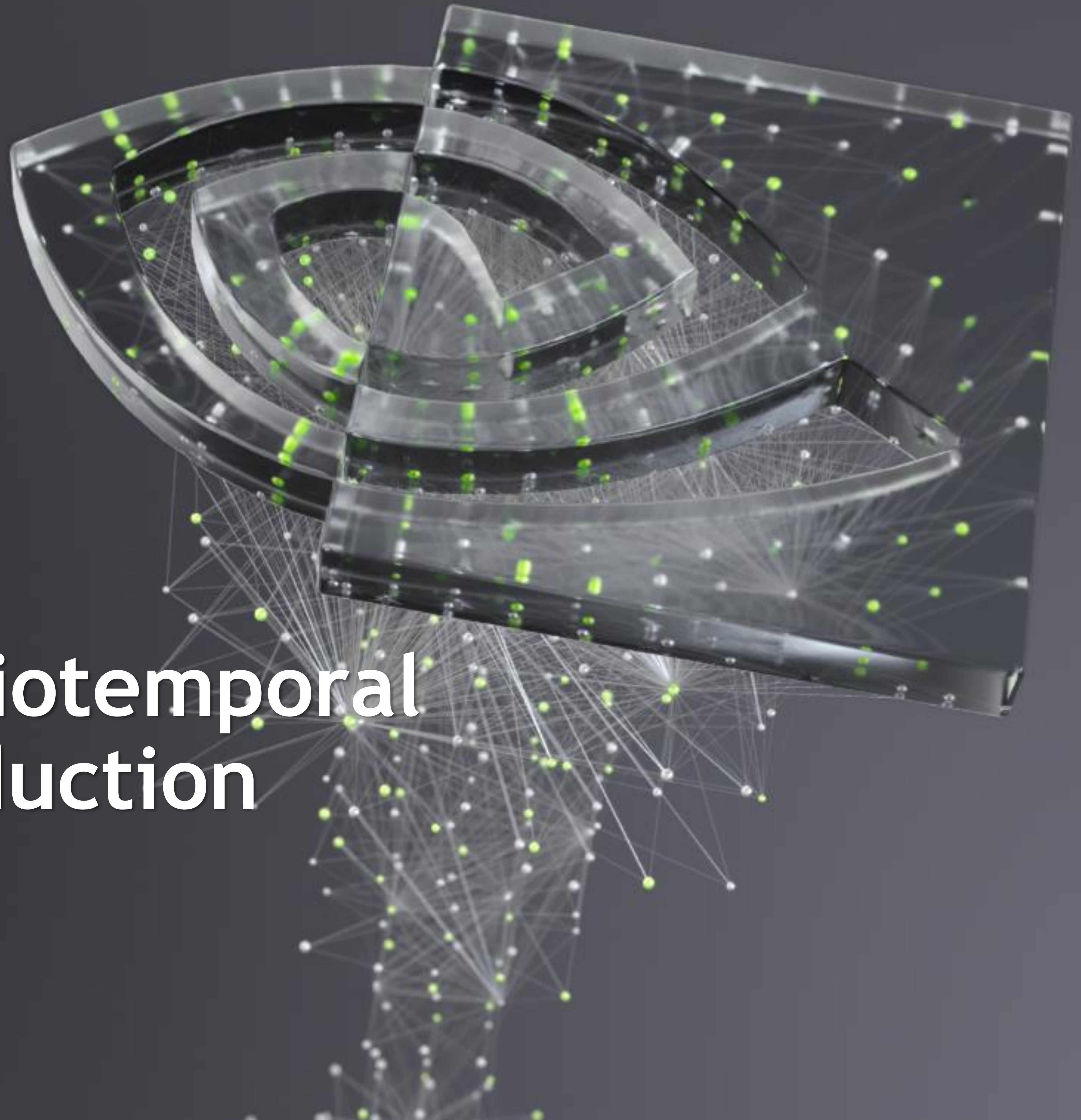


NVIDIA

Rearchitecting Spatiotemporal Resampling for Production

Chris Wyman and Alexey Panteleev

High Performance Graphics, July 7, 2021

IMPORTANT RESAMPLING FOR DIRECT LIGHTING

Bitterli et al. 2020, “Spatiotemporal reservoir resampling for real-time ray tracing…”

Leverage correlations to improve quality

60x better than prior state of art

IMPORTANT RESAMPLING FOR DIRECT LIGHTING

Bitterli et al. 2020, "Spatiotemporal reservoir resampling for real-time ray tracing…"

Leverage correlations to improve quality

- 60x better than prior state of art

For use in production, requires:

- Lower, more consistent cost
- Varied light primitives
- Fewer shadow rays
- Tunable quality / perf knobs
- Better handle on bias

GOALS OF OUR HPG PAPER

Move ReSTIR from research prototype (closer) to “production ready”

GOALS OF OUR HPG PAPER

Move ReSTIR from research prototype (closer) to “production ready”

Improve clarity for those reimplementing the algorithm:

- Improve intuition for underlying mathematics
- Develop visual intuition for causes (and avoidance) of bias

GOALS OF OUR HPG PAPER

Move ReSTIR from research prototype (closer) to "production ready"

Improve clarity for those reimplementing the algorithm:

- Improve intuition for underlying mathematics
- Develop visual intuition for causes (and avoidance) of bias

Please see paper or my other talks (e.g., at GTC)

GOALS OF OUR HPG PAPER

Move ReSTIR from research prototype (closer) to “production ready”

Improve clarity for those reimplementing the algorithm:

- Improve intuition for underlying mathematics
- Develop visual intuition for causes (and avoidance) of bias

Please see paper or my other talks (e.g., at GTC)

Improve performance, including:

- Improve memory coherence (use resampling **reshape** computations more efficiently)
- Reduce number of shadow rays required
- Decouple shading from reuse (improves quality at a give performance)

GOALS OF OUR HPG PAPER

Move ReSTIR from research prototype (closer) to "production ready"

Improve clarity for those reimplementing the algorithm:

- Improve intuition for underlying mathematics
- Develop visual intuition for causes (and avoidance) of bias

Please see paper or my other talks (e.g., at GTC)

Improve performance, including:

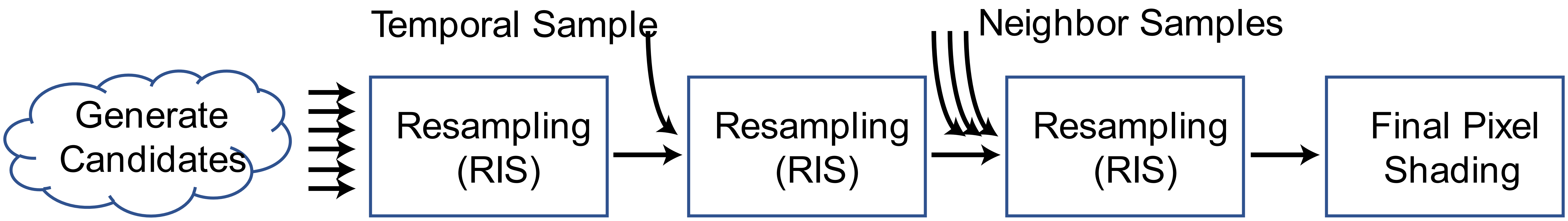
- Improve memory coherence (use resampling **reshape** computations more efficiently)
- Reduce number of shadow rays required
- Decouple shading from reuse (improves quality at a give performance)

Going to discuss today

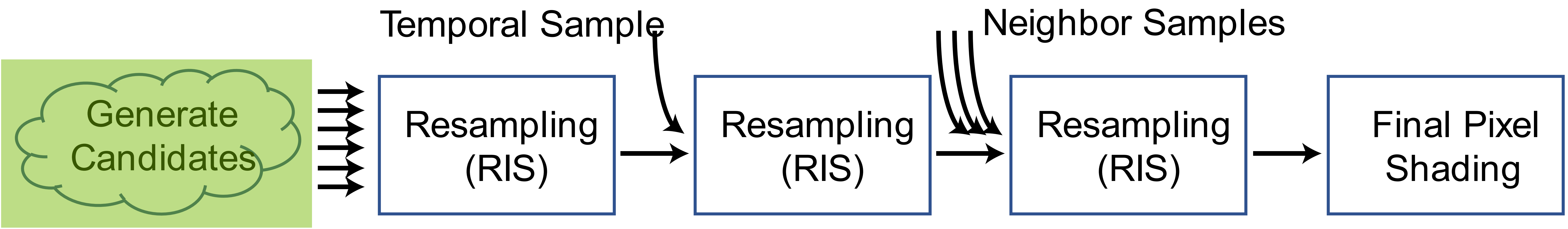
QUICK REVIEW

HOW DOES RESAMPLING WORK?

HOW DOES RESAMPLING WORK?

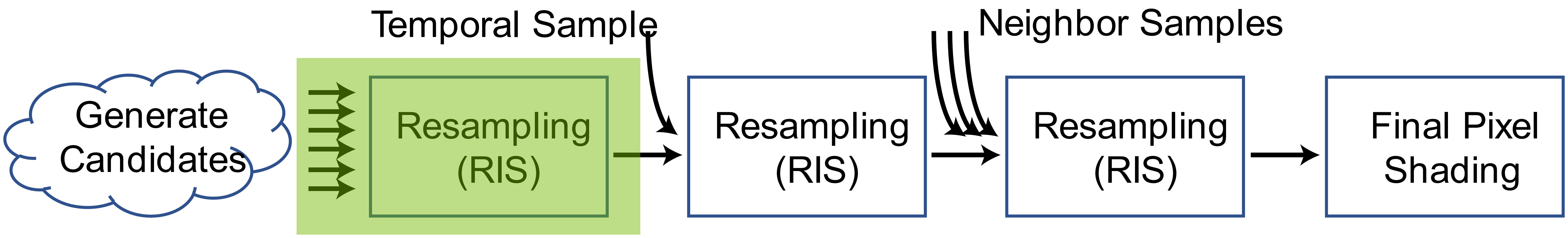


HOW DOES RESAMPLING WORK?



Each pixel (naively) selects a few candidate lights

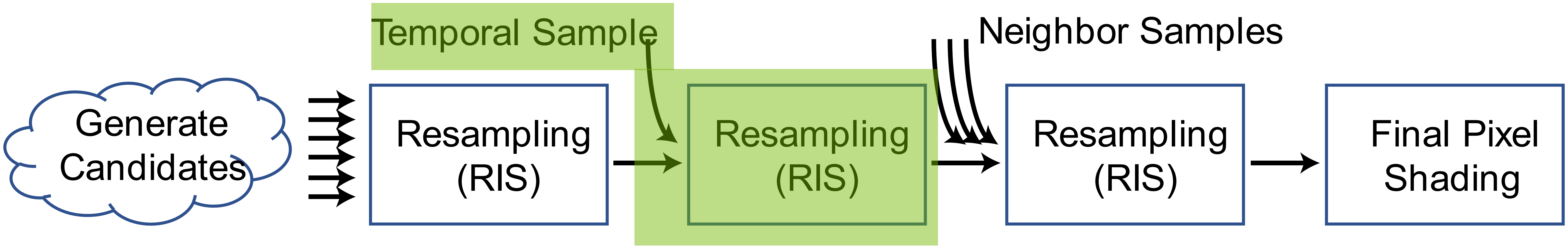
HOW DOES RESAMPLING WORK?



Each pixel (naively) selects a few candidate lights

These get **reweighted** to select one, proportional to their contributions at the current pixel

HOW DOES RESAMPLING WORK?

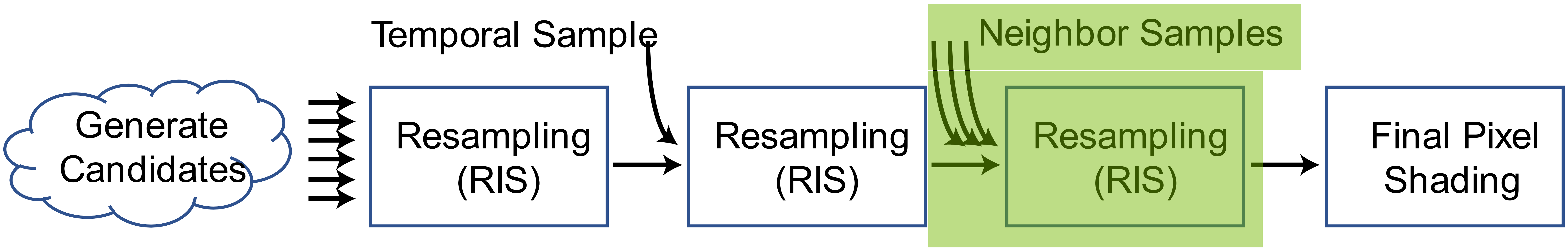


Each pixel (naively) selects a few candidate lights

These get **reweighted** to select one, proportional to their contributions at the current pixel

Reuse (and reweight) last frame's final samples, to increase sample count

HOW DOES RESAMPLING WORK?



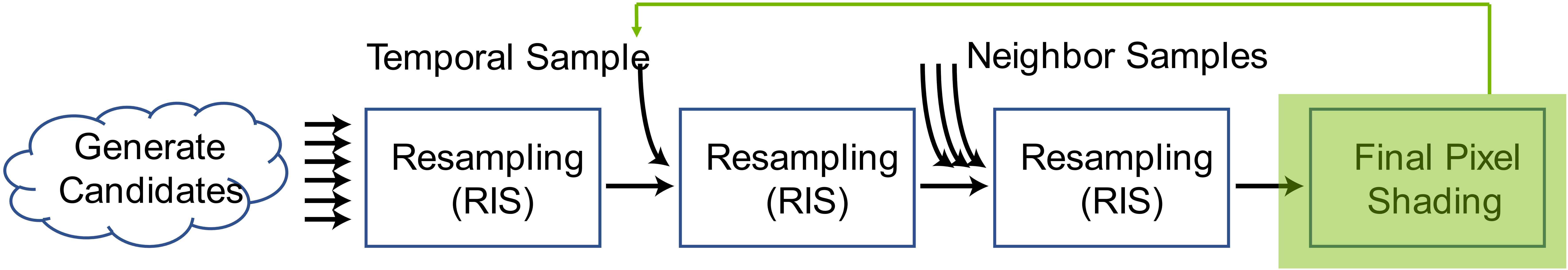
Each pixel (naively) selects a few candidate lights

These get **reweighted** to select one, proportional to their contributions at the current pixel

Reuse (and reweight) last frame’s final samples, to increase sample count

Reuse (and reweight) selected samples at neighbor pixels, to increase sample count

HOW DOES RESAMPLING WORK?



Each pixel (naively) selects a few candidate lights

These get **reweighted** to select one, proportional to their contributions at the current pixel

Reuse (and reweight) last frame's final samples, to increase sample count

Reuse (and reweight) selected samples at neighbor pixels, to increase sample count

Shade the final light selections (after iterative reweighting)

(And save samples to reuse next frame)

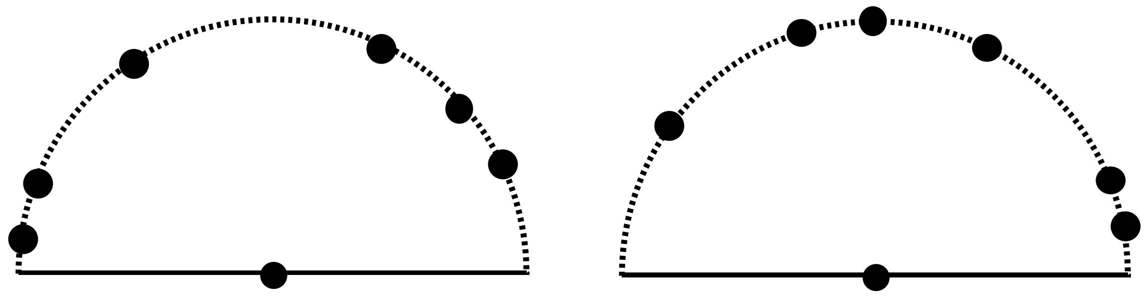
IMPROVING MEMORY COHERENCE

MEMORY INCOHERENCE

Why does it occur?

Complex scene with 3 million dynamic lights

Each pixel independently selects some



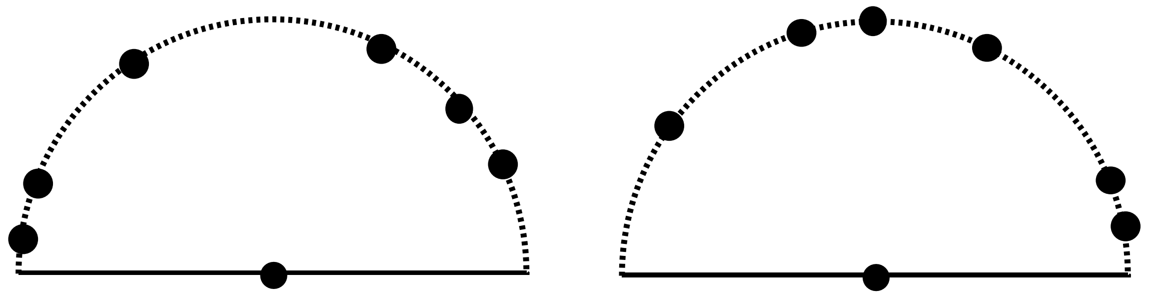
MEMORY INCOHERENCE

Why does it occur?

Complex scene with 3 million dynamic lights

Each pixel independently selects some

- Extremely incoherent; trashes cache



Roughly 60% of cost in [Bitterli et al. 2020]

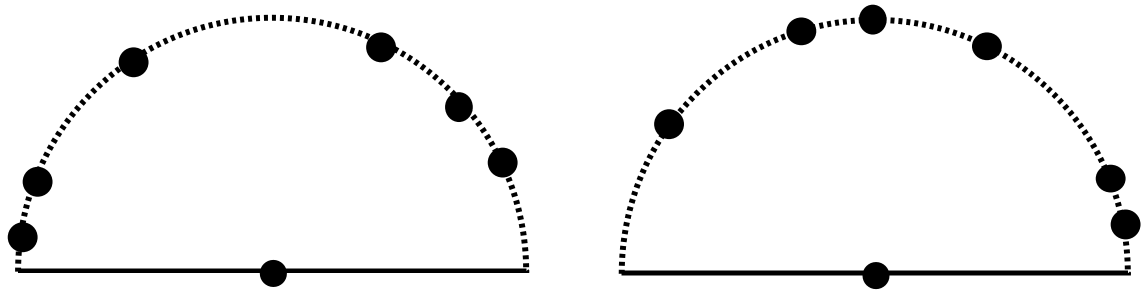
MEMORY INCOHERENCE

Why does it occur?

Complex scene with 3 million dynamic lights

Each pixel independently selects some

- Extremely incoherent; trashes cache
- Cost depends on cache hierarchy
- Sampling varies >20x between scenes



Roughly 60% of cost in [Bitterli et al. 2020]

CAN WE MAKE LOOKUPS (MORE) COHERENT?

Ensure light samples for adjacent pixels are in same cache line?

CAN WE MAKE LOOKUPS (MORE) COHERENT?

Ensure light samples for adjacent pixels are in same cache line?

Use same 32 candidate lights for all pixels in a screen tile?

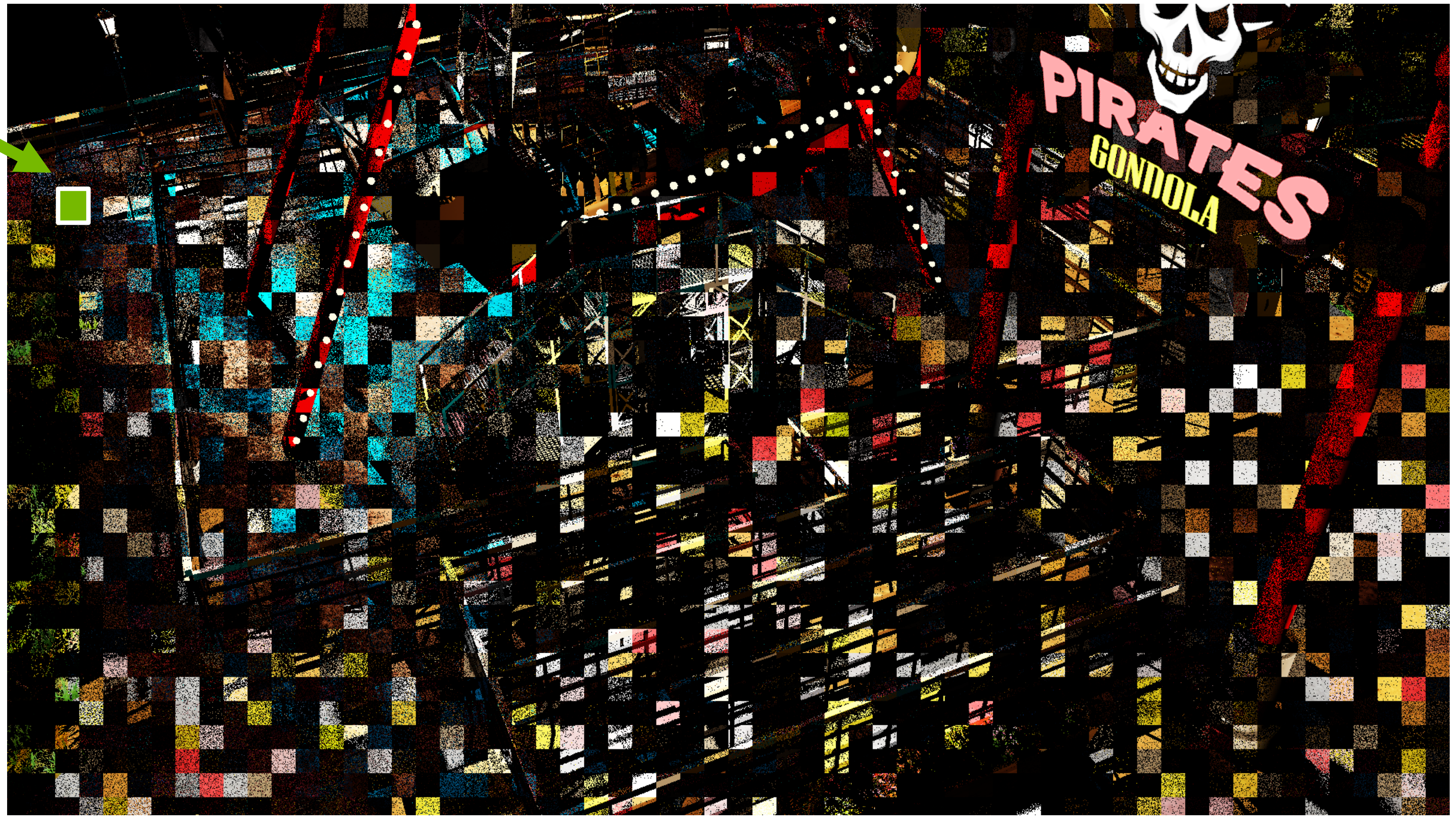
Standard, uncorrelated RIS

$$F \approx \frac{1}{N} \sum_{i=1}^{N} \left[\frac{f(\omega_i)}{\hat{p}(\omega_i)} \frac{1}{M} \sum_{j=1}^{M} \frac{\hat{p}(\omega_{ij})}{q(\omega_{ij})} \right]$$

Correlated form of RIS

$$\approx \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(\omega_i)}{\hat{p}(\omega_i)} \right) \left(\frac{1}{M} \sum_{j=1}^{M} \frac{\hat{p}(\omega_j)}{q(\omega_j)} \right)$$

I.e., M=32
The same 32 for all pixels in a block



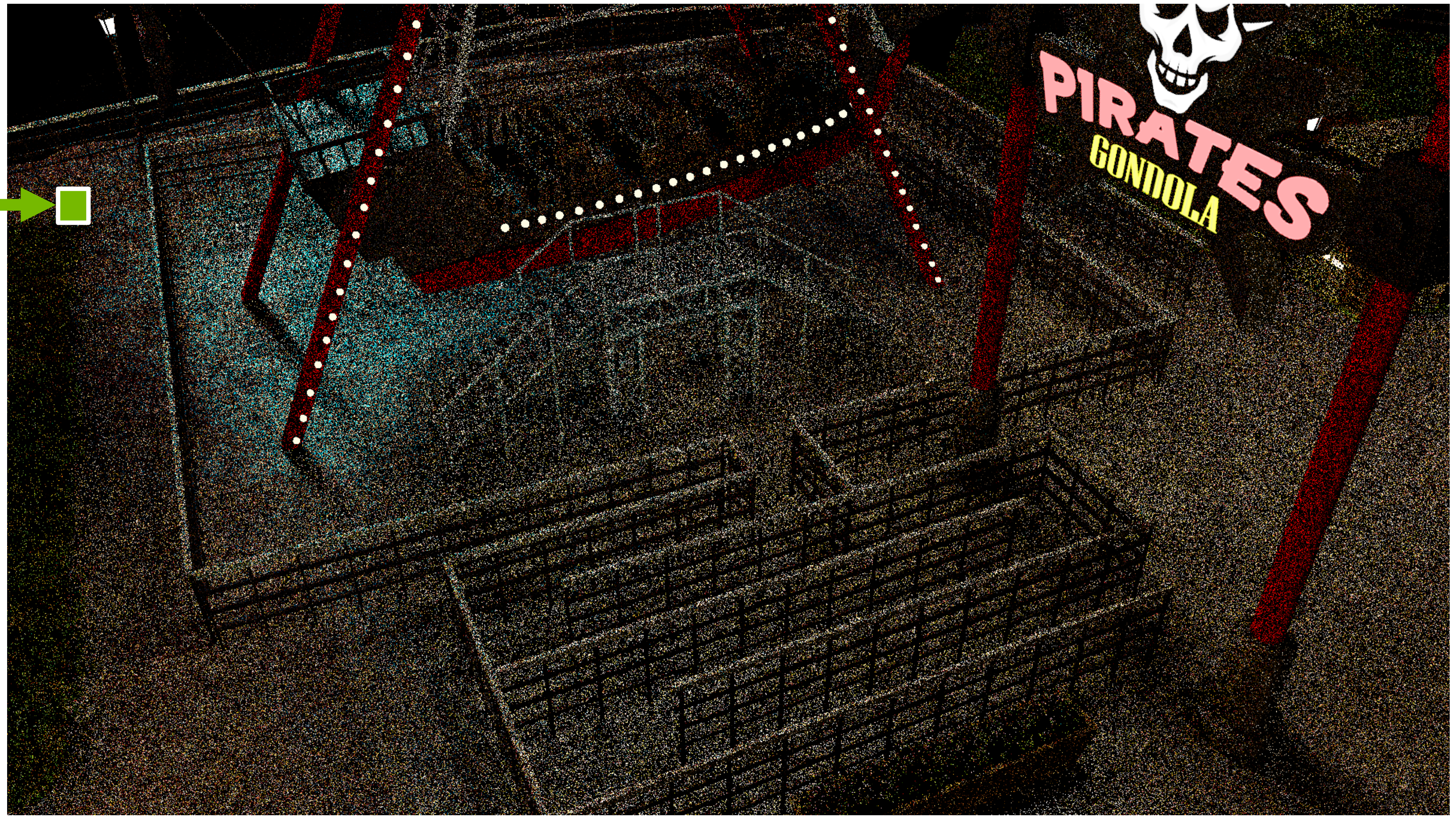
(Surprisingly after spatiotemporal reuse, this still looks decent; but spatial blocks are visible)

CAN WE MAKE LOOKUPS (MORE) COHERENT?

Ensure light samples for adjacent pixels are in same cache line?

~~Use same 32 candidate lights for all pixels in a screen tile?~~

Pick 16k lights per tile; each pixel picks 32 of those



(But this just moves incoherency from per-pixel to per-block sampling)

$$\approx \left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(\omega_i)}{\hat{p}(\omega_i)}\right)\left(\frac{1}{M}\sum_{j=1}^{M}\frac{\hat{p}(\omega_j)}{q(\omega_j)}\right)$$

I.e., M=32

Pick different 32 samples from 16k lights for each tile

CAN WE MAKE LOOKUPS (MORE) COHERENT?

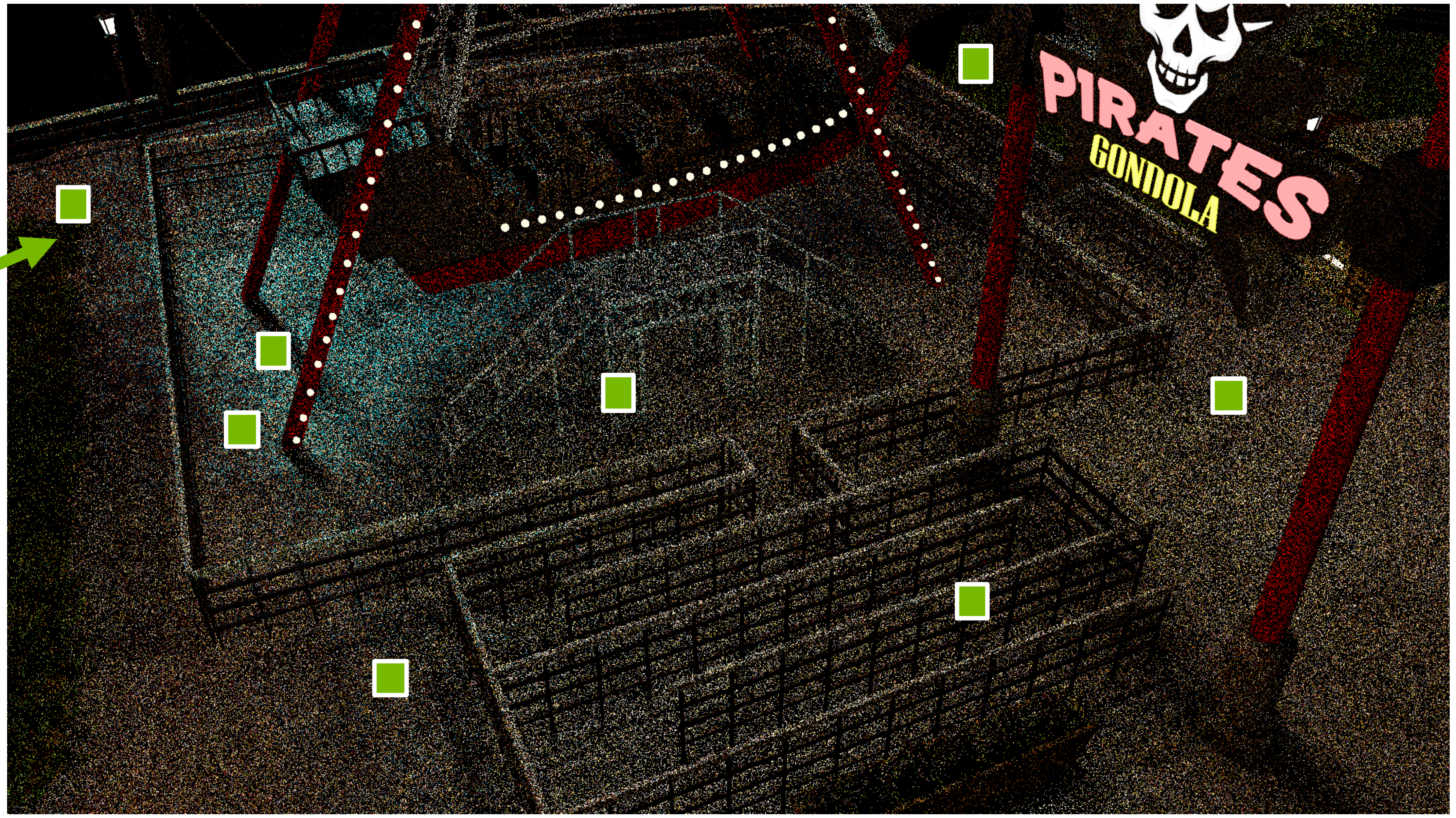
Ensure light samples for adjacent pixels are in same cache line?

~~Use same 32 candidate lights for all pixels in a screen tile?~~

~~Pick 16k lights per tile; each pixel picks 32 of those~~

Reuse a few blocks of lights over entire image? Say 256?

(Each block with 16k lights; a pixel picks 32 of them)



(Removes incoherency from inner loop; reduces number of incoherent lookups)

$$\approx \left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(\omega_i)}{\hat{p}(\omega_i)}\right)\left(\frac{1}{M}\sum_{j=1}^{M}\frac{\hat{p}(\omega_j)}{q(\omega_j)}\right)$$

I.e., M=32

Pixels pick 32 from 16k lights; same 16k used for multiple tiles

CAN WE MAKE LOOKUPS (MORE) COHERENT?

Ensure light samples for adjacent pixels are in same cache line?

~~Use same 32 candidate lights for all pixels in a screen tile?~~

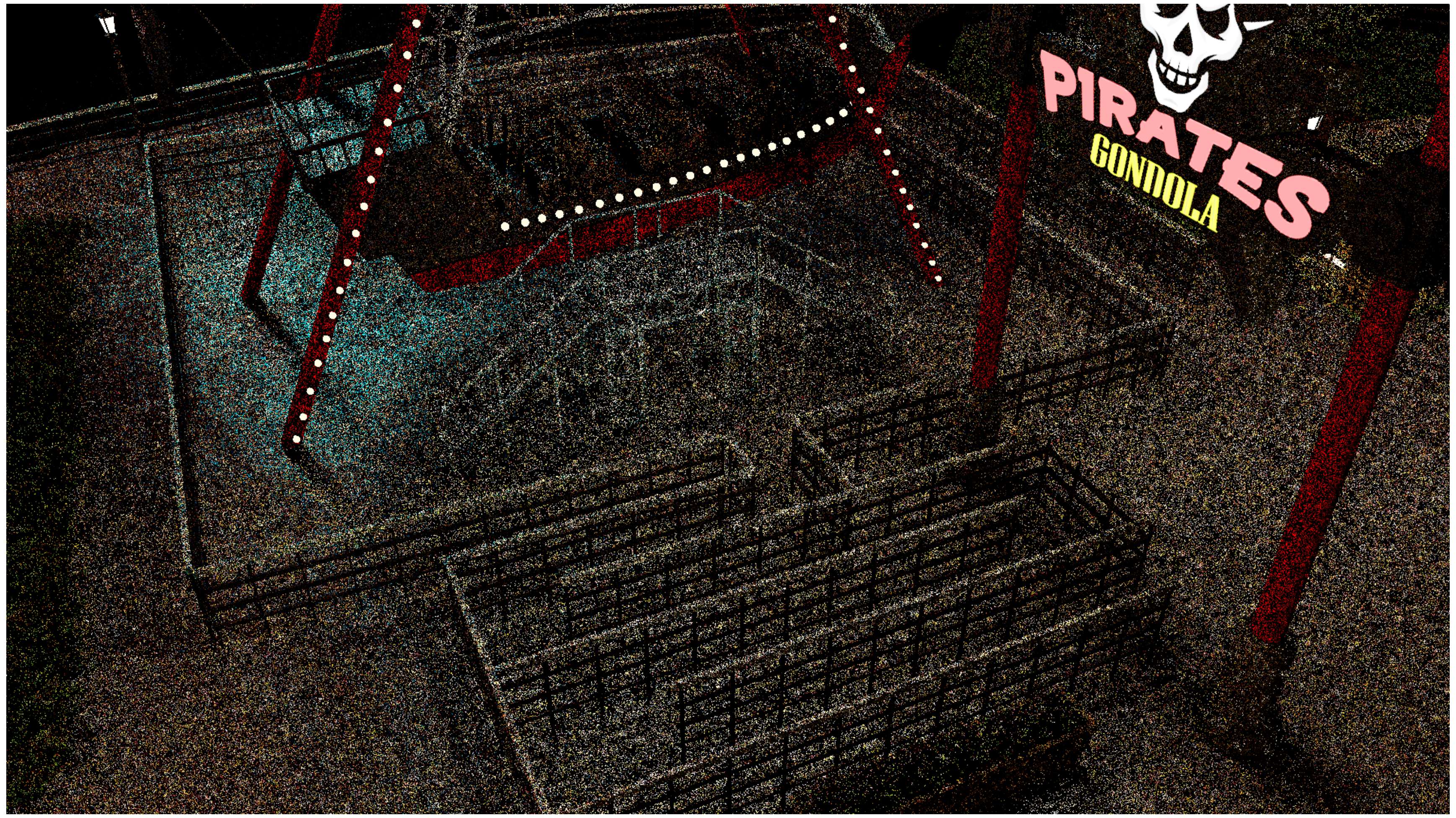
~~Pick 16k lights per tile; each pixel picks 32 of those~~

~~Reuse a few blocks of lights over entire image? Say 256?~~

~~(Each block with 16k lights; a pixel picks 32 of them)~~

Optimizing params: 128 blocks of 1024 light samples each

Reused over 8x8 pixel tile; each pixel selects 32 lights



(Perf sweet spot across scenes; slight correlation noticeable in some scenes)

CAN WE MAKE LOOKUPS (MORE) COHERENT?

Ensure light samples for adjacent pixels are in same cache line?

~~Use same 32 candidate lights for all pixels in a screen tile?~~

~~Pick 16k lights per tile; each pixel picks 32 of those~~

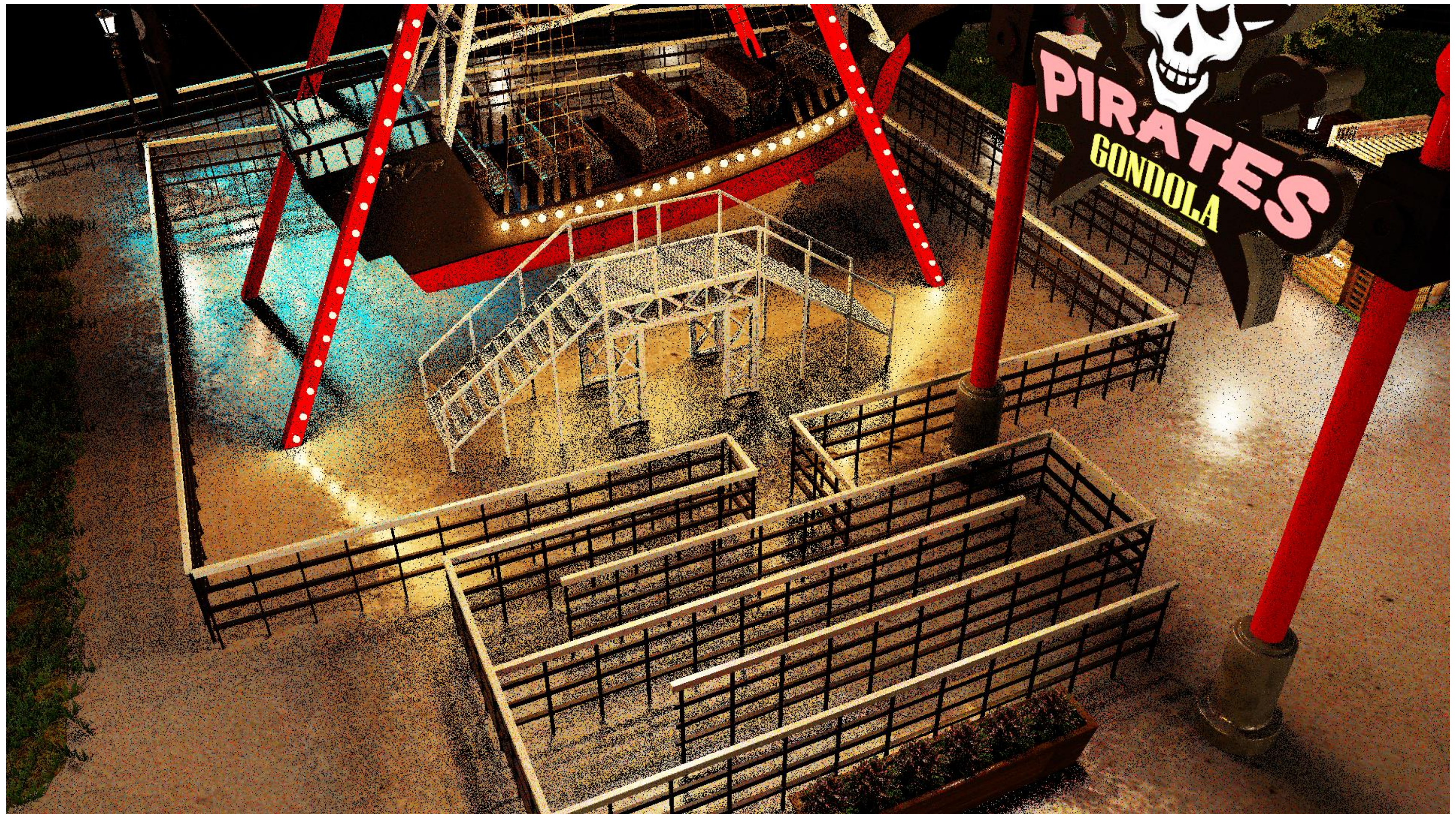
~~Reuse a few blocks of lights over entire image? Say 256?~~

~~(Each block with 16k lights; a pixel picks 32 of them)~~

Optimizing params: 128 blocks of 1024 light samples each

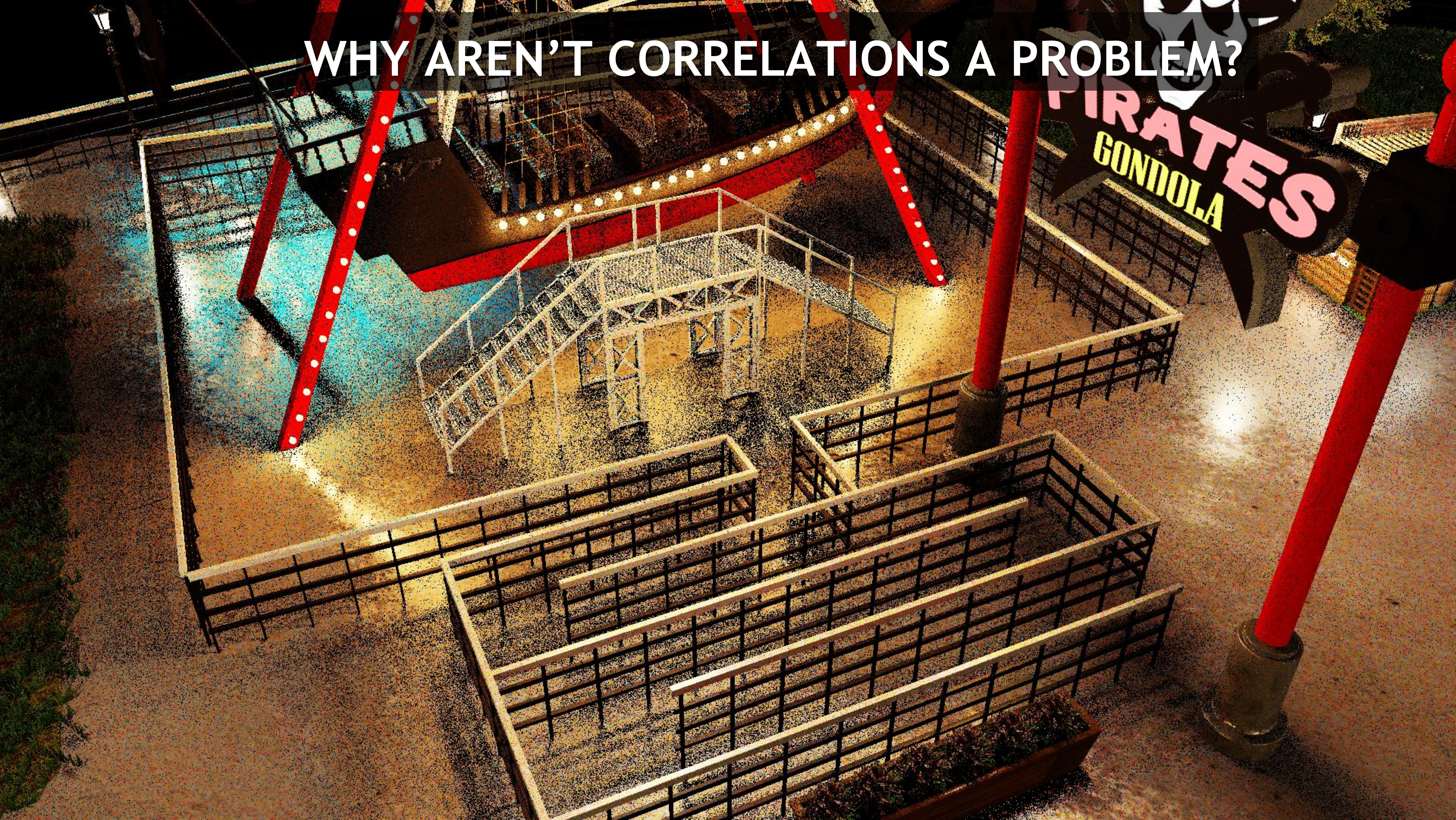
Reused over 8x8 pixel tile; each pixel selects 32 lights

Applying spatiotemporal resampling hides minor correlations

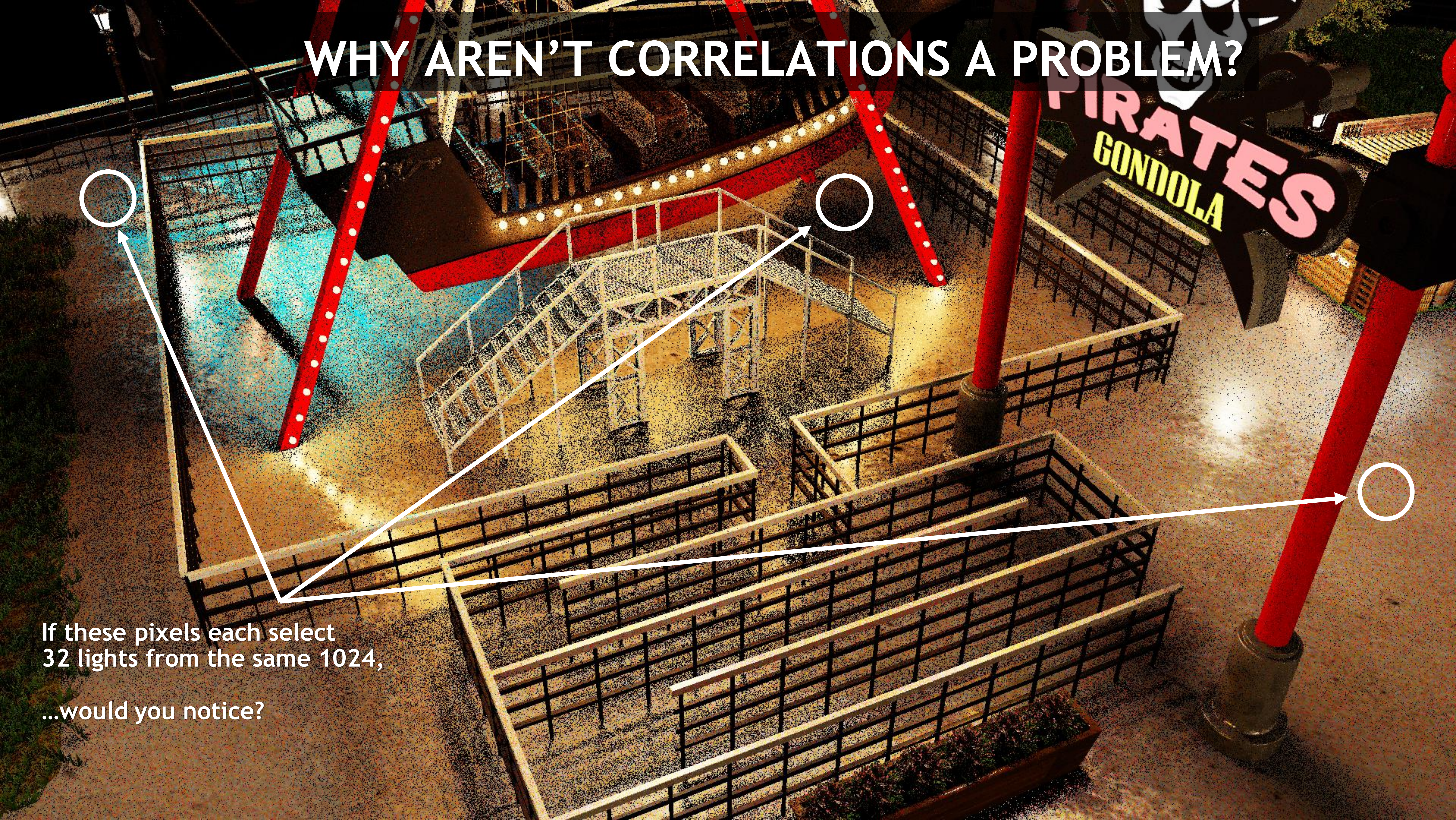


Sampling quality indistinguishable; 21x faster candidate sampling here
18.1 ms → 0.84 ms

WHY AREN’T CORRELATIONS A PROBLEM?



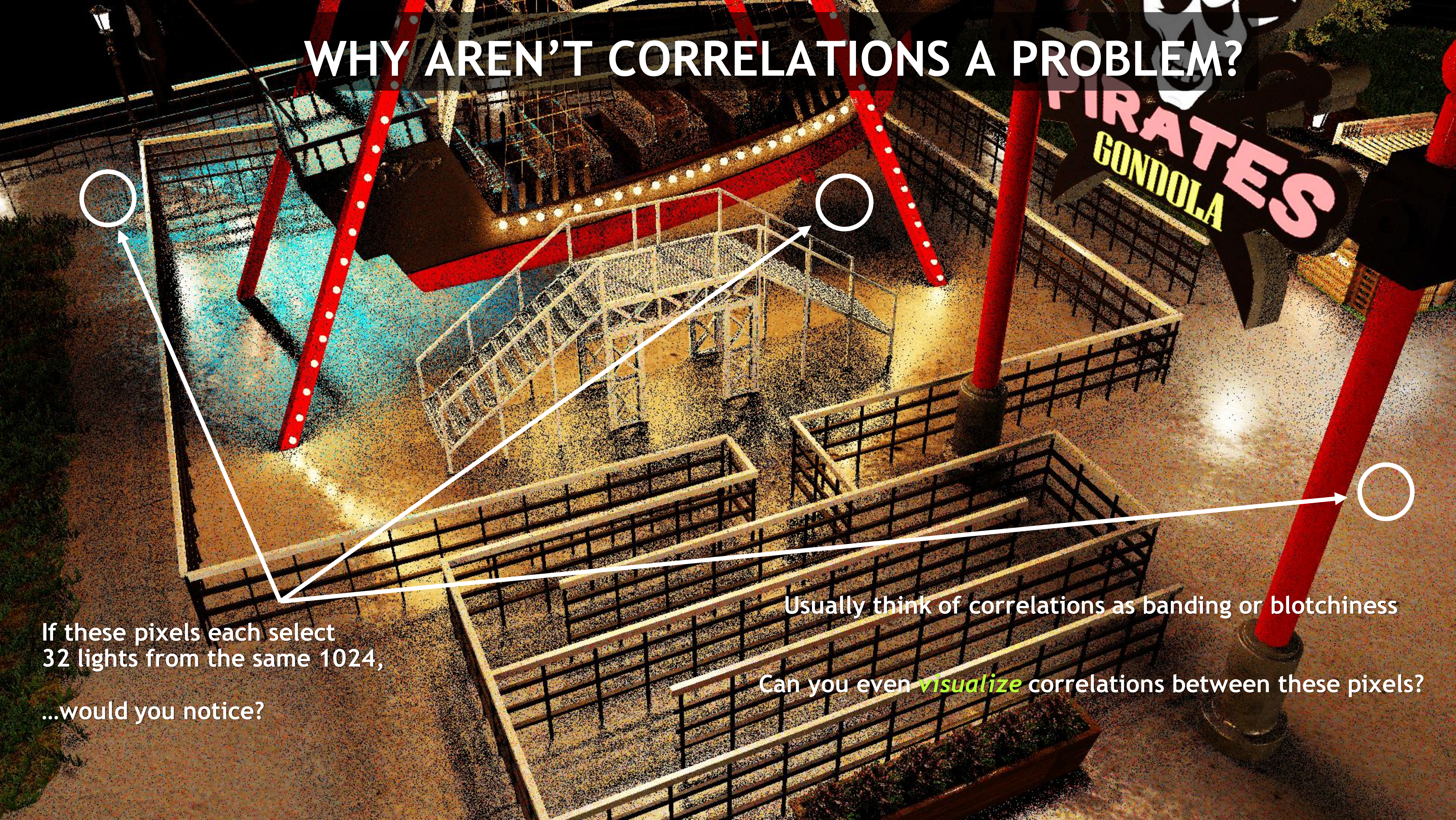
WHY AREN'T CORRELATIONS A PROBLEM?



If these pixels each select
32 lights from the same 1024,

…would you notice?

WHY AREN'T CORRELATIONS A PROBLEM?



If these pixels each select
32 lights from the same 1024,

…would you notice?

Usually think of correlations as banding or blotchiness

Can you even *visualize* correlations between these pixels?

REDUCING RAY TRACING COSTS

CAREFUL SHADOW RAY SELECTION

Tracing rays has significant cost; up to 60% of original ReSTIR costs

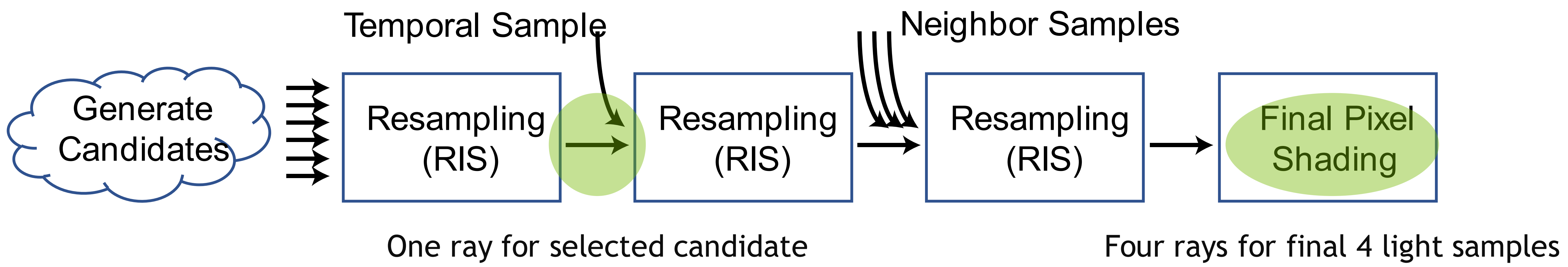
CAREFUL SHADOW RAY SELECTION

Tracing rays has significant cost; up to 60% of original ReSTIR costs

Bitterli et al. [2020] traced 5 rays per pixel

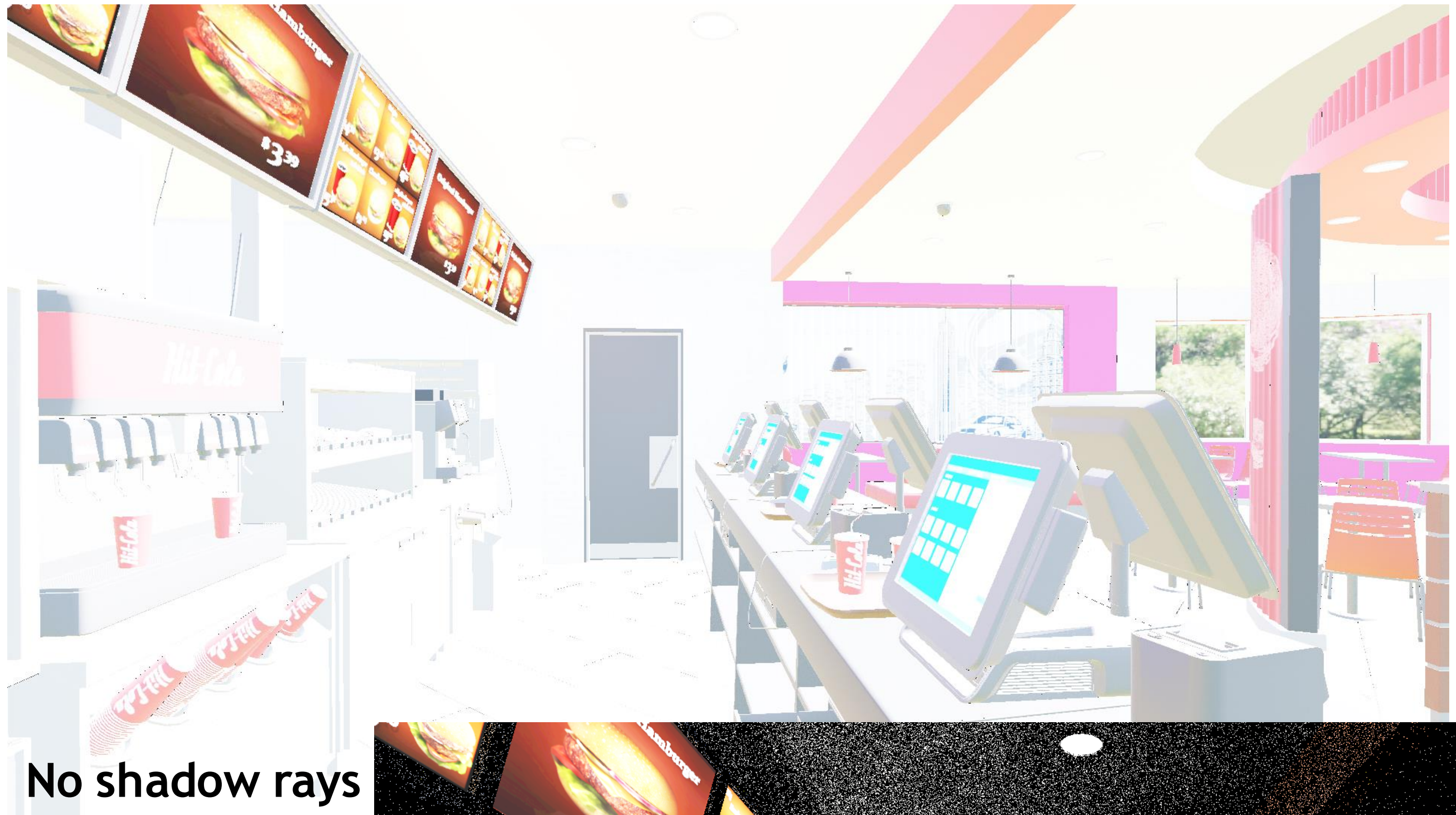
Some modern games use ¼ rays per pixel; so up to 20x current budgets!

Why 5 rays? How important are these?

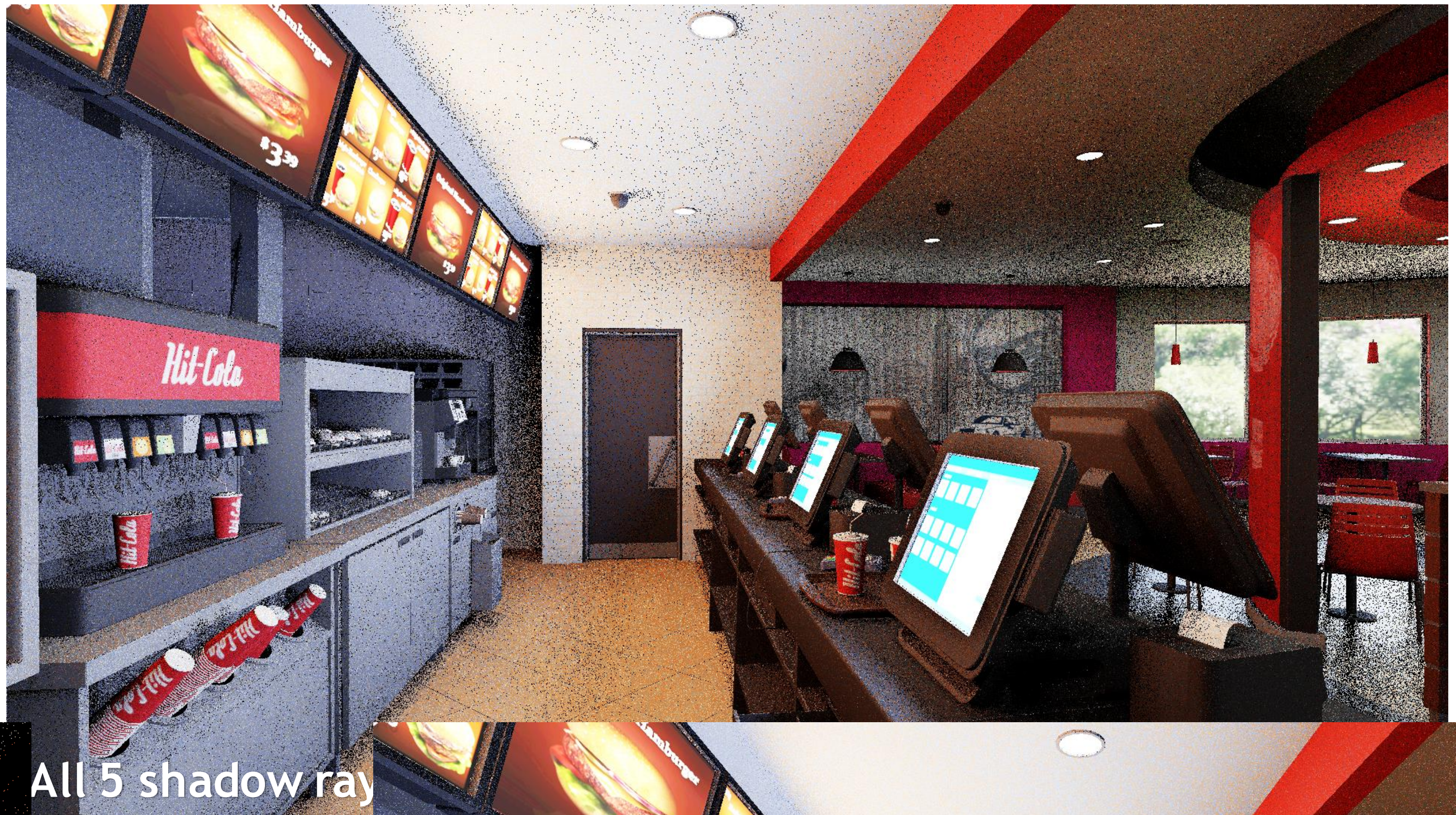


WHICH RAYS ARE IMPORTANT?

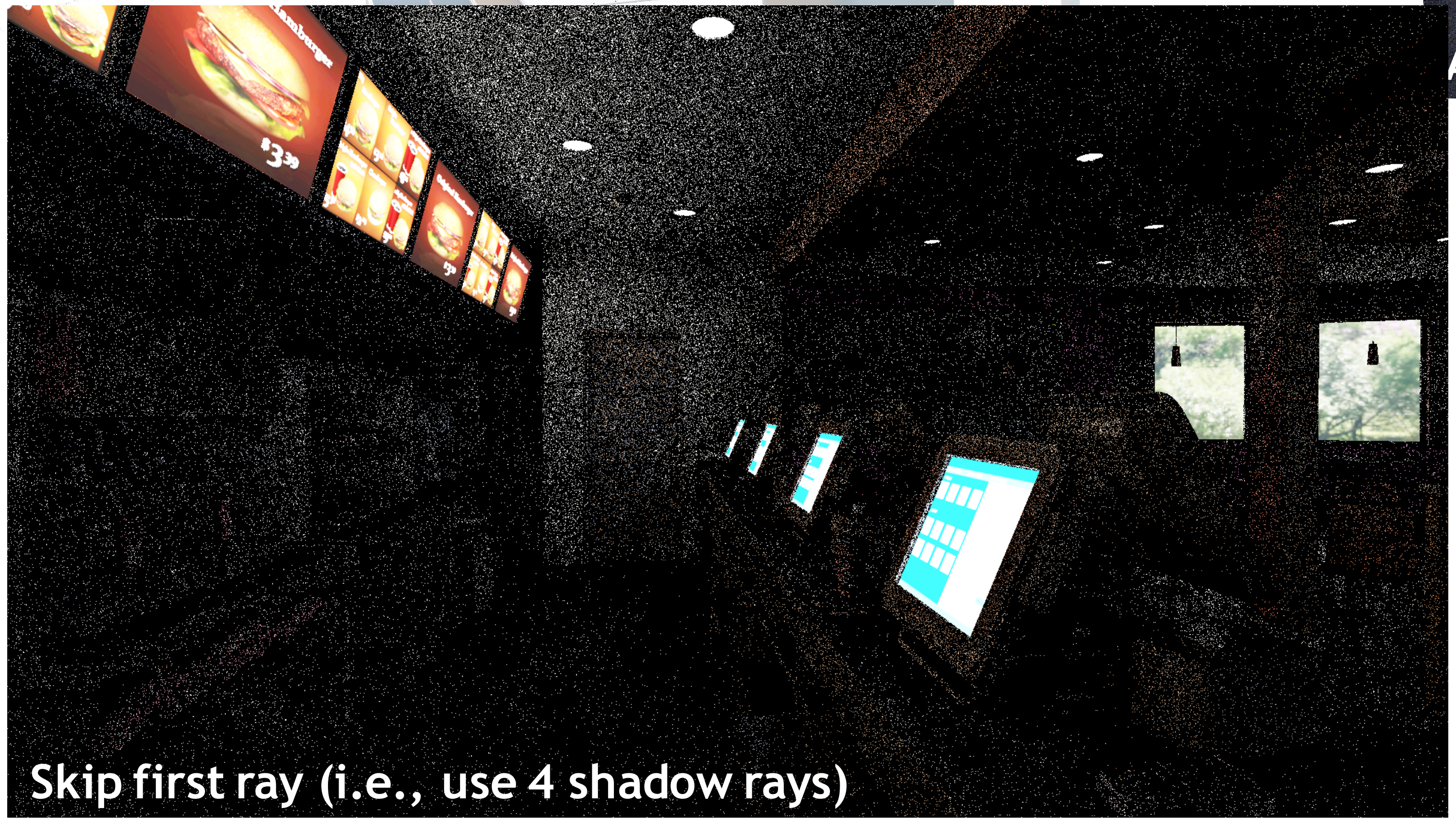
WHICH RAYS ARE IMPORTANT?



No shadow rays



All 5 shadow ray

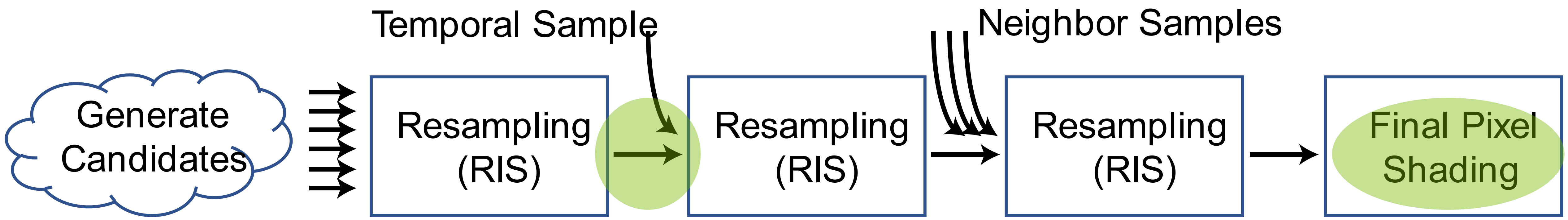


Skip first ray (i.e., use 4 shadow rays)

Use *only* first ray

FIRST RAY IS MOST IMPORTANT

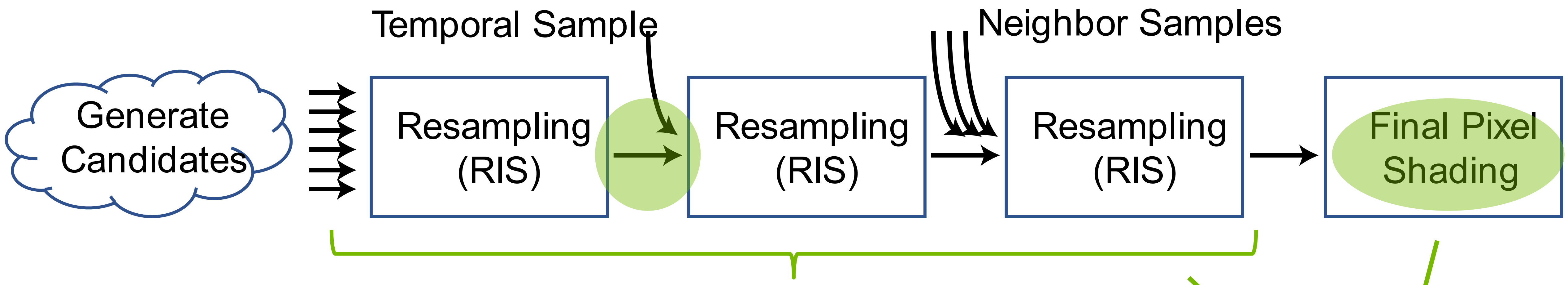
It prevents sample pollution (reusing really bad lights)…



Why allocate 1 important shadow ray and 4 less-important rays?

FIRST RAY IS MOST IMPORTANT

It prevents sample pollution (reusing really bad lights)…



Why allocate 1 important shadow ray and 4 less-important rays?

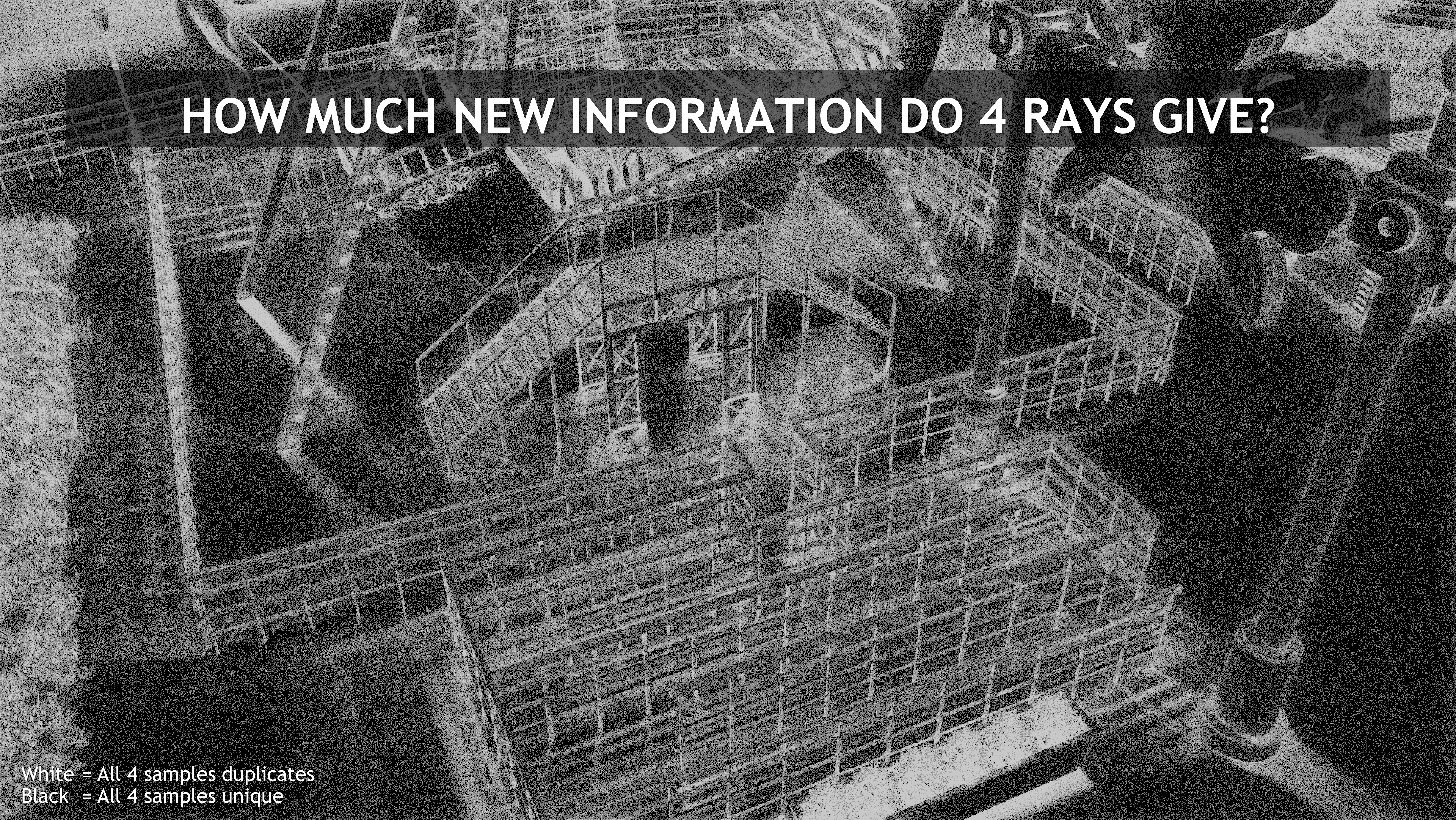
Idea: ReSTIR generally does not reduce visibility noise

So, need extra samples to reduce variance in final visibility

$$F \approx \frac{1}{N} \sum_{i=1}^{N} \left[\frac{f(\omega_i)}{\hat{p}(\omega_i)} \frac{1}{M} \sum_{j=1}^{M} \frac{\hat{p}(\omega_{ij})}{q(\omega_{ij})} \right]$$

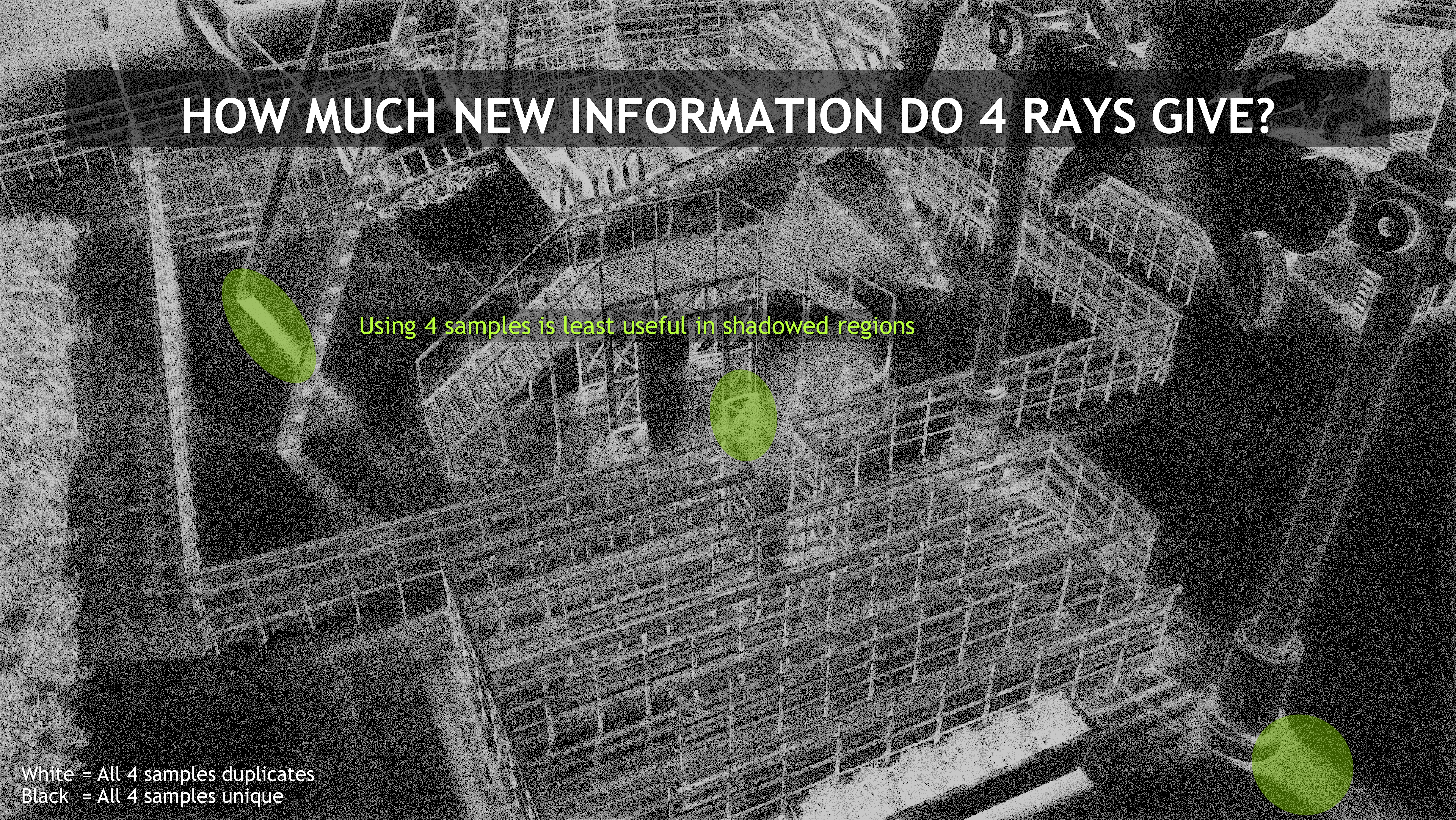
HOW MUCH NEW INFORMATION DO 4 RAYS GIVE?

HOW MUCH NEW INFORMATION DO 4 RAYS GIVE?



White = All 4 samples duplicates
Black = All 4 samples unique

HOW MUCH NEW INFORMATION DO 4 RAYS GIVE?



HOW MUCH NEW INFORMATION DO 4 RAYS GIVE?

Having 4 unique samples isn't useful here; ReSTIR does fine where visibility is stable

White = All 4 samples duplicates
Black = All 4 samples unique

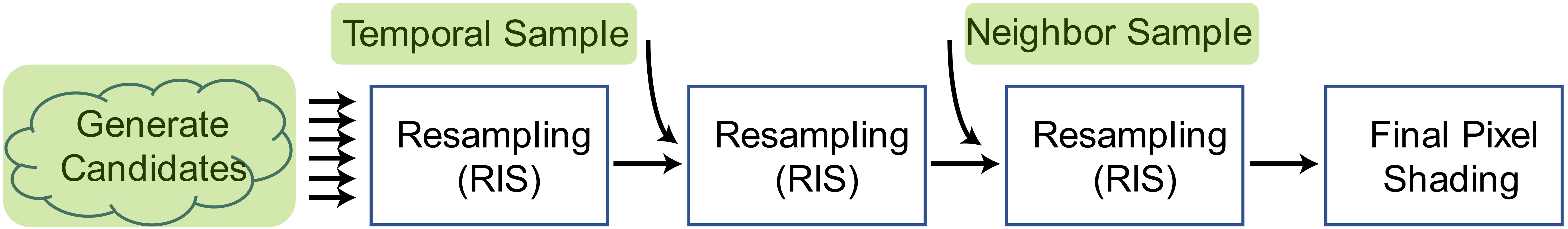
TAKE AWAY: NO MORE THAN 2 RAYS NEEDED!

Extra rays wasted most of the time

DECOUPLING SHADING AND REUSE

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR

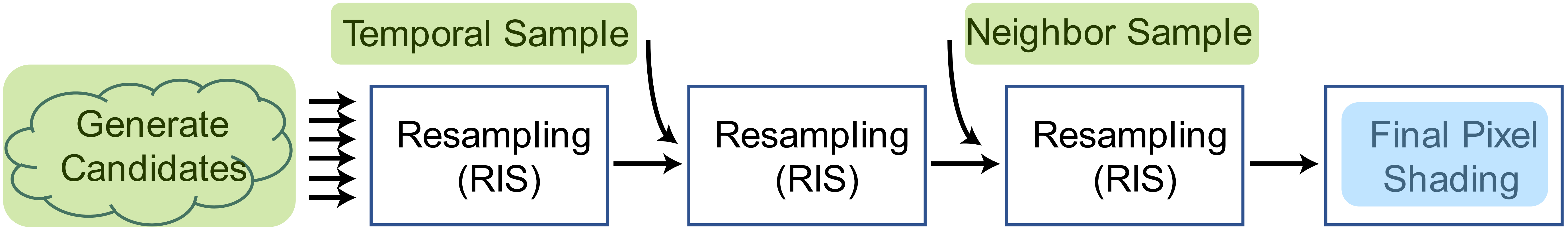


Each frame, we touch M+2 light samples

- M initial candidates (32 in our paper)
- 1 reused temporal light
- 1 reused spatial light

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



Each frame, we touch M+2 light samples

M initial candidates (32 in our paper)

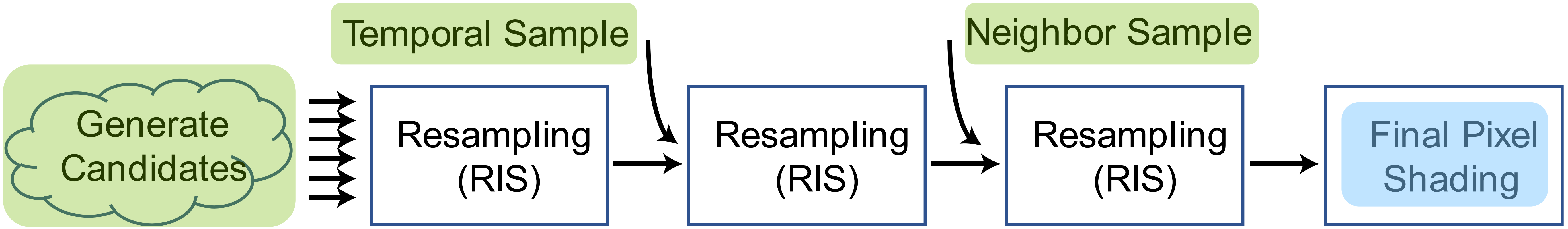
1 reused temporal light

1 reused spatial light

But we only shade one!

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



Each frame, we touch M+2 light samples

M initial candidates (32 in our paper)

1 reused temporal light

1 reused spatial light

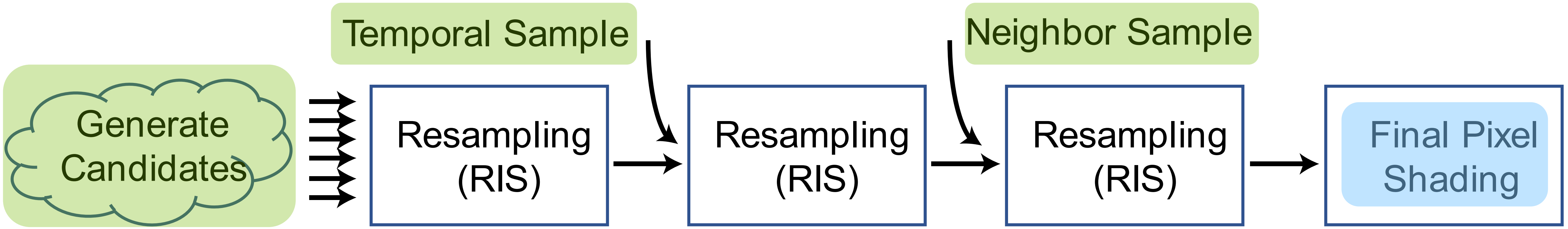
But we only shade one!

Do all computations for others

(Except shadow queries)

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some "wasted" work going underused in ReSTIR



Each frame, we touch M+2 light samples

- M initial candidates (32 in our paper)
- 1 reused temporal light
- 1 reused spatial light

But we only shade one!

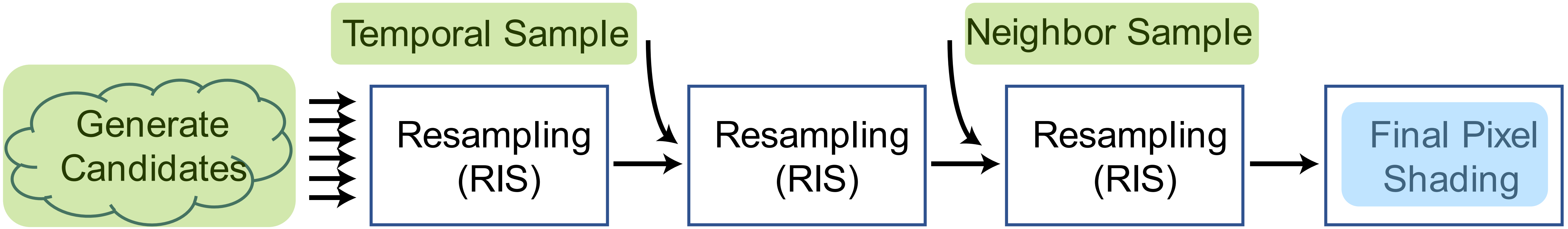
- Do all computations for others
- (Except shadow queries)

Goals of shading & reuse different

- Shading = best quality this frame
- Reuse = best quality in future

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some "wasted" work going underused in ReSTIR



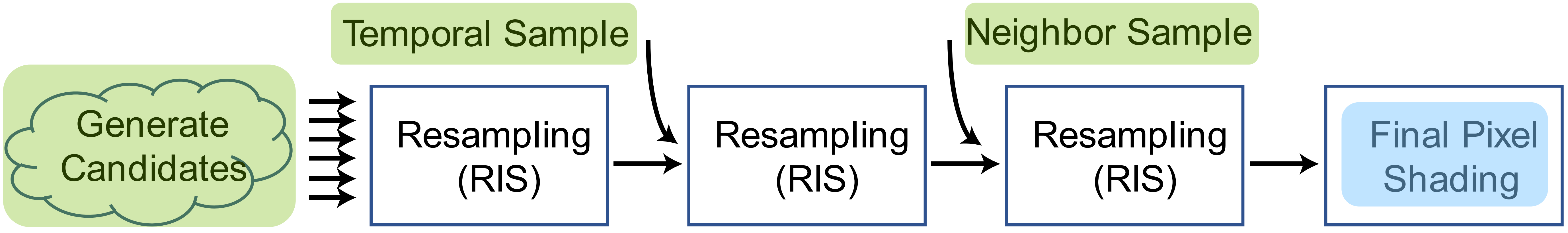
Each frame, we touch M+2 light samples

- M initial candidates (32 in our paper)
- 1 reused temporal light
- 1 reused spatial light

Why not use all M+2 to shade?

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



Each frame, we touch M+2 light samples

M initial candidates (32 in our paper)

1 reused temporal light

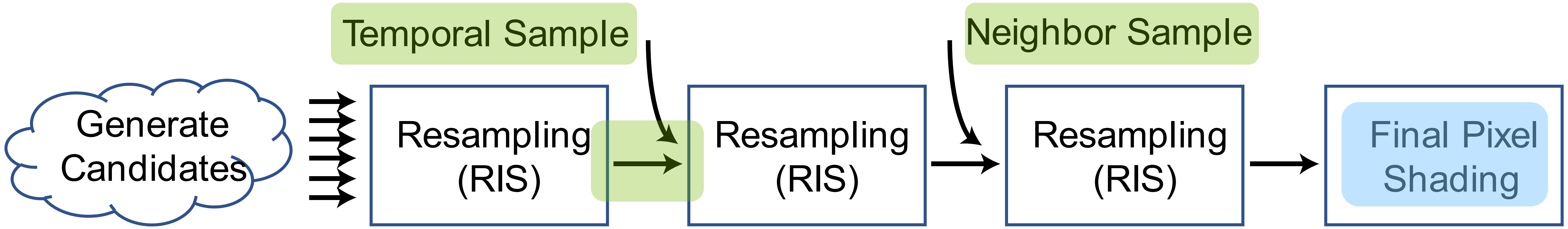
1 reused spatial light

Why not use all M+2 to shade?

… too many shadow rays!

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR

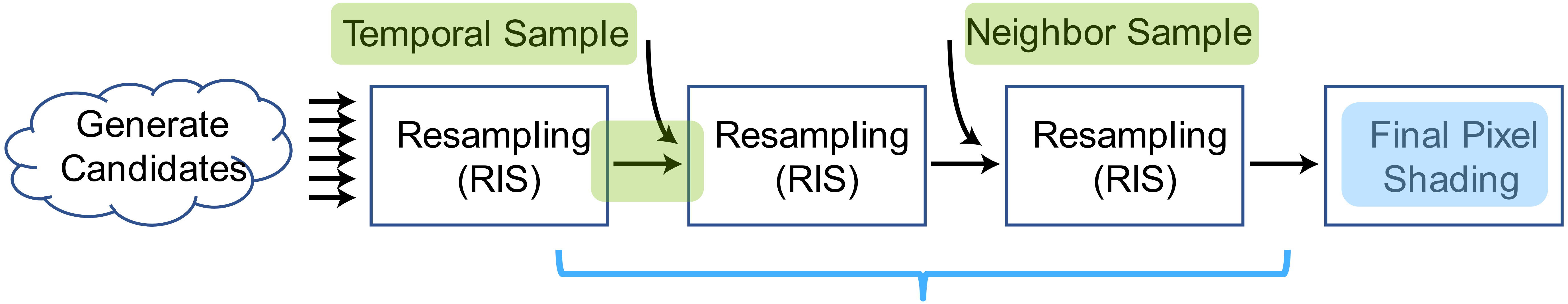


What if we ignore the M candidates?

Realize first RIS pass selects just one

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



What if we ignore the M candidates?

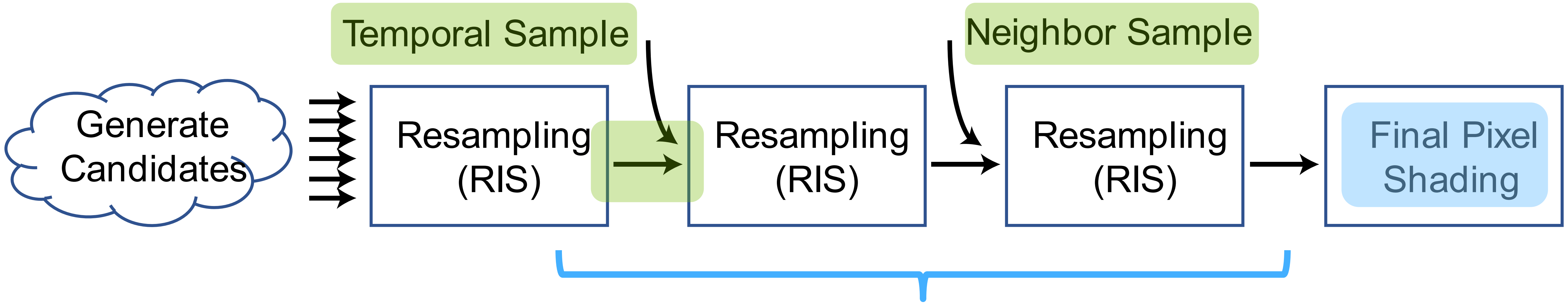
Realize first RIS pass selects just one

Three light samples; one selected for shading

Why not shade all three?

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



What if we ignore the M candidates?

Realize first RIS pass selects just one

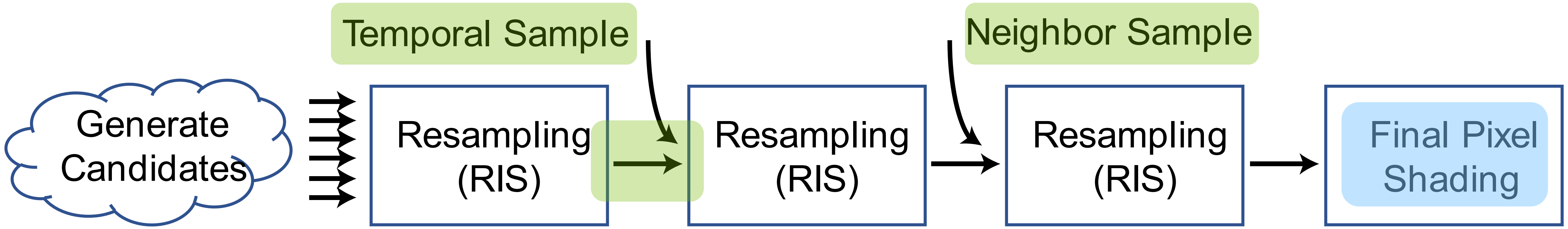
Three light samples; one selected for shading

Why not shade all three?

... requires 3 rays per pixel (instead of 2)
... but also increase image quality

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR

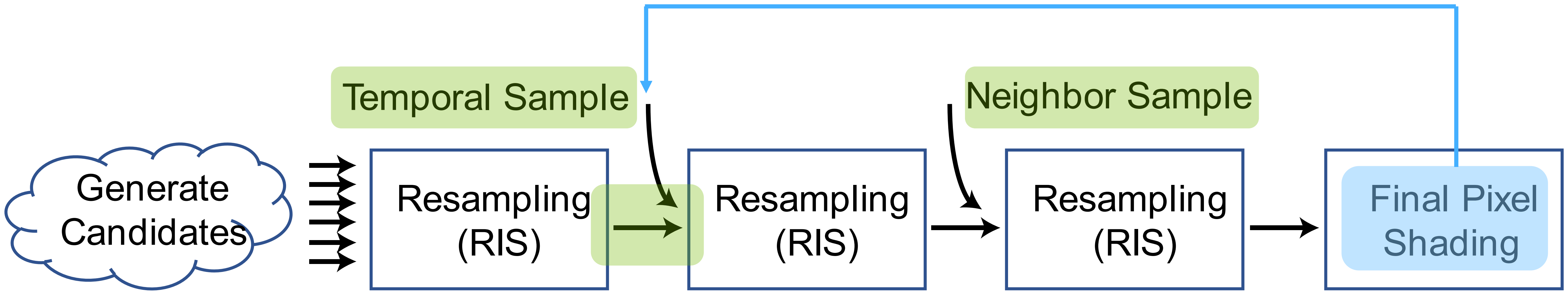


If we do this, consider…

Now have visibility for all 3 samples

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some "wasted" work going underused in ReSTIR



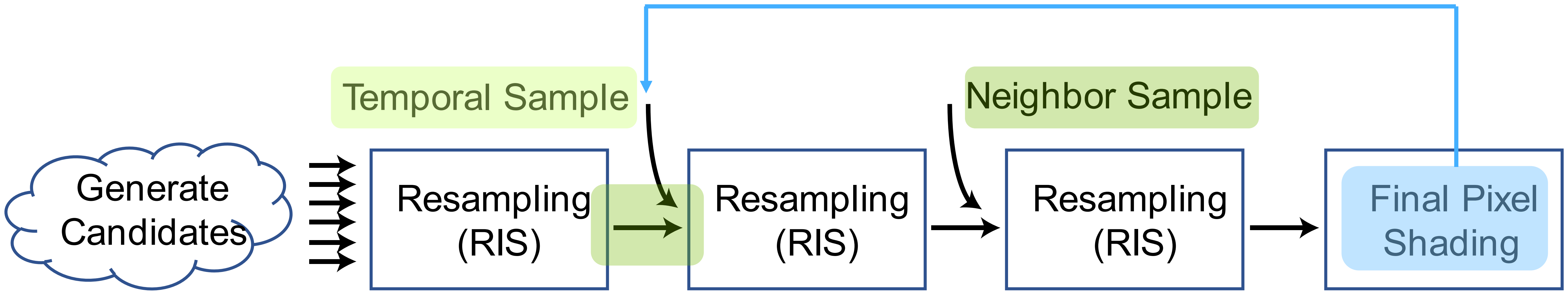
If we do this, consider…

Now have visibility for all 3 samples

Including the one reused in future frames

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



If we do this, consider…

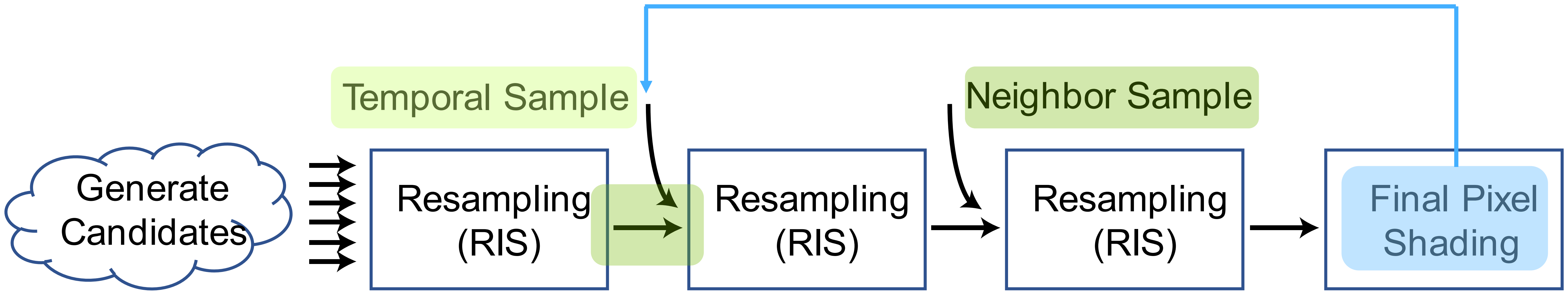
Now have visibility for all 3 samples

Including the one reused in future frames

Reuse this visibility query? Then only 2 rays per pixel, again

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some “wasted” work going underused in ReSTIR



If we do this, consider…

Now have visibility for all 3 samples

Including the one reused in future frames

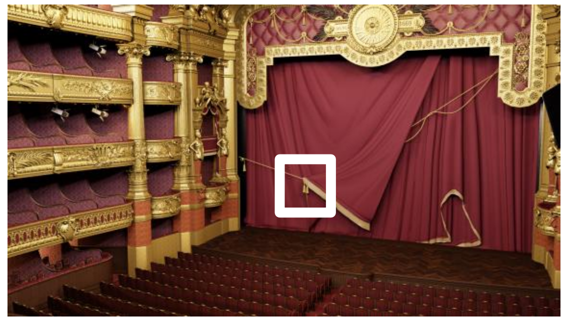
Reuse this visibility query? Then only 2 rays per pixel, again

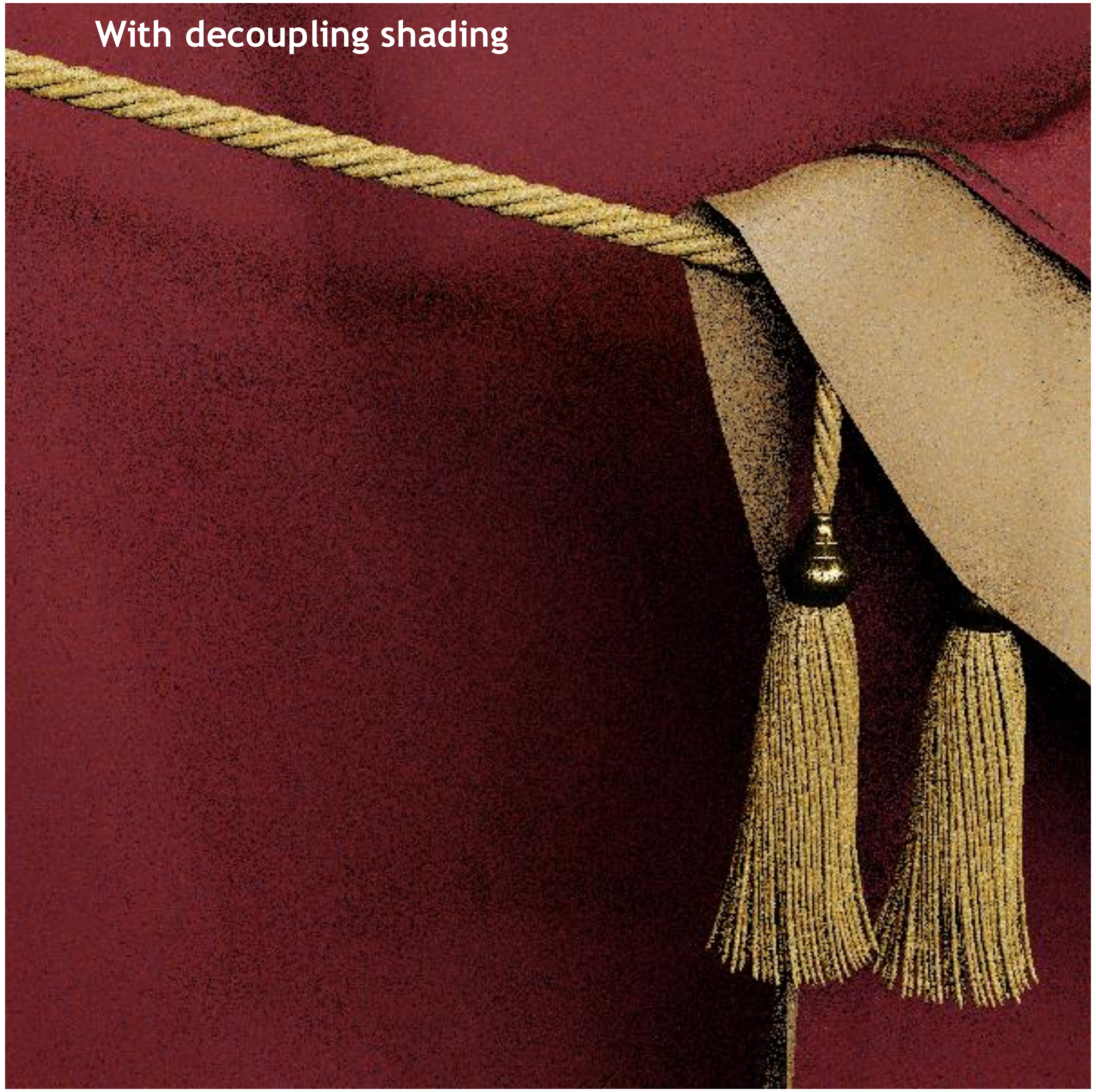
May add some transient temporal bias

But I couldn’t see to capture an example

RESULTS OF DECOUPLING

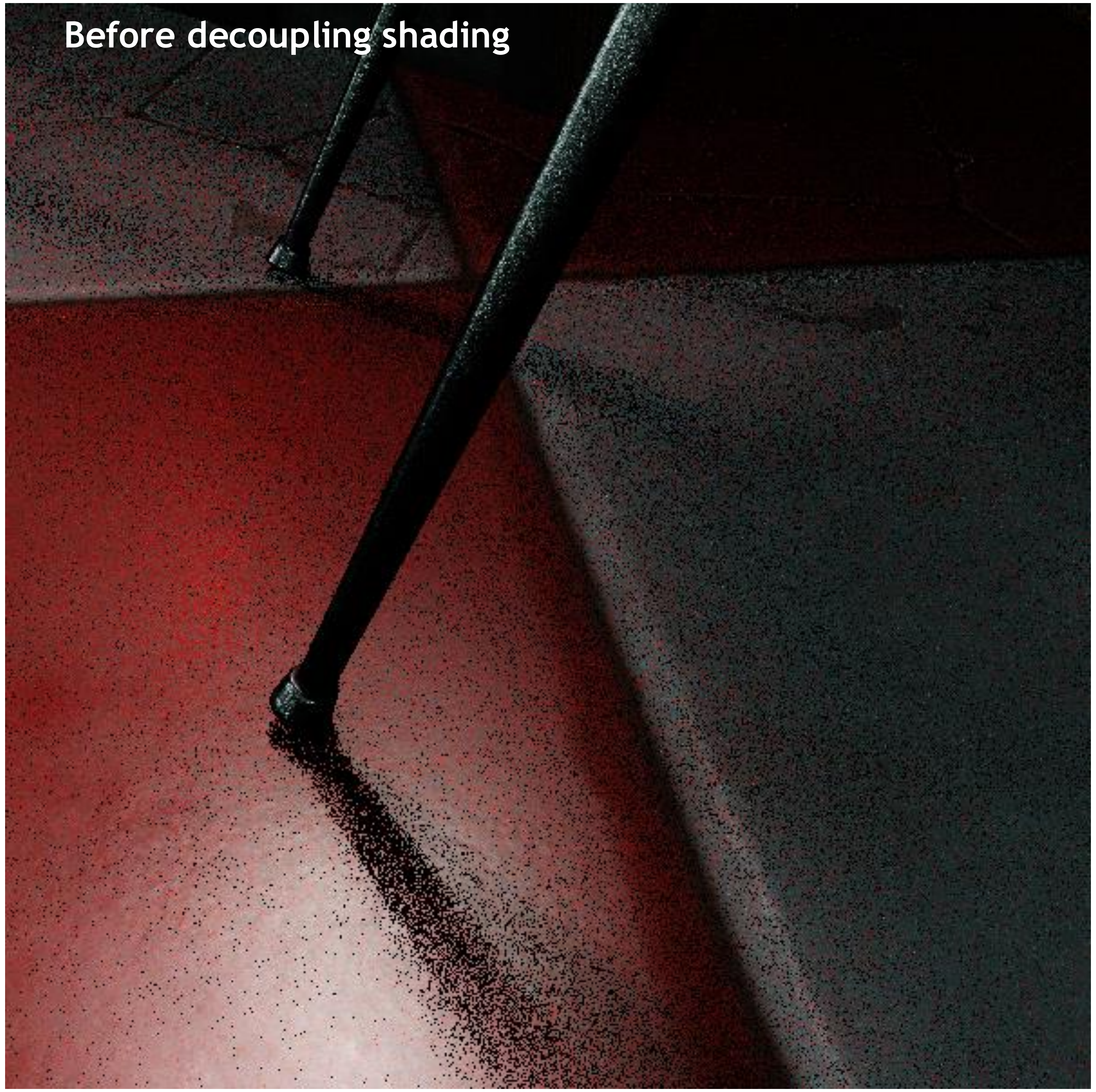
Both the same cost; touching the same memory

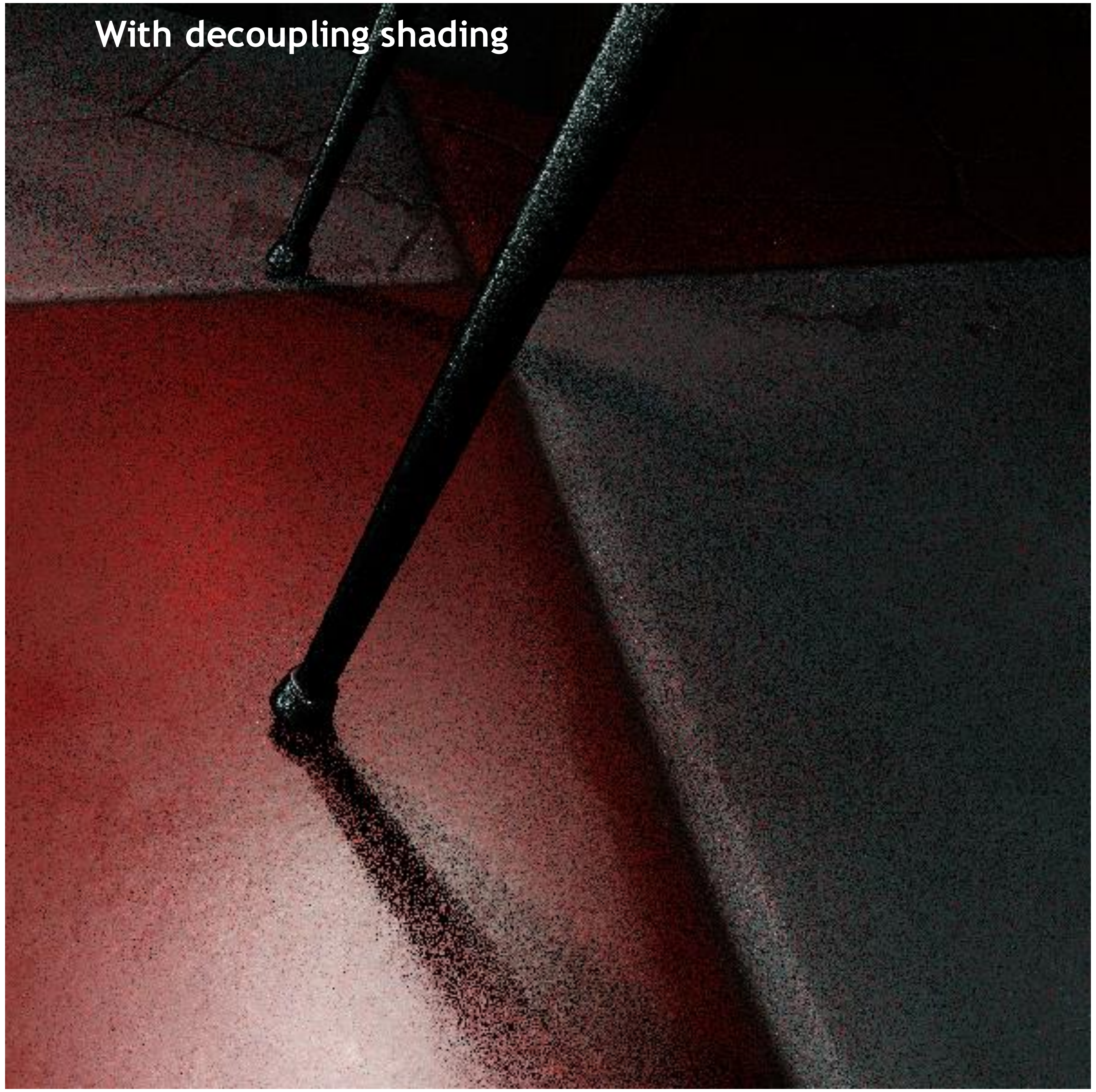




RESULTS OF DECOUPLING

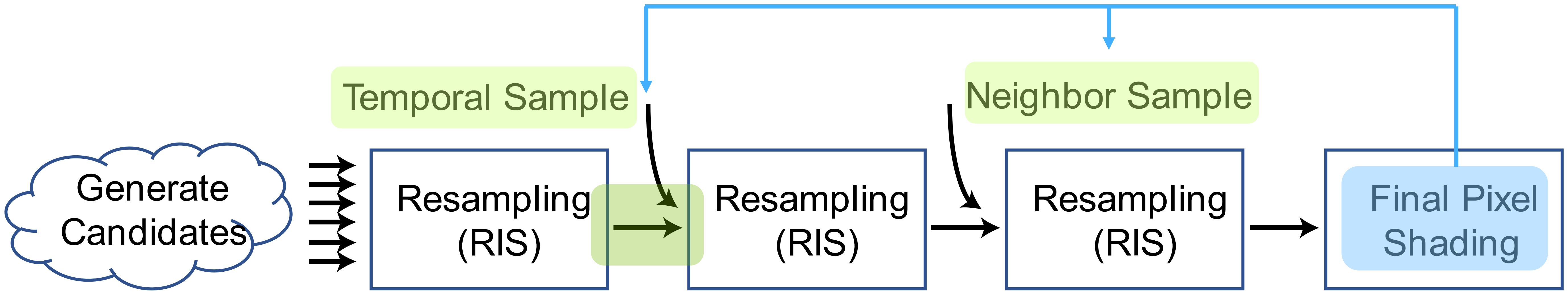
Both the same cost; touching the same memory





DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some "wasted" work going underused in ReSTIR

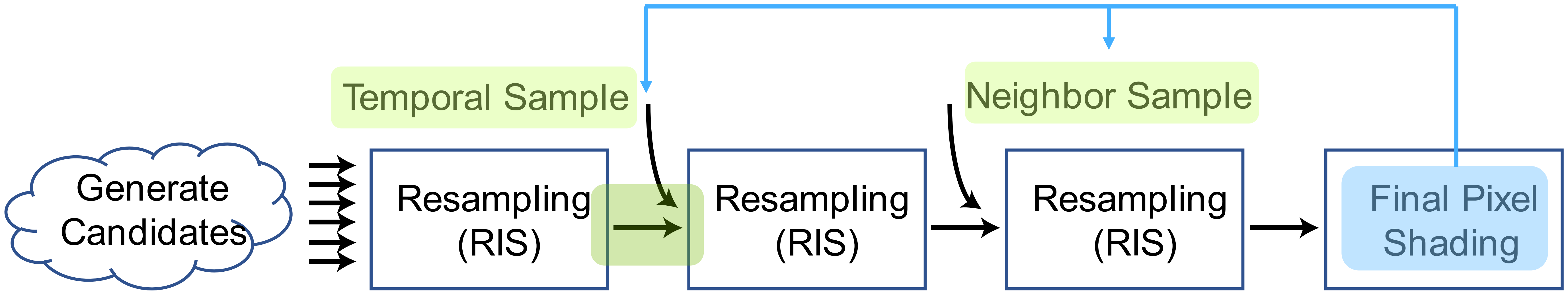


Can also ask:

Can we reuse visibility here? Reduces to 1 ray per pixel

DECOUPLING SHADING AND REUSE

Decreasing parameters → Uncovered some "wasted" work going underused in ReSTIR



Can also ask:

Can we reuse visibility here? Reduces to 1 ray per pixel

This **does** add bias, but a controllable knob

TRADING QUALITY FOR PERFORMANCE

Reusing (some) neighbor shadow rays

2 rays per pixel
2.1 ms (3090)
6.3 ms (2080Q)

(No reuse for spatial samples)

1 ray per pixel
1.6 ms (3090)
5.0 ms (2080Q)

(Always reuse for spatial samples)

TRADING QUALITY FOR PERFORMANCE

Reusing (some) neighbor shadow rays

2 rays per pixel
2.1 ms (3090)
6.3 ms (2080Q)

1.75 rays per pixel
2.0 ms (3090)
6.0 ms (2080Q)

1.5 rays per pixel
1.9 ms (3090)
5.7 ms (2080Q)

1.25 rays per pixel
1.8 ms (3090)
5.4 ms (2080Q)

1 ray per pixel
1.6 ms (3090)
5.0 ms (2080Q)

(No reuse for spatial samples)

Reuse visibility for neighbors within (varying) distances from current pixel

(Always reuse for spatial samples)

PERFORMANCE RESULTS

SOME PERFORMANCE COMPARISONS

On GeForce RTX 3090

	Presampling	Candidates	Trace Rays	Reuse & Shade	Total Lighting
Ours	0.02 ms	0.8 ms	0.5 ms	0.6 ms	1.9 ms

	Candidates	First Ray	Reuse	Shade & Trace	Total Lighting
Bitterli et al.	1.3 ms	0.3 ms	2.9 ms	2.1 ms	8.4 ms

Note: Times do not sum to total, thanks to uncounted synchronization costs

SOME PERFORMANCE COMPARISONS

On GeForce RTX 3090

	Presampling	Candidates	Trace Rays	Reuse & Shade	Total Lighting
Ours	0.02 ms	0.8 ms	0.5 ms	0.6 ms	1.9 ms

	Candidates	First Ray	Reuse	Shade & Trace	Total Lighting
Bitterli et al.	1.3 ms	0.3 ms	2.9 ms	2.1 ms	8.4 ms

Note: Times do not sum to total, thanks to uncounted synchronization costs

4.4x faster

SOME PERFORMANCE COMPARISONS

On GeForce RTX 3090

	Presampling	Candidates	Trace Rays	Reuse & Shade	Total Lighting
Ours	0.04 ms	0.8 ms	3.1 ms	0.6 ms	4.6 ms

	Candidates	First Ray	Reuse	Shade & Trace	Total Lighting
Bitterli et al.	18.3 ms	2.6 ms	3.9 ms	4.3 ms	31.0 ms

Note: Times do not sum to total, thanks to uncounted synchronization costs

SOME PERFORMANCE COMPARISONS

On GeForce RTX 3090

	Presampling	Candidates	Trace Rays	Reuse & Shade	Total Lighting
Ours	0.04 ms	0.8 ms	3.1 ms	0.6 ms	4.6 ms

	Candidates	First Ray	Reuse	Shade & Trace	Total Lighting
Bitterli et al.	18.3 ms	2.6 ms	3.9 ms	4.3 ms	31.0 ms

Note: Times do not sum to total, thanks to uncounted synchronization costs

6.7x faster

SUMMARY

Goal: Improve performance, coherence, quality over previous ReSTIR research

SUMMARY

Goal: Improve performance, coherence, quality over previous ReSTIR research

Specific improvements:

Reshape computations, moving incoherent candidate generation out of inner loop

SUMMARY

Goal: Improve performance, coherence, quality over previous ReSTIR research

Specific improvements:

Reshape computations, moving incoherent candidate generation out of inner loop

Reduce shadow rays traced; many were redundant, duplicate, or irrelevant

SUMMARY

Goal: Improve performance, coherence, quality over previous ReSTIR research

Specific improvements:

- Reshape computations, moving incoherent candidate generation out of inner loop
- Reduce shadow rays traced; many were redundant, duplicate, or irrelevant
- Decouple sample shading and reuse; allows shading more than we reuse

SUMMARY

Goal: Improve performance, coherence, quality over previous ReSTIR research

Specific improvements:

- Reshape computations, moving incoherent candidate generation out of inner loop
- Reduce shadow rays traced; many were redundant, duplicate, or irrelevant
- Decouple sample shading and reuse; allows shading more than we reuse

Other aspects discussed in the paper:

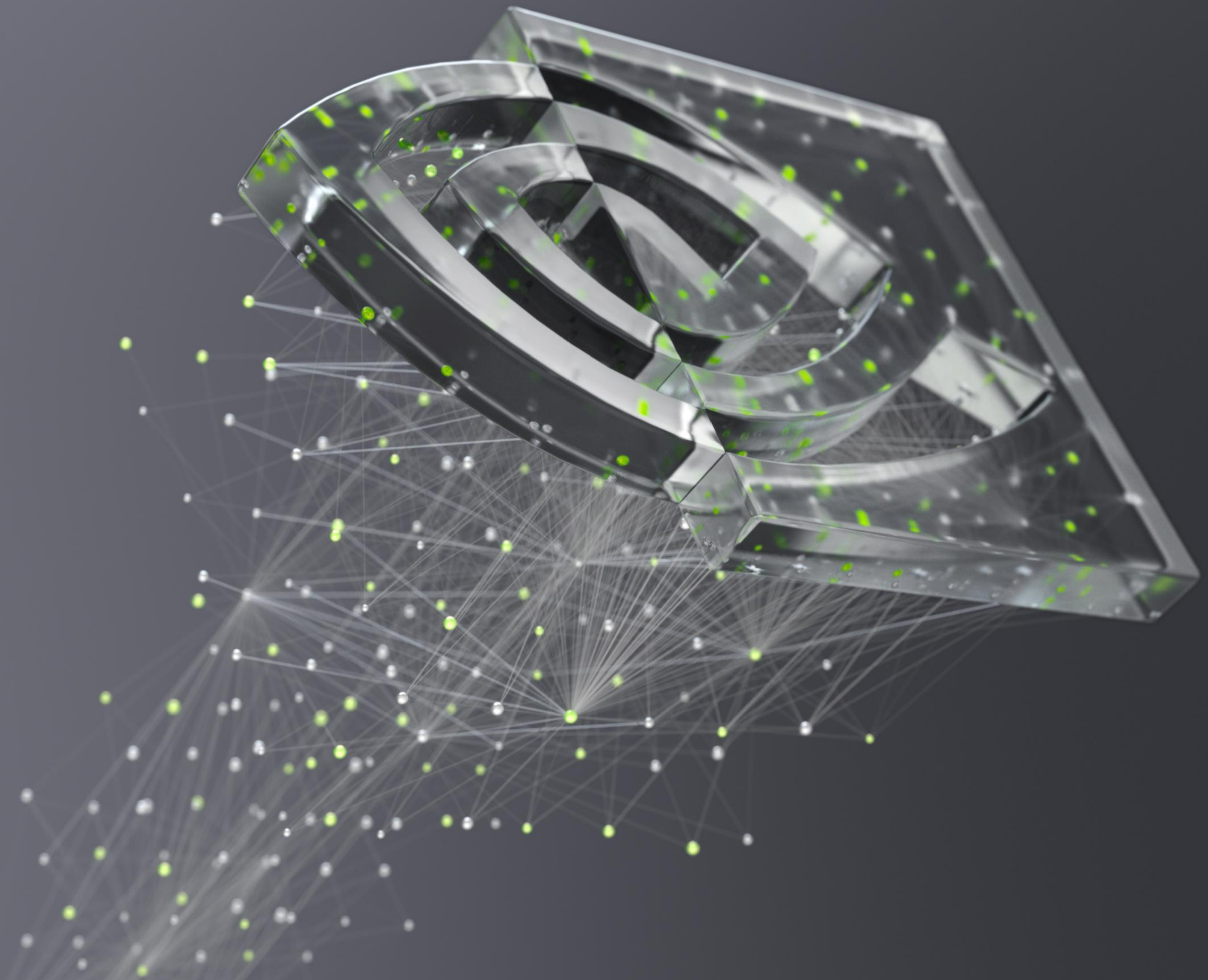
- Reduce spatial samples reused; removed samples had high cost and little quality impact
- Unify heuristic-based techniques and MIS for reducing bias and noise
- Intuitive visual discussion of bias
- Aggregate ReSTIR math in one place

THANKS!

E-mail: cwyman@nvidia.com
Twitter: @_cwyman_

Exemplar code & sample app:
https://developer.nvidia.com/rtxdi

#SaveThePl
44.99$
Boarding Area
Boarding Area
Boarding Area
Litter
Stops Here
CAUTION
WET
FLOOR



NVIDIA