Sample-based Monte Carlo Denoising using a Kernel-Splatting Network

Michaël Gharbi[1,2], Tzu-Mao Li[2], Miika Aittala[2], Jaakko Lehtinen[3,4], Frédo Durand[2]

Adobe

MIT CSAIL

Aalto-yliopisto

NVIDIA

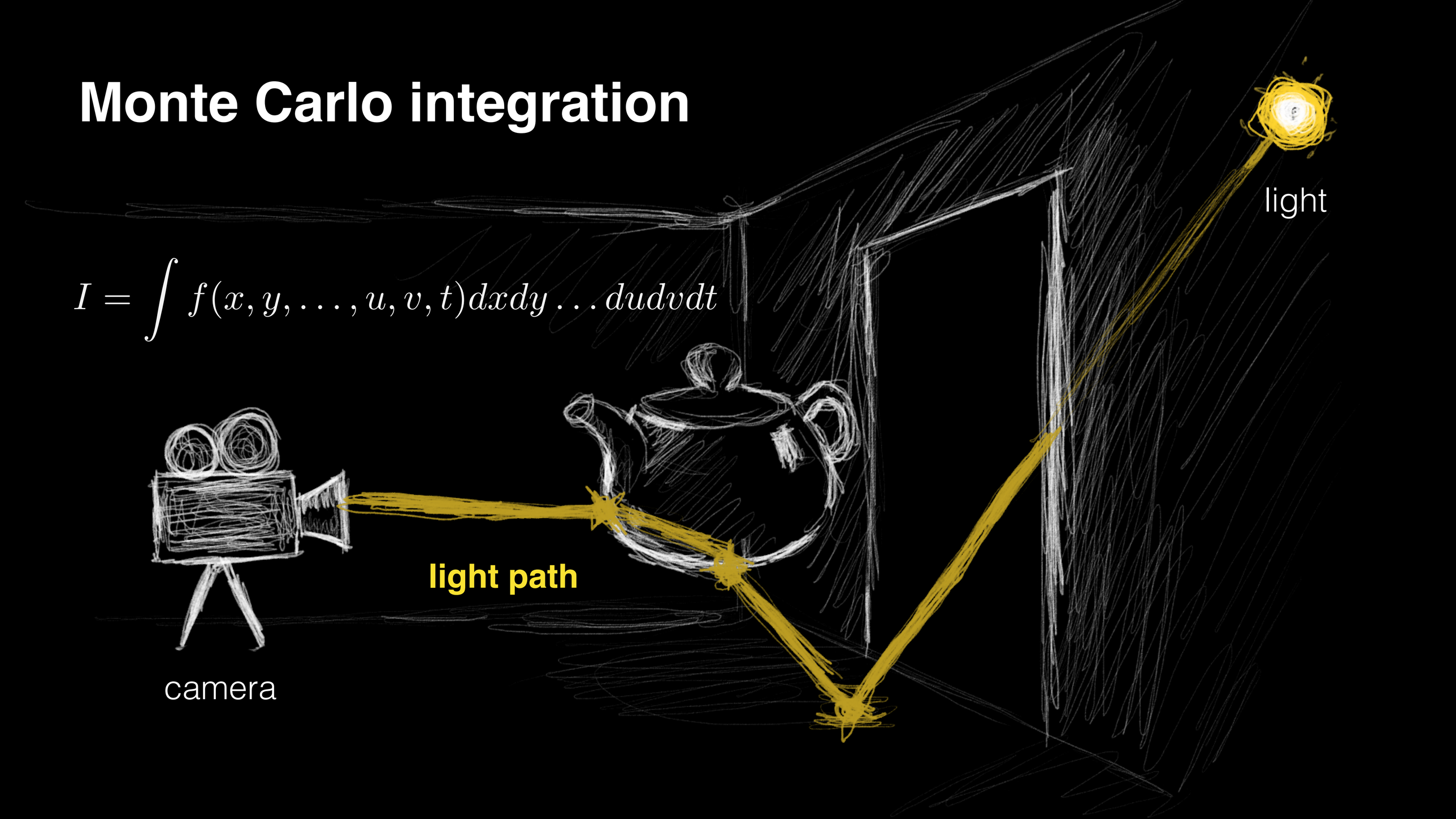
Monte Carlo rendering

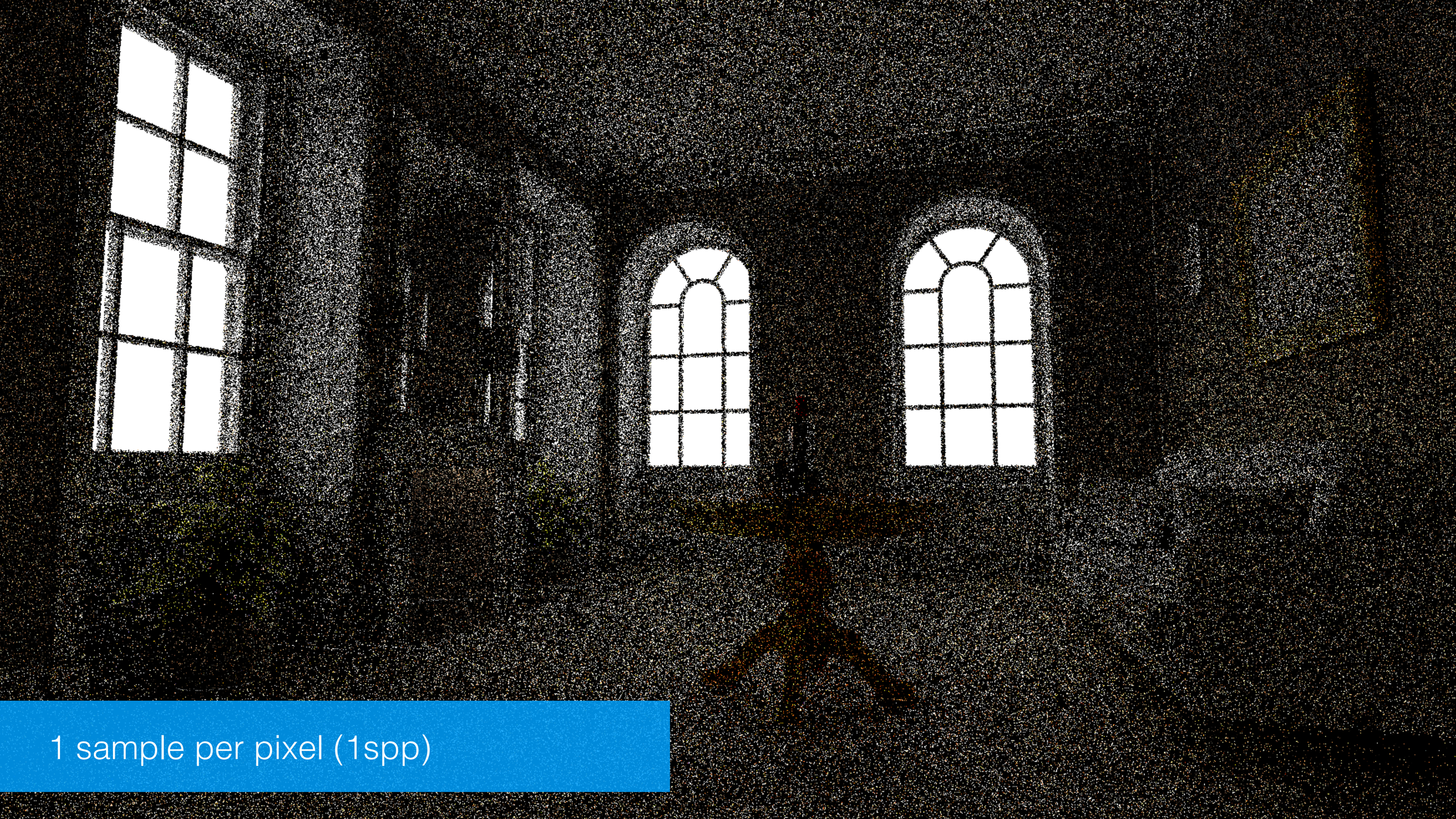
Monte Carlo integration

$$I = \int f(x, y, \ldots, u, v, t) dx dy \ldots du dv dt$$

Monte Carlo integration

$$I = \int f(x, y, \ldots, u, v, t) dx dy \ldots dudvdt$$





1 sample per pixel (1spp)


32spp


our result | 32spp


input | 32spp

Goal: faster rendering, fewer samples

- Produce high-quality renderings

Goal: faster rendering, fewer samples

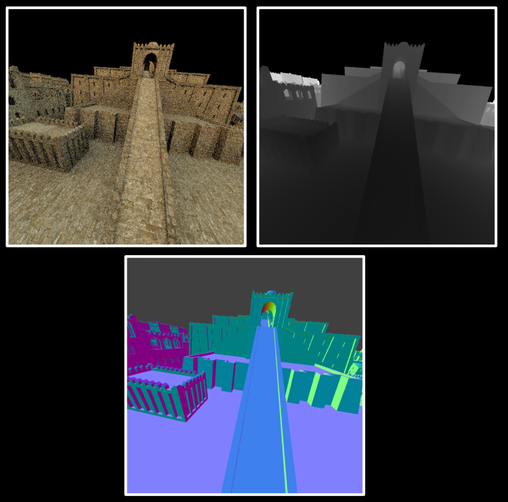
- Produce high-quality renderings
- Filtering and reuse samples
  - ray-tracing is costly
  - variance decreases only linearly with sample count

Goal: faster rendering, fewer samples

- Produce high-quality renderings
- Filtering and reuse samples
  - ray-tracing is costly
  - variance decreases only linearly with sample count
- General transport effects
  - depth of field, motion blur, etc

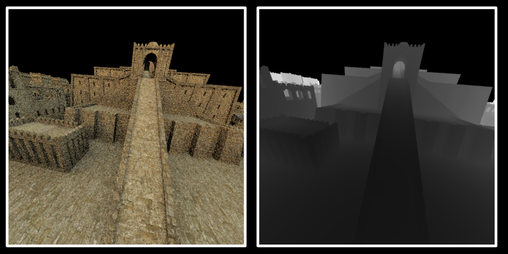
Previous work: Monte Carlo denoising

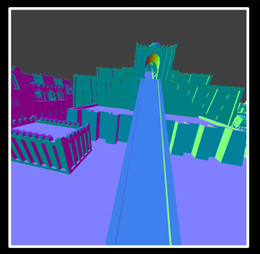
- Image-space adaptive sampling and reconstruction
  - often guided by aux. features (normals, depth, albedo)
  - Overbeck [2009], Dammertz [2010], Rouselle [2011], Li [2012], Sen & Darabi [2012], Bitterli [2016]



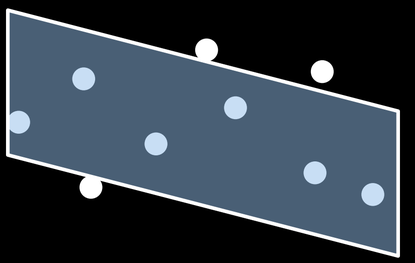
Previous work: Monte Carlo denoising

- Image-space adaptive sampling and reconstruction
  - often guided by aux. features (normals, depth, albedo)
  - Overbeck [2009], Dammertz [2010], Rouselle [2011], Li [2012], Sen & Darabi [2012], Bitterli [2016]



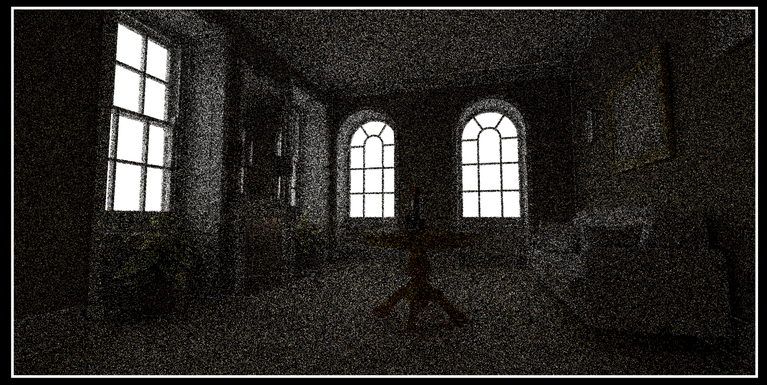


- High-dimensional filtering
  - nearest neighbor filter [Hashisuka 2008]
  - sample re-projection [Lehtinen 2011; Lehtinen 2012]
  - sheared filters [Egan 2009; Egan 2011; Yan 2015; Wu 2017]

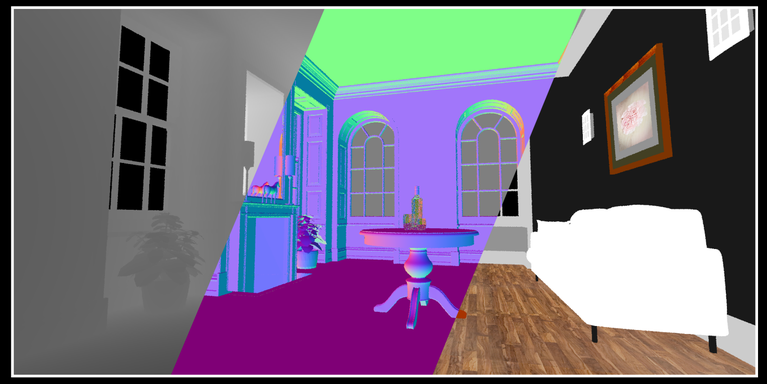


Deep-learning-based denoising

[Bako 2017; Chaitanya 2017; Vogels 2018]



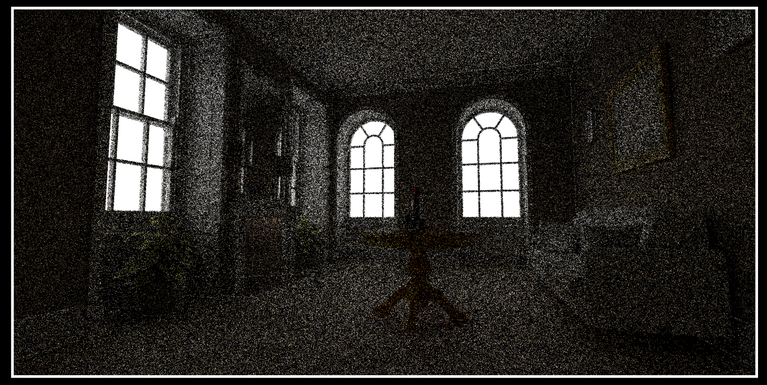
noisy image



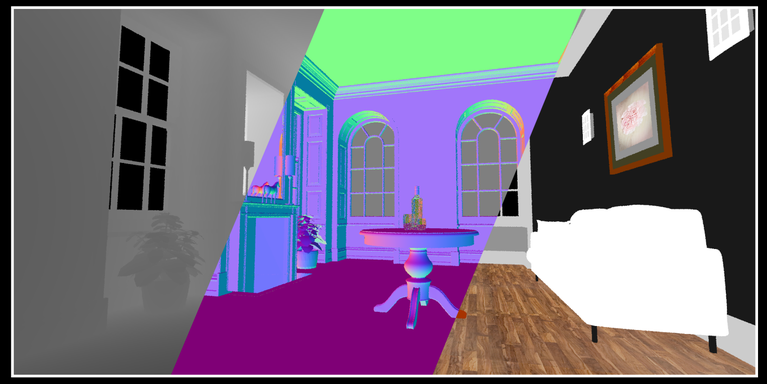
averaged auxiliary buffers
depth, normals, albedo

Deep-learning-based denoising

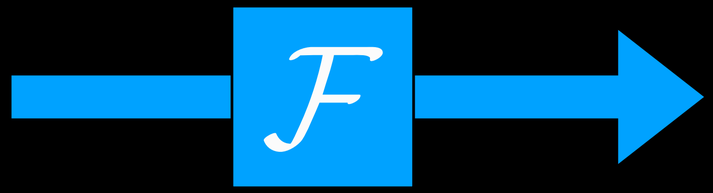
[Bako 2017; Chaitanya 2017; Vogels 2018]



noisy image



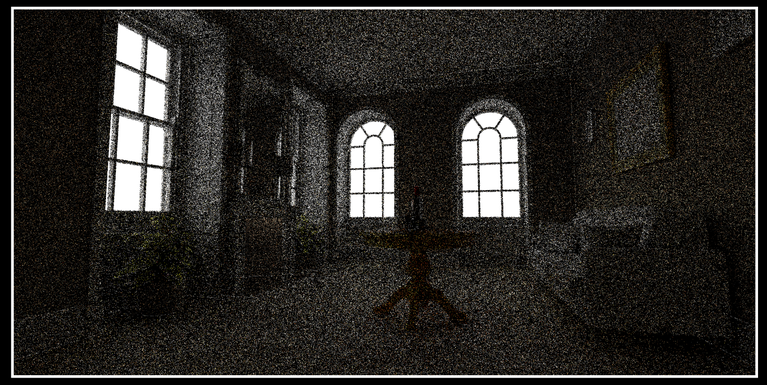
averaged auxiliary buffers
depth, normals, albedo



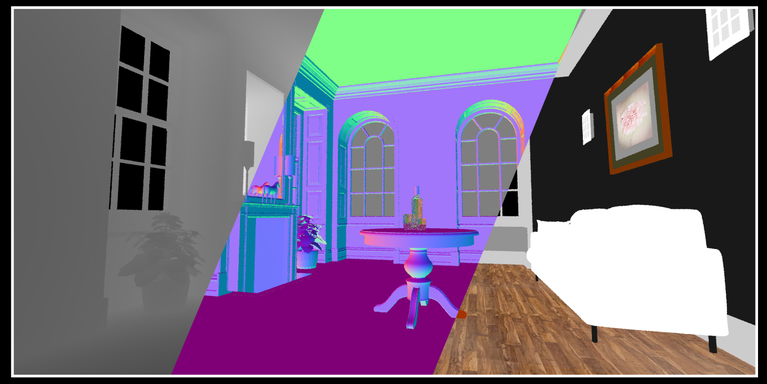
neural network

Deep-learning-based denoising

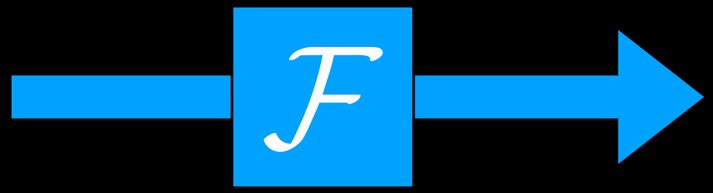
[Bako 2017; Chaitanya 2017; Vogels 2018]



noisy image



averaged auxiliary buffers
depth, normals, albedo

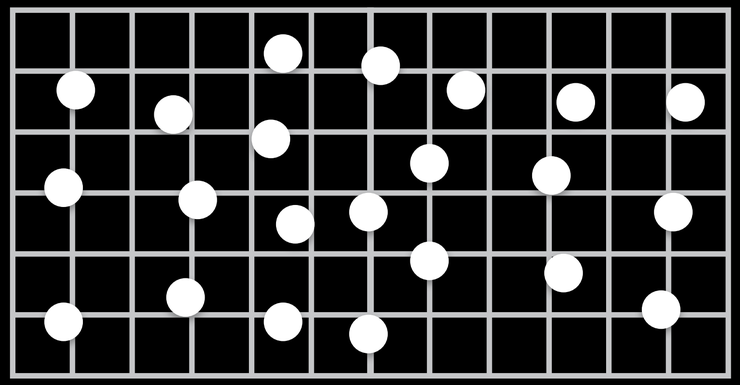


neural network

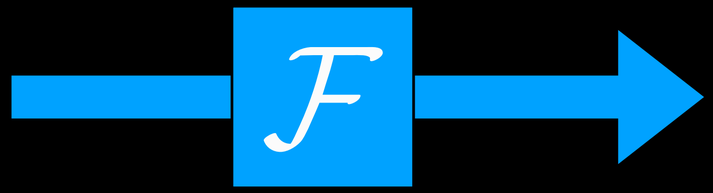
denoised image

Our approach

Learning to denoise, from the samples



input samples
(radiance + auxiliary features)

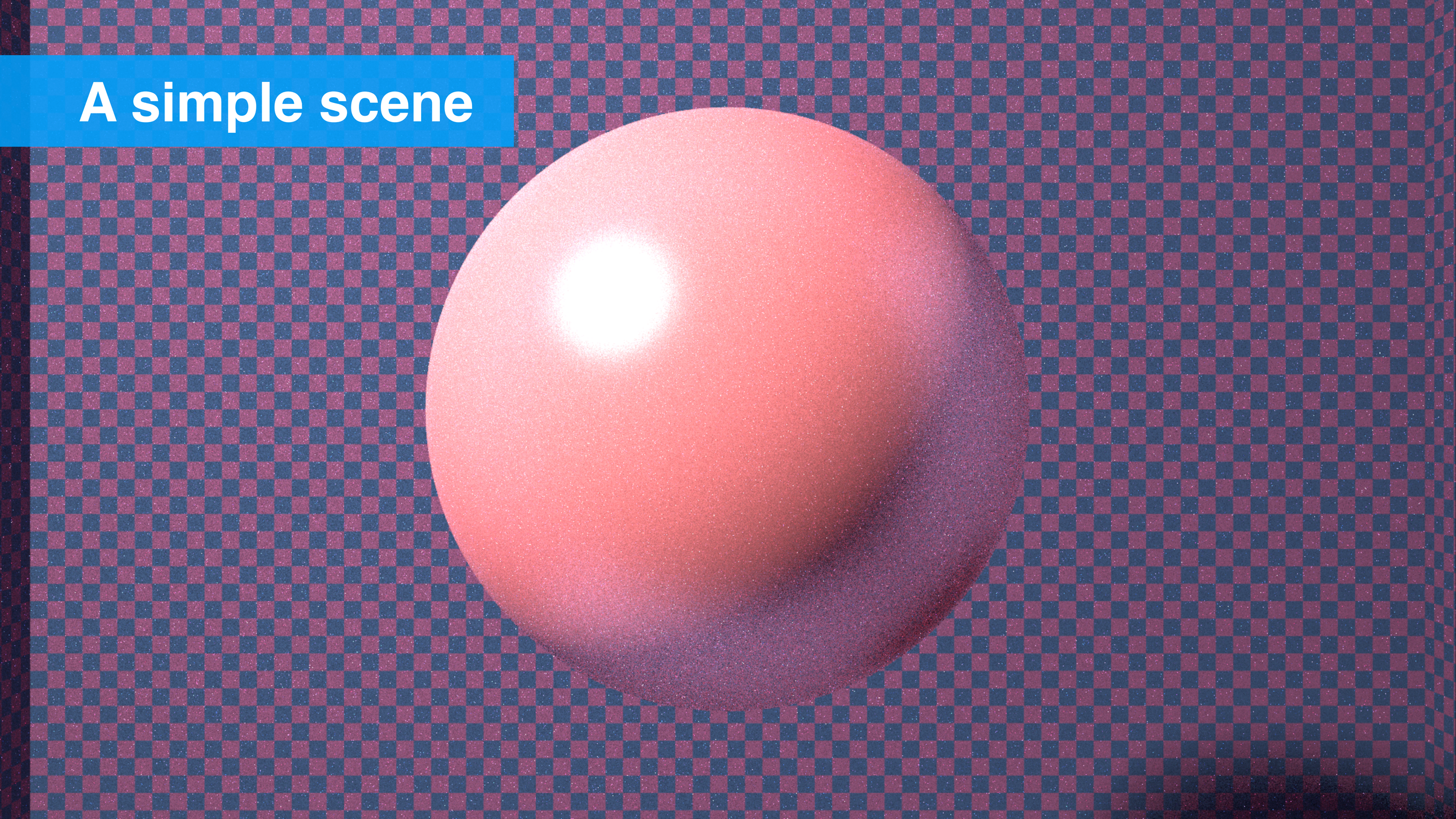


neural network

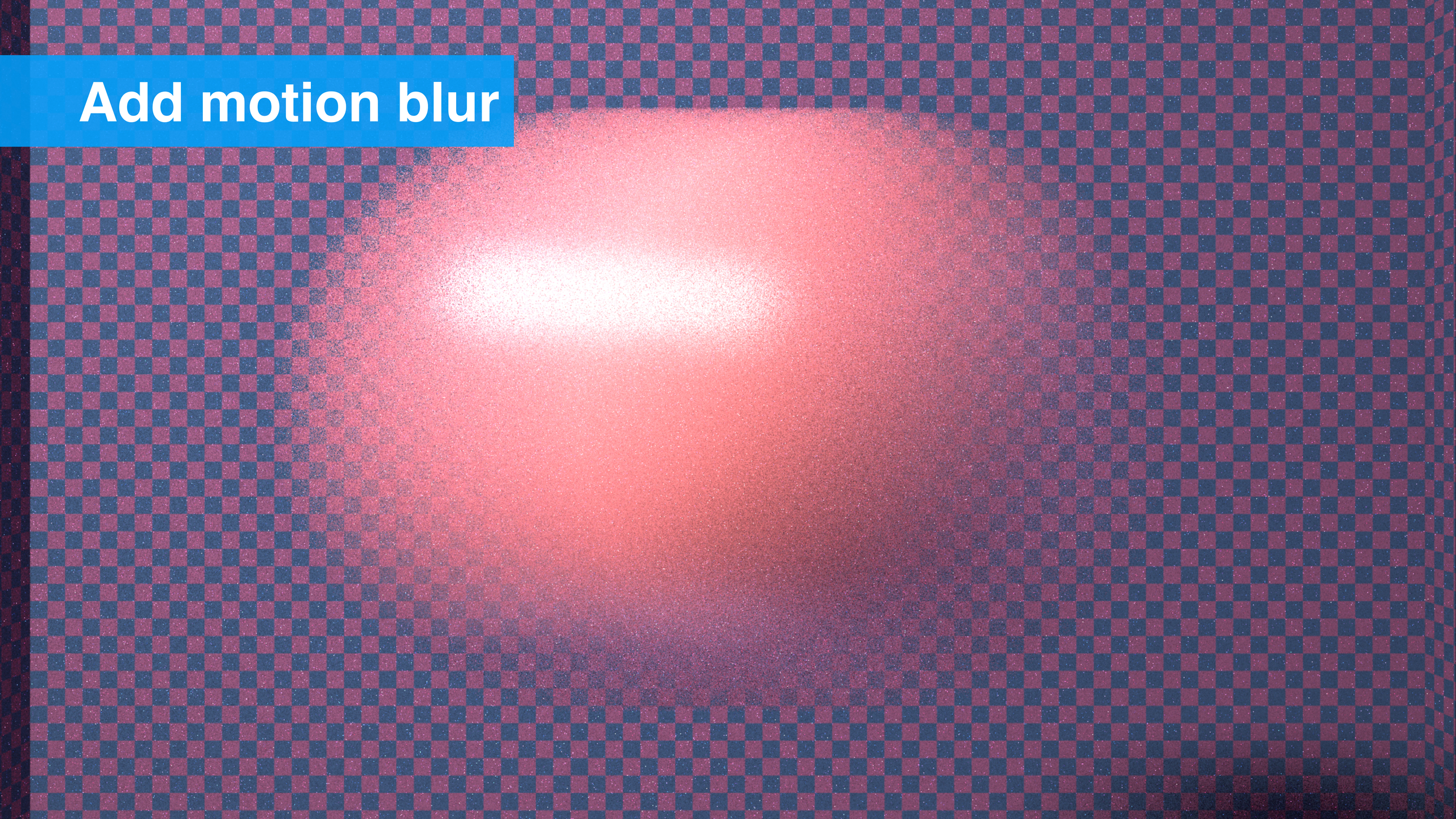
denoised image

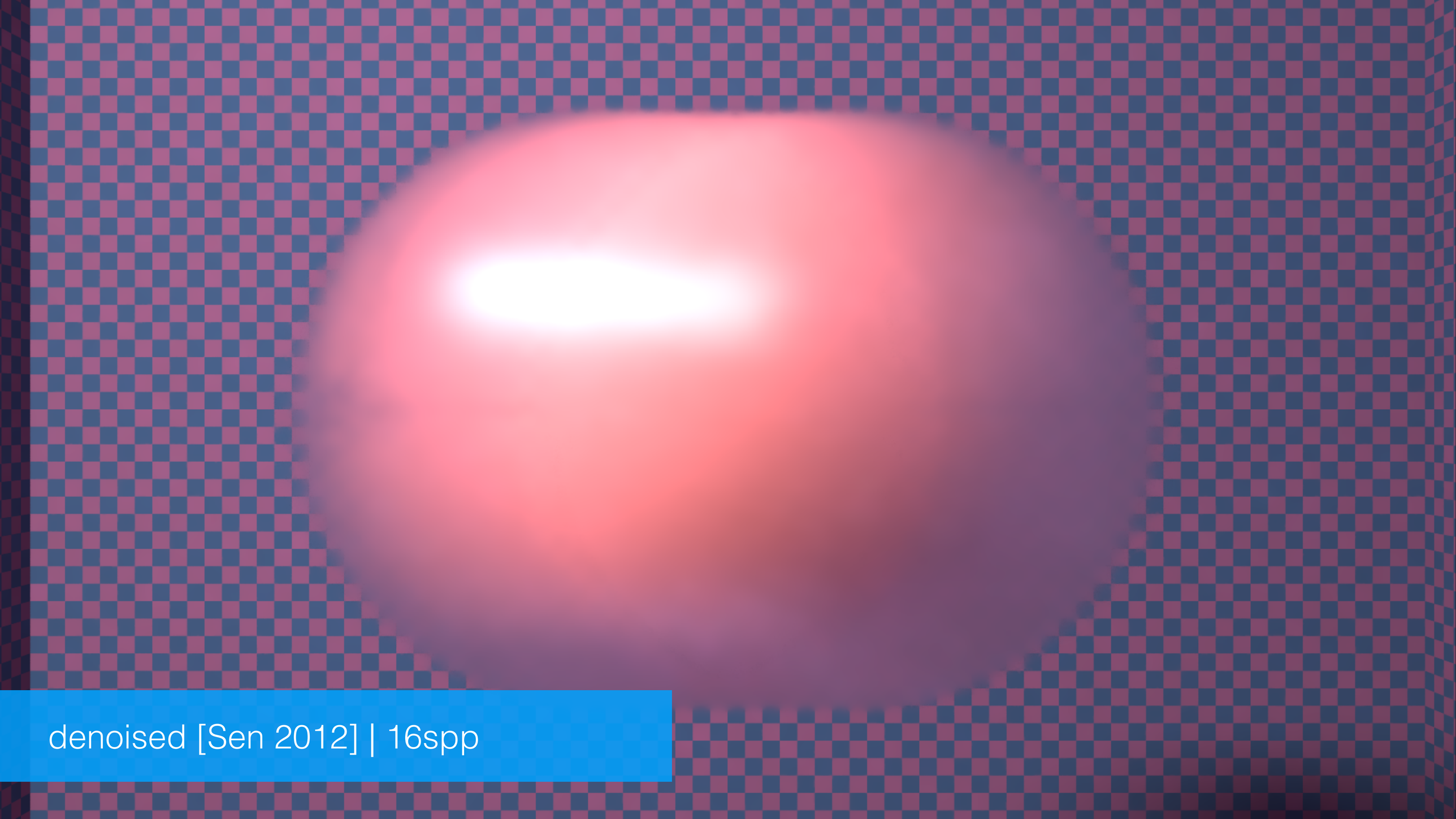
Why care about individual samples?

A simple scene

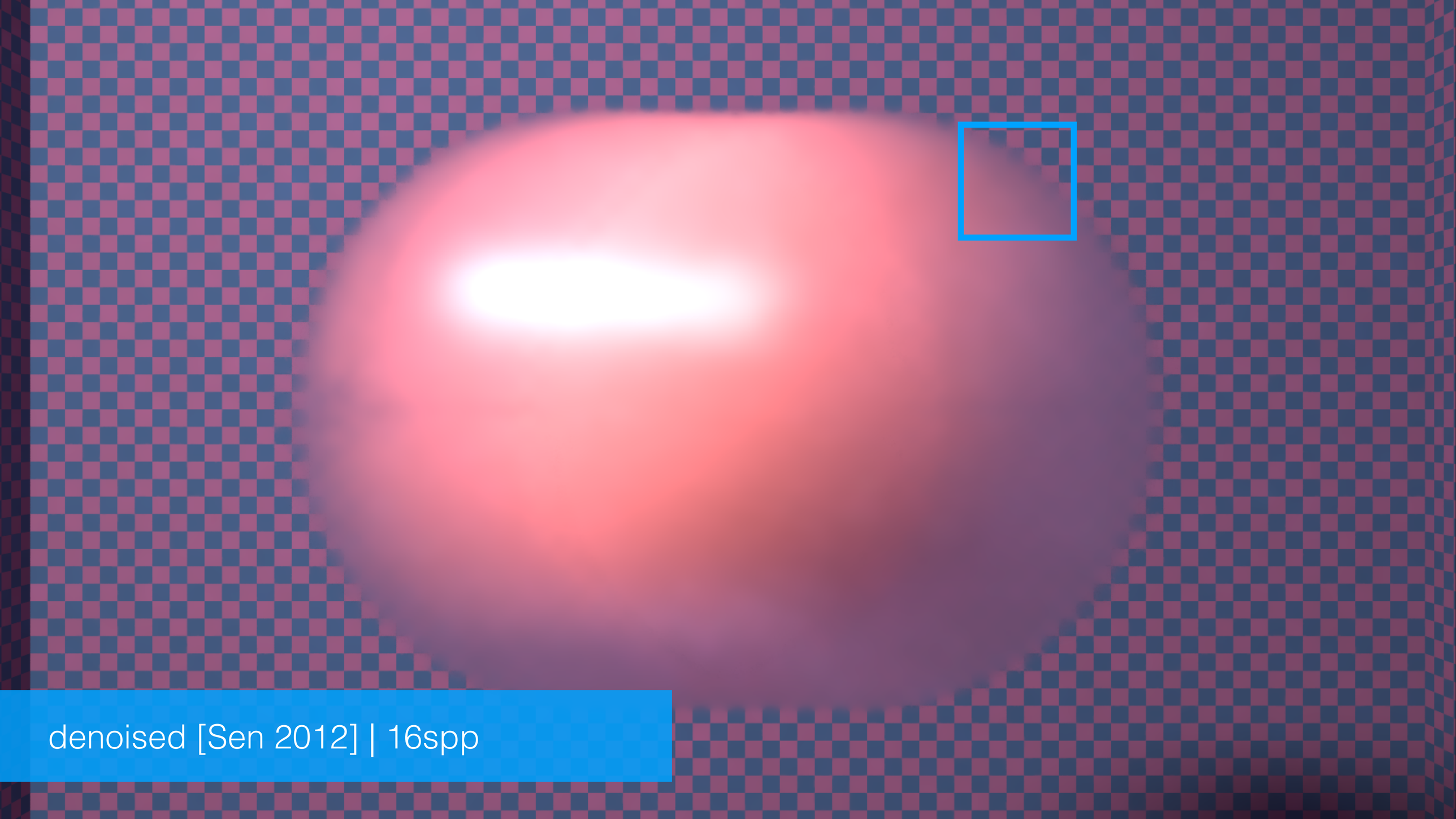


Add motion blur

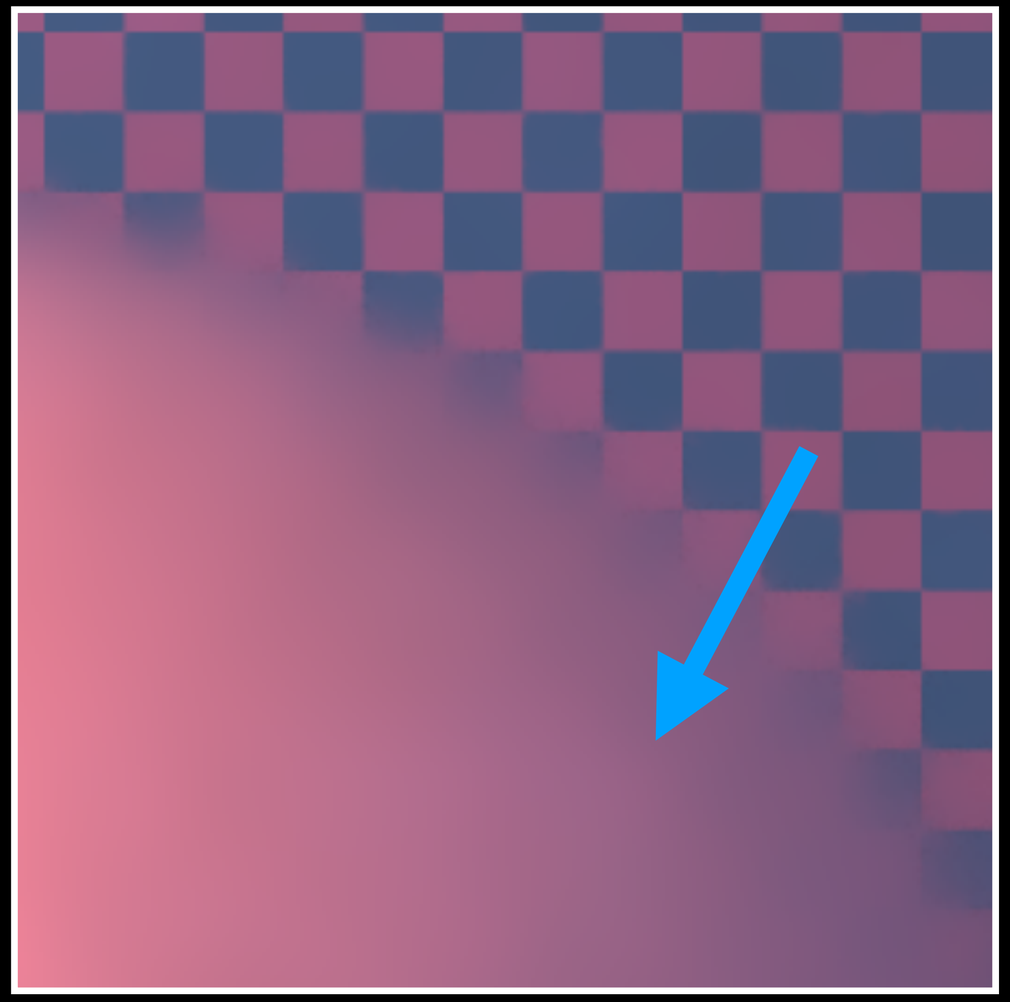




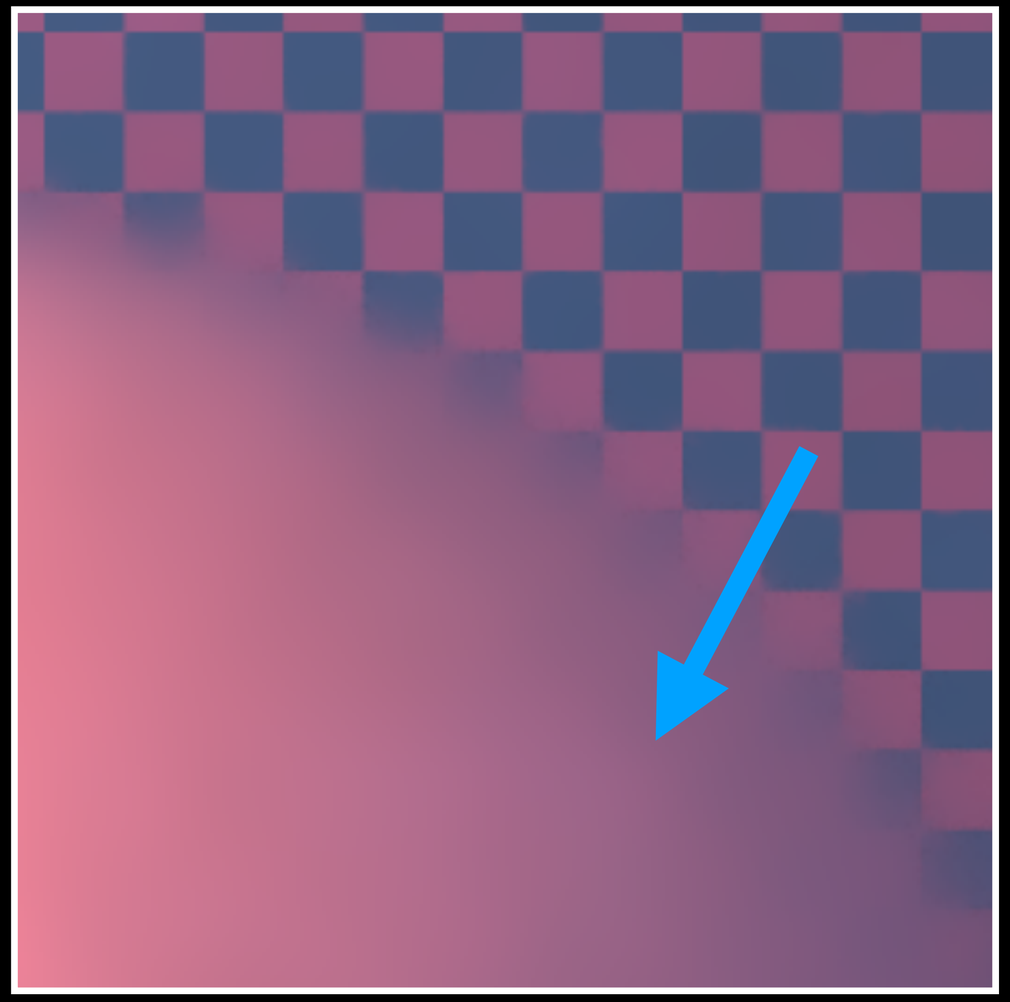
denoised [Sen 2012] | 16spp



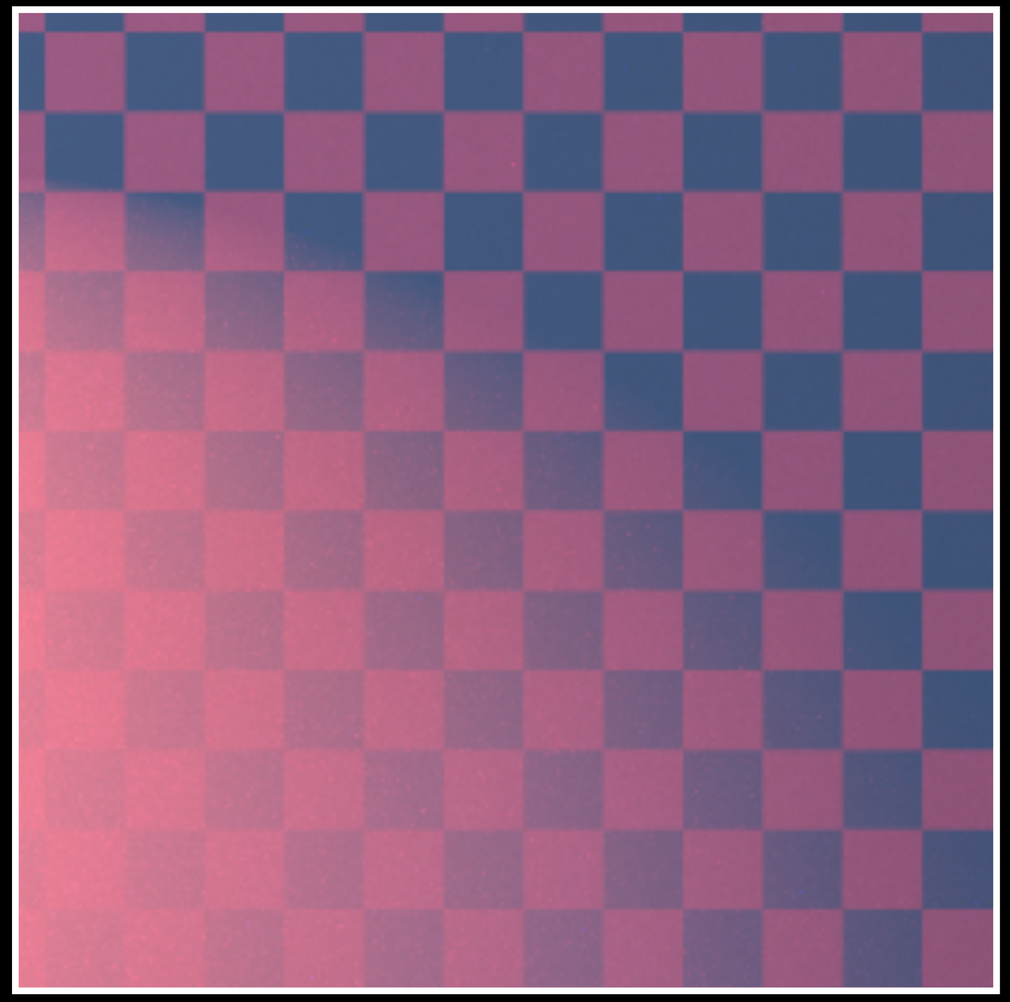
denoised [Sen 2012] | 16spp



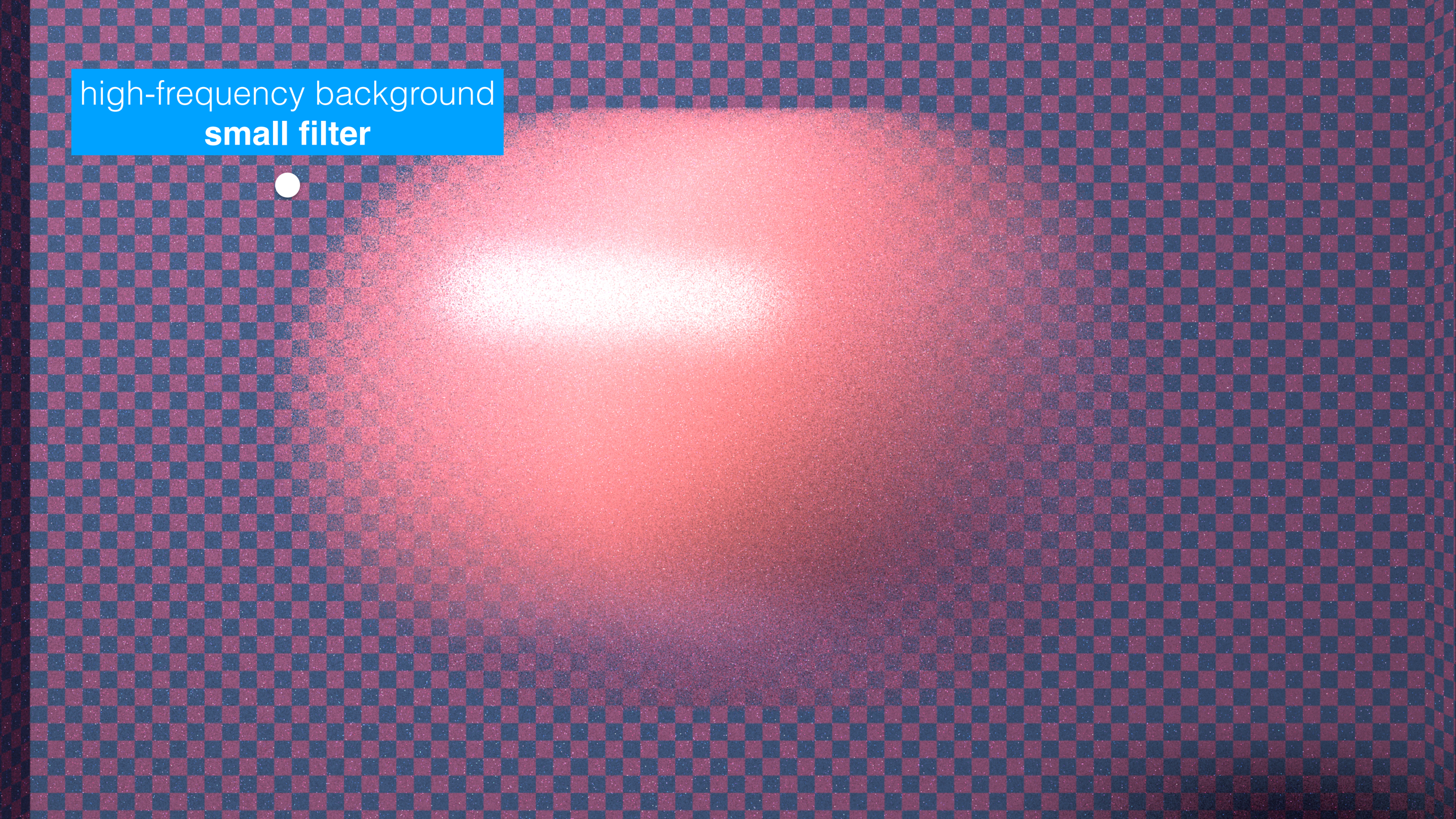
[Sen 2012] | 16spp

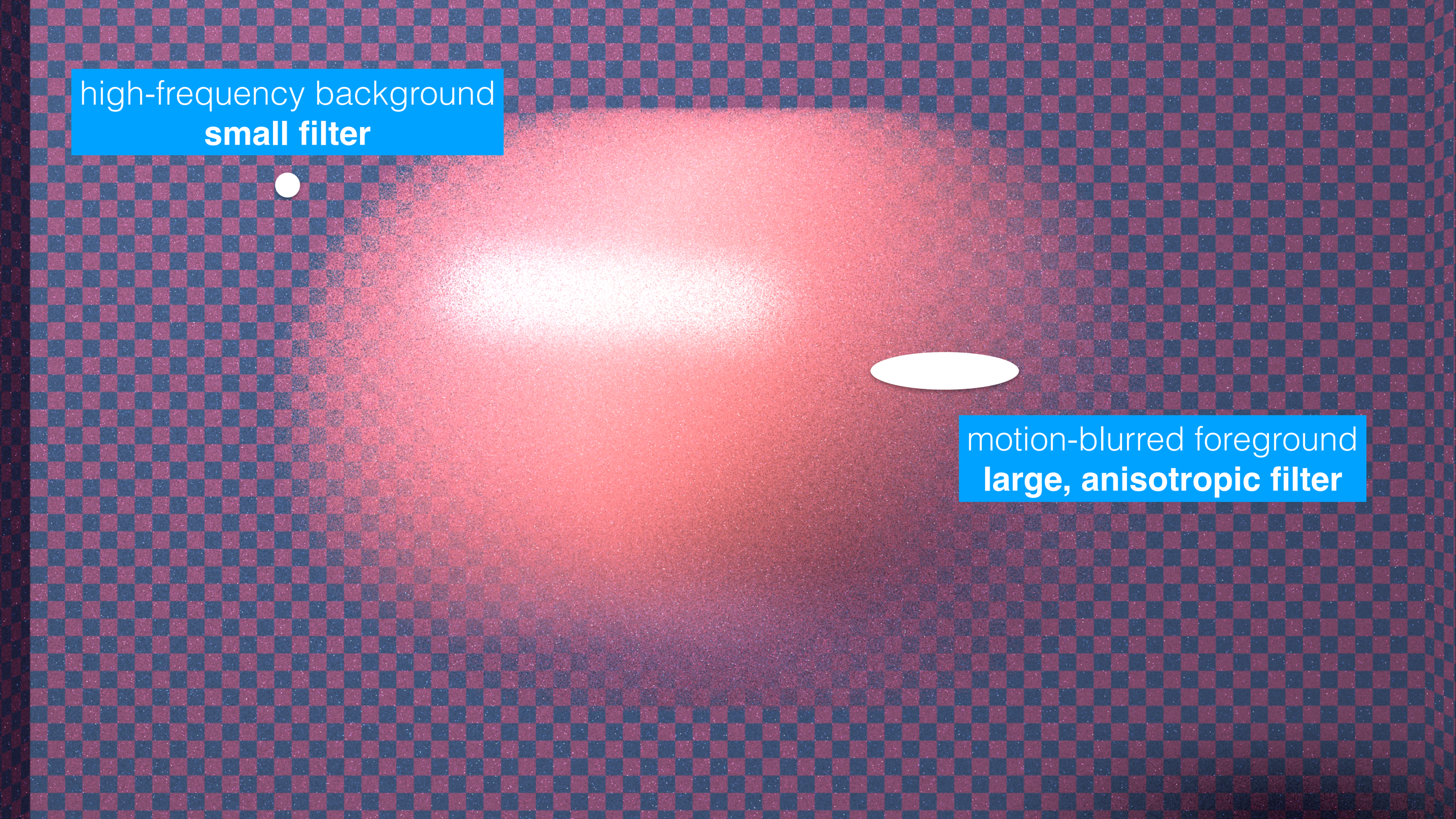


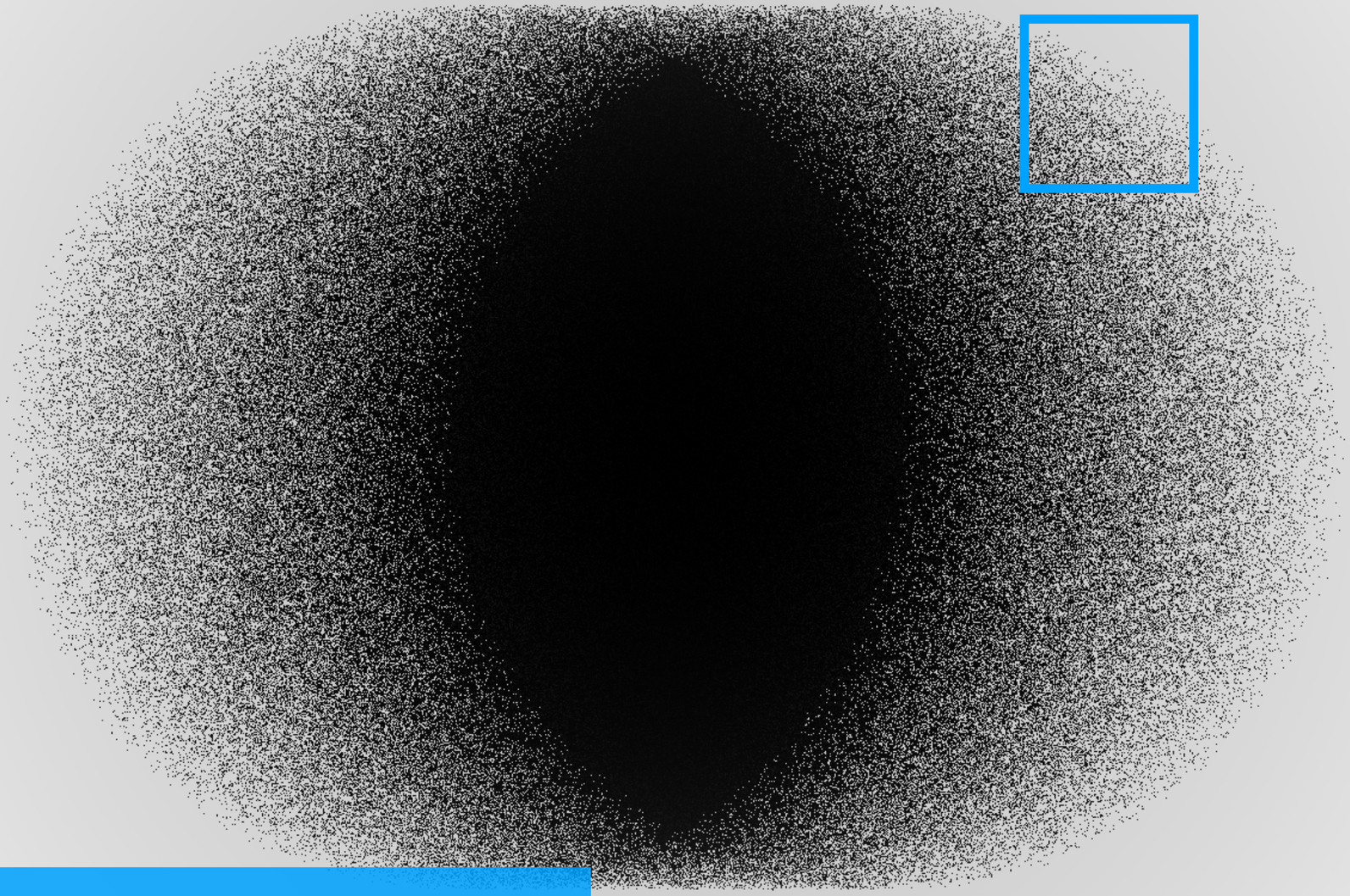
[Sen 2012] | 16spp



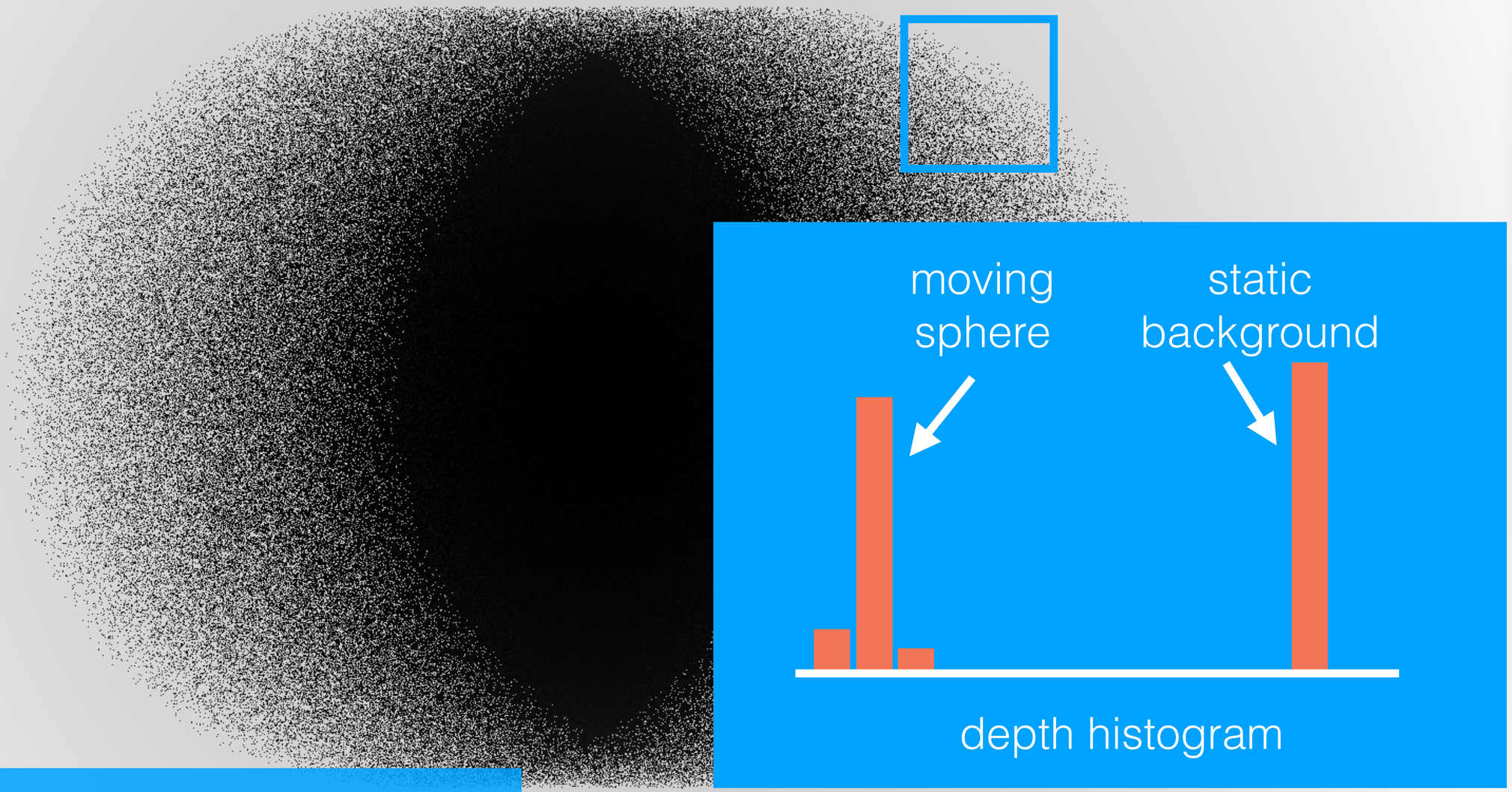
ground truth | 8192spp


high-frequency background
small filter

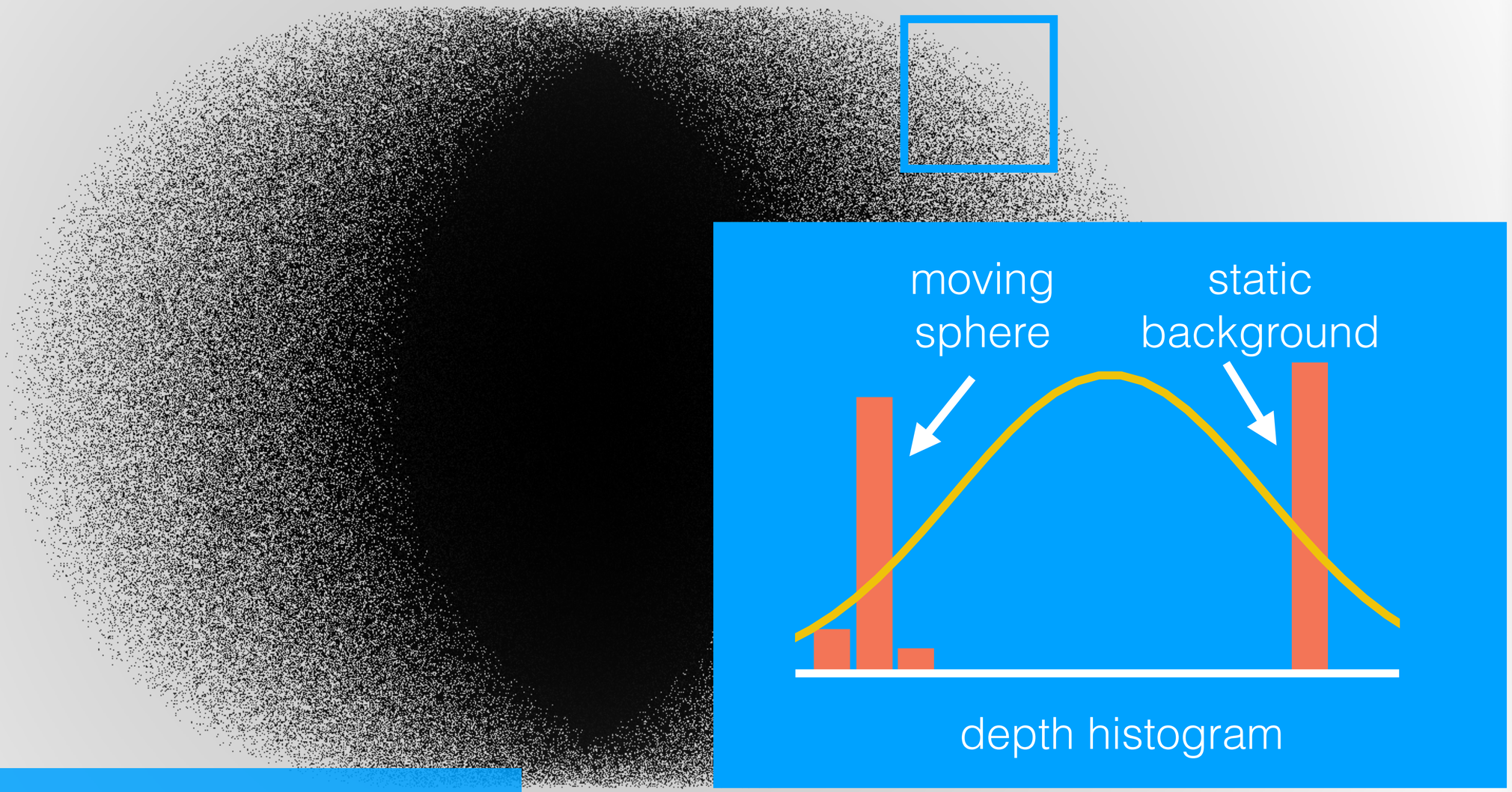

high-frequency background
small filter
motion-blurred foreground
large, anisotropic filter



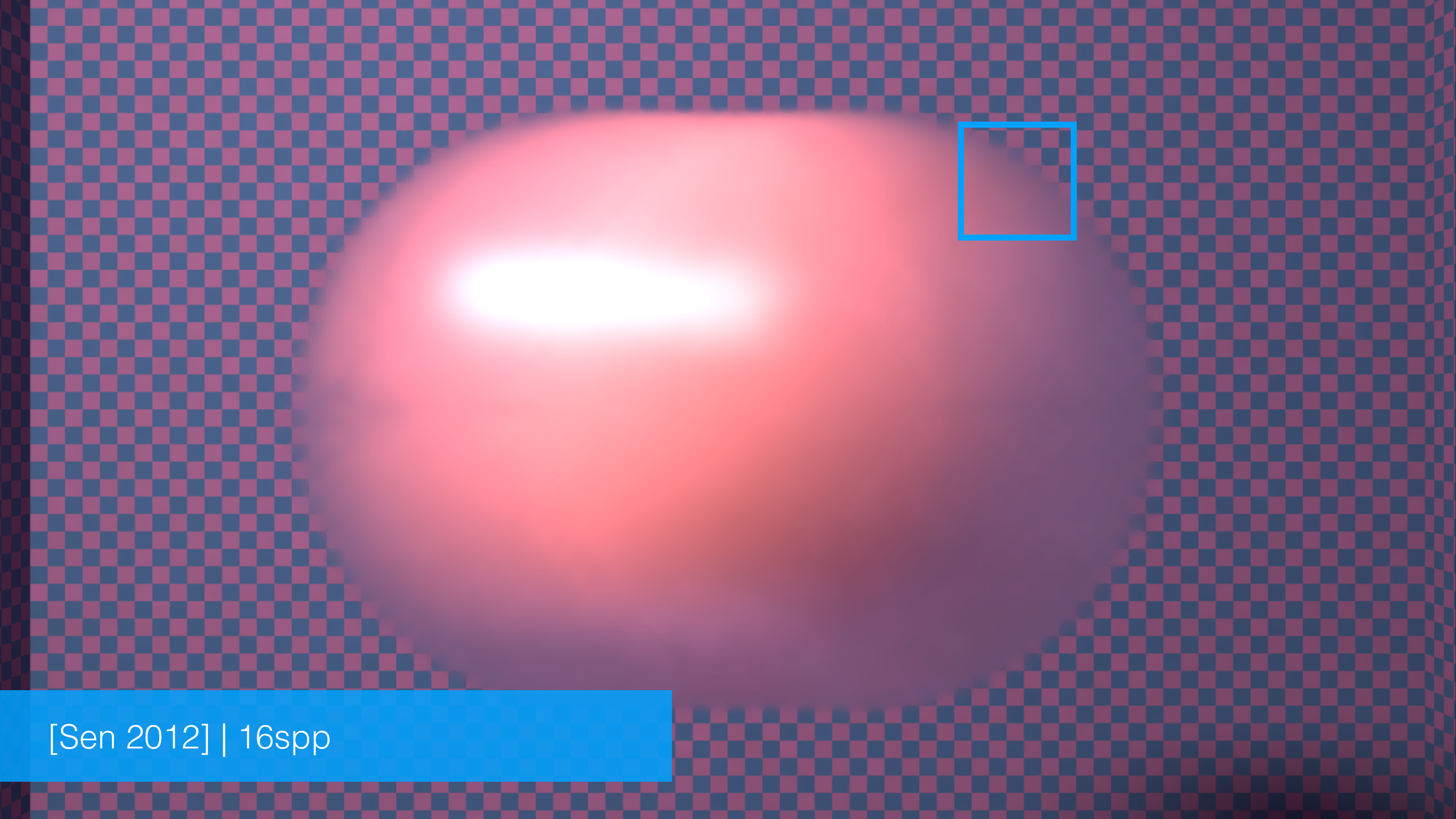
auxiliary depth buffer

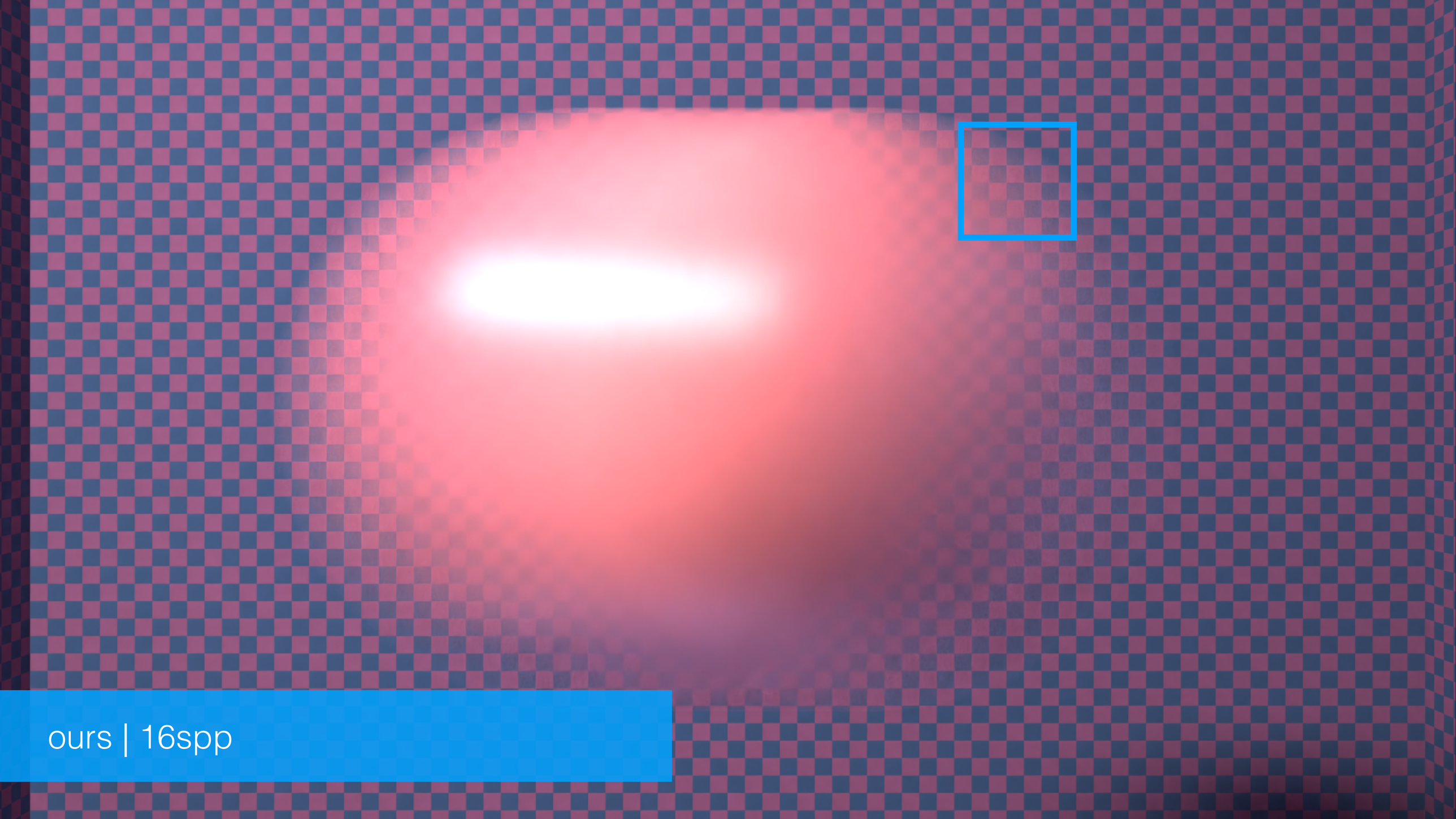


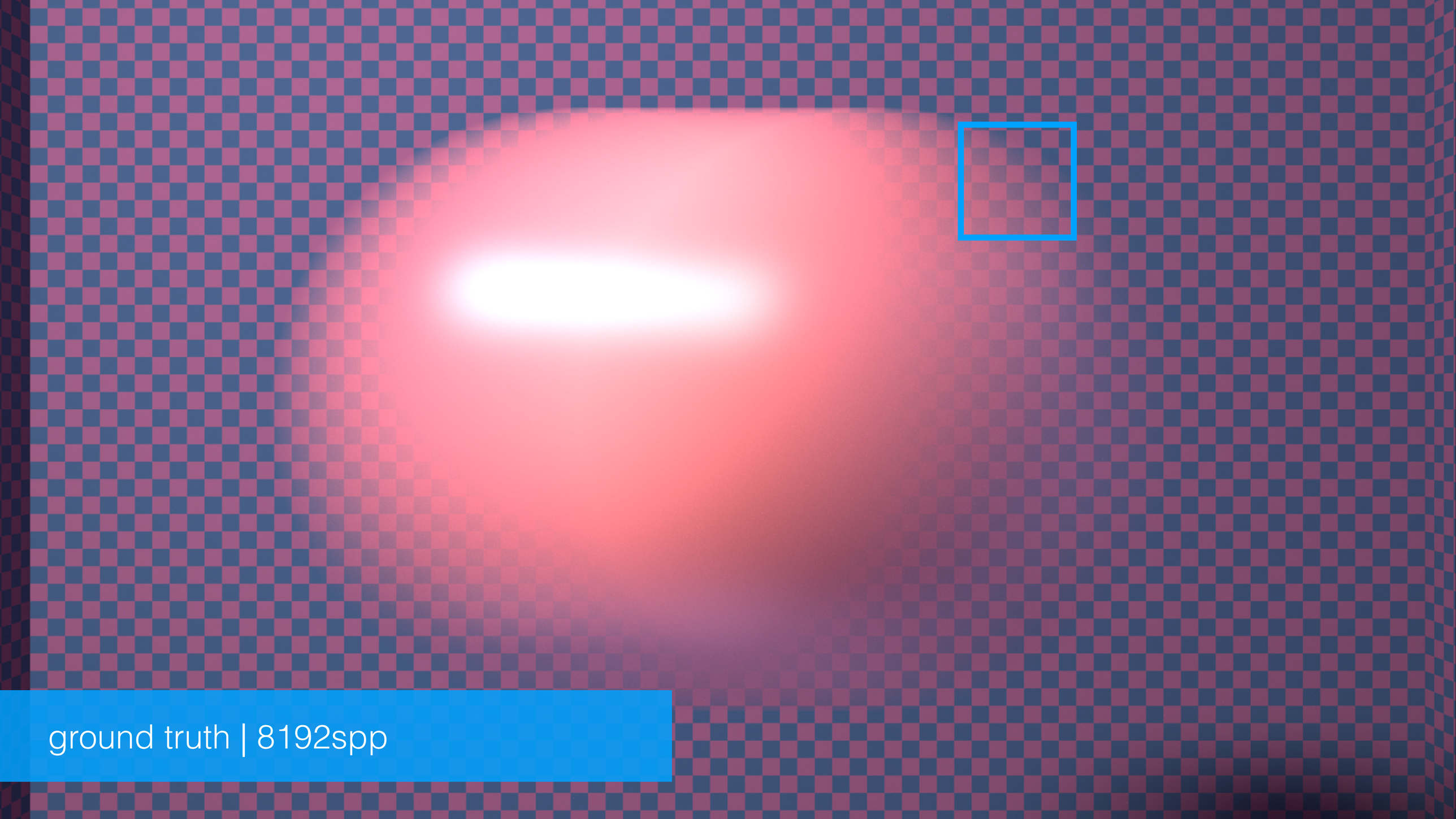
auxiliary depth buffer



auxiliary depth buffer

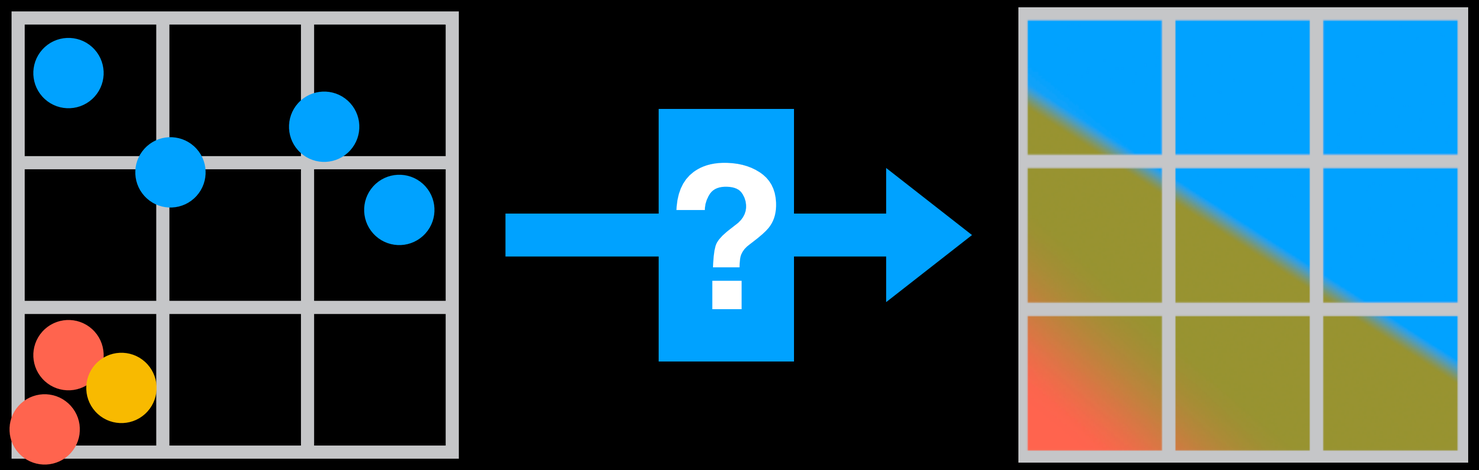

[Sen 2012] | 16spp


ours | 16spp

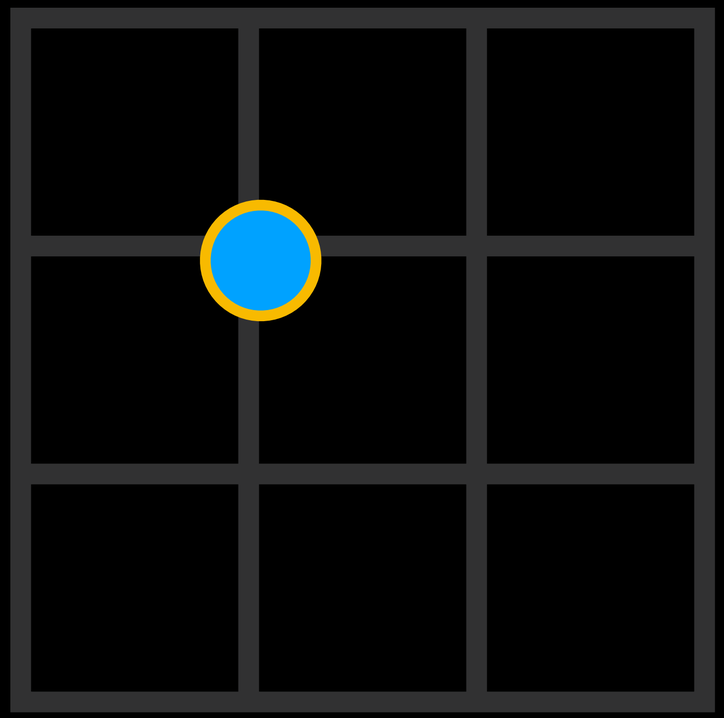


ground truth | 8192spp

Mapping samples to pixels: challenges

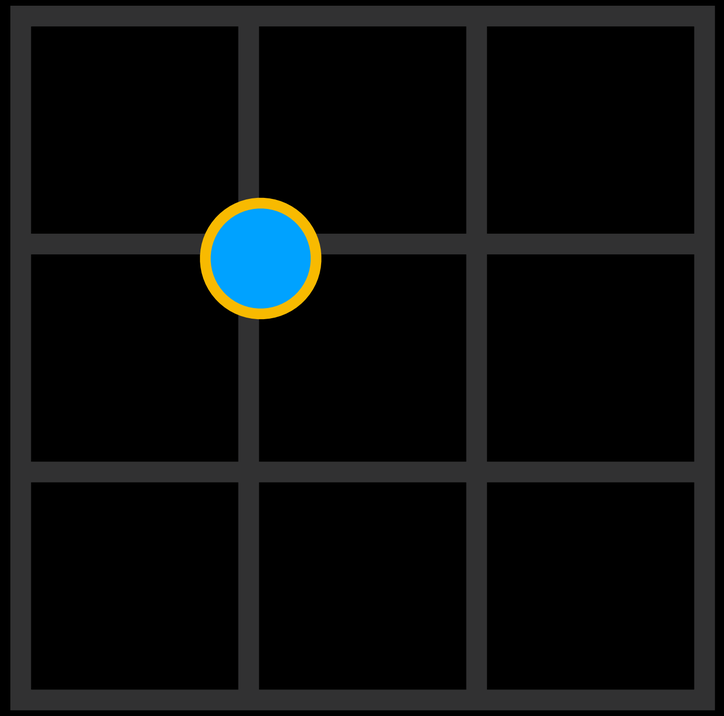


Samples need to know their surroundings

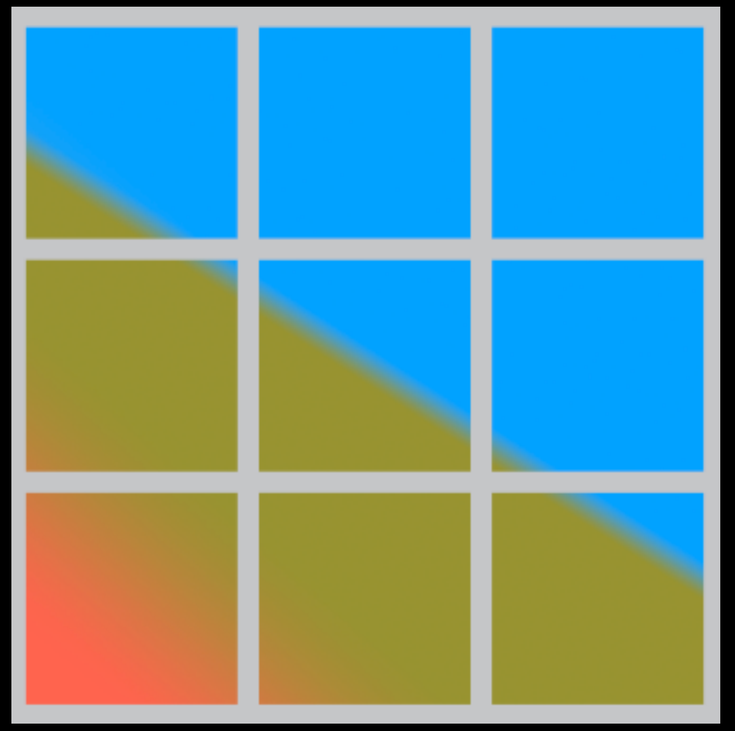


unordered samples

Samples need to know their surroundings

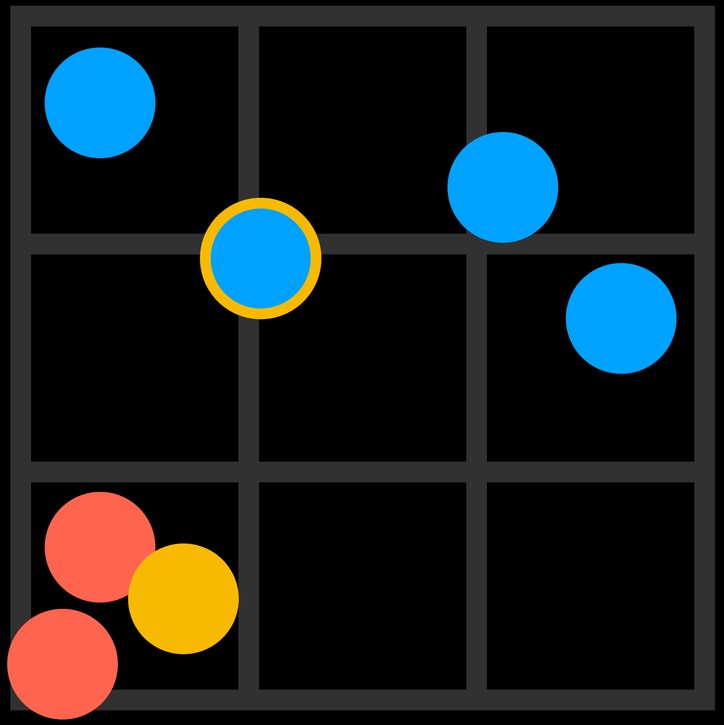


unordered samples

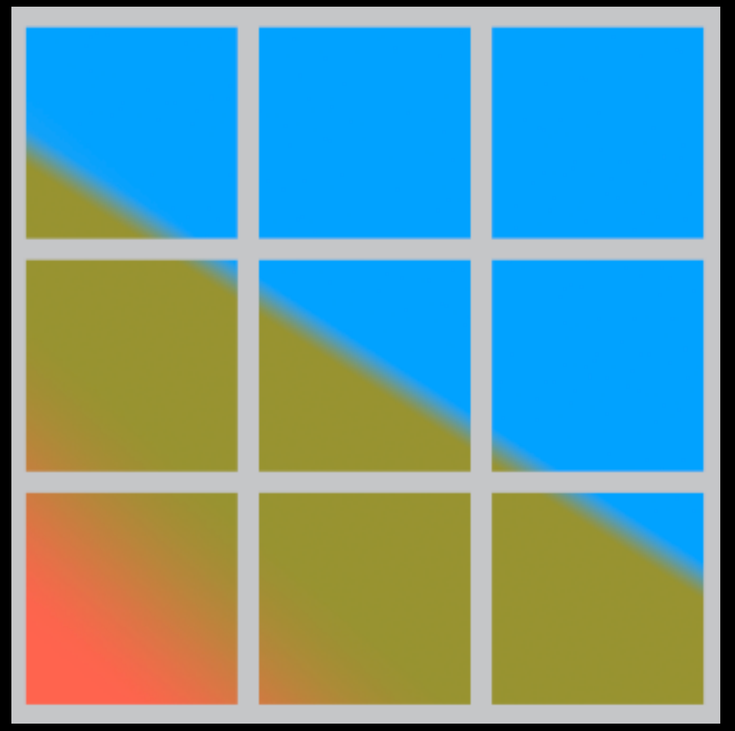


target reconstruction

Samples need to know their surroundings



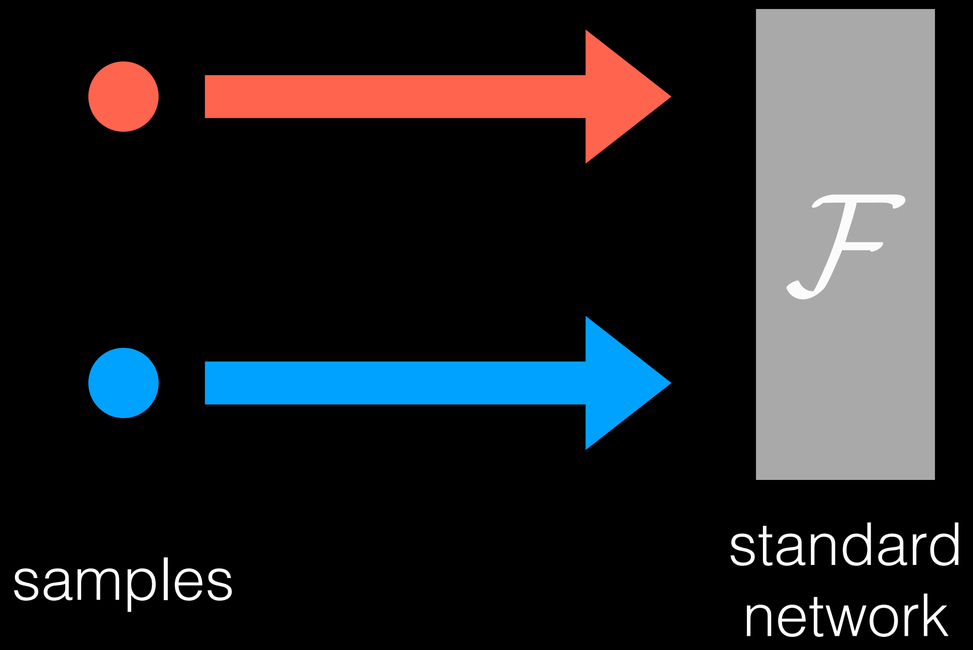
unordered samples



target reconstruction

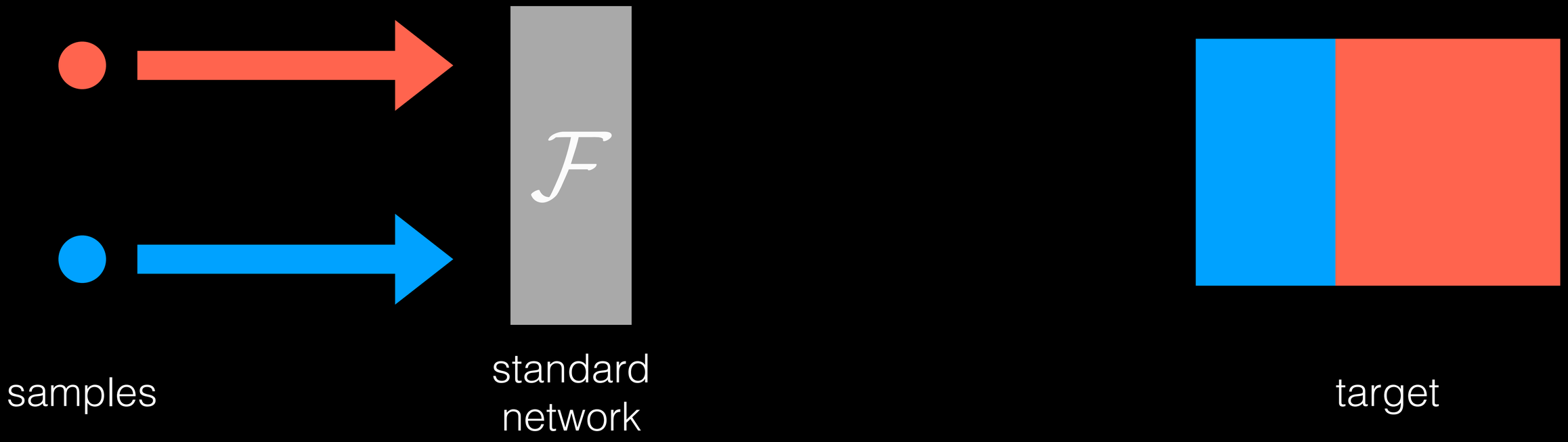
Standard networks have implicit ordering

and cannot track samples



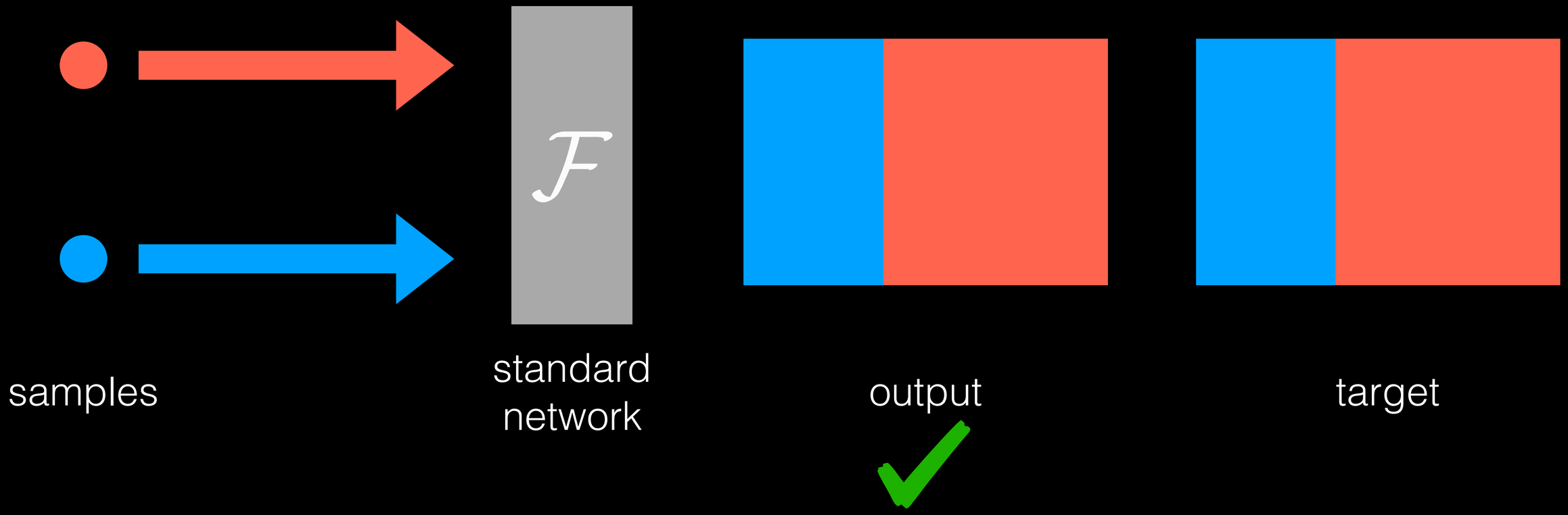
Standard networks have implicit ordering

and cannot track samples



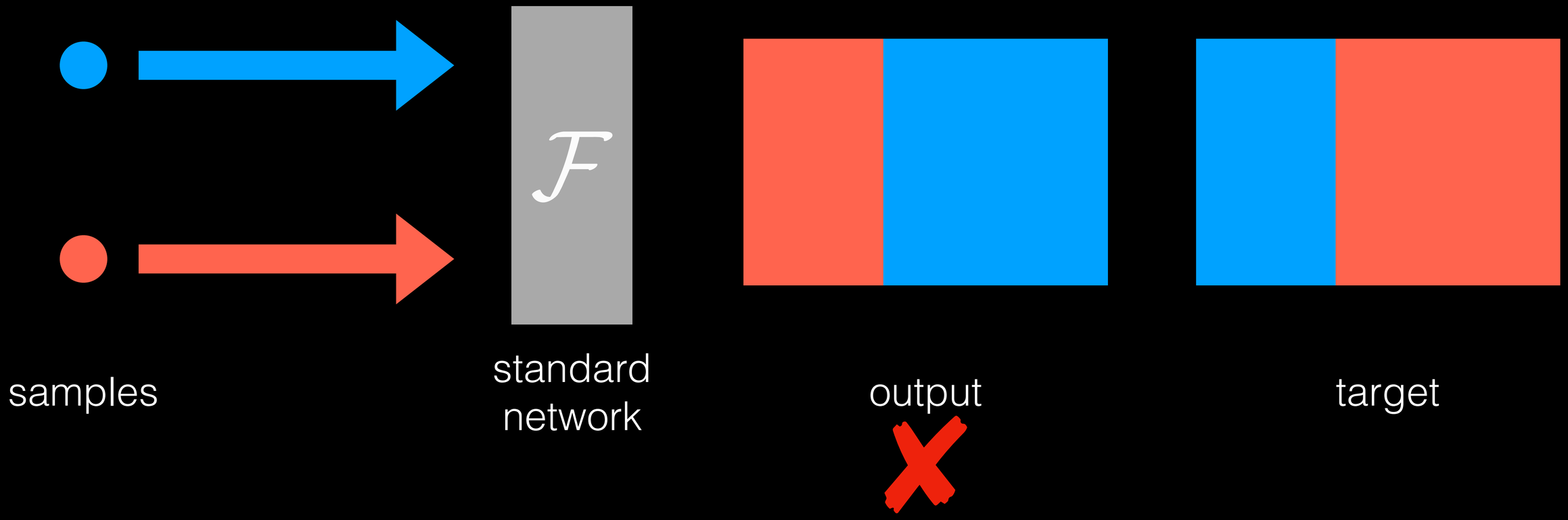
Standard networks have implicit ordering

and cannot track samples



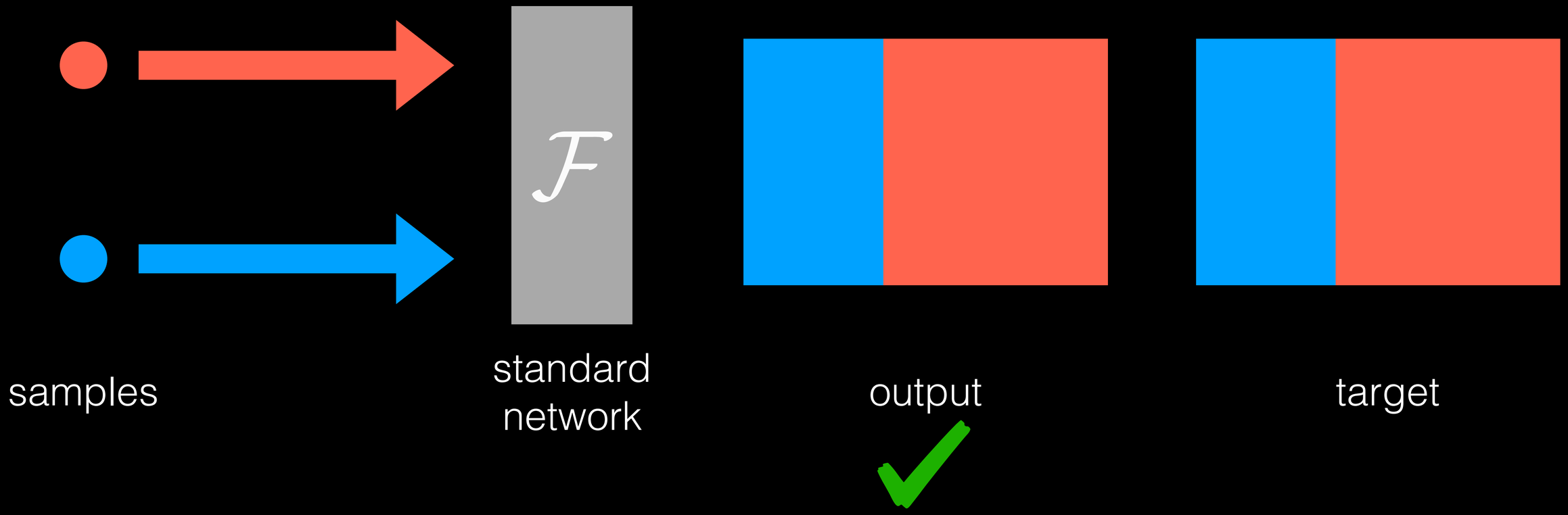
Standard networks have implicit ordering

and cannot track samples



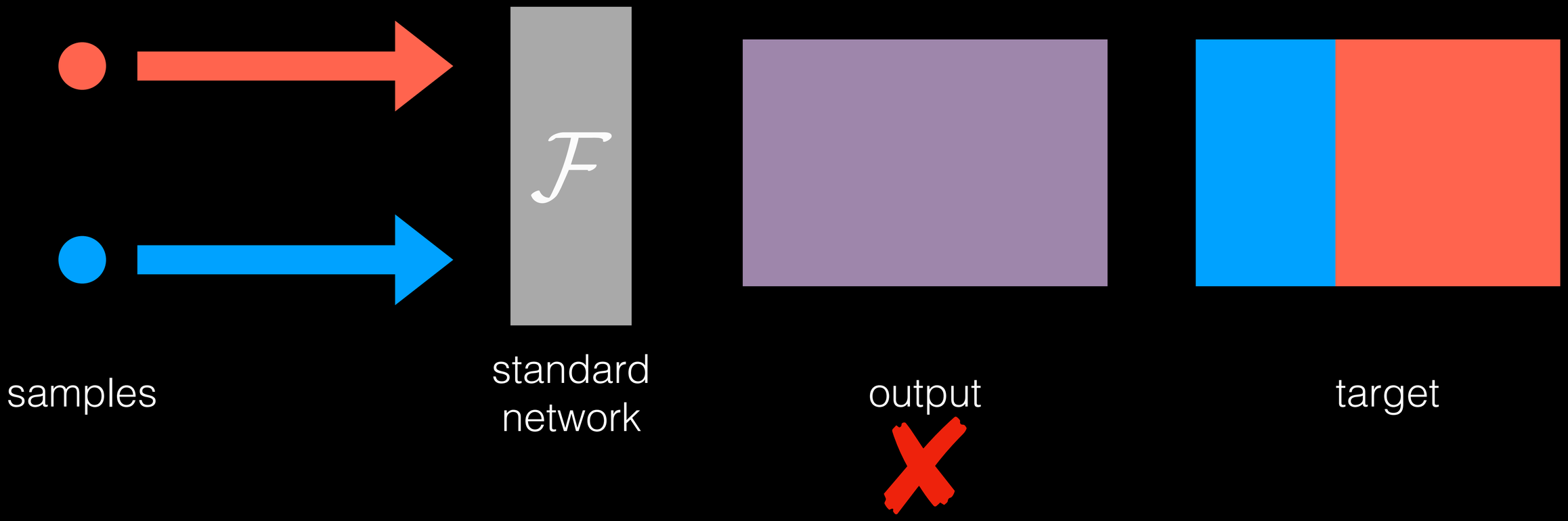
Standard networks have implicit ordering

and cannot track samples



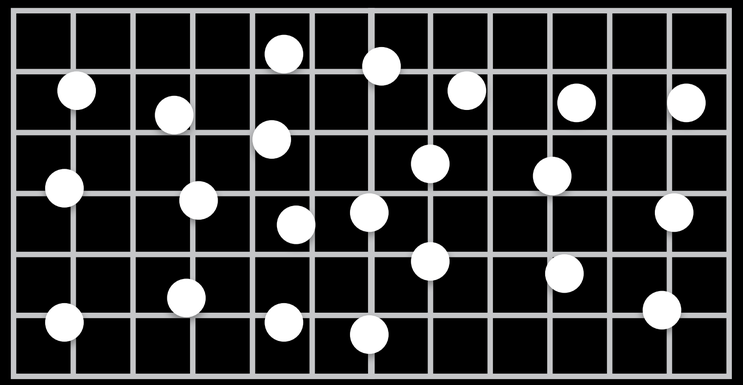
No better than averaging samples

and cannot track samples



Method: map unordered samples to pixels

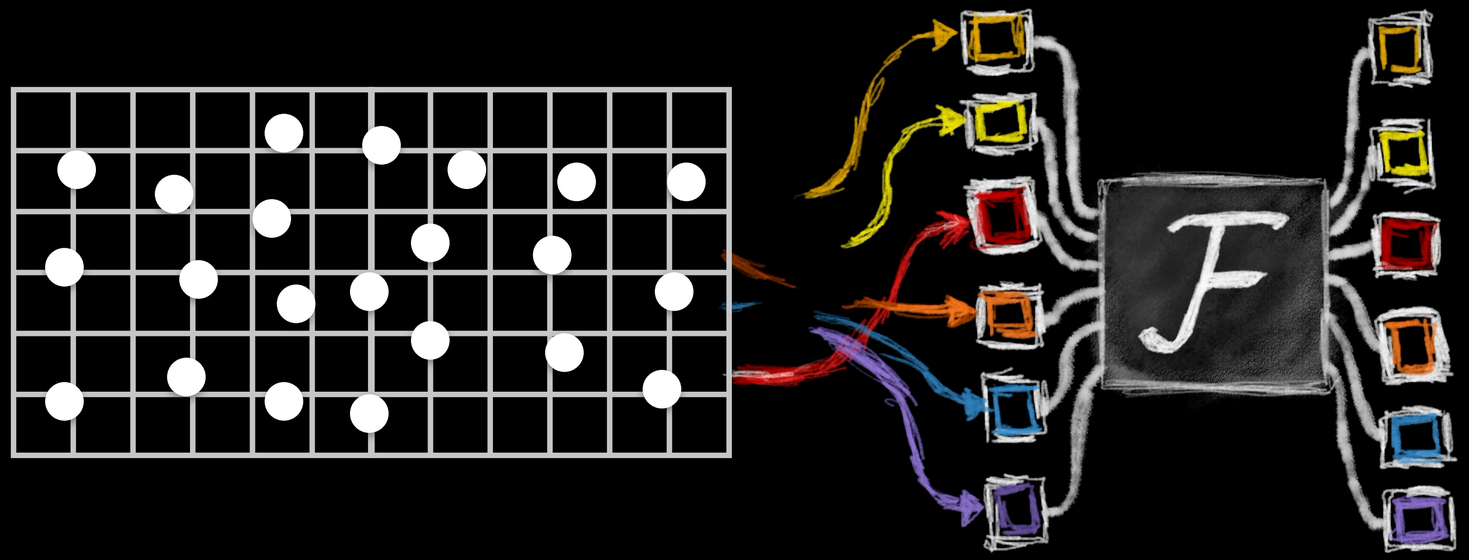
using splatting kernels



unordered samples

Method: map unordered samples to pixels

using splatting kernels

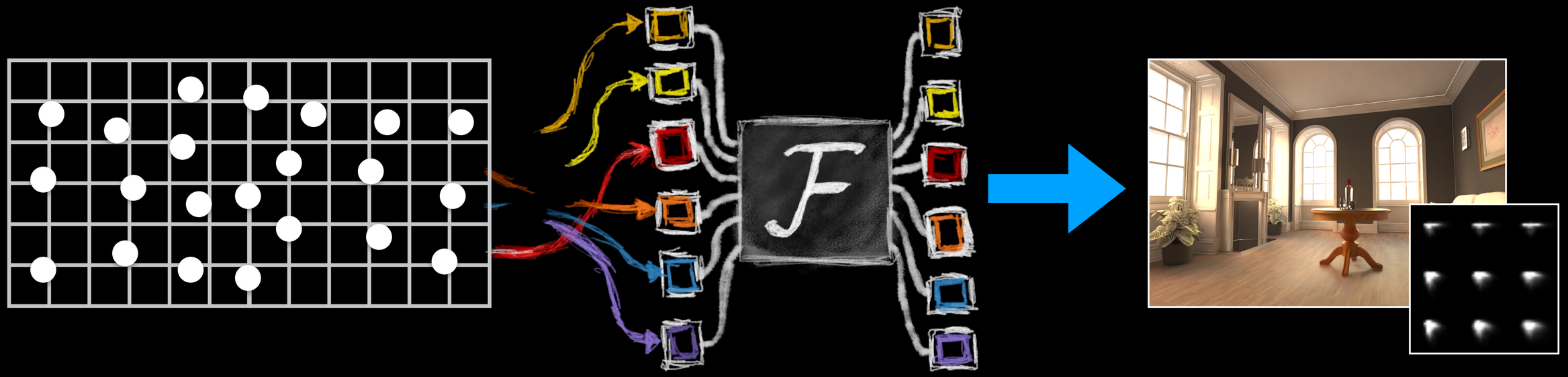


unordered samples

permutation-invariant neural network

Method: map unordered samples to pixels

using splatting kernels

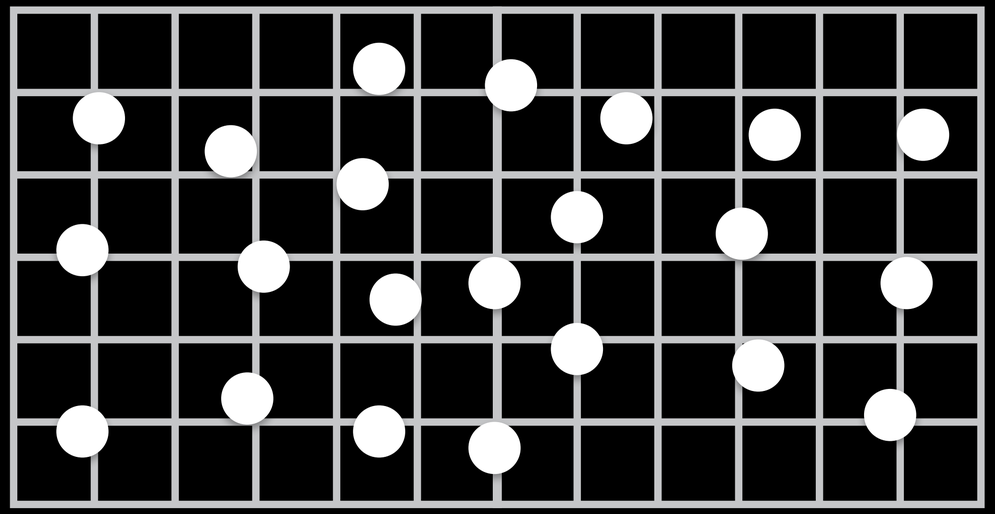


unordered samples

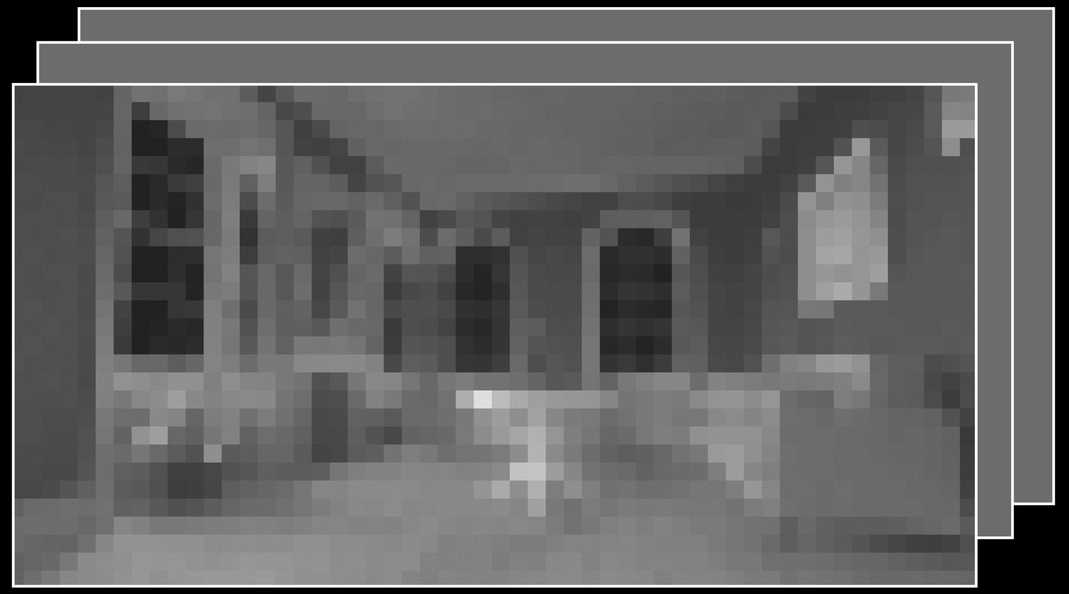
permutation-invariant neural network

splatting kernels reconstruction

Two representations: samples/pixels

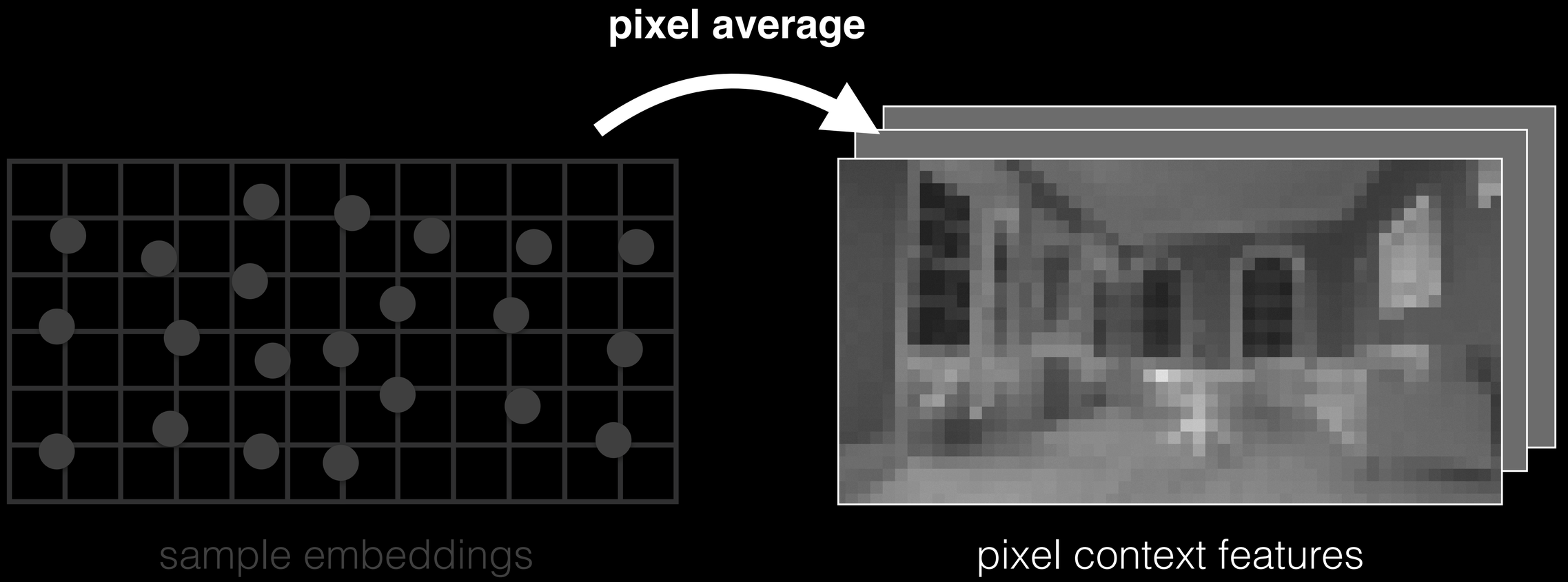


sample embeddings

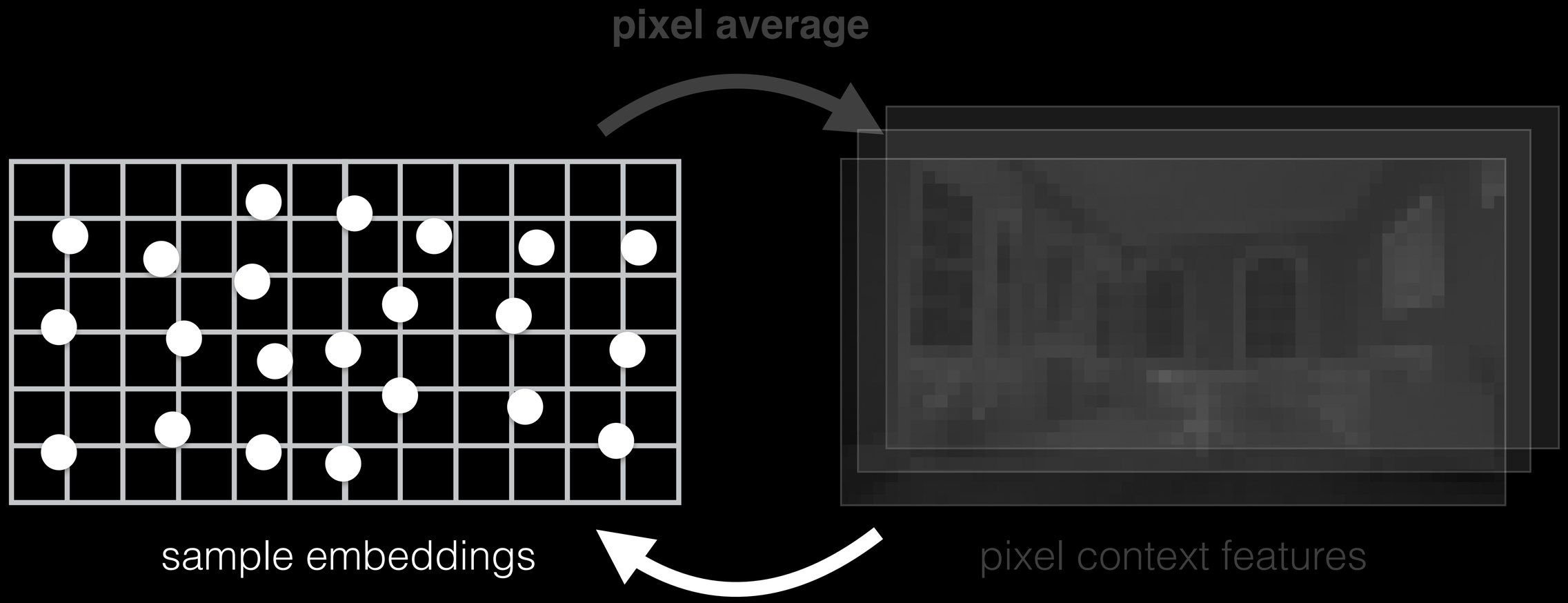


pixel context features

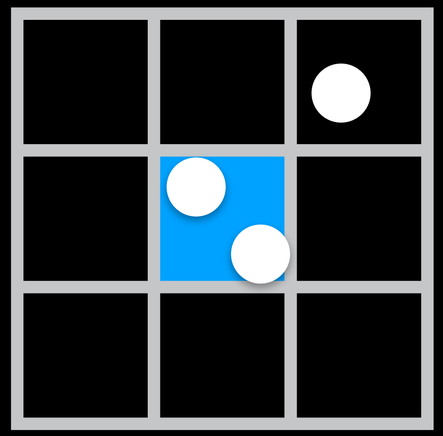
Two representations: samples/pixels



Two representations: samples/pixels

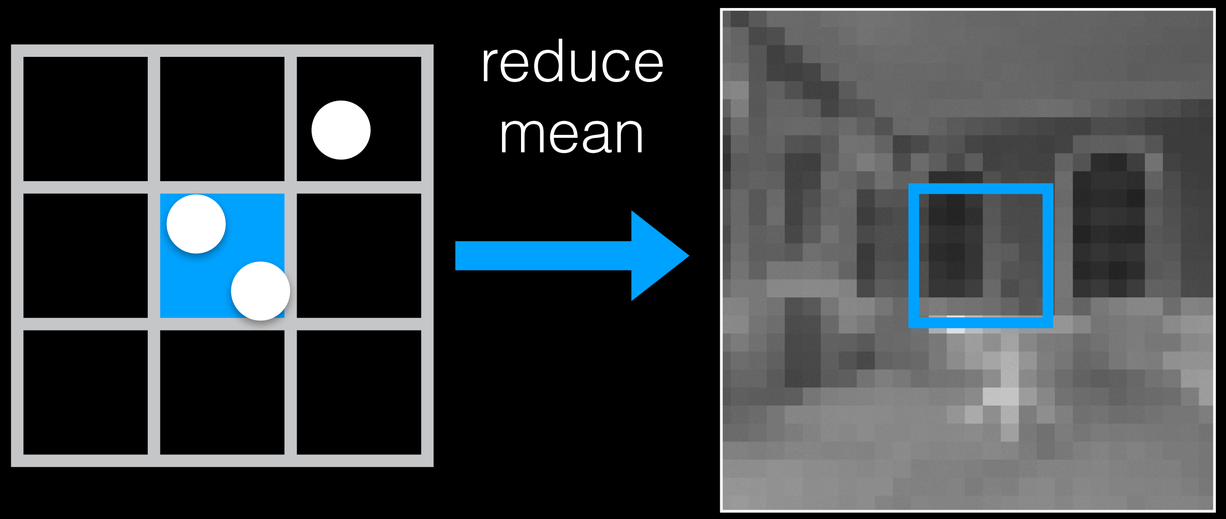


Update pixel context

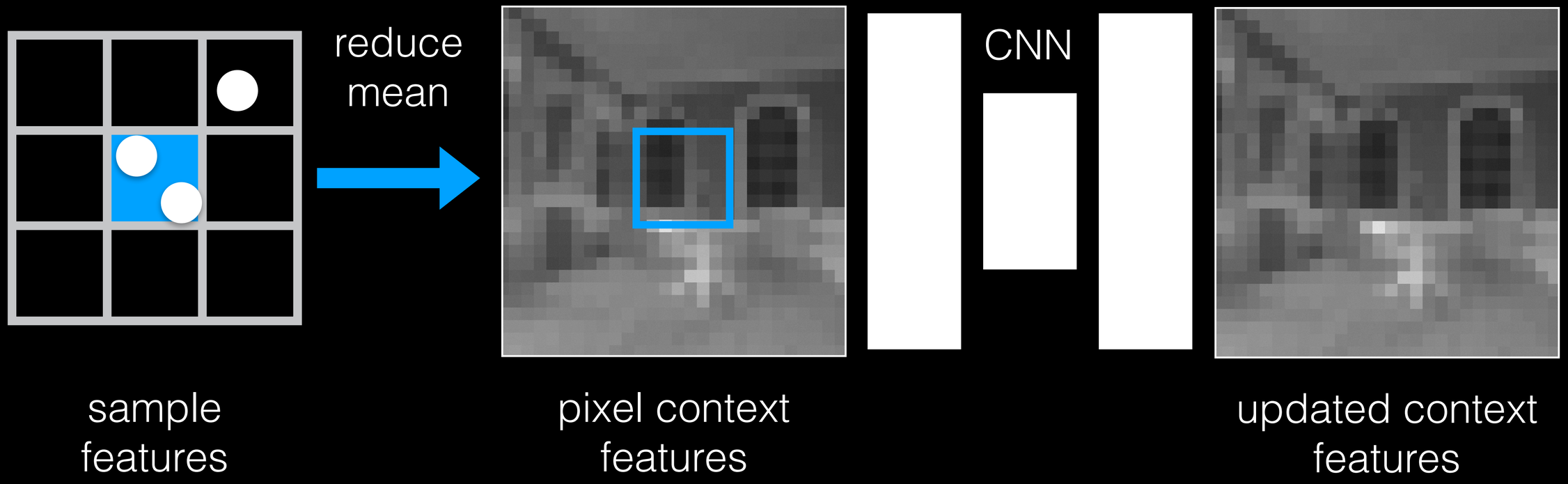


sample
features

Update pixel context



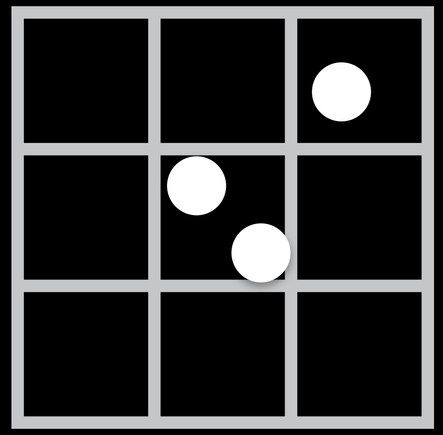
Update pixel context



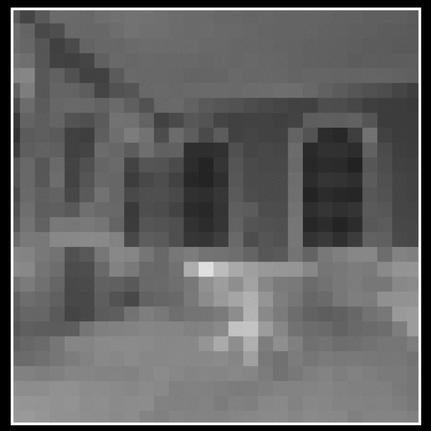
Independently process samples

conditioned on pixel context

input
sample features

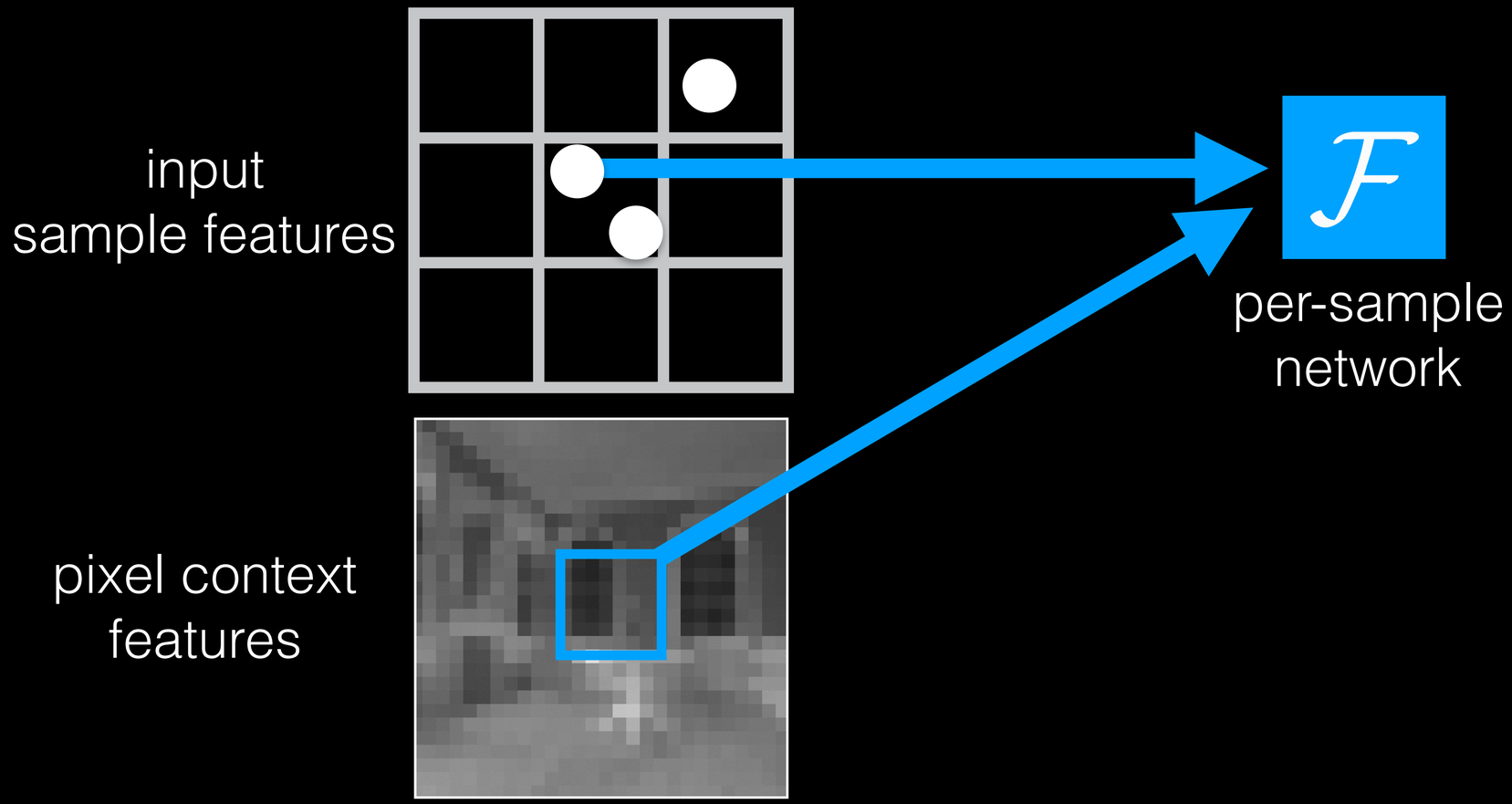


pixel context
features



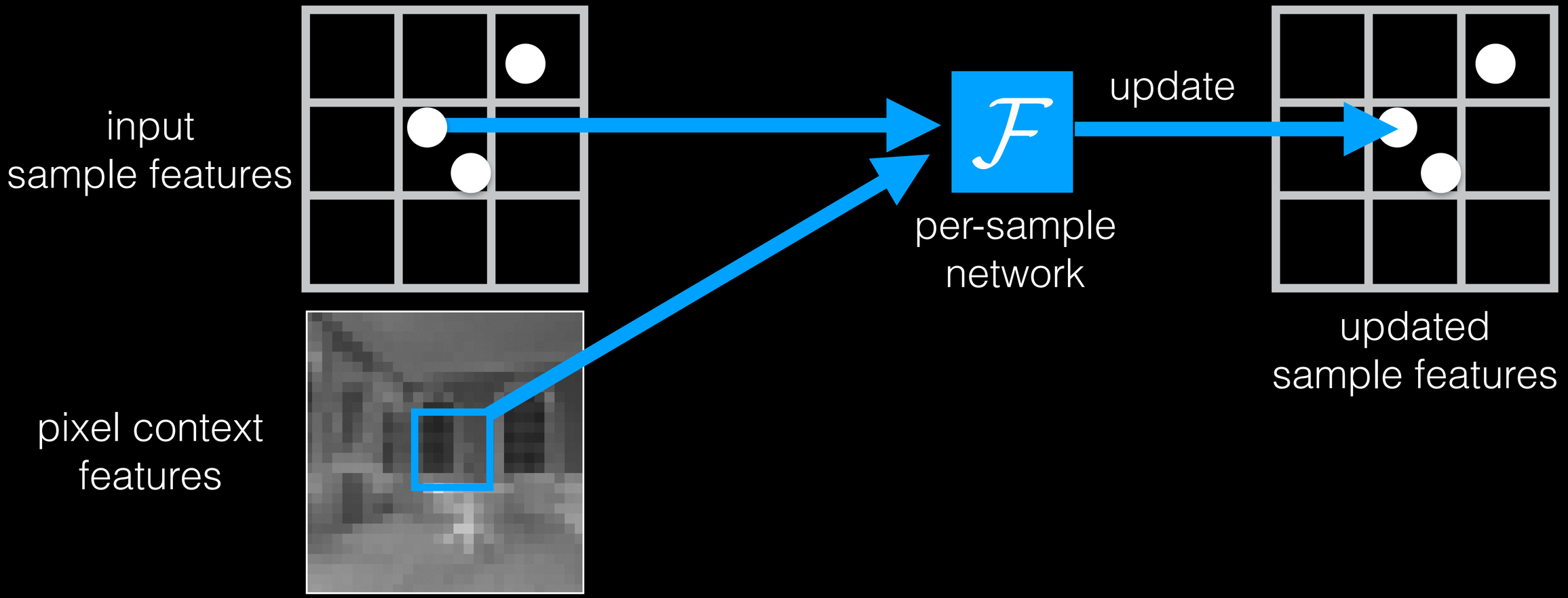
Independently process samples

conditioned on pixel context

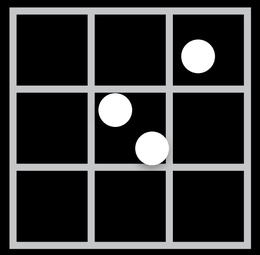


Independently process samples

conditioned on pixel context

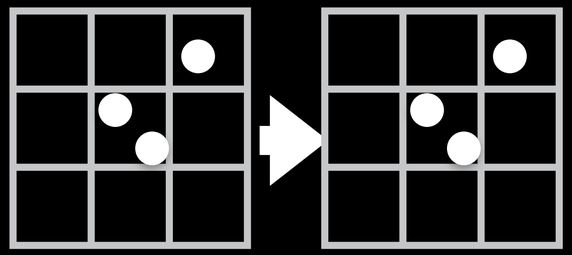


Permutation-invariant network



input
samples

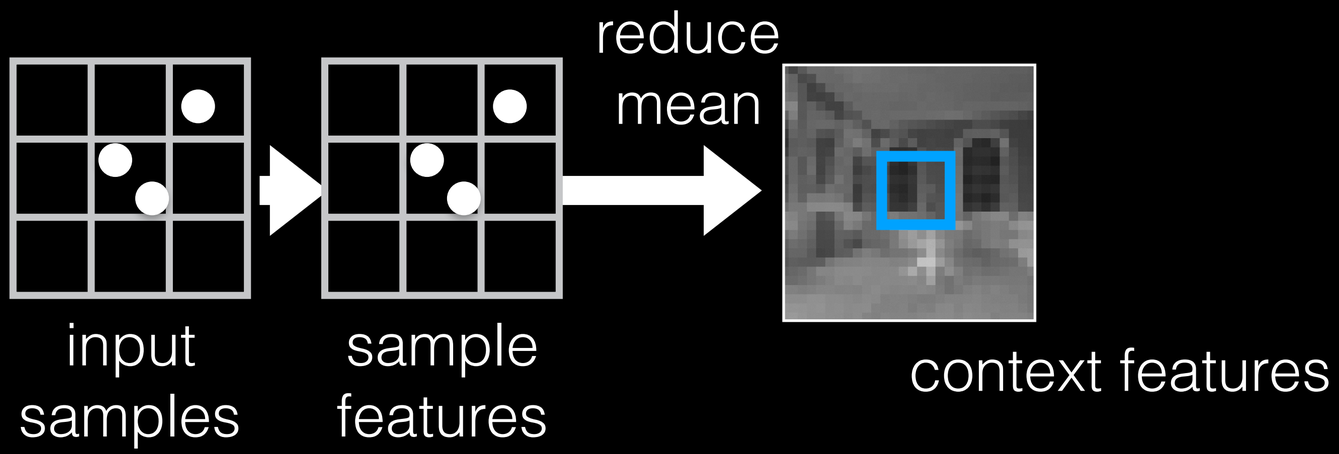
Permutation-invariant network



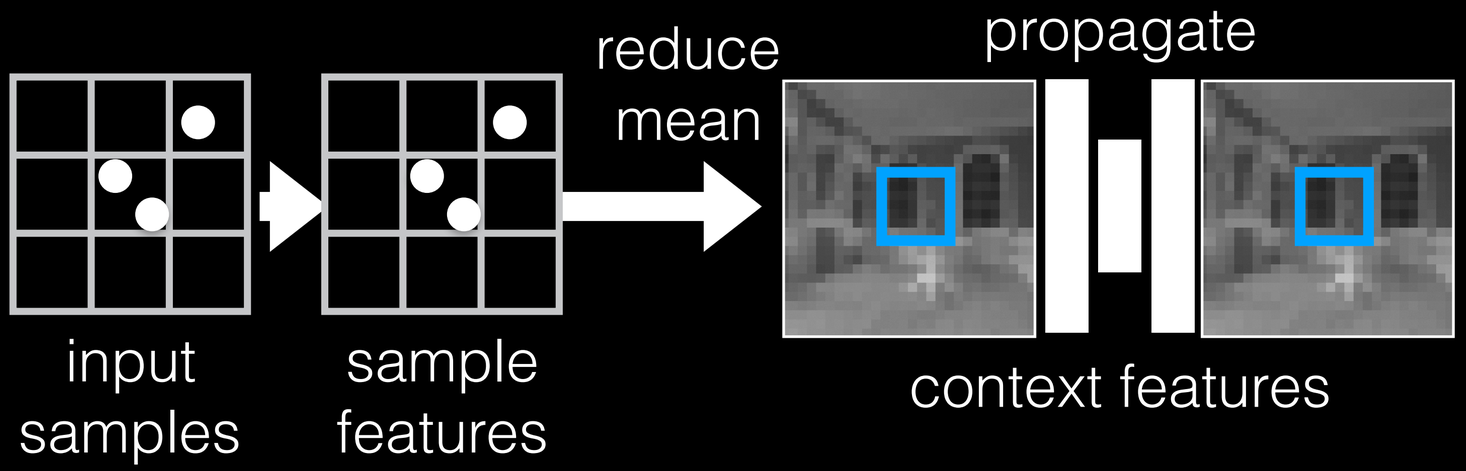
input samples

sample features

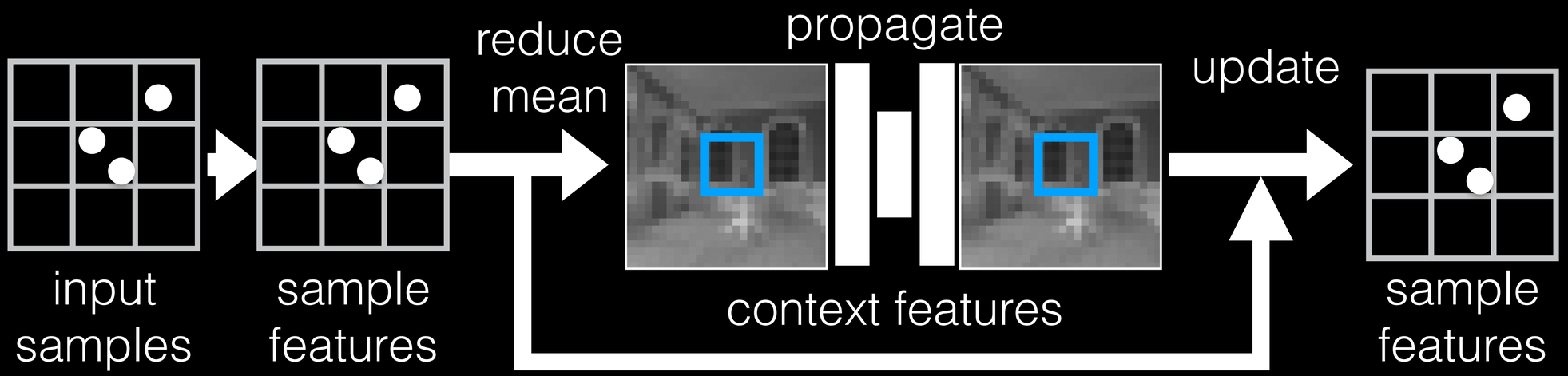
Permutation-invariant network



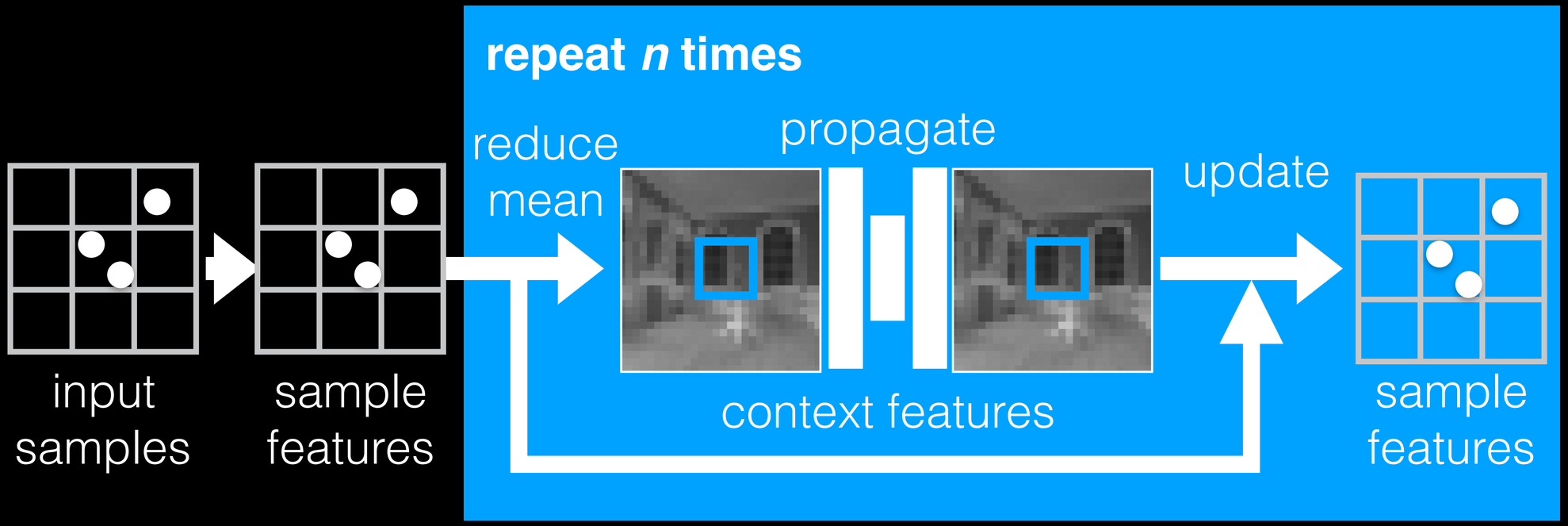
Permutation-invariant network



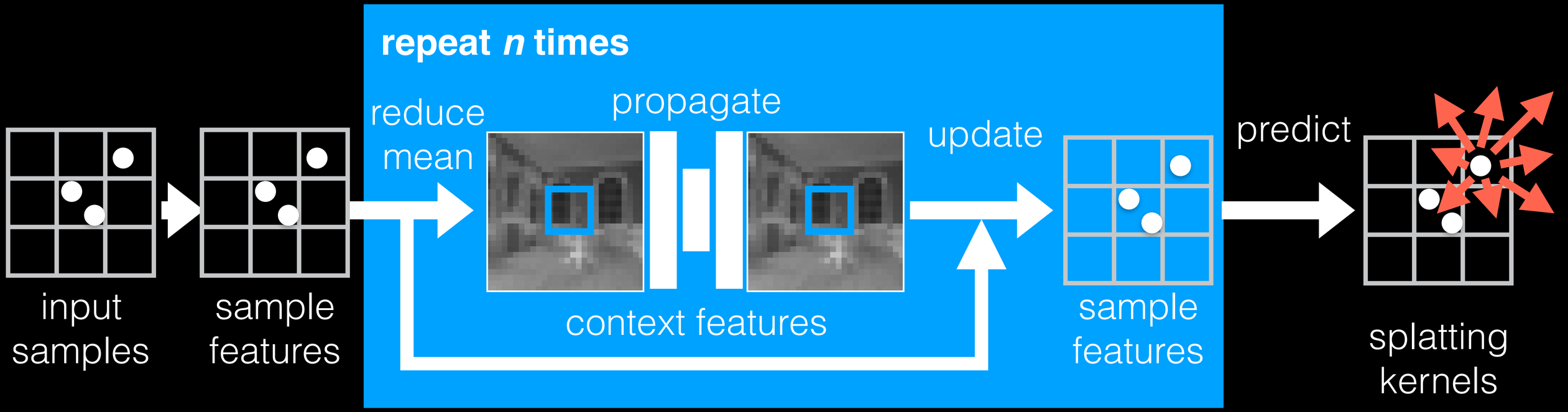
Permutation-invariant network



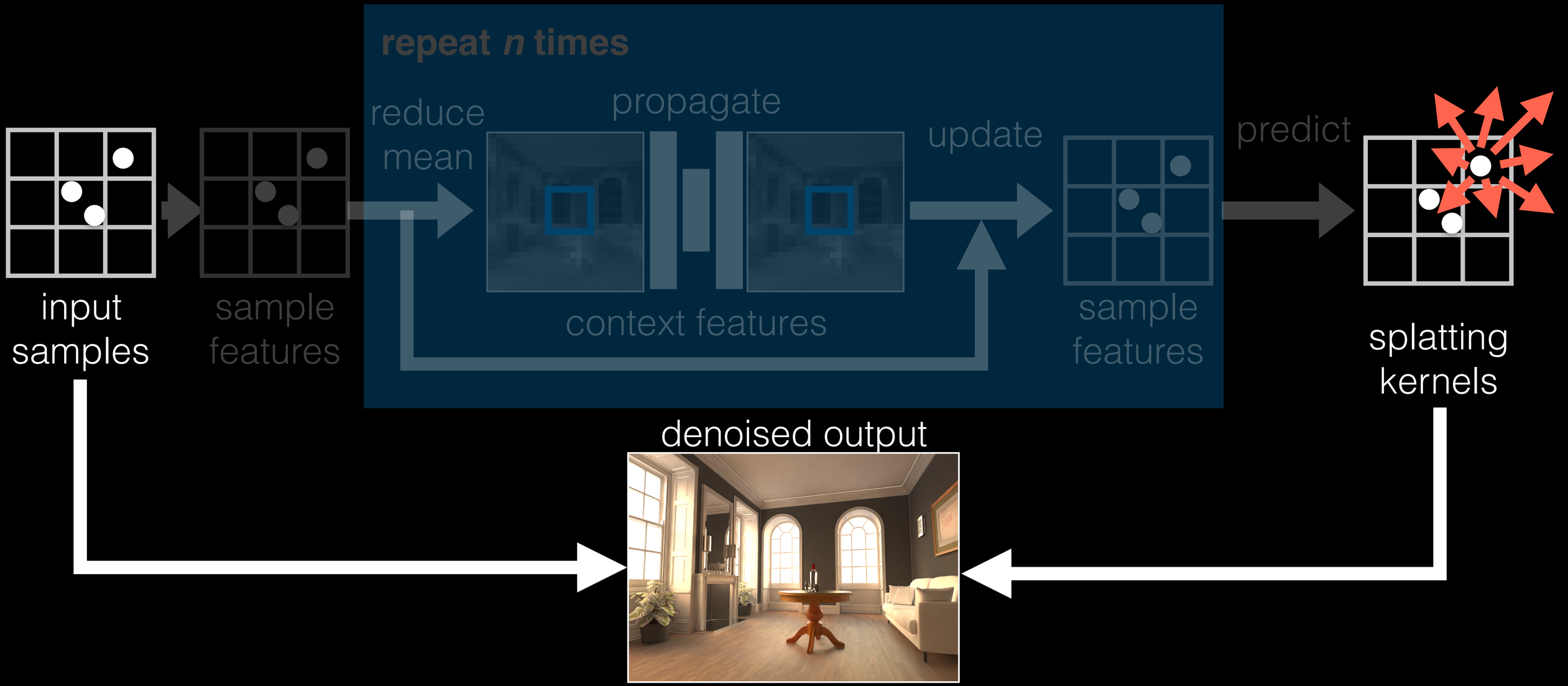
Permutation-invariant network



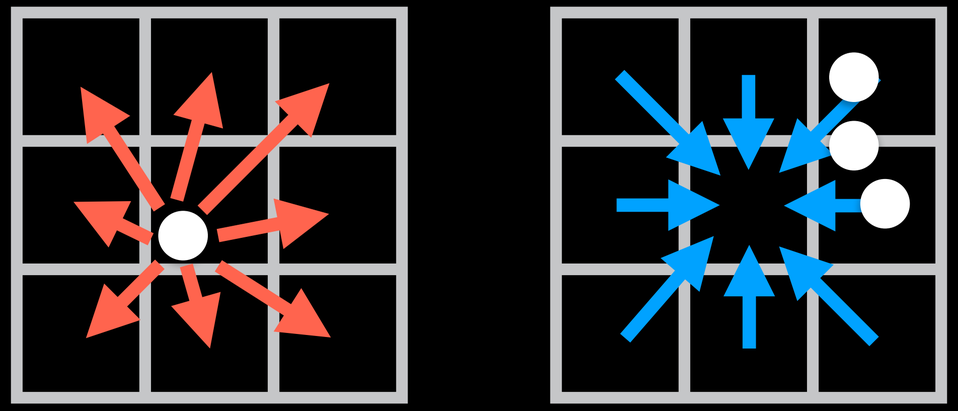
Permutation-invariant network



Permutation-invariant network

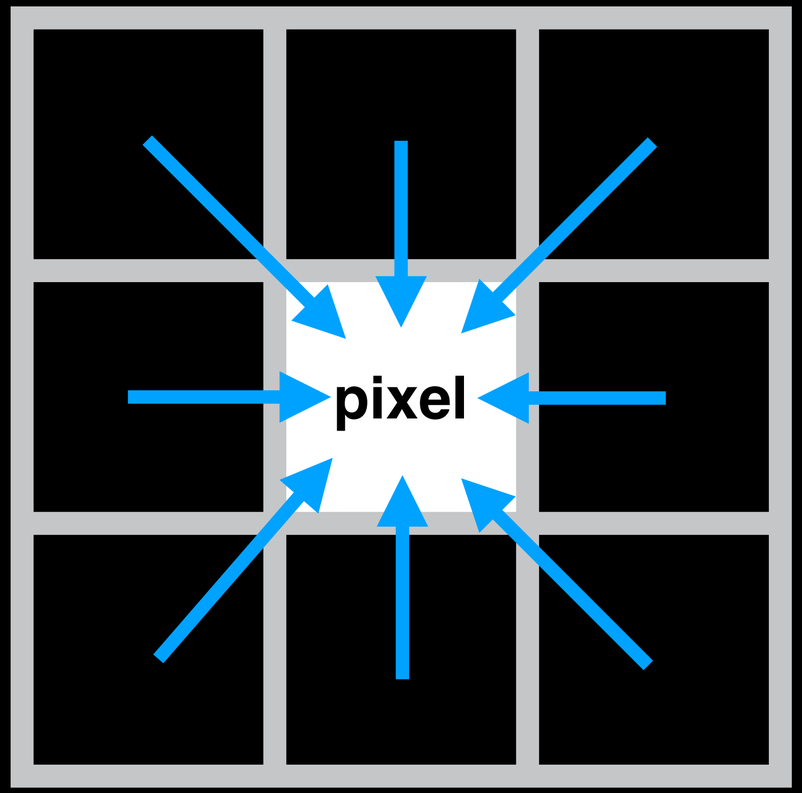


Why splatting kernels?



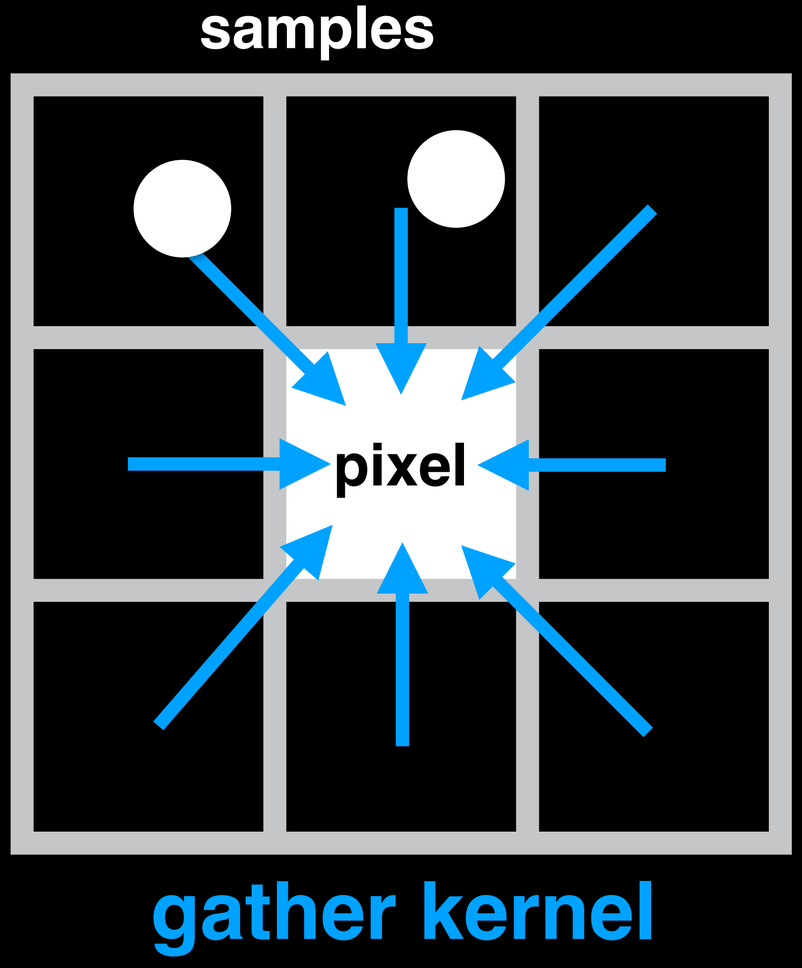
Why not *gather* kernels?

[Bako 2017; Vogels 2018] predict a kernel per pixel



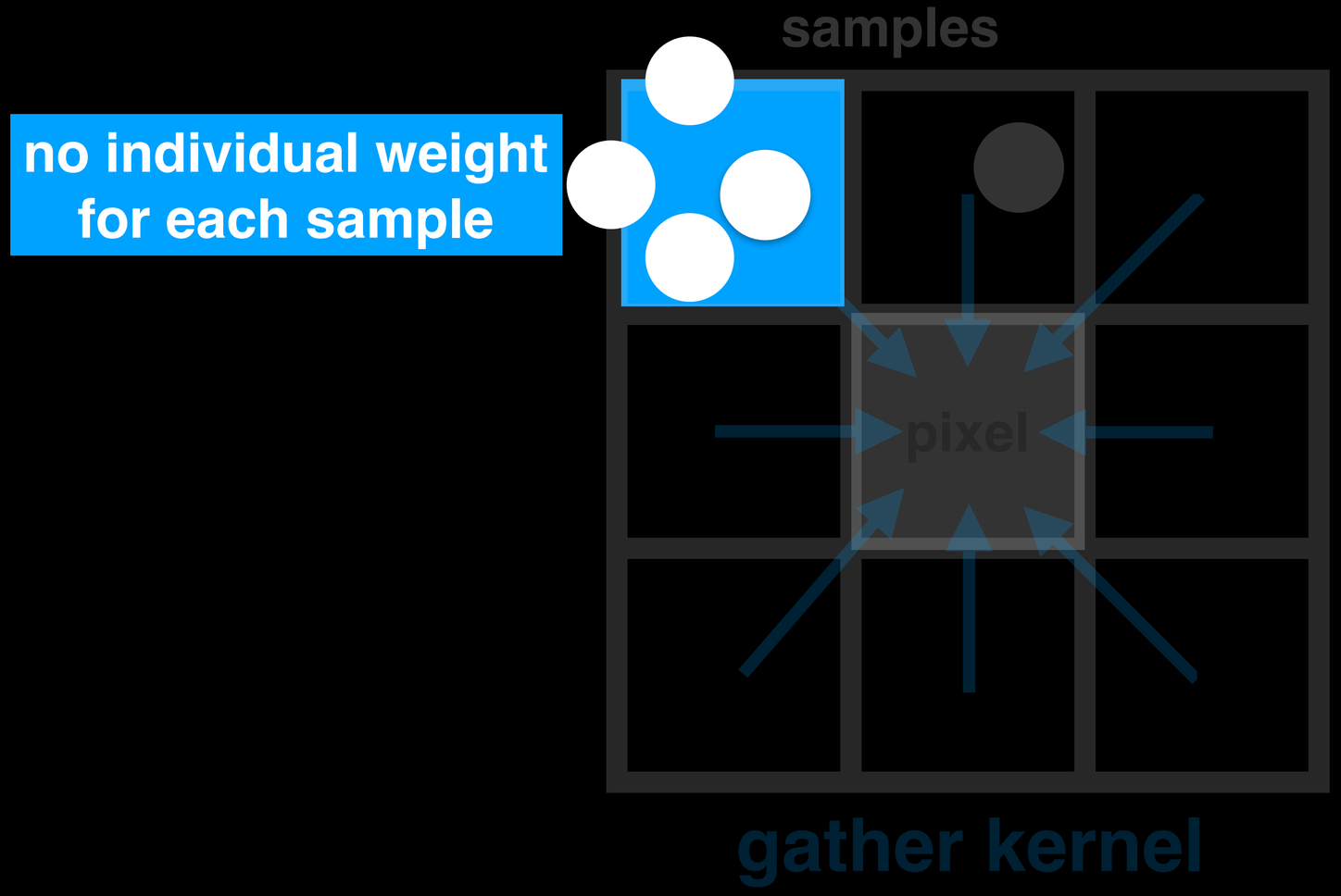
Why not *gather* kernels?

[Bako 2017; Vogels 2018] predict a kernel per pixel



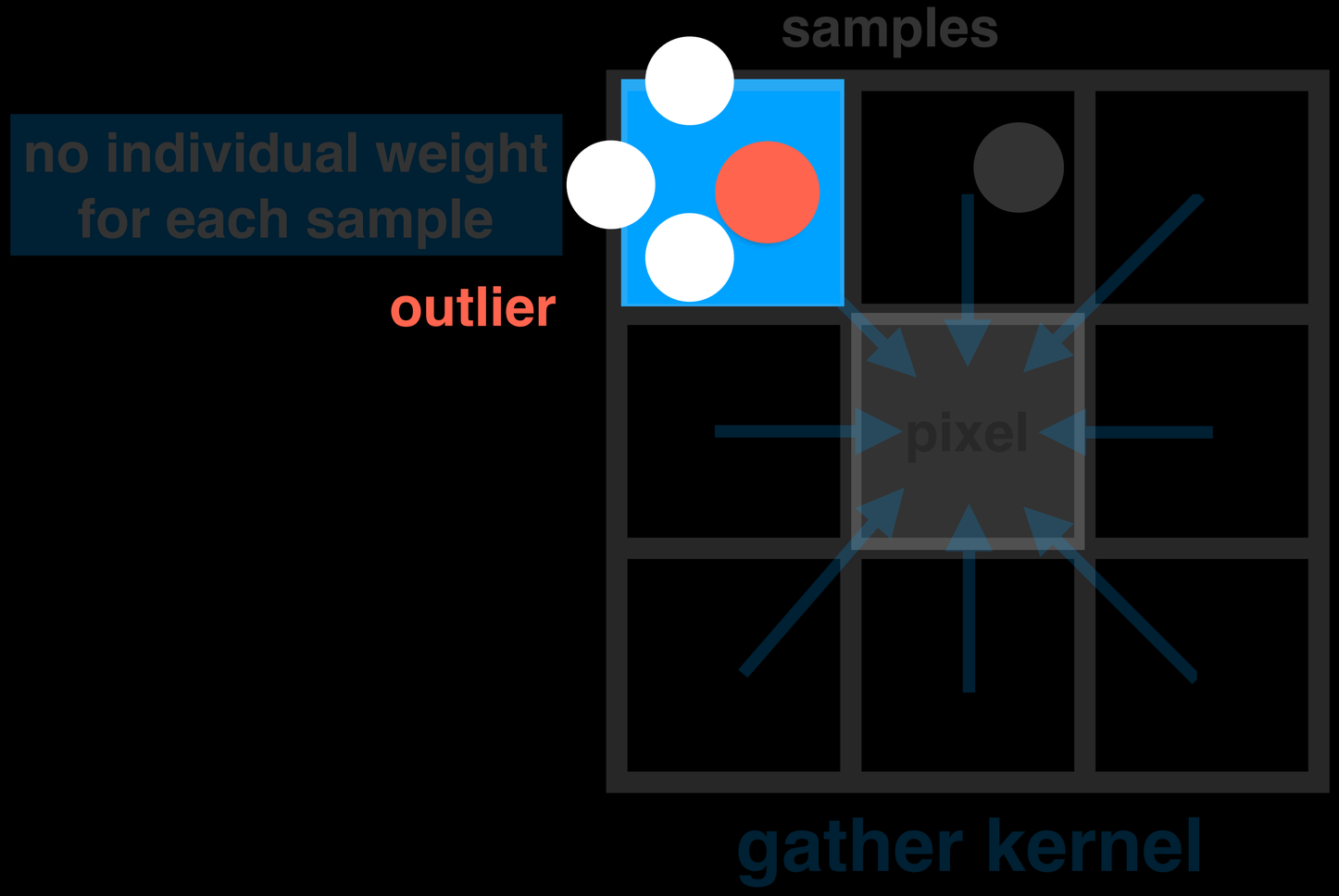
Why not *gather* kernels?

[Bako 2017; Vogels 2018] predict a kernel per pixel

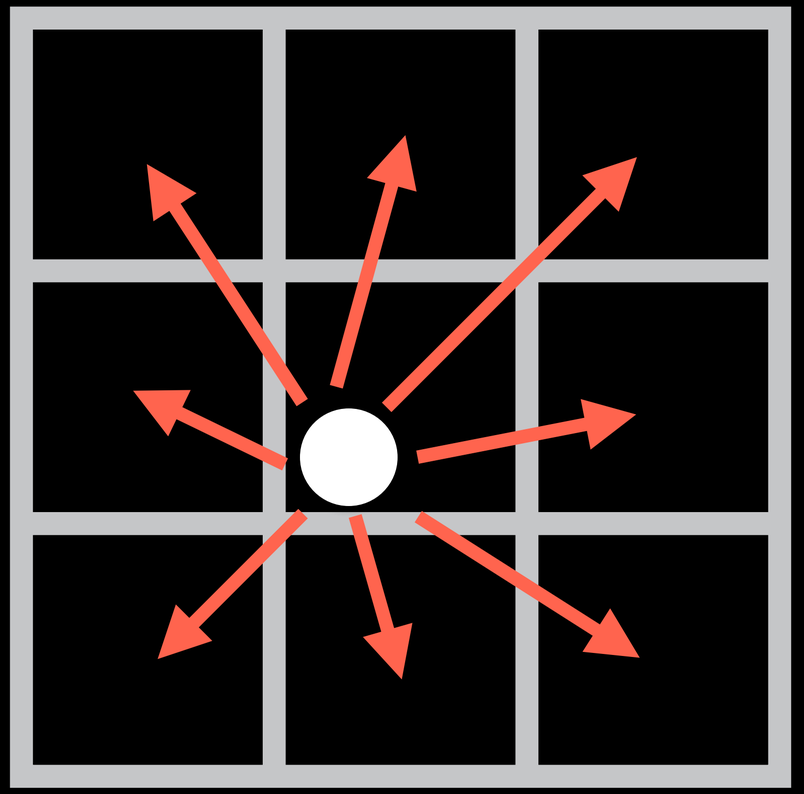


Why not *gather* kernels?

[Bako 2017; Vogels 2018] predict a kernel per pixel

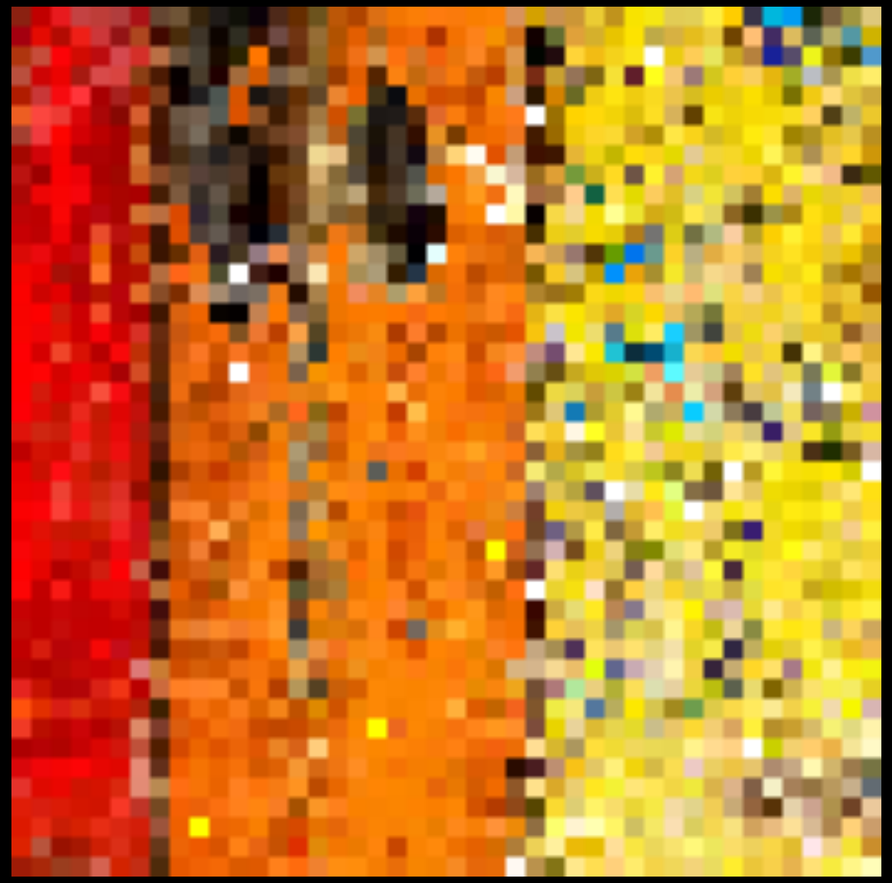


Splatting kernels: more natural for samples



splatting kernel

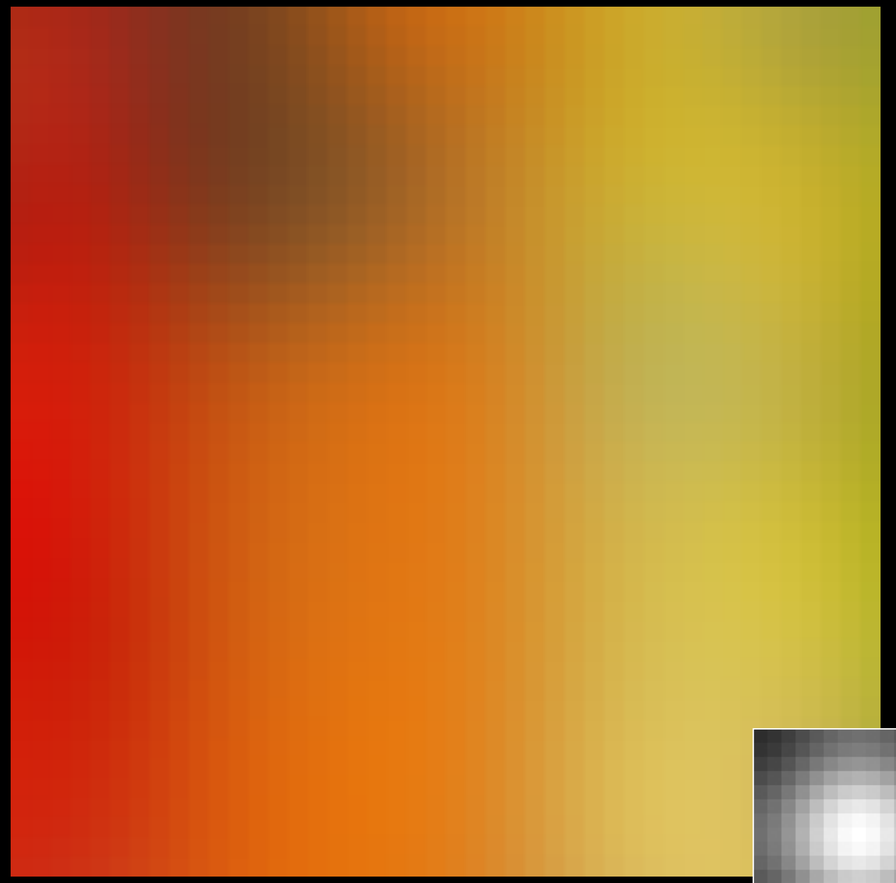
Toy example: learn Gaussian blur



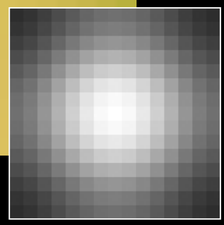
input

kernel-predicting network

$\mathcal{F}$

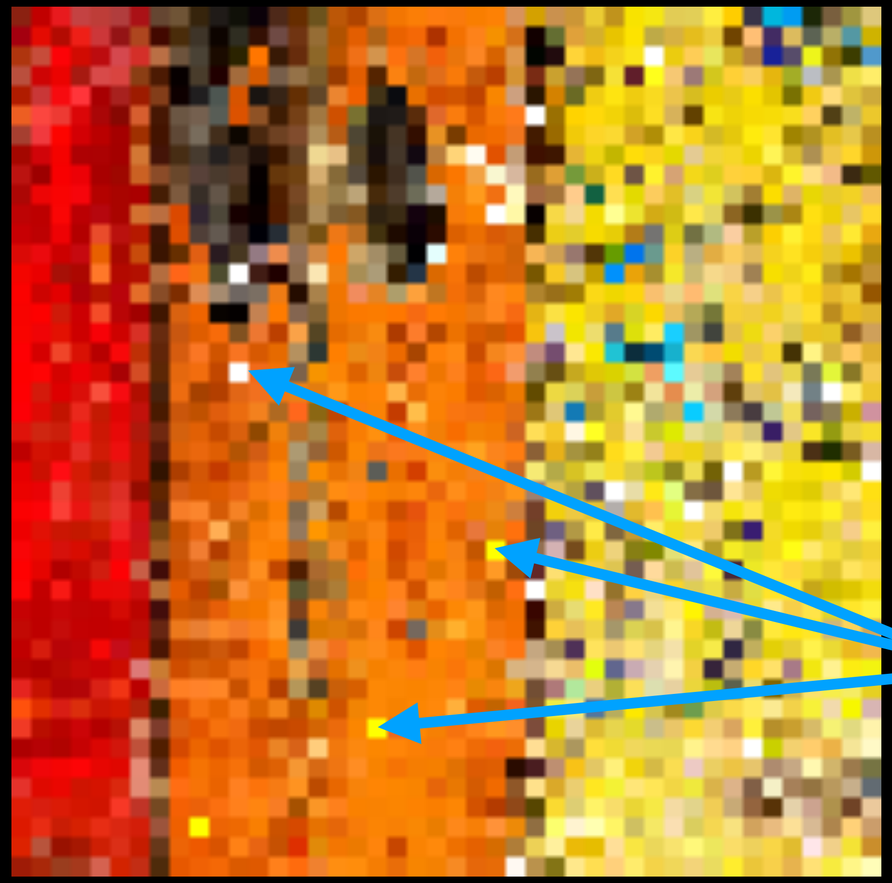


target: Gaussian blur



Toy example: learn Gaussian blur

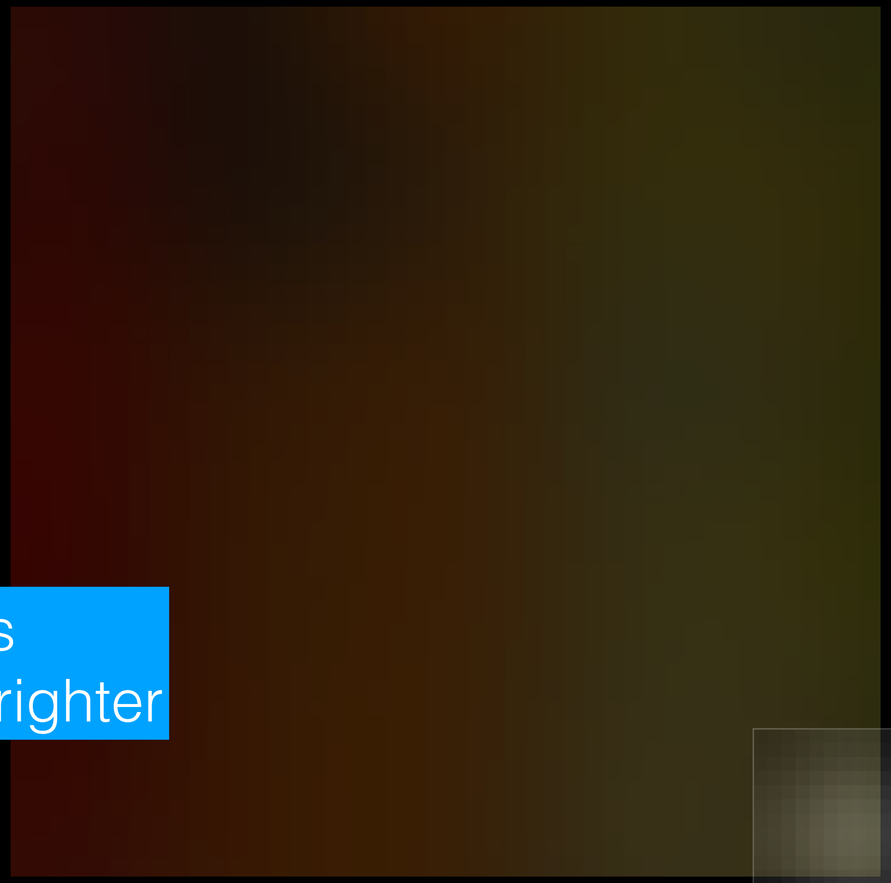
with outliers



kernel-predicting network

$\mathcal{F}$

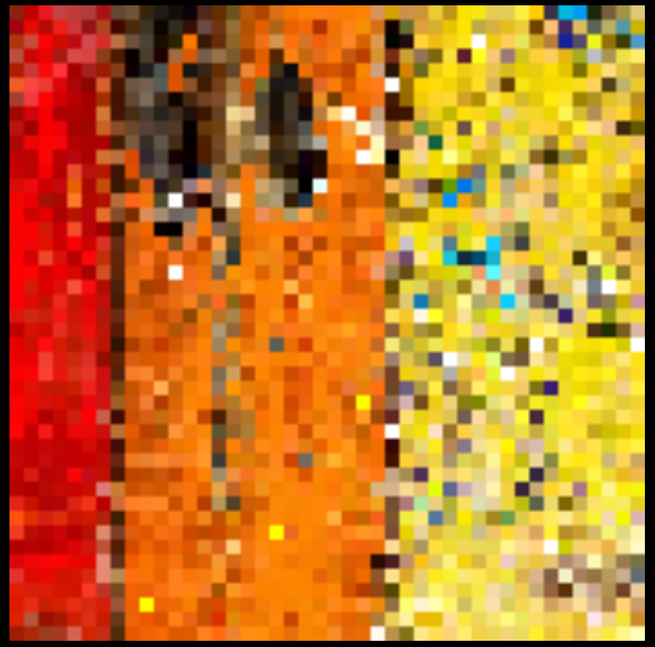
1 % outliers
100x — 1000x brighter



input

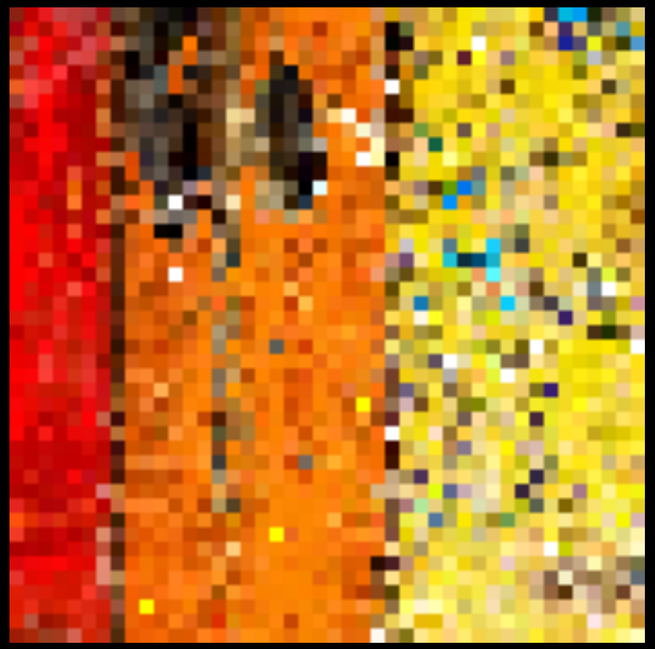
target: Gaussian blur

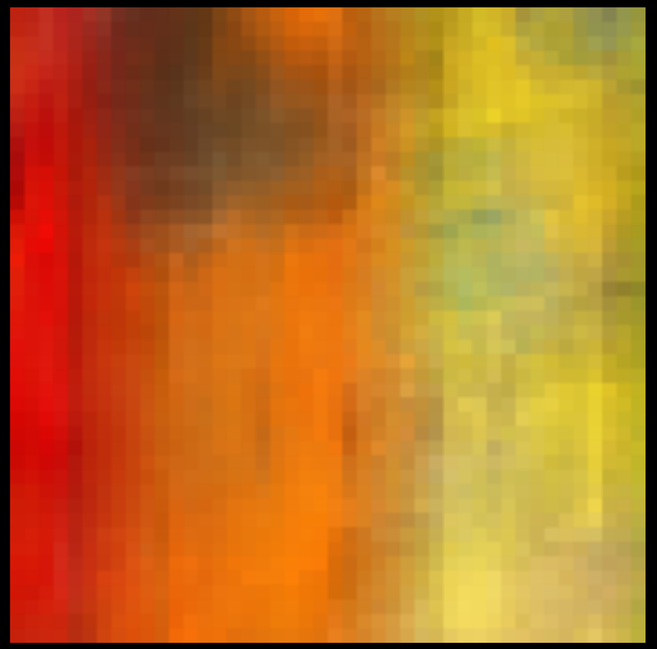
input



target

input

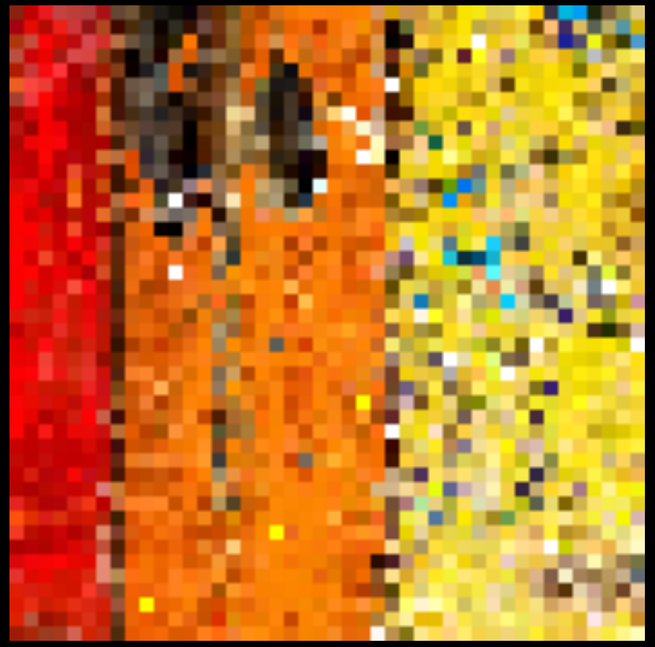


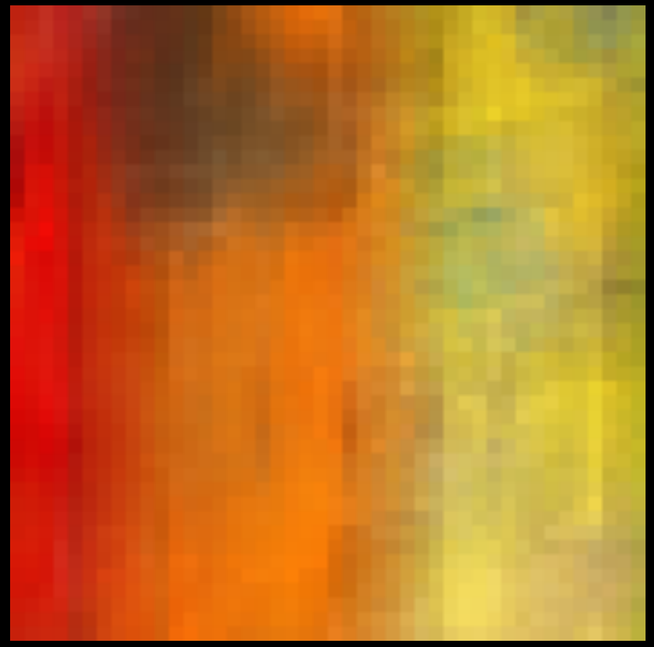


network with
gather kernels
~[Bako 2017]

target

input



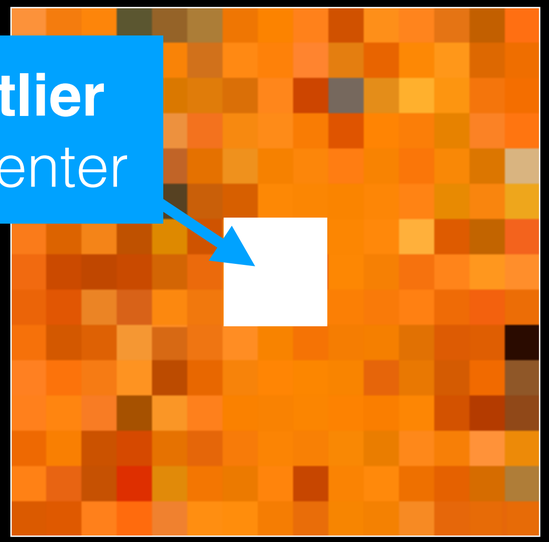


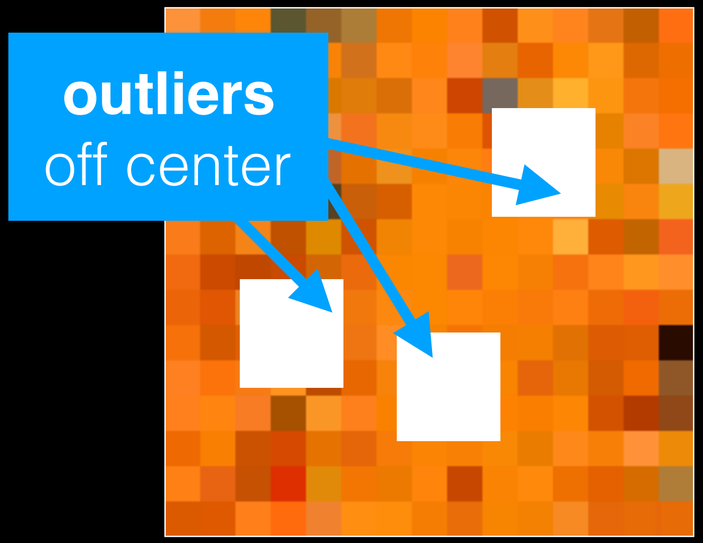
network with
gather kernels
~[Bako 2017]

target

network with
splatting kernels
(ours)

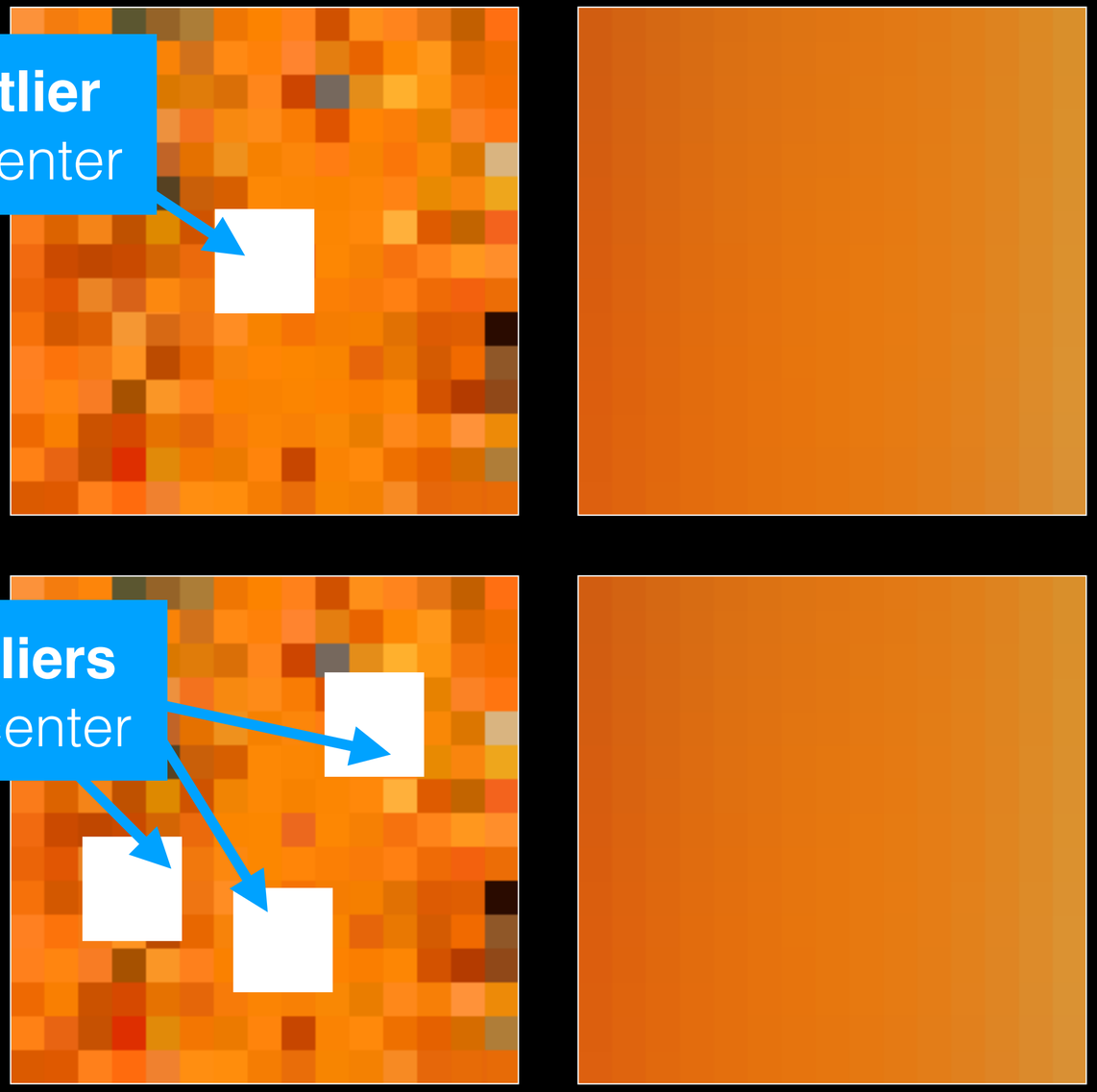
input





input

target

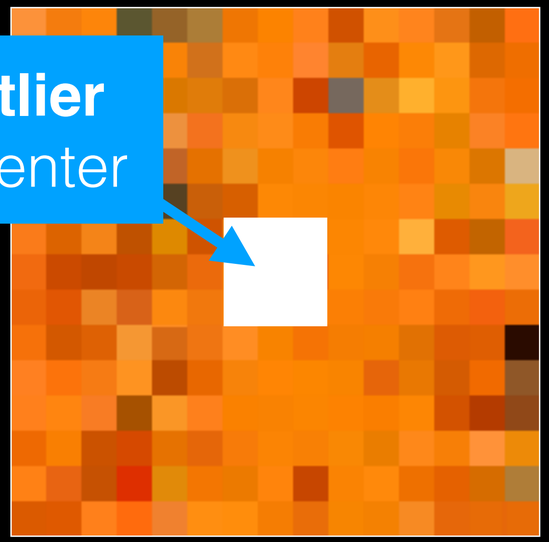


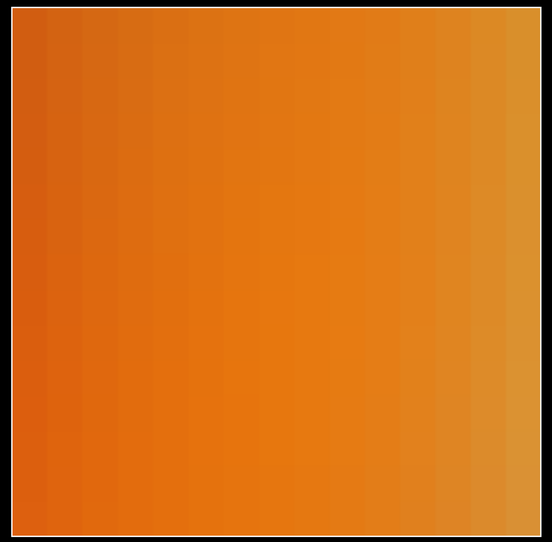
input

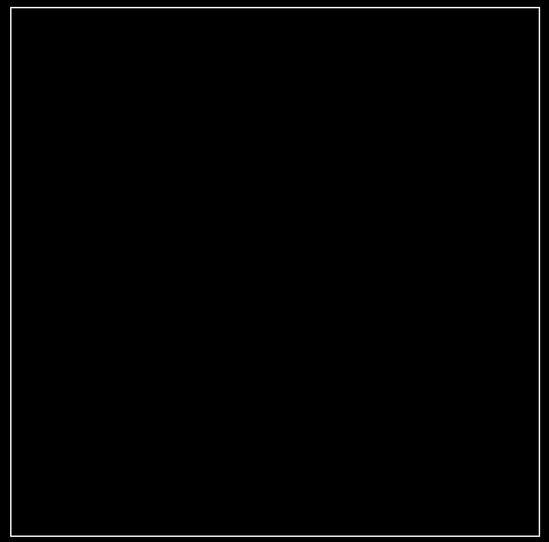
target

expected
splatting kernel

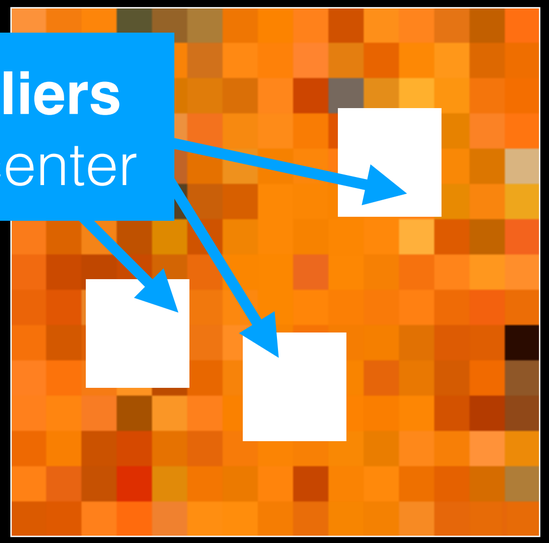
outlier
in center

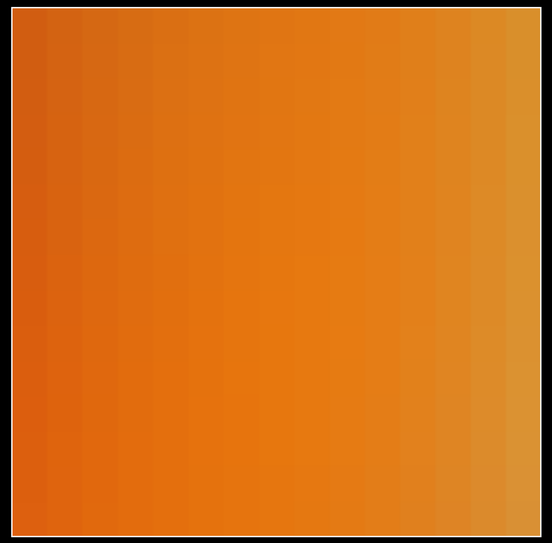






outliers
off center



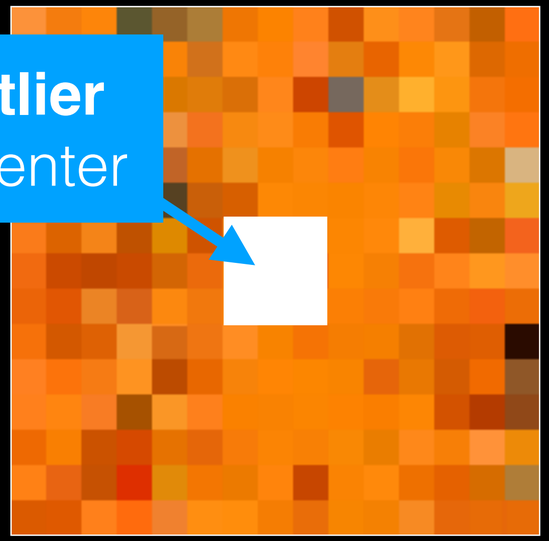


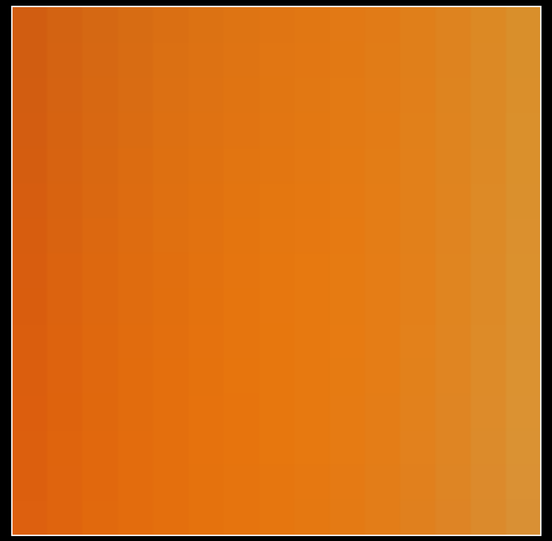
input

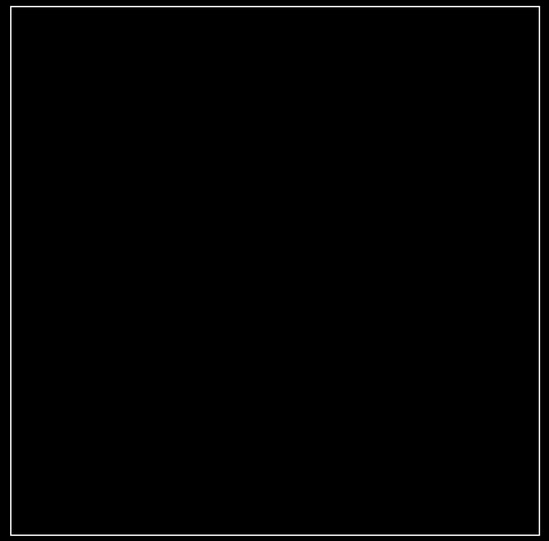
target

expected
splatting kernel

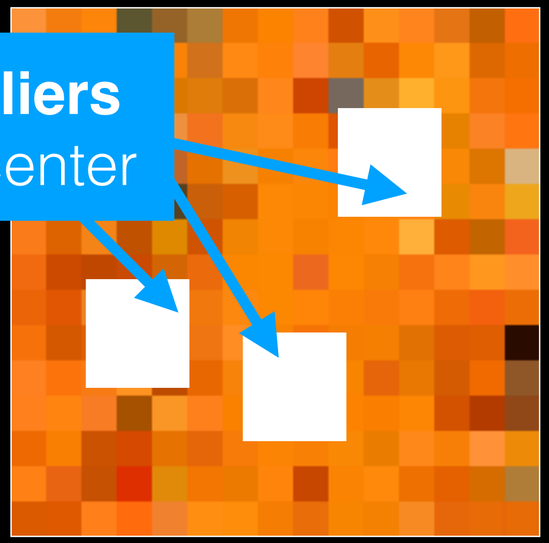
outlier
in center

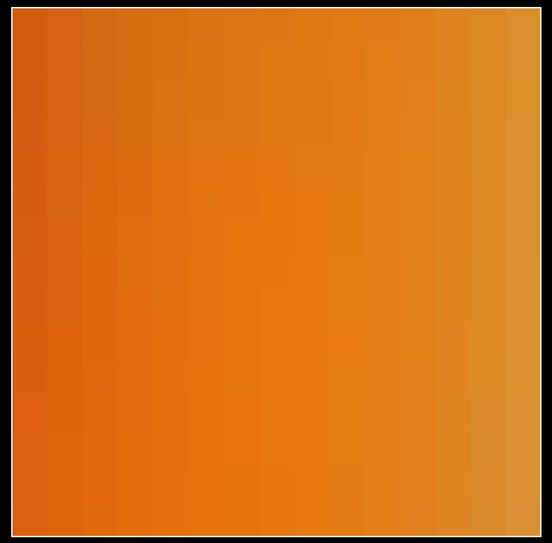


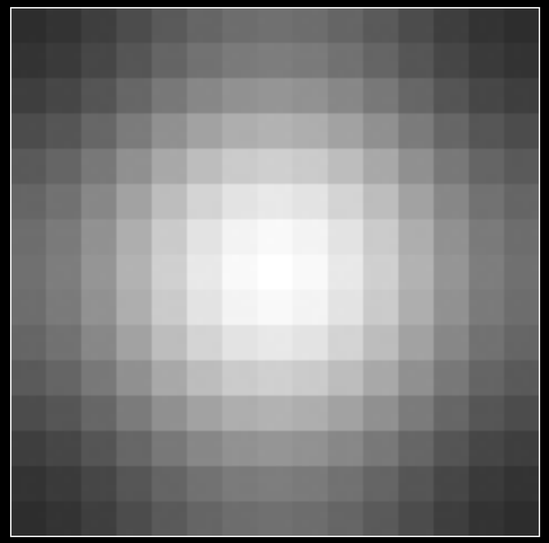




outliers
off center







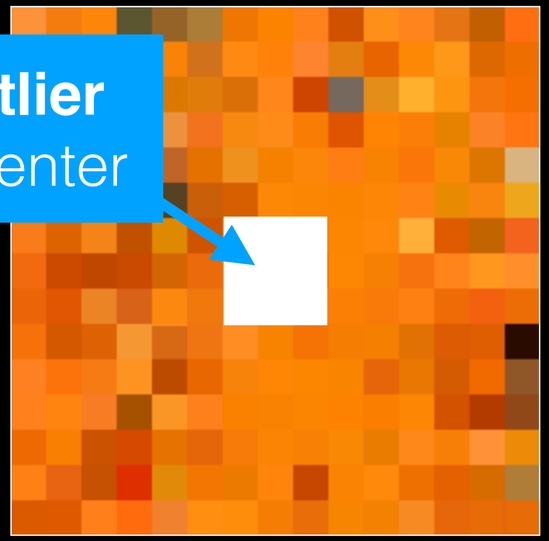
input

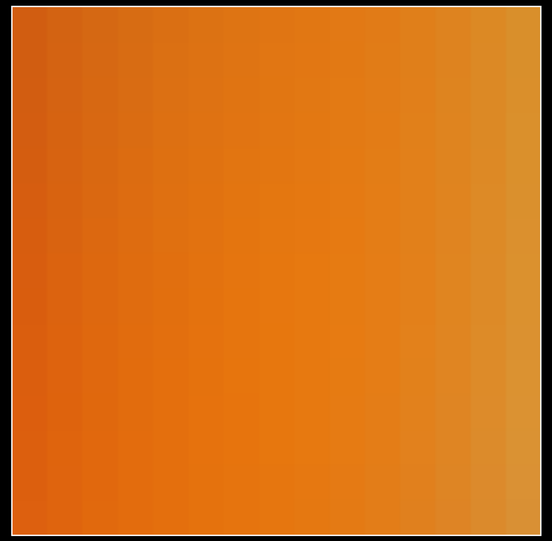
target

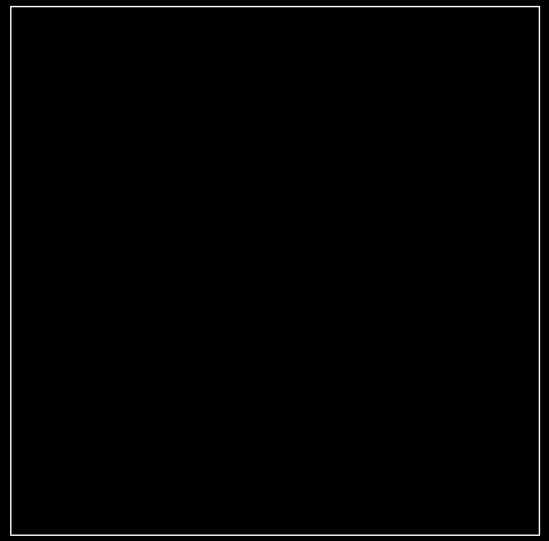
expected
splatting kernel

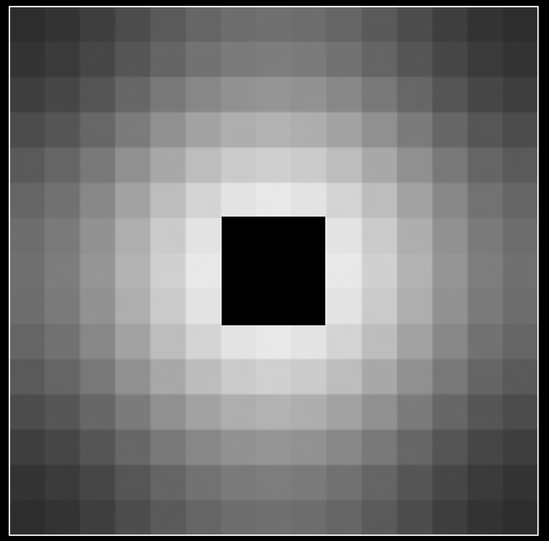
expected
gather kernel

outlier
in center

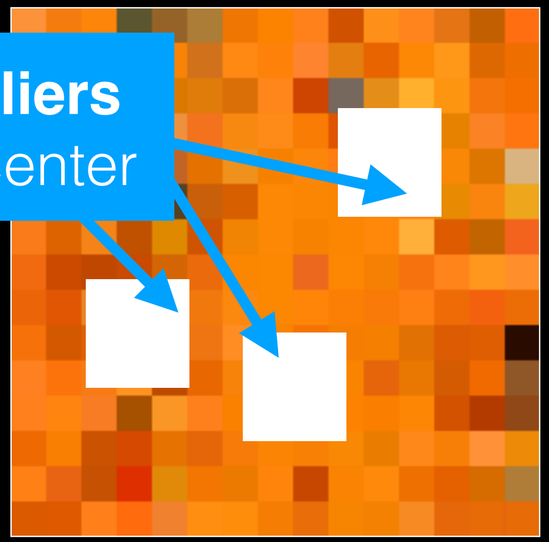


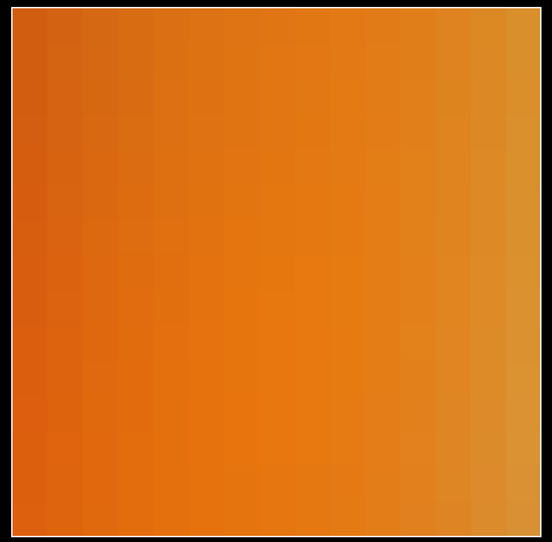


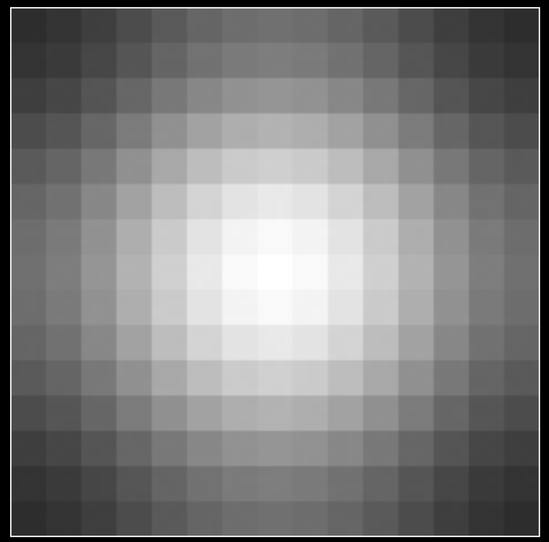




outliers
off center







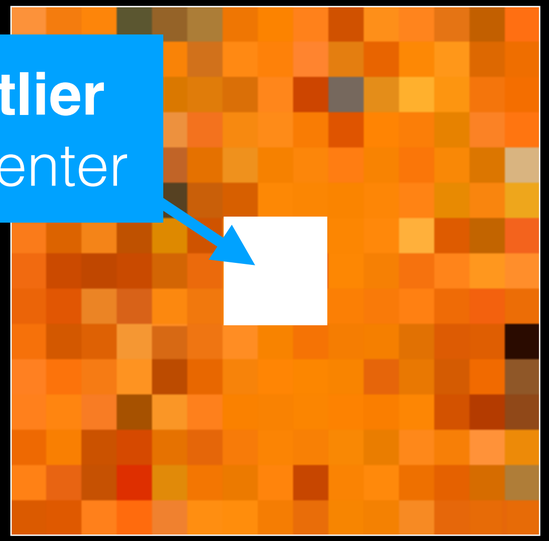
input

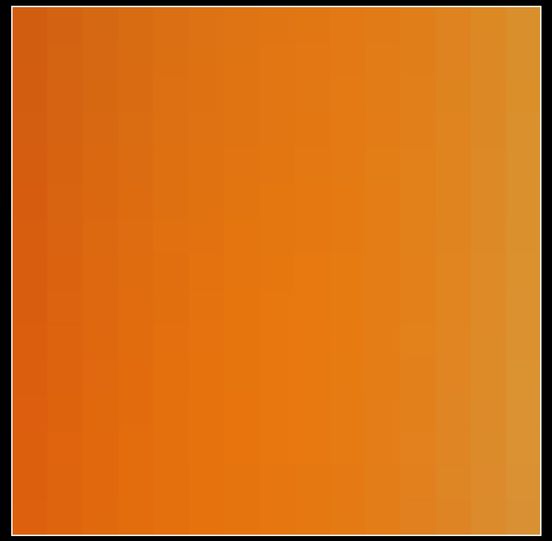
target

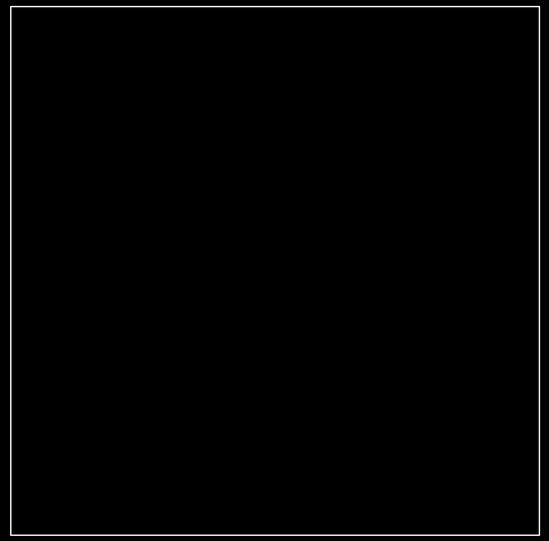
expected
splatting kernel

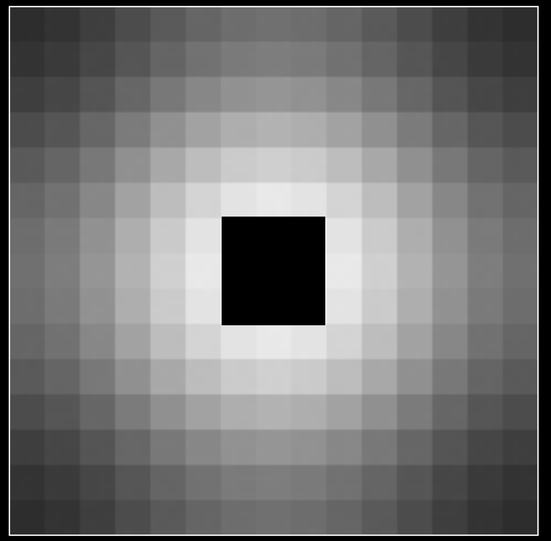
expected
gather kernel

outlier
in center

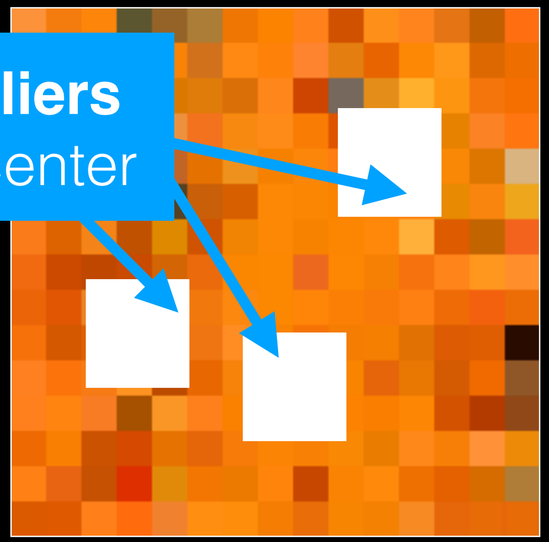


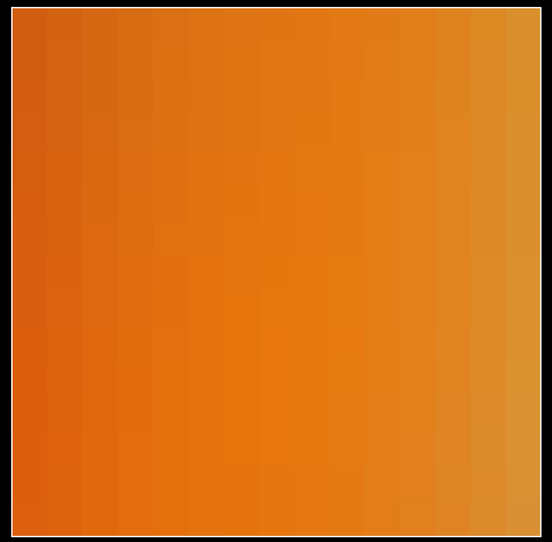


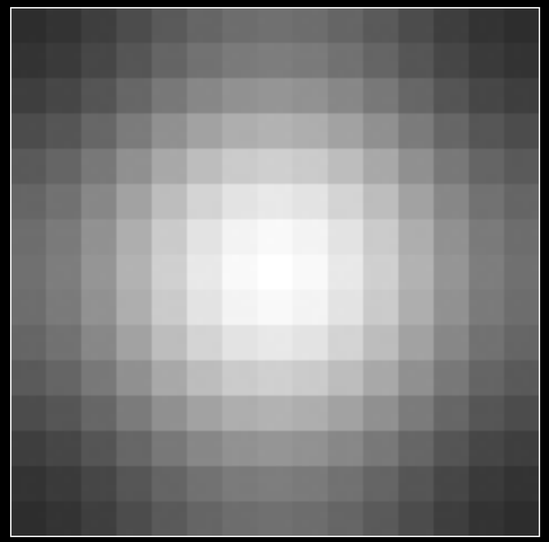


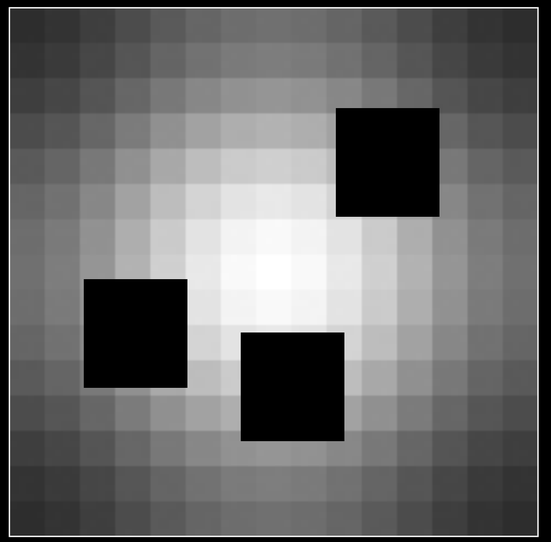


outliers
off center









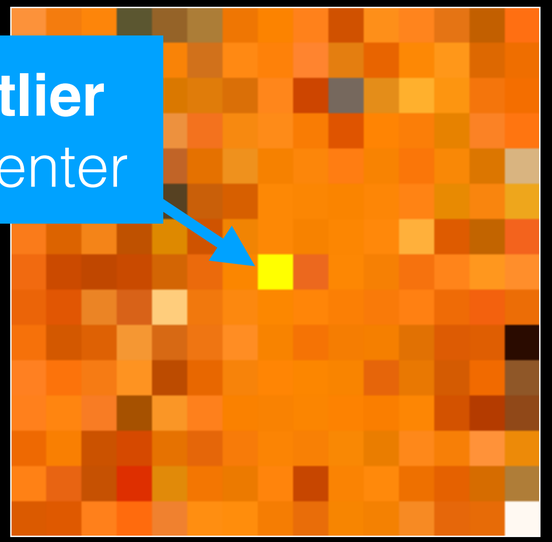
input

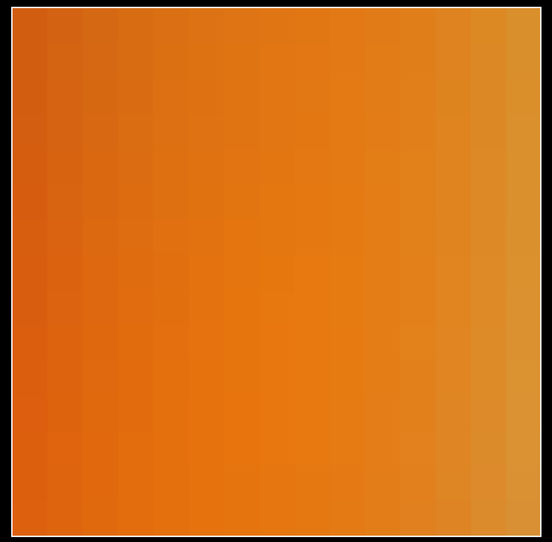
target

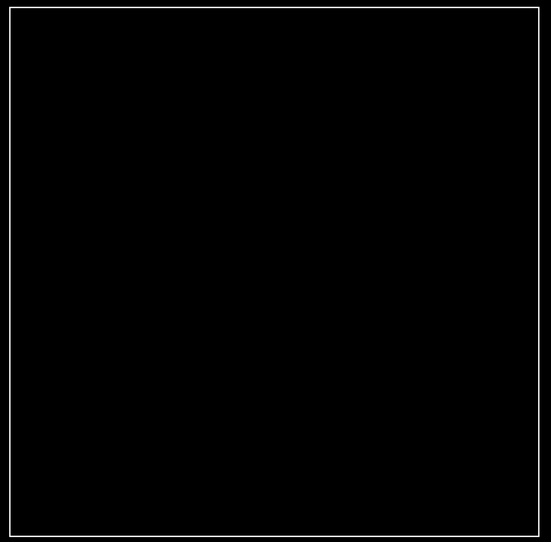
splatting kernel

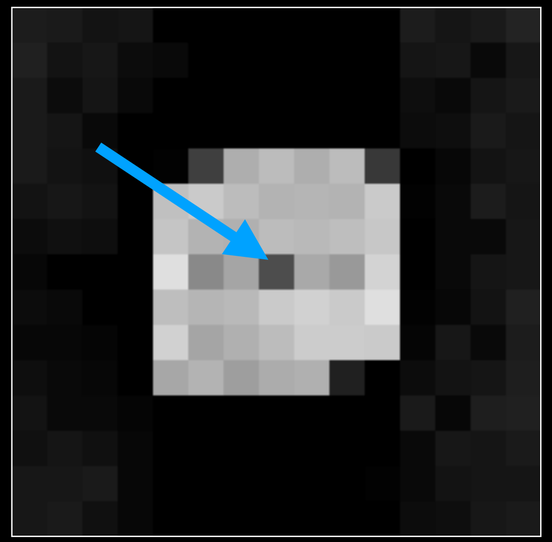
gather kernel

outlier
in center

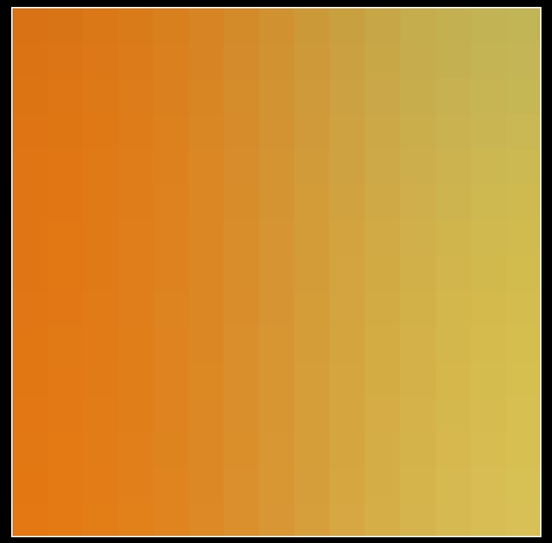


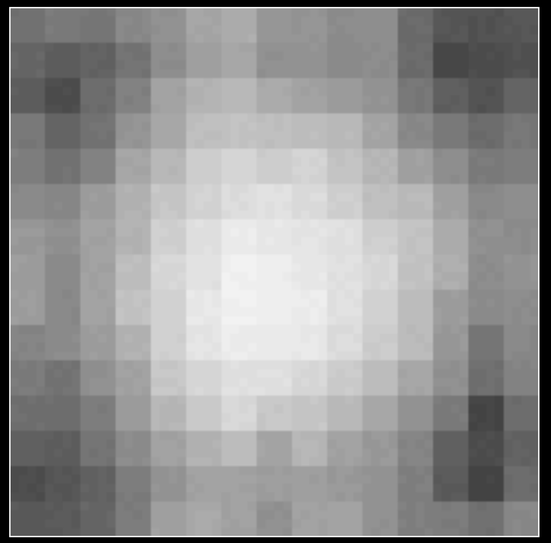


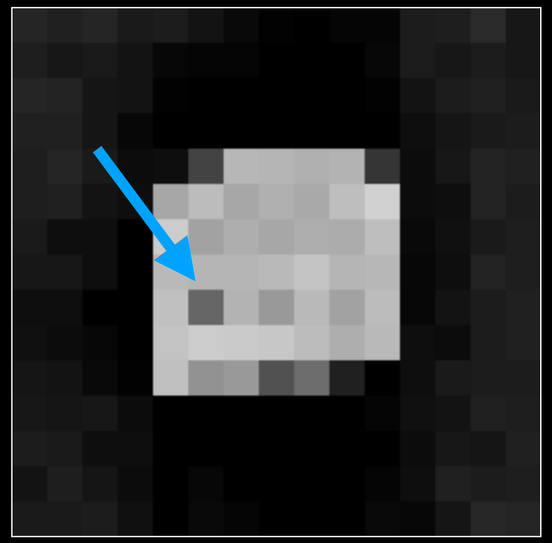




outliers
off center





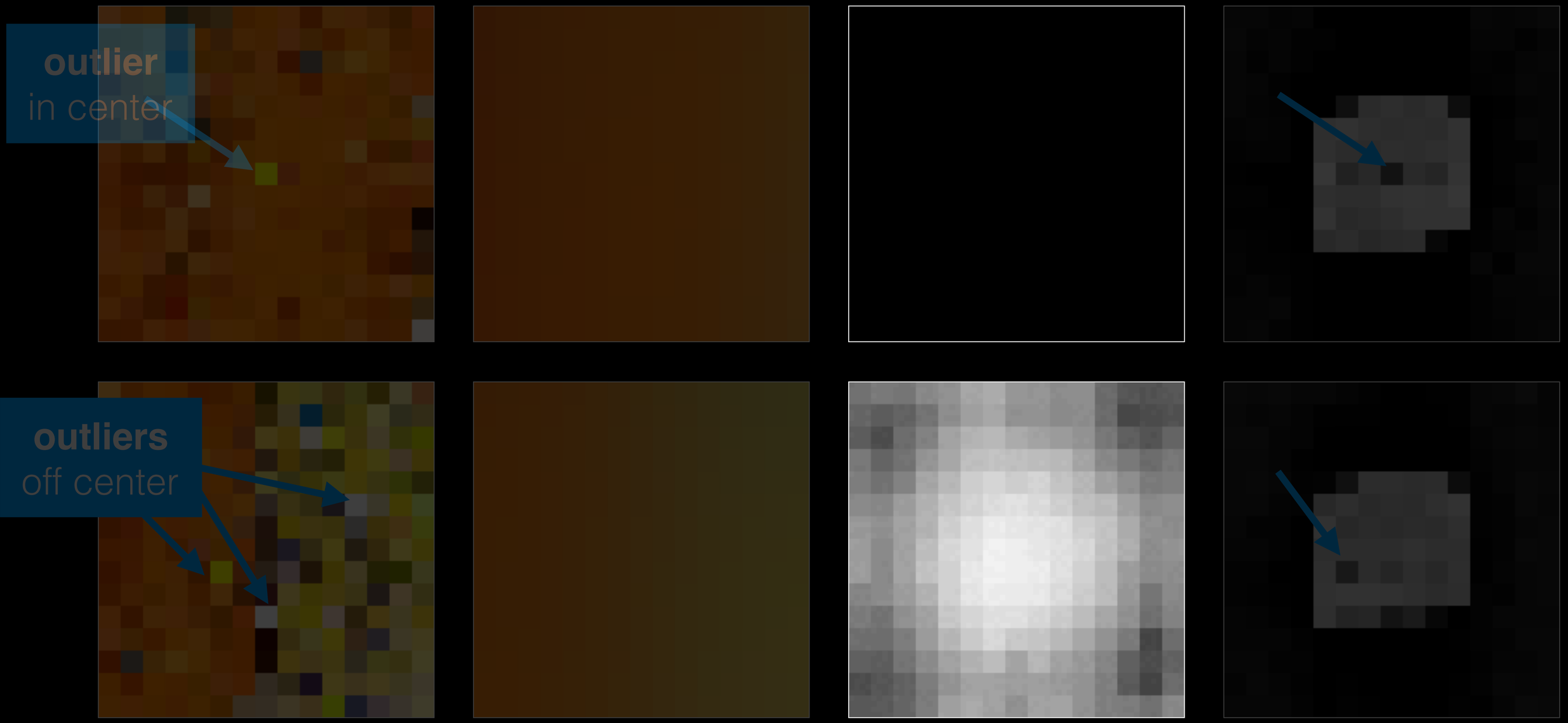


input

target

splatting kernel

gather kernel



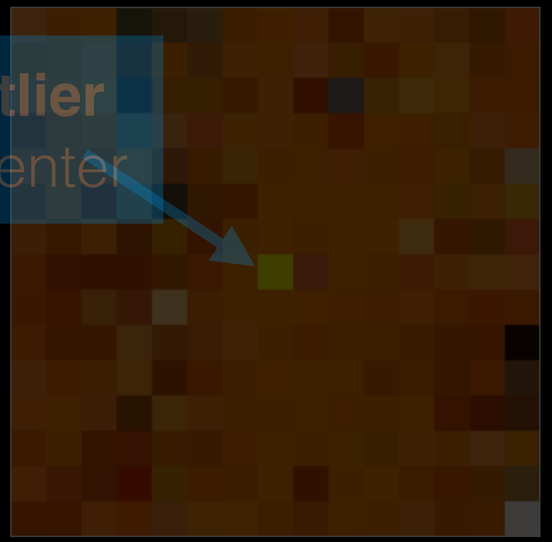
input

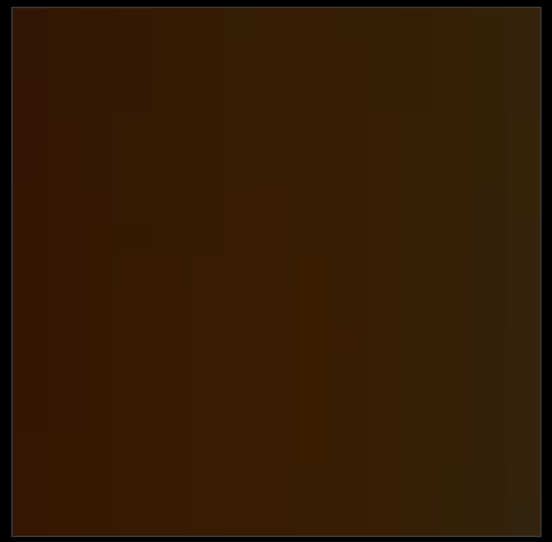
target

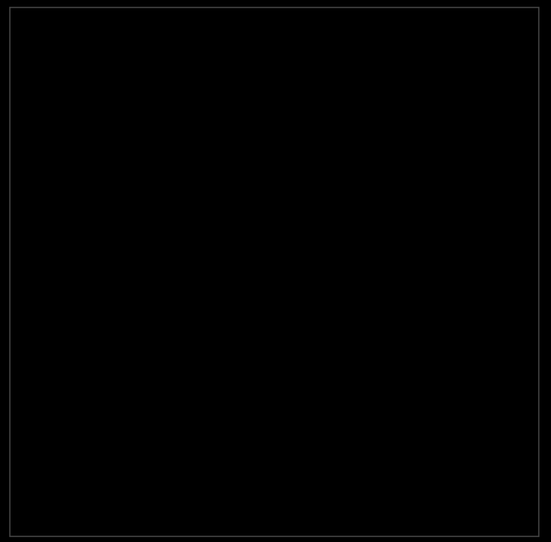
splatting kernel

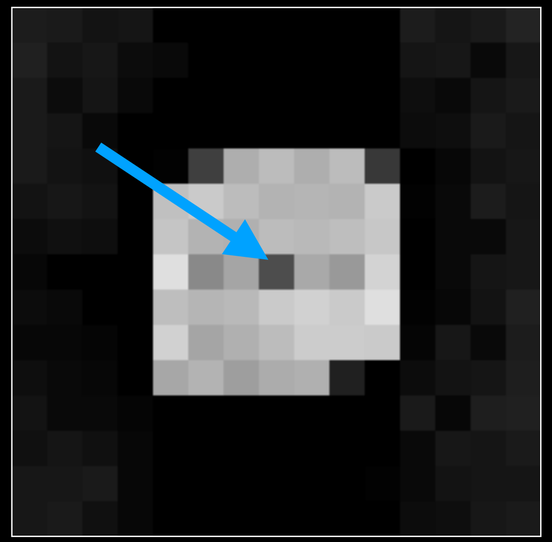
gather kernel

outlier
in center

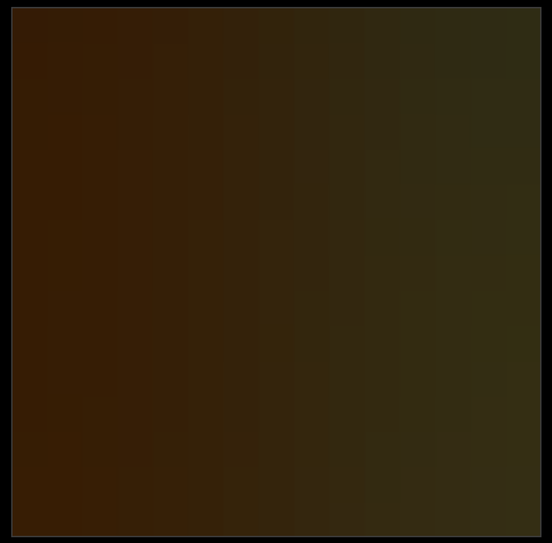


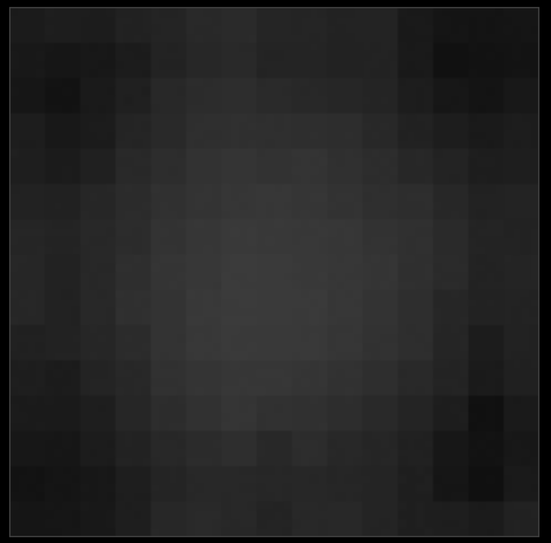


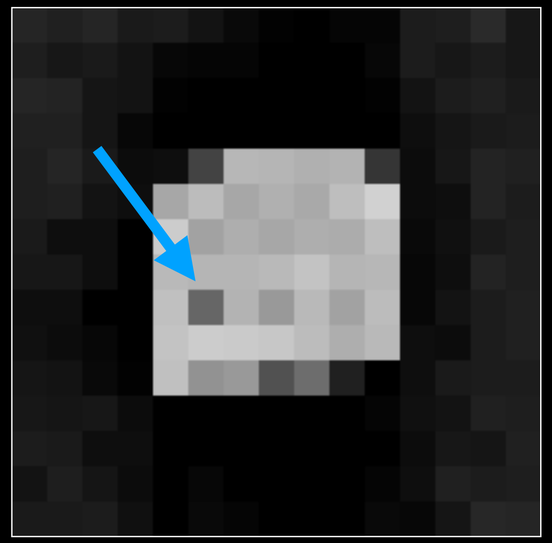


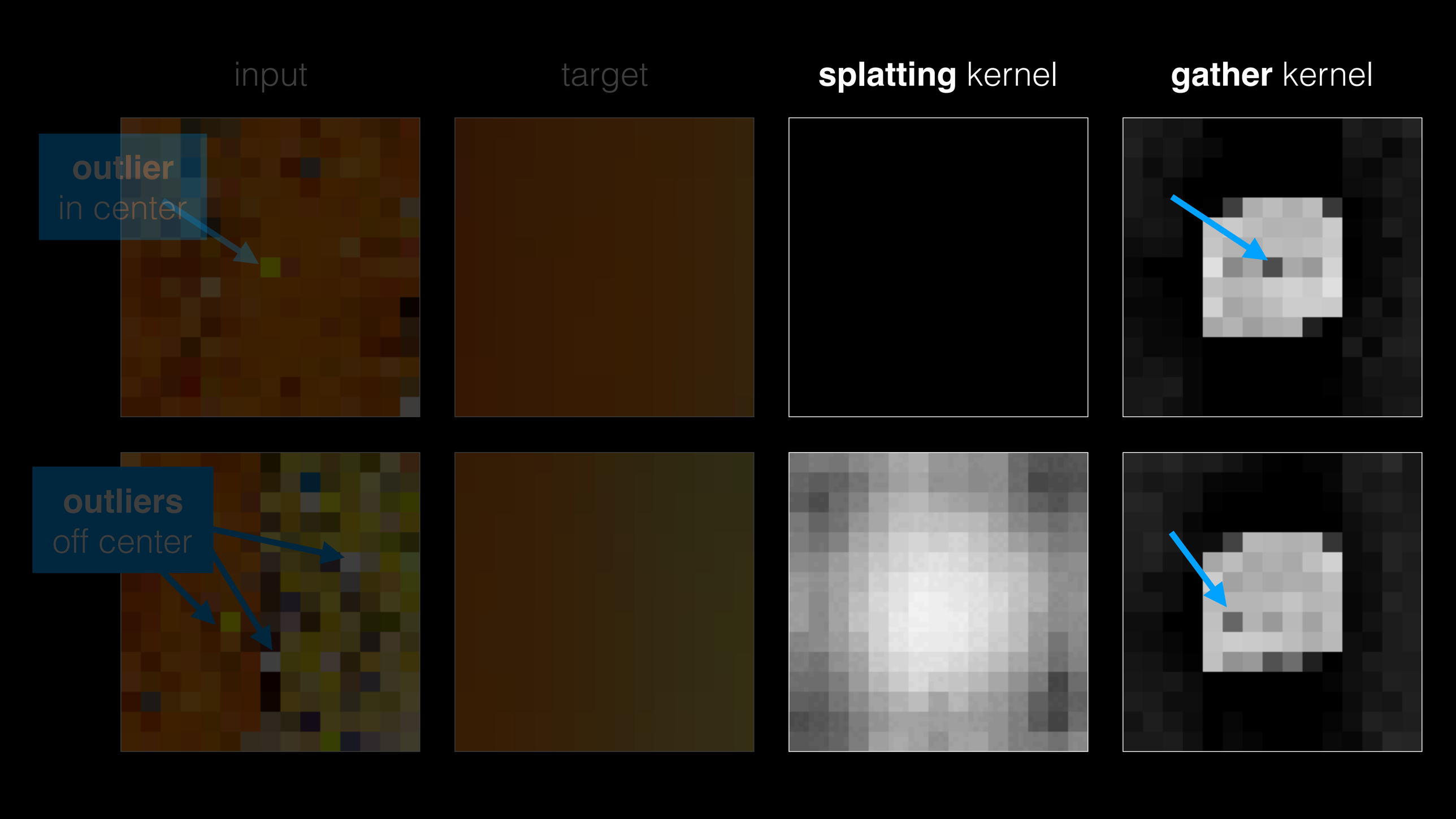


outliers
off center

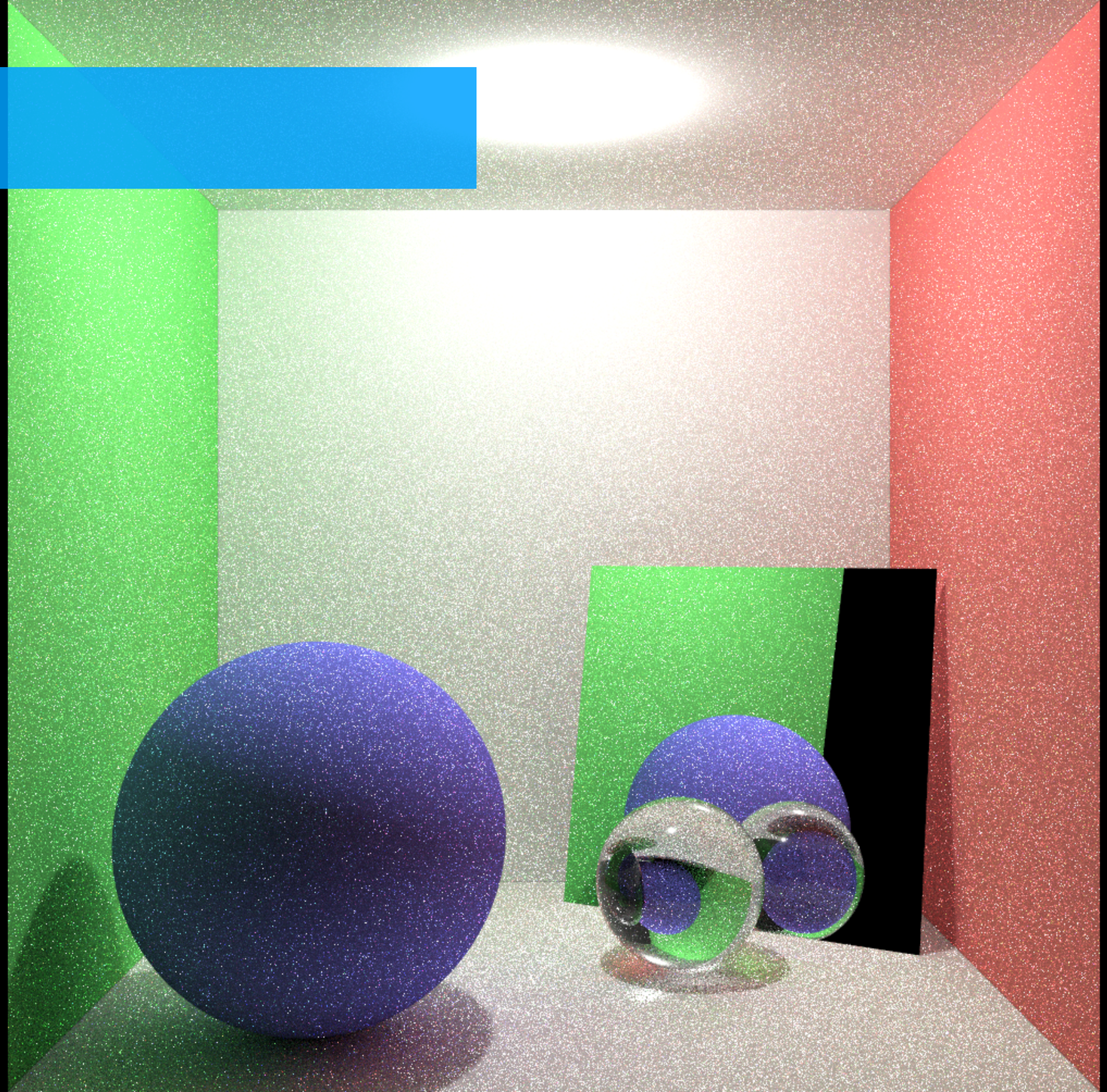


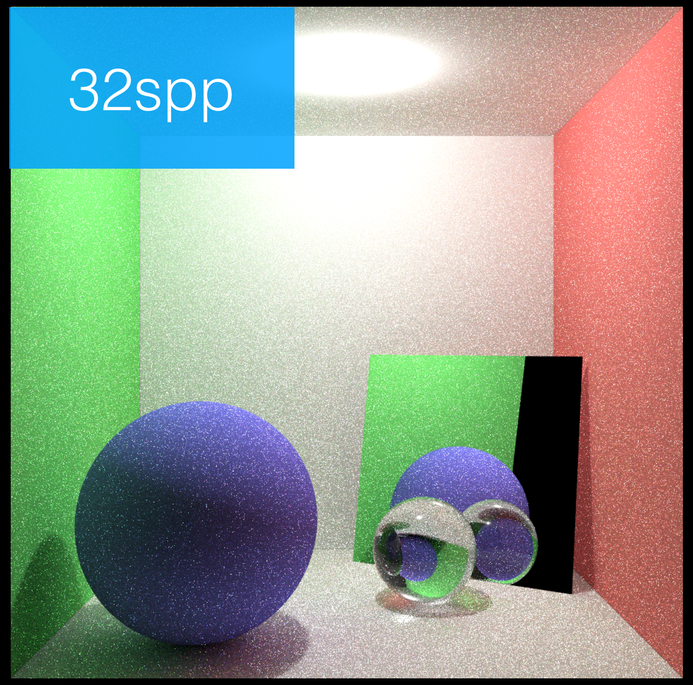


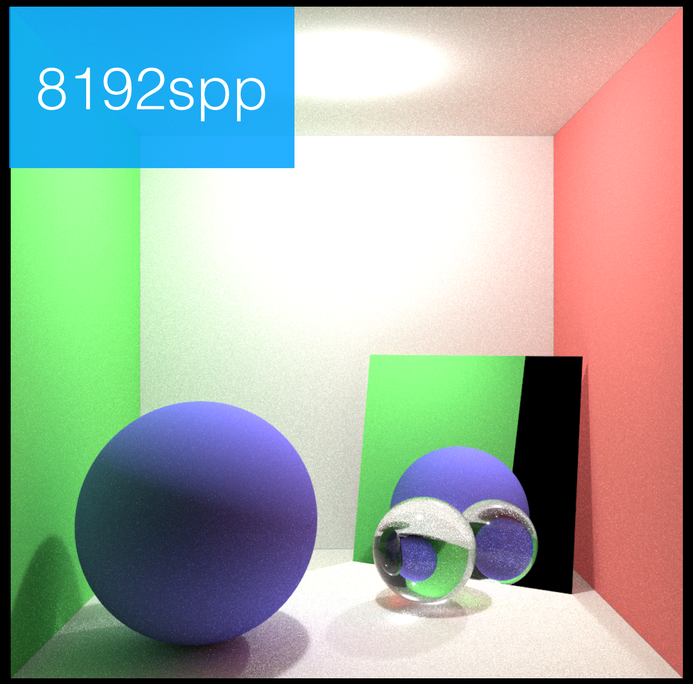


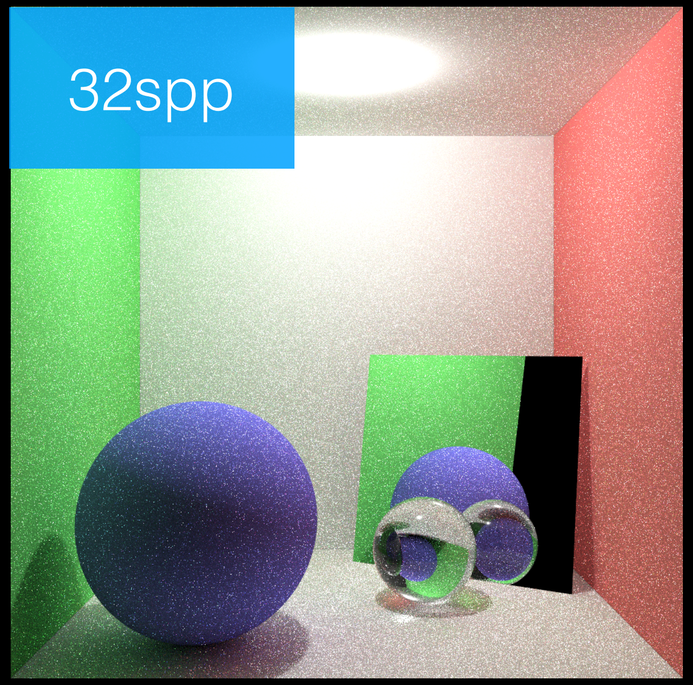

input
target
splatting kernel
gather kernel
outlier
in center
outliers
off center

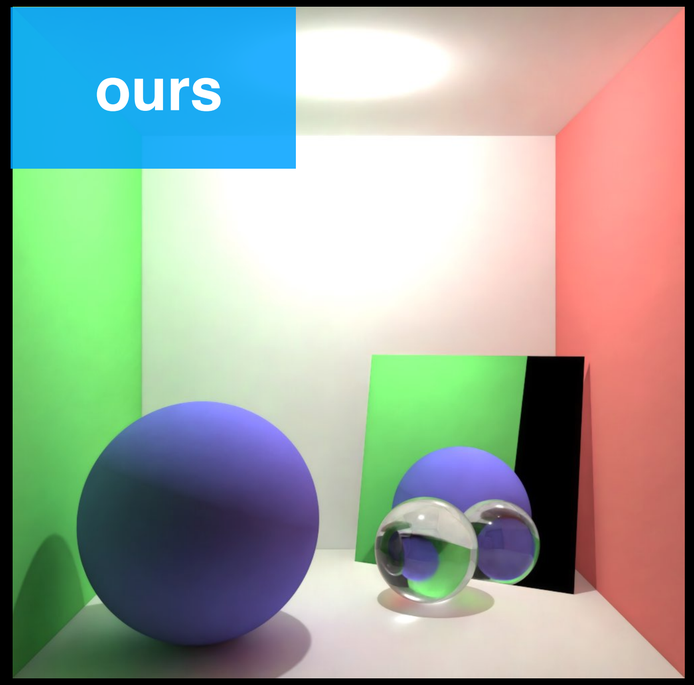
input | 32spp

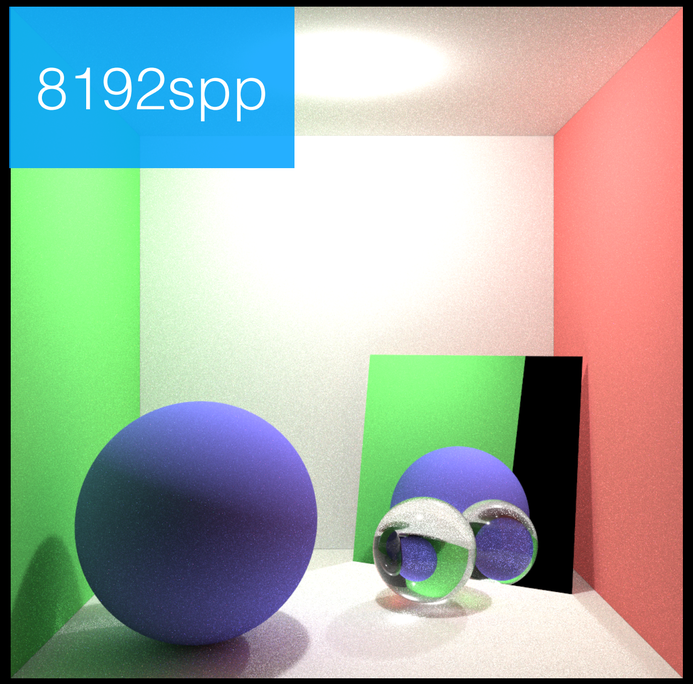


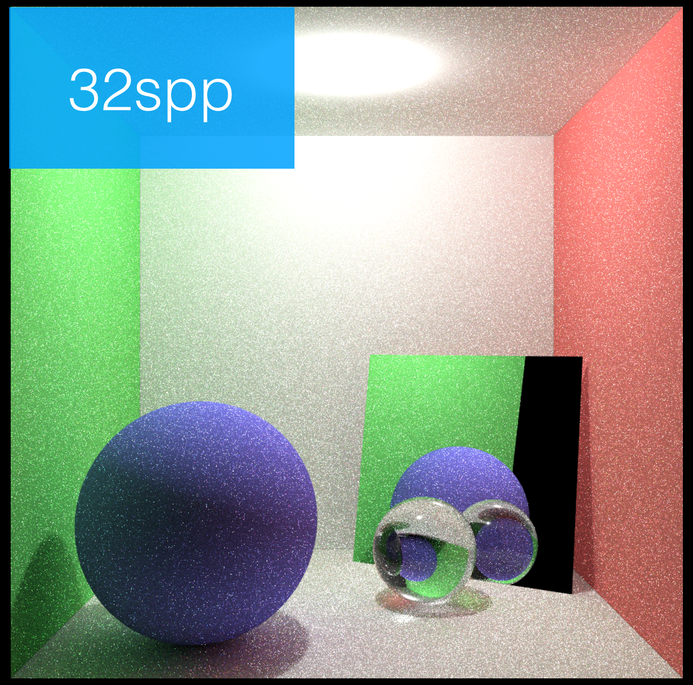

32spp

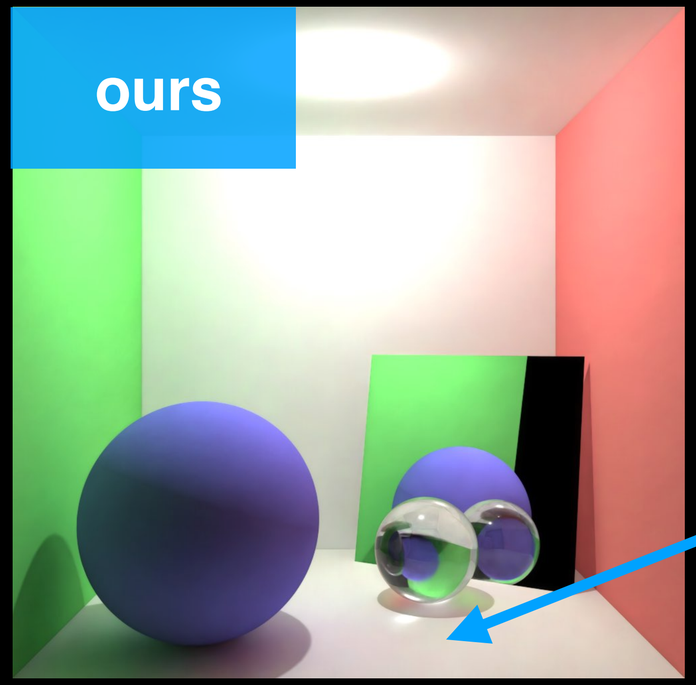

8192spp


32spp

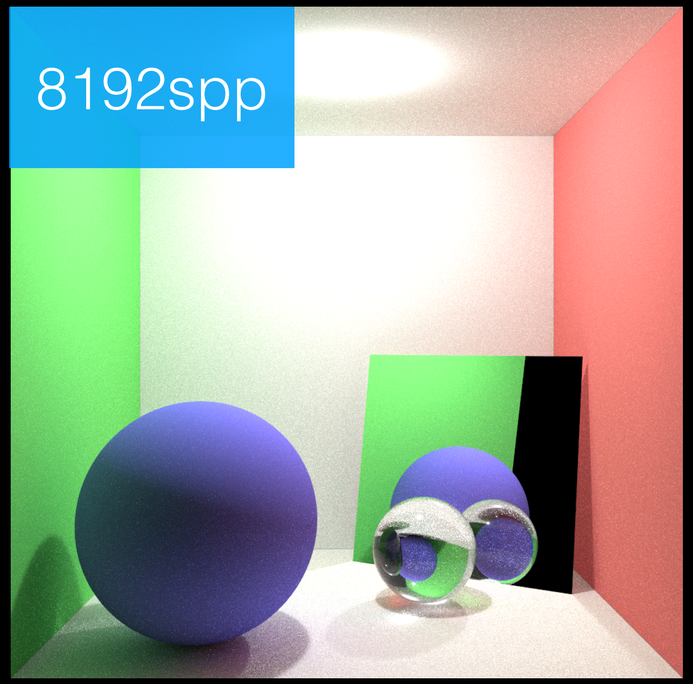

ours

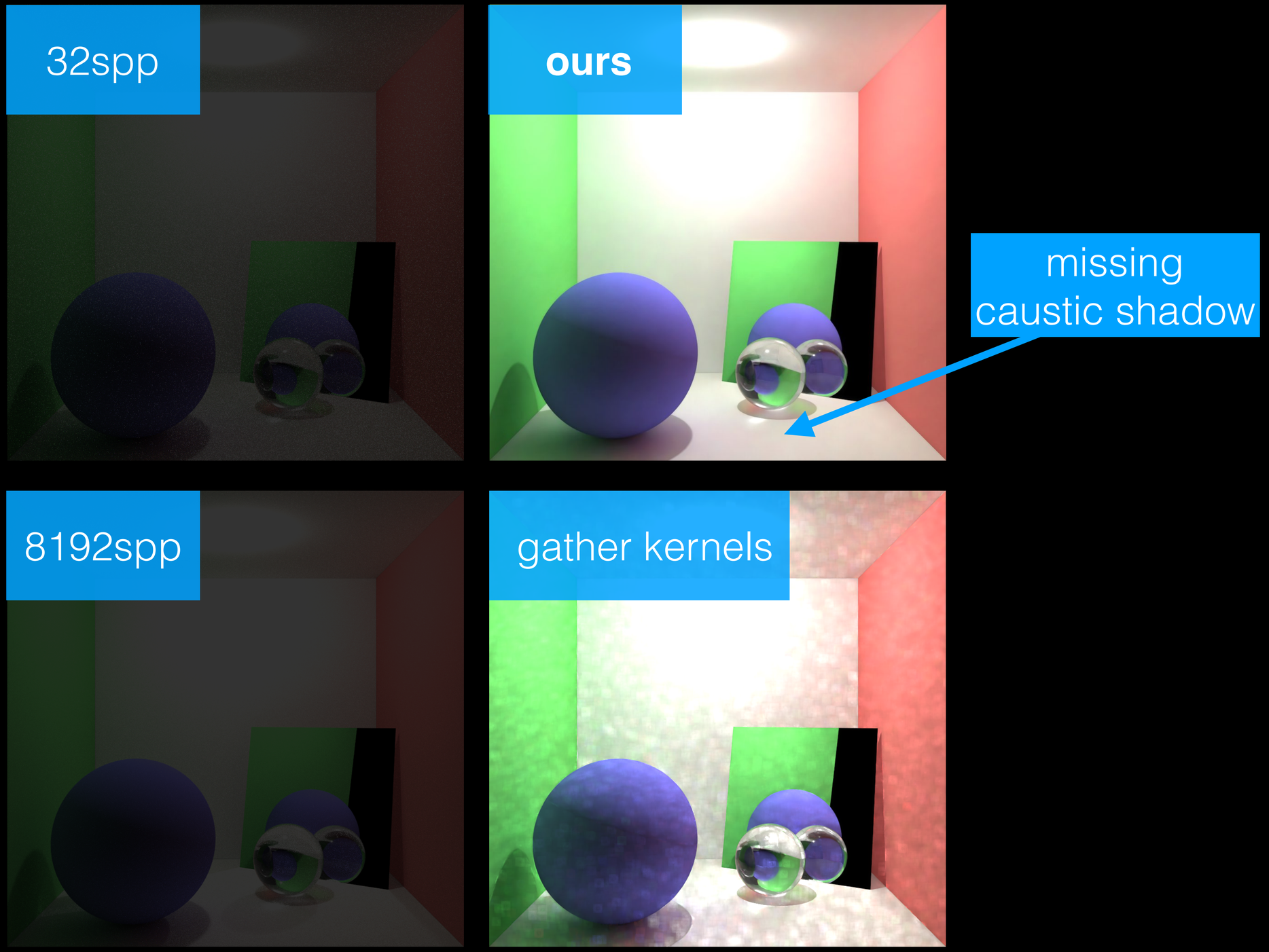

8192spp

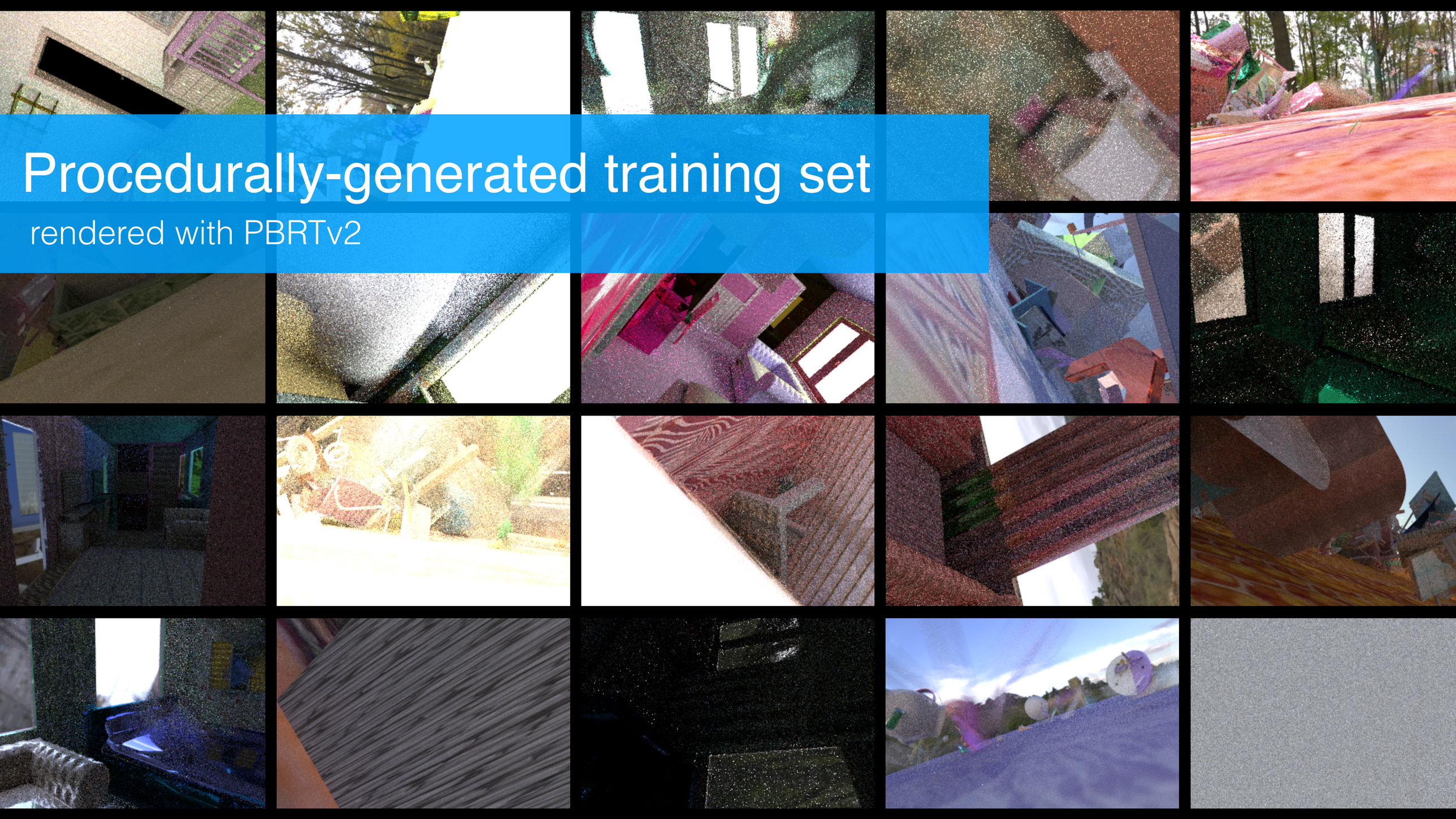




missing
caustic shadow




32spp
ours
missing
caustic shadow
8192spp
gather kernels


Procedurally-generated training set
rendered with PBRTv2

Results

Evaluation

- 55 test scenes

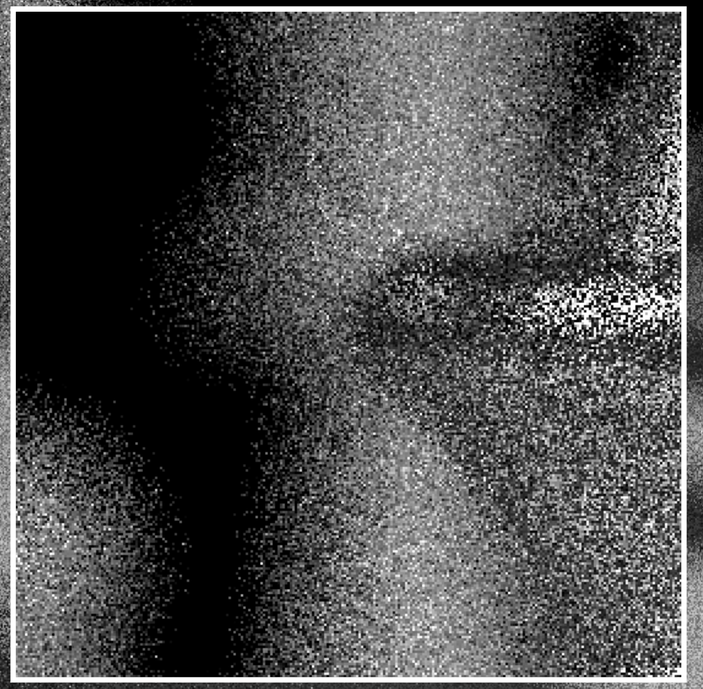
Evaluation

- 55 test scenes
- 5 reference denoisers
  - algorithmic and learning-based

Evaluation

- 55 test scenes
- 5 reference denoisers
  - algorithmic and learning-based
- Ablation study
  - splatting vs. gather kernels
  - per-sample vs. per-pixel
  - constant number of network parameters

input | 4spp



Bako et al. [2017] | 4spp

ours | 4spp

reference | 8192spp

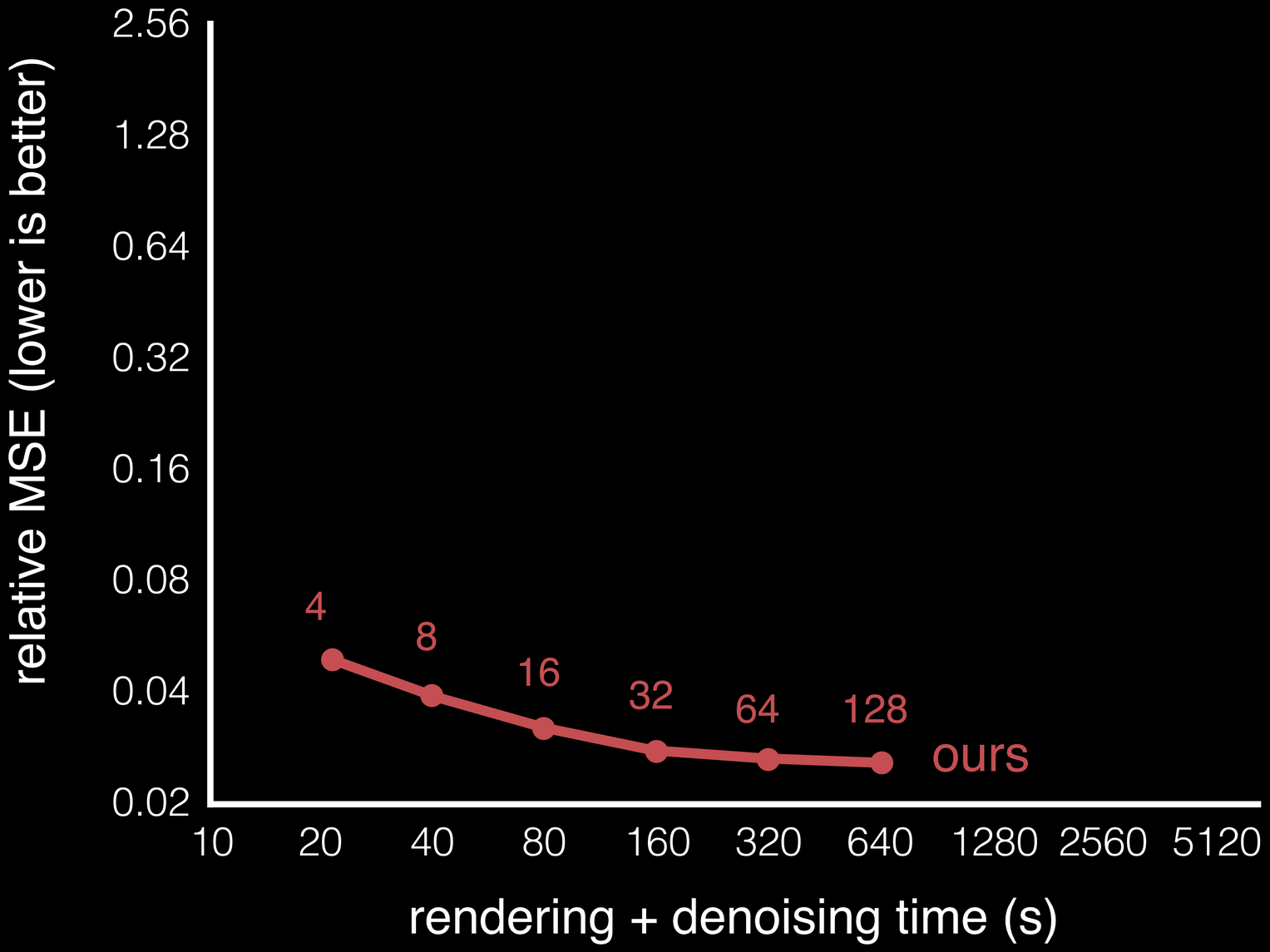
input | 4spp

Bako et al. [2017] | 4spp

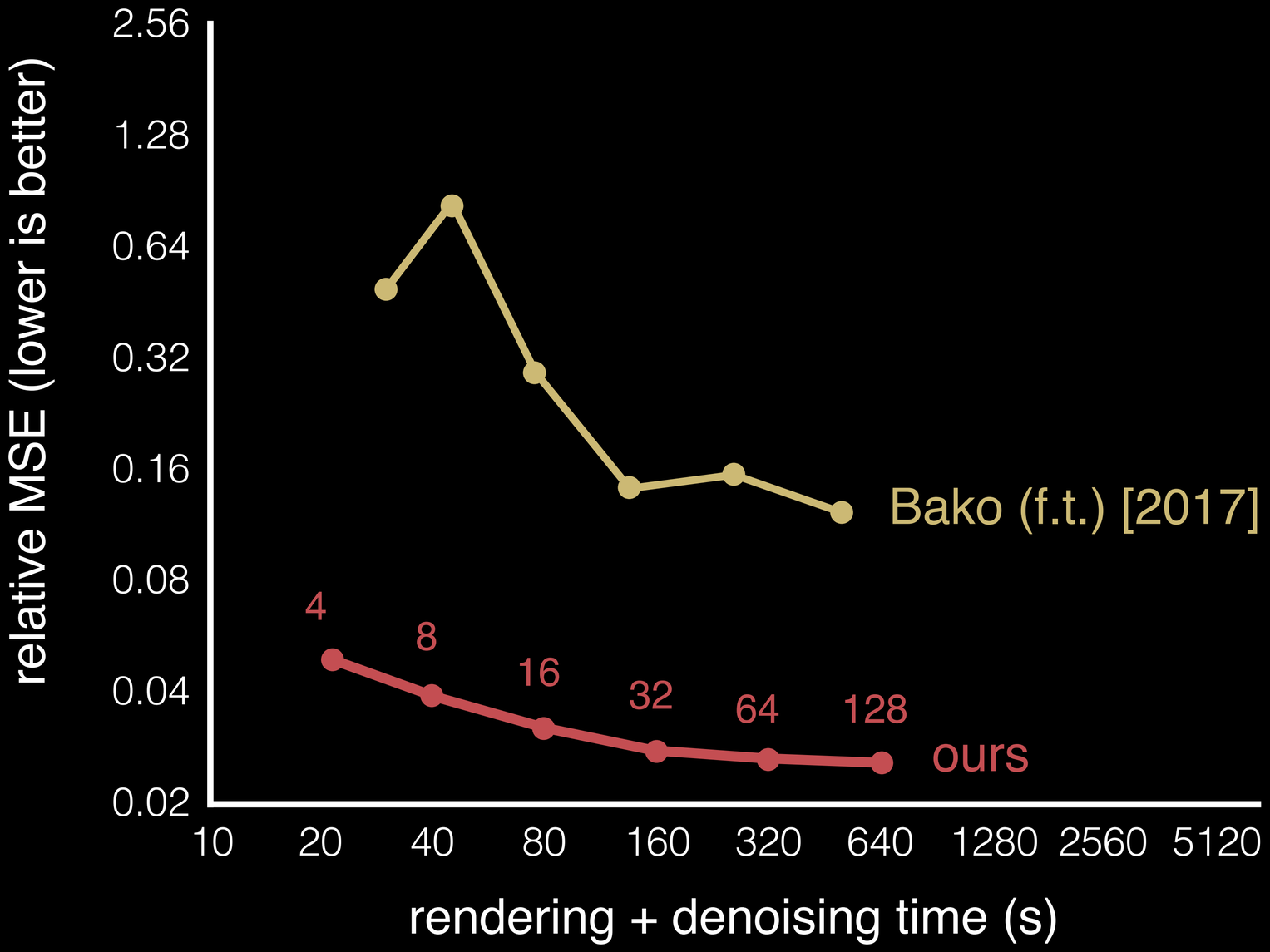
ours | 4spp

reference | 8192spp

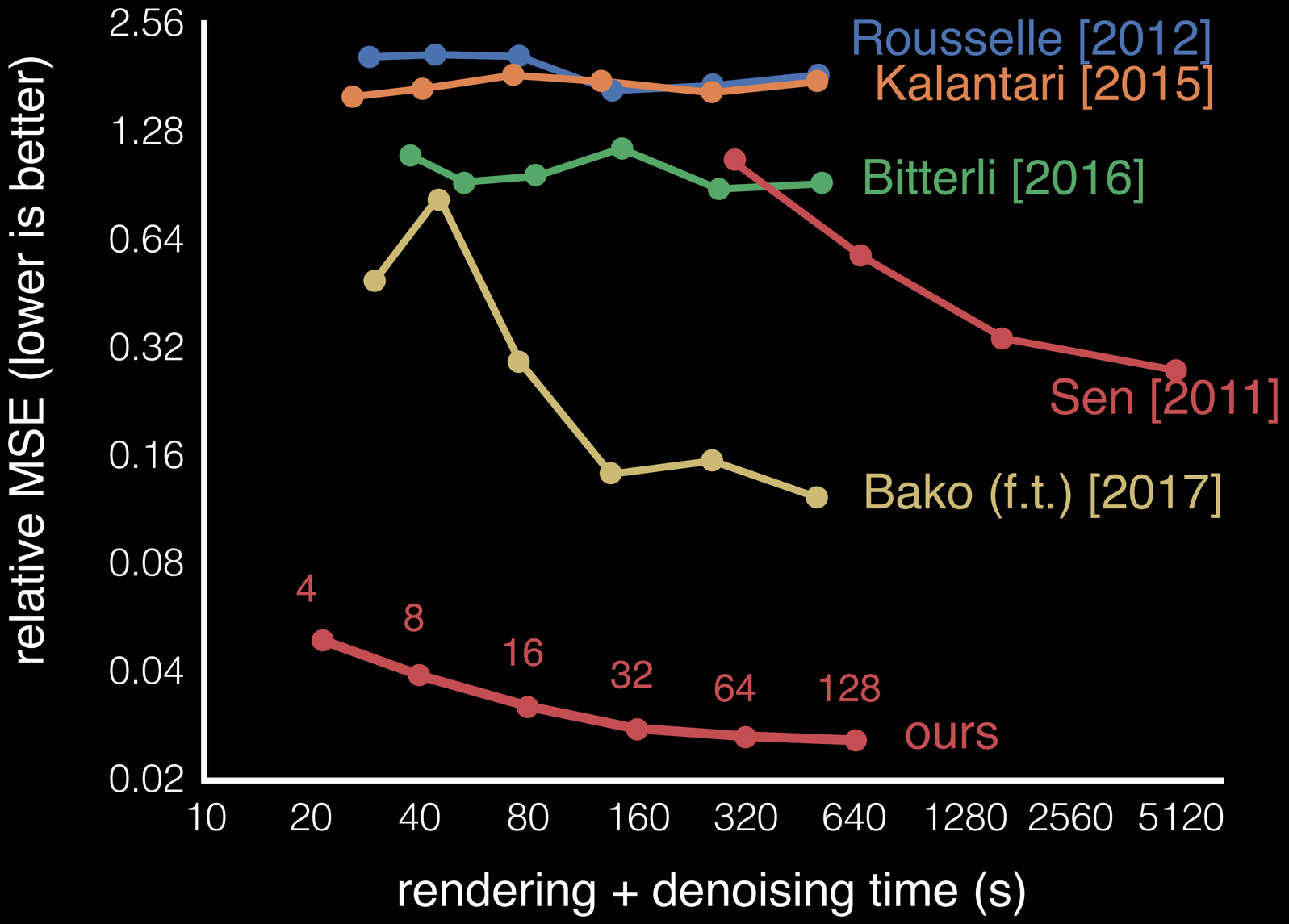
rMSE vs. running time (rendering + denoising)



rMSE vs. running time (rendering + denoising)



rMSE vs. running time (rendering + denoising)



Discussion

- We investigate how to better use individual samples
  - with deep learning

Discussion

- We investigate how to better use individual samples
  - with deep learning
- Runtime scales linearly with sample count
  - Also linear memory footprint
  - More applicable to few sample regime (4—32spp)

Discussion

- We investigate how to better use individual samples
  - with deep learning
- Runtime scales linearly with sample count
  - Also linear memory footprint
  - More applicable to few sample regime (4—32spp)
- Samples sum radiance over complete light path

Conclusion

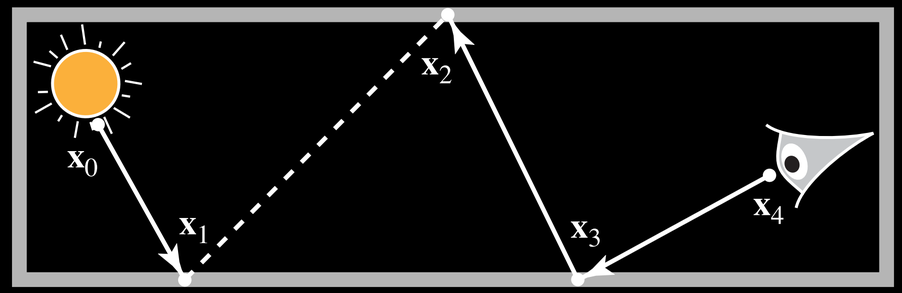
- Samples provide richer information
  - compared to per-pixel statistics
- Splatting kernels
  - high-quality output, simpler prediction
  - robust to severe noise
- Permutation-invariant network
  - map from arbitrary number of samples to pixels

Michaël GHARBI mgharbi@adobe.com
code/models/data - www.mgharbi.com

Extras

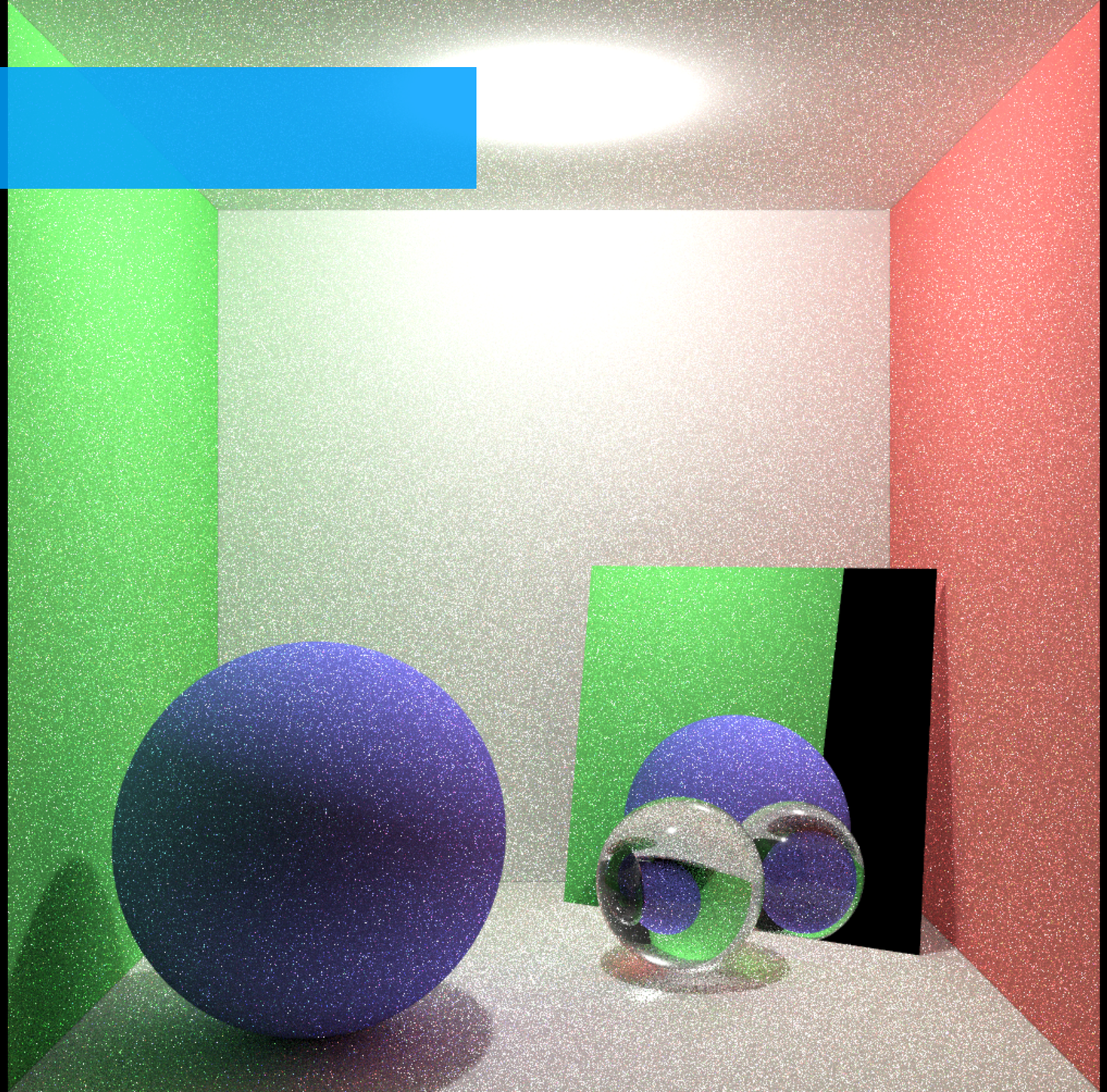
Data generation

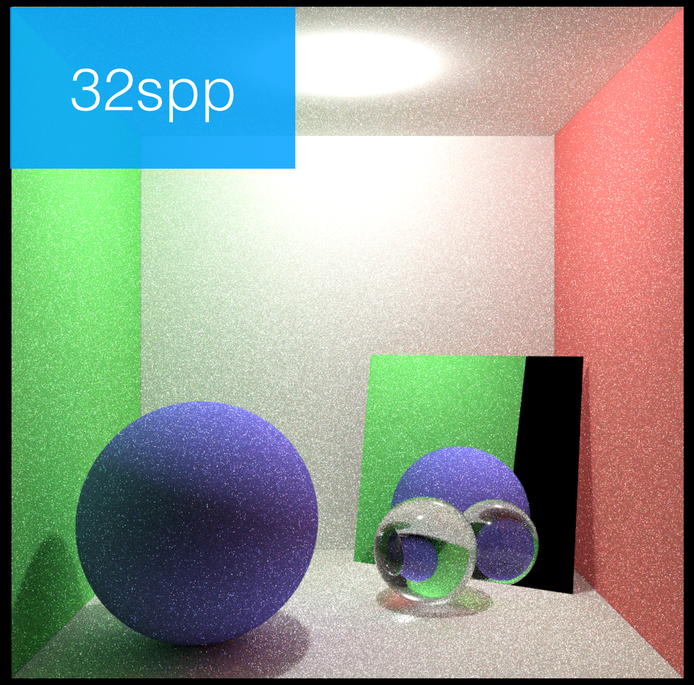
- Custom scene generator
  - geometry from SunCG [Song 2017] and ShapeNet [Chang 2015]
  - textures from Cimpoi et al. [2014]
  - randomized materials, cameras, lights
- Path tracing with fixed path length
  - PBRTv2 [Pharr 2004]
- Store samples and path information
  - radiance at multiple bounces along the path
  - depth, normals, albedo, x, y, t, lens position, etc

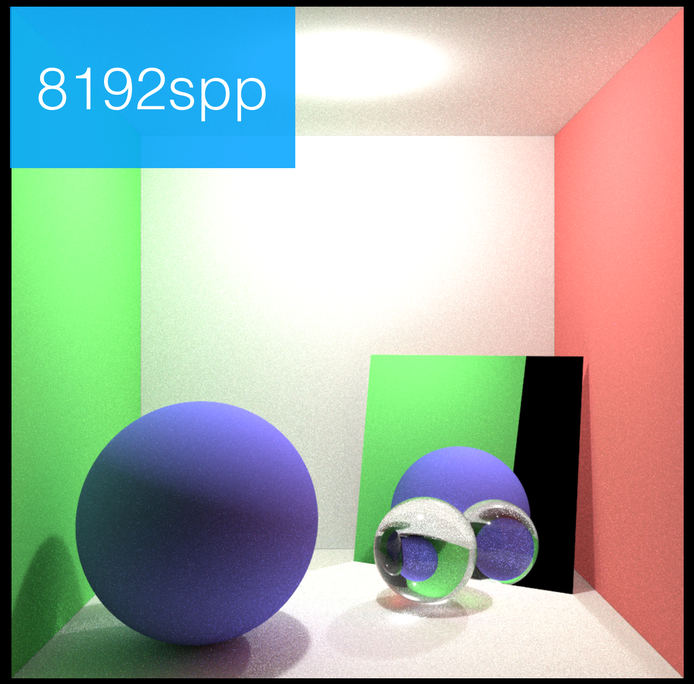


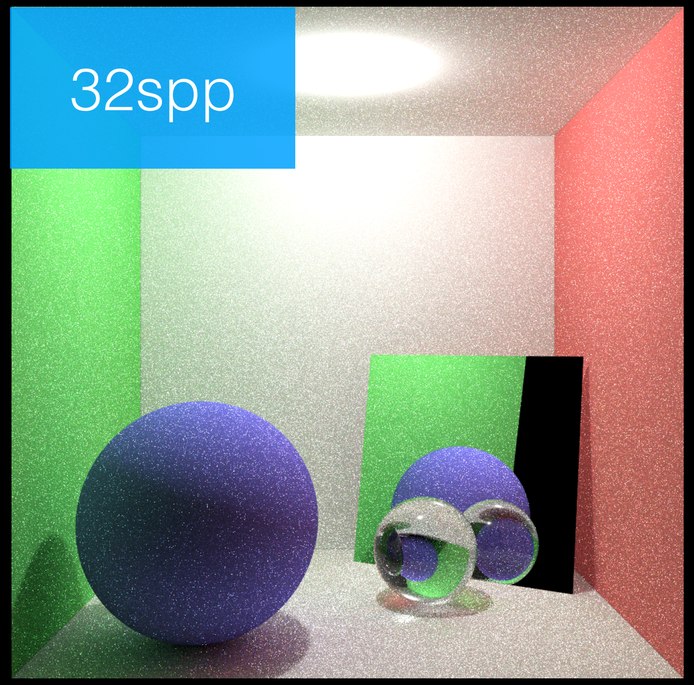
Model ablations

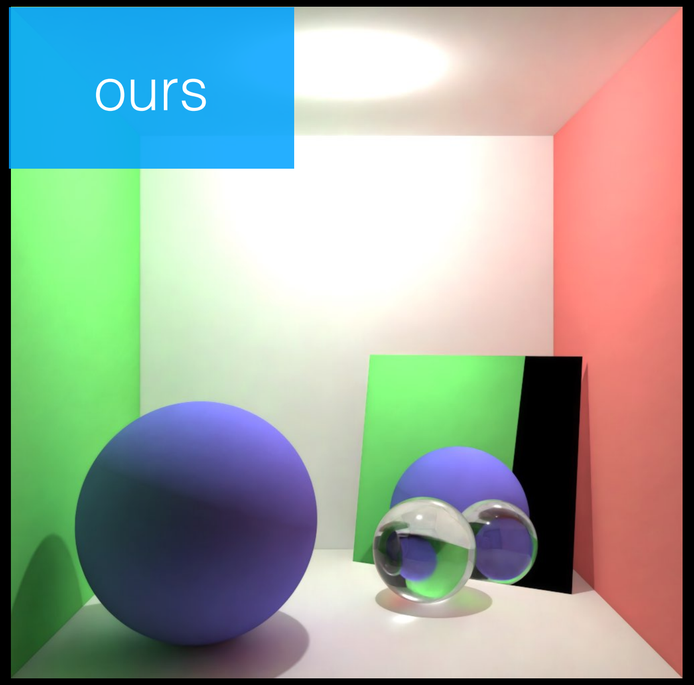
input | 32spp

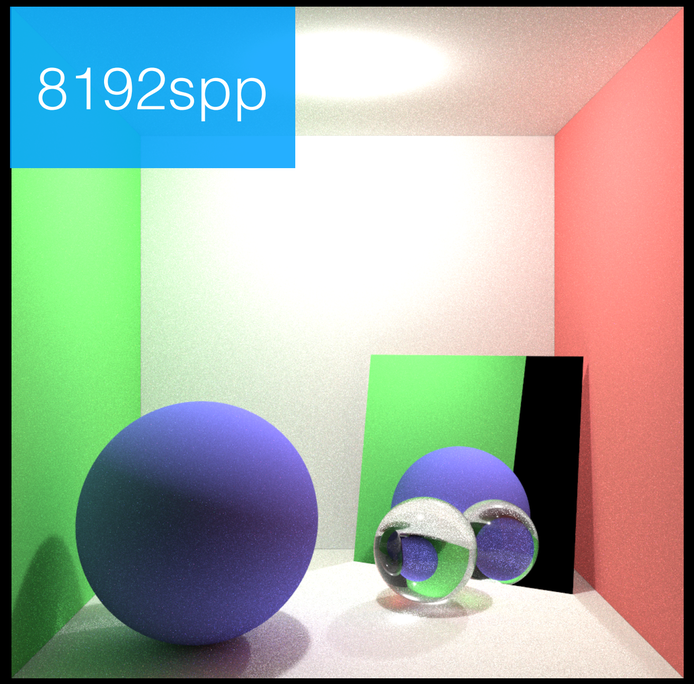


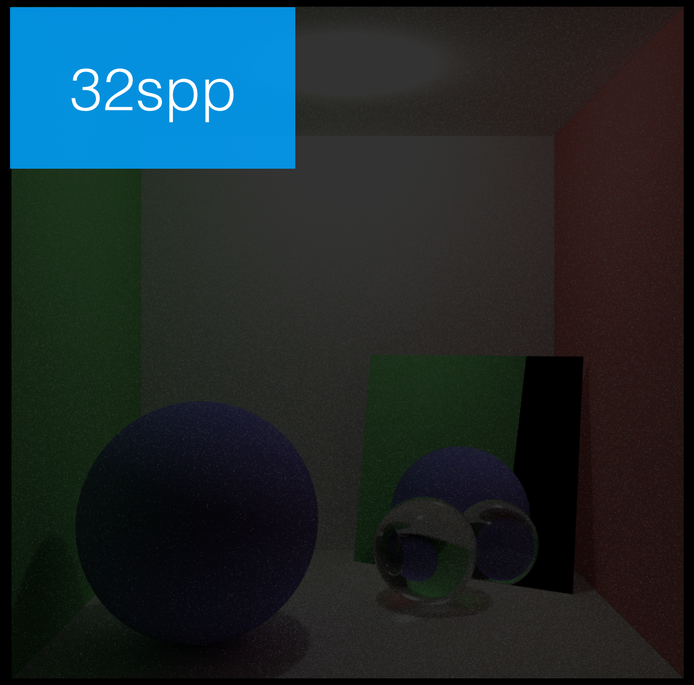

32spp

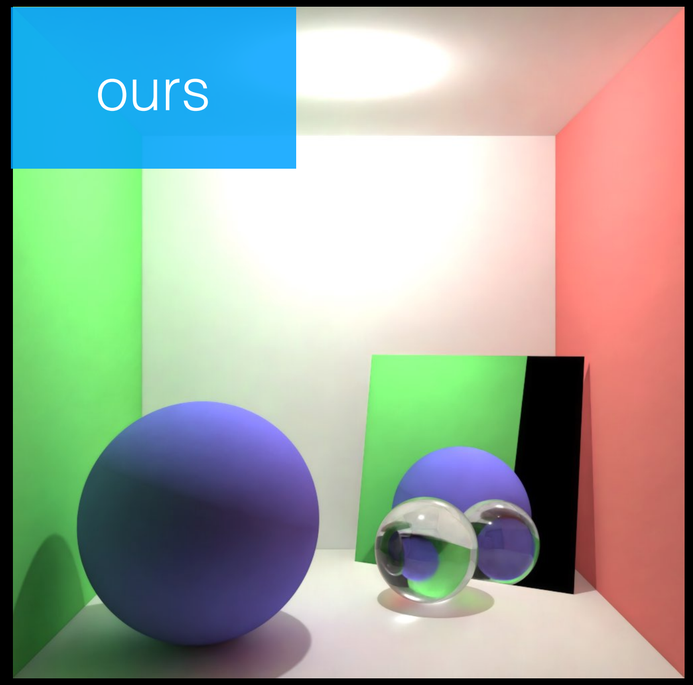

8192spp

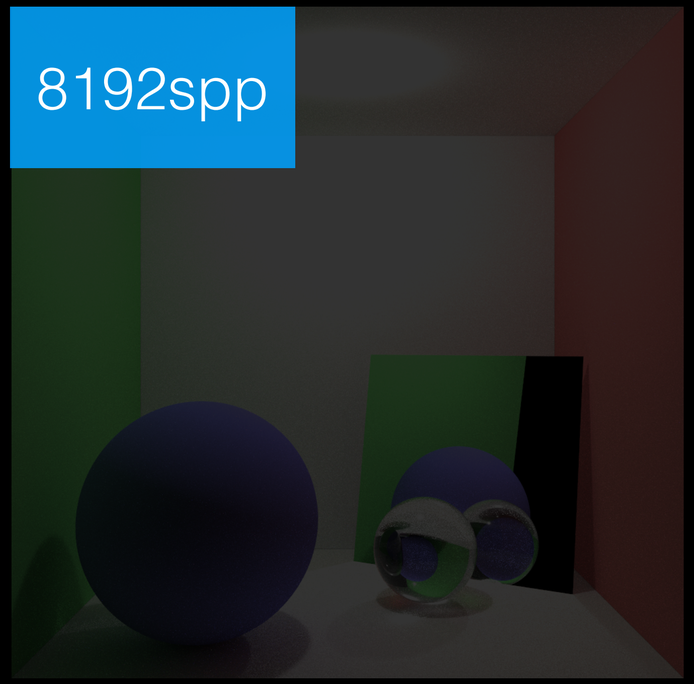

32spp

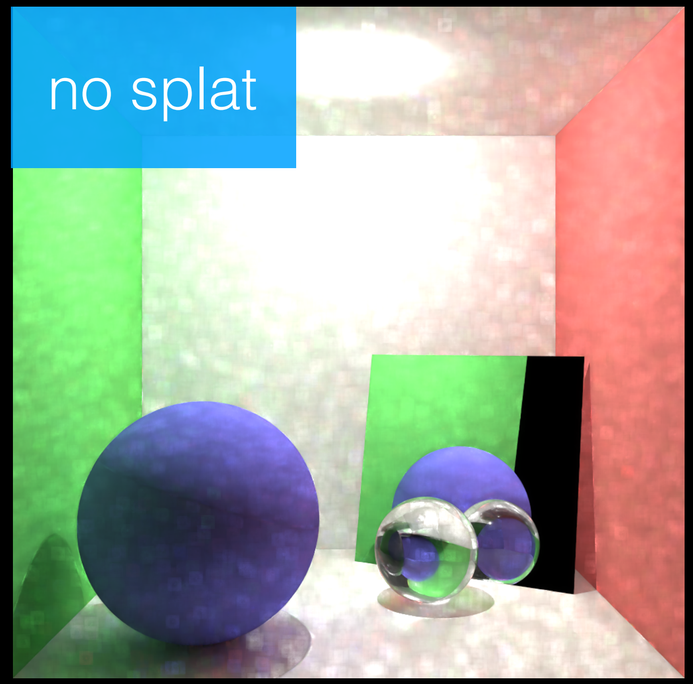

ours

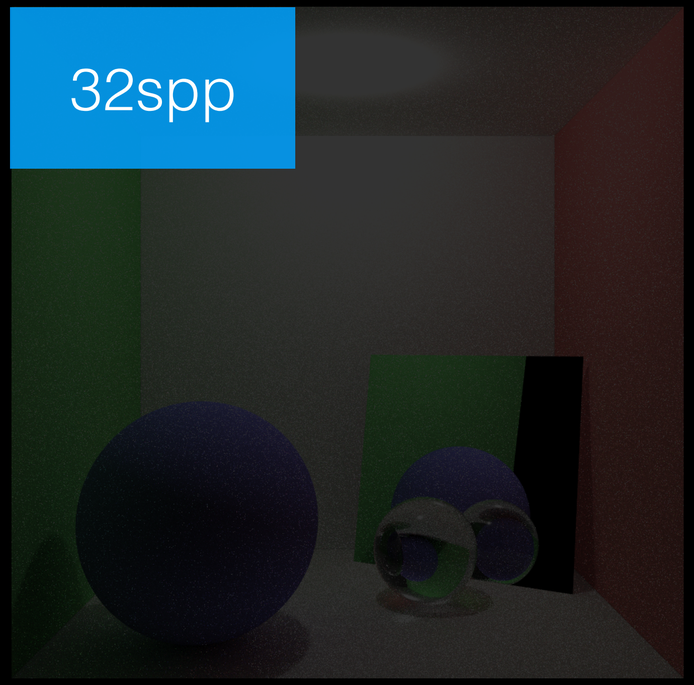

8192spp

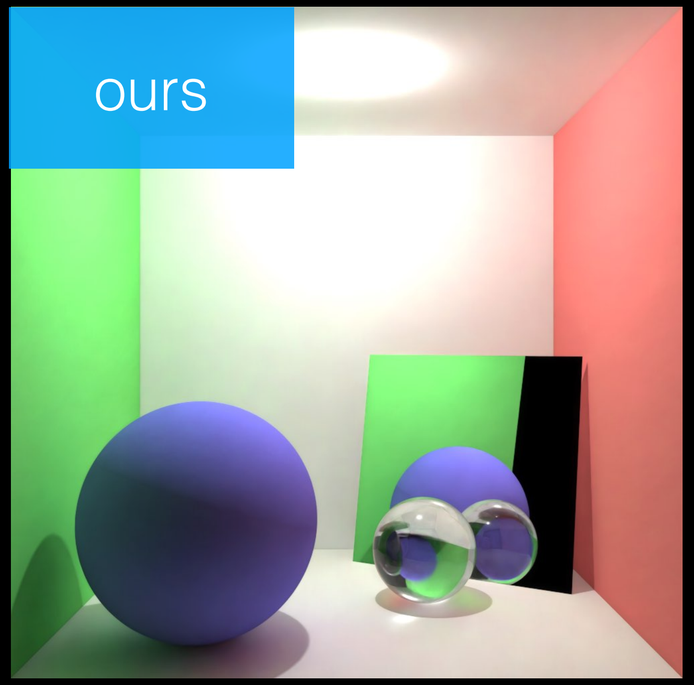

32spp

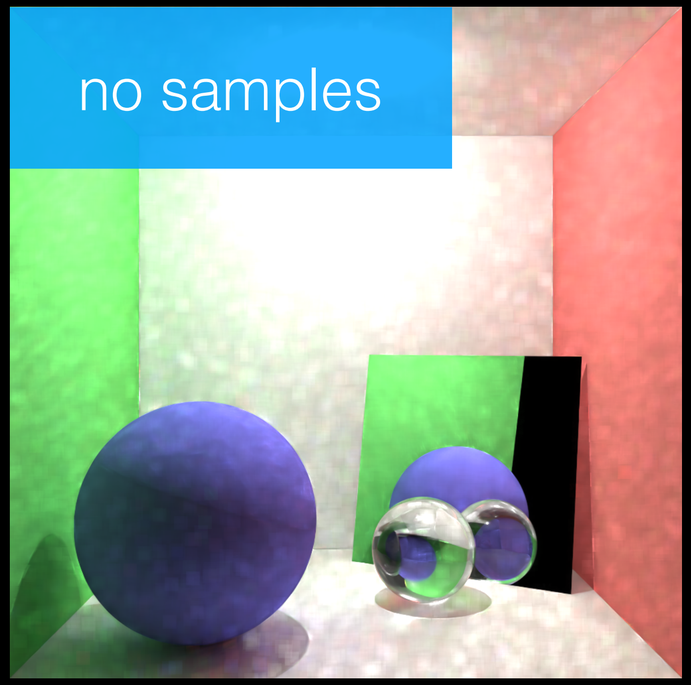

ours

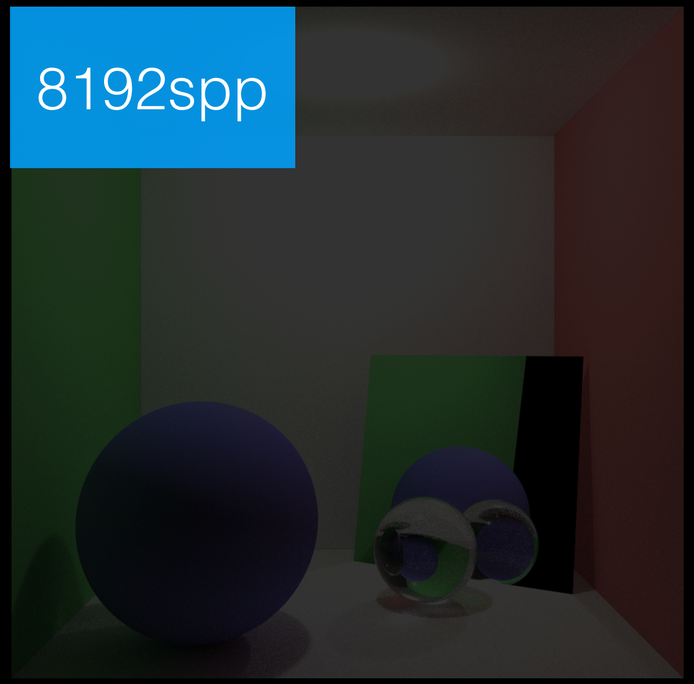

8192spp

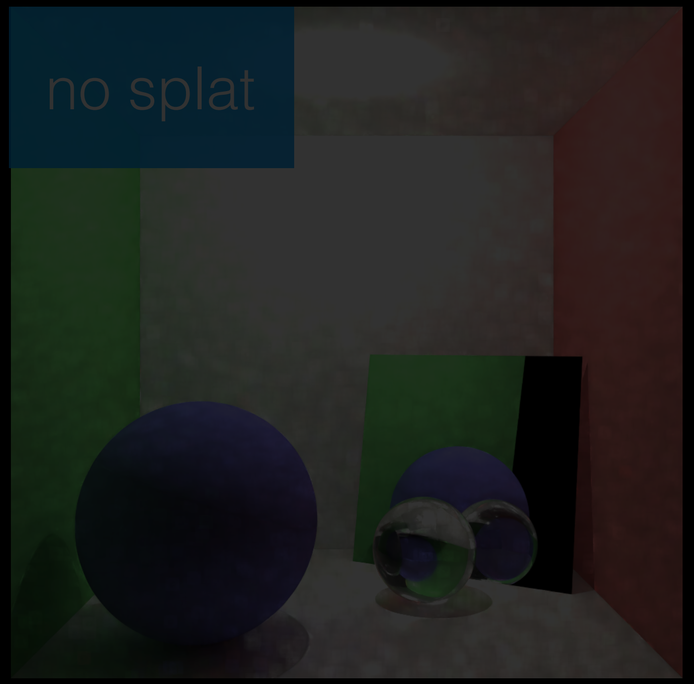

no splat

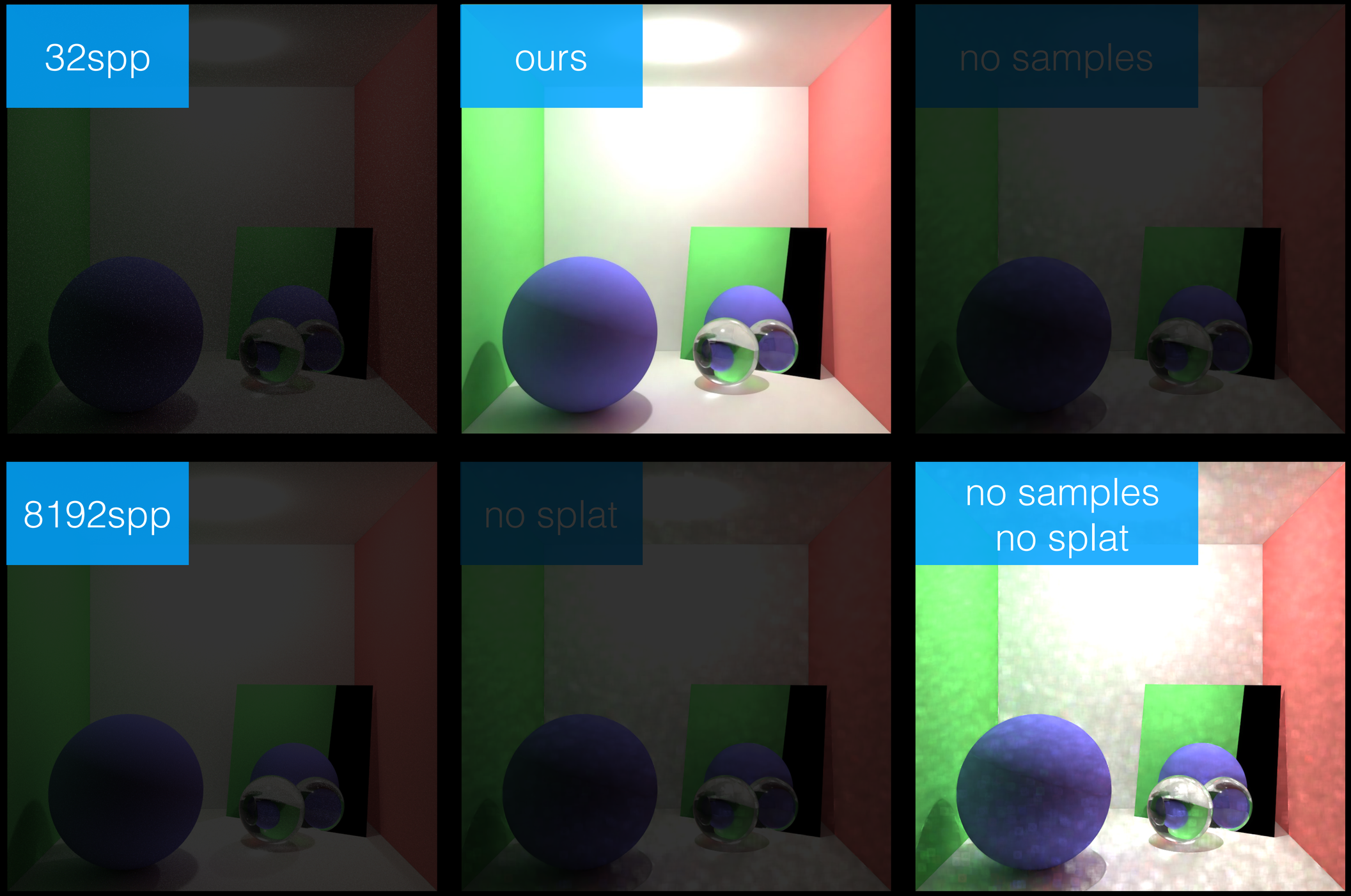

32spp

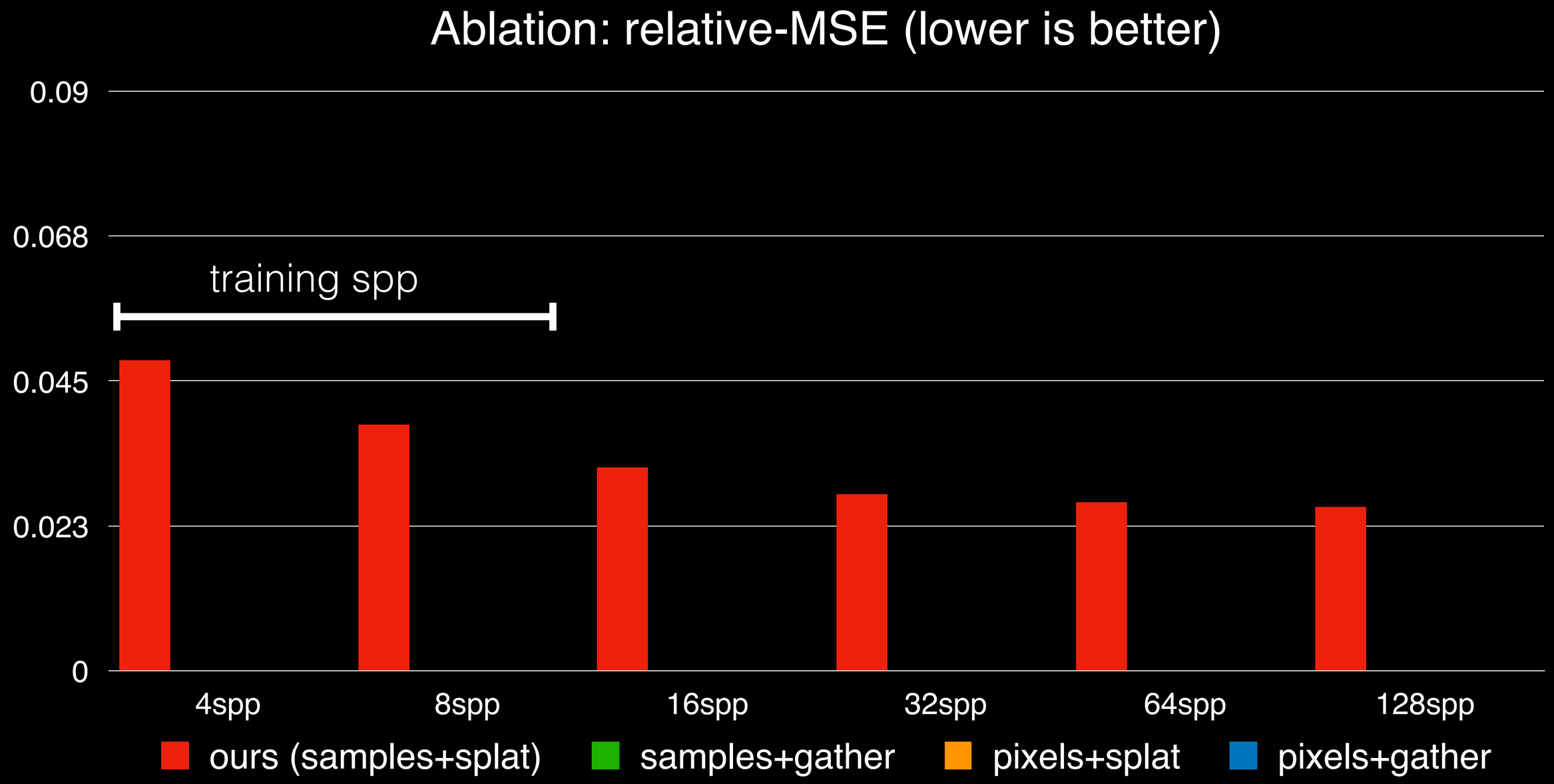

ours

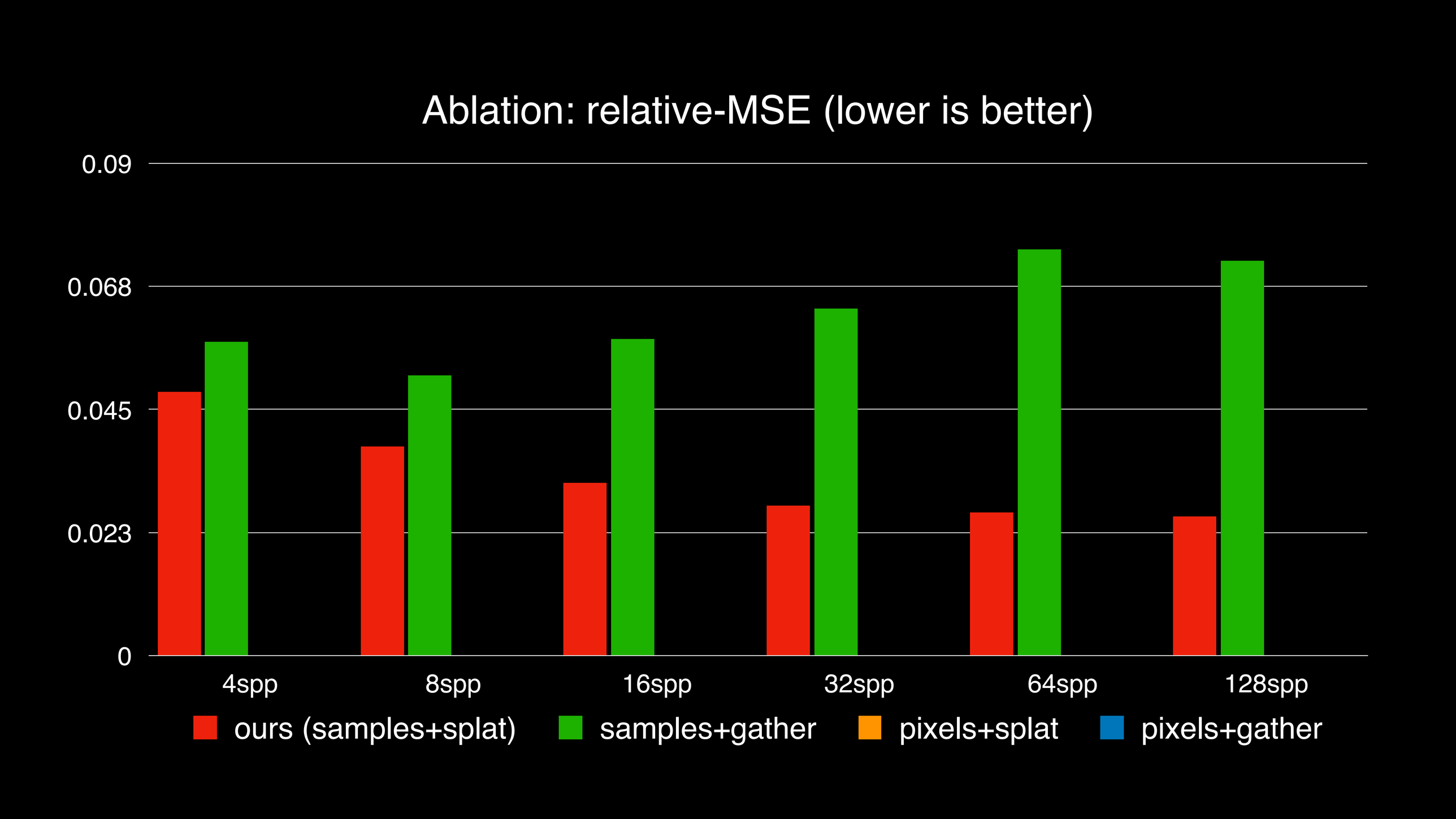

no samples

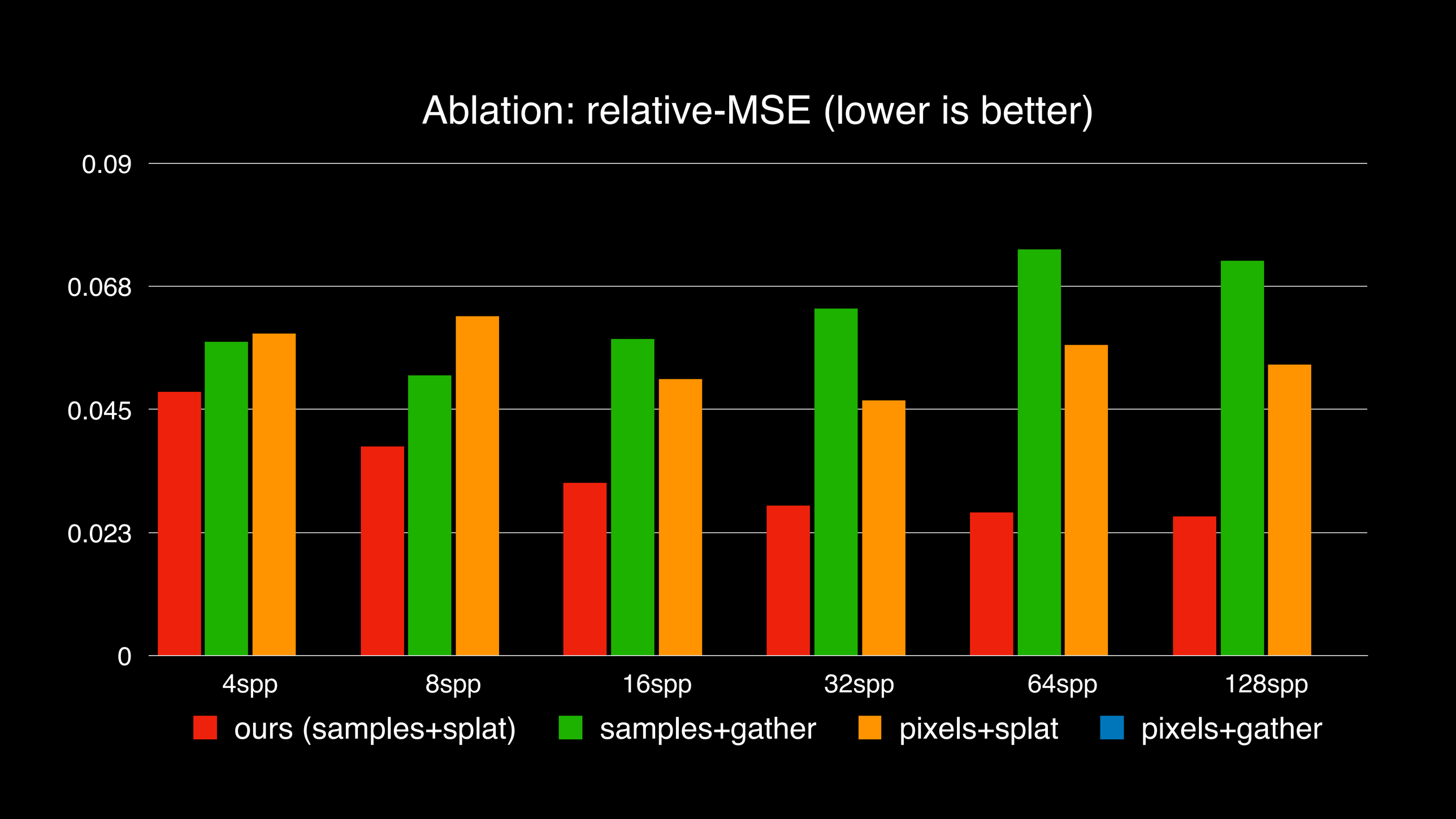

8192spp

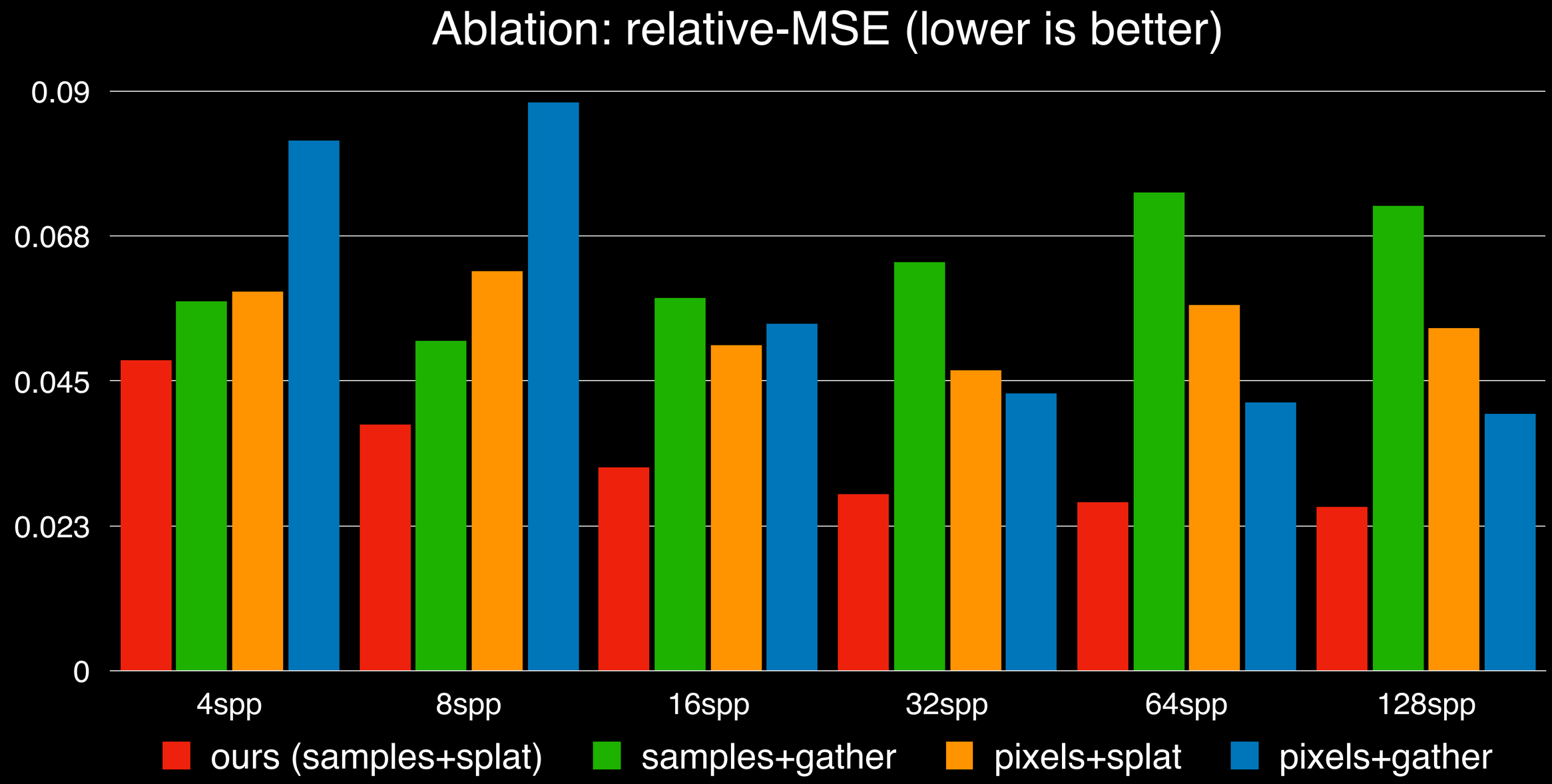

no splat


32spp
ours
no samples
8192spp
no splat
no samples
no splat


Ablation: relative-MSE (lower is better)
0.09
0.068
0.045
0.023
0
training spp
4spp
8spp
16spp
32spp
64spp
128spp
ours (samples+splat)
samples+gather
pixels+splat
pixels+gather

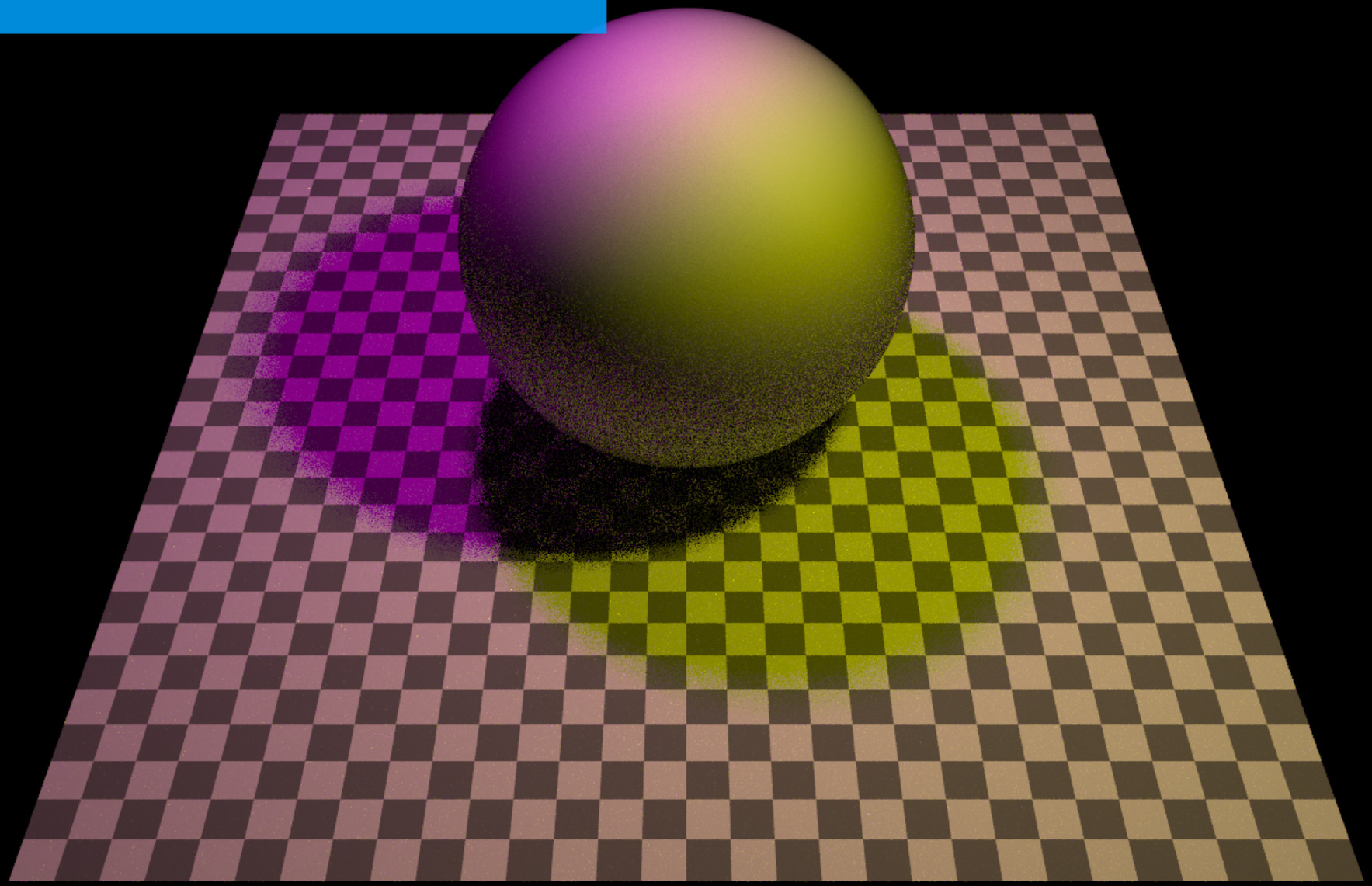

Ablation: relative-MSE (lower is better)
0.09
0.068
0.045
0.023
0
4spp
8spp
16spp
32spp
64spp
128spp
ours (samples+splat)
samples+gather
pixels+splat
pixels+gather


Ablation: relative-MSE (lower is better)
0.09
0.068
0.045
0.023
0
4spp
8spp
16spp
32spp
64spp
128spp
ours (samples+splat)
samples+gather
pixels+splat
pixels+gather

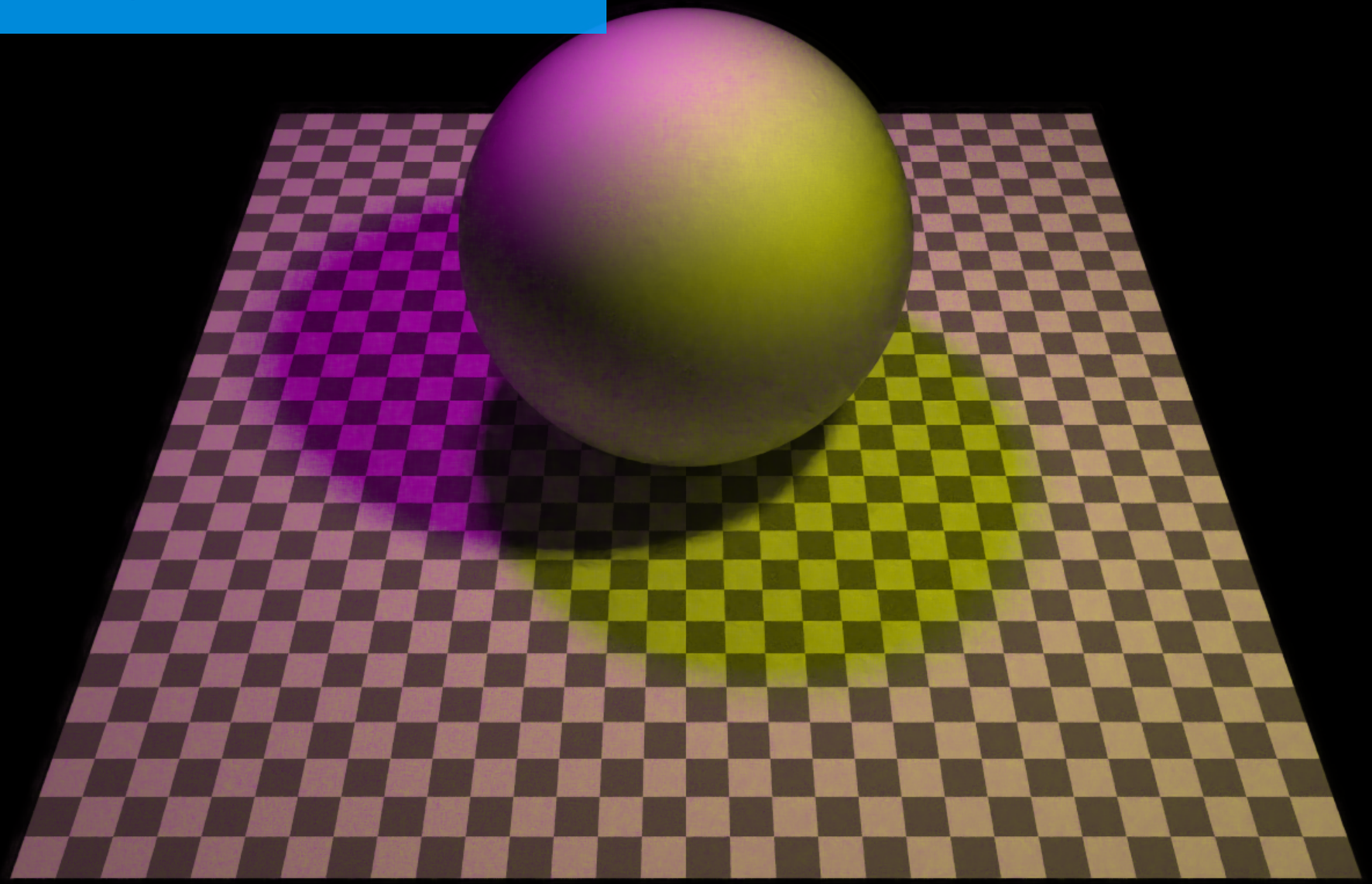

Ablation: relative-MSE (lower is better)
0.09
0.068
0.045
0.023
0
4spp
8spp
16spp
32spp
64spp
128spp
ours (samples+splat)
samples+gather
pixels+splat
pixels+gather

Extra results

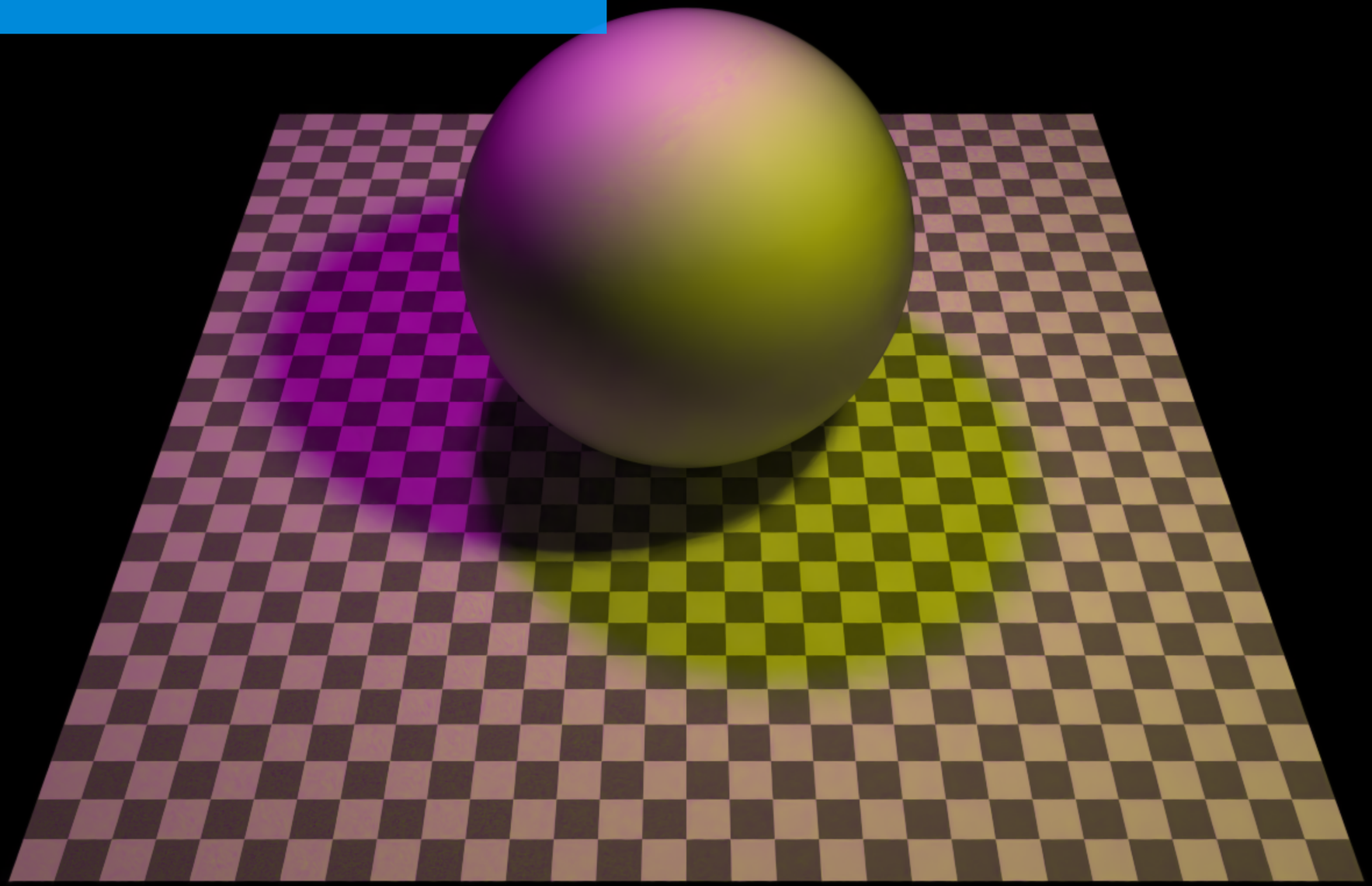
input | 8spp



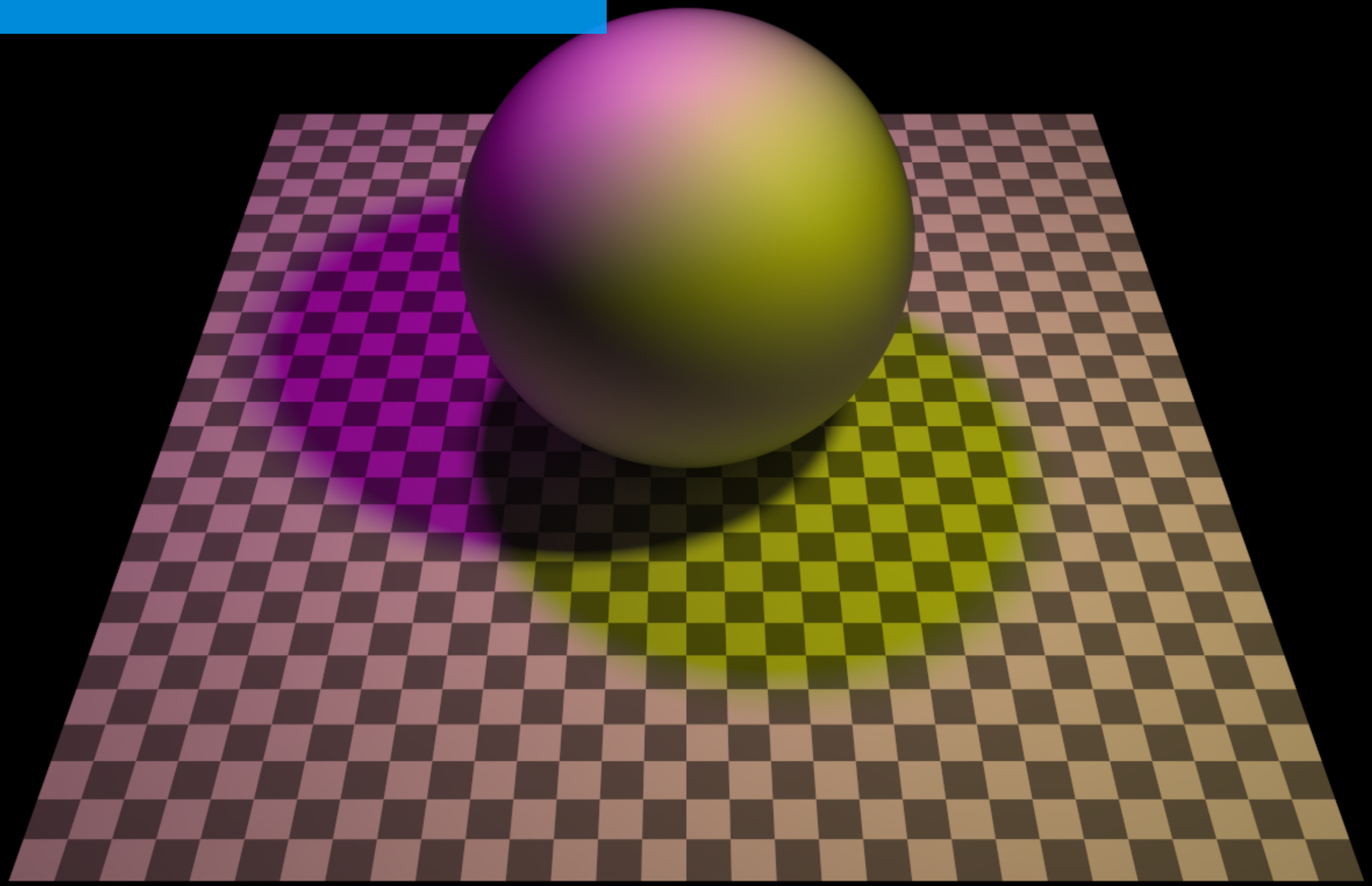
Bako et al. [2017] | 8spp

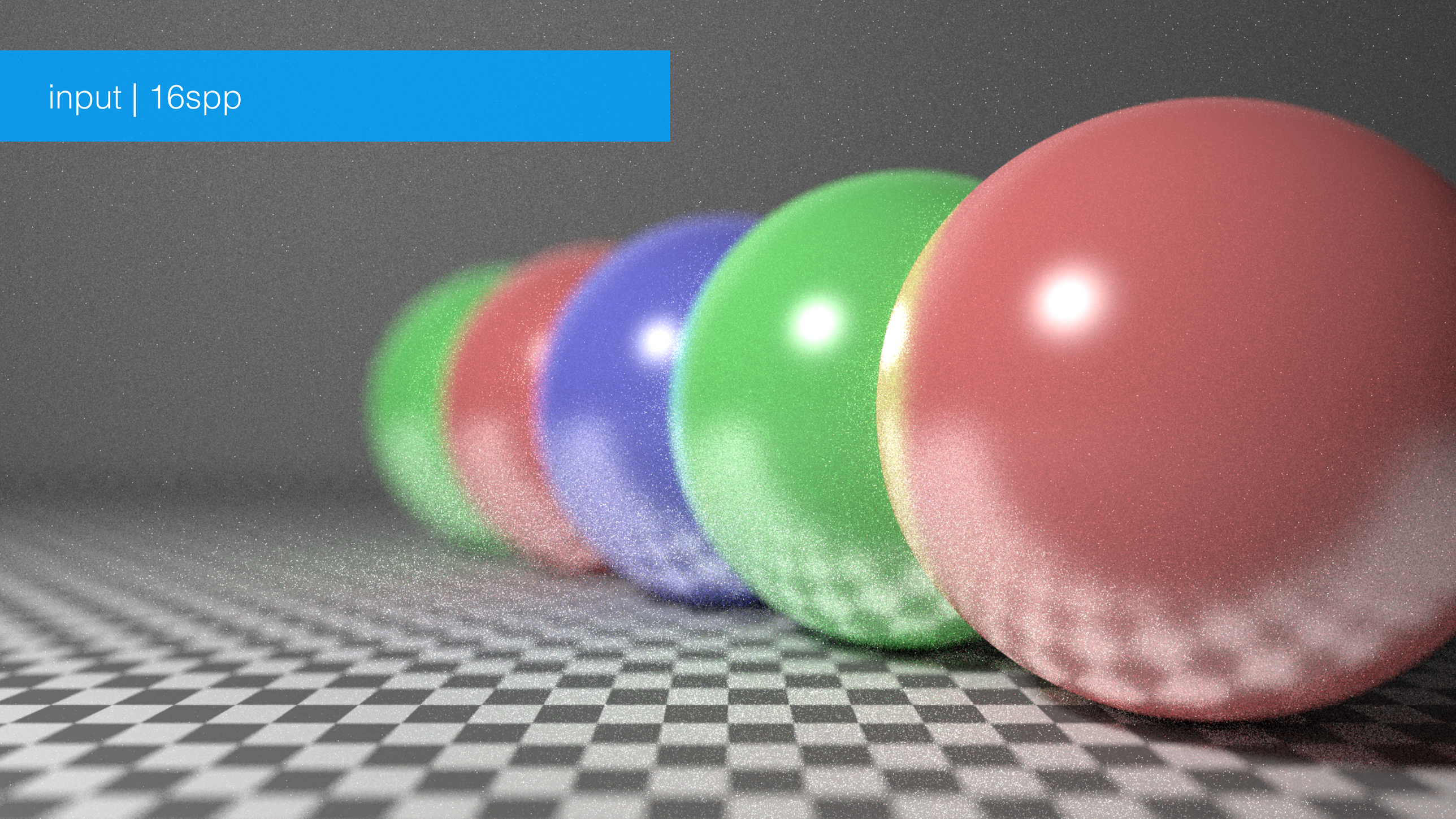


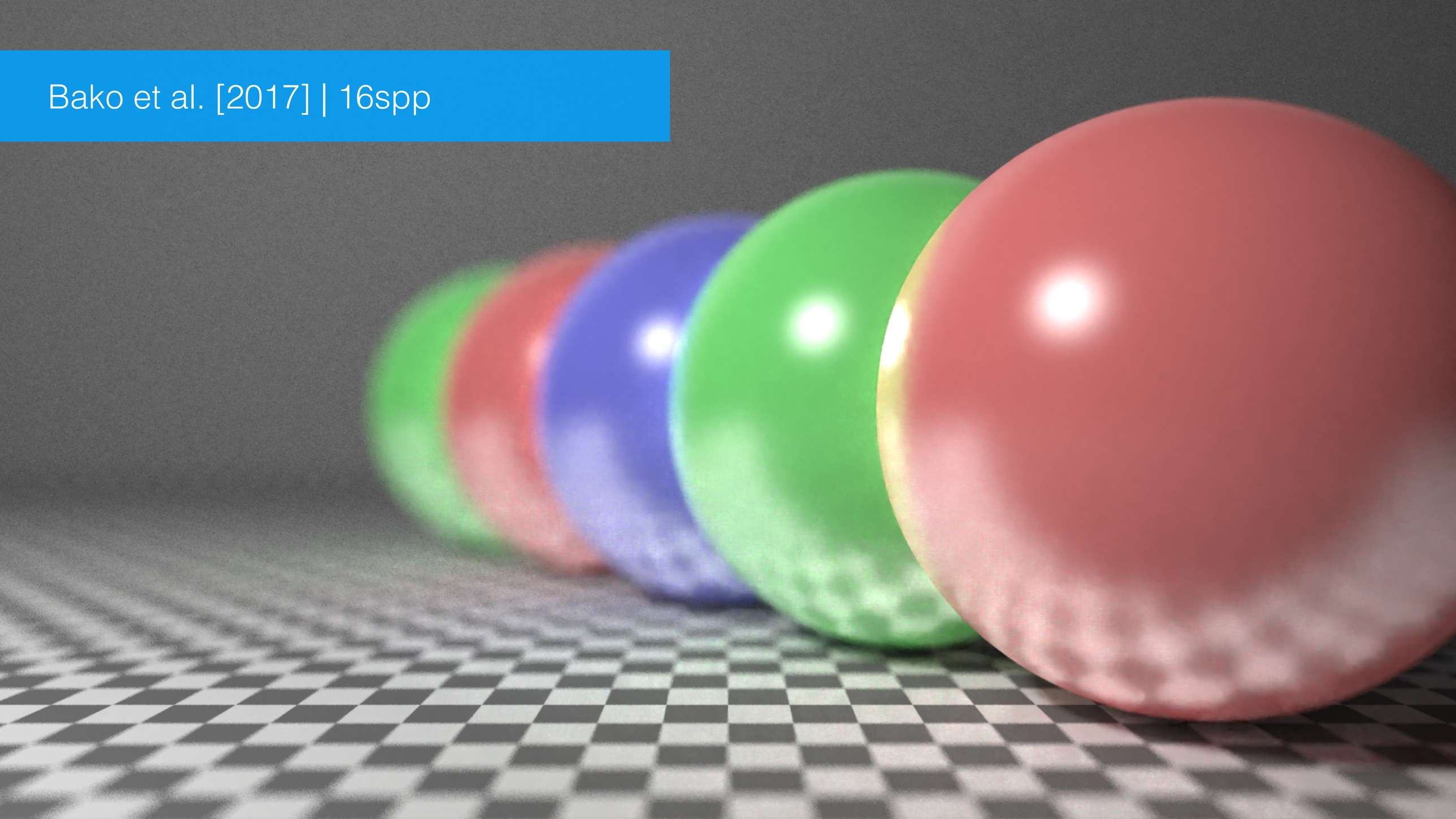
ours | 8spp

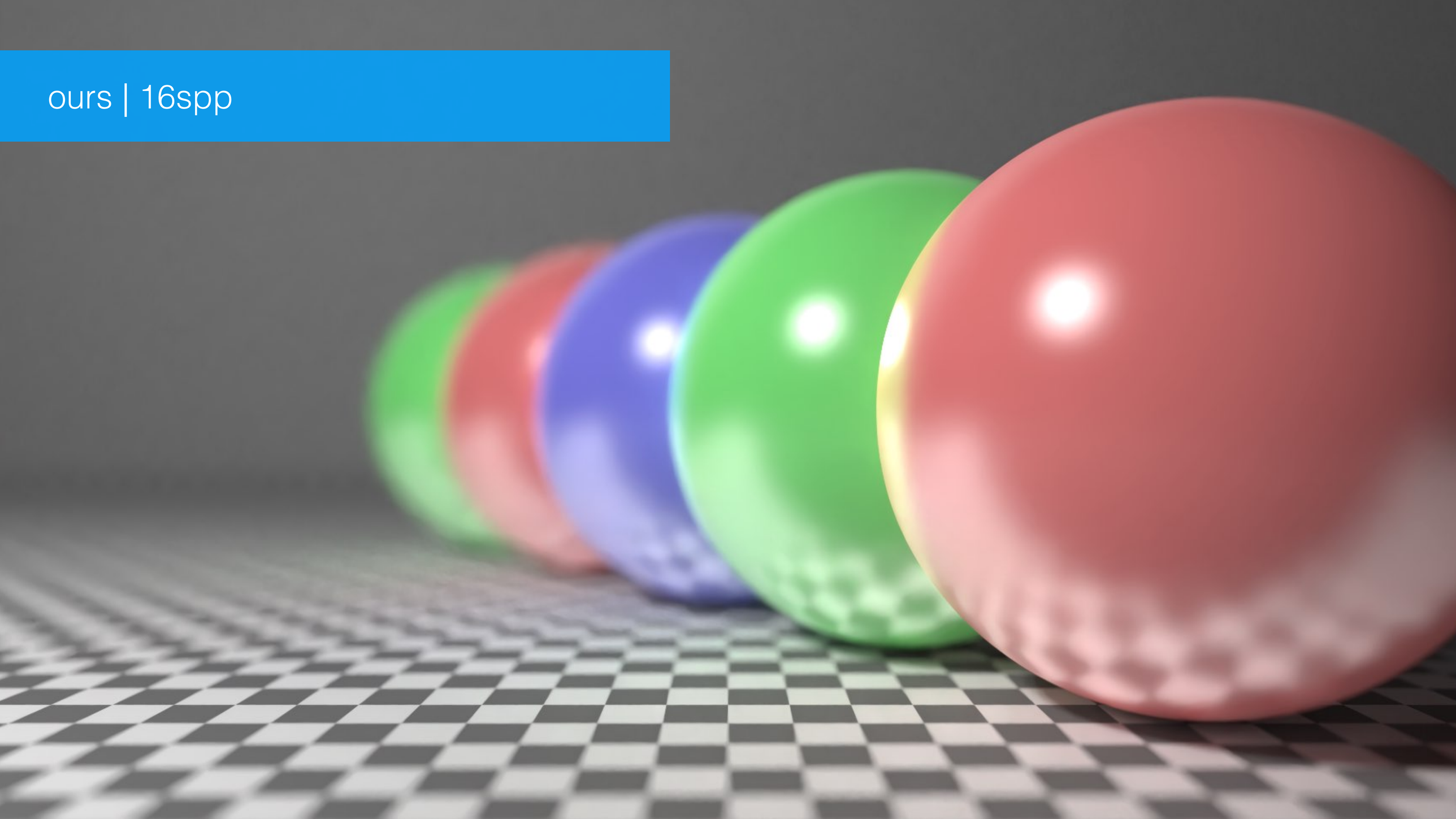


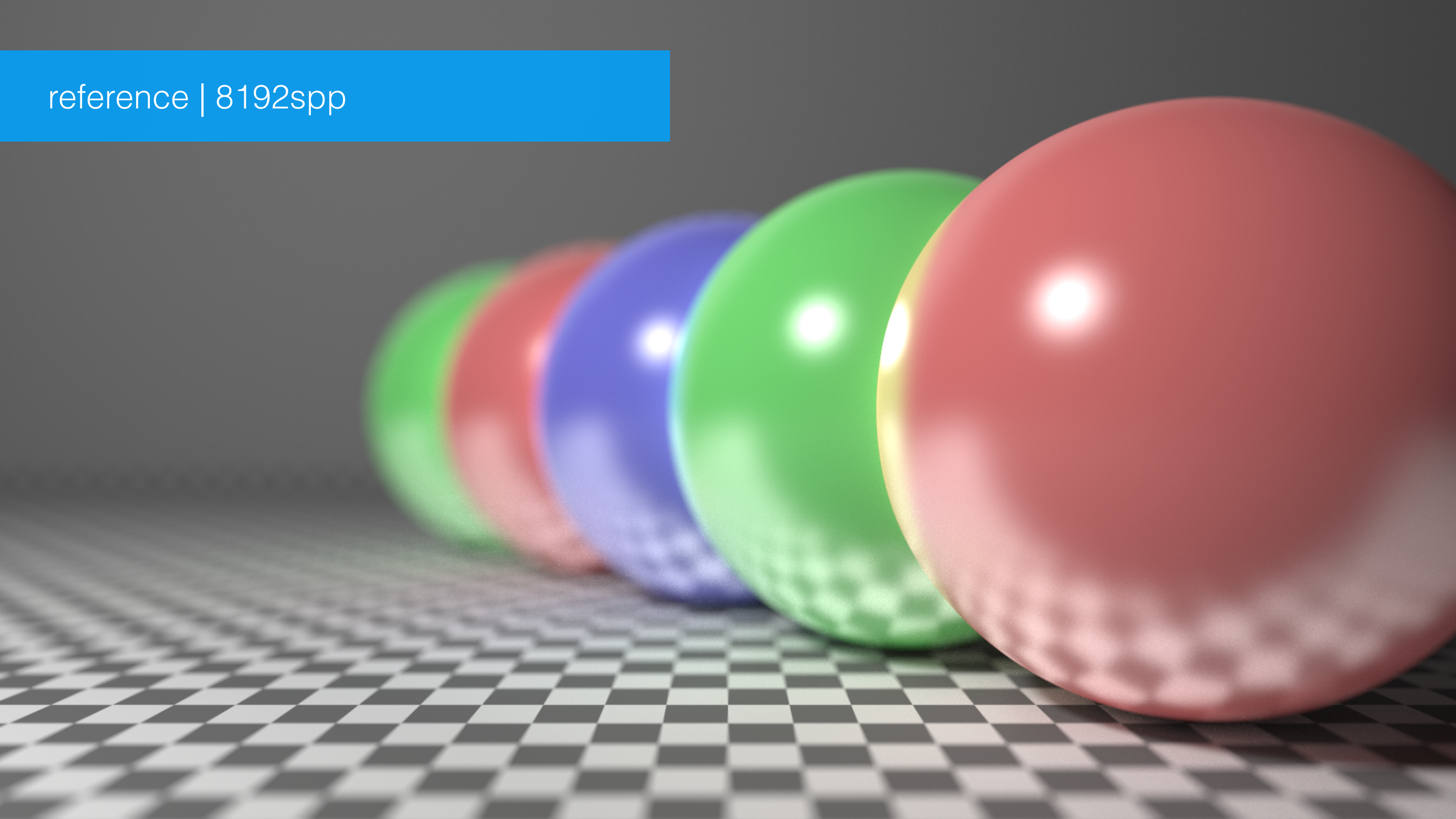
reference | 8192spp

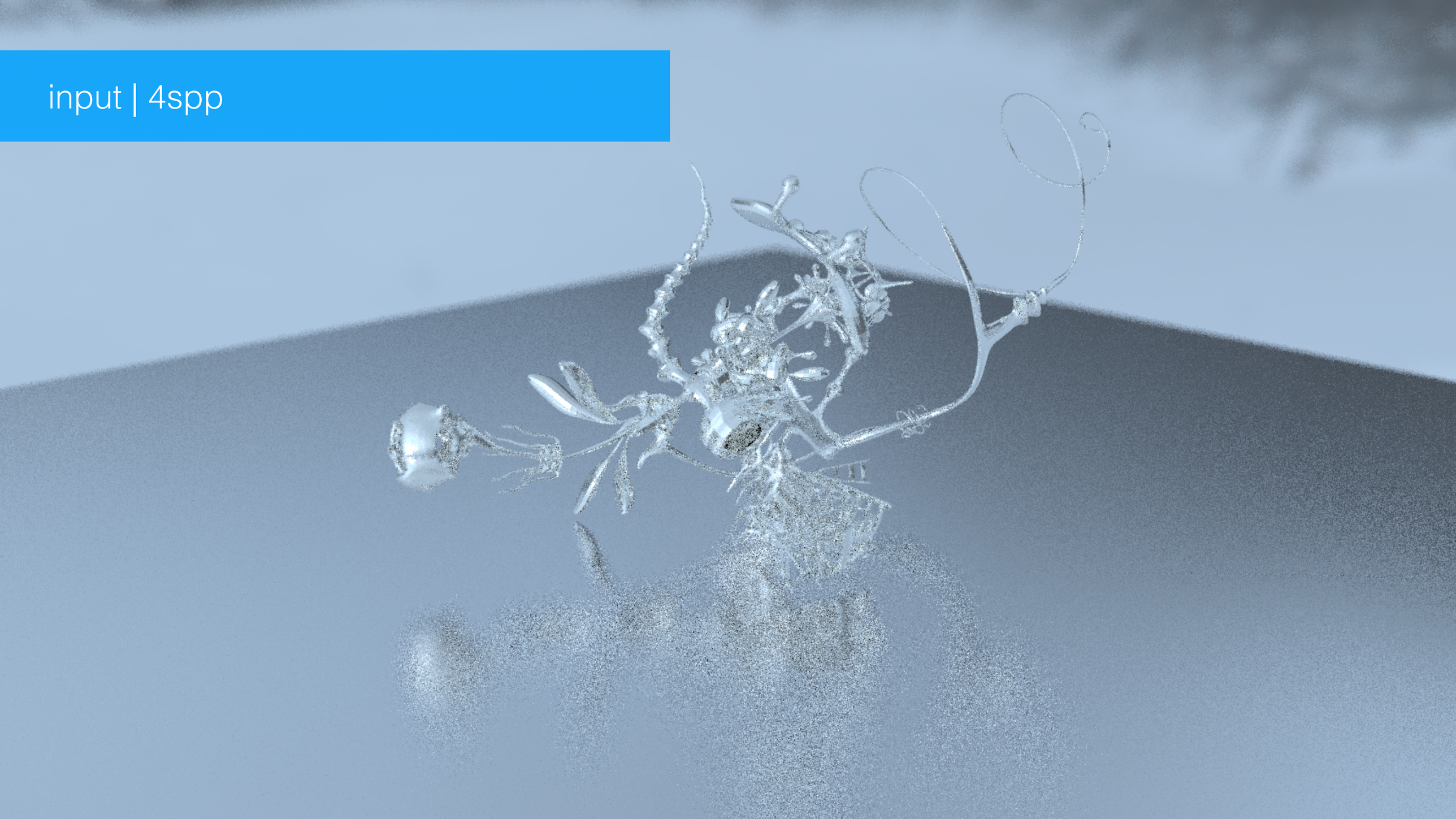


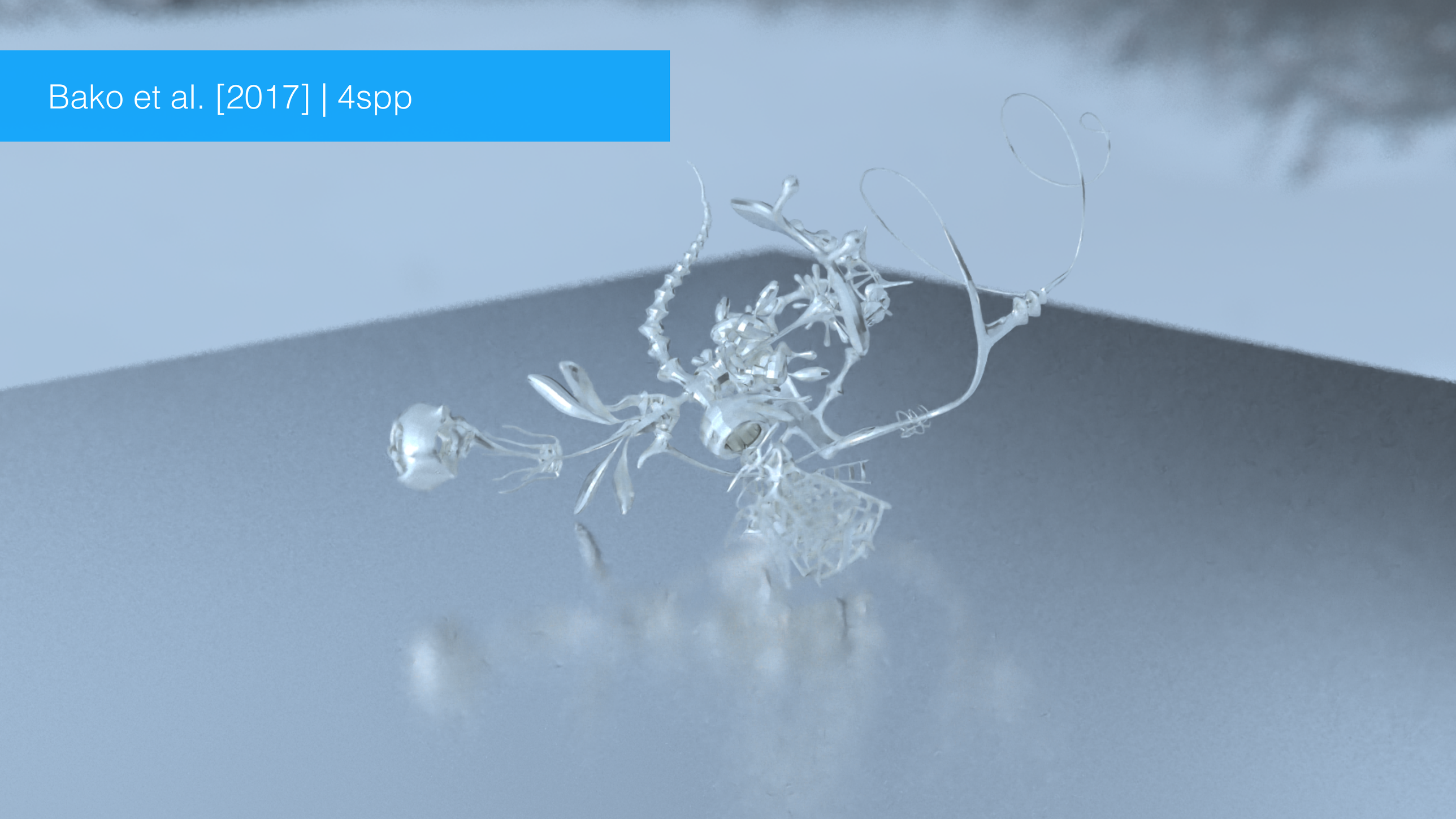

input | 16spp

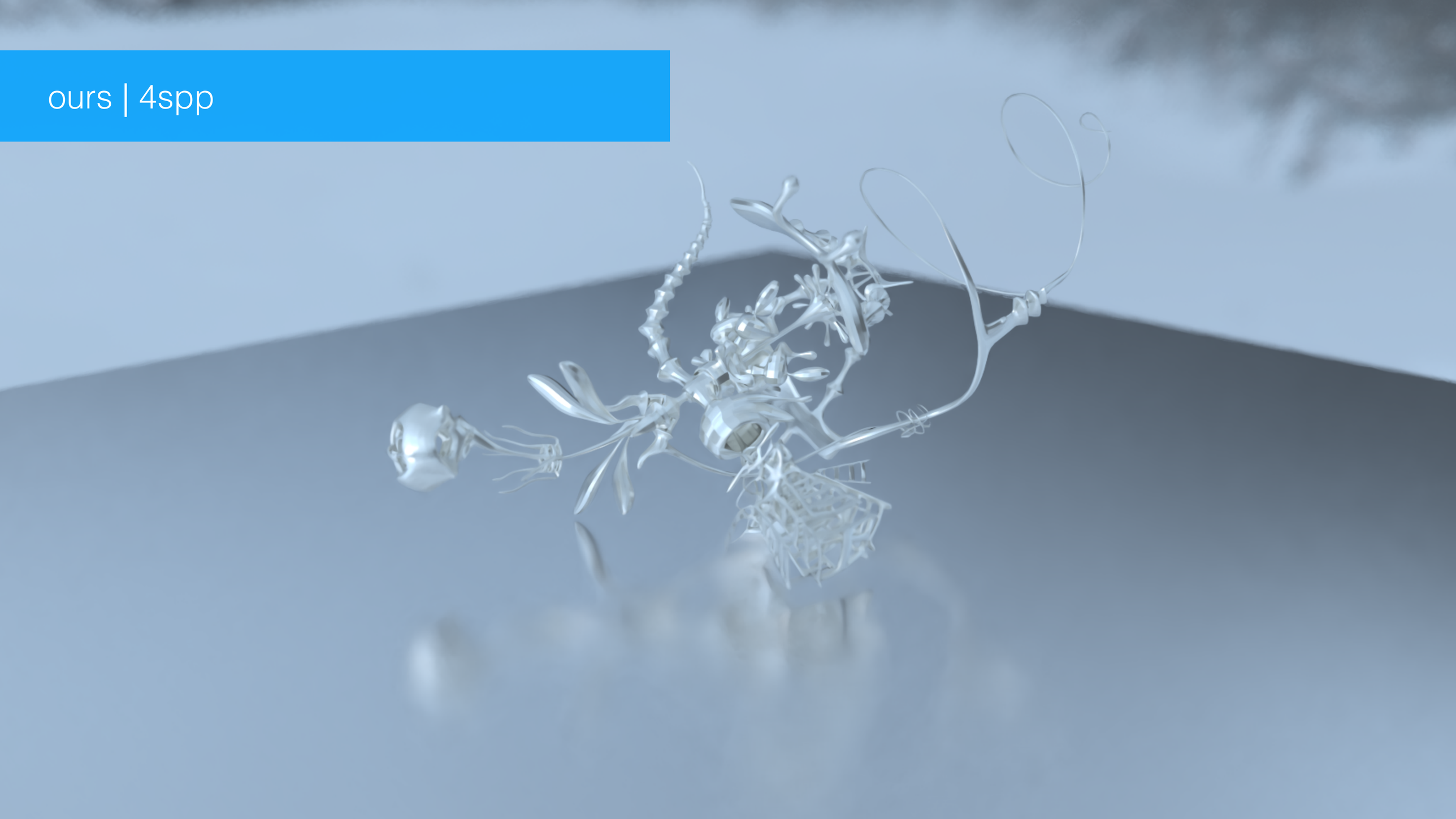

Bako et al. [2017] | 16spp

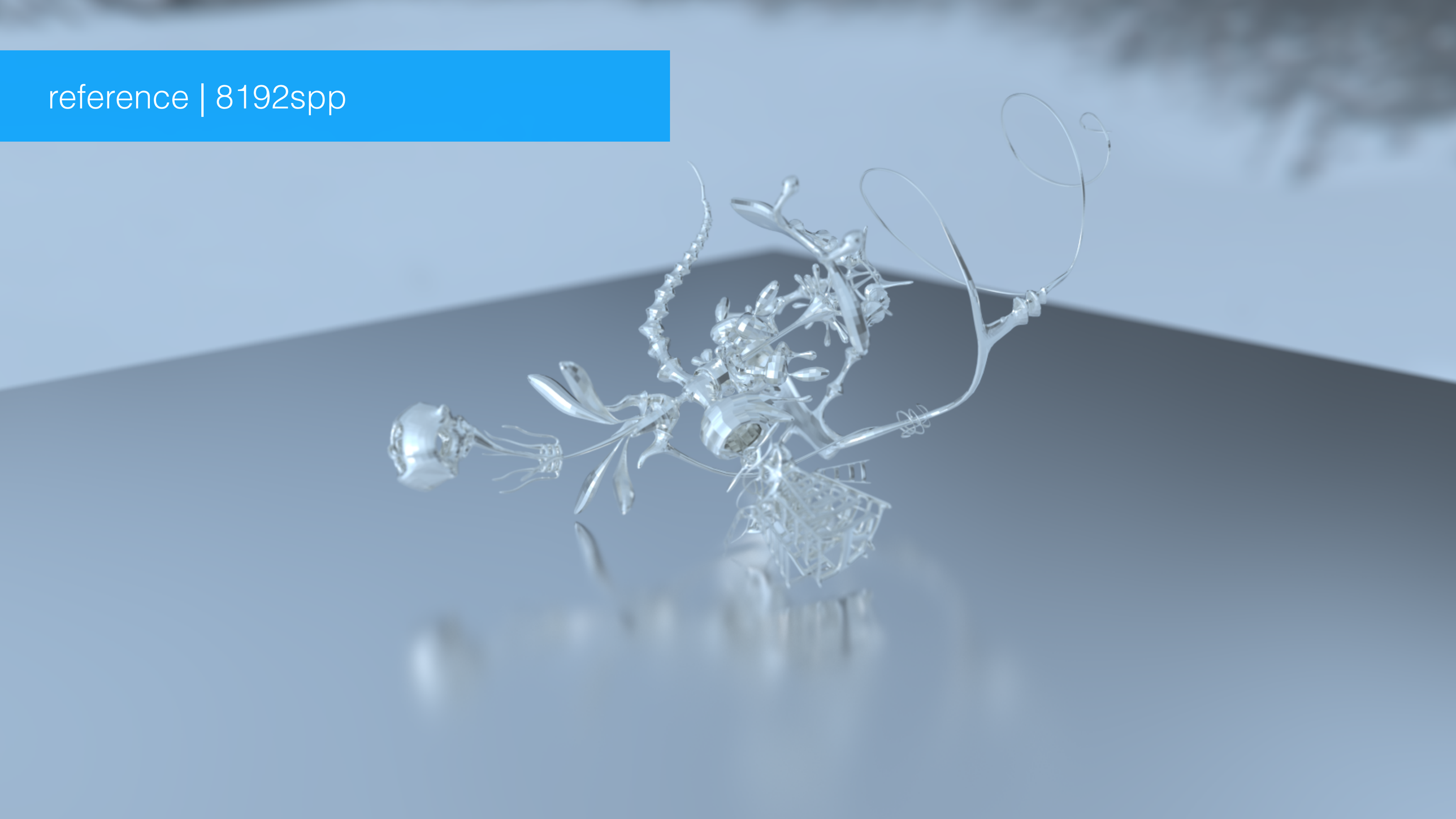

ours | 16spp


reference | 8192spp


input | 4spp


Bako et al. [2017] | 4spp


ours | 4spp


reference | 8192spp


input | 4spp


Bako et al. [2017] | 4spp


ours | 4spp

reference | 8192spp