

Light-Cone Shell Queries for Scalable Time-Gated Rendering



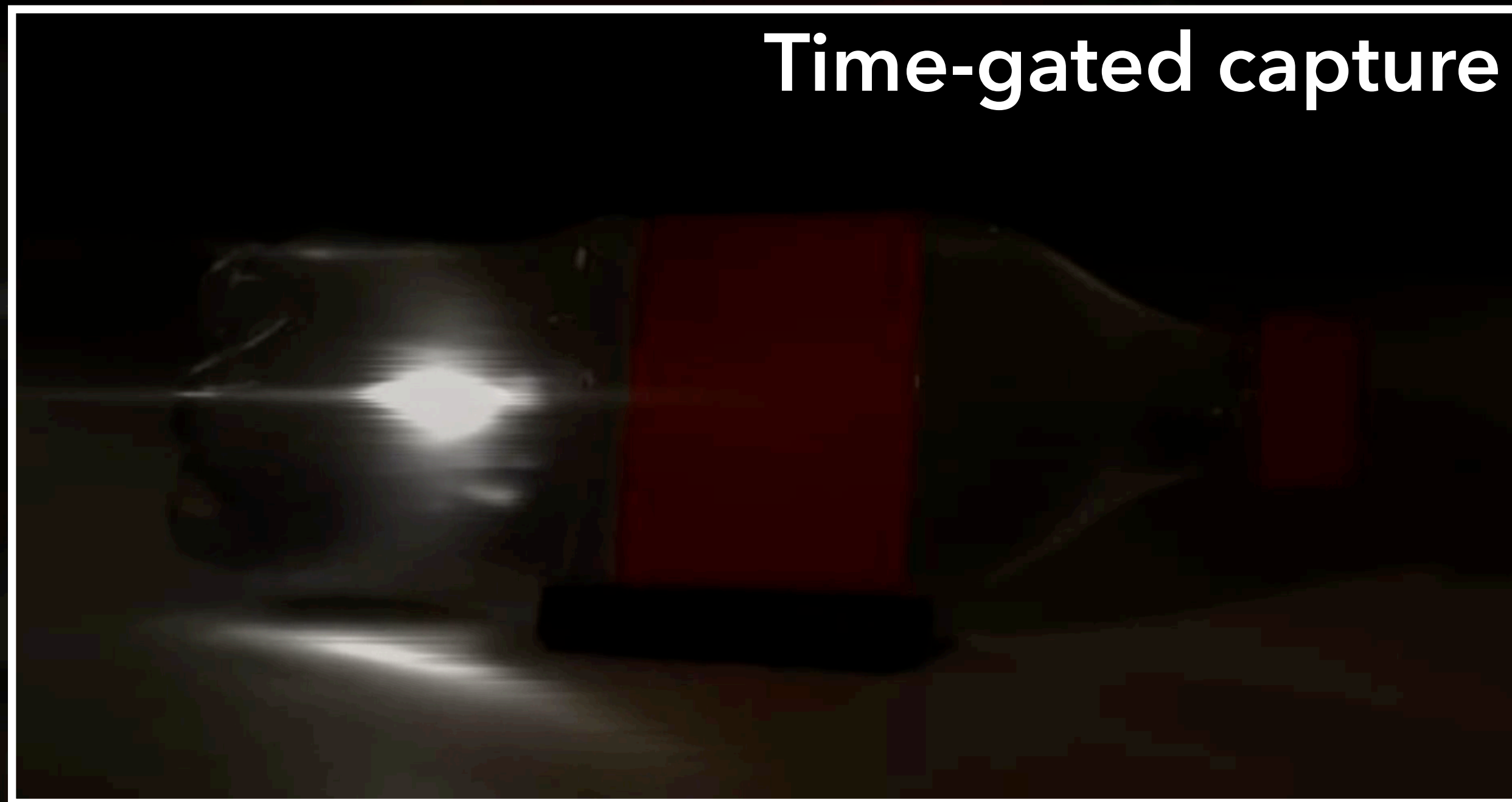
Jack Cui, Juhyeon Kim, Adithya Pediredla, Wojciech Jarosz
Dartmouth College

RGB camera



"Laser pulse shooting through a bottle and visualized at a trillion frames per second" <https://www.youtube.com/watch?v=-fSqFWcb4rE>

Time-gated capture



Camera Culture Group, MIT Media Lab

Time-gated cameras

RGB Camera



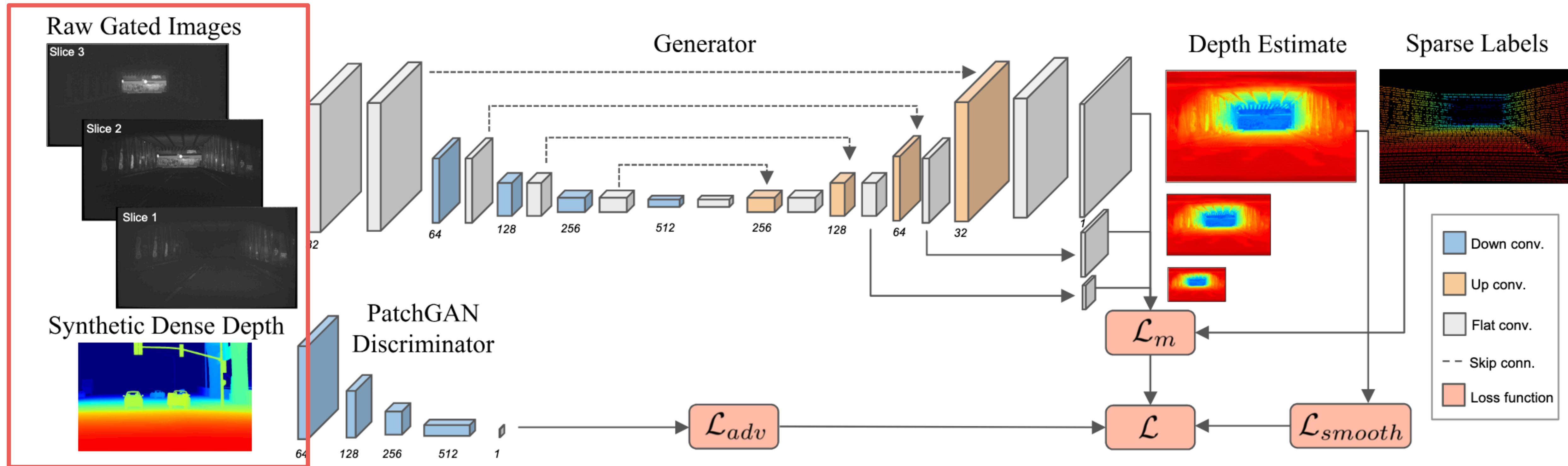
Gated Camera



Gruber et al. 2019, "Gated2Depth: Real-time Dense Lidar from Gated Images"

Why simulate?

Gruber et al. 2019, "Gated2Depth: Real-time Dense Lidar from Gated Images"

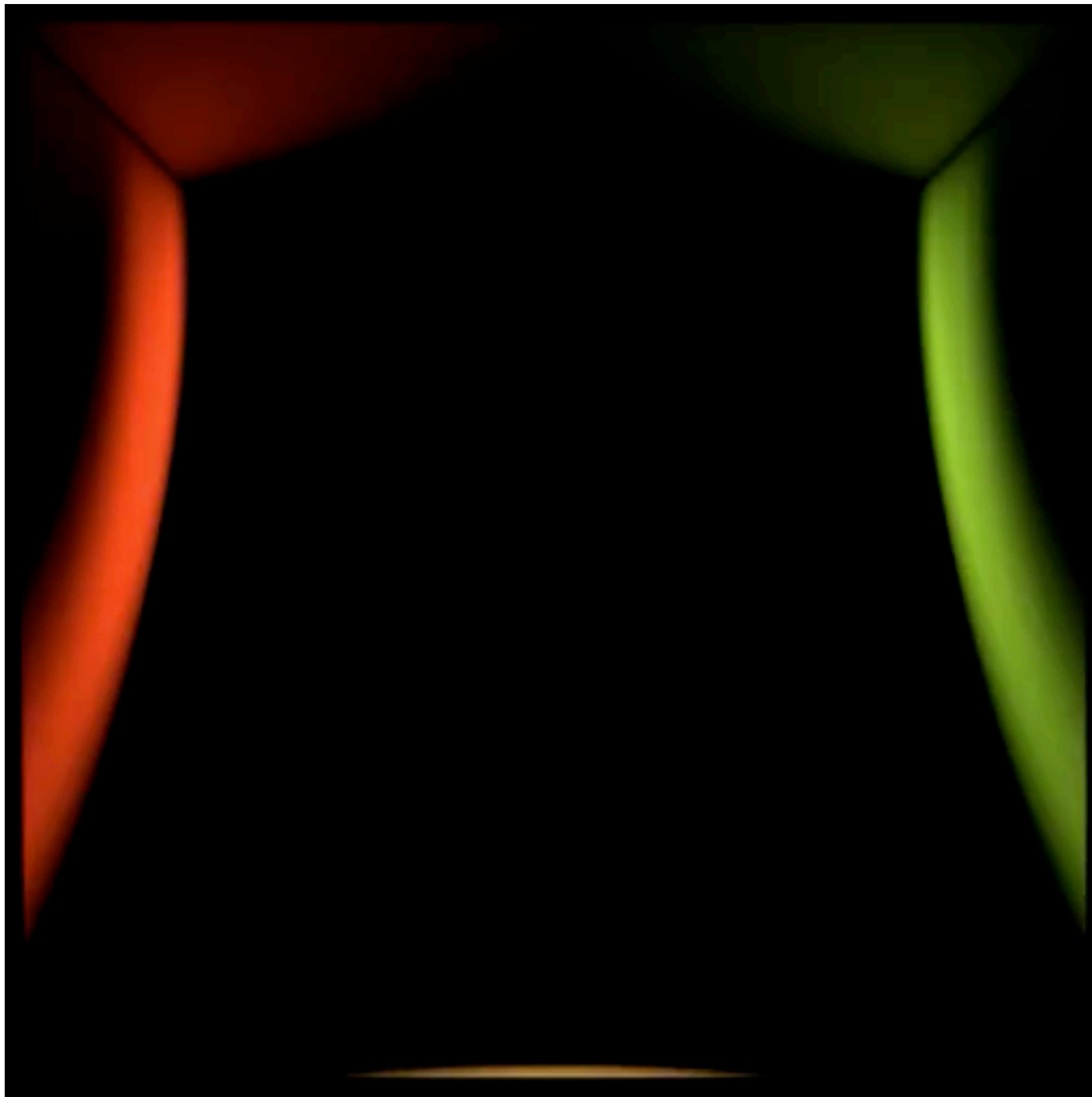


Gated images paired with ground-truth depth are required

Data scarce!

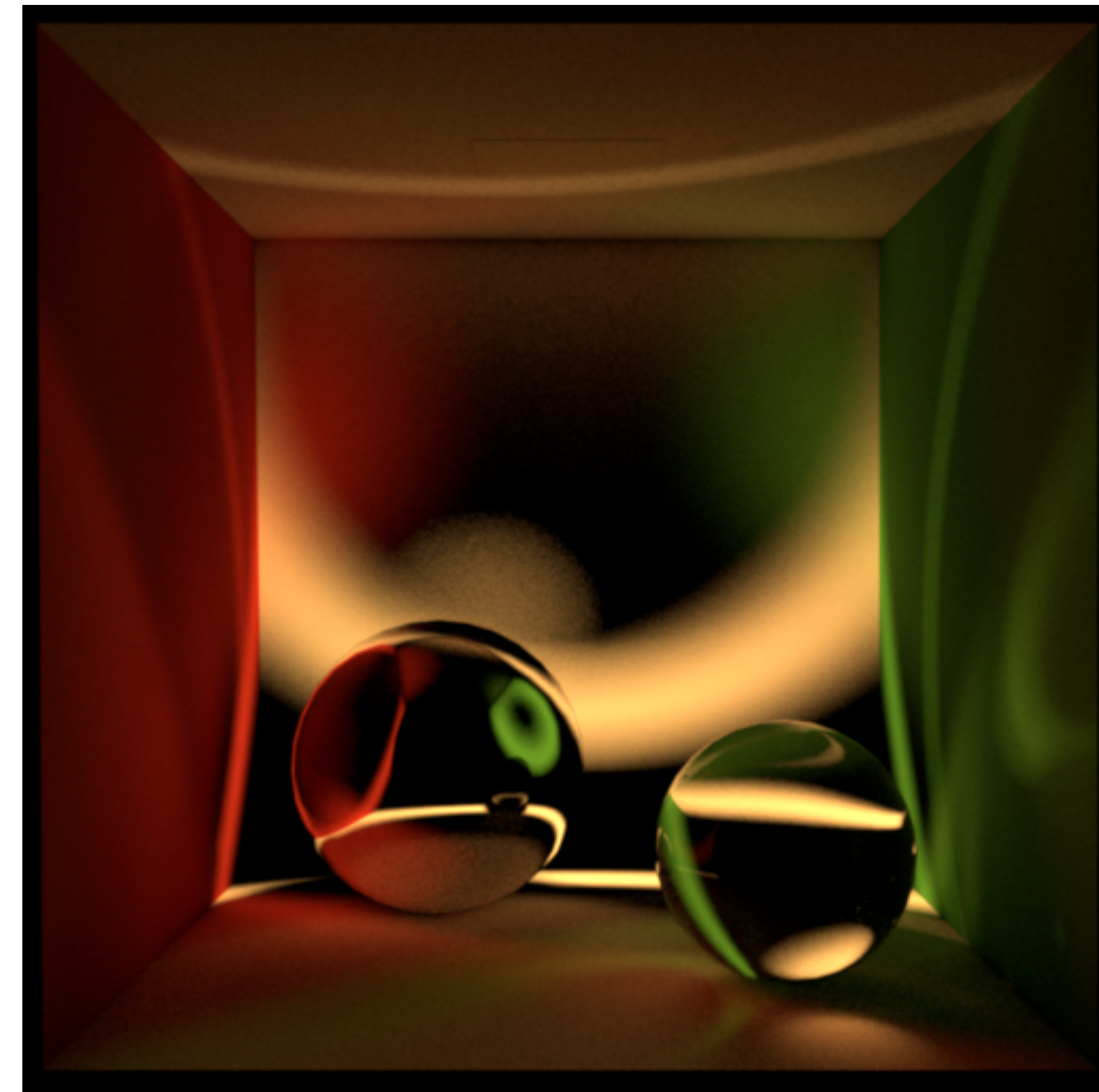
Time-of-flight rendering

Transient



[Jarabo et al. 2014] [Marco et al. 2017] [Liu et al. 2022] [He et al. 2024]

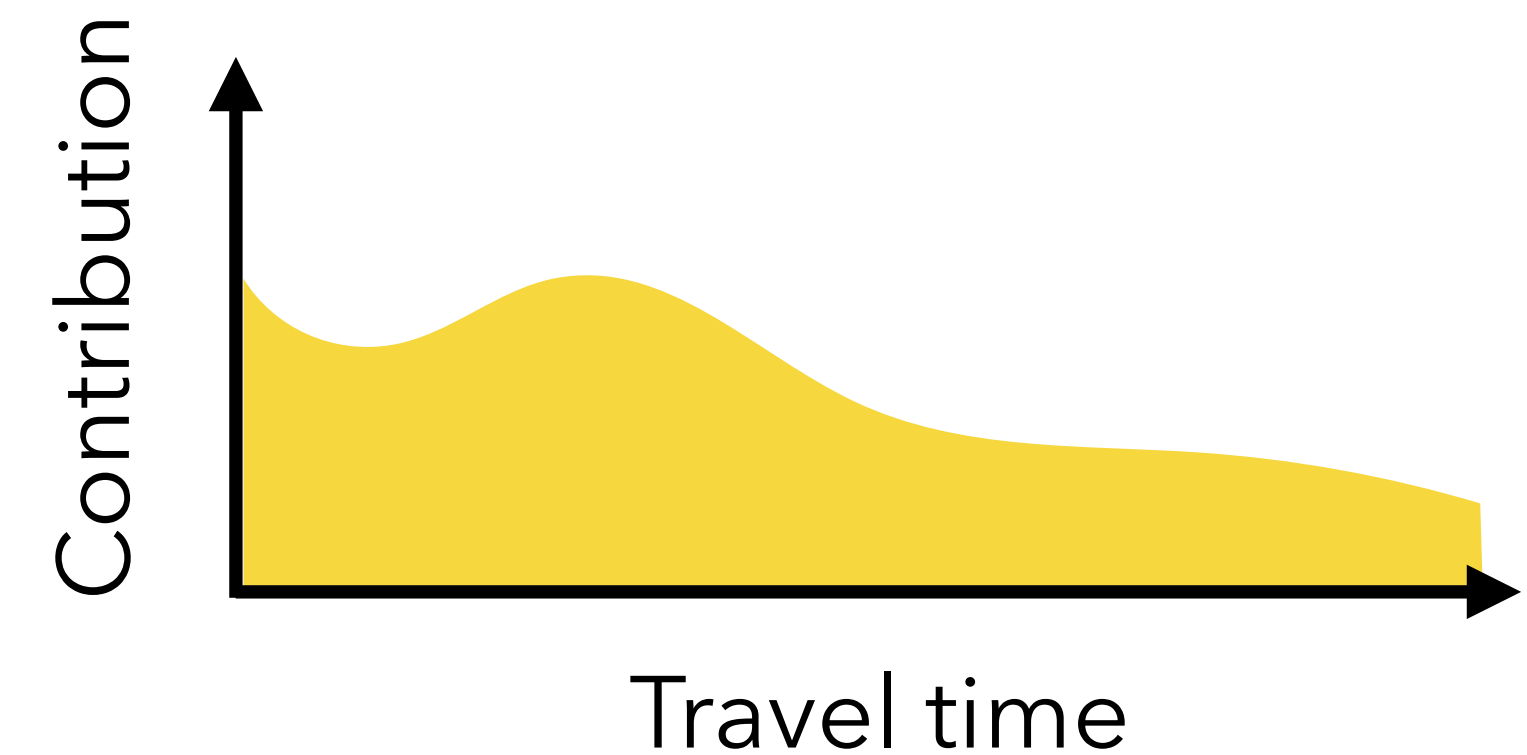
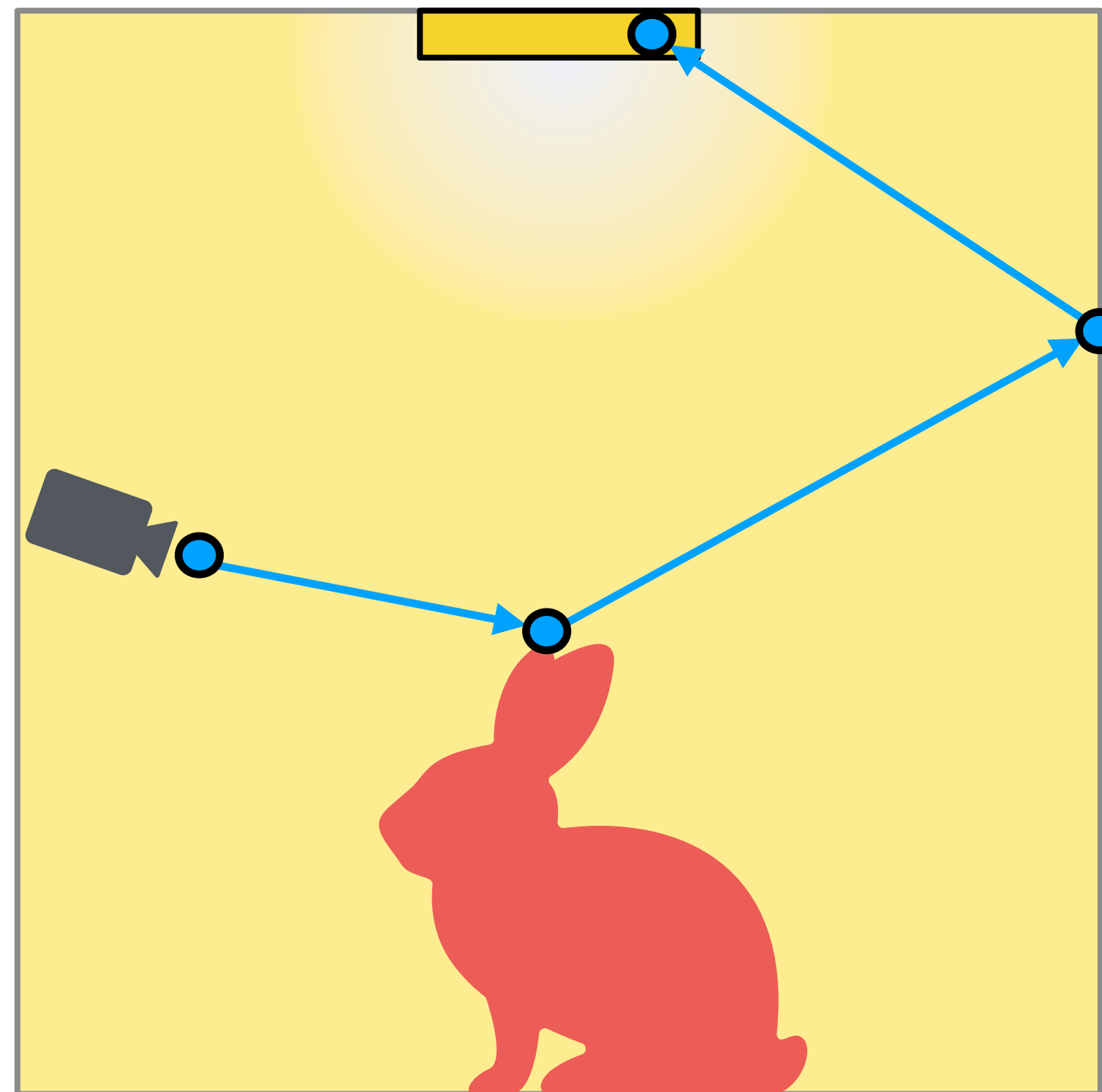
Time-gated



[Pediredla et al. 2019]

Steady-state rendering

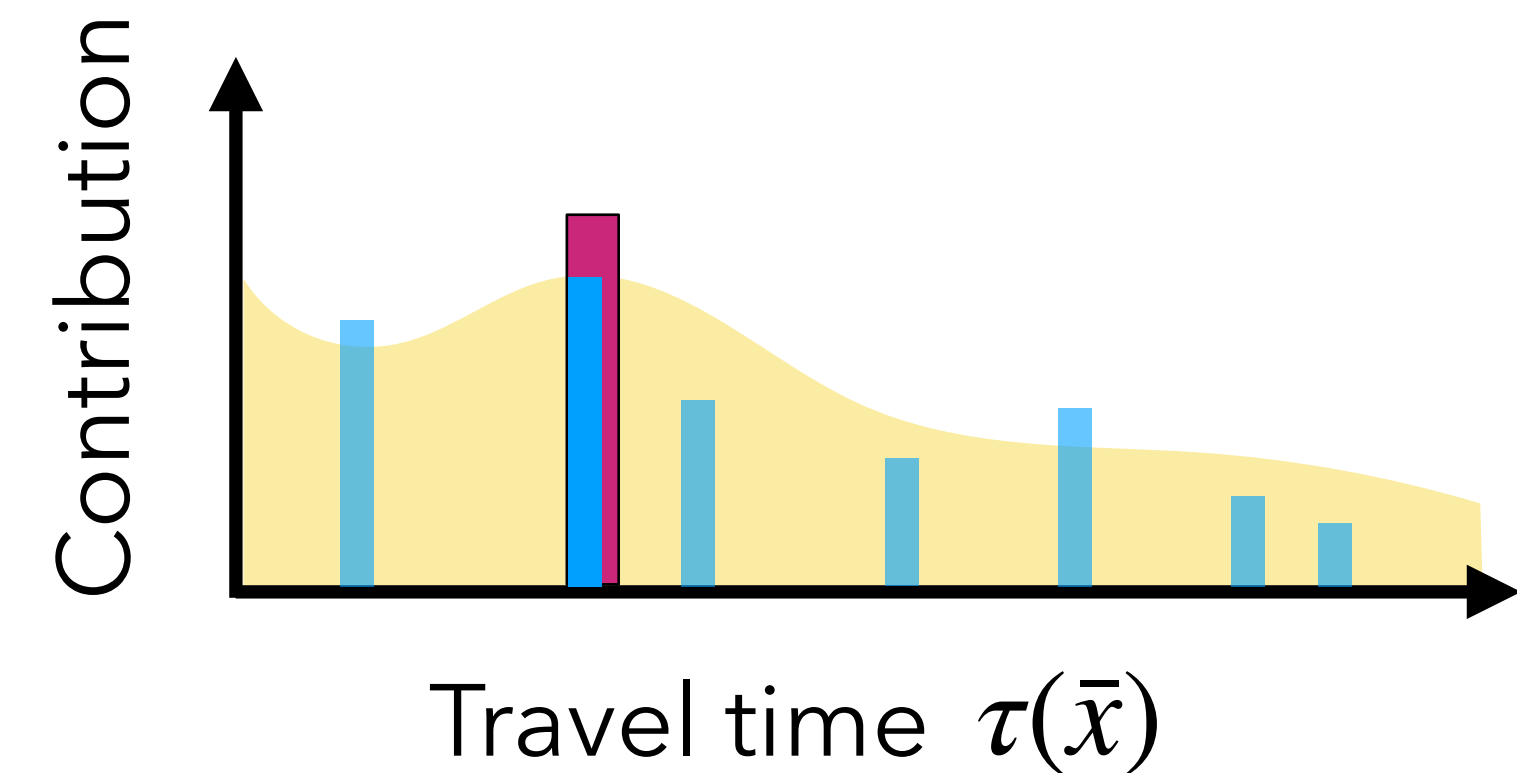
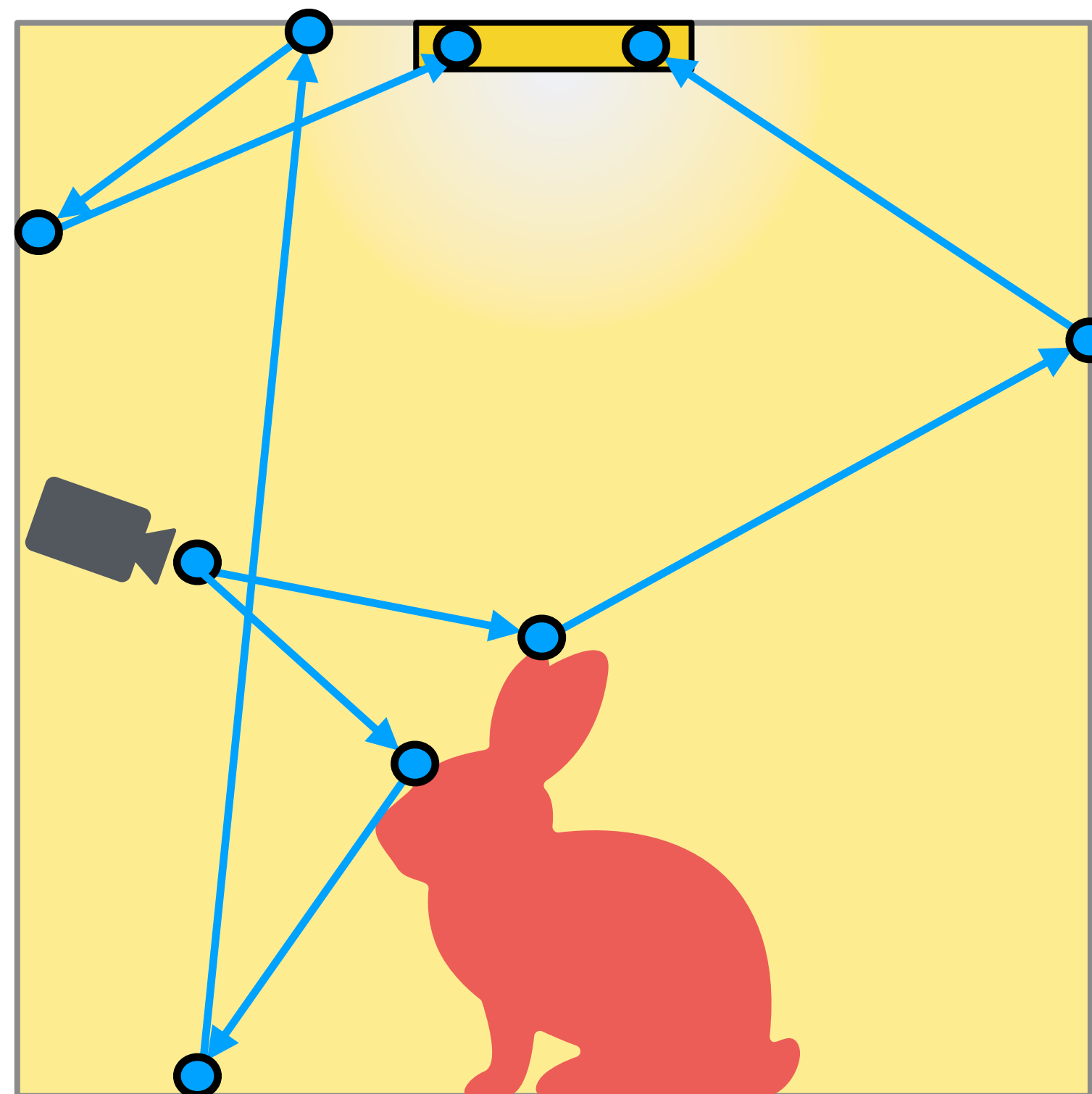
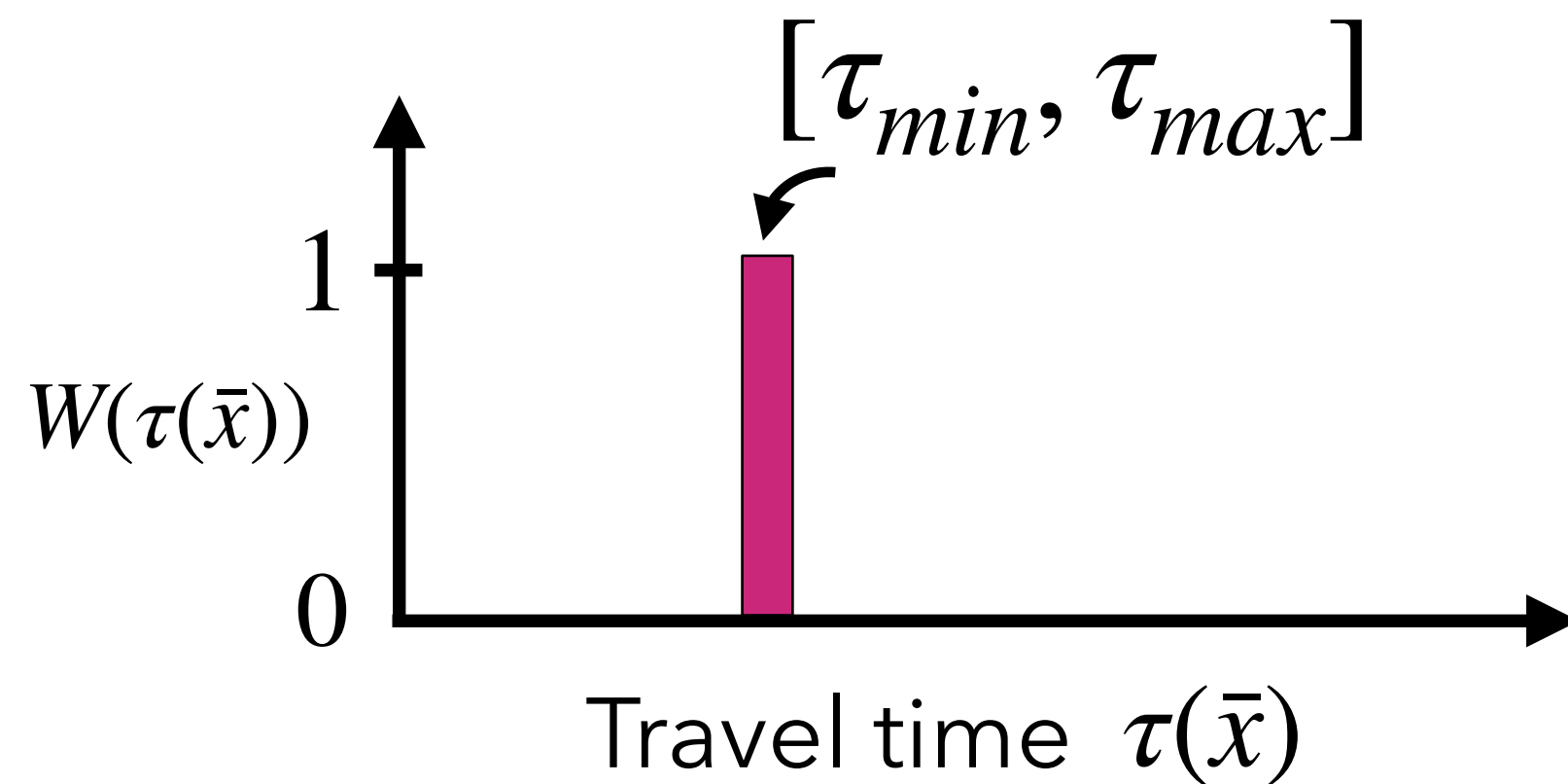
Steady-state $I = \int_{\Omega} f(\bar{x}) d\mu(\bar{x})$



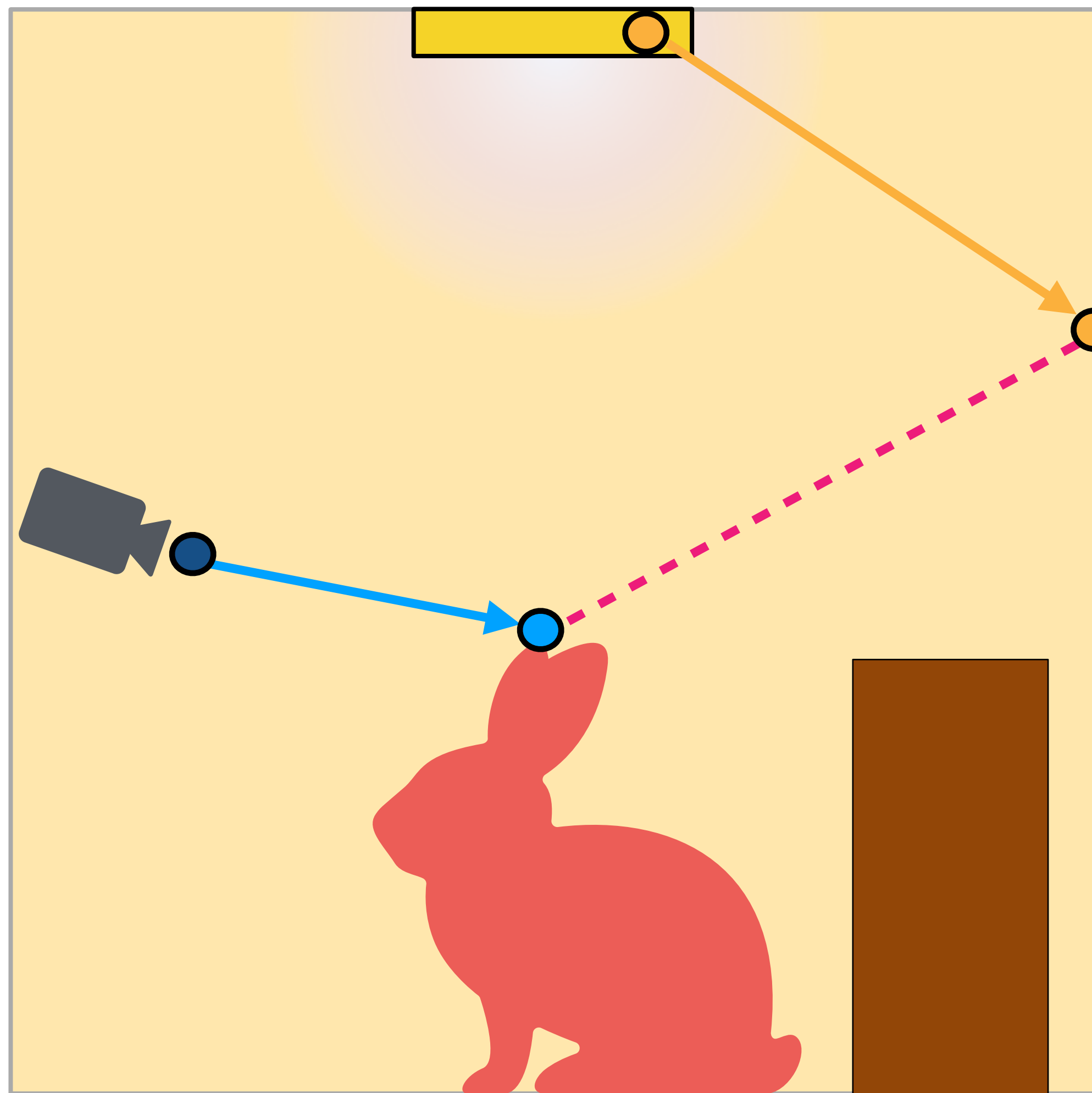
Time-gated rendering

Time-gated $I_\tau = \int_{\Omega} W(\tau(\bar{x})) f(\bar{x}) d\mu(\bar{x})$

Travel-time importance term



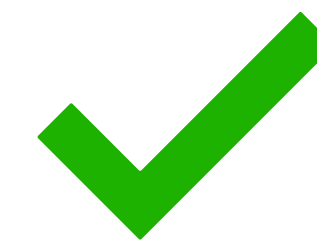
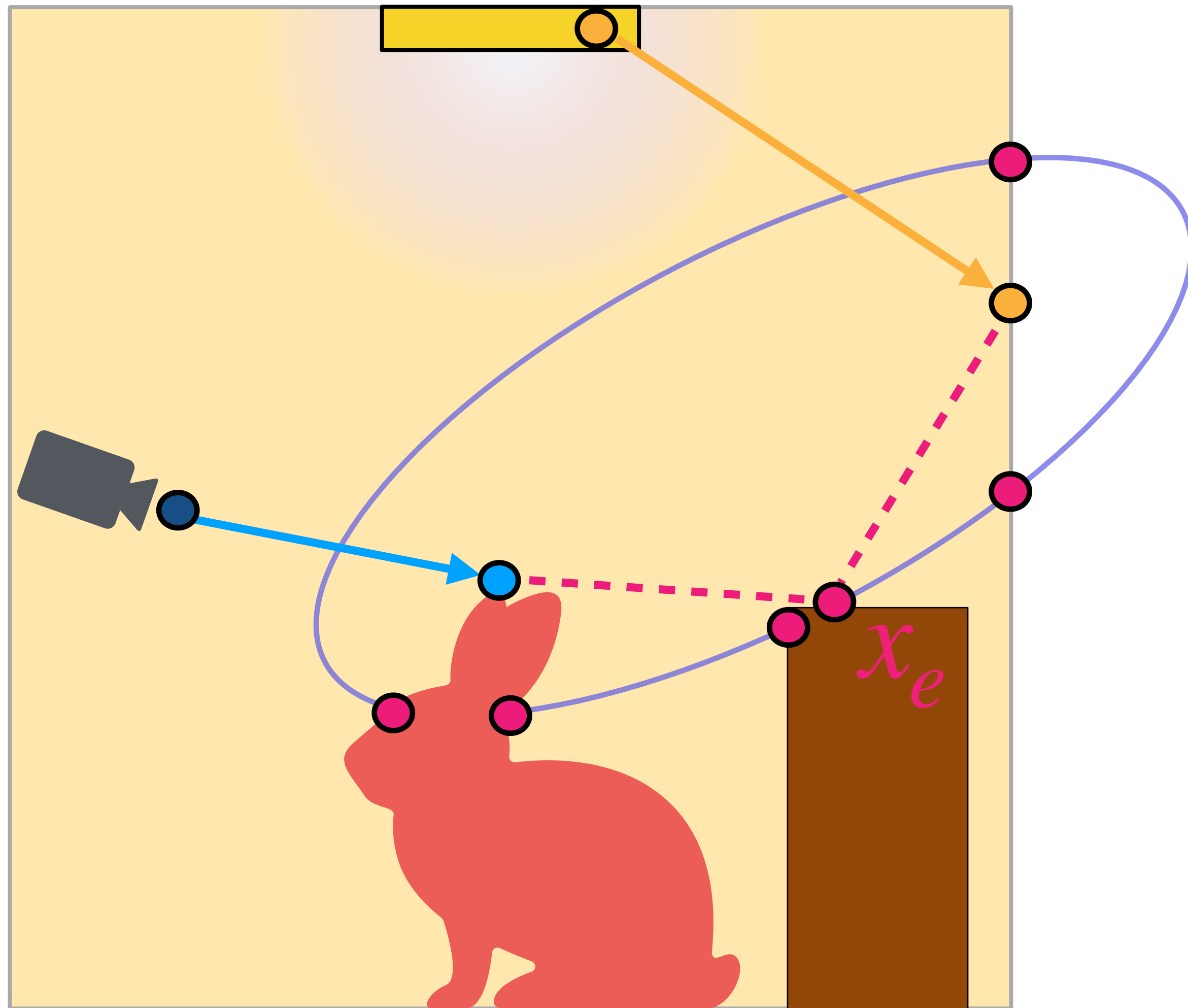
Why naive sampling fails



No control over travel time!

Ellipsoidal path connections (EPC)

"Ellipsoidal Path Connections for Time-Gated Rendering" [Pediredla et al. 2019]



Samples paths of a target travel time

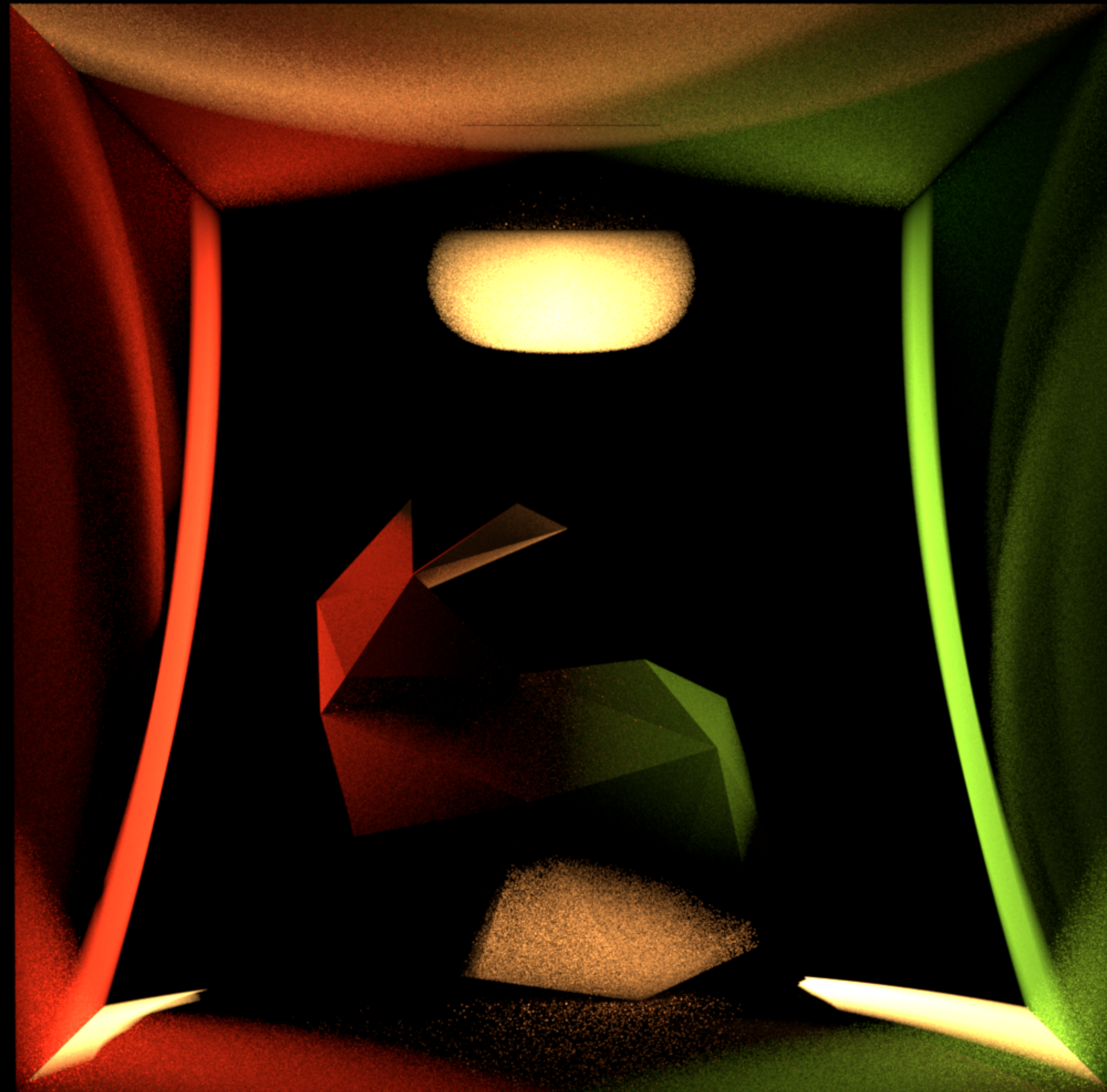
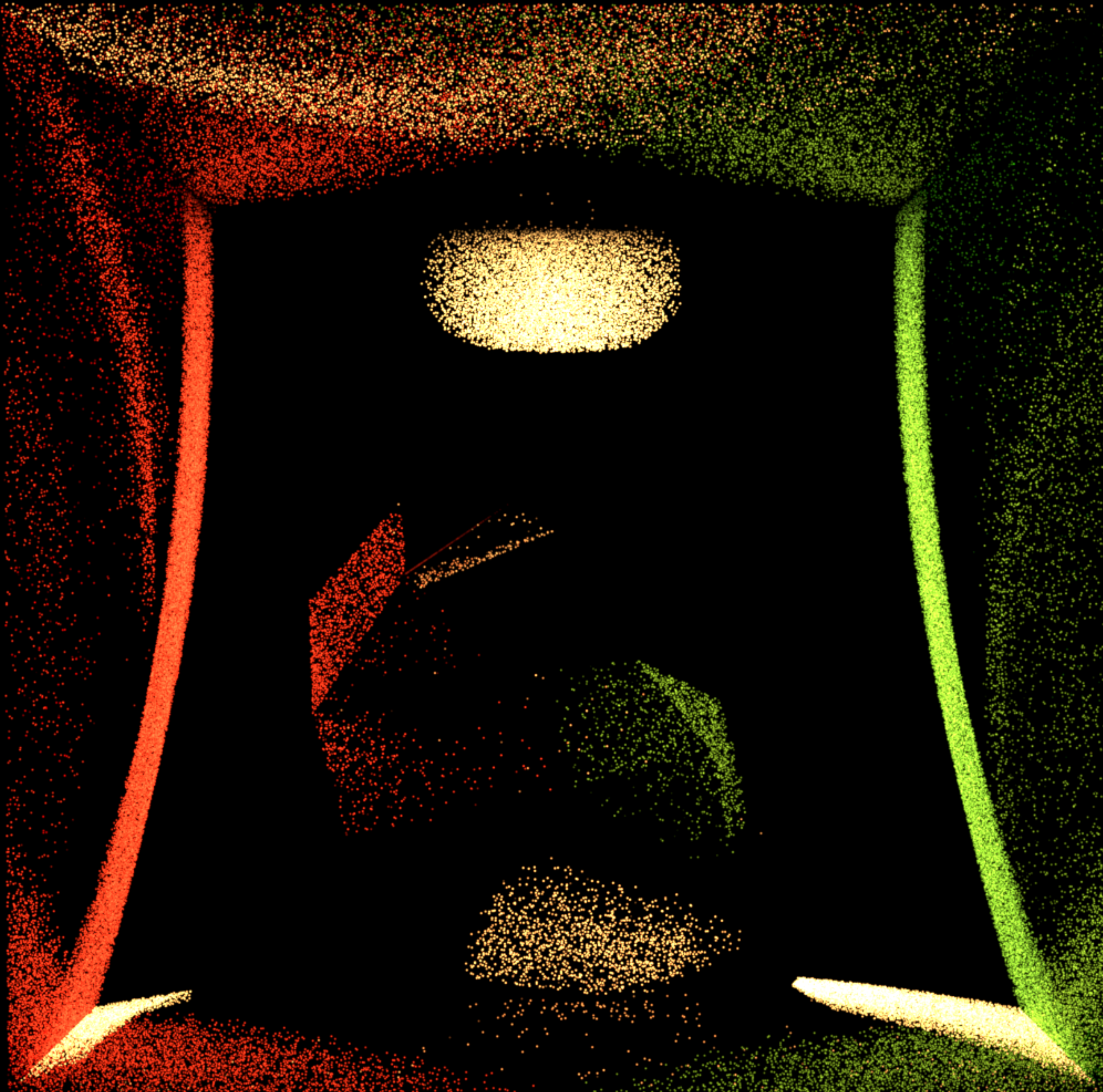


Cost coupled to triangle count and density

BDPT (30s)

23 triangles

EPC (30s)





226k triangles

EPC 1spp (16 min)

BDPT (30s)

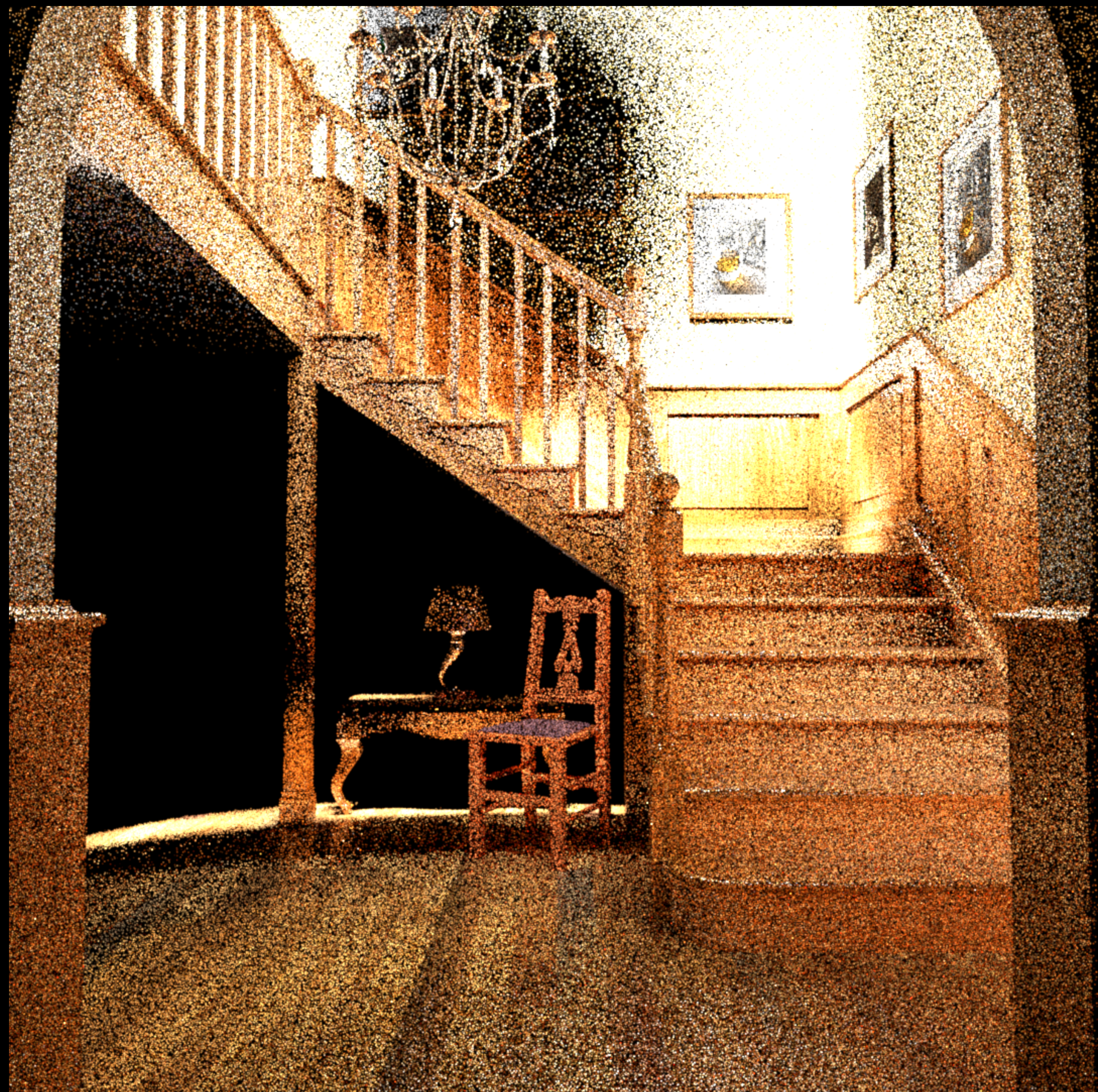


Goal: decouple time-gating cost from
scene complexity

Ours (30s)

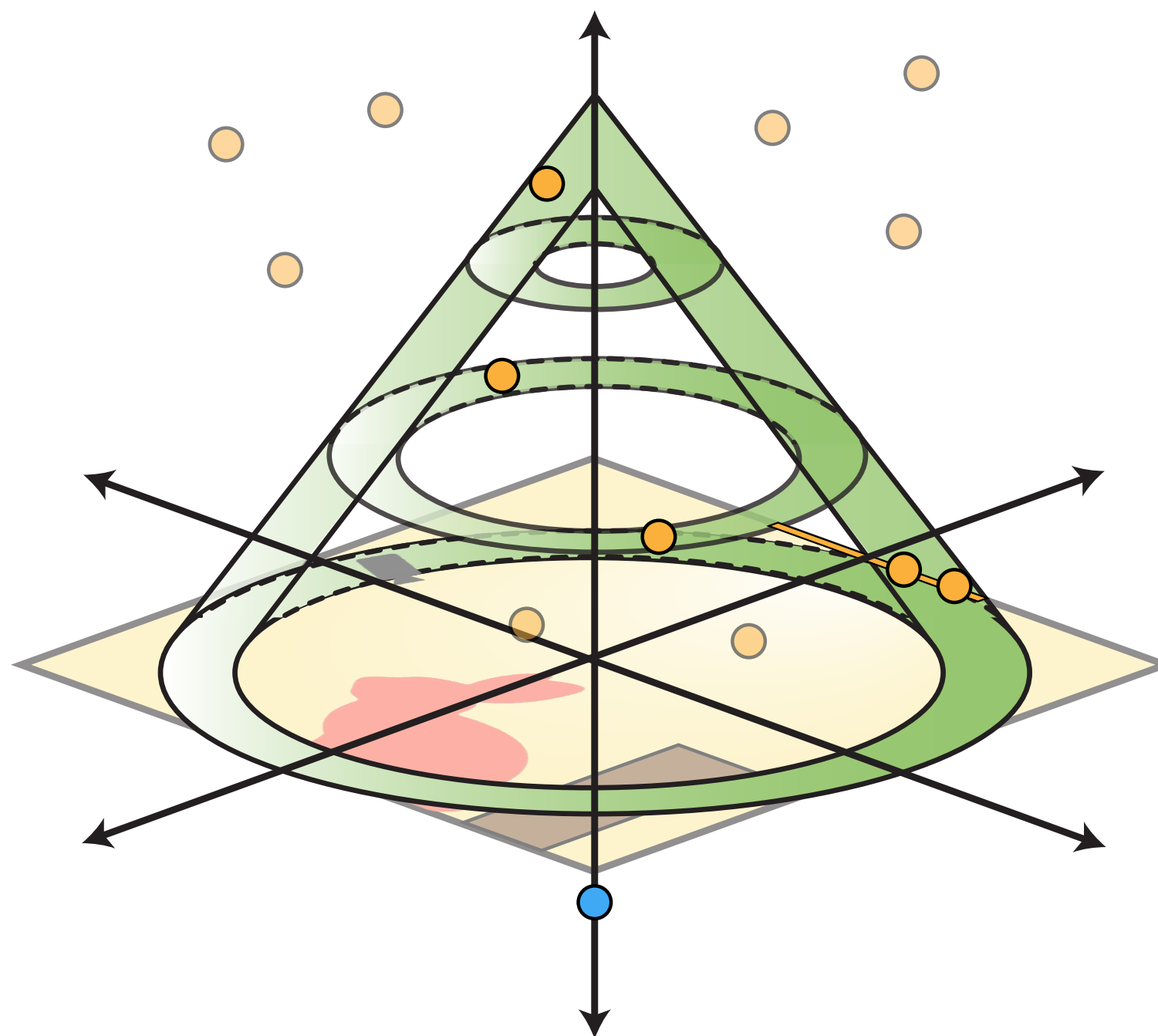


BDPT (30s)



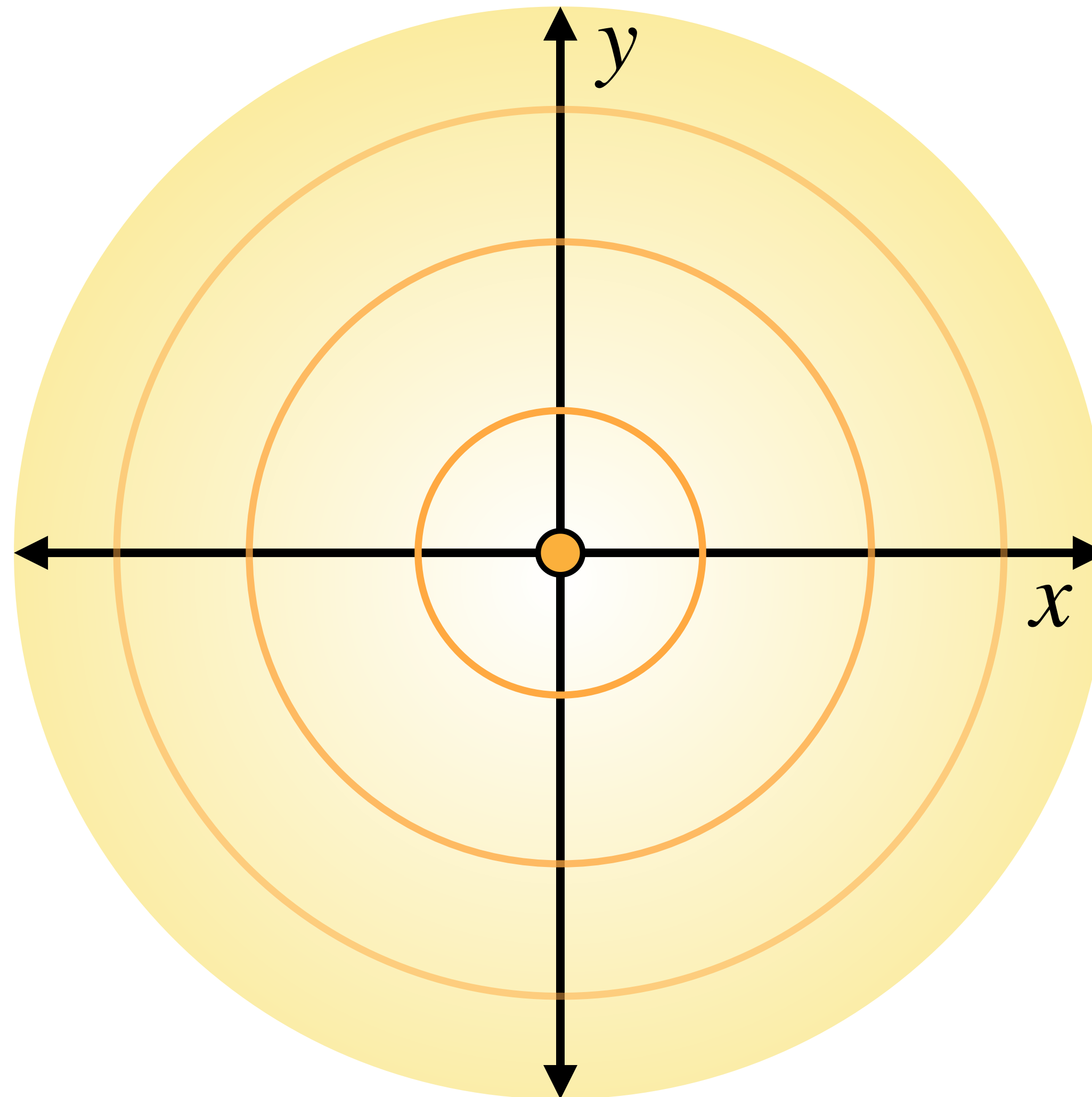
Our contribution

1. Formulating time-gated rendering with light-cone shells
2. Accelerating light-cone shell queries

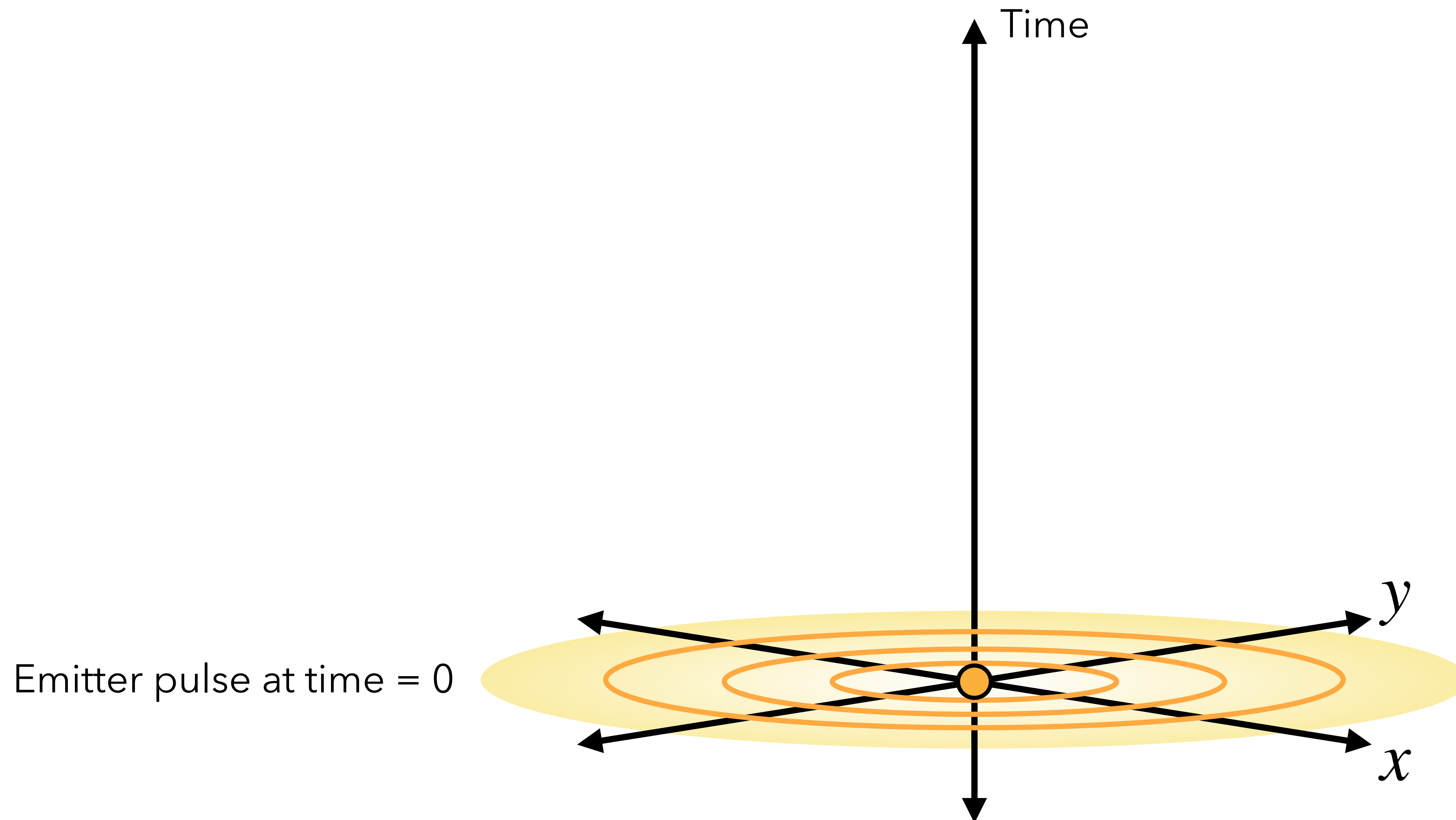


The light cone

Emitter pulse

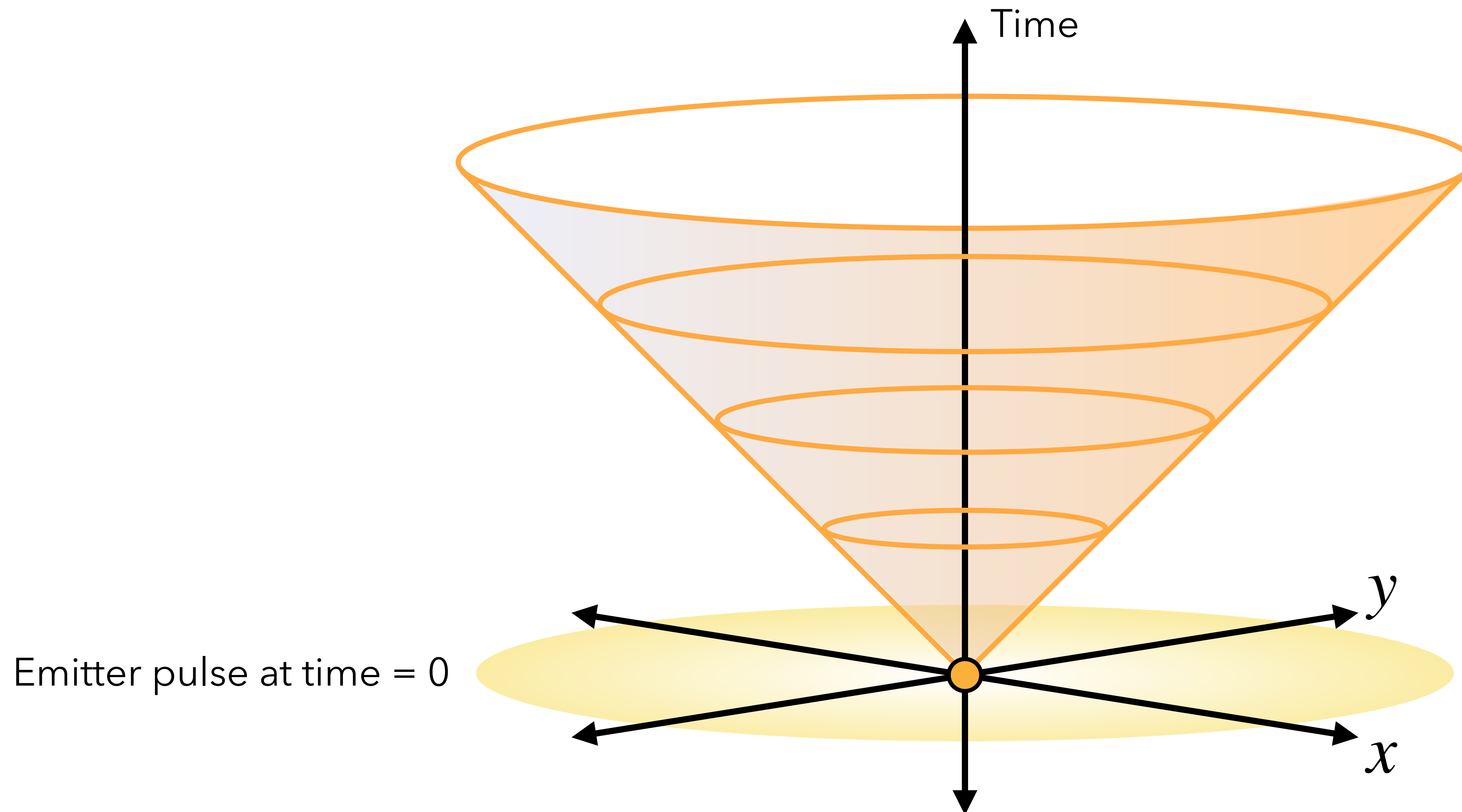


The light cone

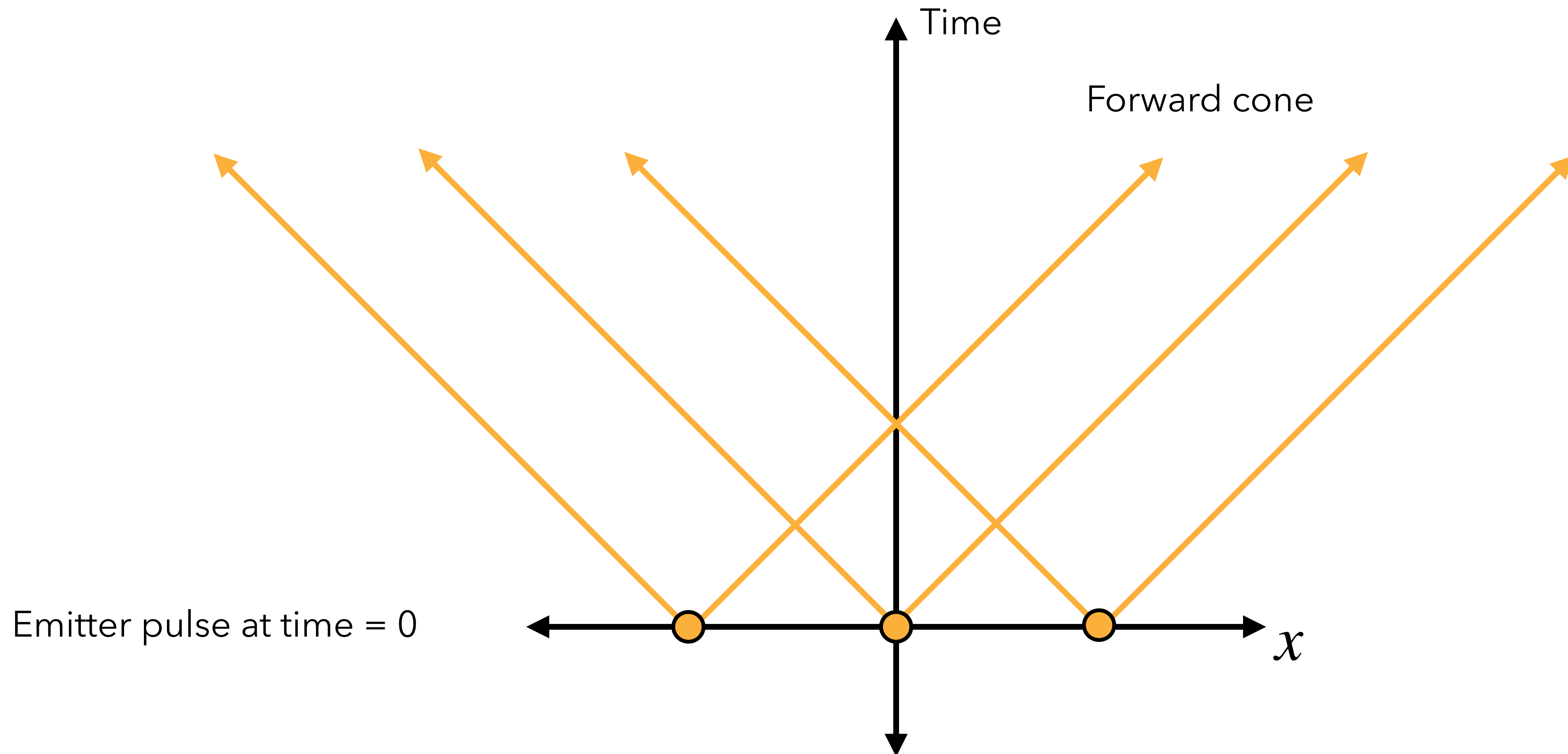


The light cone

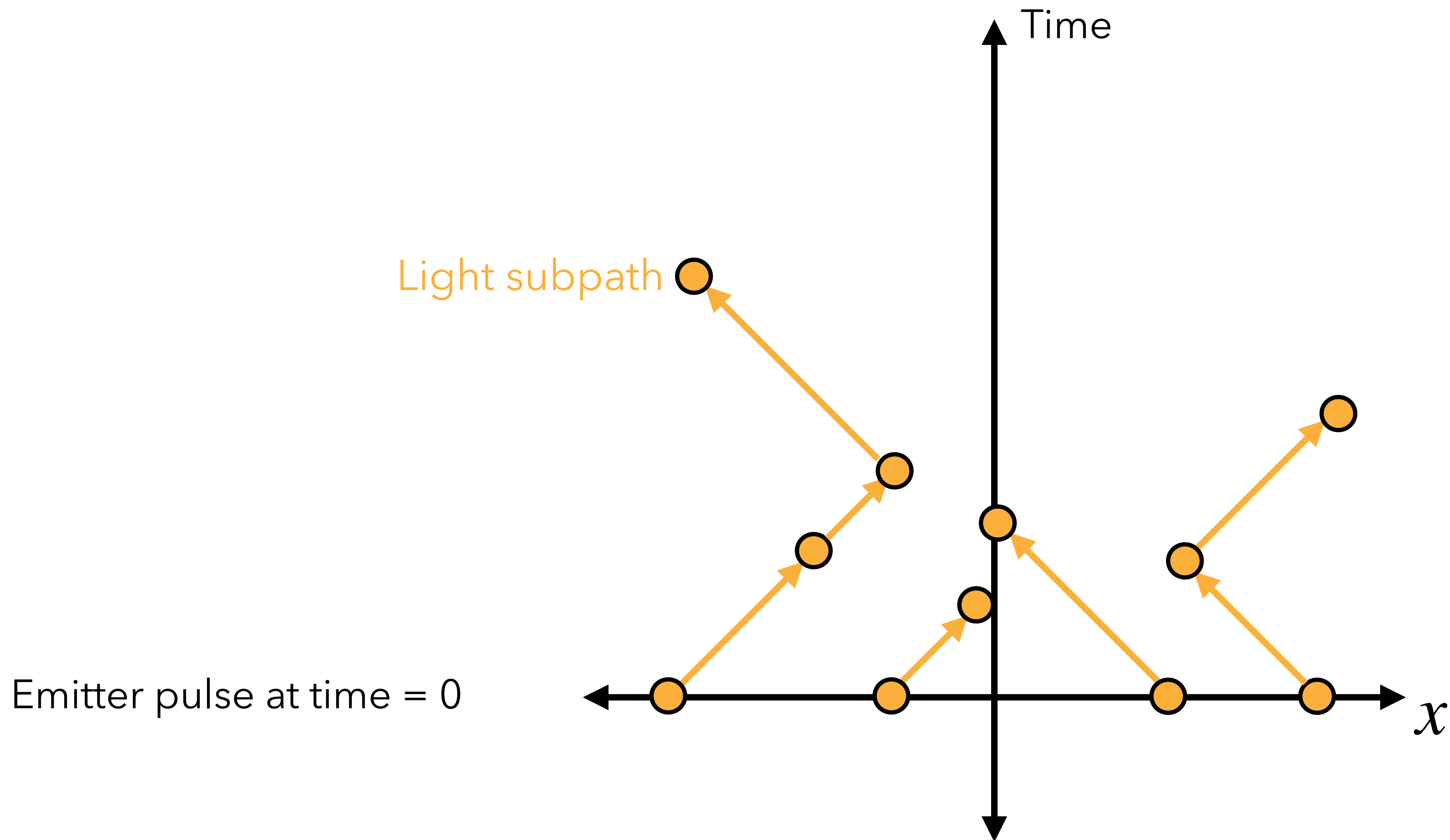
[Minkowski 1908]



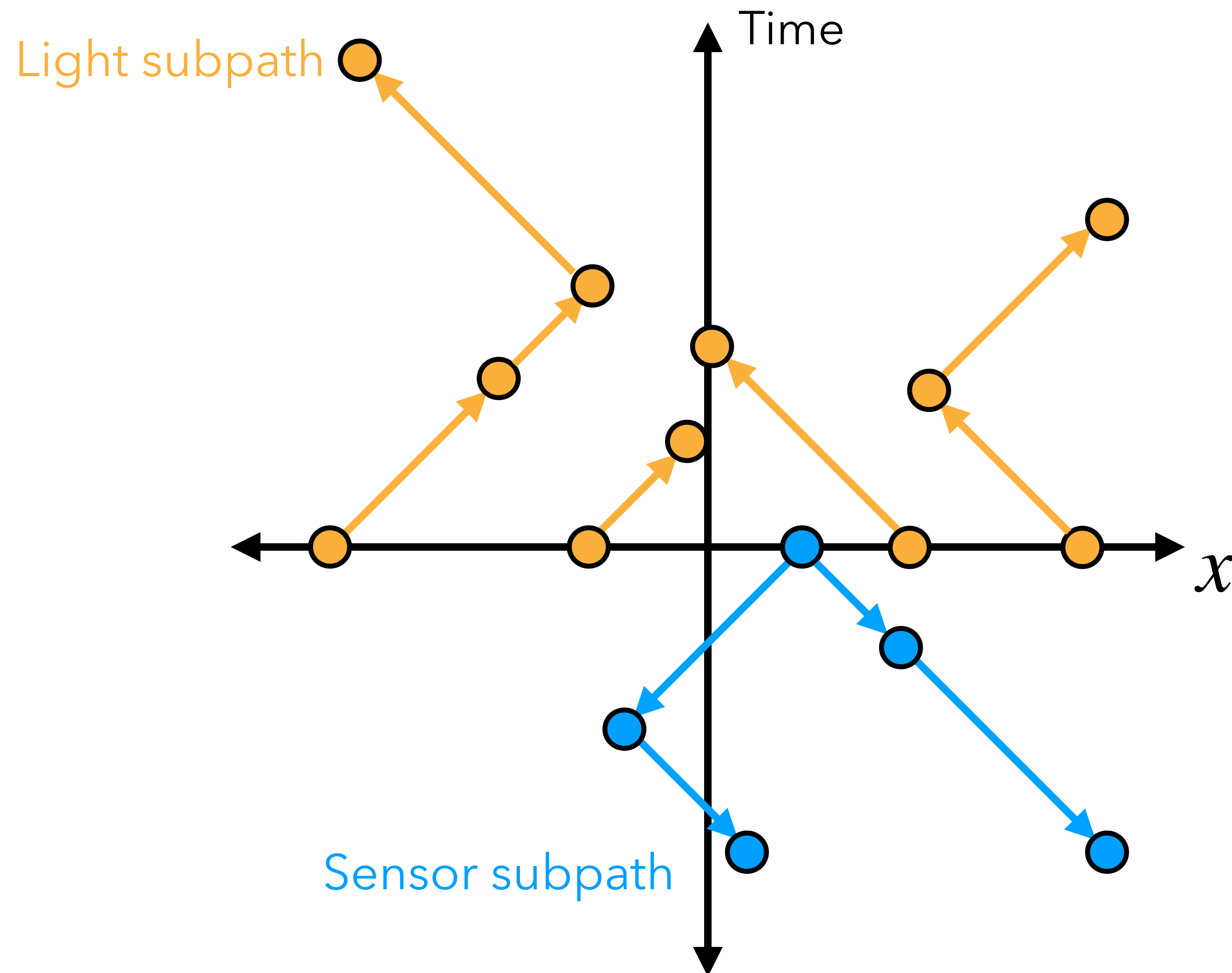
Tracing segments in spacetime



Tracing light subpaths



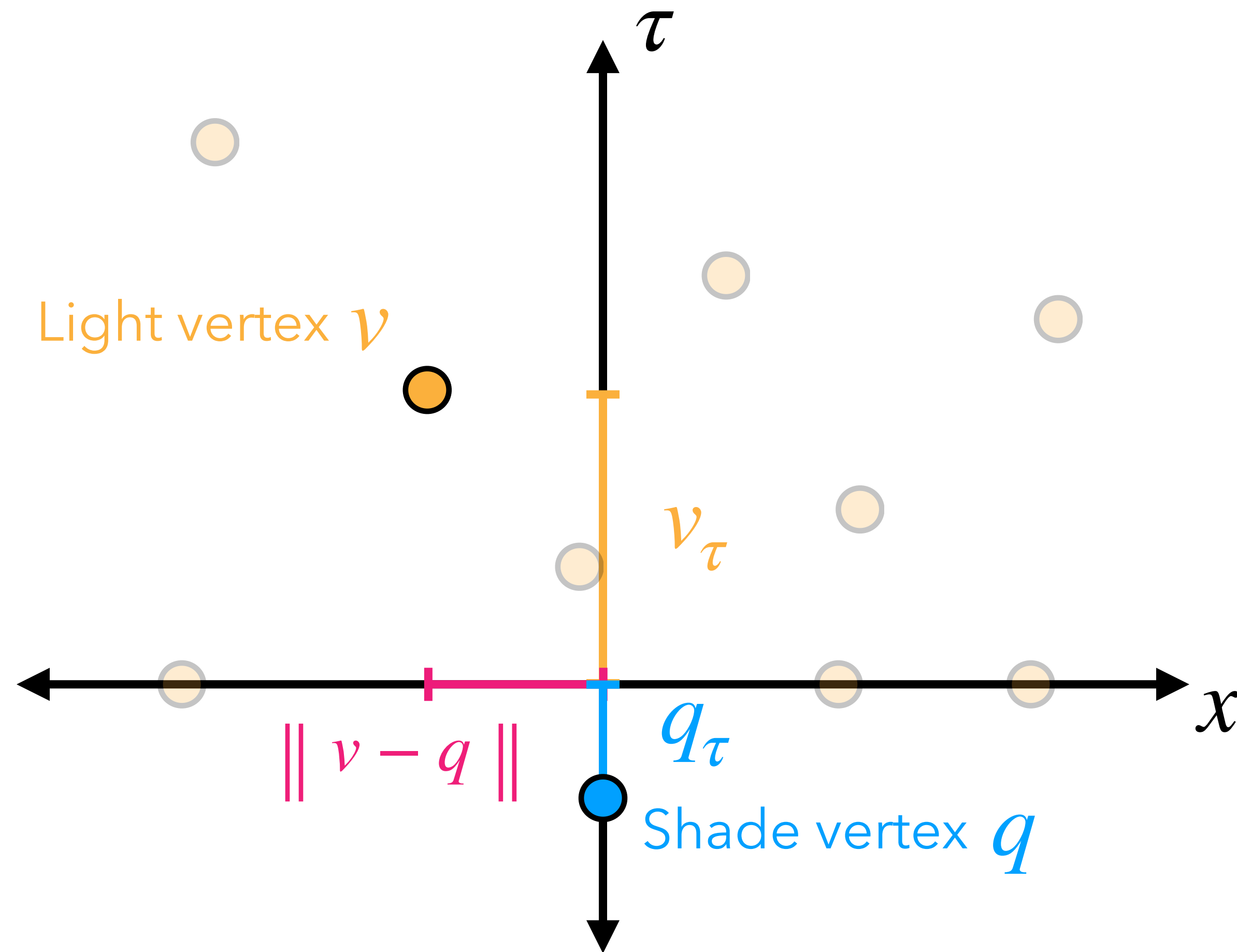
Tracing sensor subpaths



Making a time-valid connection

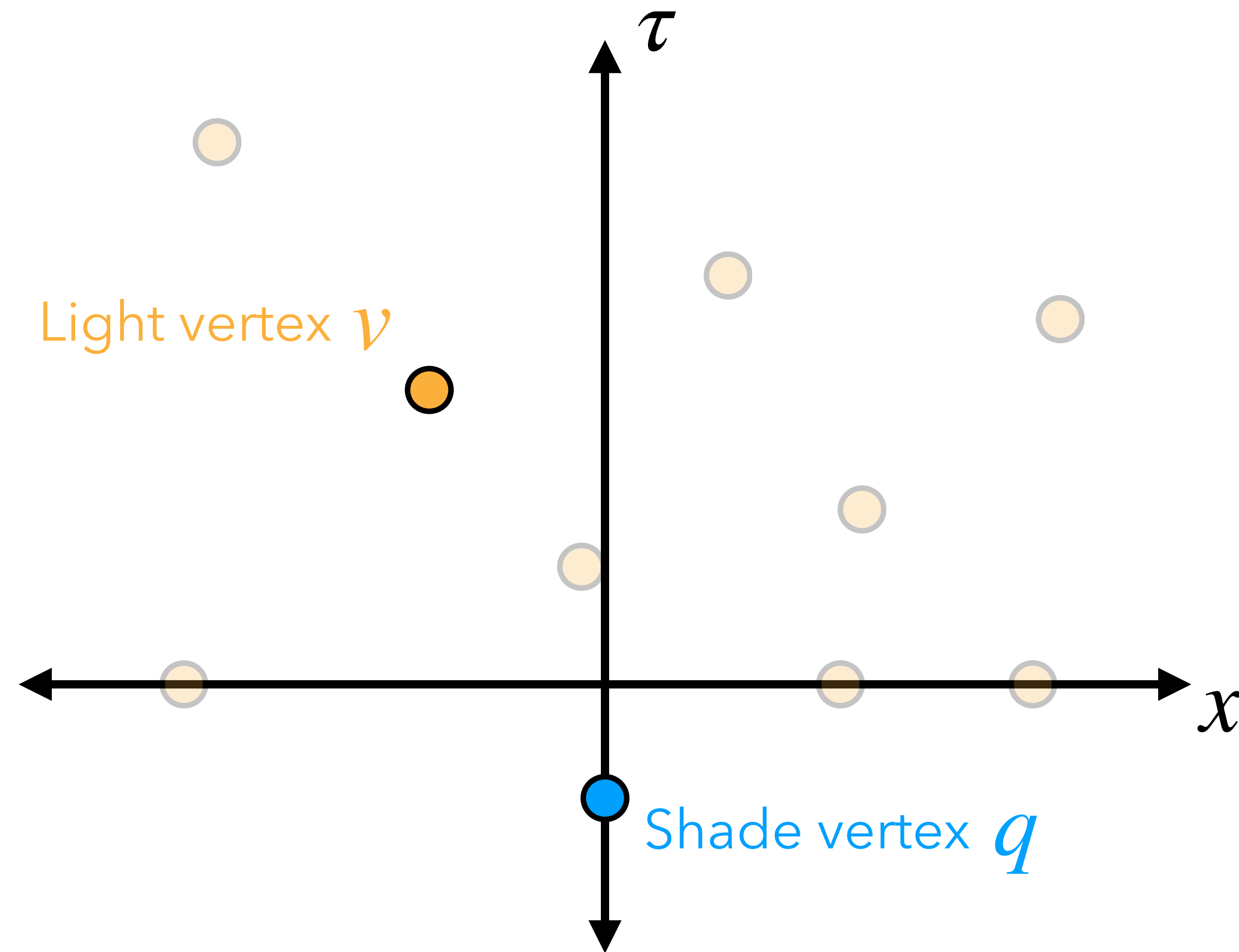
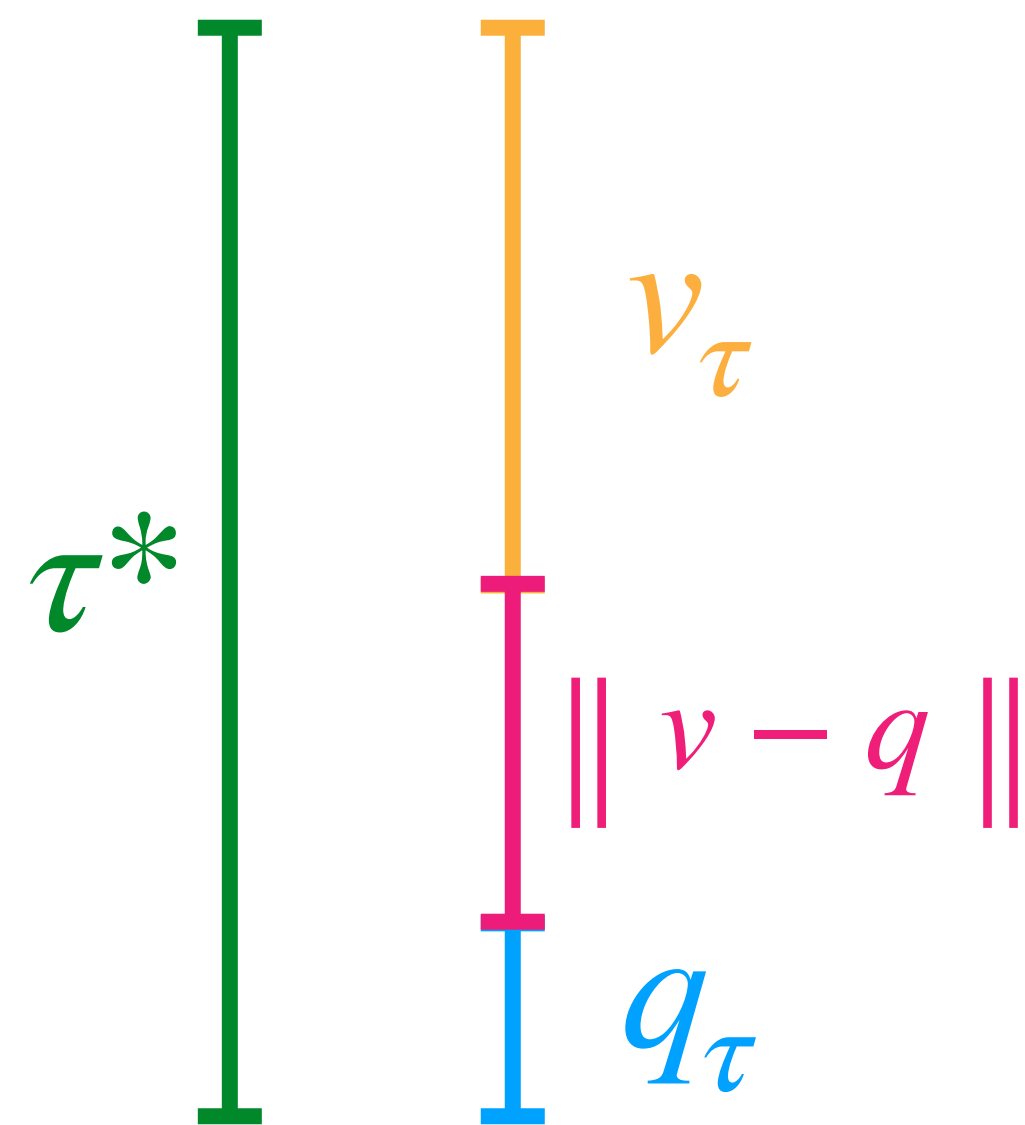
Target

τ^*



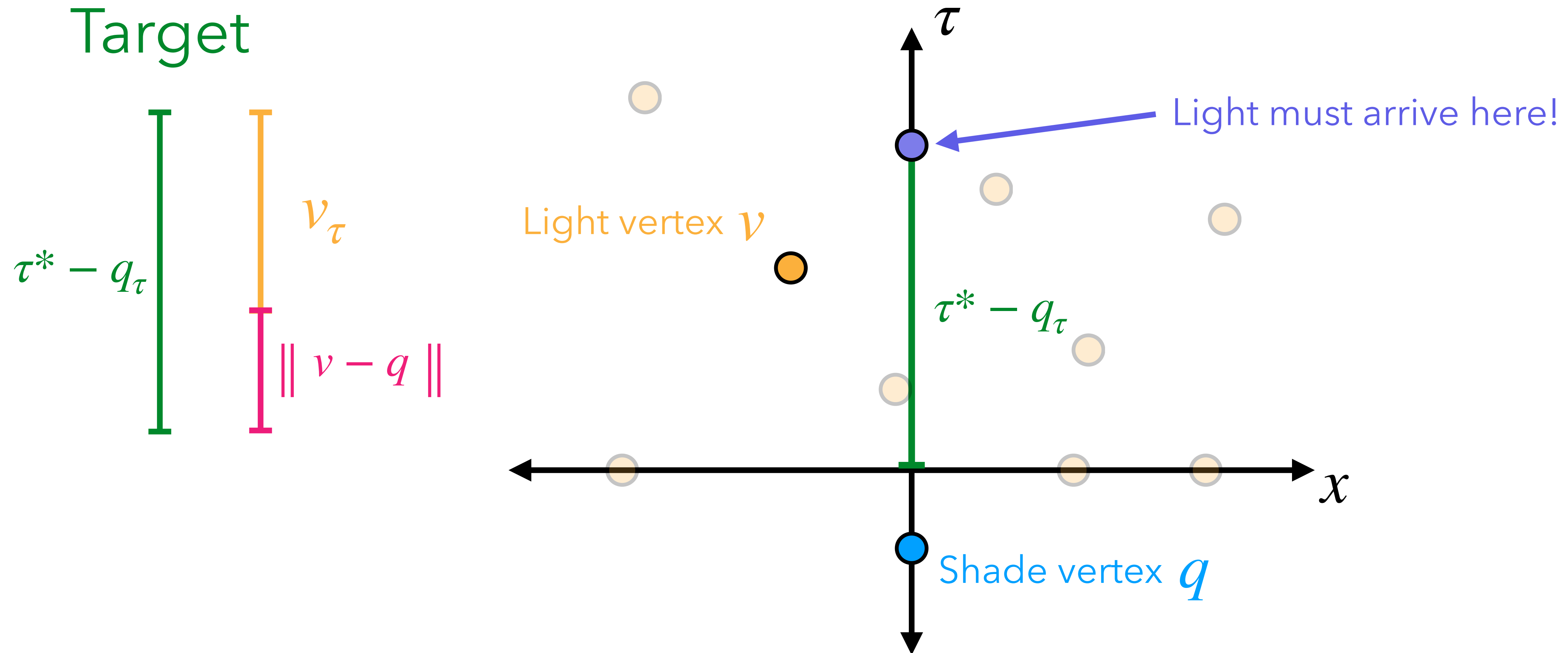
Making a time-valid connection

Target

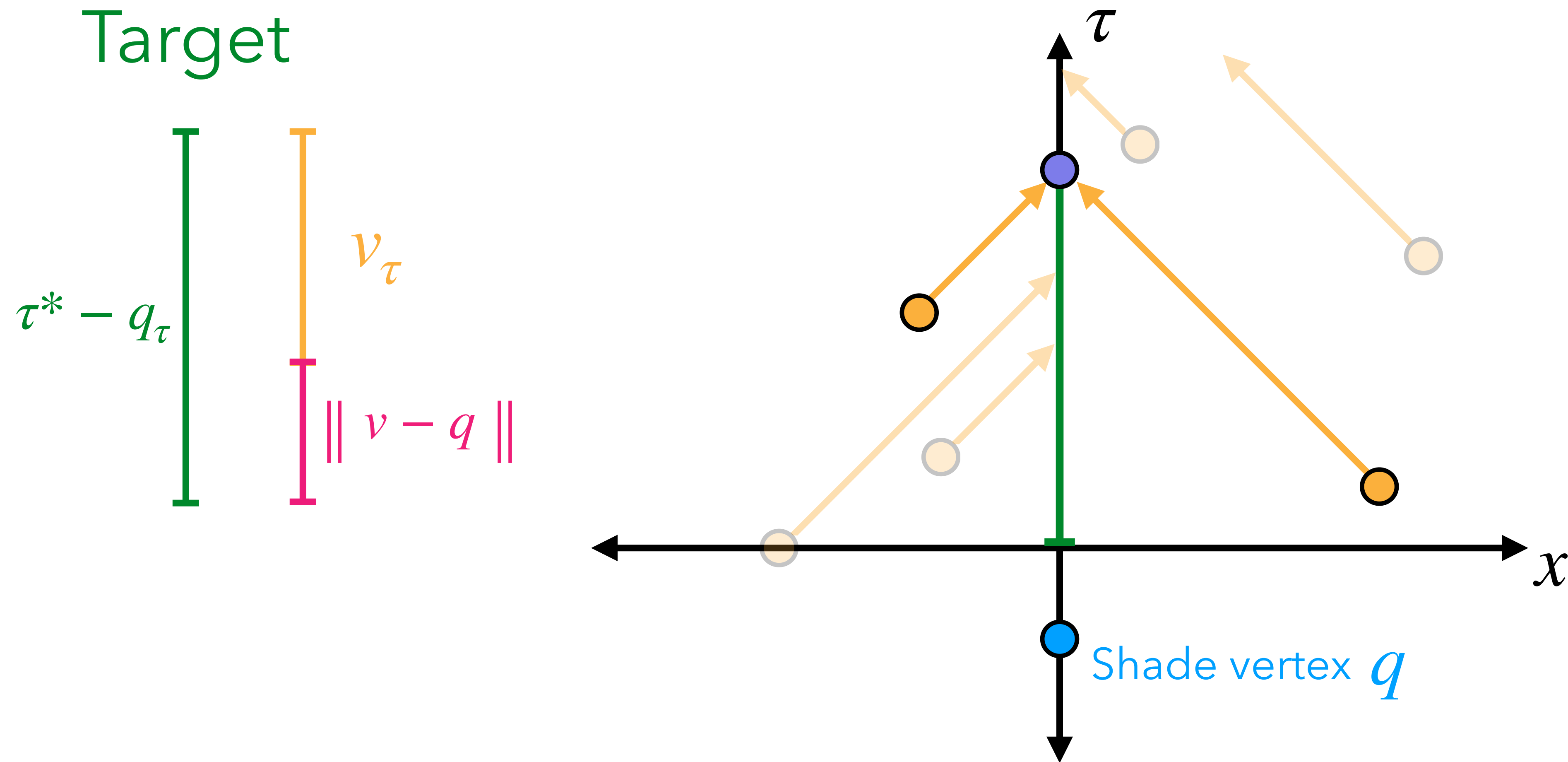


Making a time-valid connection

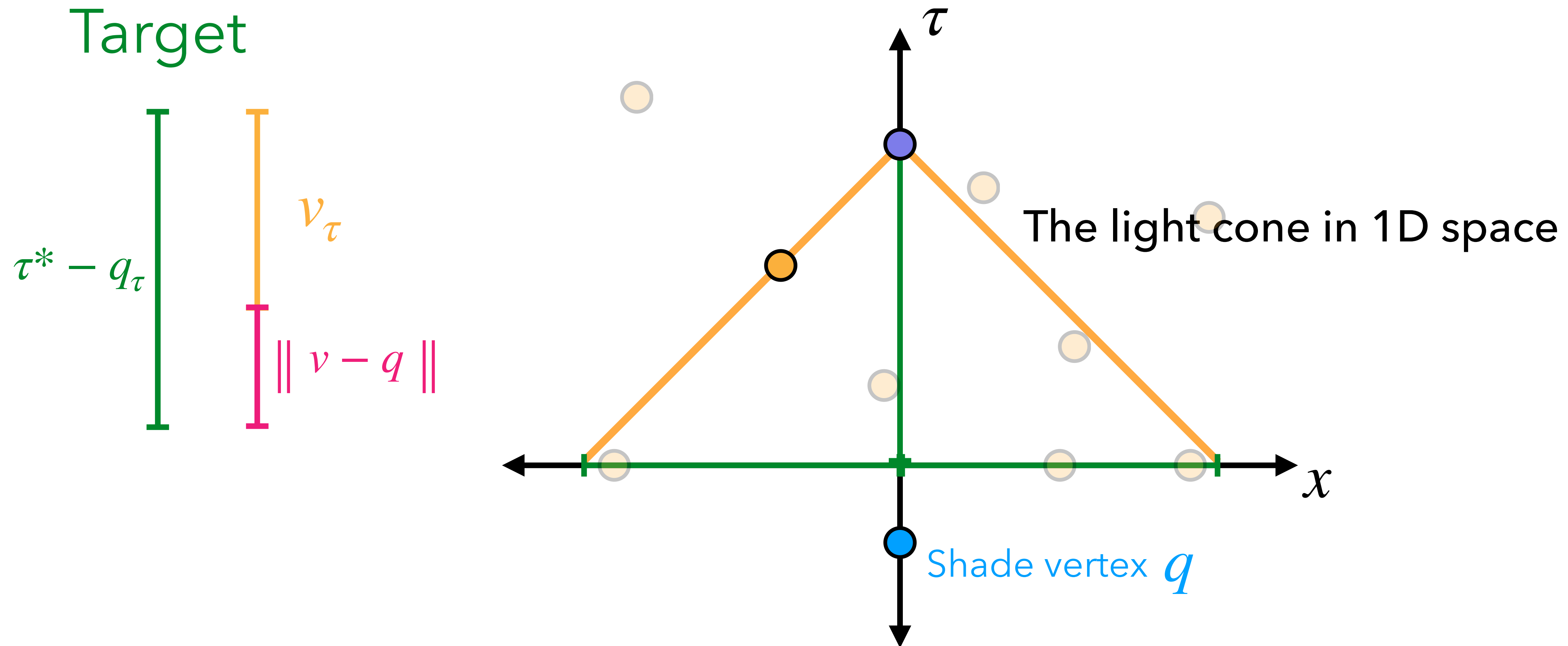
Target



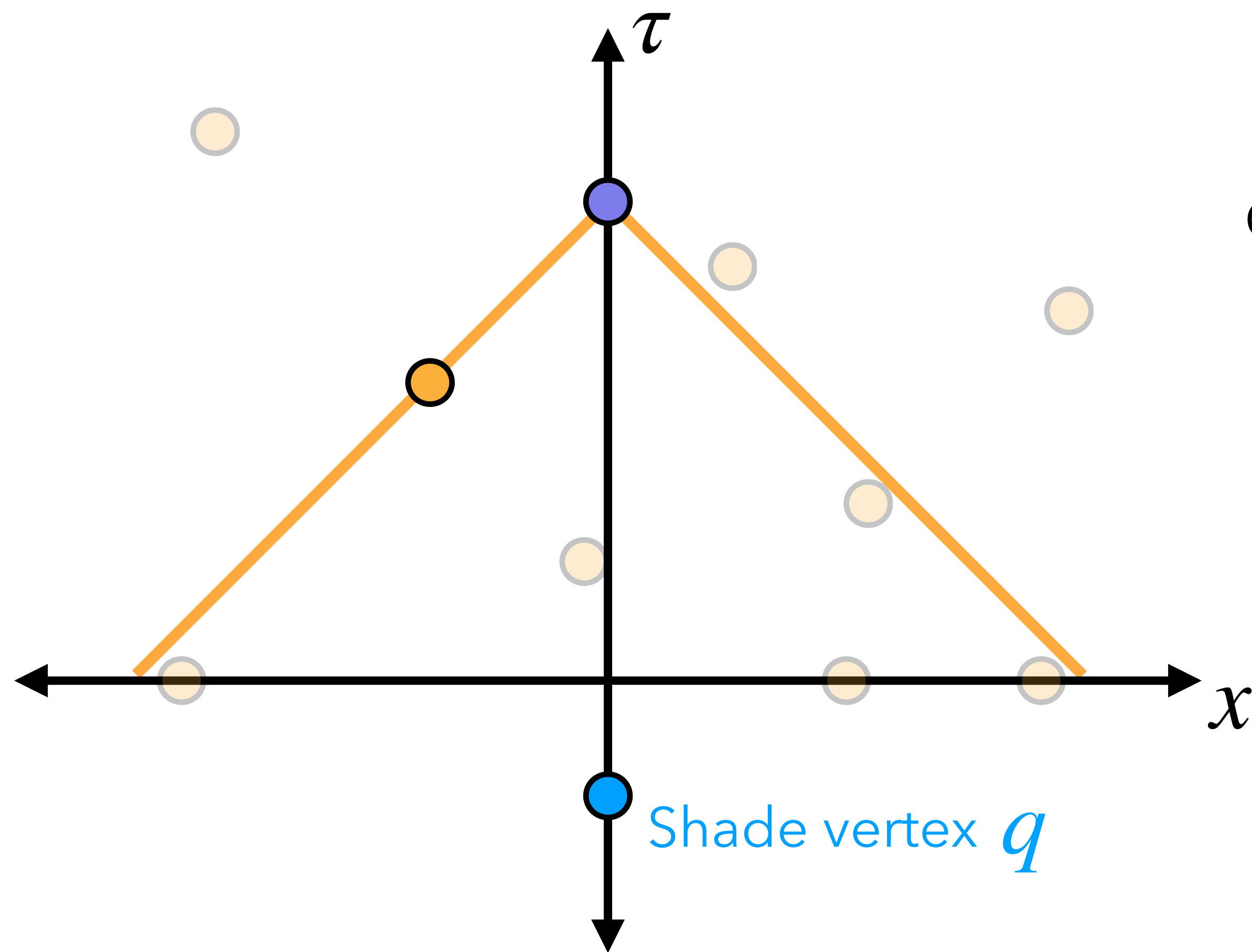
The light cone in our setting



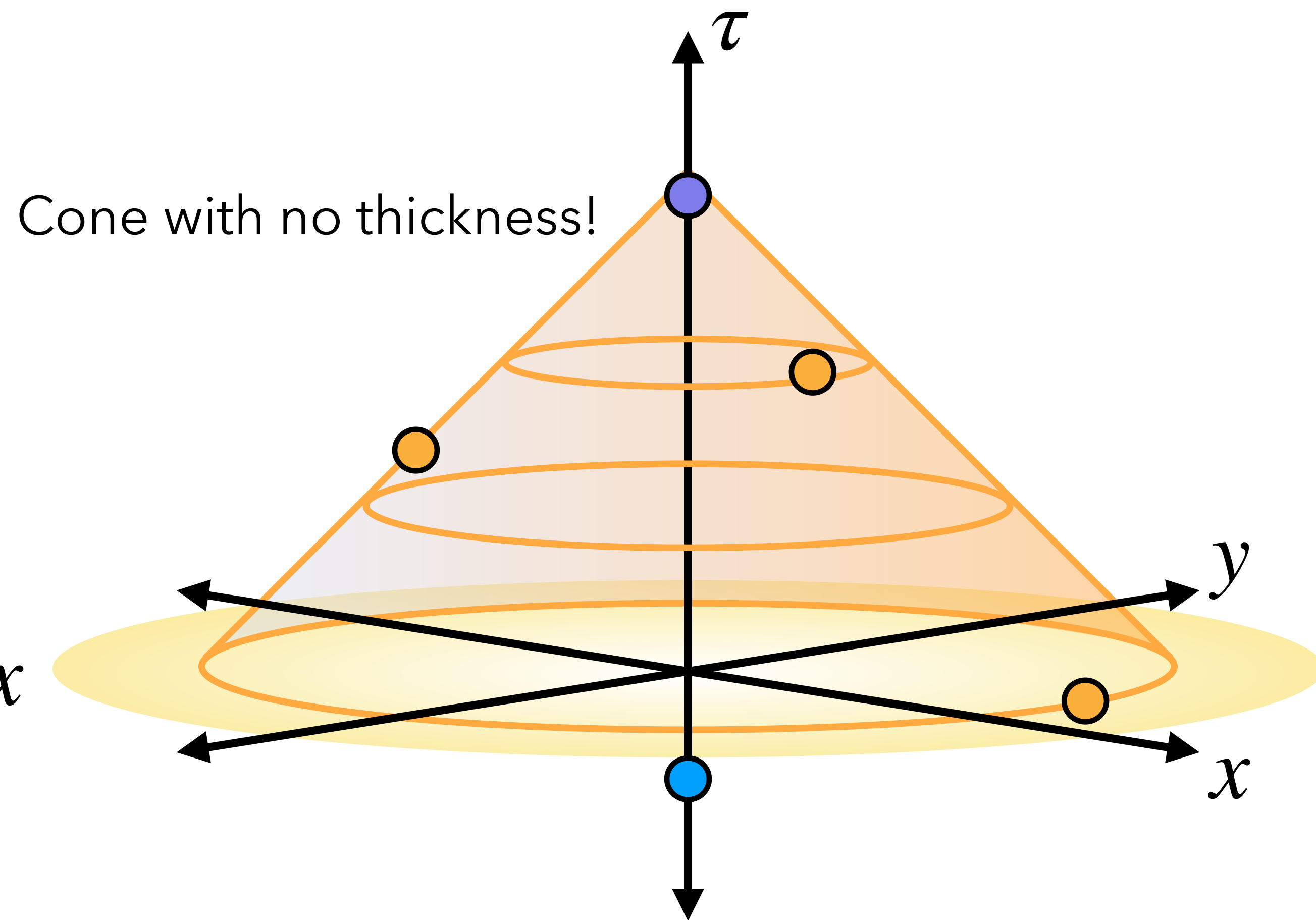
The light cone in our setting



The light cone in our setting

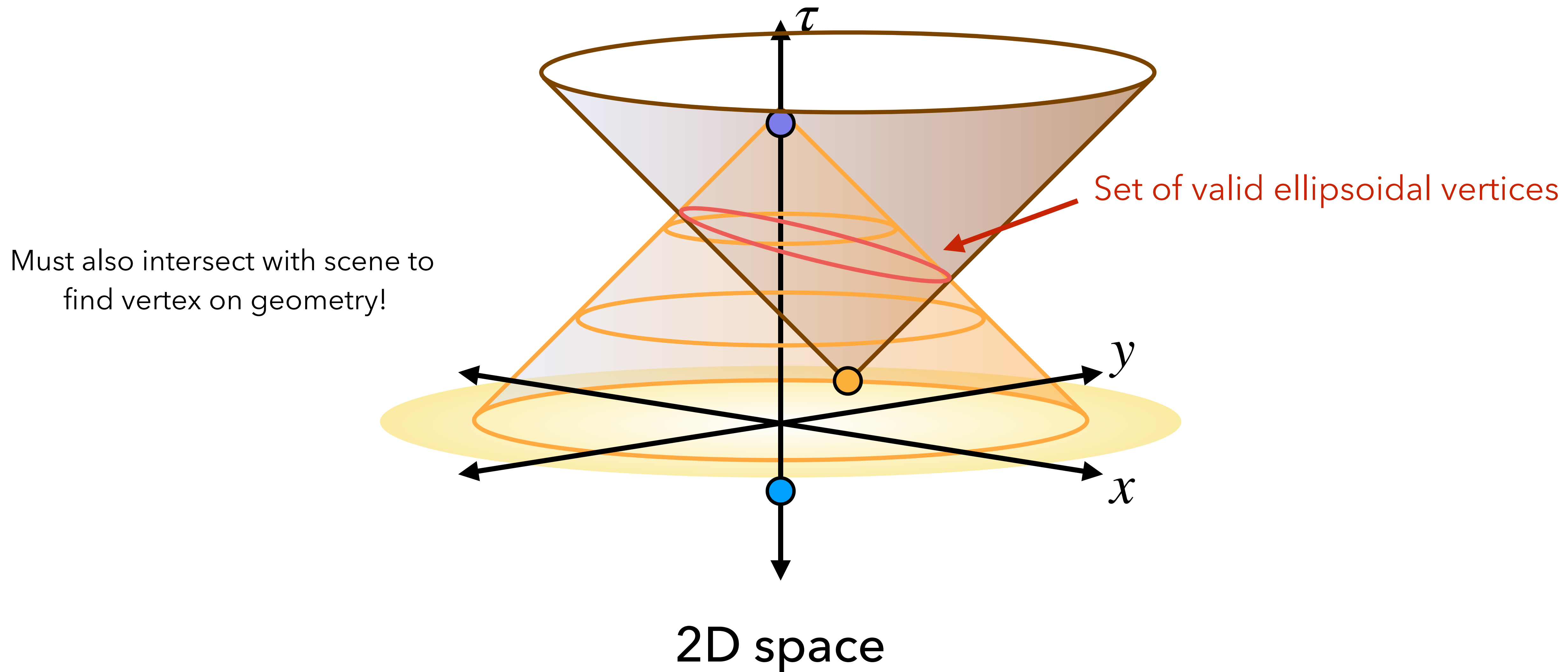


1D space

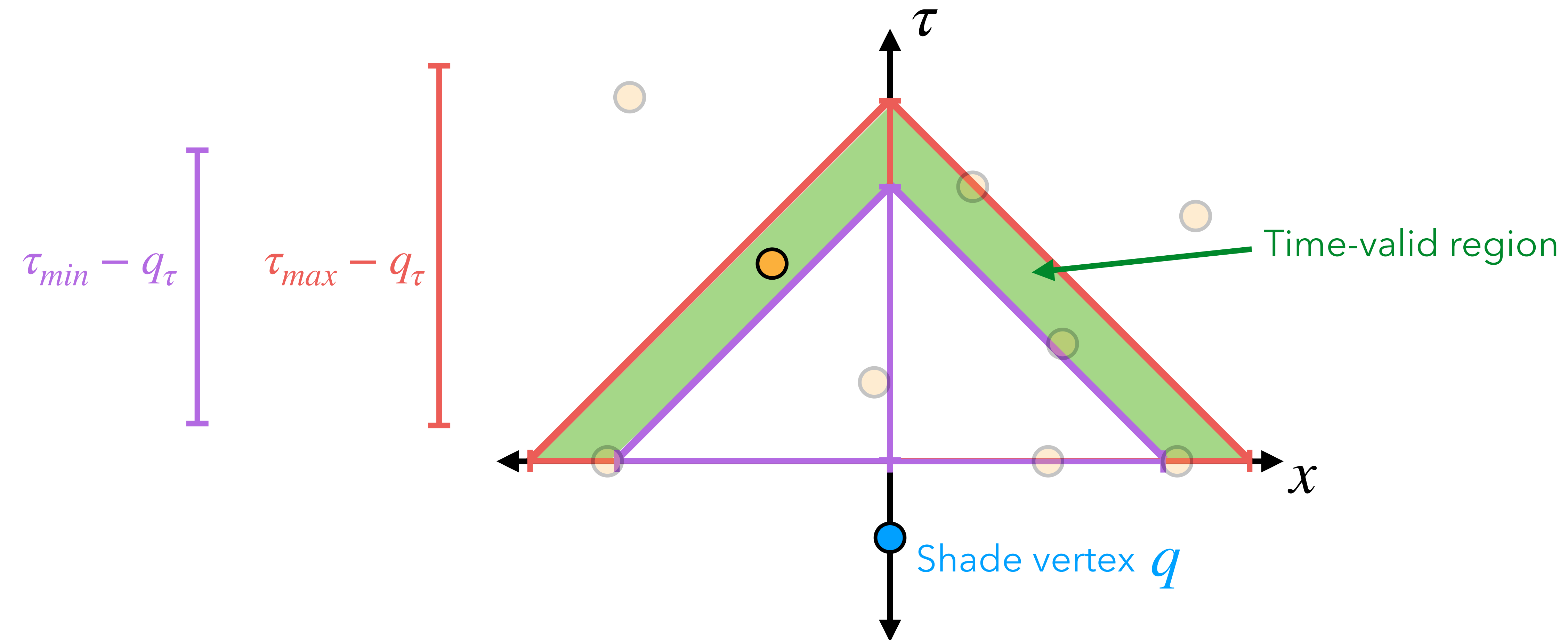


2D space

EPC as the intersection of light cones



The light-cone shell

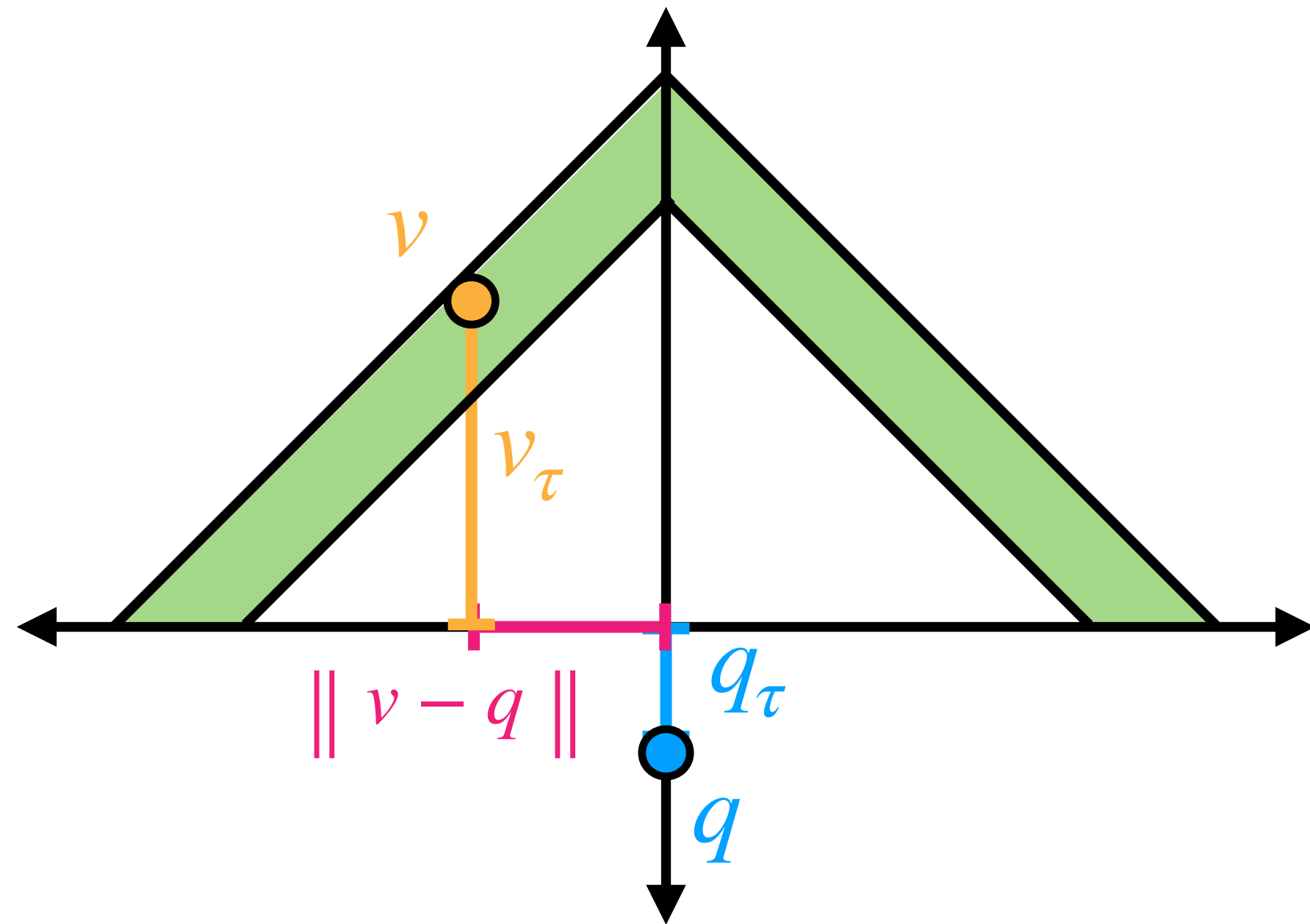


Light-cone shell queries

Observation: validity test is cheap!

$$\tau_{min} \leq v_{\tau} + \|v - q\| + q_{\tau} \leq \tau_{max}$$

Light vertex time Connection distance Shade vertex time

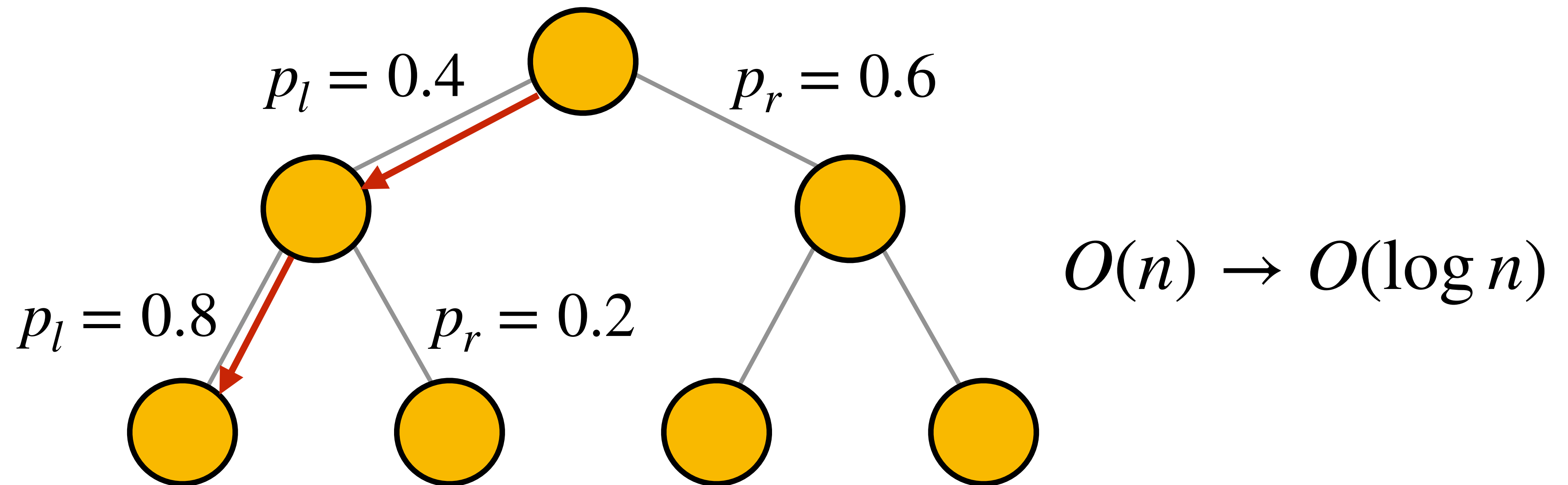


Goal: given stored light vertices,
find those in the shell

Accelerating the query

"Stochastic Lightcuts" [Yuksel 2019]

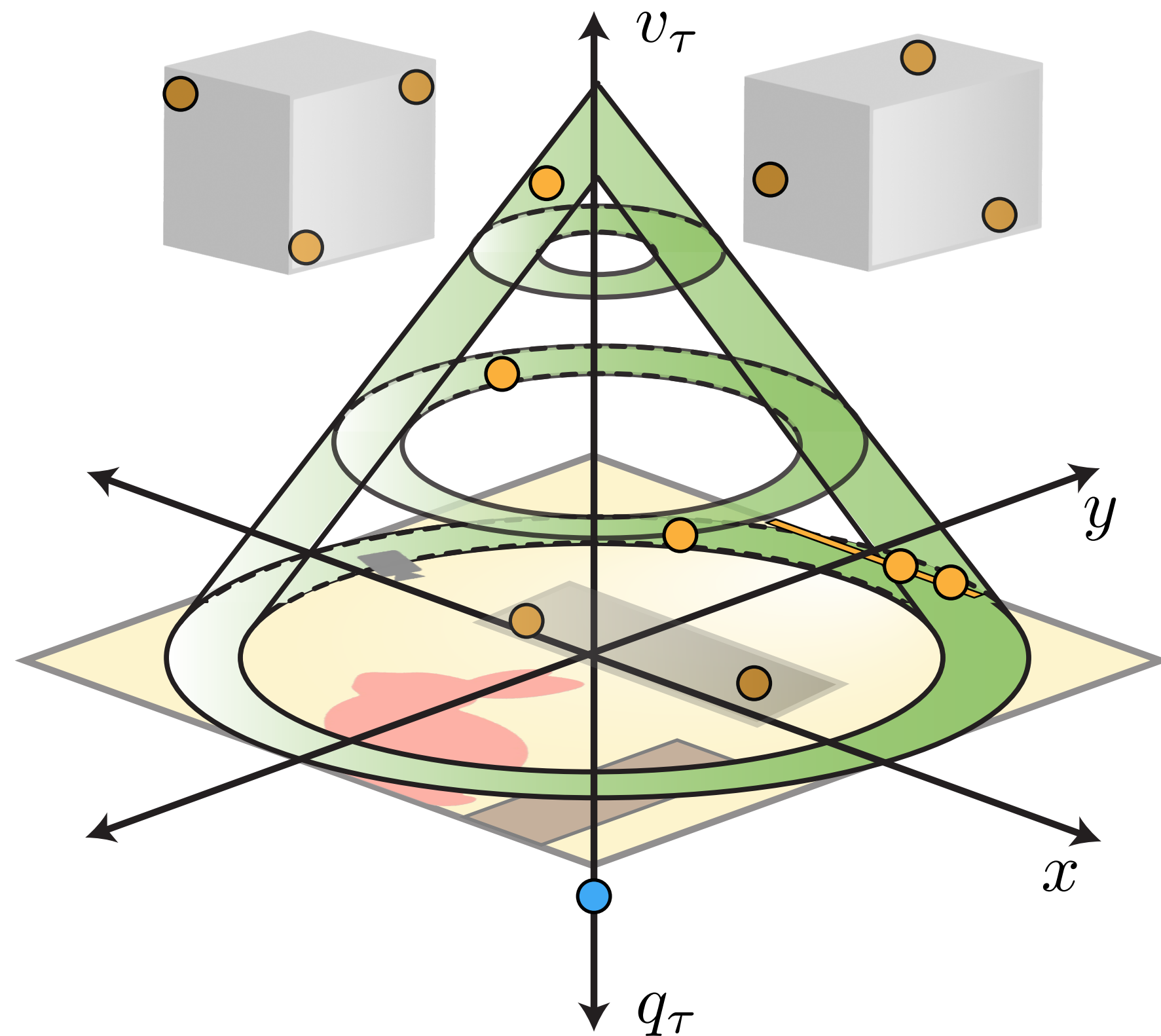
4D balanced binary tree, stochastic lightcuts traversal



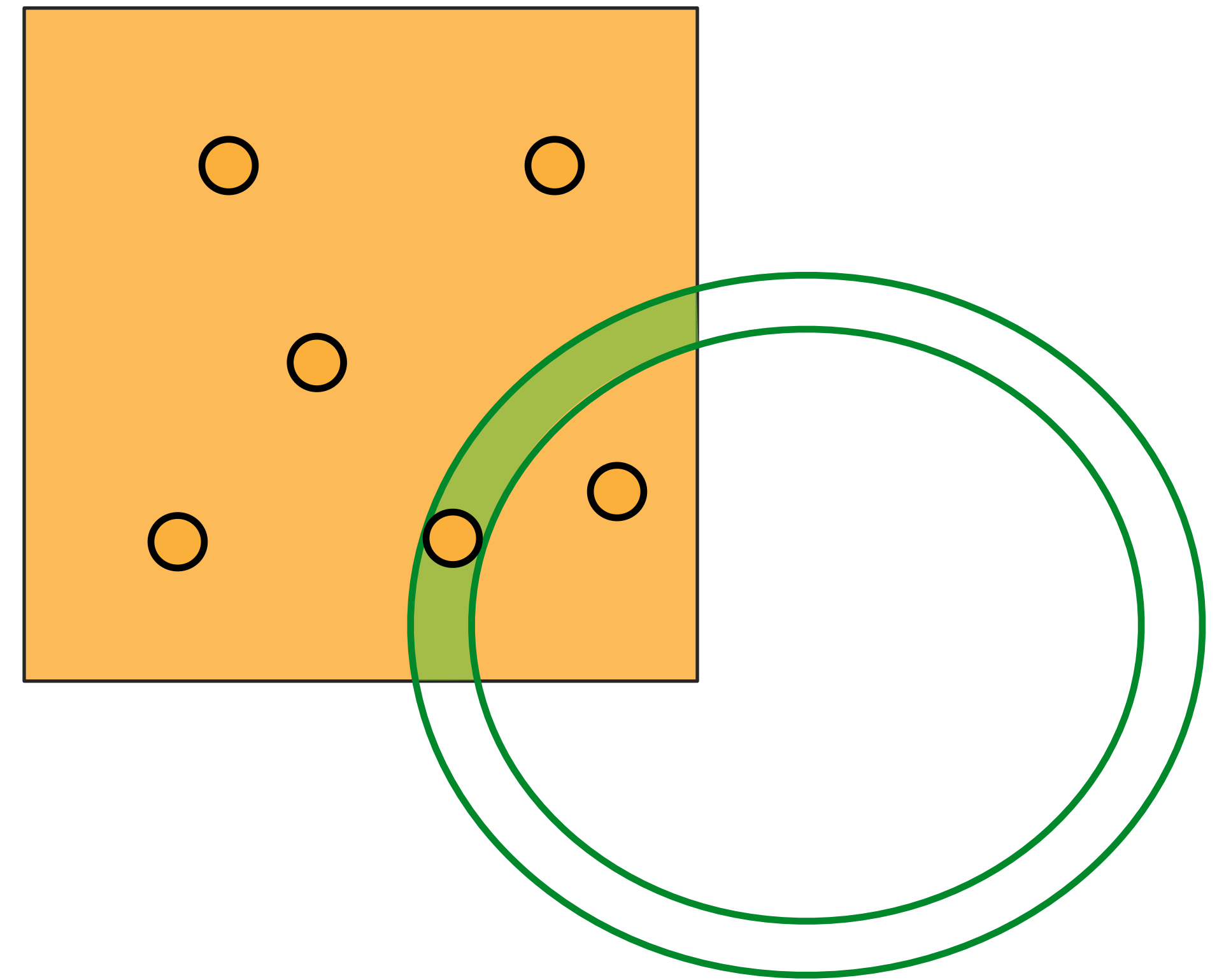
But this doesn't take into account the time gate!

Shell overlap drives traversal

Pruned



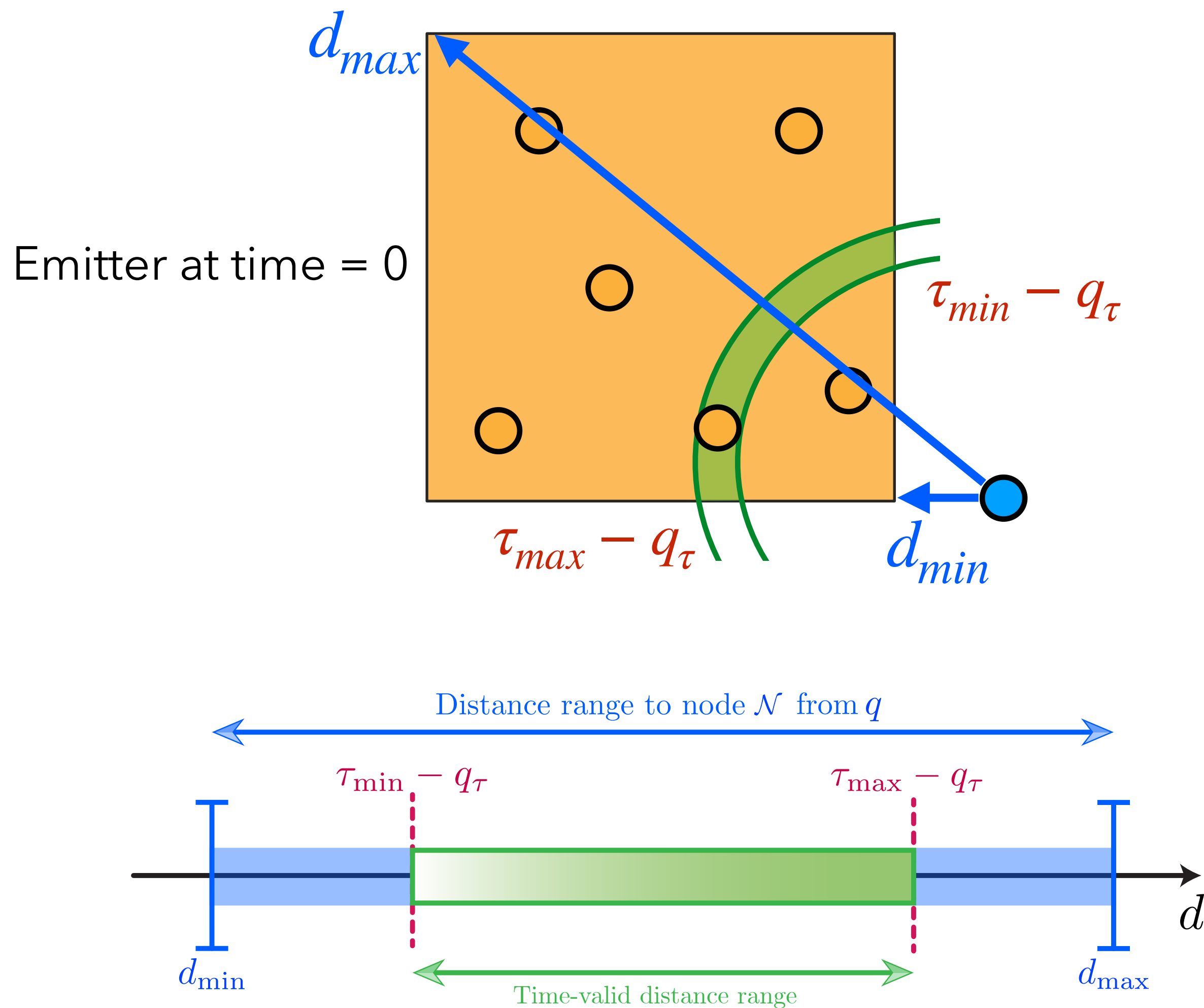
Prune if completely outside of shell



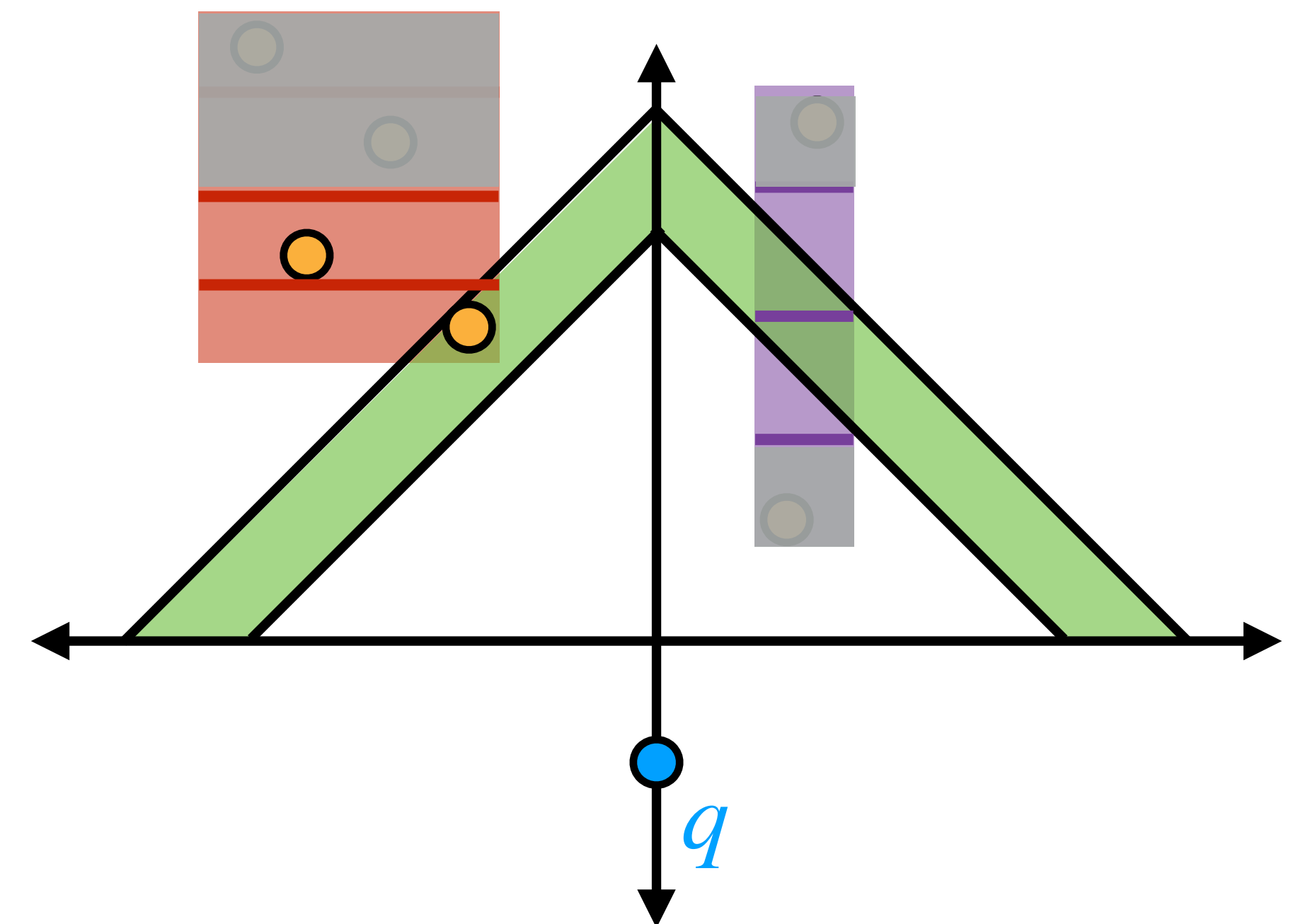
Estimate overlap

Estimating the overlap

Spatial overlap



Temporal overlap



Only consider slabs touching the shell

A time-valid connection accelerator

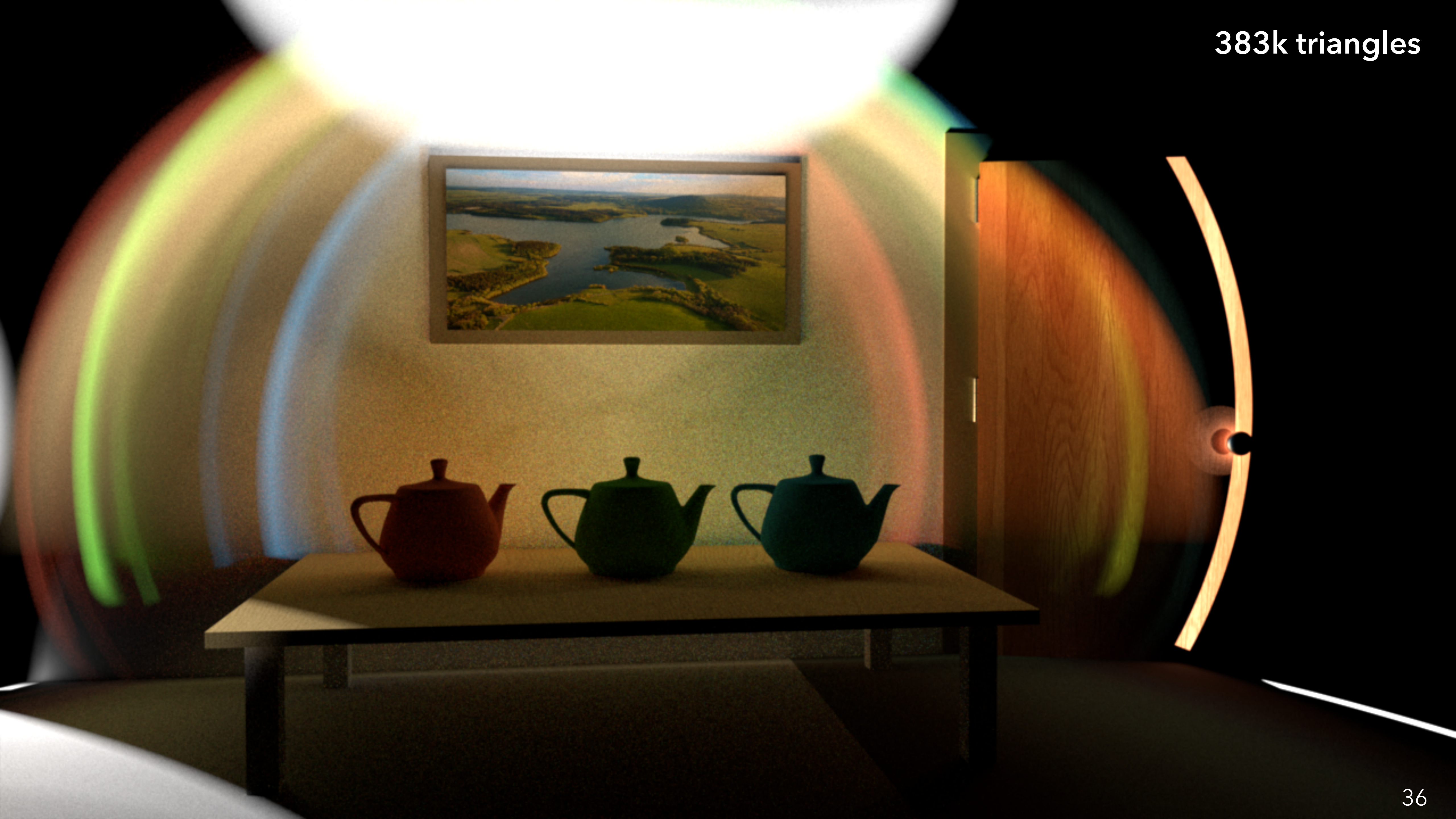
Drops into any algorithm that connects subpaths, we instantiate it in a BDPT framework (Mitsuba 0.6)

1. **Trace** light subpaths, store vertices
2. **Build** acceleration structure
3. **Connect** – every connection is a shell query
 - MIS: standard BDPT weights (singularity suppression)
 $\approx$ probabilistic connections [Popov et al. 2015]

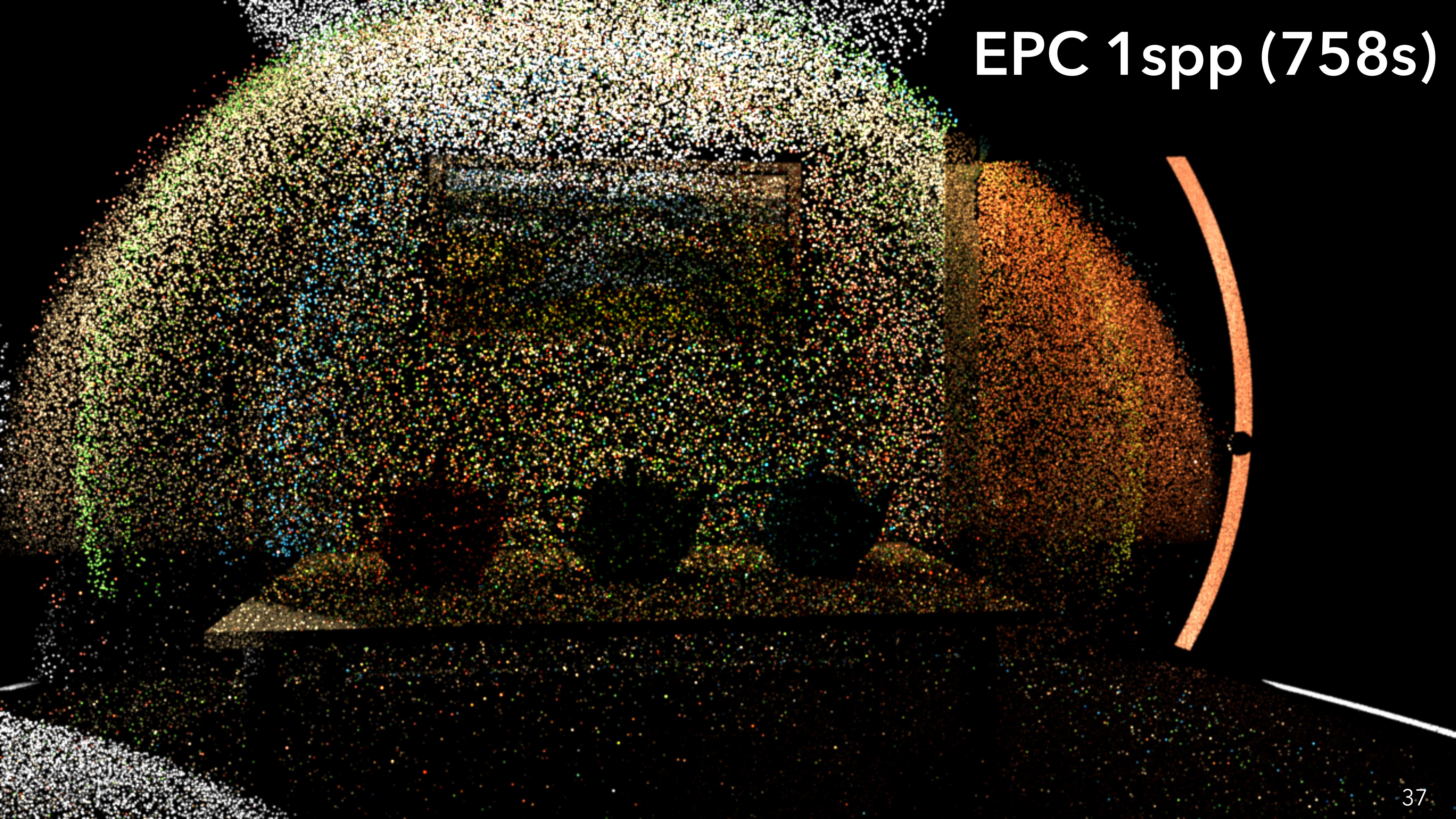


Results

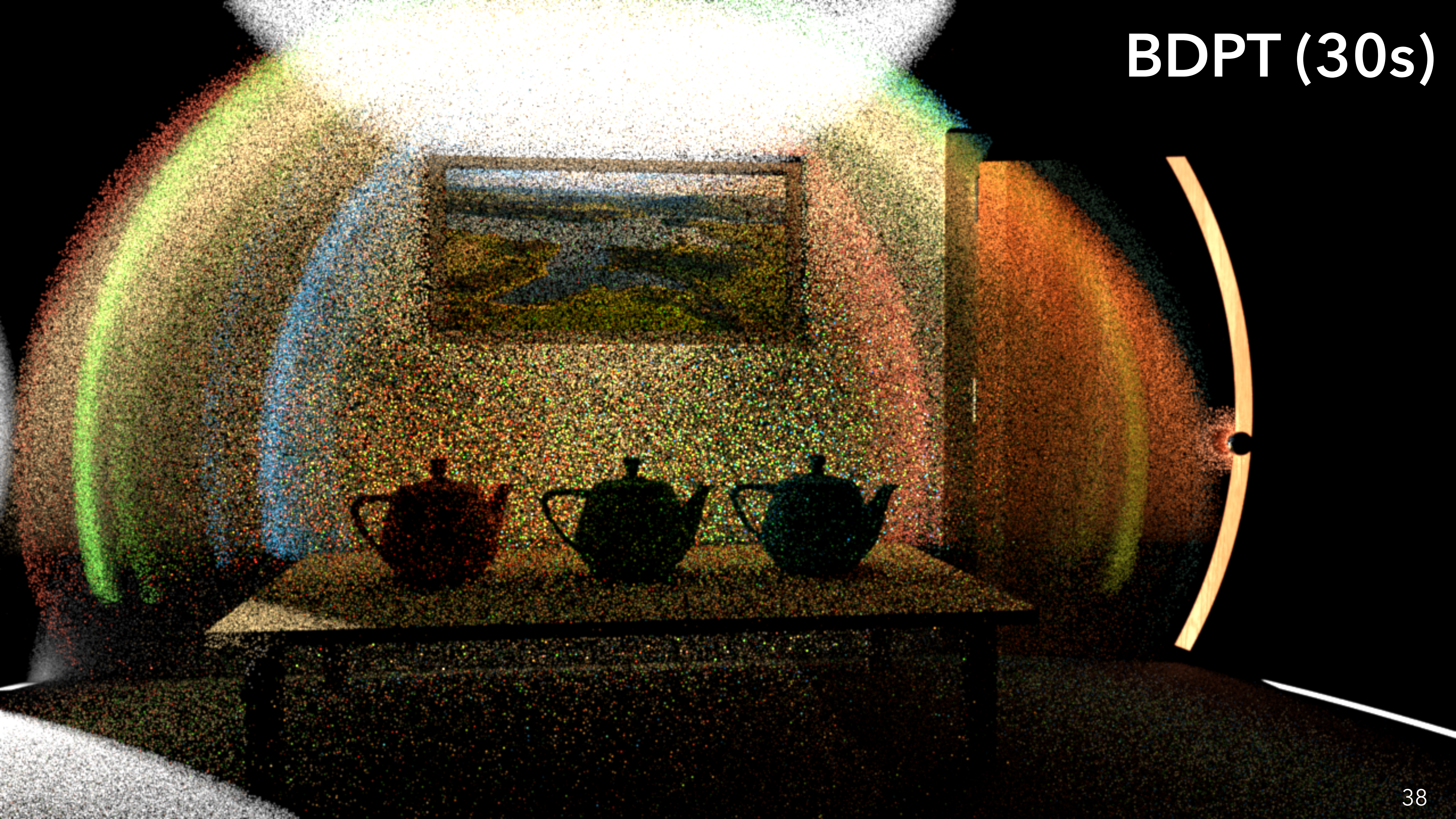
383k triangles



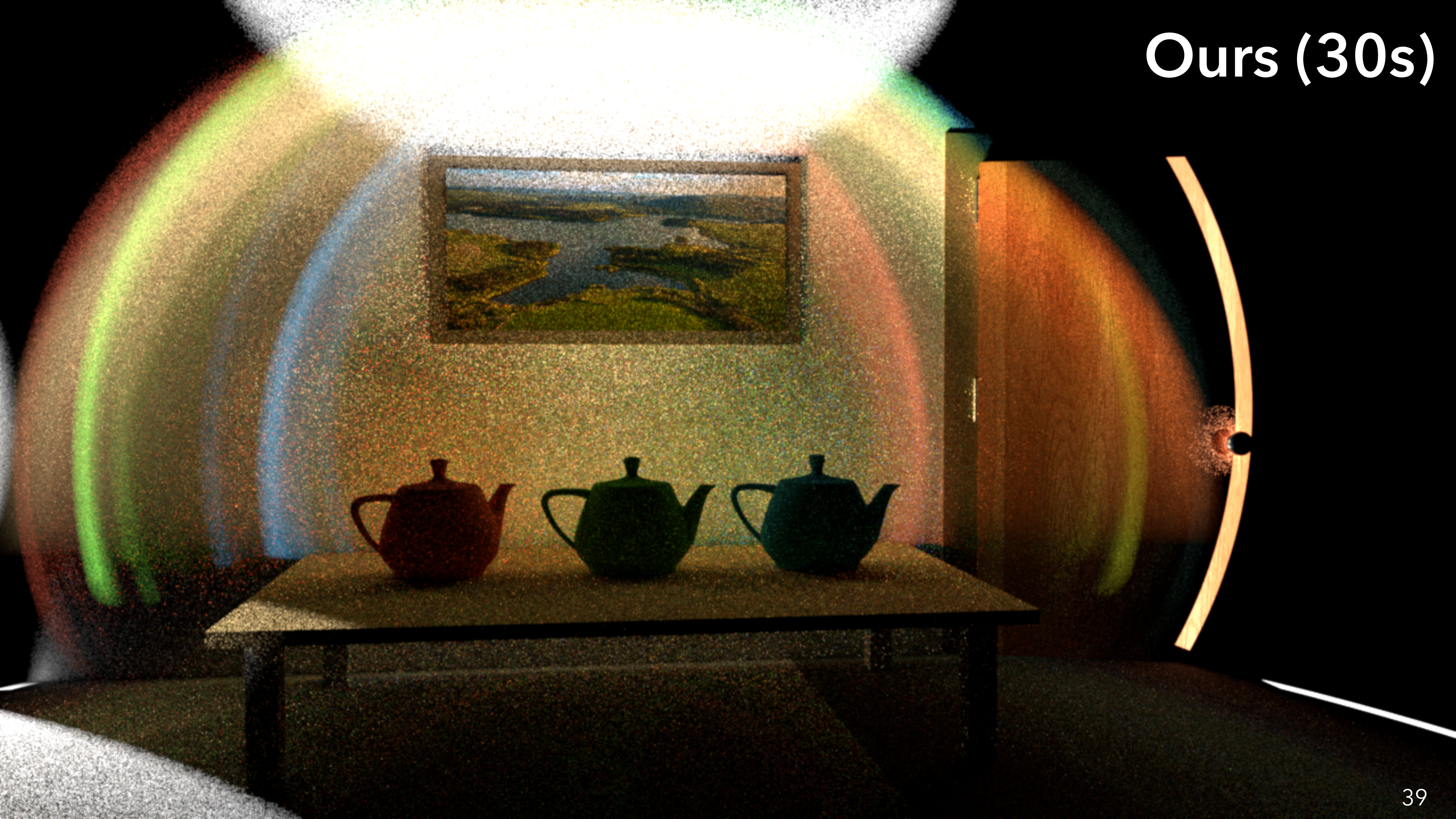
EPC 1spp (758s)



BDPT (30s)



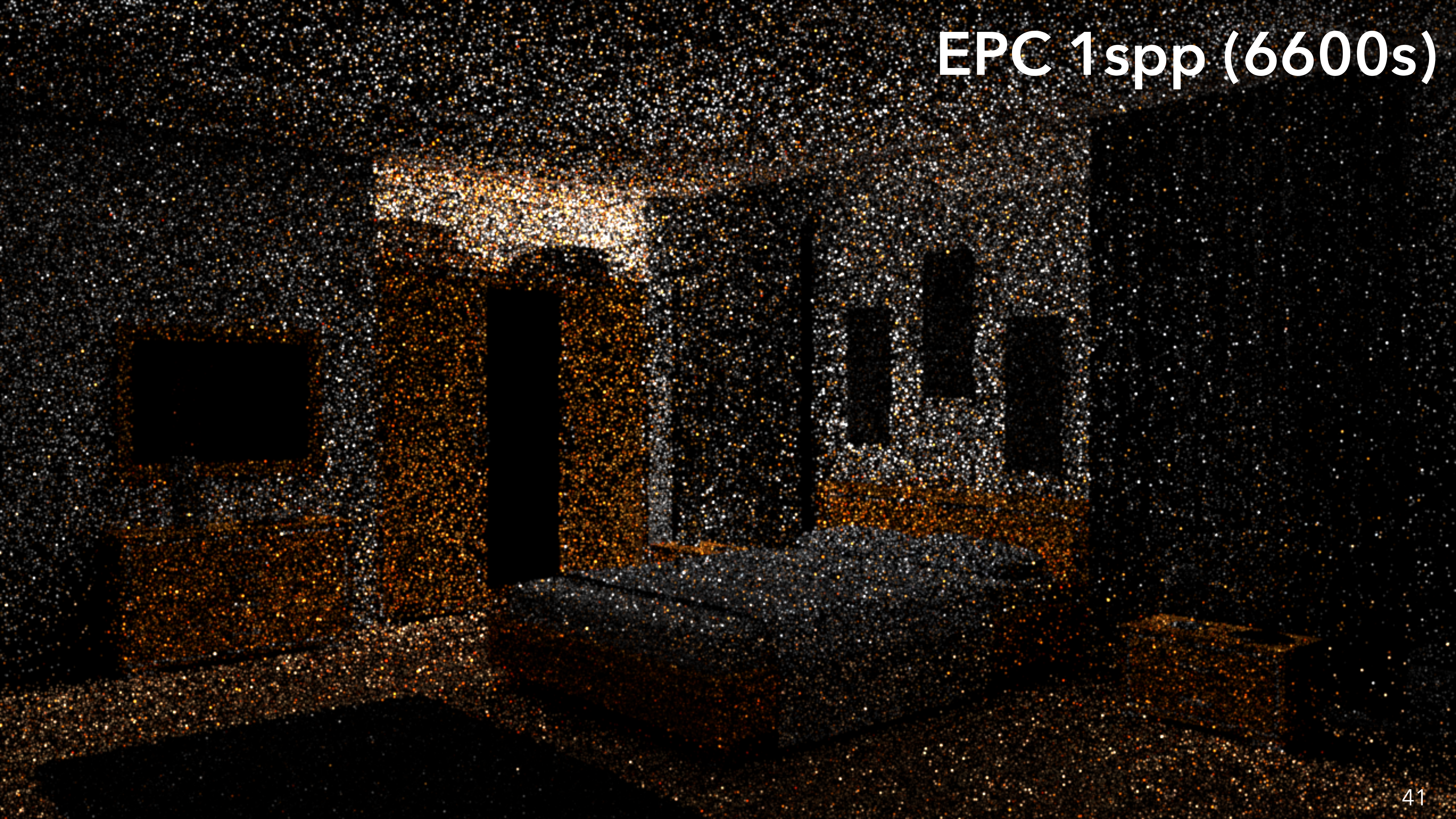
Ours (30s)



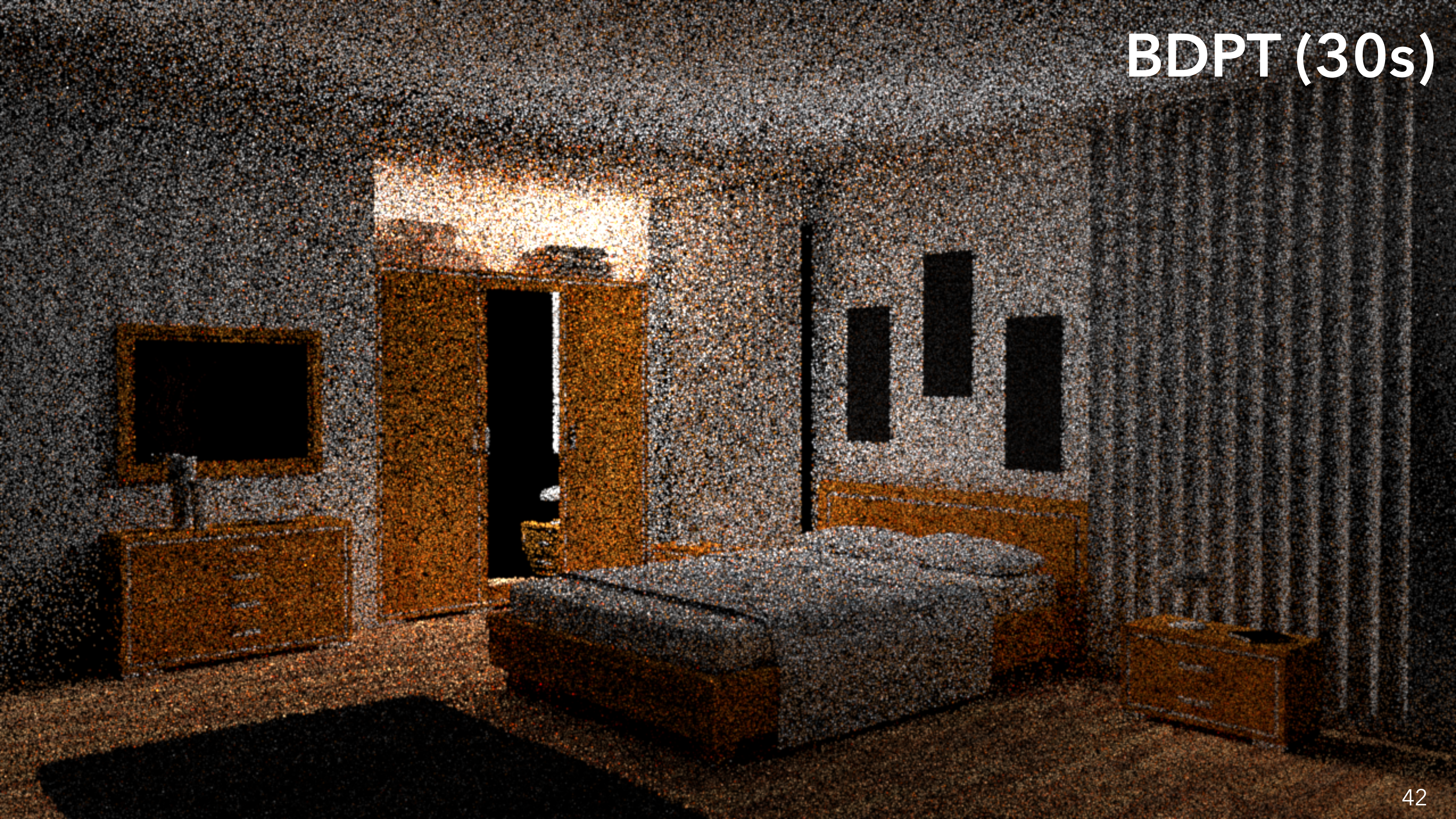
1.5m triangles



EPC 1spp (6600s)



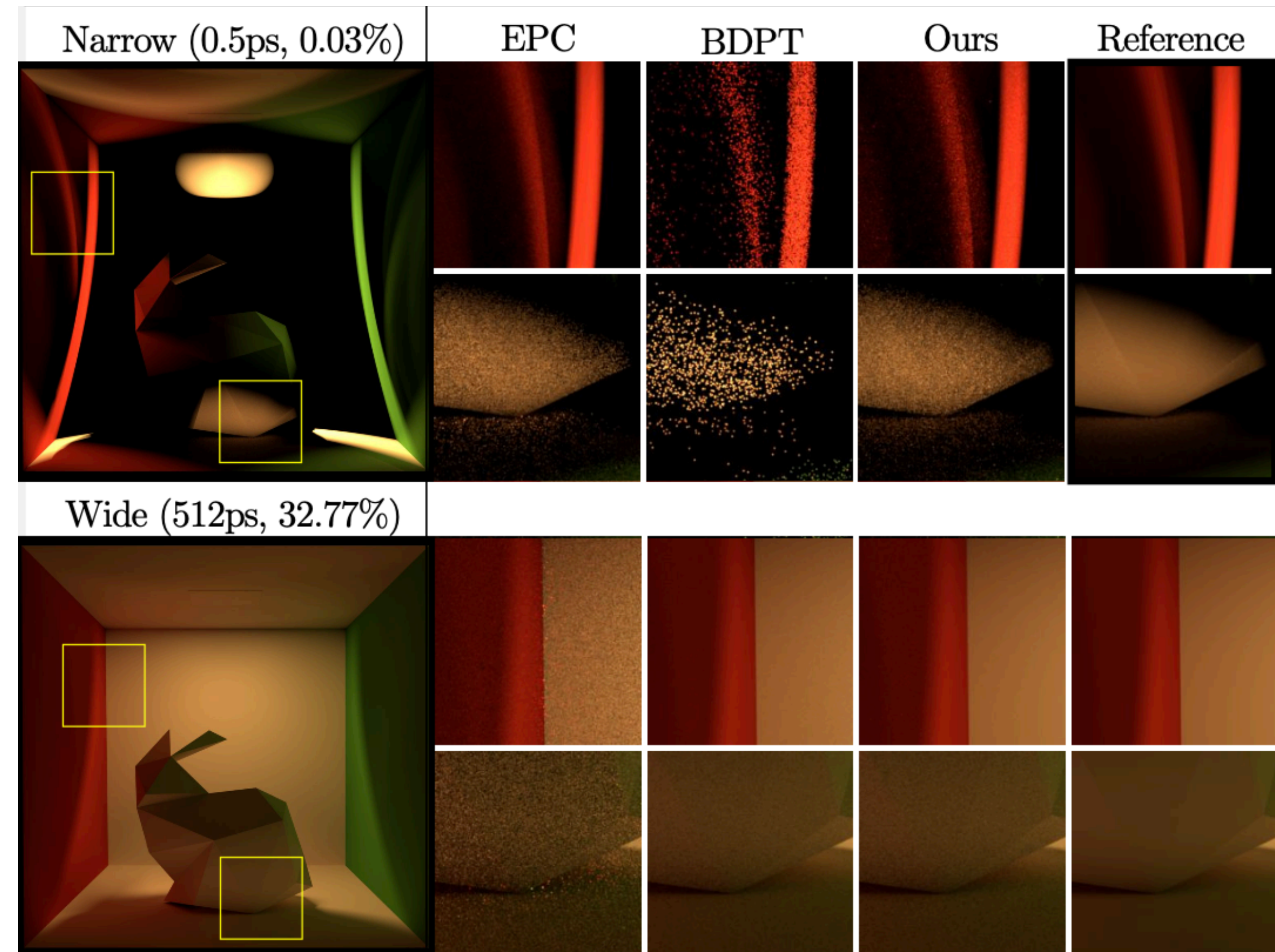
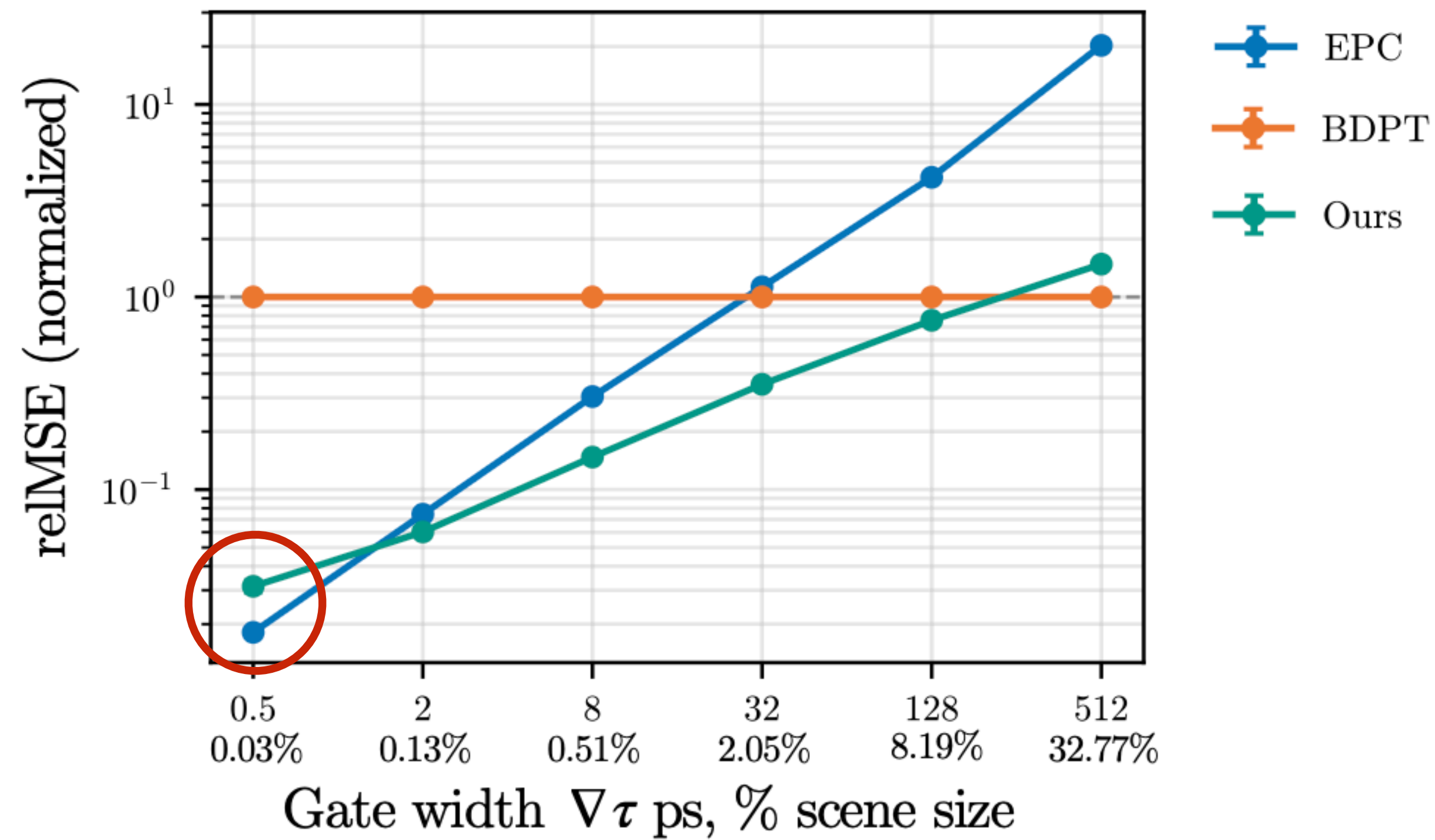
BDPT (30s)



Ours (30s)



Gate width sweep



66 triangles



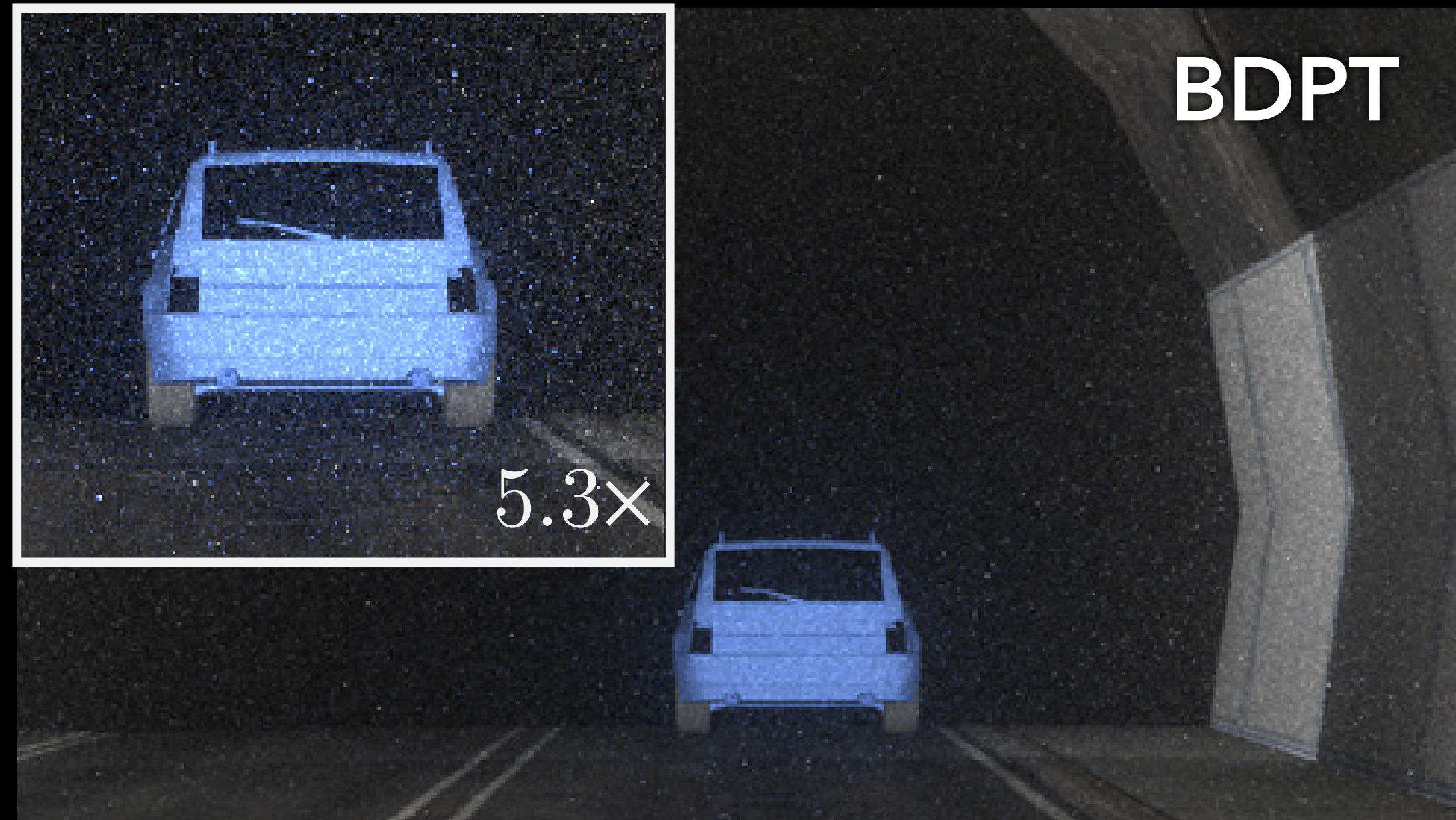
Applications

Steady-state

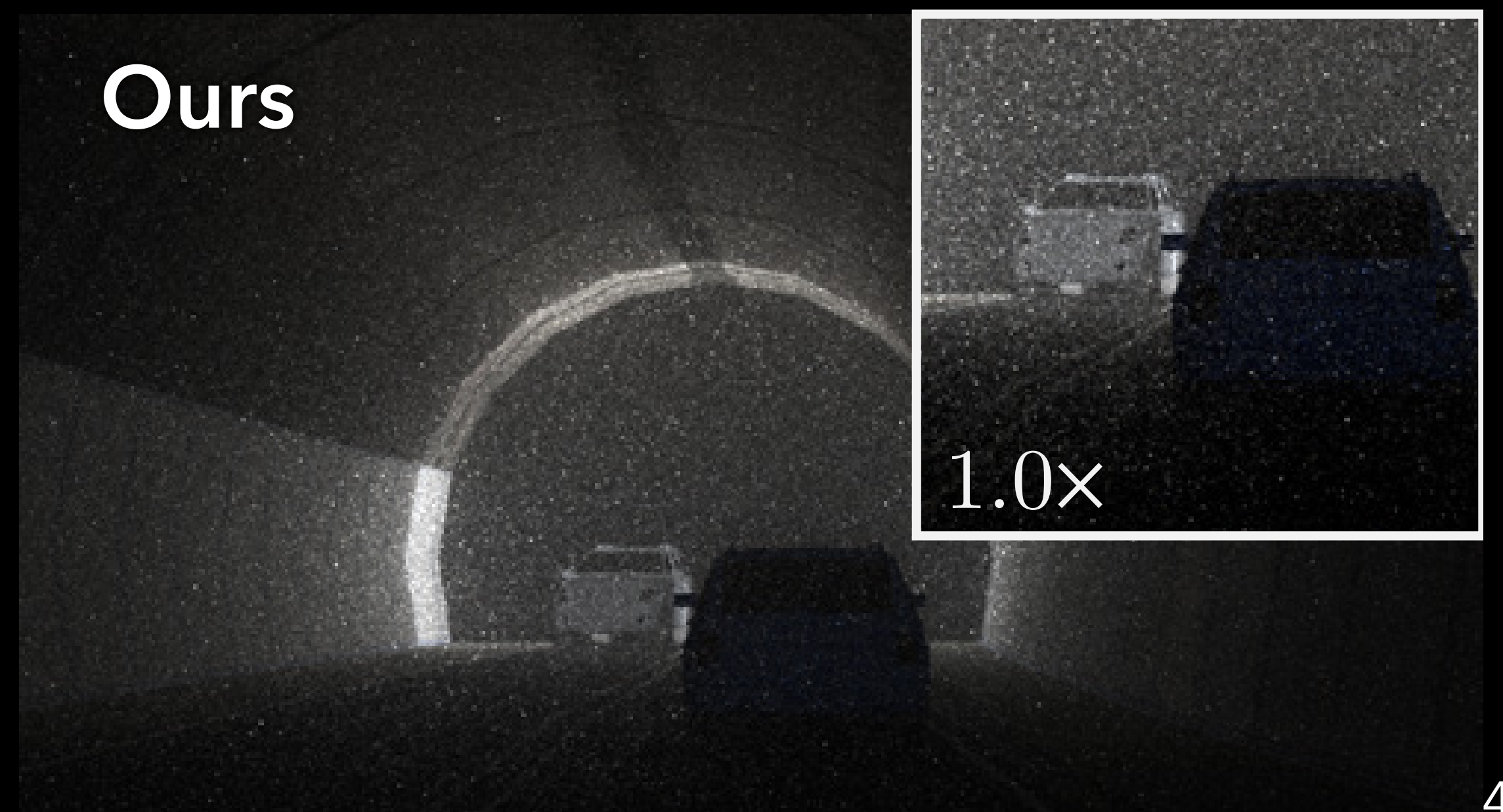
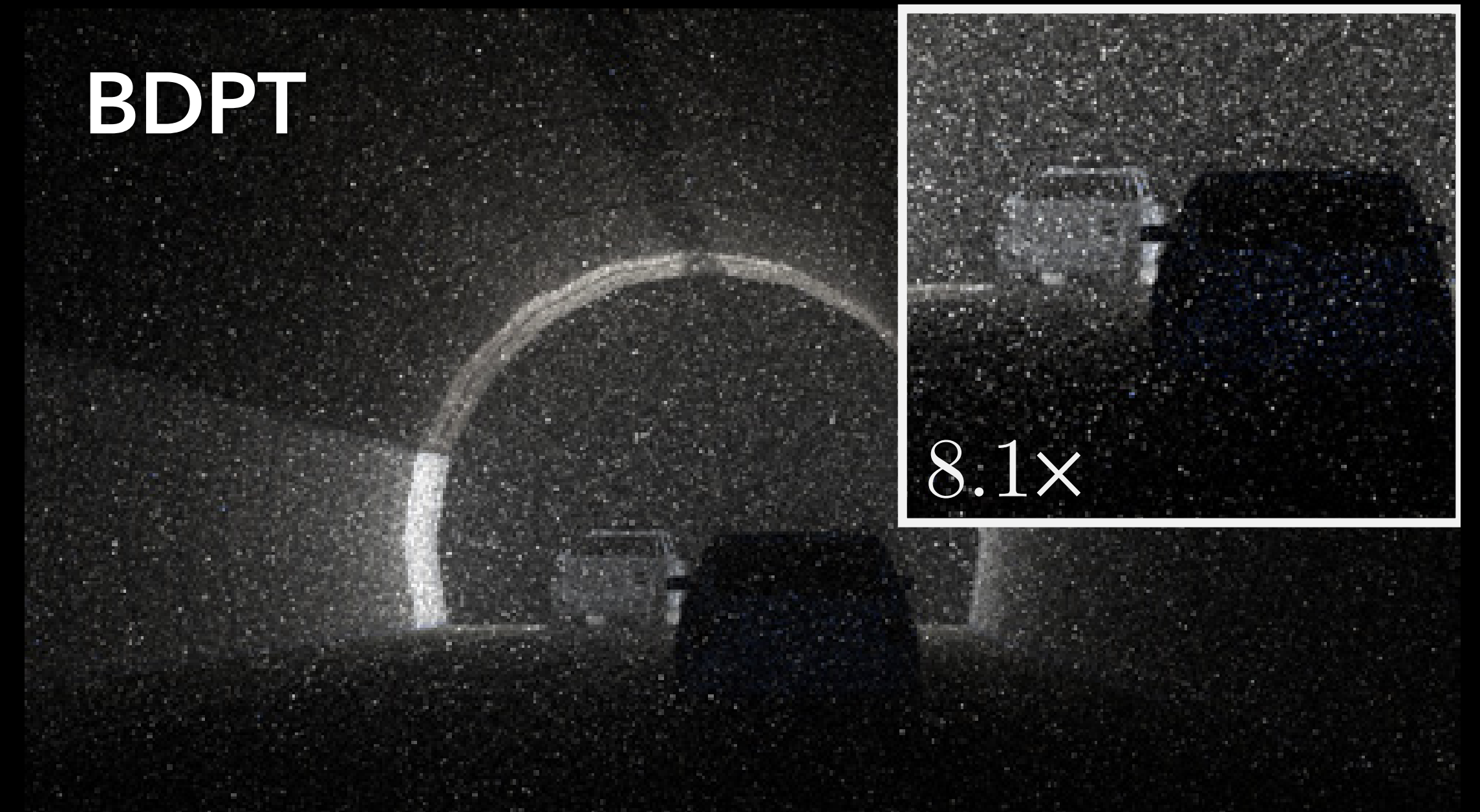




Gate 1



Gate 2



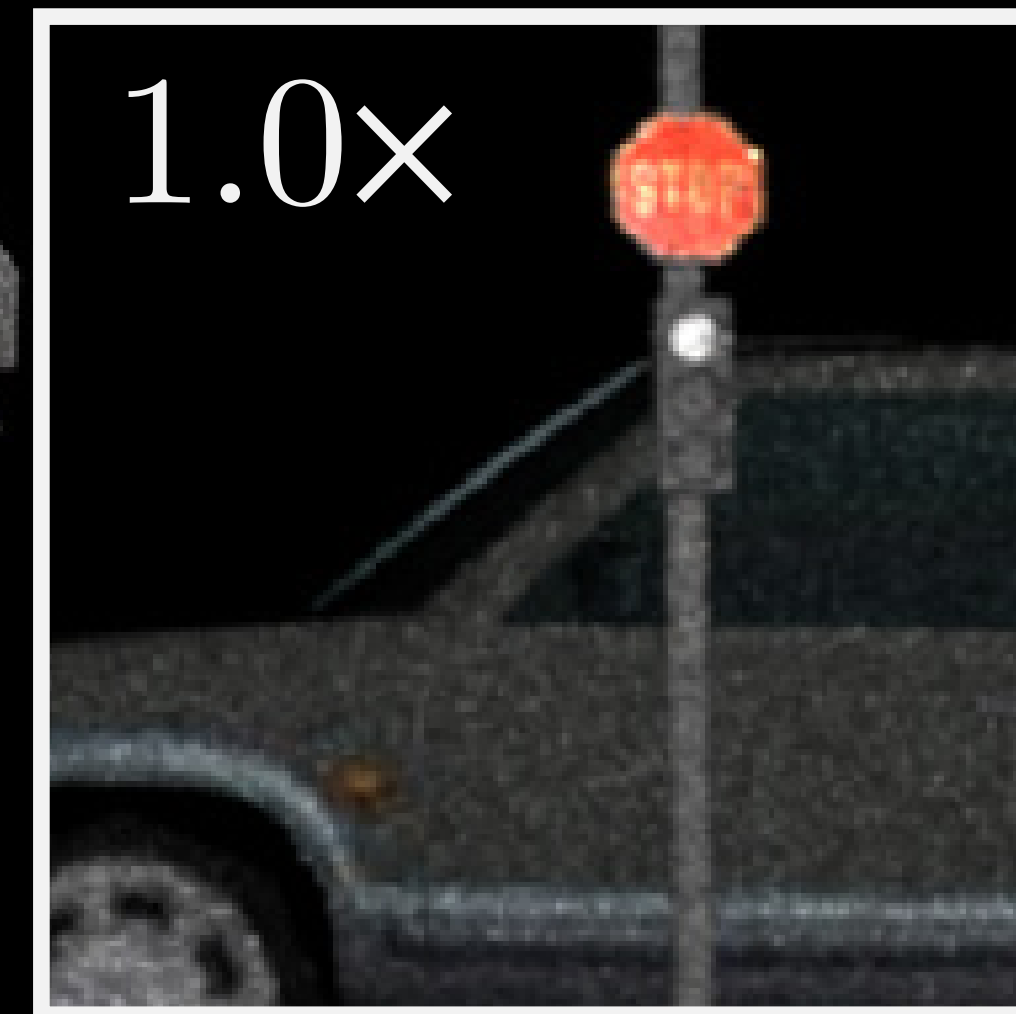
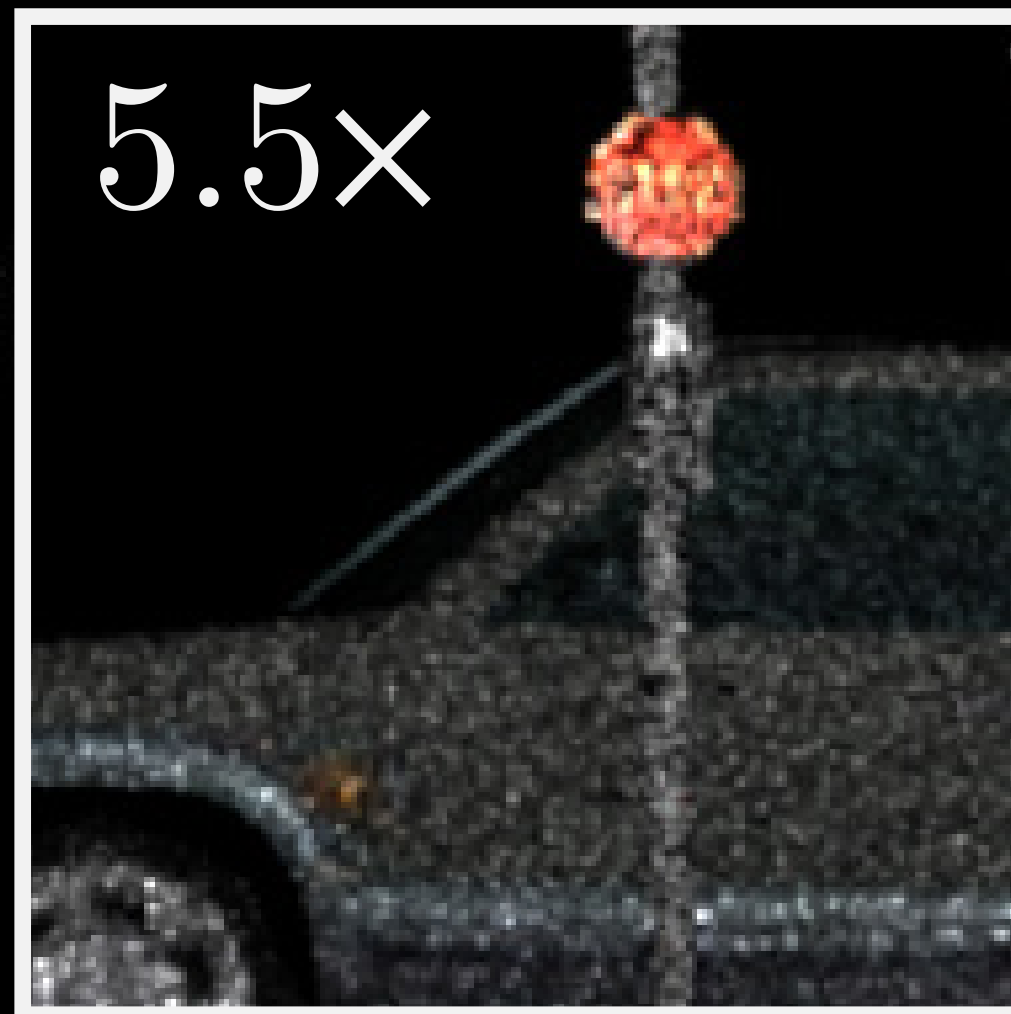
Proximity Detection

Steady-state



BDPT

Ours



Conclusion

1. **Formulated** time-gated rendering with light-cone shells
(*global path constraint* $\rightarrow$ *a single region in space-time*)

2. **Accelerated** light-cone shell queries (*overlap-driven traversal*)

Future work:

- Caustics via VCM/UPS [Georgiev et al. 2012, Hachisuka et al. 2012]
- Resampling-aware MIS [Tokuyoshi et al. 2019, Nabata et al. 2020, Su et al. 2022]

Light-Cone Shell Queries for Scalable Time-Gated Rendering

Thank you!

Project page



Paper + code



Juhyeon Kim



Adithya Pediredla



Wojciech Jarosz

This work was supported by NSF
award 2403122

