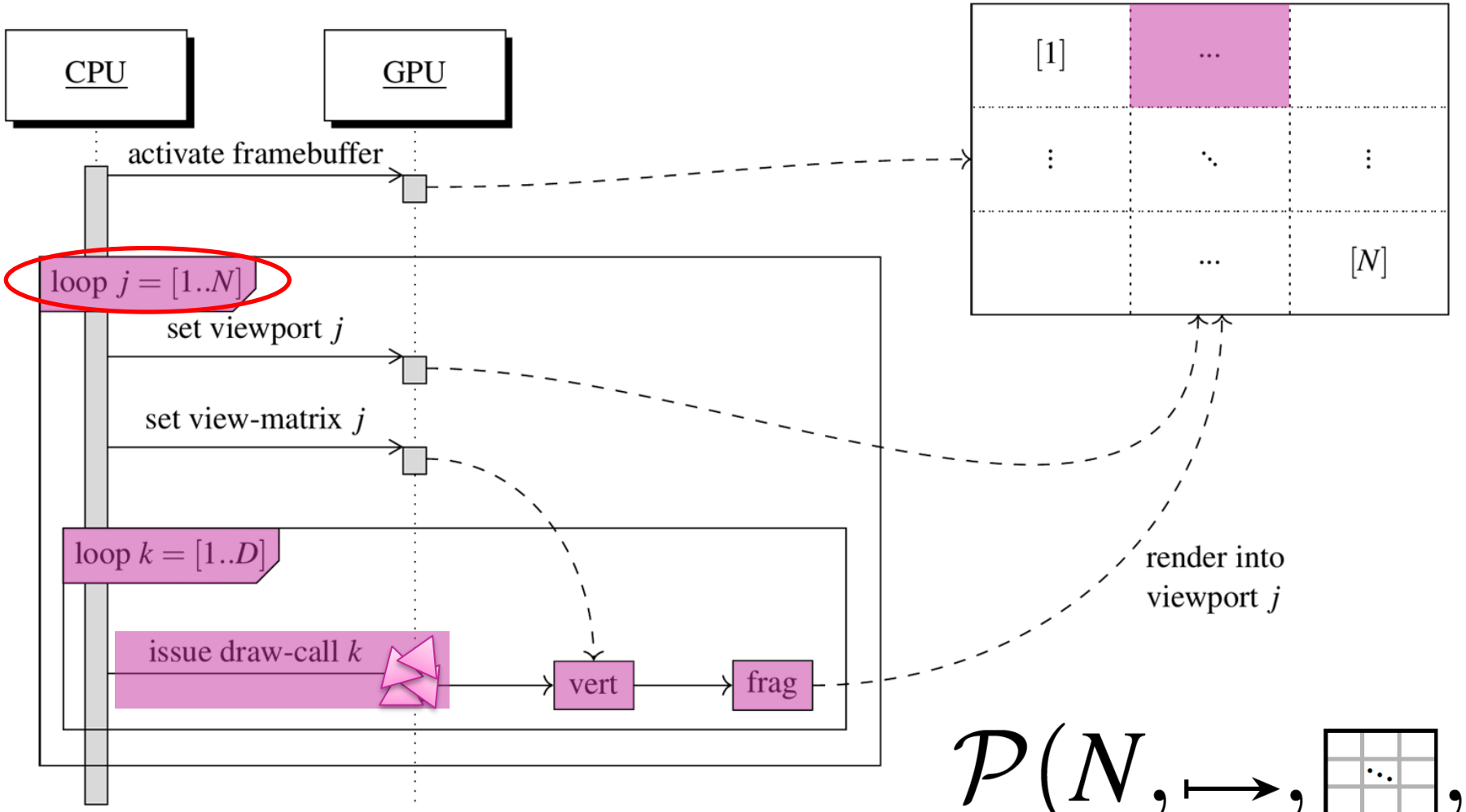
“Geometry Amplification”

The Journey of the Triangles



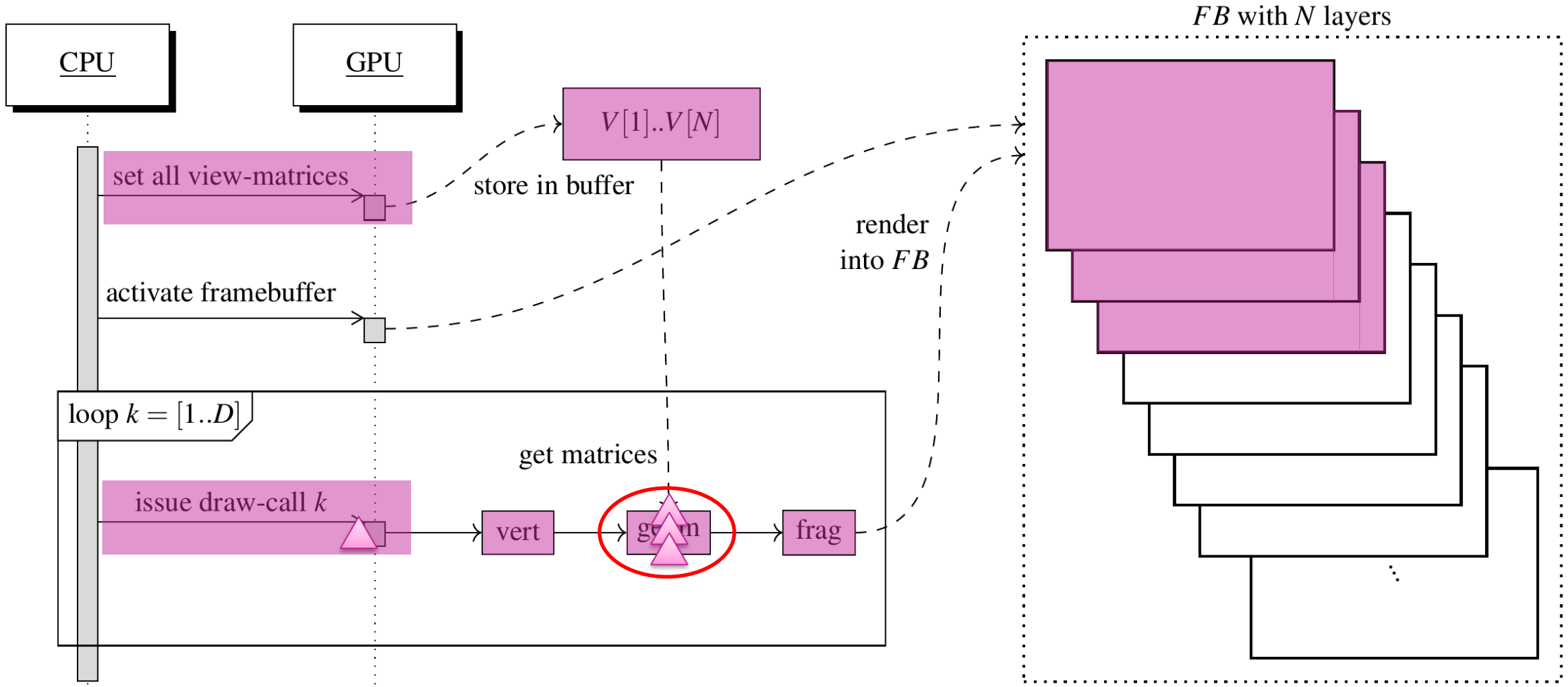
$$\mathcal{P}\left(N, \longmapsto, \boxed{\ddots}, \texttt{CLIP}_{\texttt{VP}}\right)$$

The Journey of the Triangles

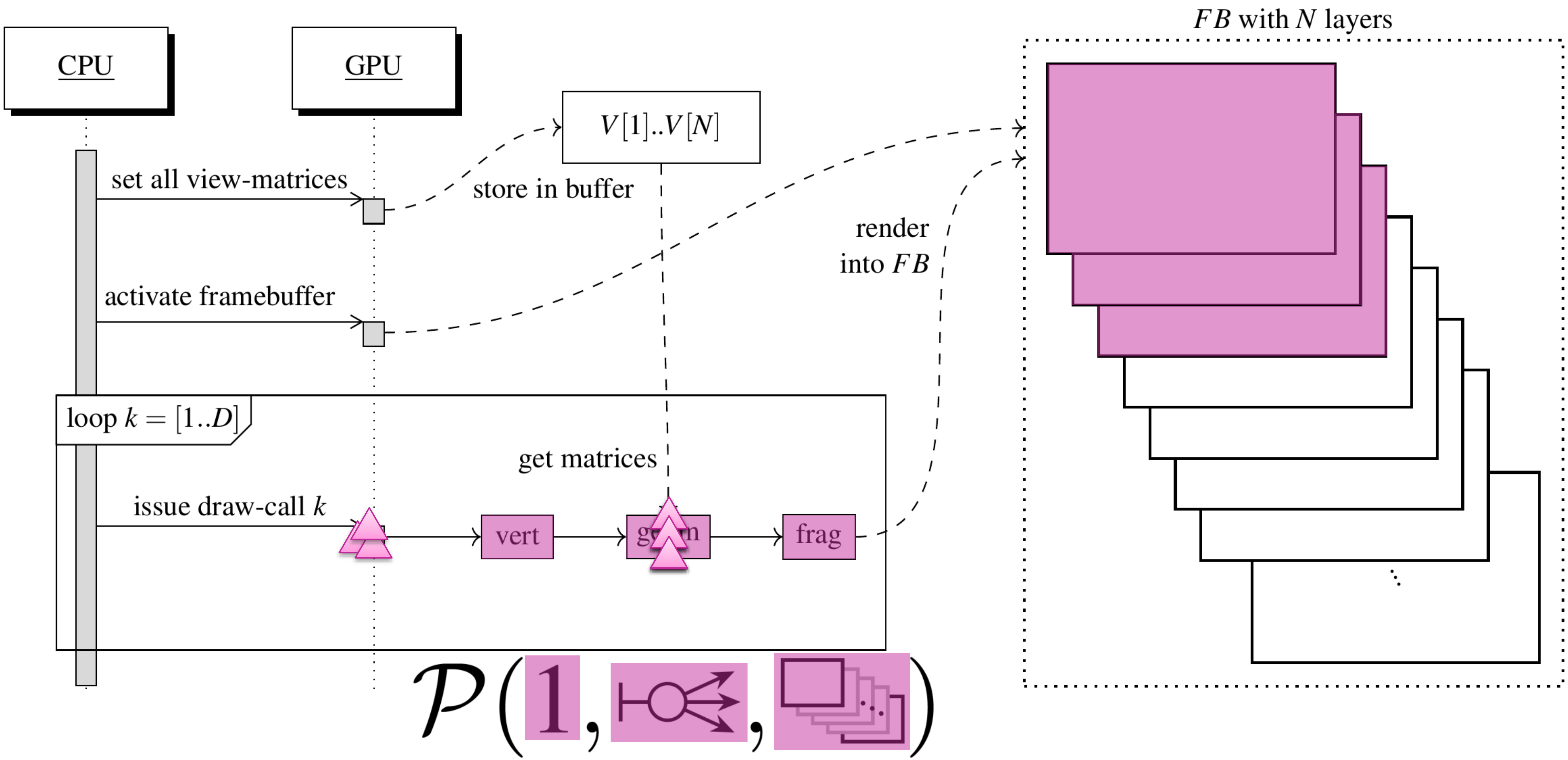


$$\mathcal{P}\left(N, \longmapsto, \boxed{\cdots}, \texttt{CLIP}_{\texttt{VP}}\right)$$

Moving the Loop into the Geometry Shader

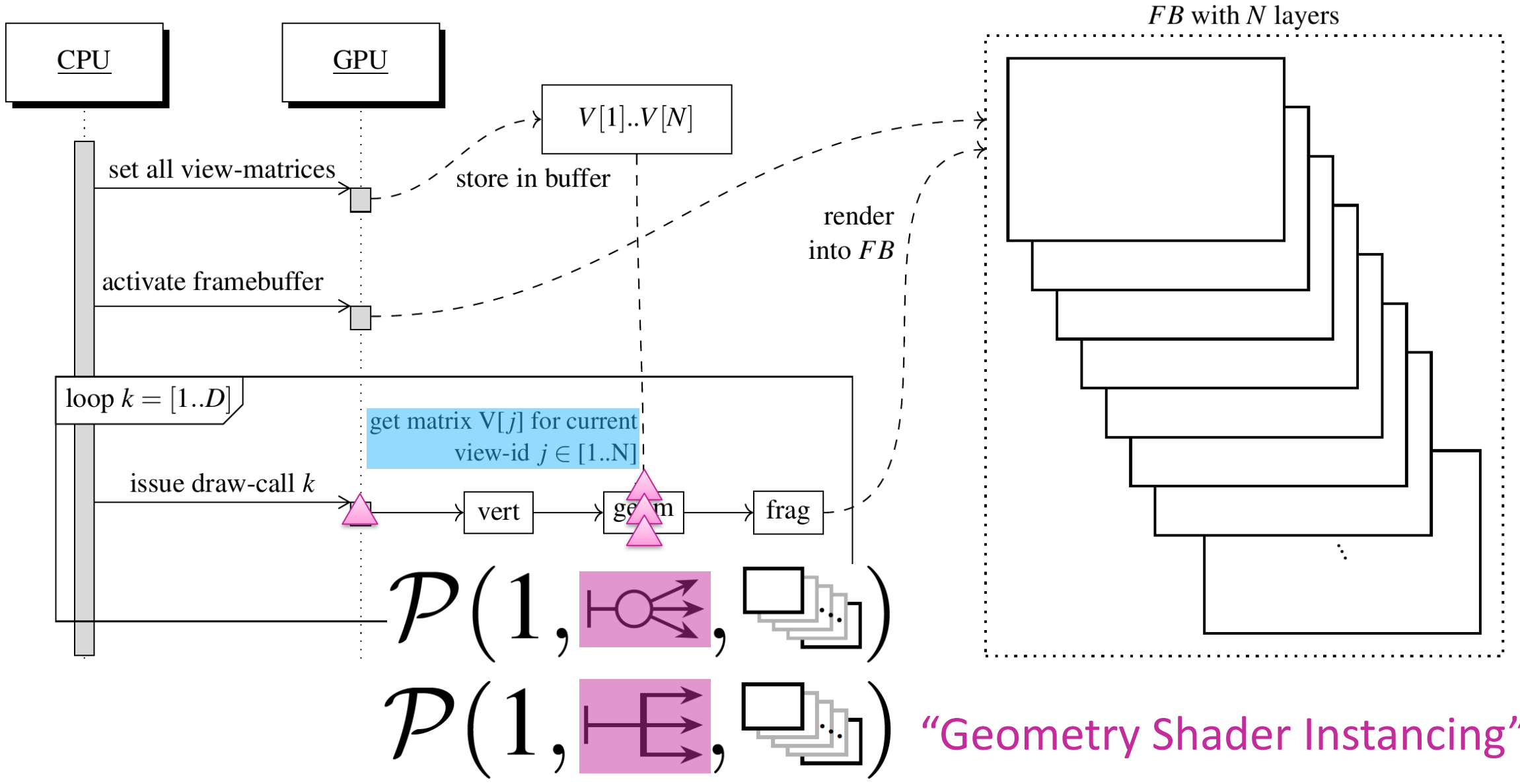


Moving the Loop into the Geometry Shader



"Geometry Shader Amplification"

Replacing the Loop with Instancing



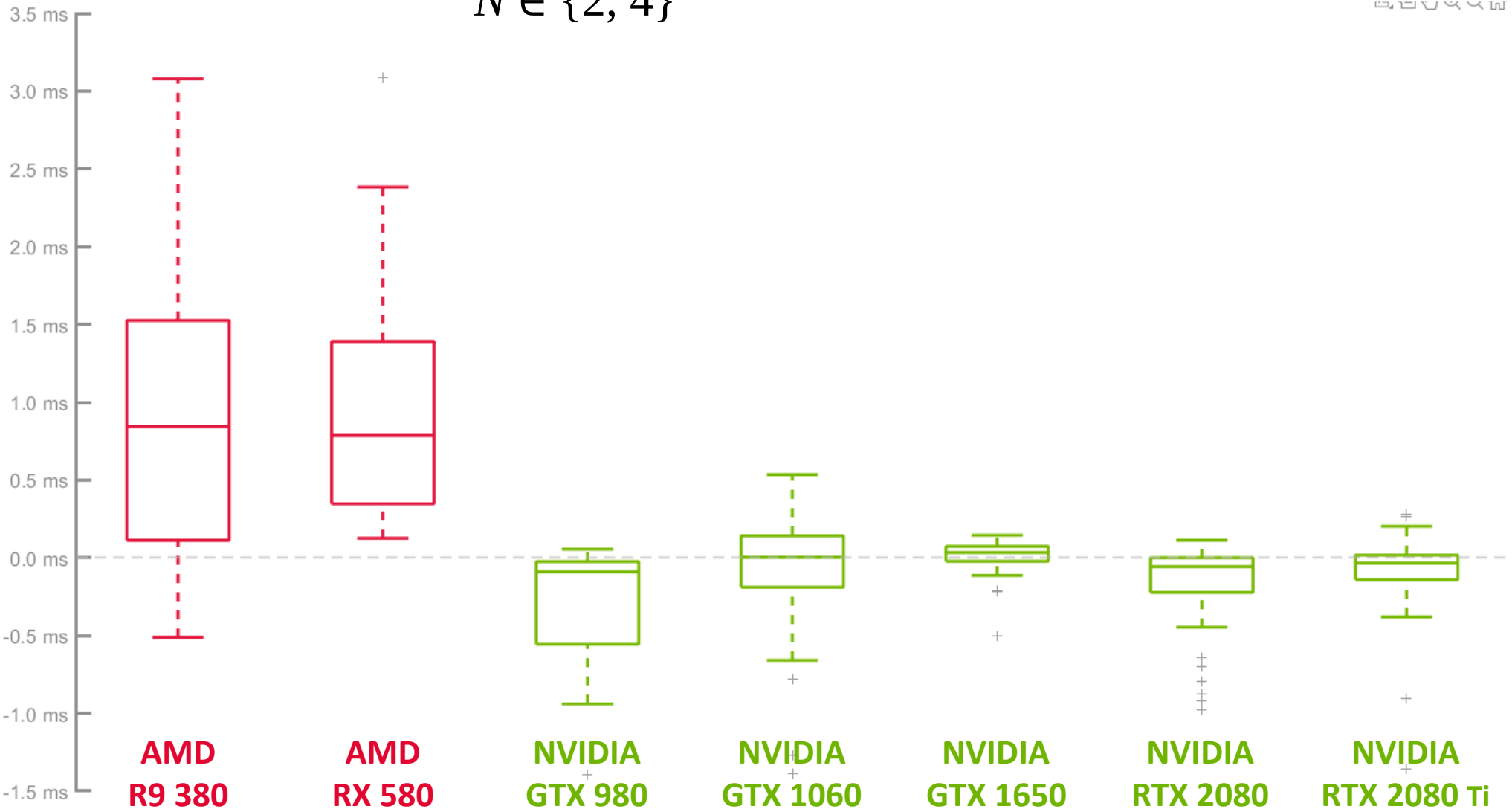
"Geometry Shader Instancing"

Performance: $\mathcal{P}(1, \mapsto\!\!\!\!\Rightarrow, \square\!\square) - \mathcal{P}(N, \longmapsto, \square\cdots\square)$

$N \in \{2, 4\}$

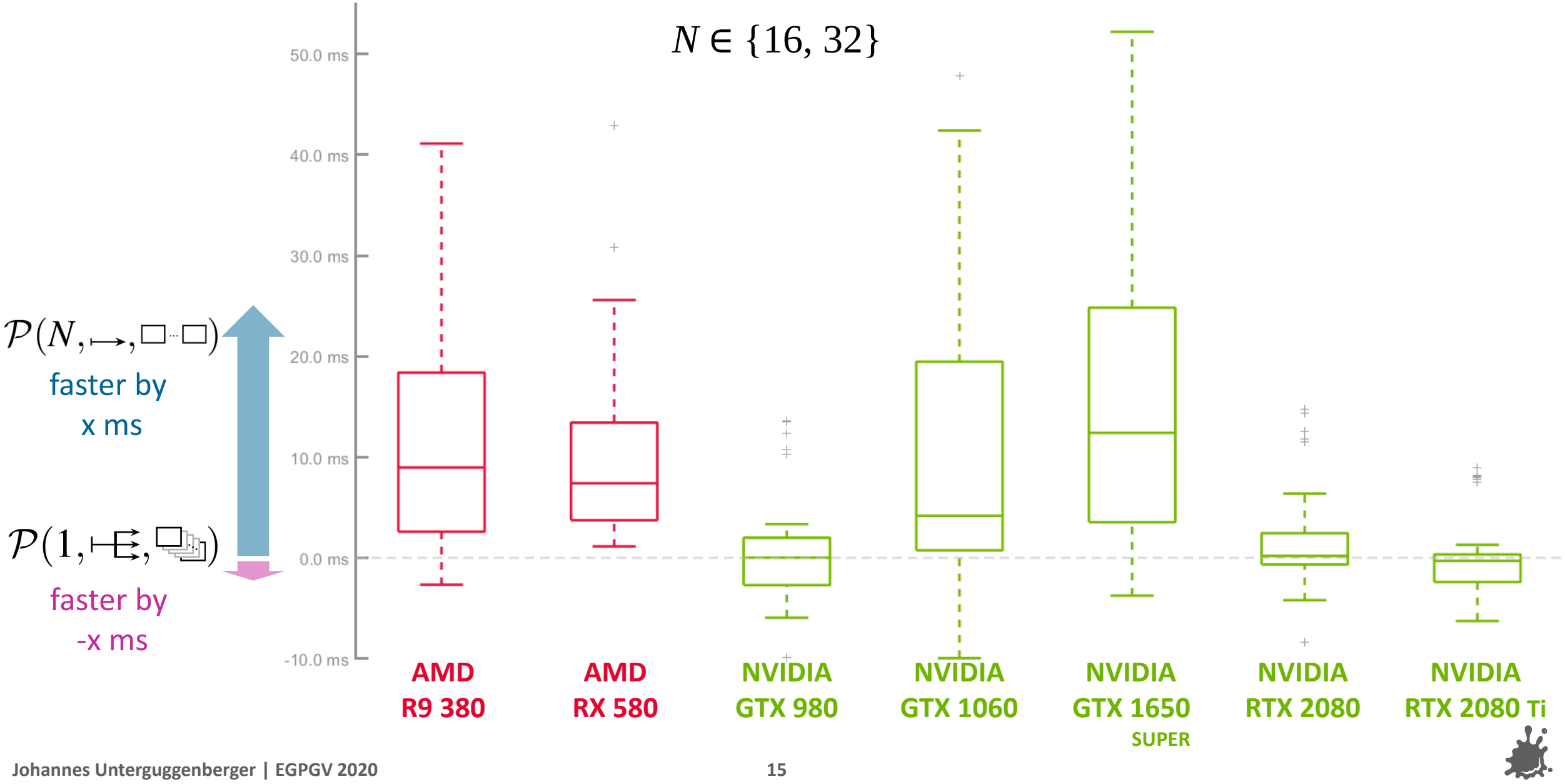
$\mathcal{P}(N, \longmapsto, \square\cdots\square)$
faster by
x ms

$\mathcal{P}(1, \mapsto\!\!\!\!\Rightarrow, \square\!\square)$
faster by
-x ms



Performance: $\mathcal{P}(1, \mapsto, \square\square) - \mathcal{P}(N, \rightarrow, \square\cdots\square)$

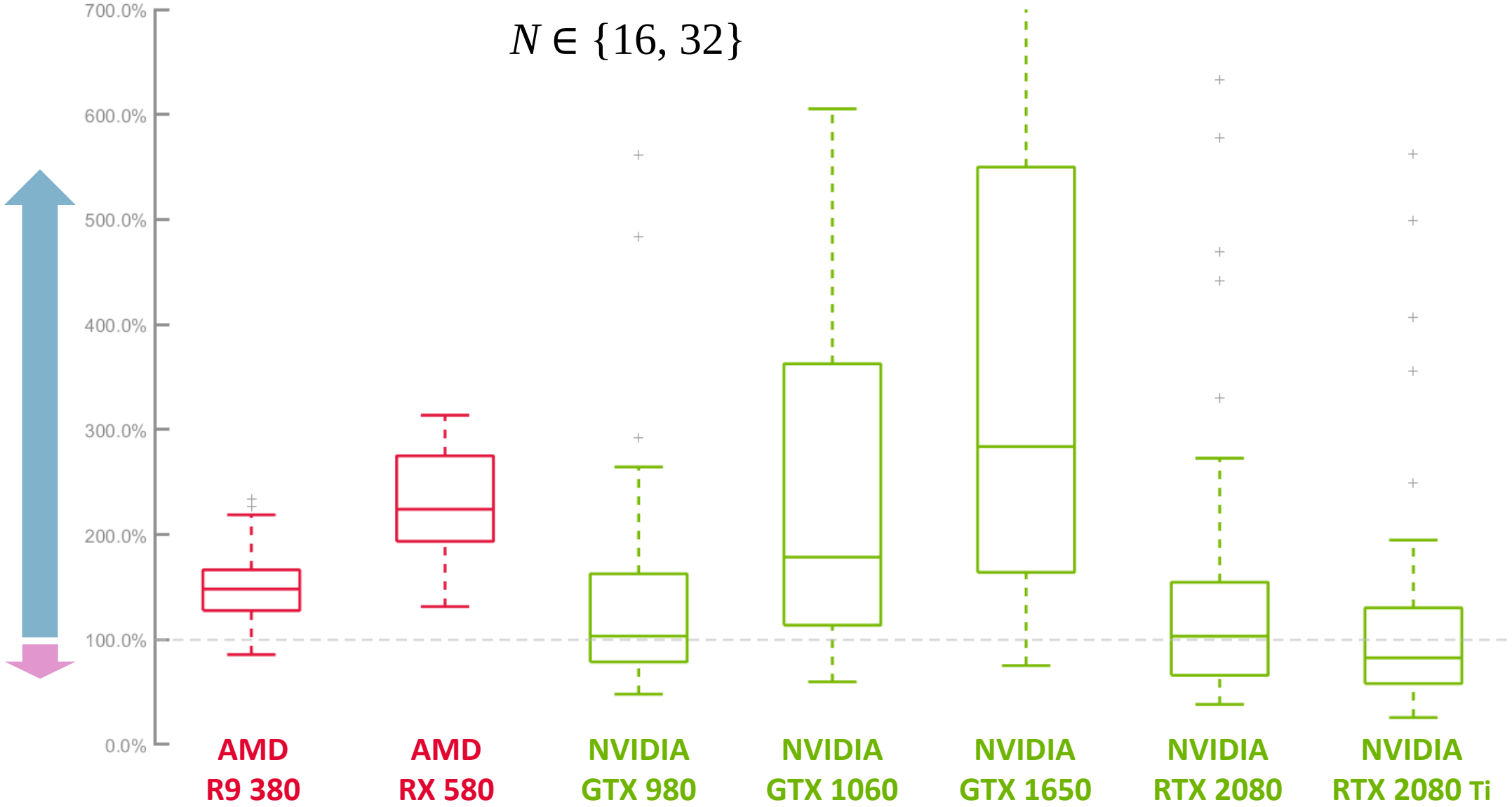
$N \in \{16, 32\}$



Performance: $\mathcal{P}(1, \mapsto, \square\square) \;/\; \mathcal{P}(N, \longmapsto, \square \cdots \square)$

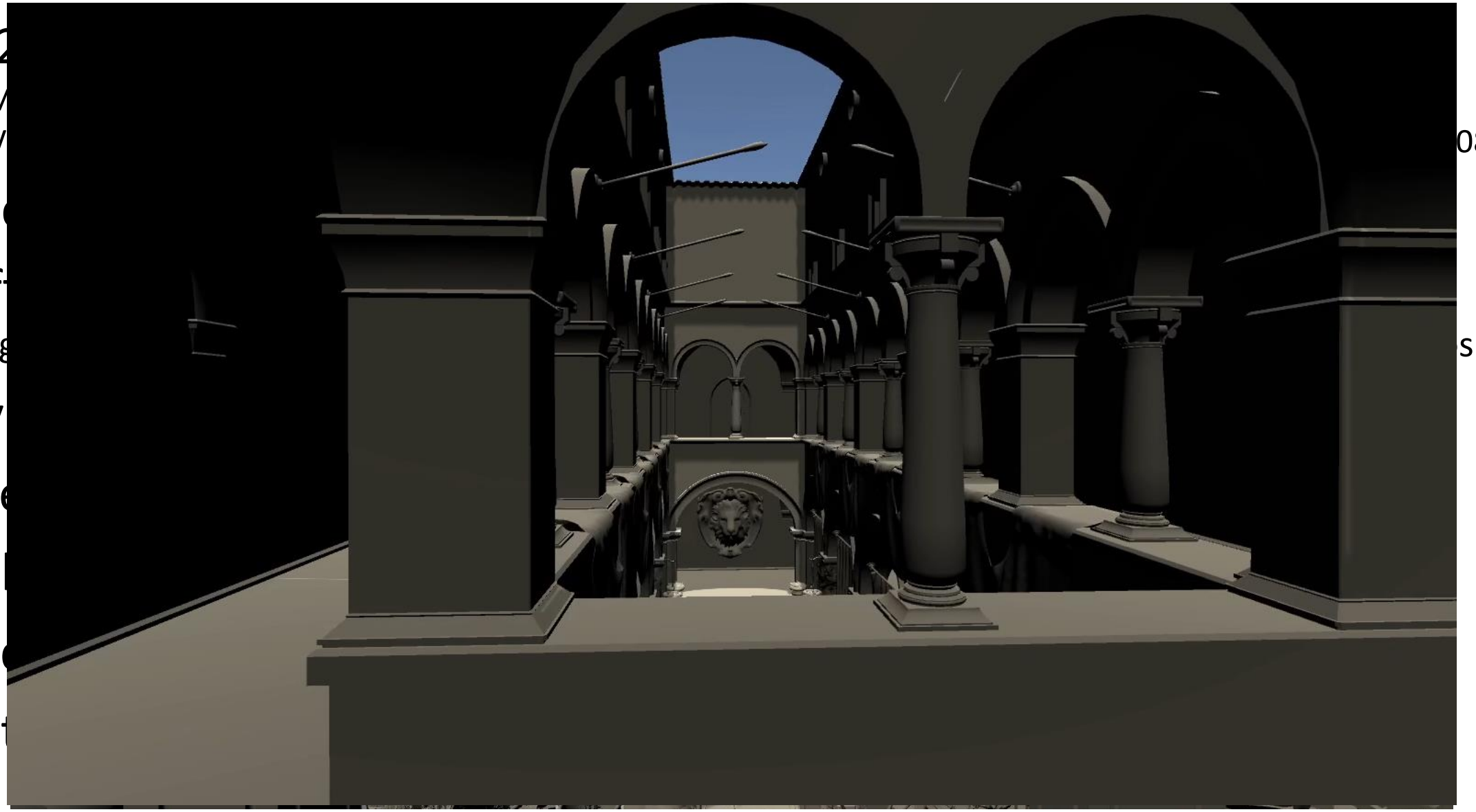
$N \in \{16, 32\}$

Performance of $\mathcal{P}(1, \mapsto, \square\square)$ w.r.t. $\mathcal{P}(N, \longmapsto, \square \cdots \square)$



Different Configurations

- 6+2
  2 AM
  6 NV
- 6 s
- Dif
  Rang
- 4 v
- 3 r
- Lig
- Ad
- Wi



080 Ti)

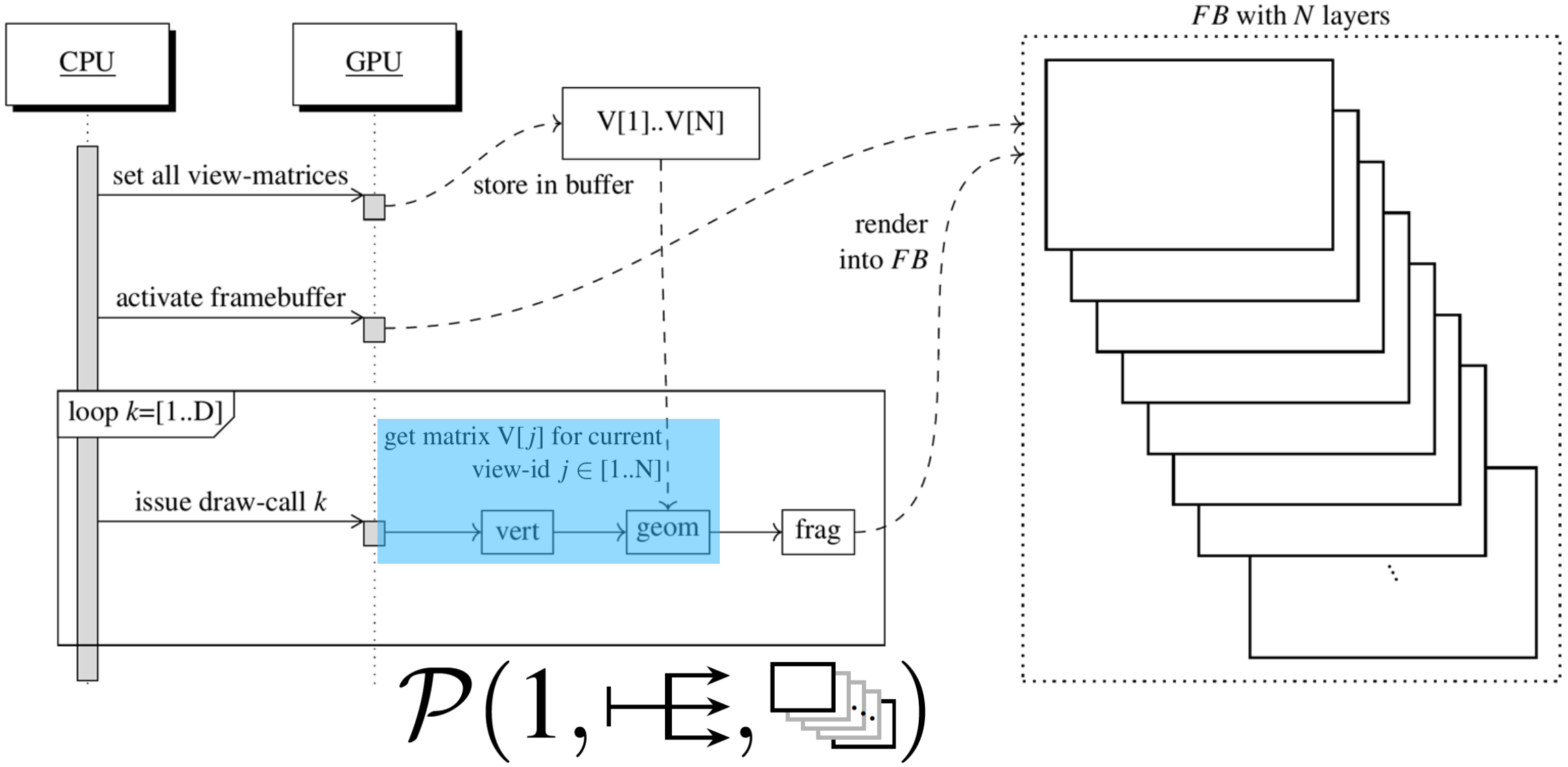
sition

3/5

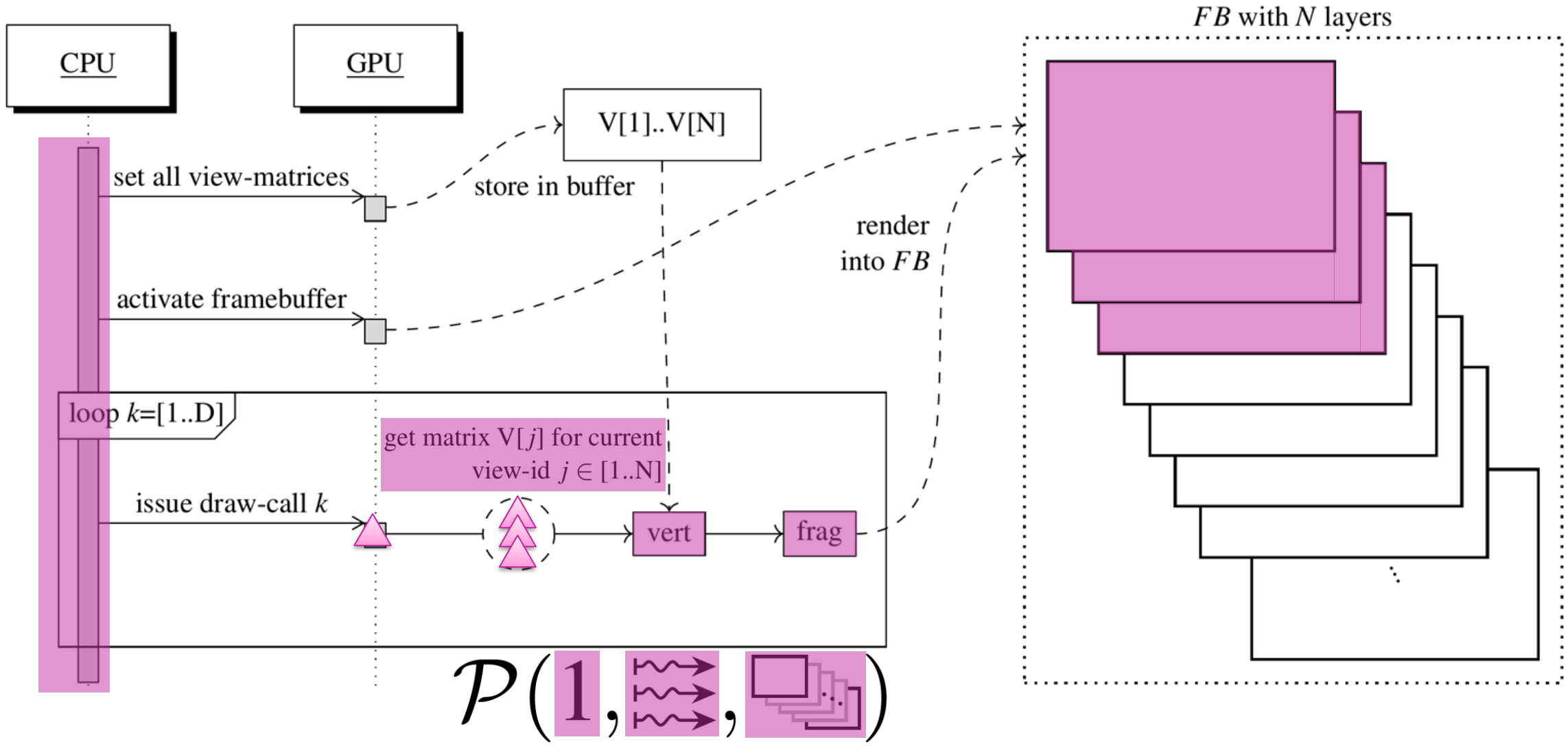
Hardware-Accelerated Multi-View Rendering

OVR_multiview

Recap: Geometry Shader Instancing

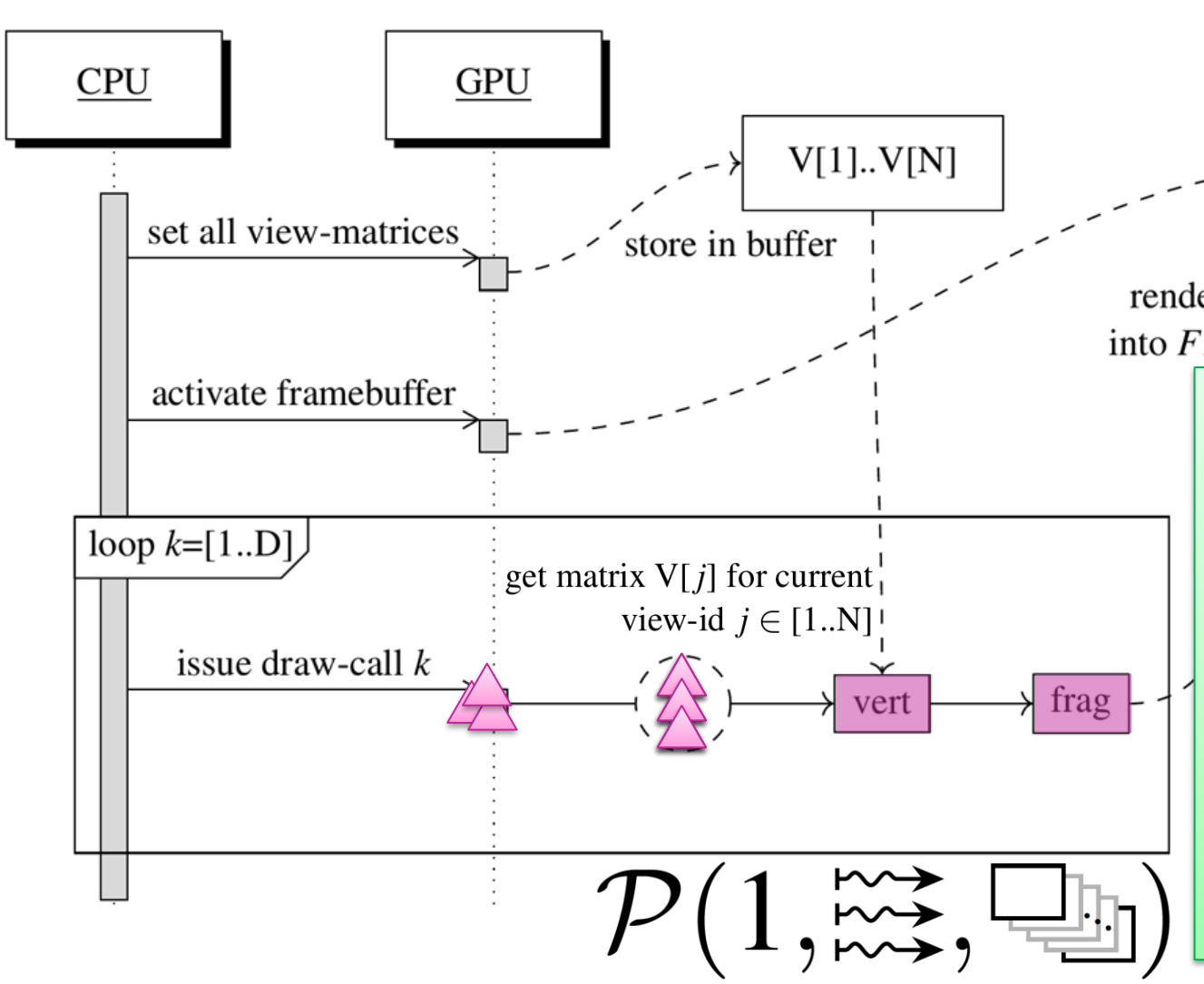


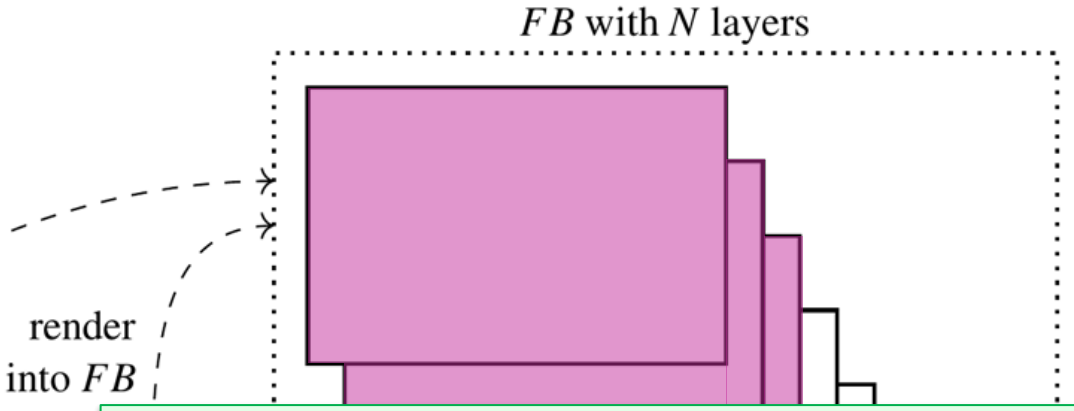
Pipeline for Hardware-Accelerated Multi-View Rendering



"Hardware-Accelerated Multi-View" or "OVR Multi-View" (Oculus VR)

Pipeline for Hardware-Accelerated Multi-View Rendering





To improve efficiency, the NVIDIA compiler analyzes the input shader and produce a compiled output that executes view independent code once, with the result shared across all output views, while view dependent attributes are necessarily computed once per output view.

NVIDIA, TURING GPU ARCHITECTURE

"Hardware-Accelerated Multi-View" or "OVR Multi-View" (Oculus VR)

Performance: $\mathcal{P}(1, \rightrightarrows, \square\square)$ / $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

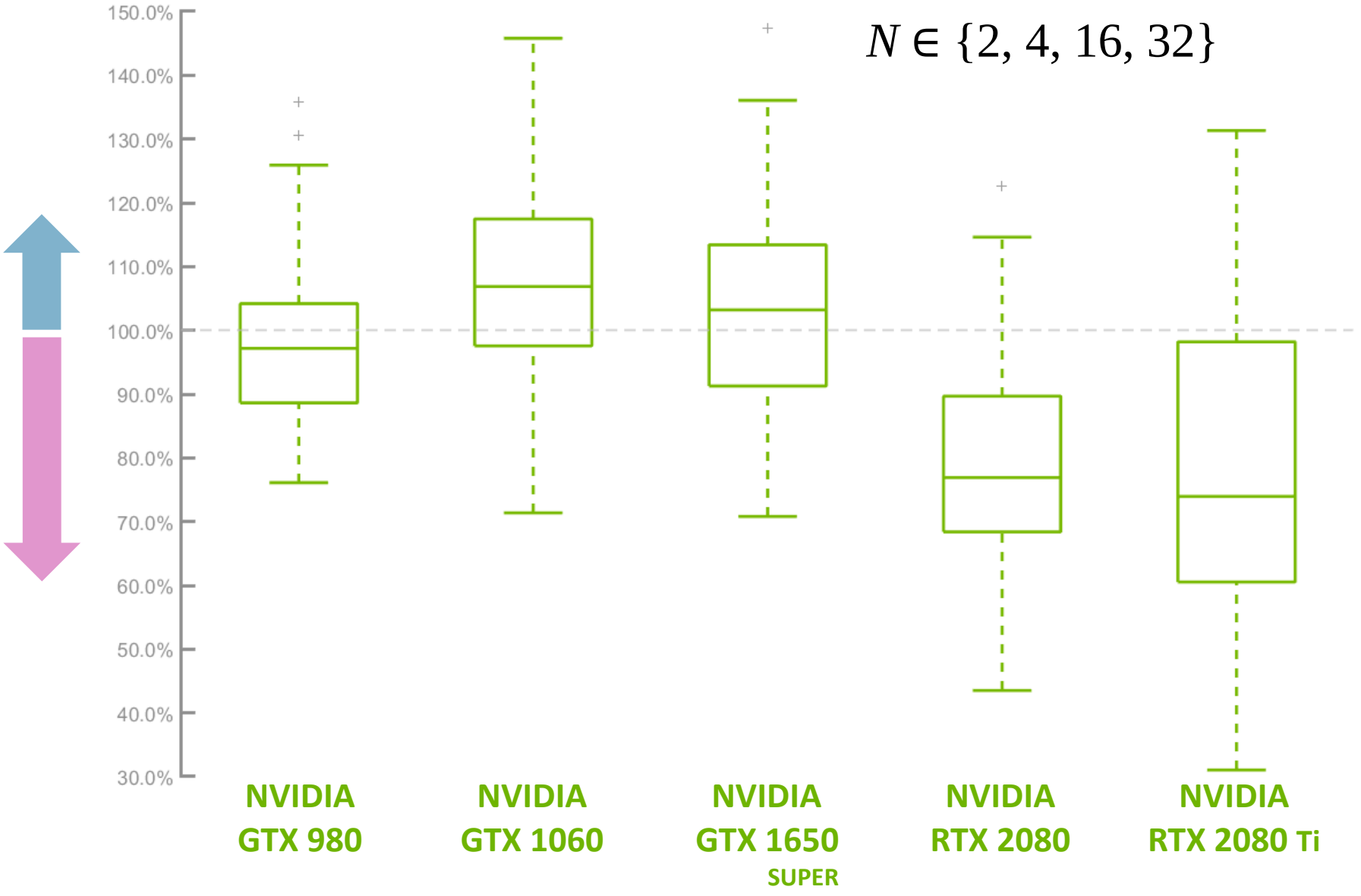
Performance of

$\mathcal{P}(1, \rightrightarrows, \square\square)$

w.r.t.

$\mathcal{P}(N, \longmapsto, \square \cdots \square)$

$N \in \{2, 4, 16, 32\}$



Performance: $\mathcal{P}(1, \rightsquigarrow, \square)$ / $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

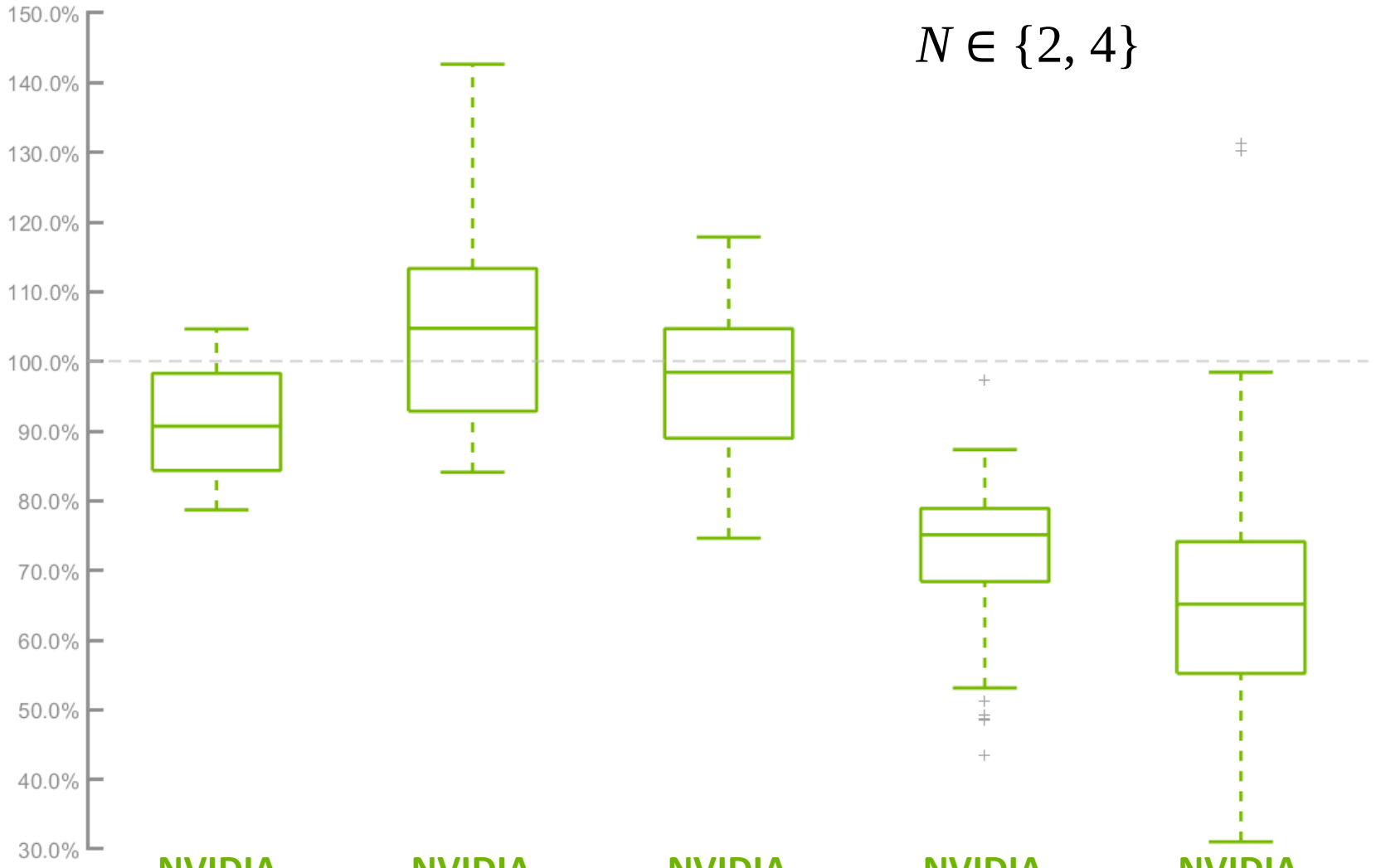
Performance of

$\mathcal{P}(1, \rightsquigarrow, \square)$

w.r.t.

$\mathcal{P}(N, \longmapsto, \square \cdots \square)$

$N \in \{2, 4\}$



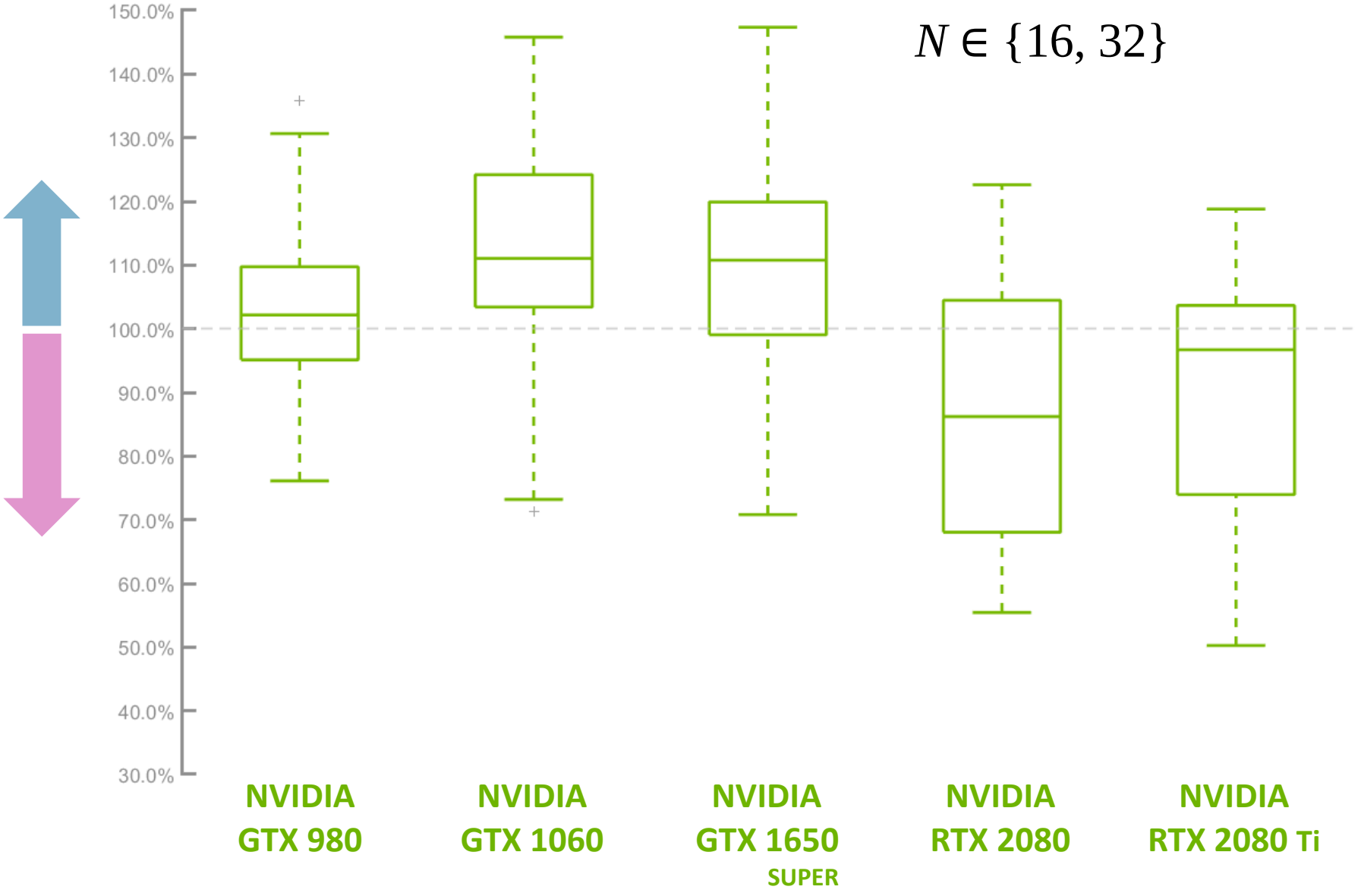
Performance: $\mathcal{P}(1, \rightsquigarrow, \square\square)$ / $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

Performance of

$\mathcal{P}(1, \rightsquigarrow, \square\square)$

w.r.t.

$\mathcal{P}(N, \longmapsto, \square \cdots \square)$



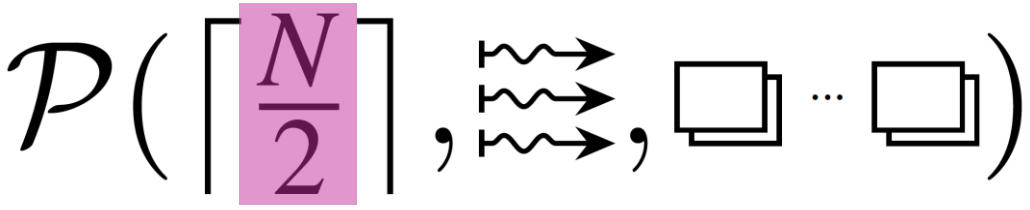
How many views can actually be hardware-accelerated with ONE draw call?

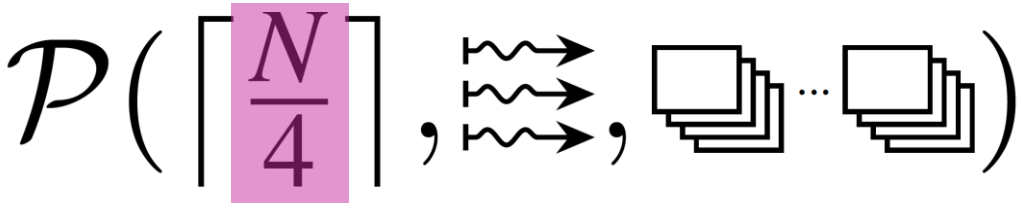
With Pascal's SMP engine, which **supports two separate projection centers**, the GPU can render the two stereo projections directly in a single rendering pass.

NVIDIA, GeForce GTX 1080 Whitepaper

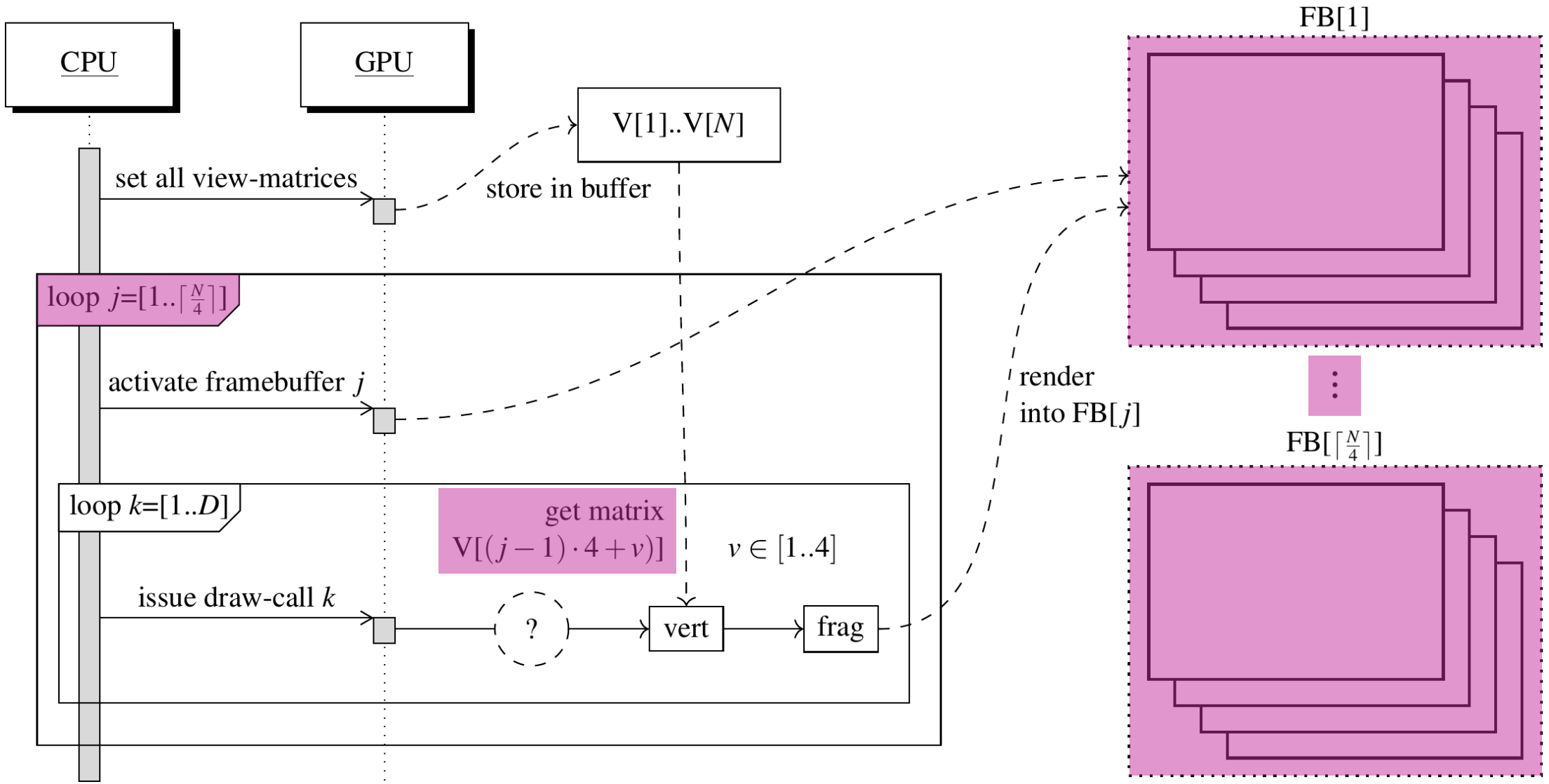
Turing hardware supports **up to four views per pass**, and up to 32 views are supported at the API level.

NVIDIA, TURING GPU ARCHITECTURE





Hardware-Acceleration for ALL Views per Draw-Call



Performance: $\mathcal{P}(1, \rightsquigarrow, \square)$ / $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

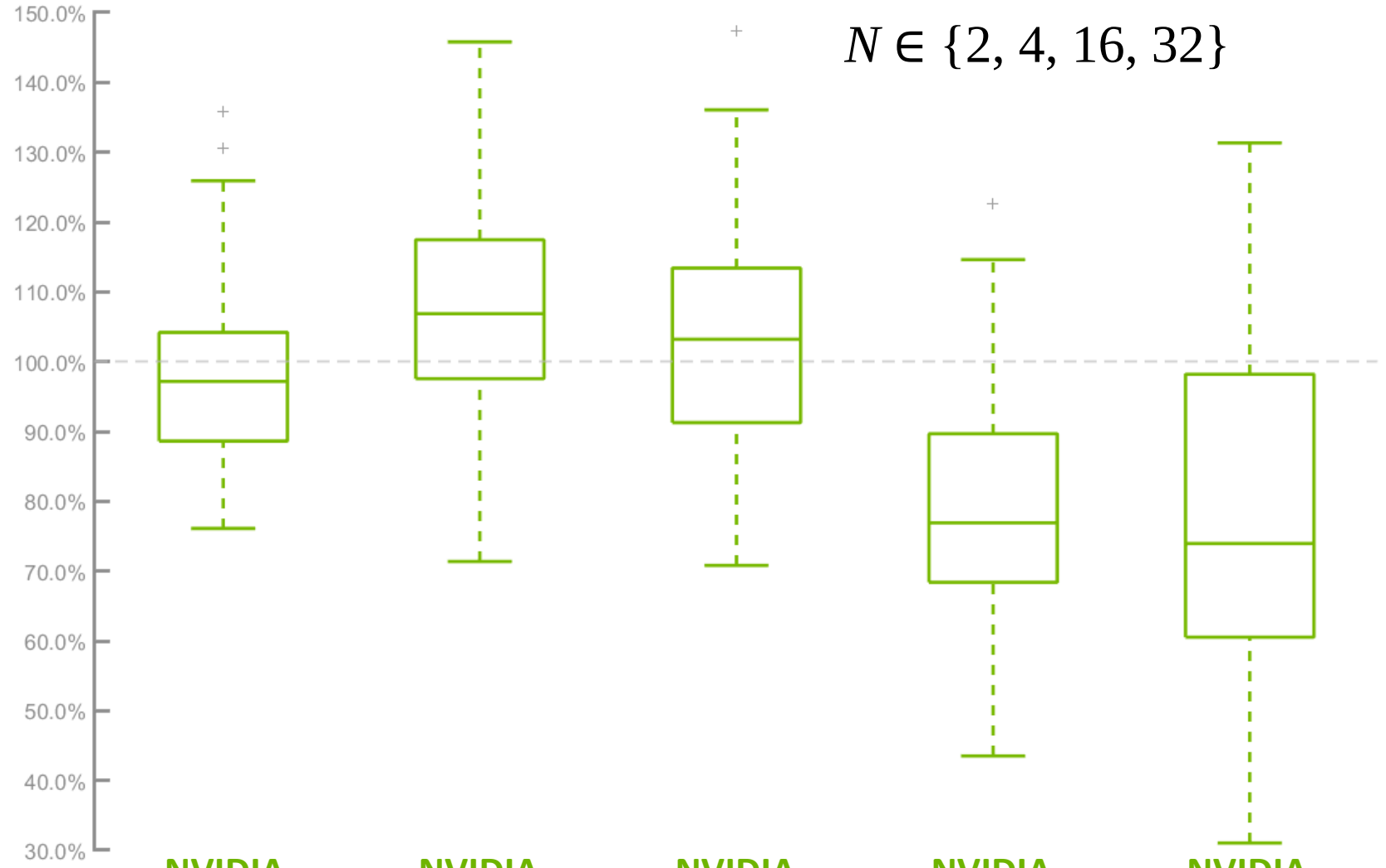
Performance of

$\mathcal{P}(1, \rightsquigarrow, \square)$

w.r.t.

$\mathcal{P}(N, \longmapsto, \square \cdots \square)$

$N \in \{2, 4, 16, 32\}$



Performance: $\mathcal{P}(\ ,\rightrightarrows,\ )$ / $\mathcal{P}(N,\longmapsto,\square\cdots\square)$

$N \in \{2, 4, 16, 32\}$

Performance of

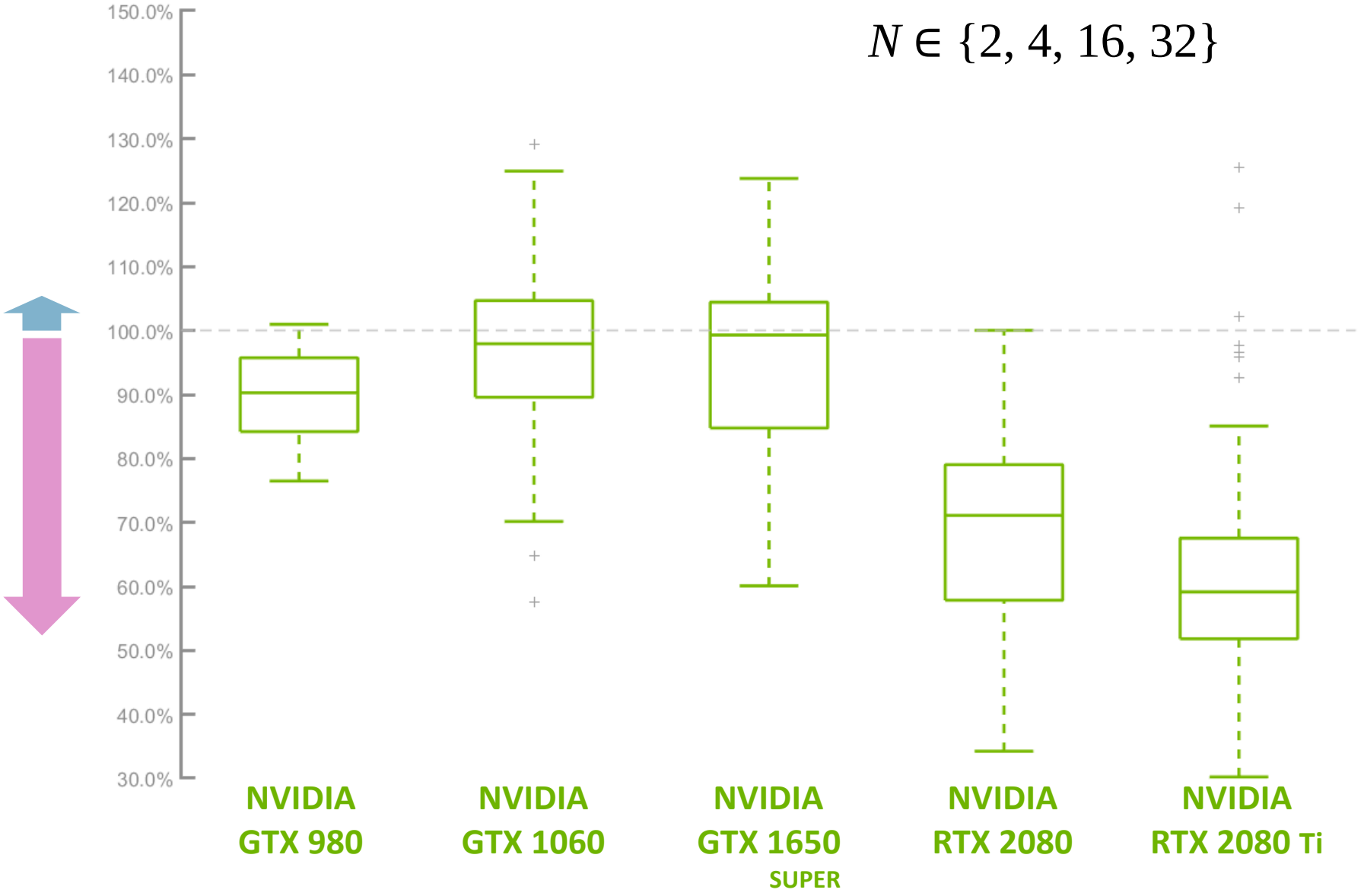
$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square\cdots\square)$

xor

$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square\cdots\square)$

w.r.t.

$\mathcal{P}(N,\longmapsto,\square\cdots\square)$



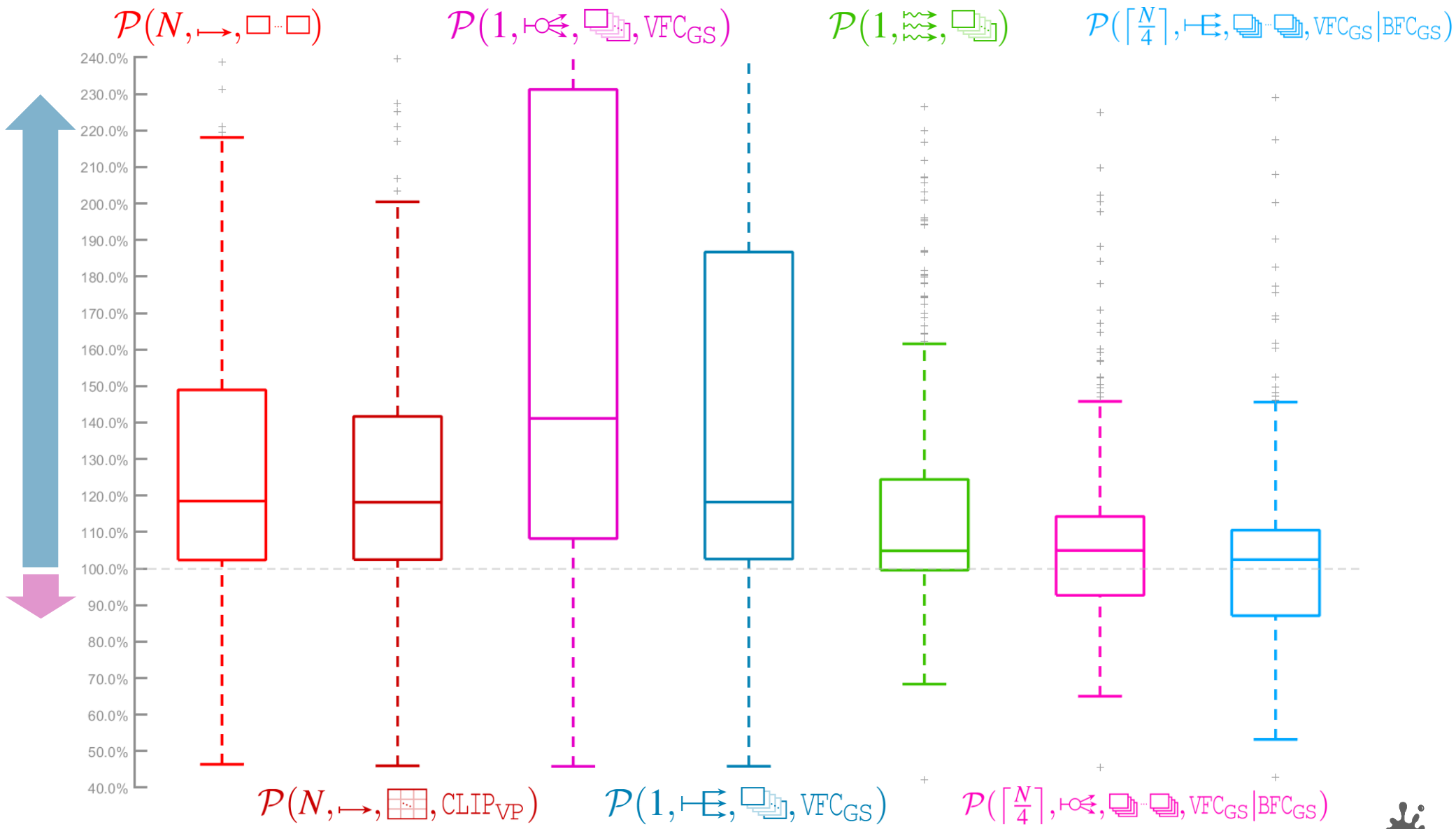
Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

Performance w.r.t.

$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$

xor

$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square \cdots \square)$

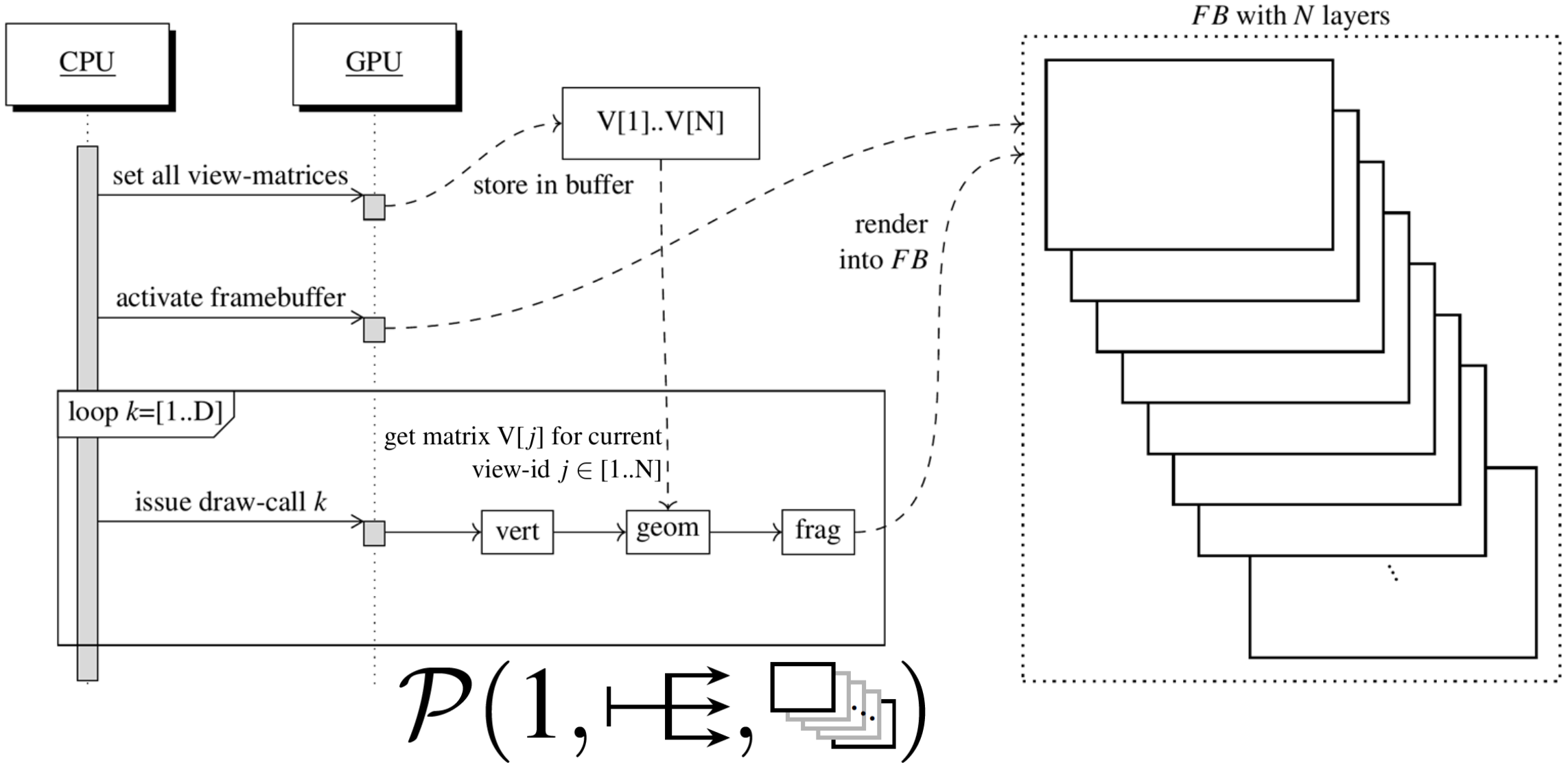


4/5

Optimizing Geometry-Shader-Based Pipeline Variants

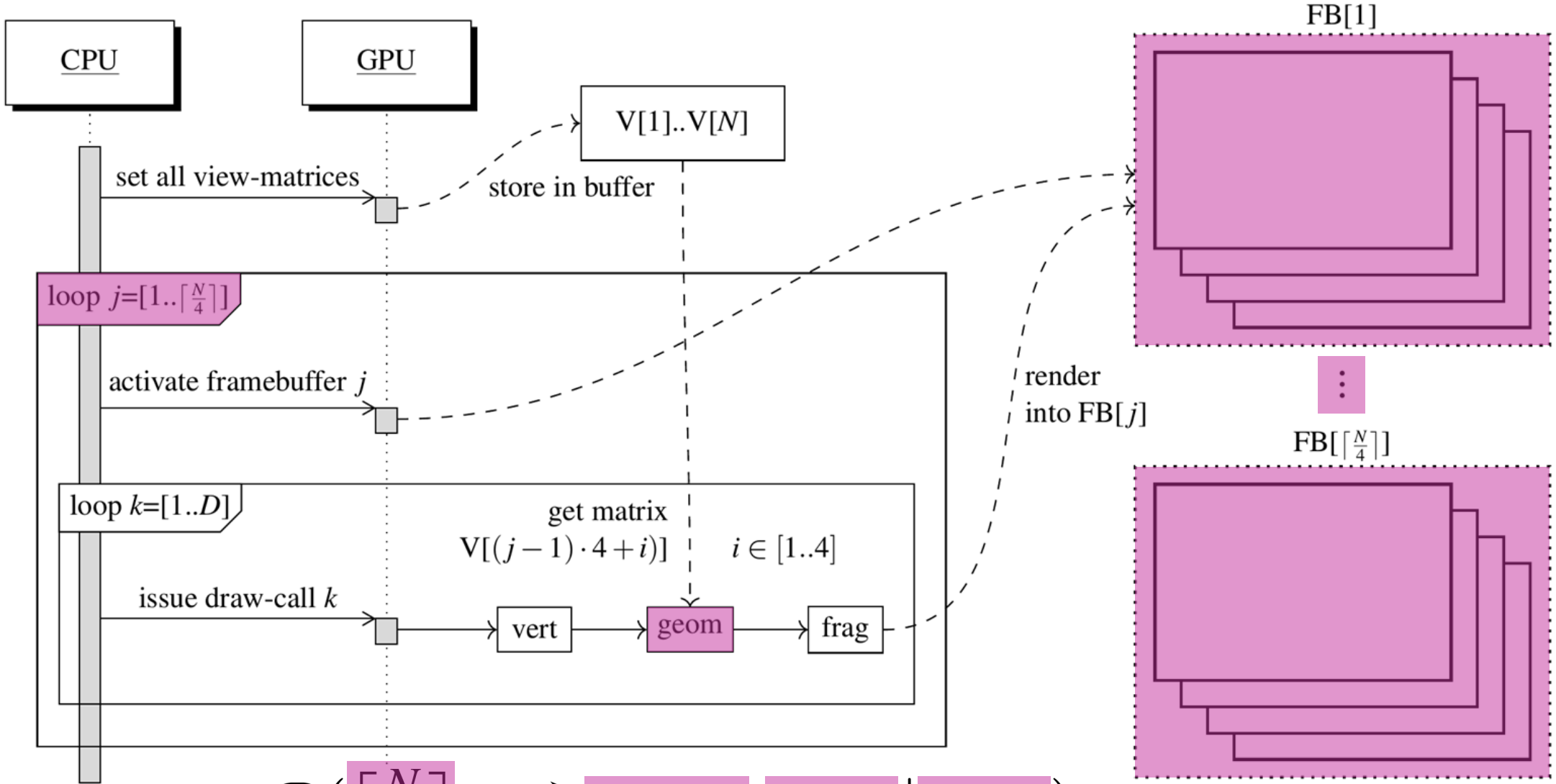
Culling, and Smaller Batches of Views

Recap: Geometry Shader Instancing



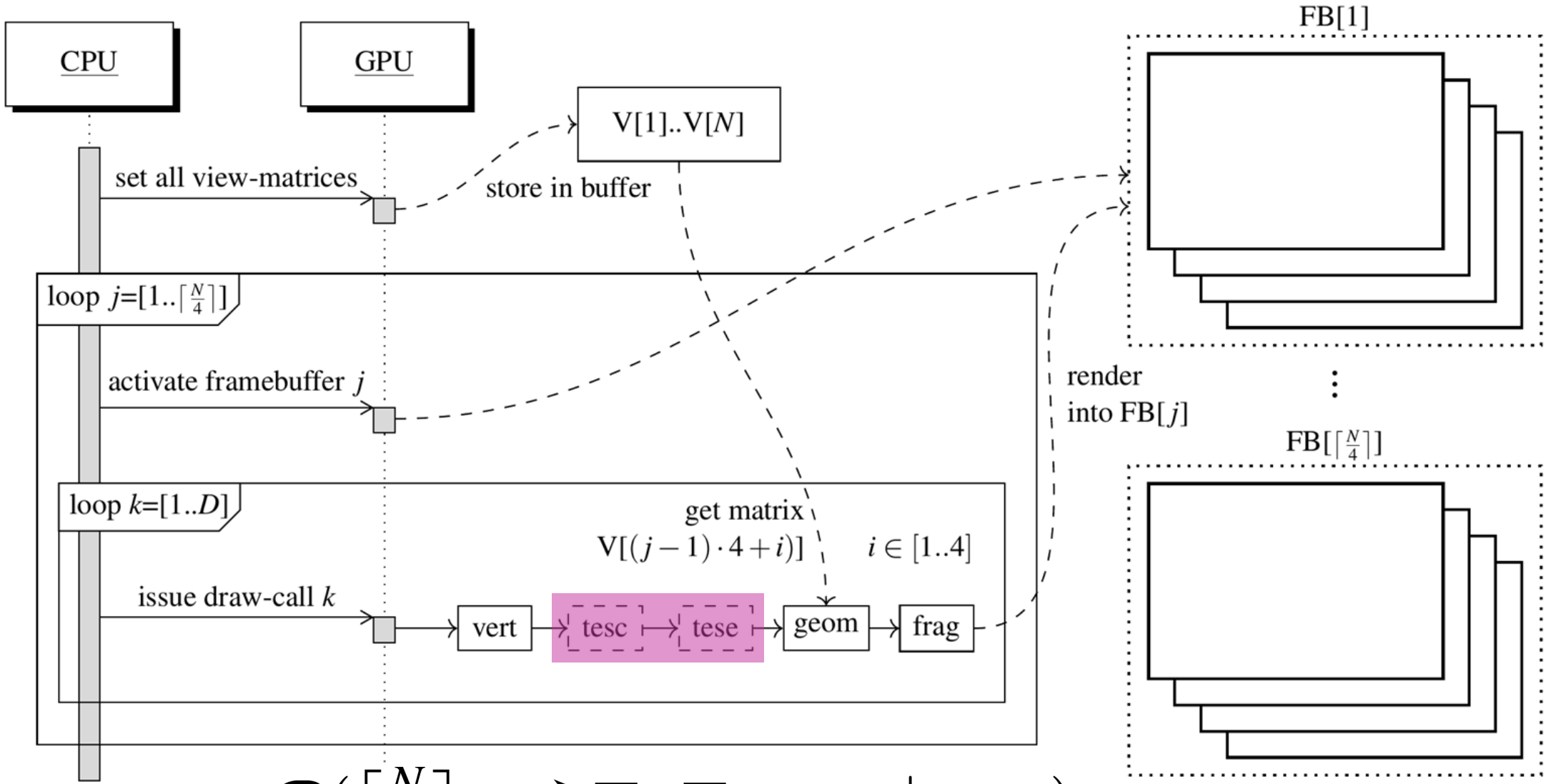
$$\mathcal{P}\left(1, \vdash\!\!\rightrightarrows, \square\right)$$

Geometry Shader Instancing OPTIMIZED



$$\mathcal{P}\left(\left\lceil \frac{N}{4} \right\rceil, \vdash\!\!\!\equiv, \square\!\!\square\!\!\square, \mathrm{VFC}_{\mathrm{GS}} \,|\, \mathrm{BFC}_{\mathrm{GS}}\right)$$

Geometry Shader Instancing OPTIMIZED



$$\mathcal{P}\left(\left\lceil \frac{N}{4} \right\rceil, \vdash\!\!\!\rightrightarrows, \square \cdots \square, \mathrm{VFC}_{\mathrm{GS}} \middle| \mathrm{BFC}_{\mathrm{GS}}\right)$$

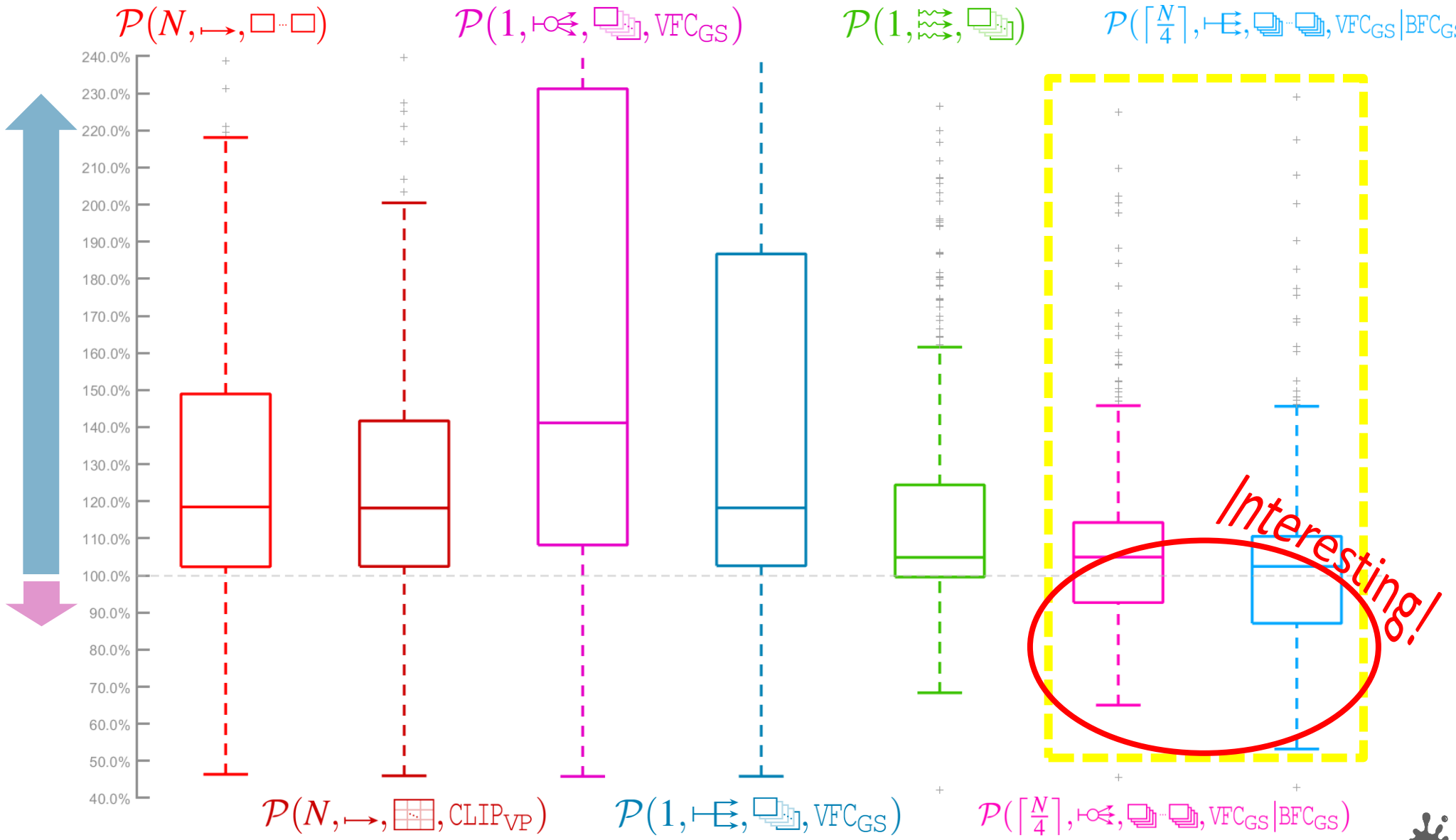
Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

Performance w.r.t.

$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$

xor

$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square\!\square \cdots \square\!\square)$



5/5

Performance Trends

from 8.9 million measurements

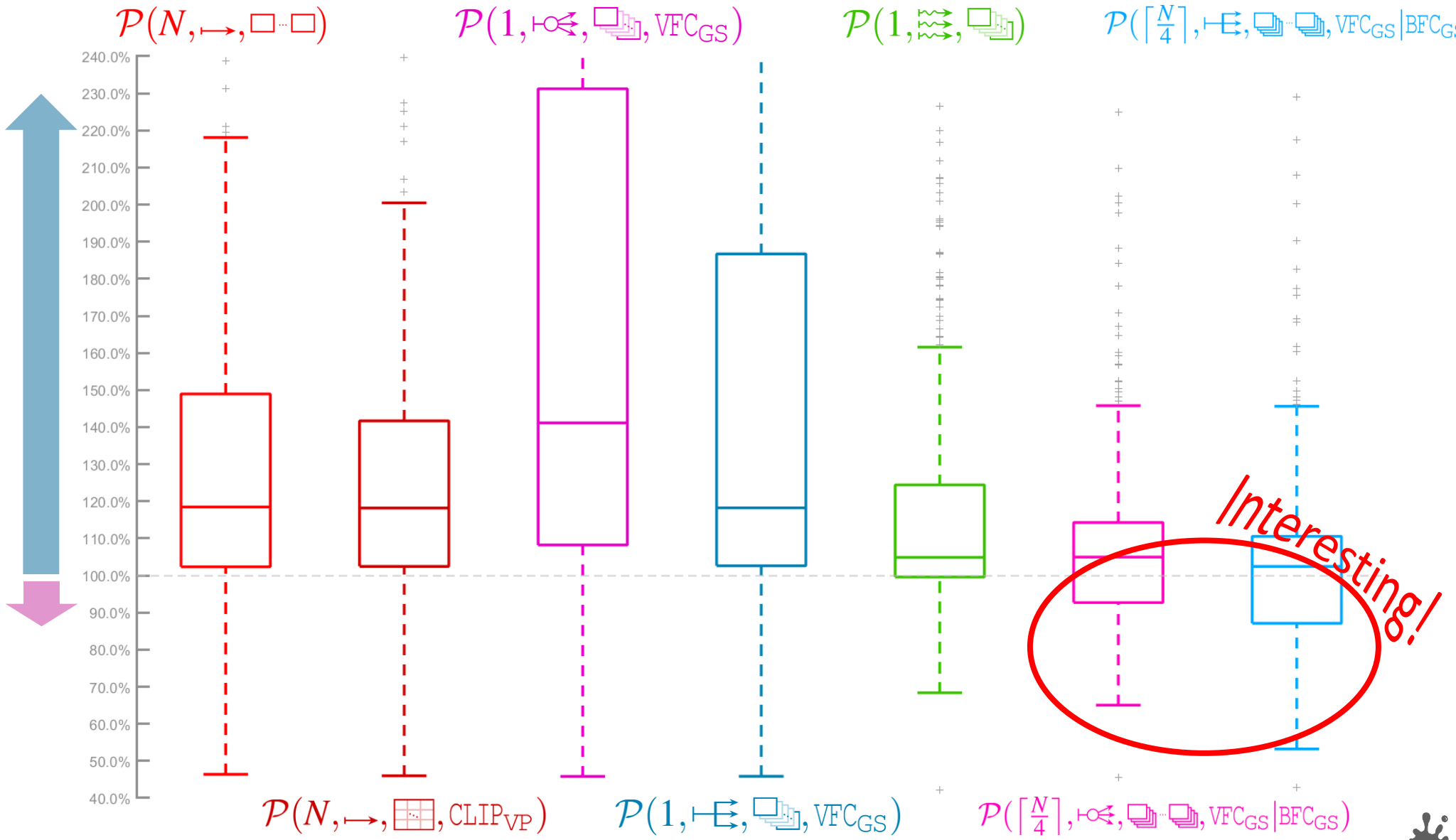
Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

Performance w.r.t.

$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$

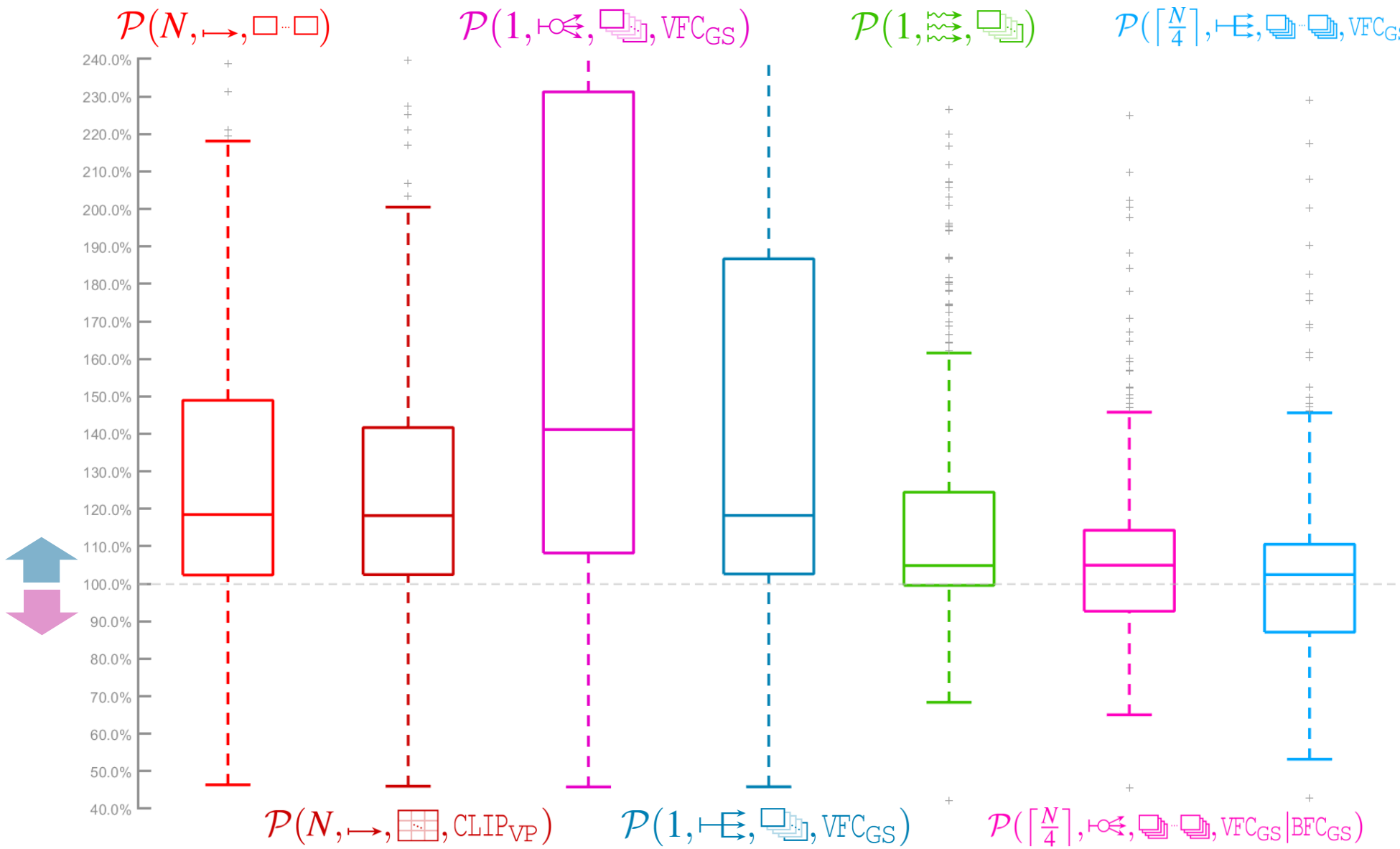
xor

$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square\!\square \cdots \square\!\square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

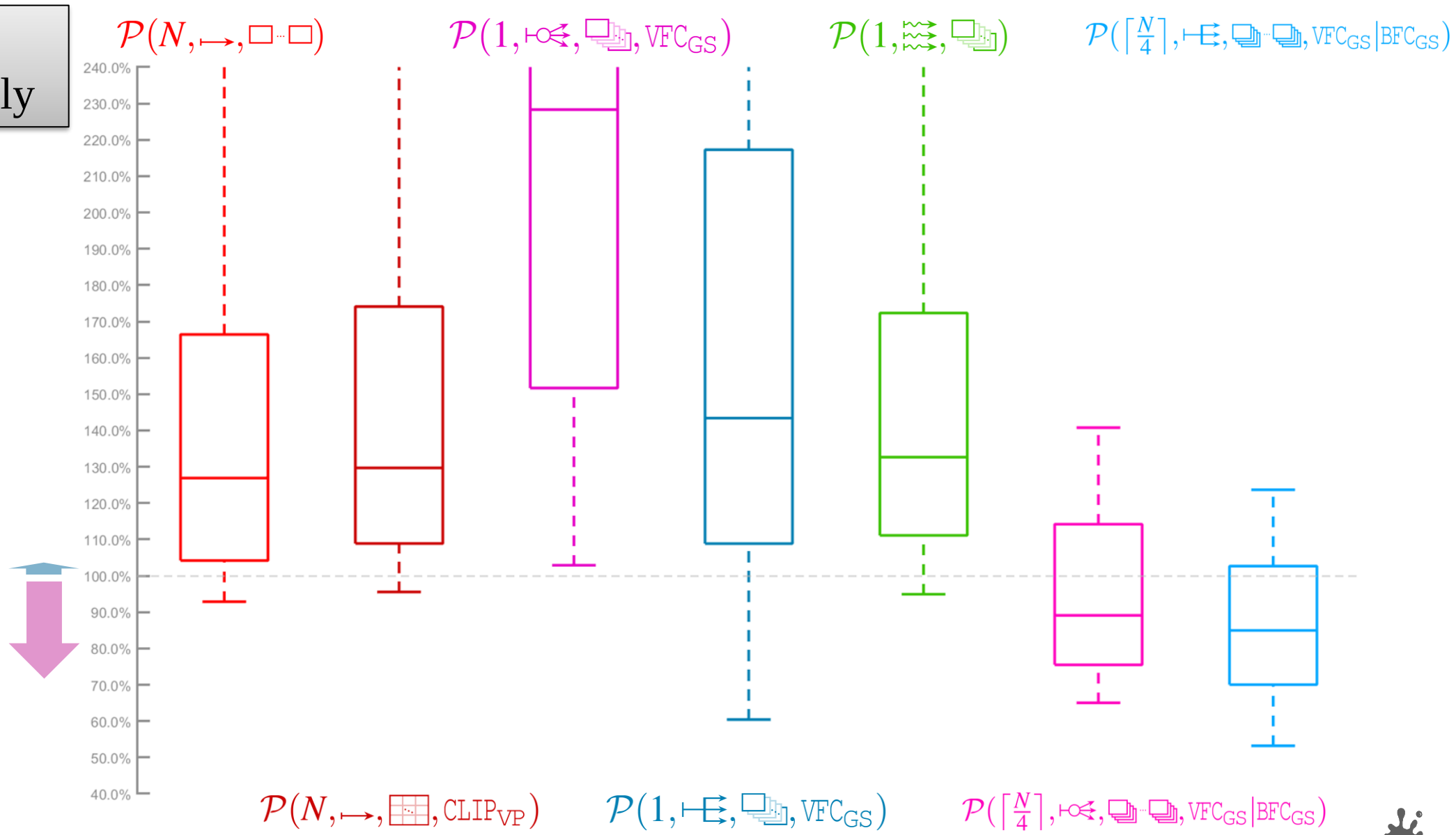
Performance w.r.t. $\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$ xor $\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square \cdots \square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only

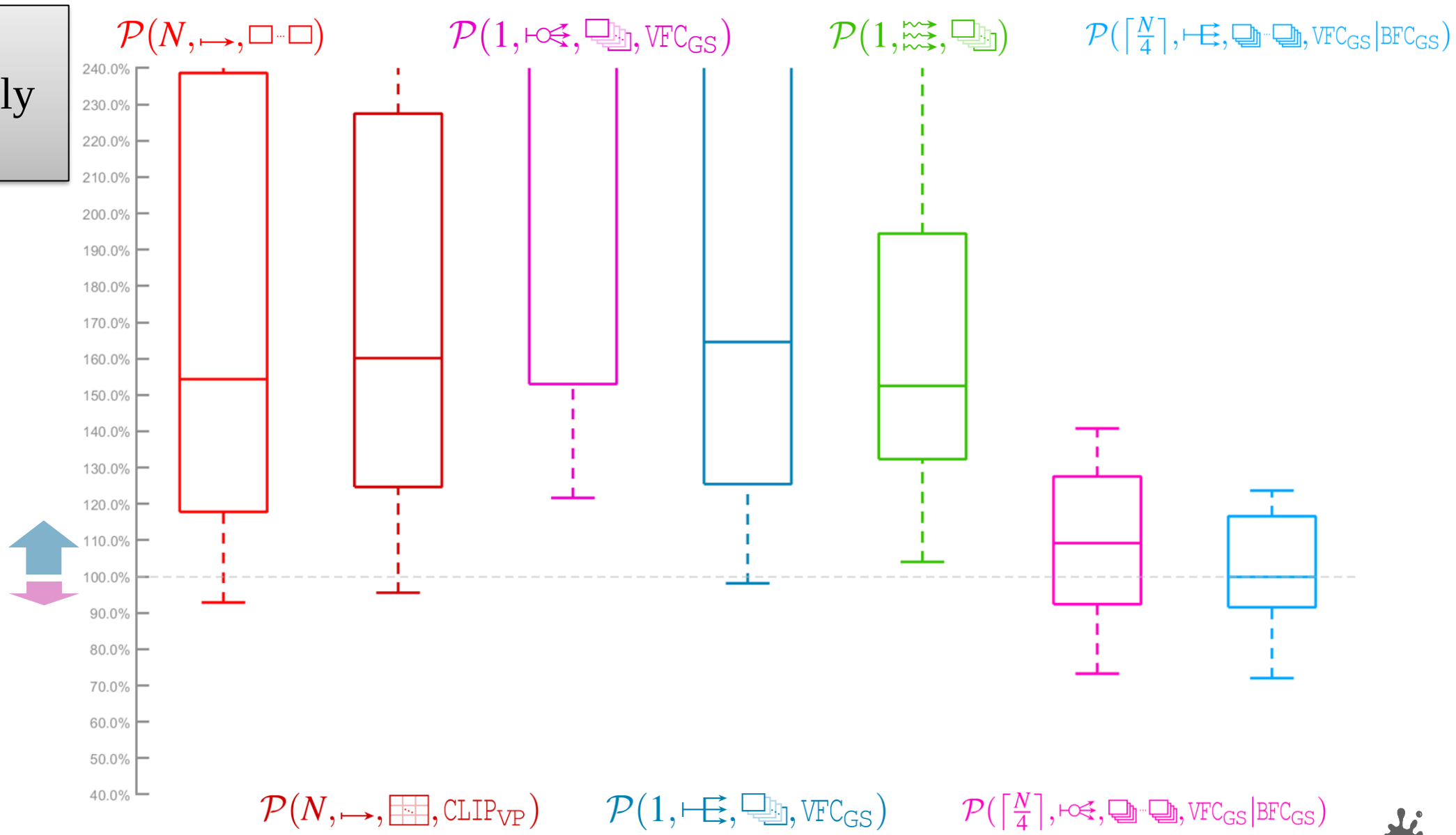
Performance
w.r.t.
$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$
xor
$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square\!\square \cdots \square\!\square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only
Turing only

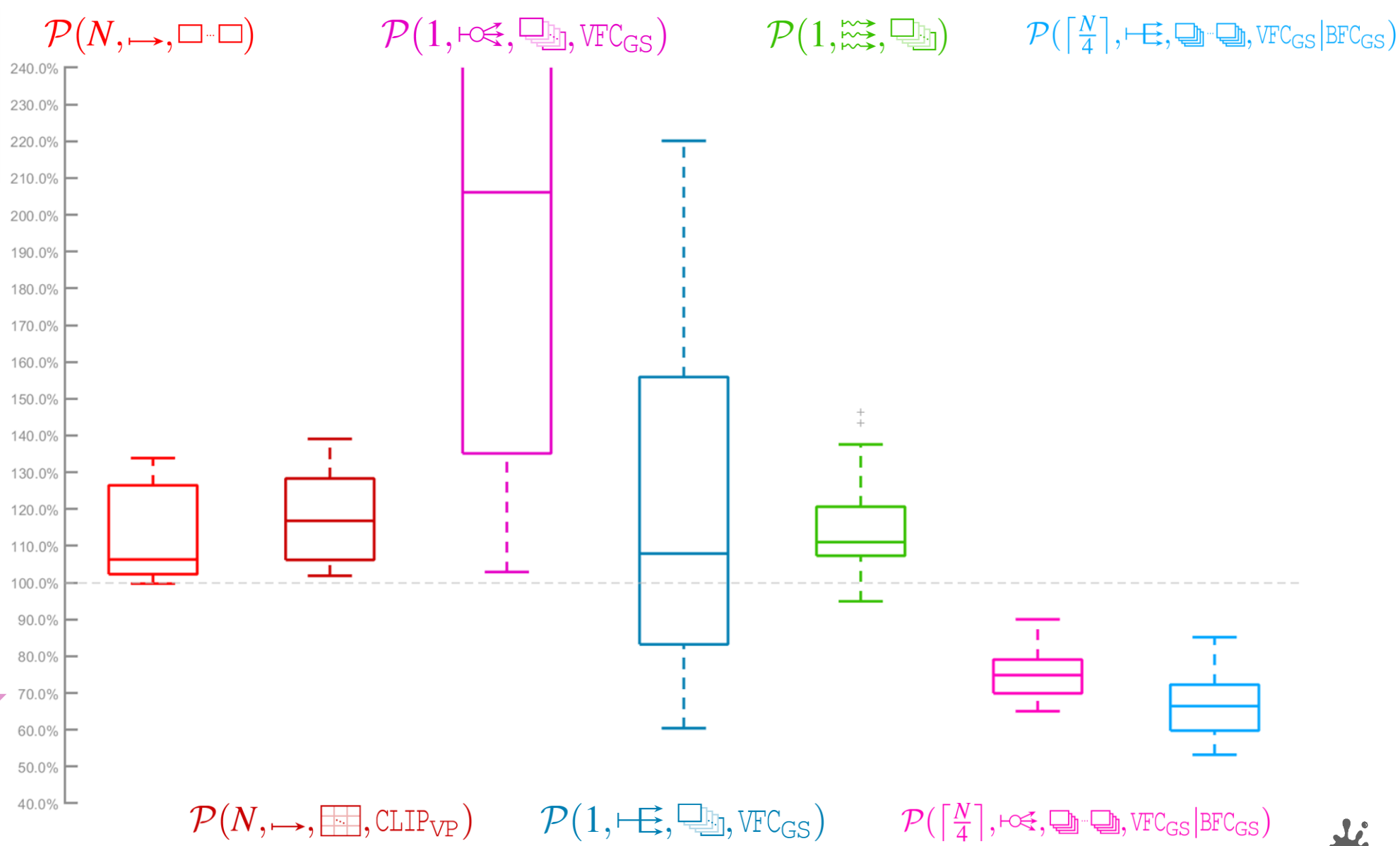
Performance
w.r.t.
$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square\cdots\square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only
Maxwell and Pascal

Performance
w.r.t.
$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only
Only low-tier GPUs
(but also Turing!)

NVIDIA GTX 1060

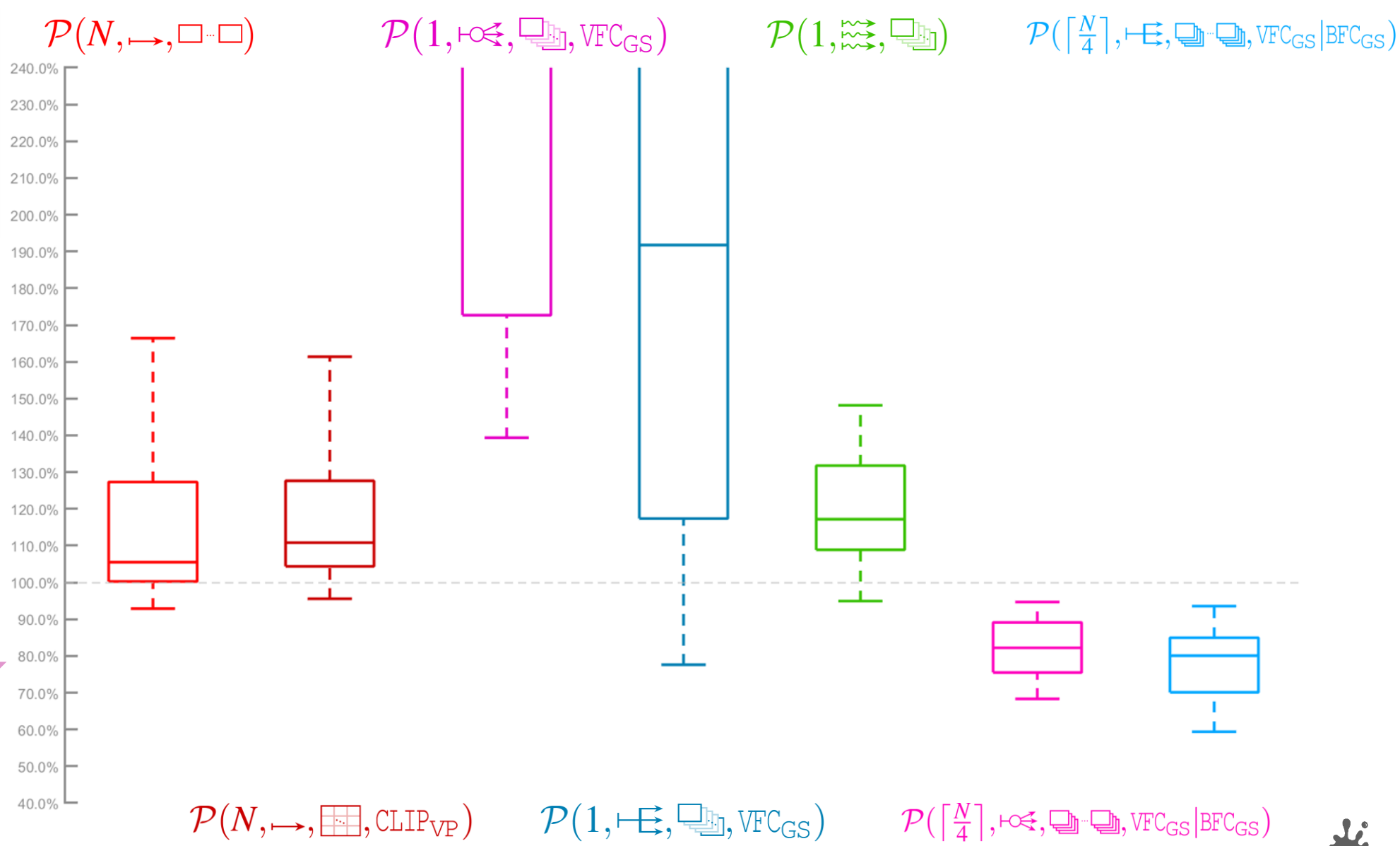
NVIDIA GTX 1650 SUPER

Performance w.r.t.

$\mathcal{P}\left(\left\lceil \frac{N}{2} \right\rceil, \rightrightarrows, \square \cdots \square\right)$

xor

$\mathcal{P}\left(\left\lceil \frac{N}{4} \right\rceil, \rightrightarrows, \square \cdots \square\right)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only
Only low-tier GPUs
(but also Turing!)
Heavy Vertex Load

NVIDIA GTX 1060

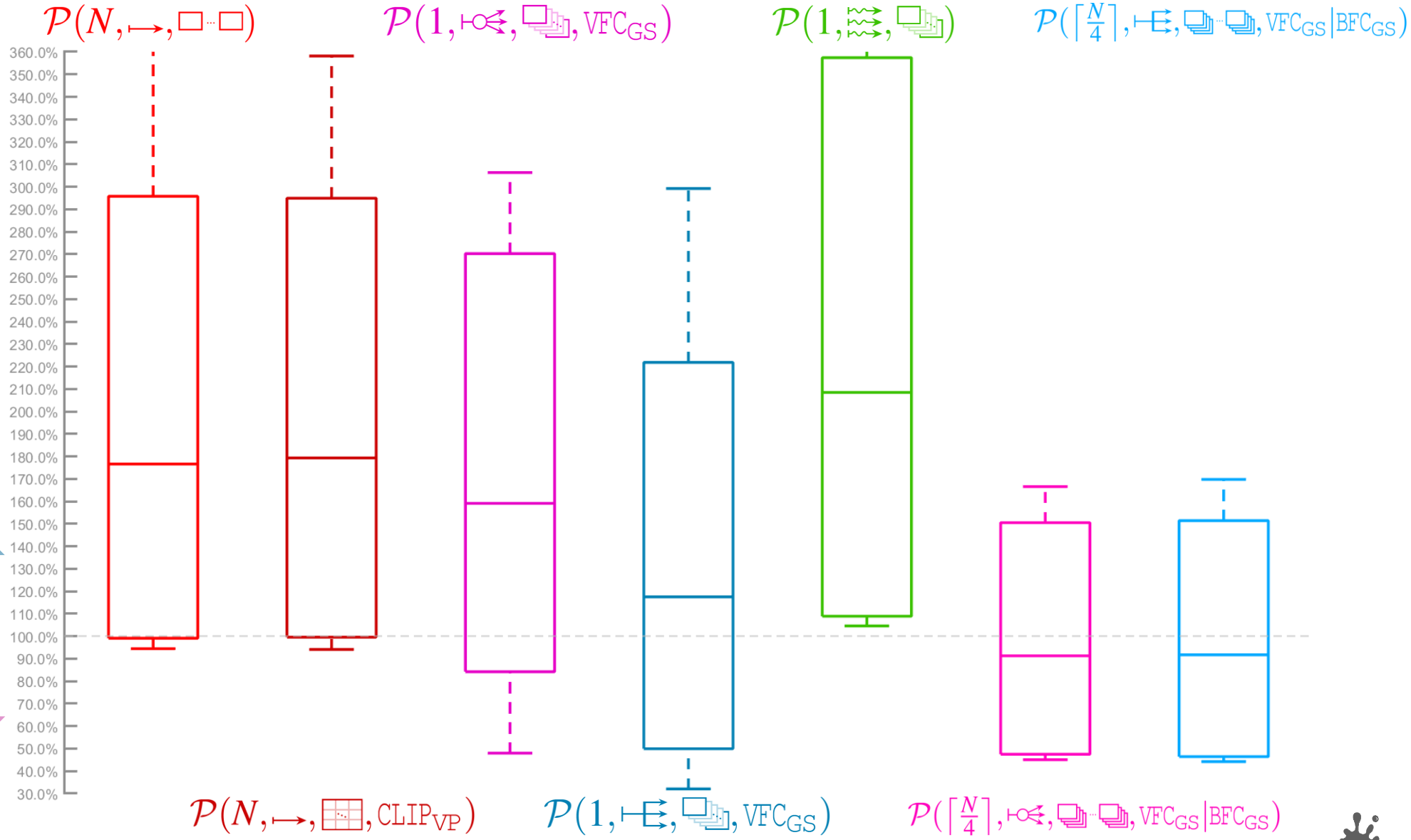
NVIDIA GTX 1650 SUPER

Performance w.r.t.

$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$

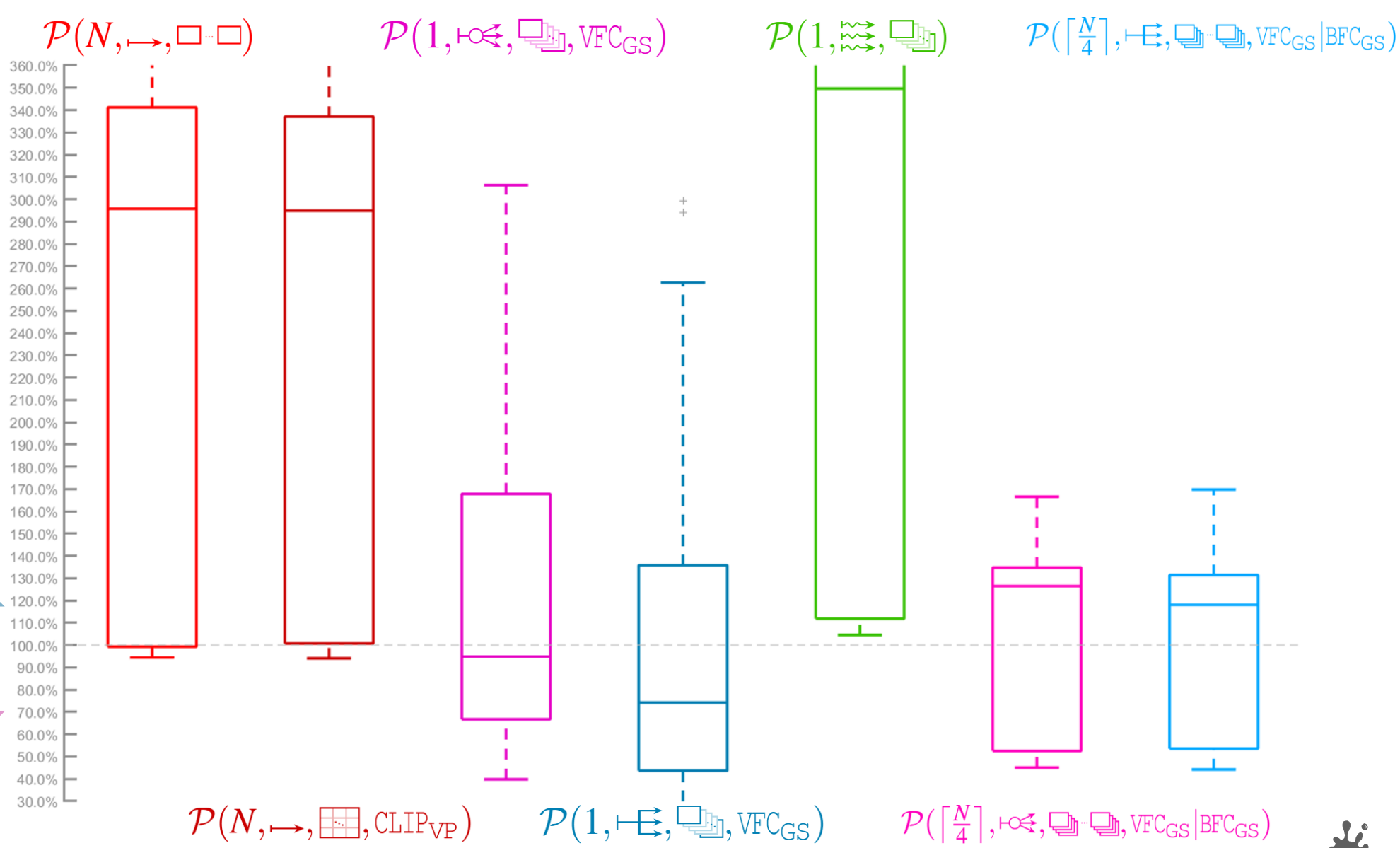
xor

$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square \cdots \square)$



Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large scenes only
Heavy Vertex Load

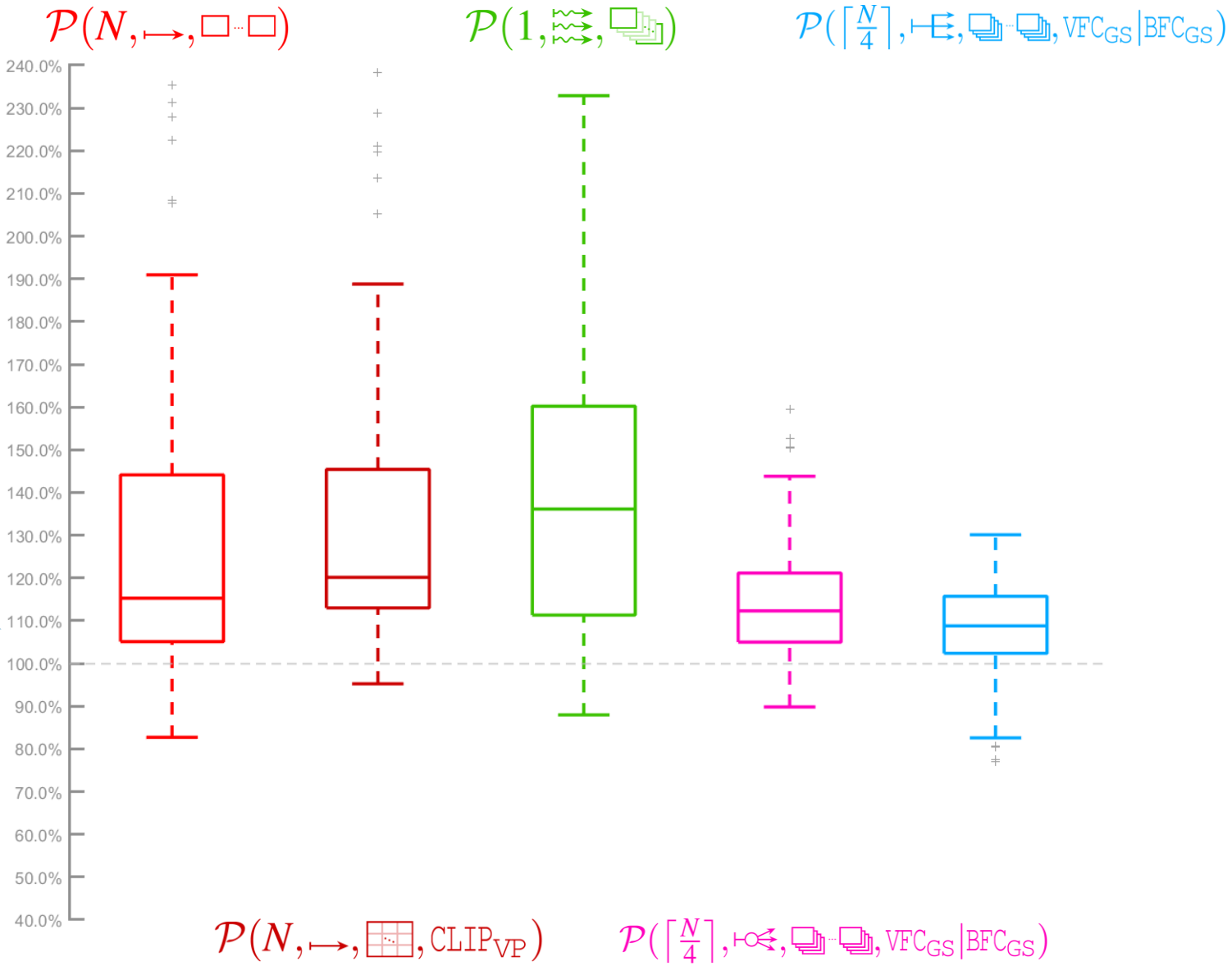


Performance
w.r.t.
$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$
xor
$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square \cdots \square)$

Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

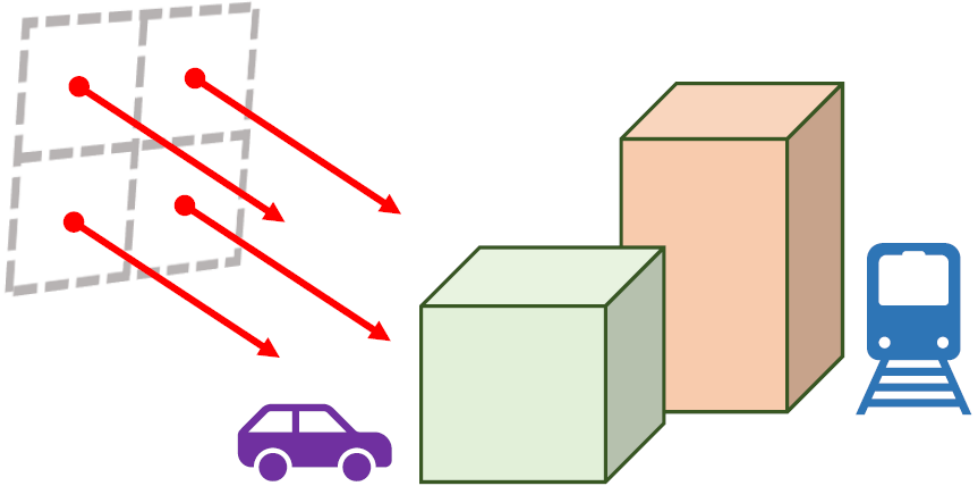
$N \in \{16, 32\}$
Large scenes only
↑ Fragment Load
(G-Buffer Renderng)

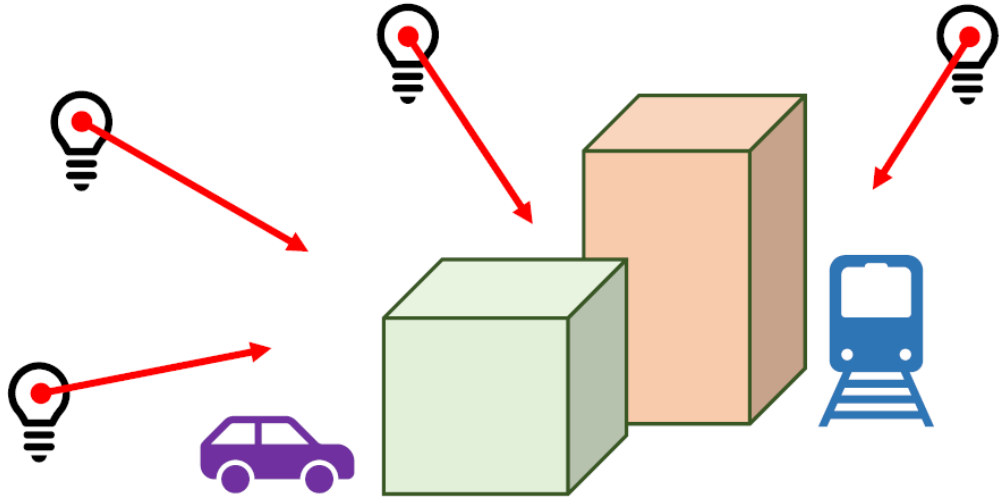
Performance
w.r.t.
$\mathcal{P}(\lceil \frac{N}{2} \rceil, \rightrightarrows, \square \cdots \square)$
xor
$\mathcal{P}(\lceil \frac{N}{4} \rceil, \rightrightarrows, \square \cdots \square)$



Viewport Discrepancy

- Lots of shared vertices for similar views
- What about incoherent view frusta? (e.g. shadow mapping)

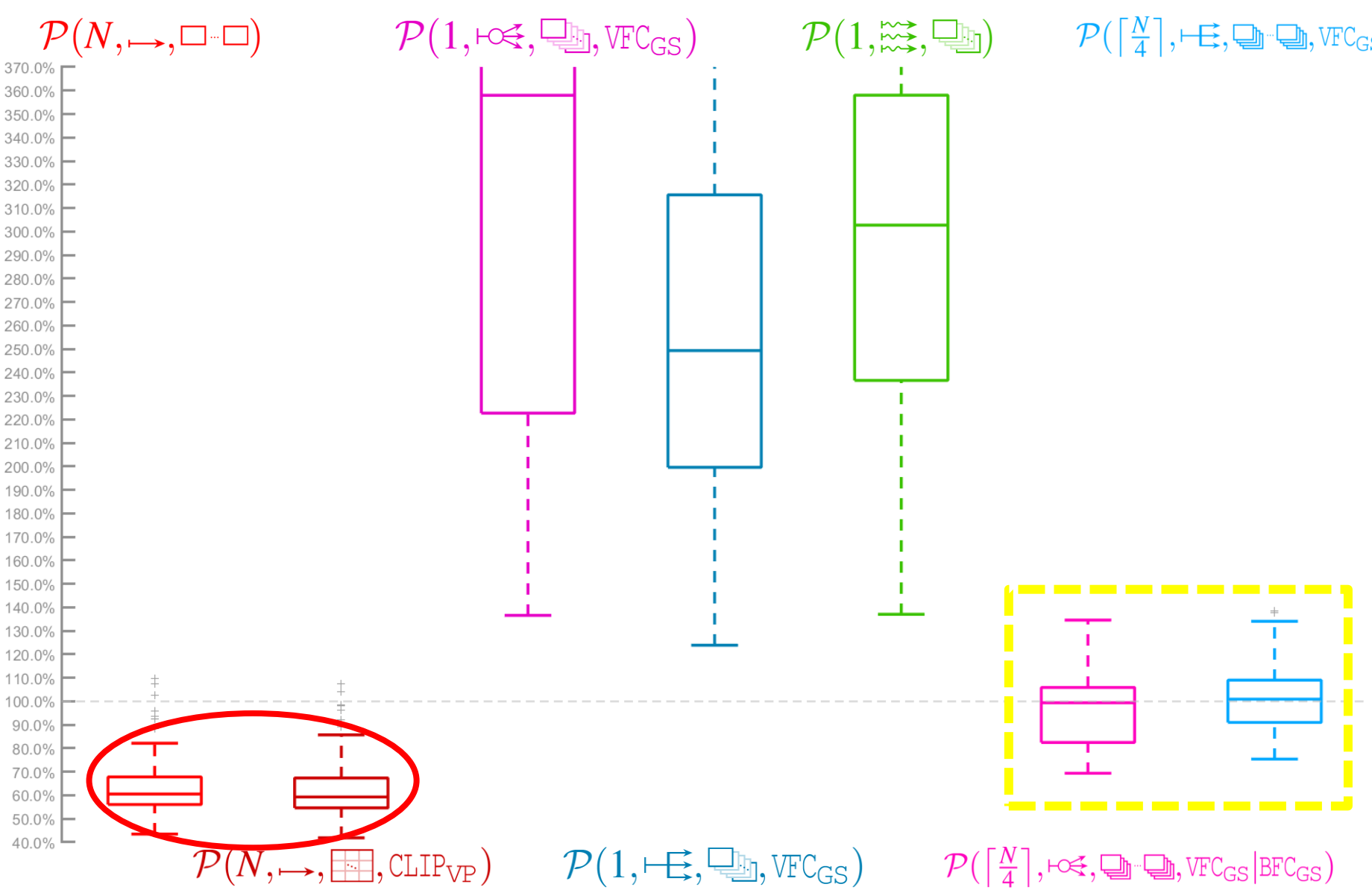




Performance w.r.t. $\mathcal{P}(\ ,\rightrightarrows,\ )$

$N \in \{16, 32\}$
Large Scenes Only
Incoherent Frusta
„Shadow Mapping“

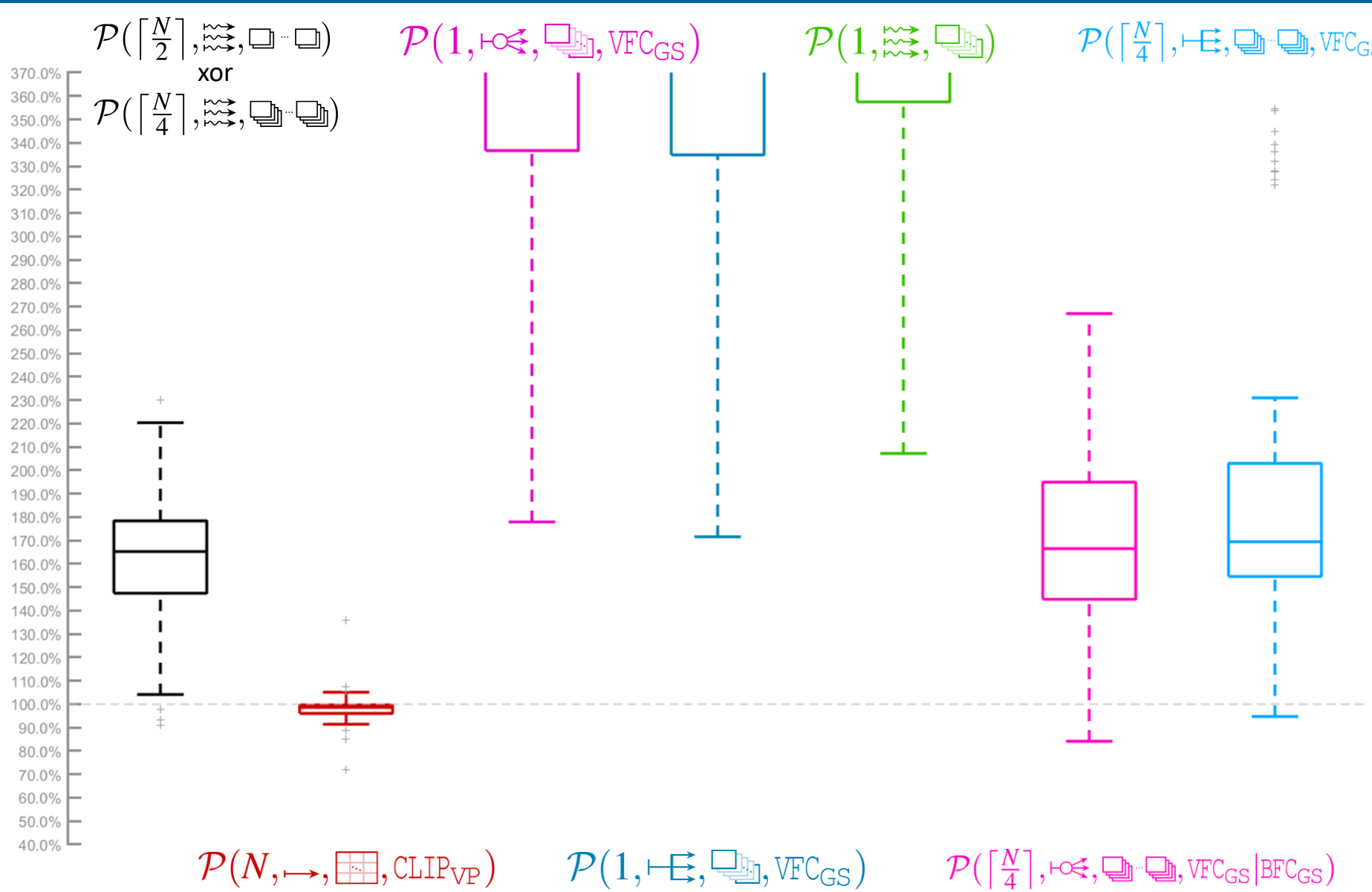
Performance
w.r.t.
$\mathcal{P}\left(\left\lceil \frac{N}{2} \right\rceil, \rightrightarrows, \square \cdots \square\right)$
xor
$\mathcal{P}\left(\left\lceil \frac{N}{4} \right\rceil, \rightrightarrows, \square\!\square \cdots \square\!\square\right)$



Performance w.r.t. $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

$N \in \{16, 32\}$
Large Scenes Only
Incoherent Frusta
„Shadow Mapping“

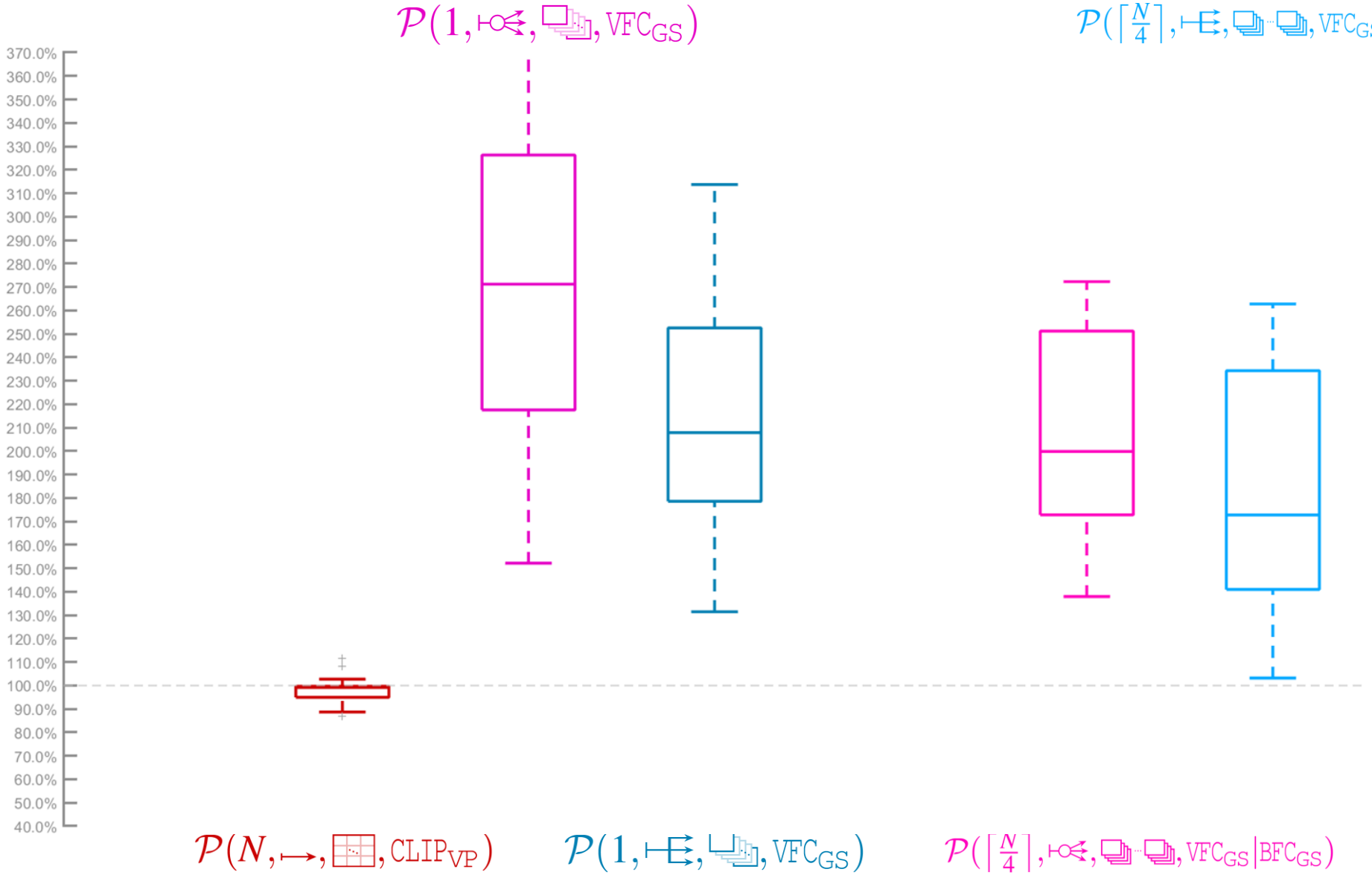
Performance
w.r.t.
$\mathcal{P}(N, \longmapsto, \square \cdots \square)$



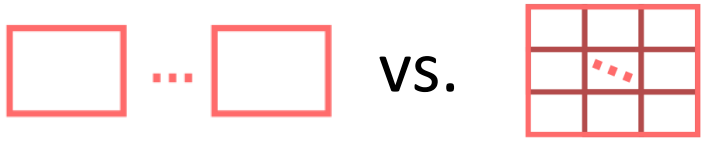
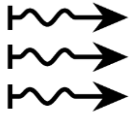
Performance w.r.t. $\mathcal{P}(N, \longmapsto, \square \cdots \square)$

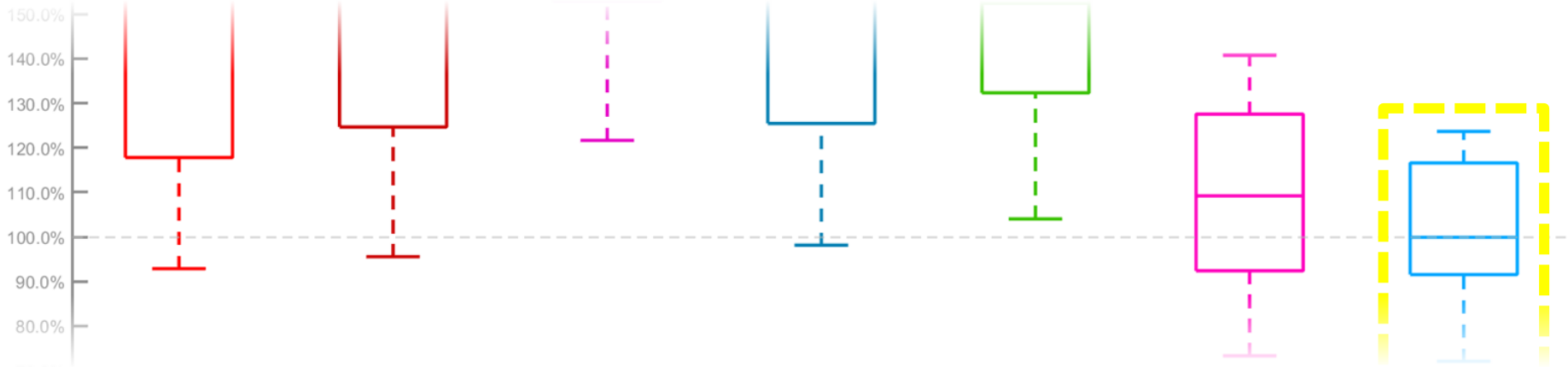
AMD
RX 580

Performance
w.r.t.
$\mathcal{P}(N, \longmapsto, \square \cdots \square)$



Conclusion

- Different Pipeline Variants  vs.
- The Journey of the Triangles 
- Hardware-Accelerated Multi-View Rendering 
- Optimizing Geometry-Shader-Based Pipeline Variants
- Performance Trends $\mathcal{P}\left(\left\lceil \frac{N}{4} \right\rceil, \mapsto, \square \cdots \square, \mathrm{VFC_{GS}} \middle| \mathrm{BFC_{GS}}\right)$



Thank You For Your Attention

Fast Multi-View Rendering for Real-Time Applications

Johannes Unterguggenberger[1], Bernhard Kerbl[1], Markus Steinberger[2], Dieter Schmalstieg[2], Michael Wimmer[1]

[1] TU Wien, Institute of Visual Computing & Human-Centered Technology, Austria

[2] Graz University of Technology, Austria