Differentiable Ray Tracing for Radio Propagations

Jérome Eertmans - April 16th, Milano

About the author

About the author

Profile:

- PhD student at UCLouvain since 09/2021;
- Electromechaninal Engineer in mechatronics;
- introduced to Ray Tracing (RT) during a student job.

Interests:

- Programming (mainly Python and Rust);
- Writing performant solutions;
- and open sourcing content (jeertmans on GitHub or eertmans.be).

Claude Oestges

Laurent Jacques

Disclaimer

$$\text{BS} \leftrightarrow \text{TX}$$

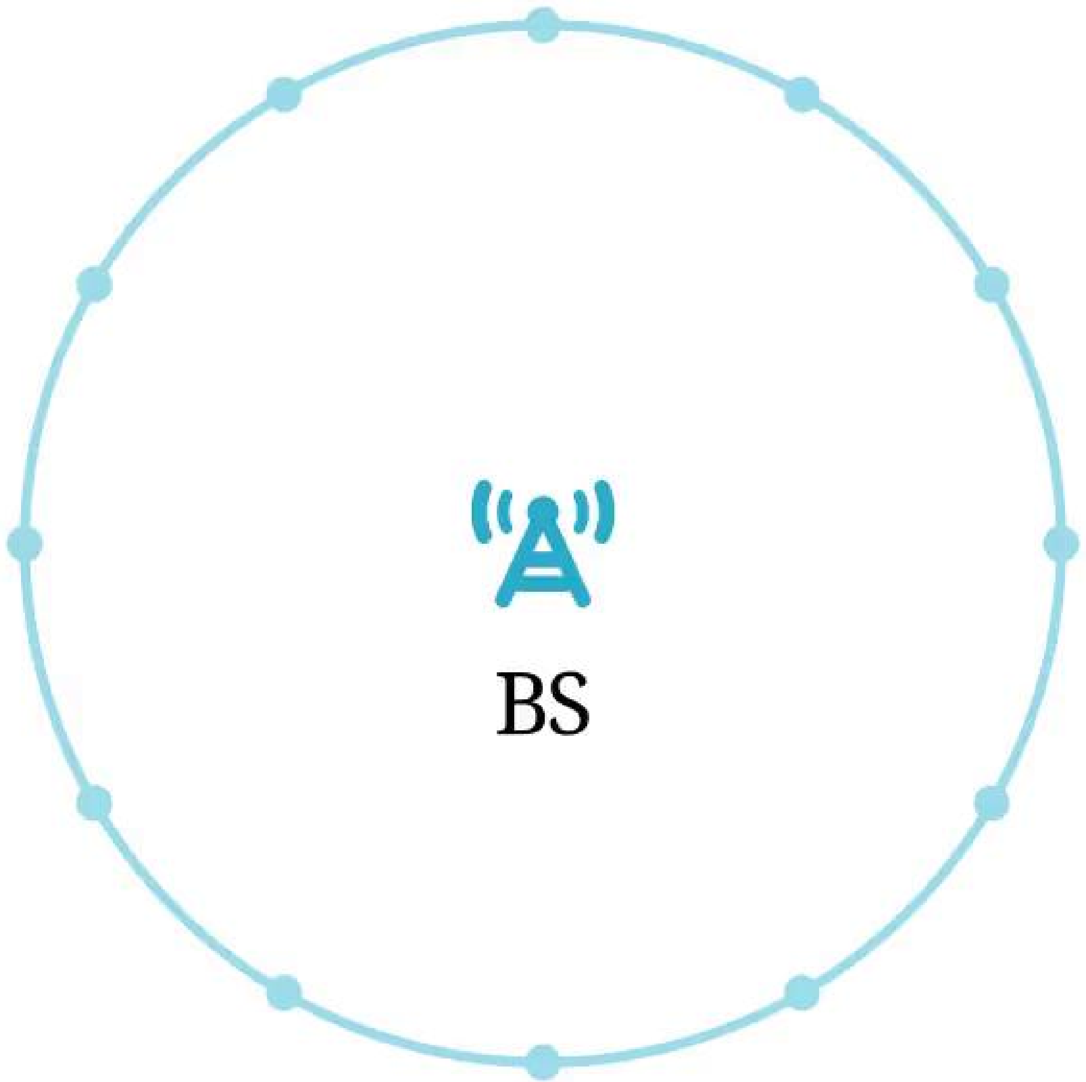
$$\text{UE} \leftrightarrow \text{RX}$$

Ray Tracing's ABC

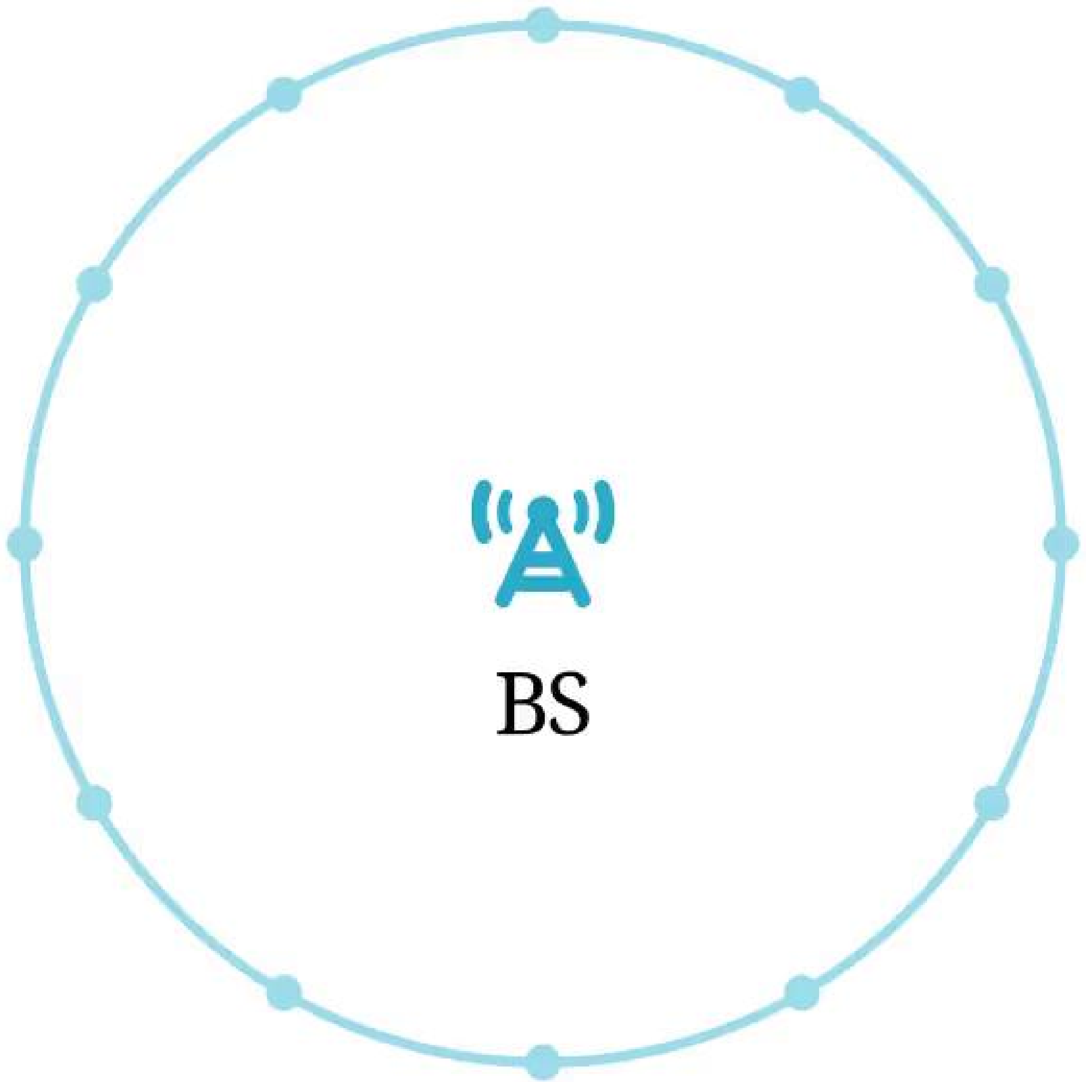
BS

Ray Tracing's ABC

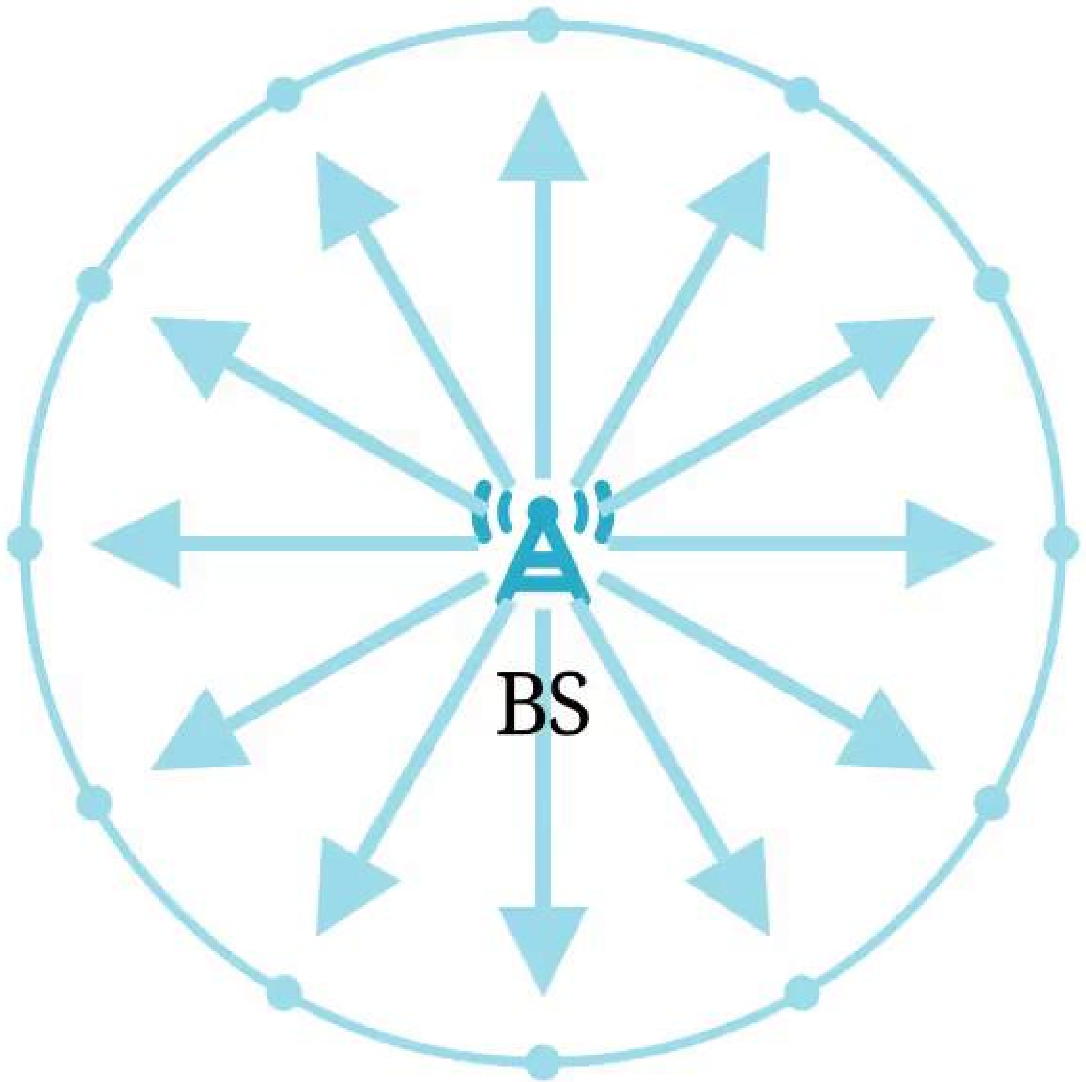
Ray Tracing's ABC

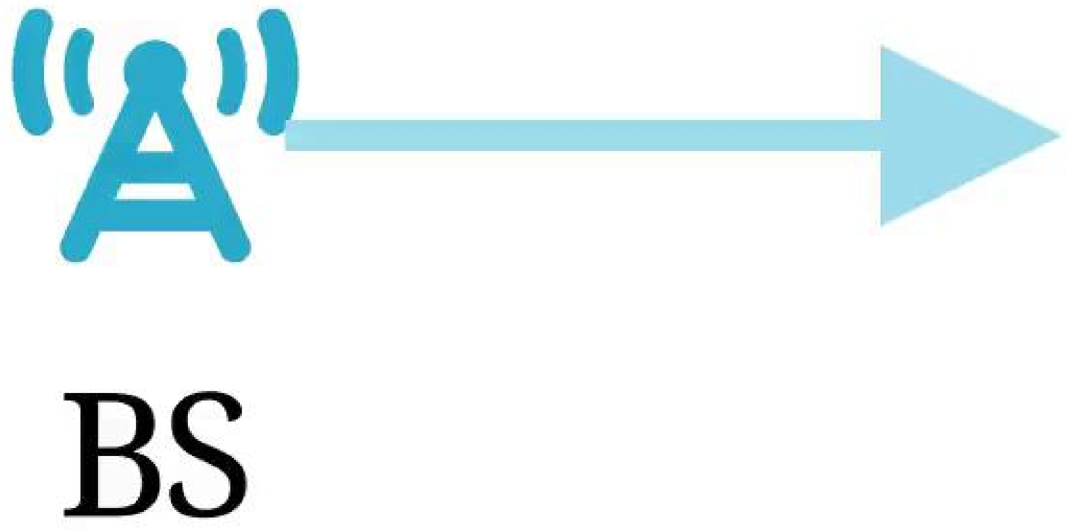


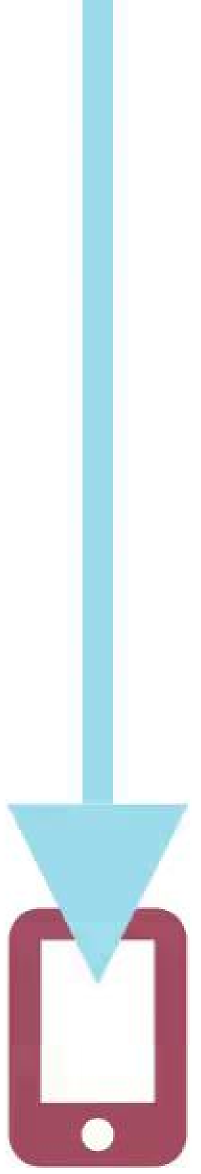
Ray Tracing's ABC



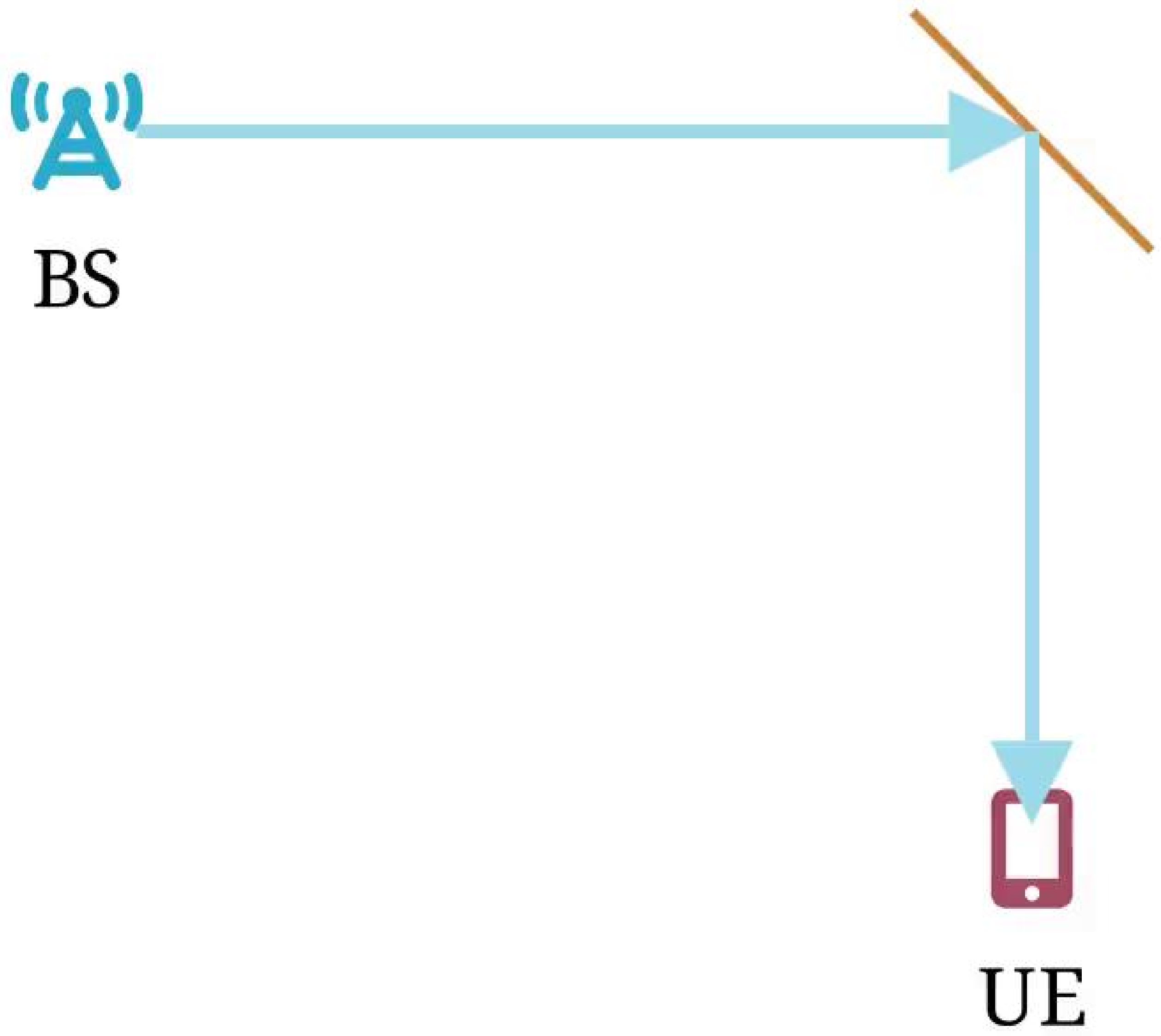
Ray Tracing's ABC



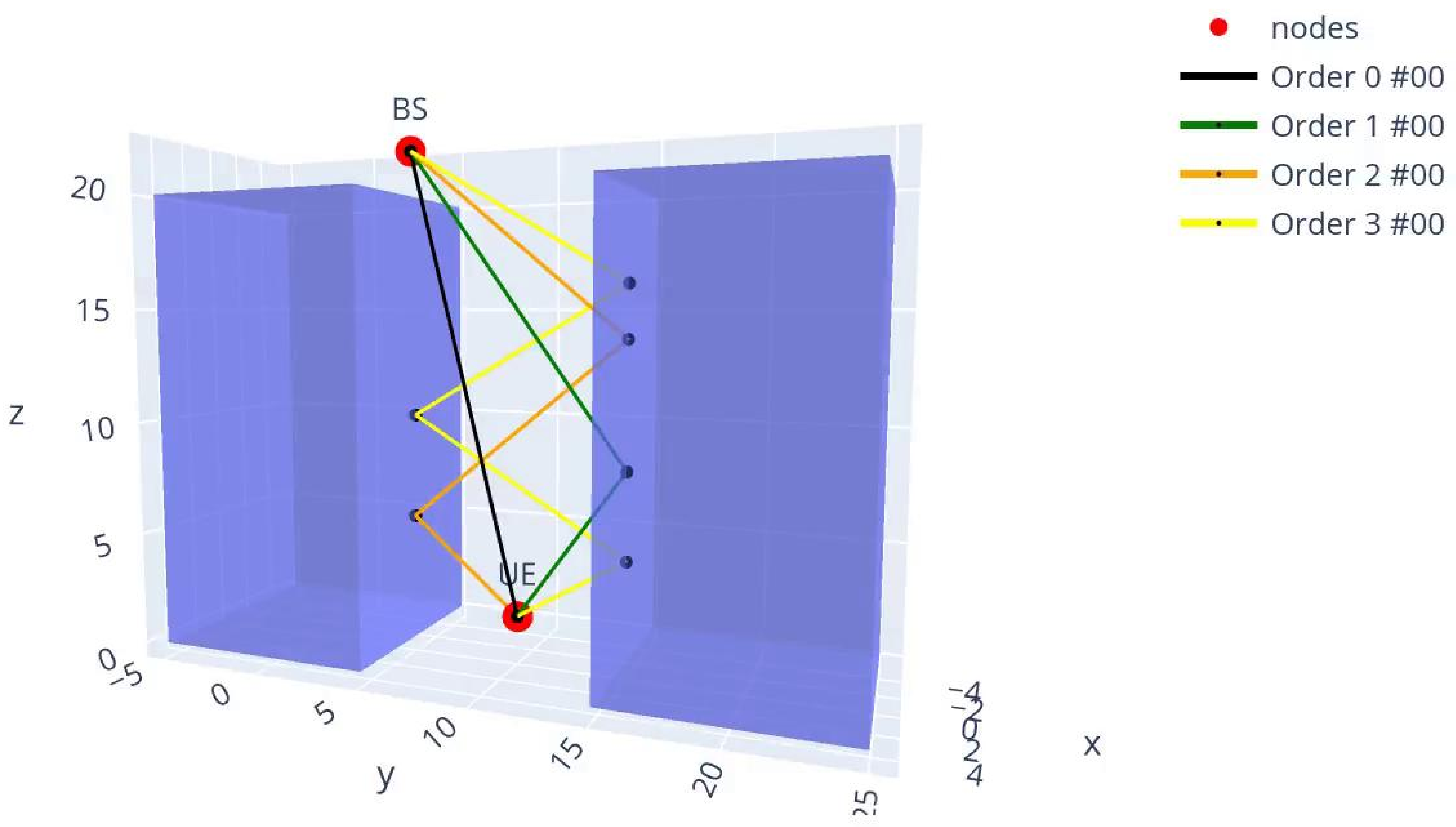

BS


UE

Ray Tracing's ABC



Ray Tracing's ABC



Ray Tracing's ABC

Electrical and Magnetic fields

$\vec{E}\ (\mathrm{V\,m^{-1}})\ \&\ \vec{B}\ (\mathrm{T})$

Ray Tracing's ABC

Electrical and Magnetic fields

$$\vec{E}\ (\mathrm{V\,m^{-1}})\ \&\ \vec{B}\ (\mathrm{T})$$

$$\vec{E}(x, y, z) = \sum_{\mathcal{P}\in\mathcal{S}} \bar{C}(\mathcal{P}) \cdot \vec{E}(\mathcal{P}_1),$$

Electrical and Magnetic fields

$$\vec{E}\ (\mathrm{V\,m^{-1}})\ \&\ \vec{B}\ (\mathrm{T})$$

$$\vec{E}(x,y,z) = \sum_{\mathcal{P}\in\mathcal{S}} \bar{C}(\mathcal{P}) \cdot \vec{E}(\mathcal{P}_1),$$

$$\text{where } \bar{C}(\mathcal{P}) = \prod_{i\in\mathcal{I}} \bar{D}_i \cdot \alpha_i \cdot e^{-j\phi_i}.$$

Ray Tracing's ABC

Input scene

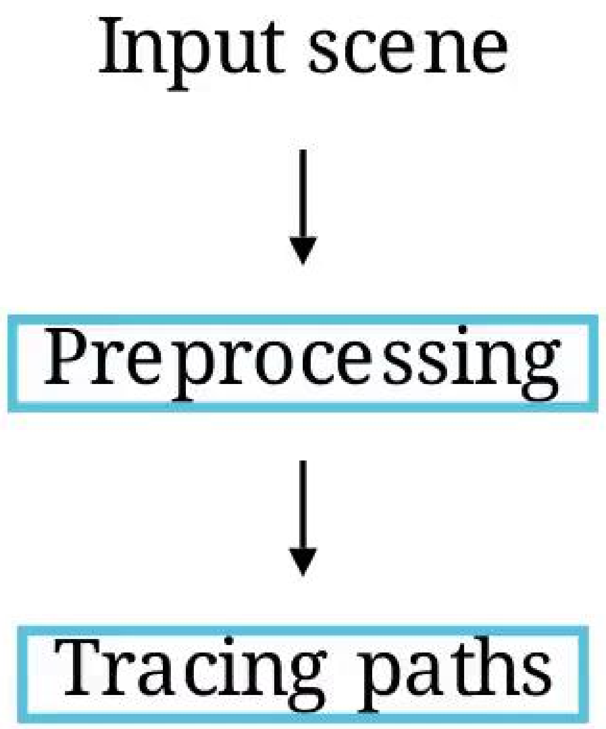
Ray Tracing's ABC

Input scene

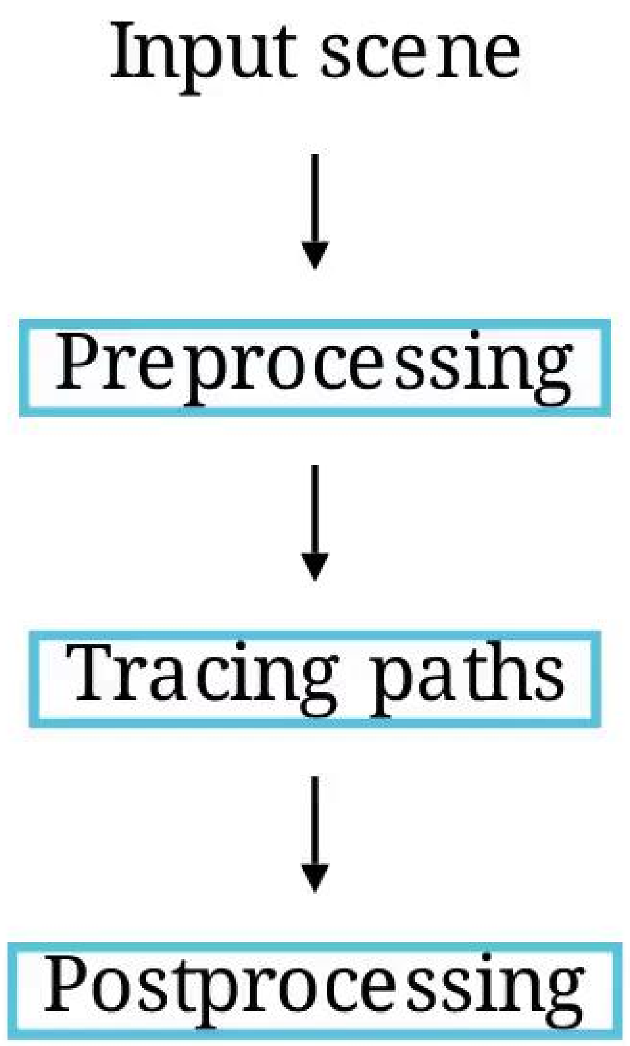
↓

Preprocessing

Ray Tracing's ABC



Ray Tracing's ABC



Ray Tracing's ABC

Input scene

↓

Preprocessing

↓

Tracing paths

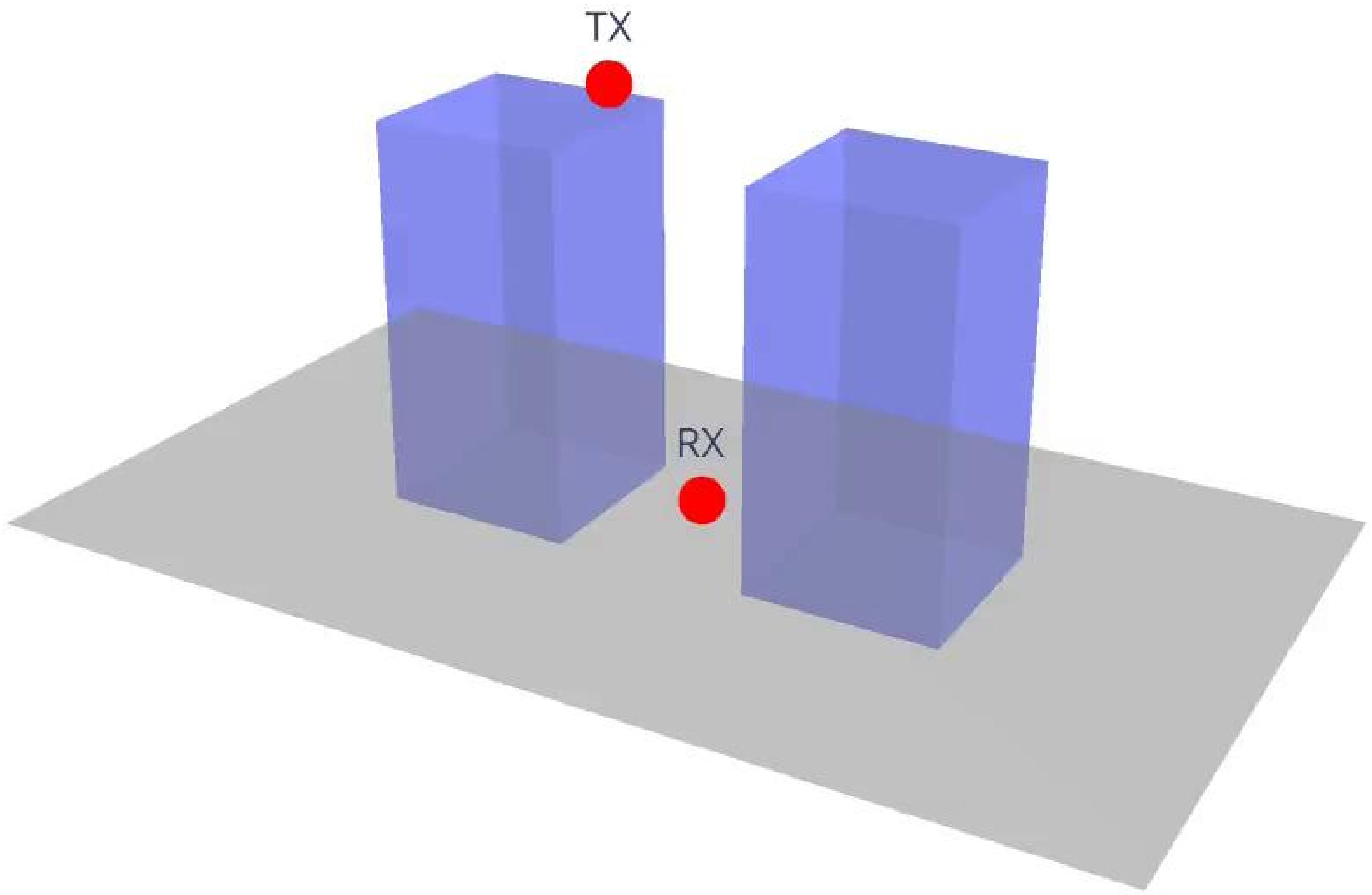
↓

Postprocessing

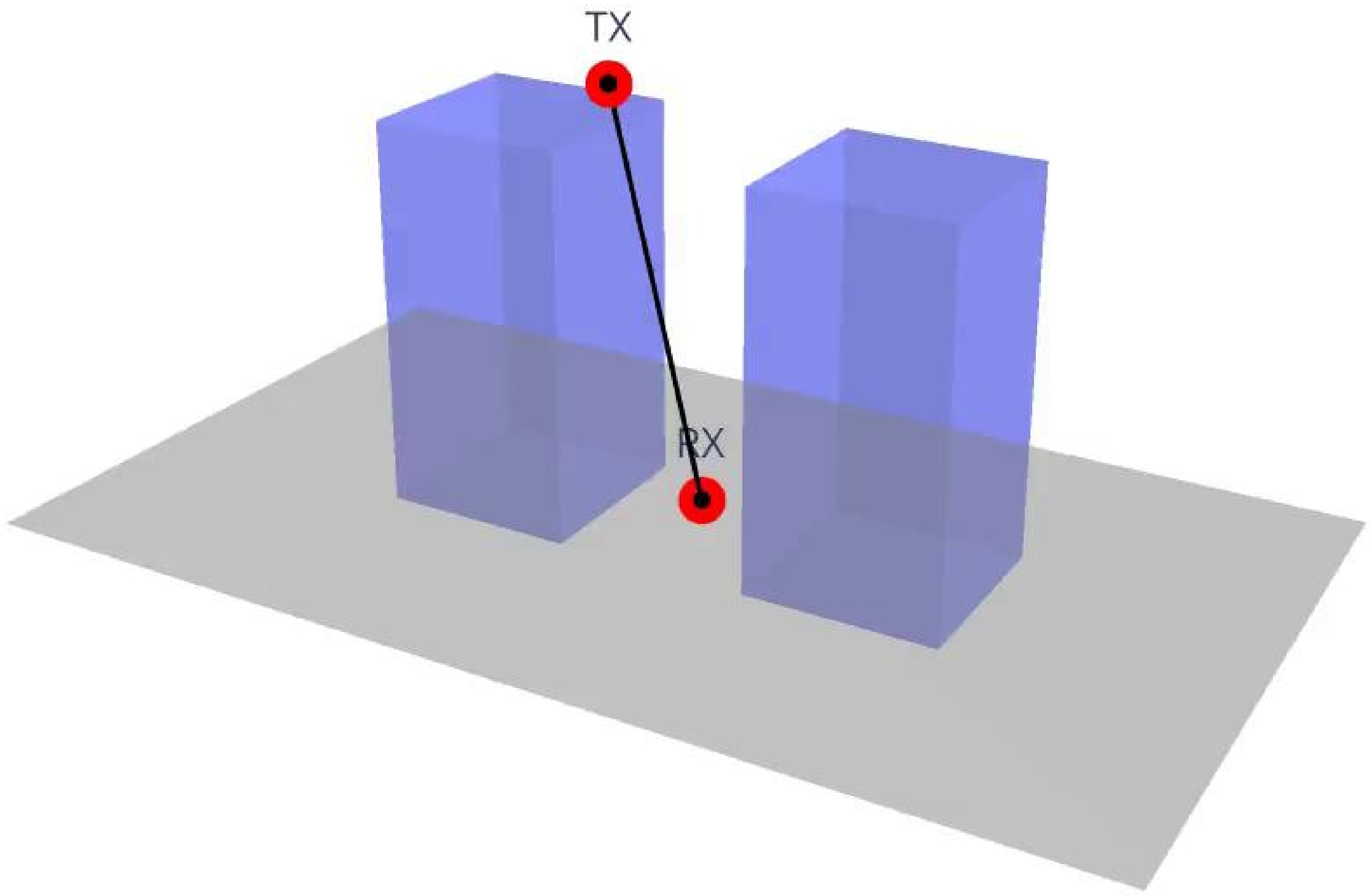
↓

EM fields

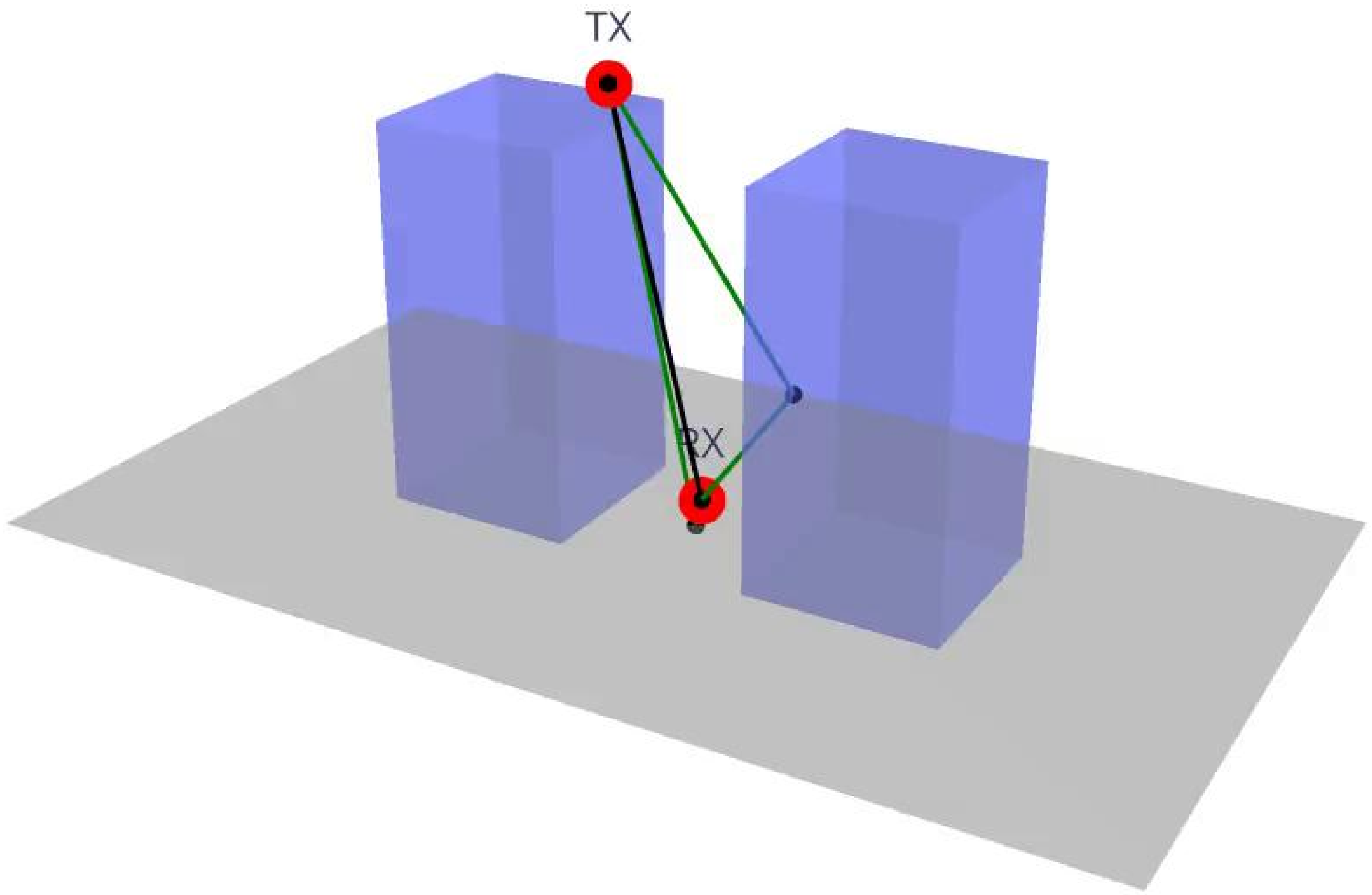
Tracing paths



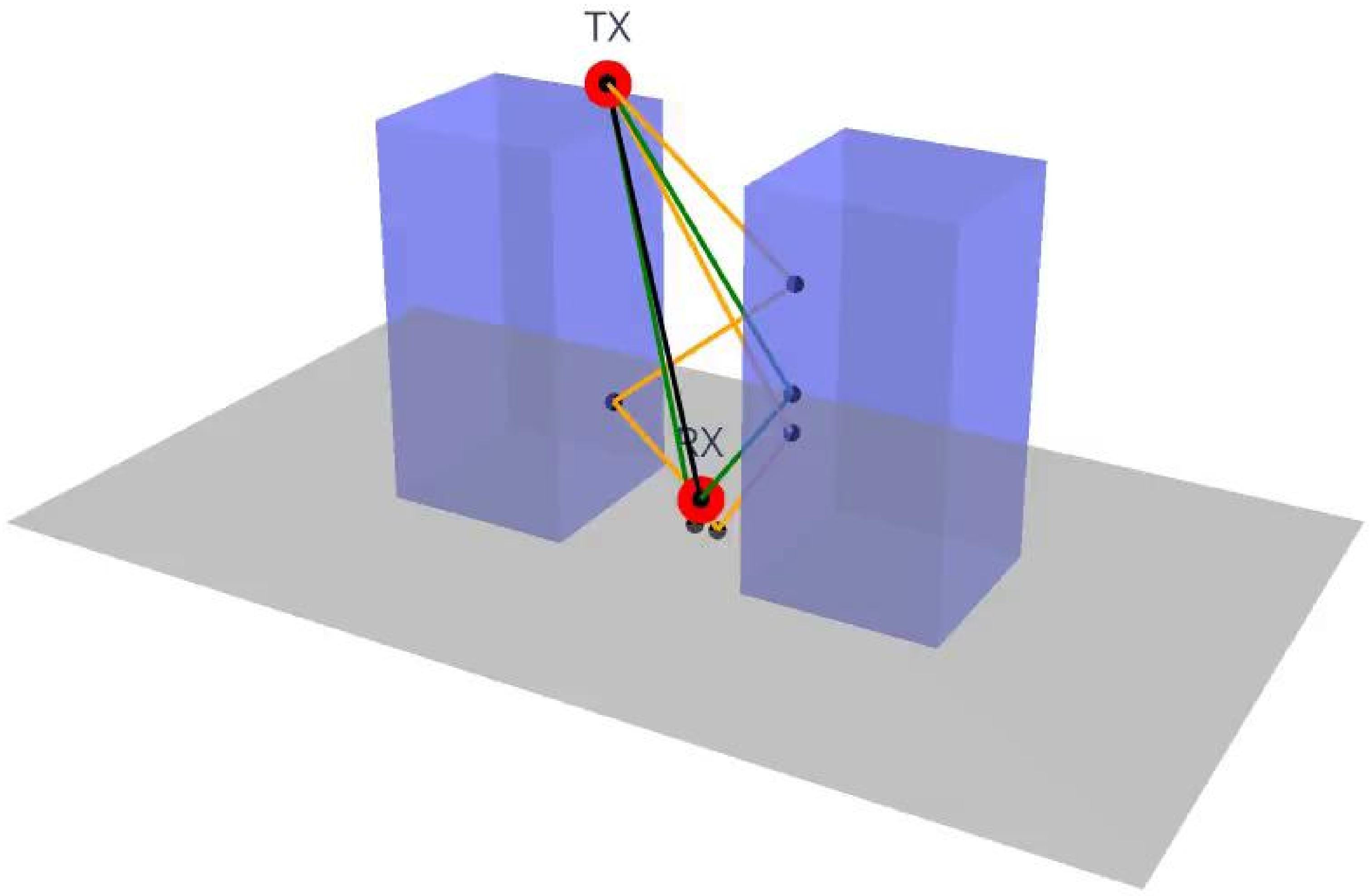
Tracing paths



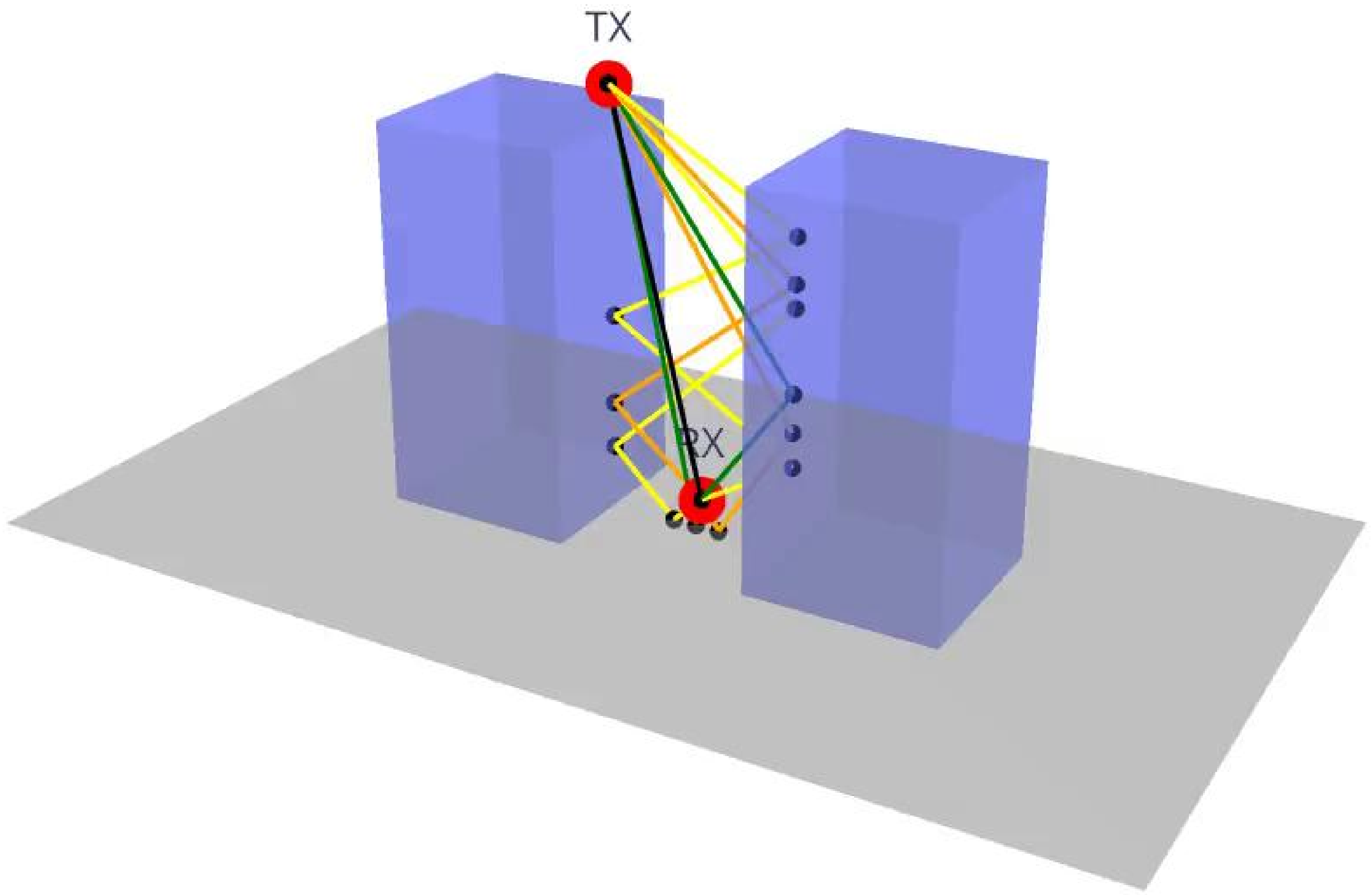
Tracing paths



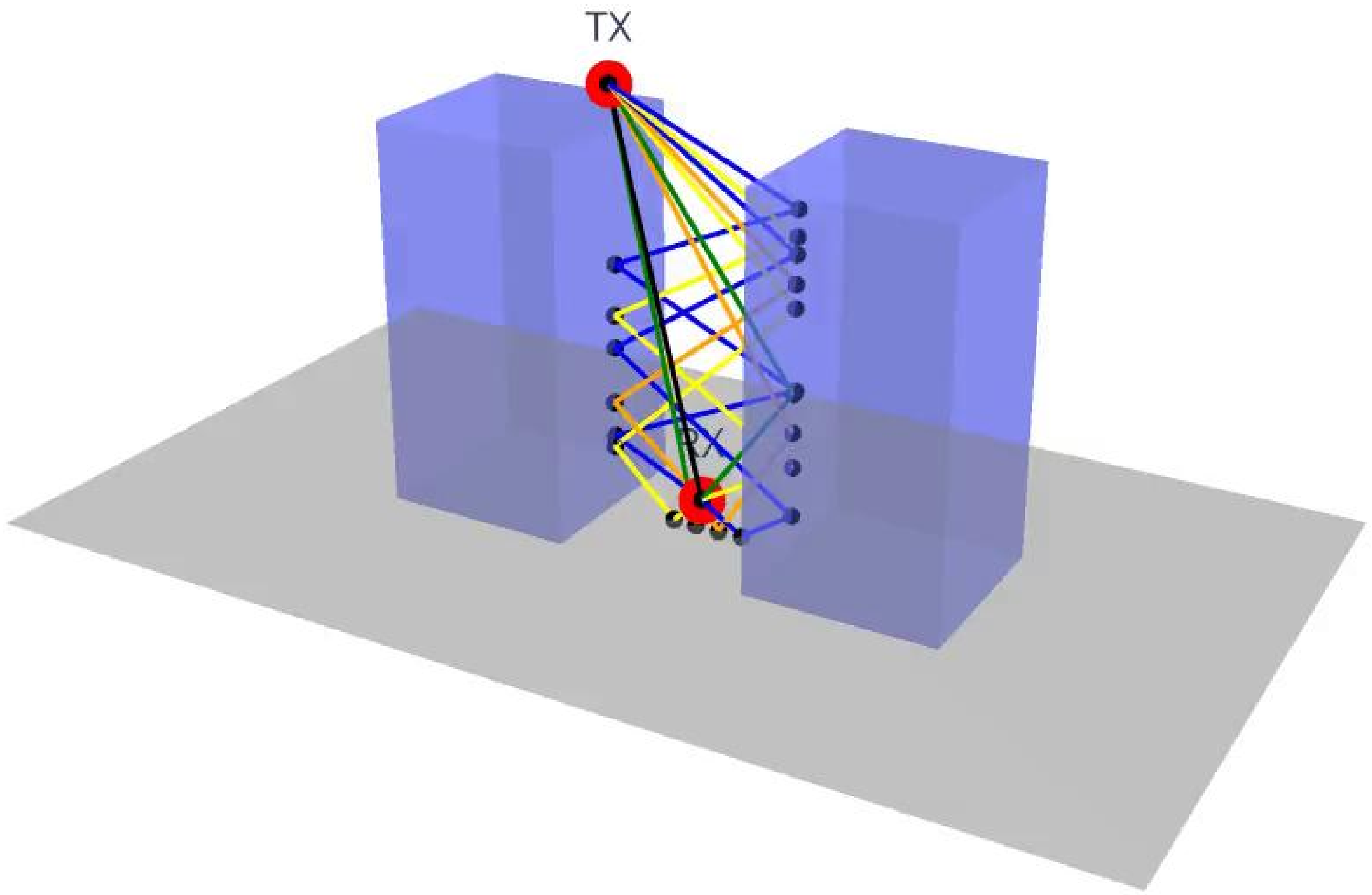
Tracing paths



Tracing paths



Tracing paths



.. previous work (EuCAP 2023)

- Image Method: refl. on planar surfaces;

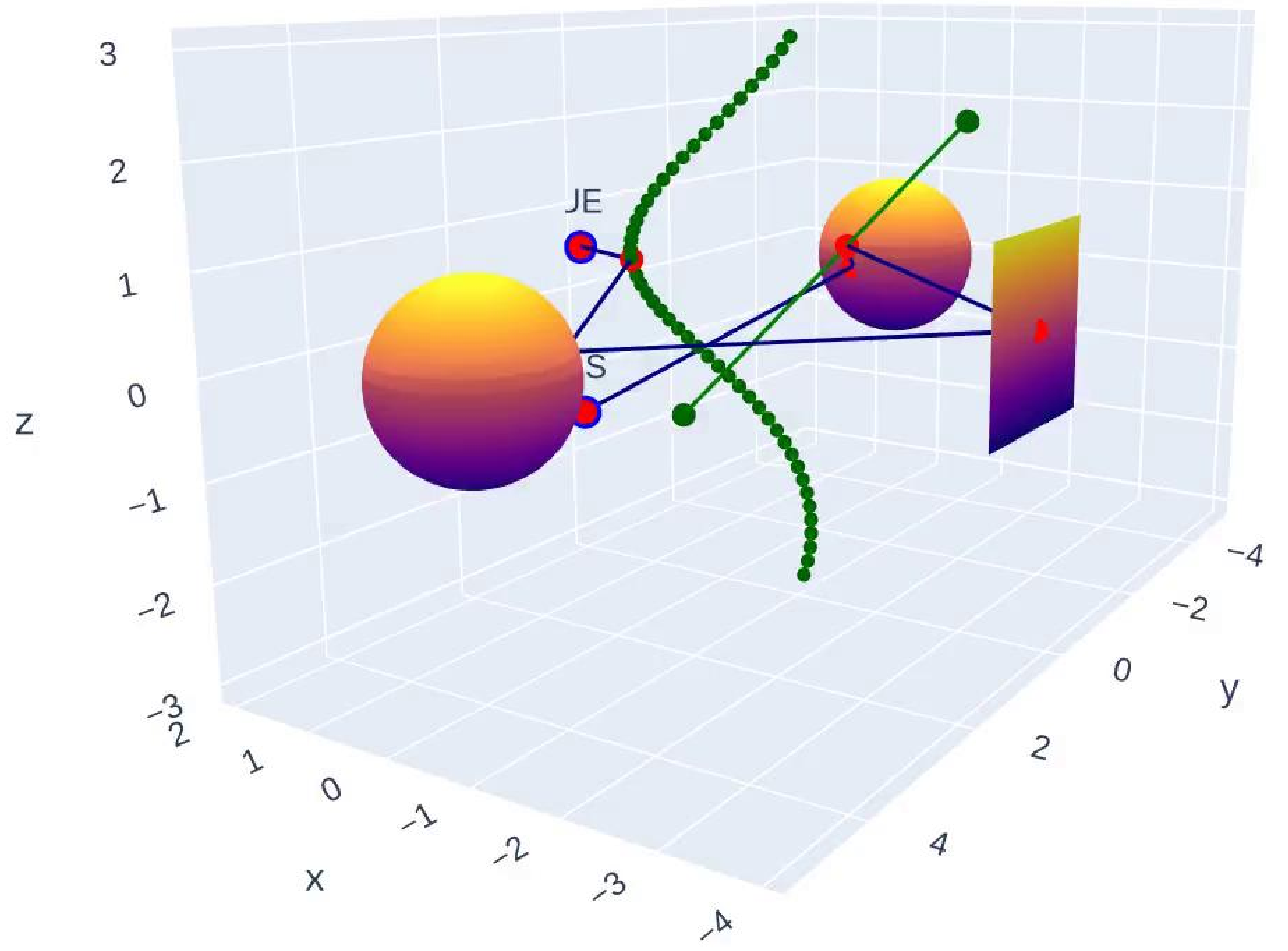
.. previous work (EuCAP 2023)

- Image Method: refl. on planar surfaces;
- Fermat-based min.: refl. and diff., etc. (convex on planar surfaces);

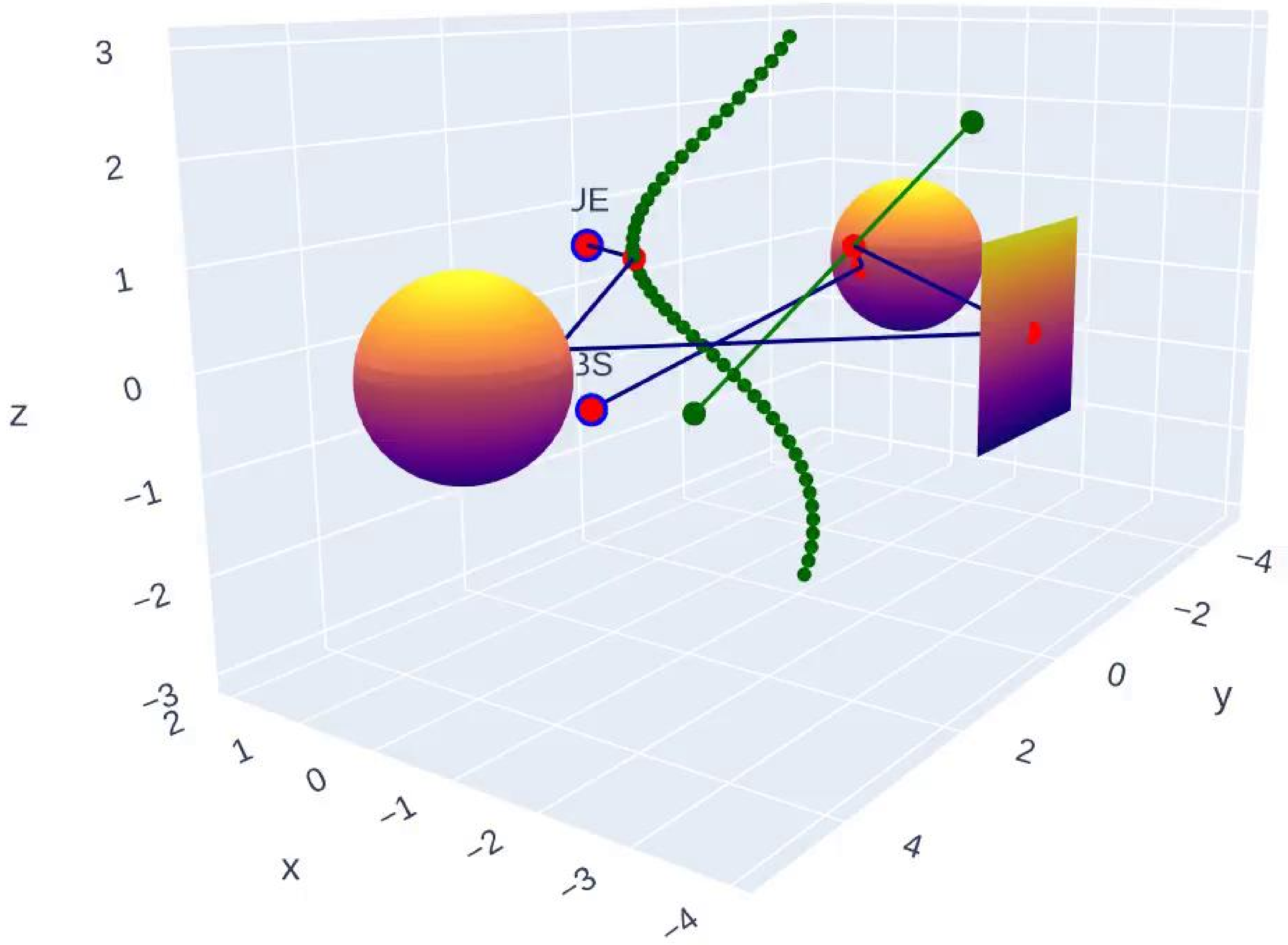
.. previous work (EuCAP 2023)

- Image Method: refl. on planar surfaces;
- Fermat-based min.: refl. and diff., etc. (convex on planar surfaces);
- Min-Path-Tracing: refl., diff., etc on any object.

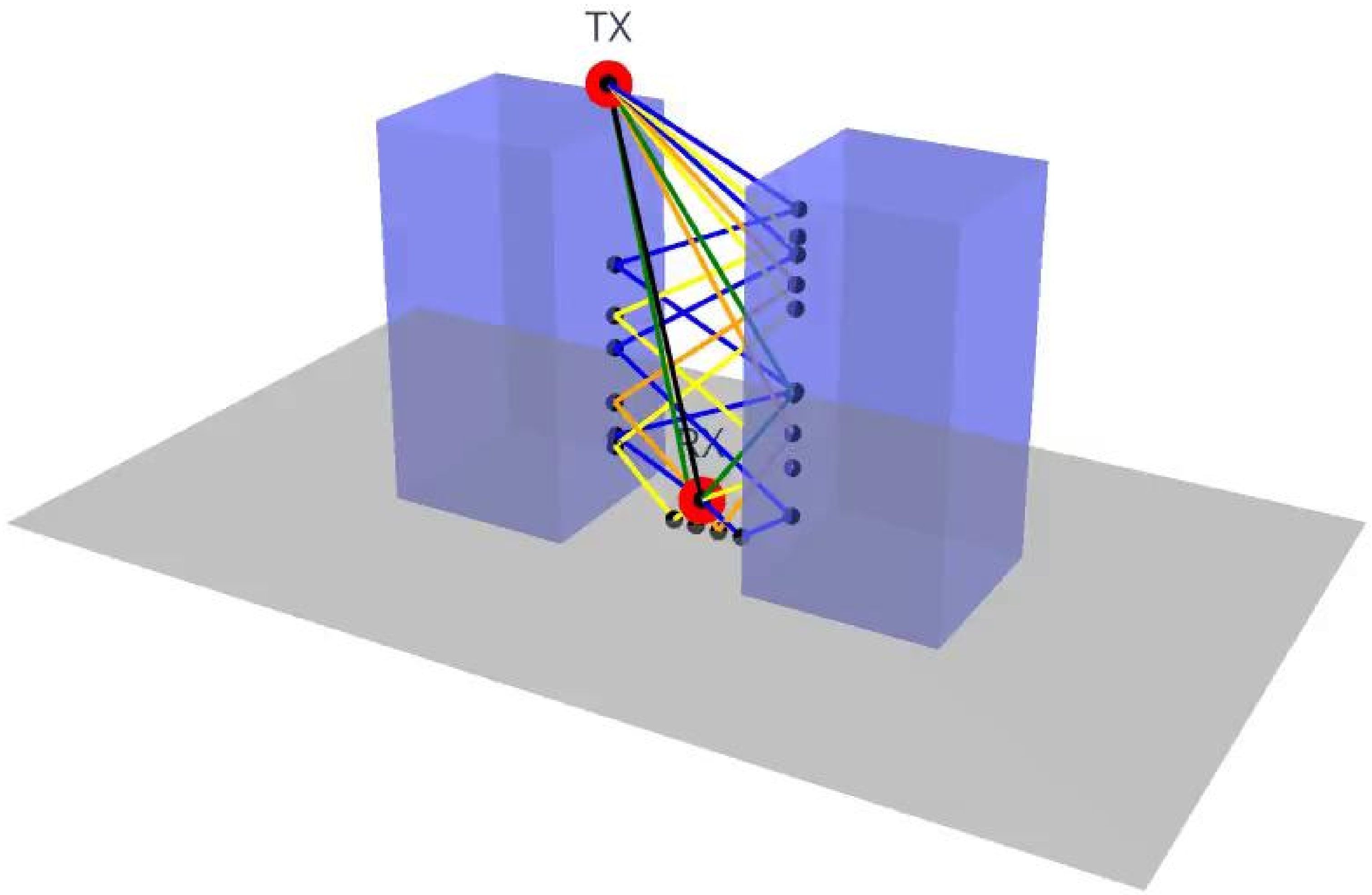
.. previous work (EuCAP 2023)



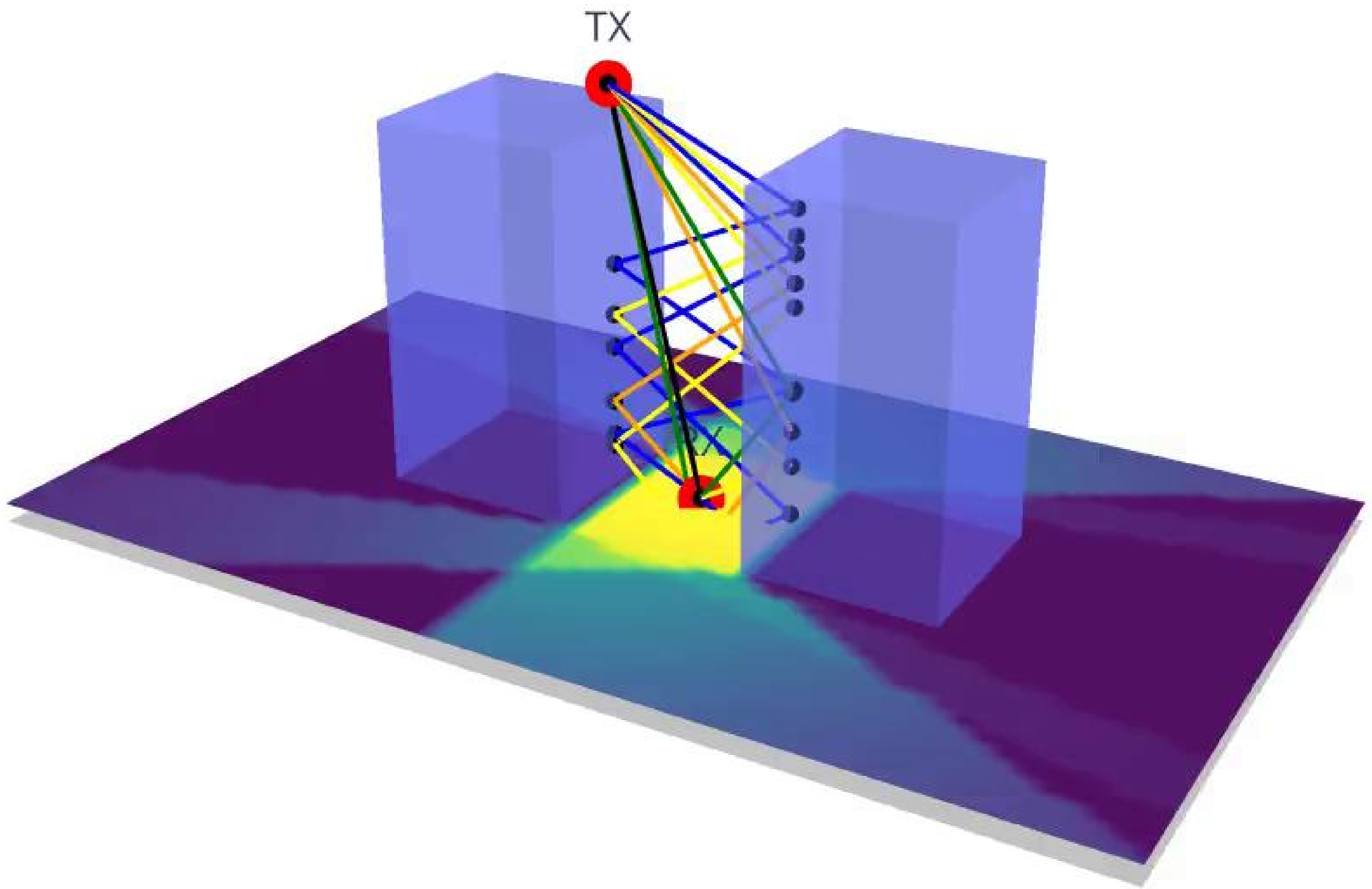
.. previous work (EuCAP 2023)



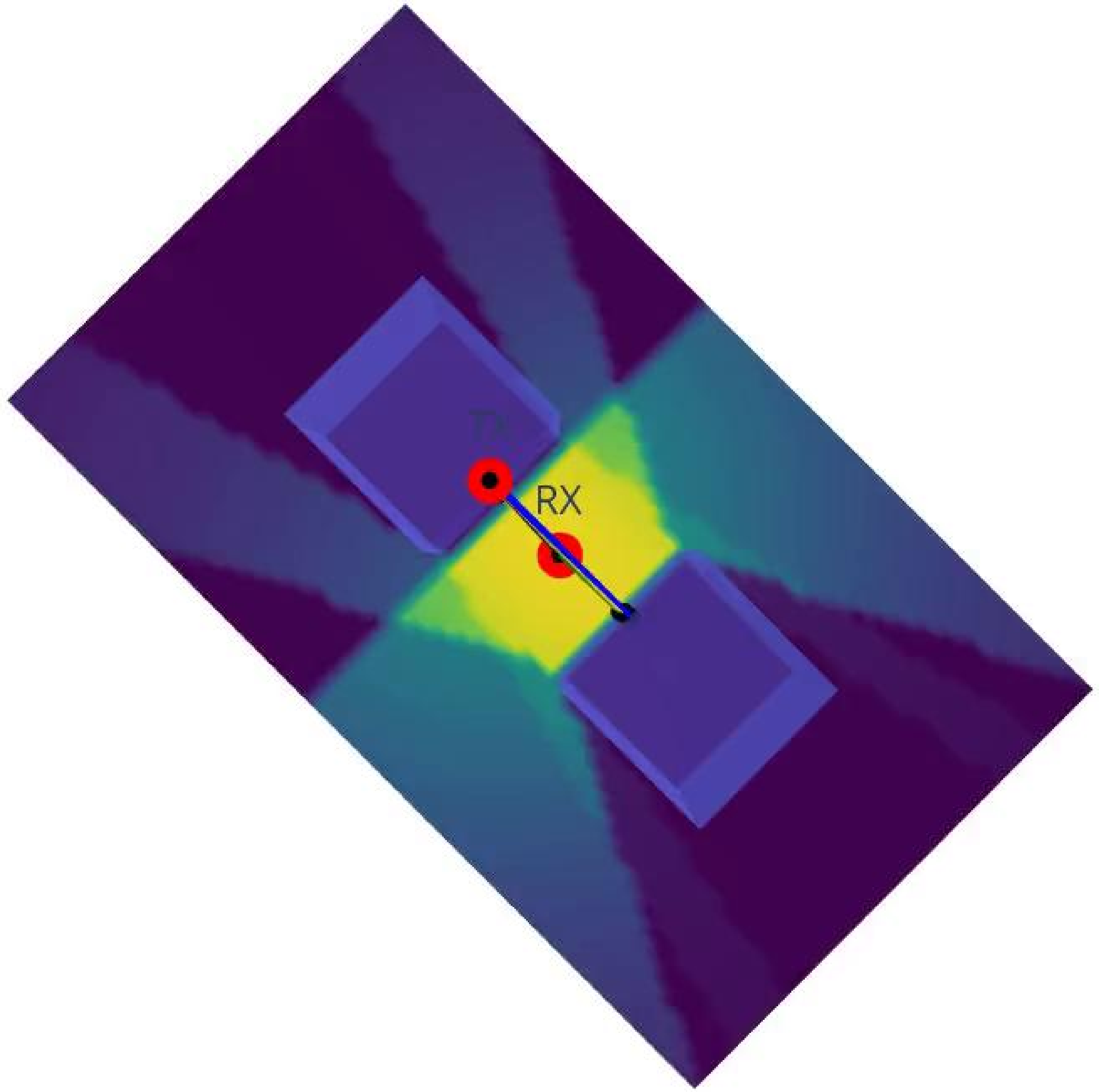
Differentiable Ray Tracing



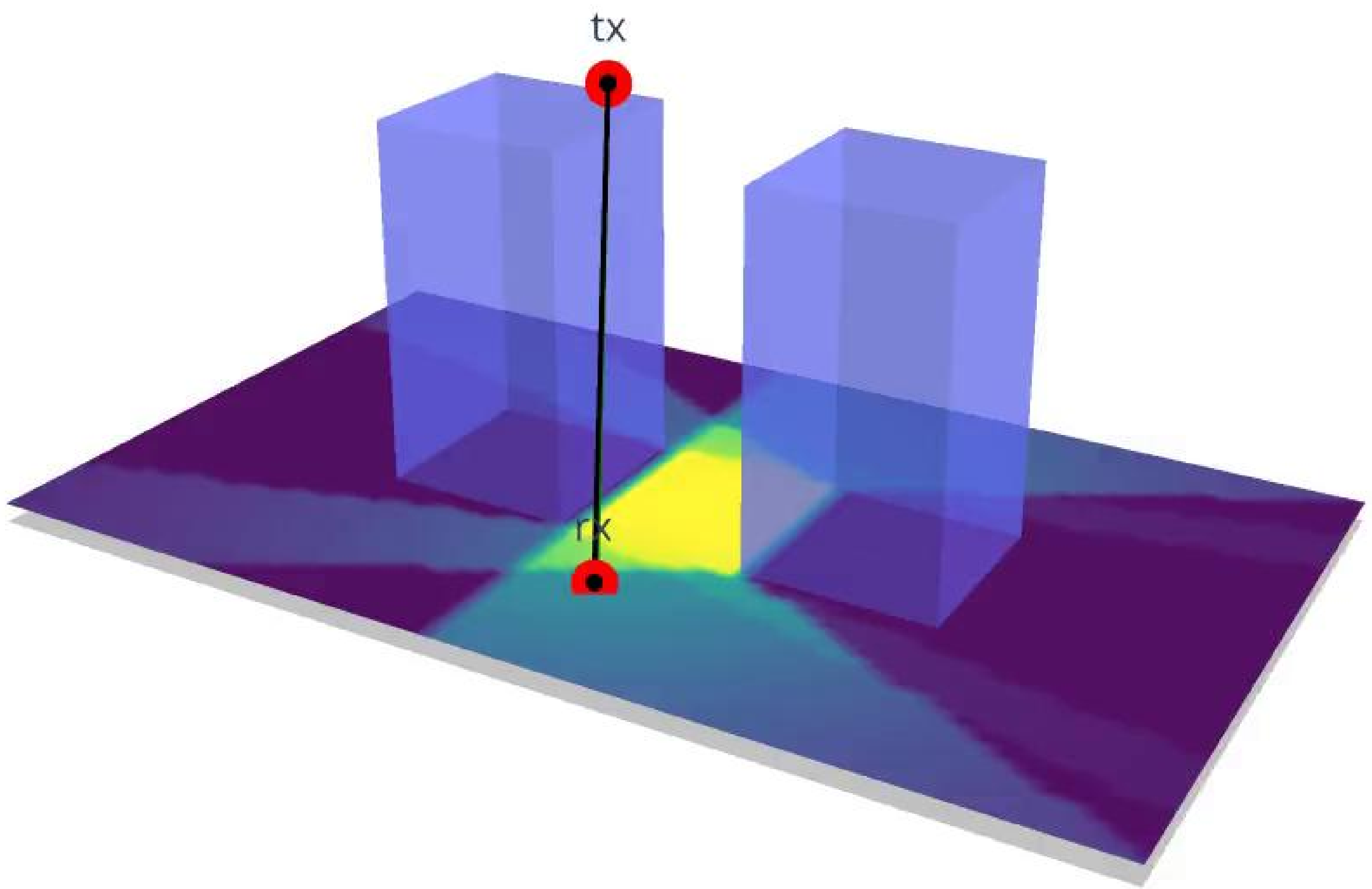
Differentiable Ray Tracing



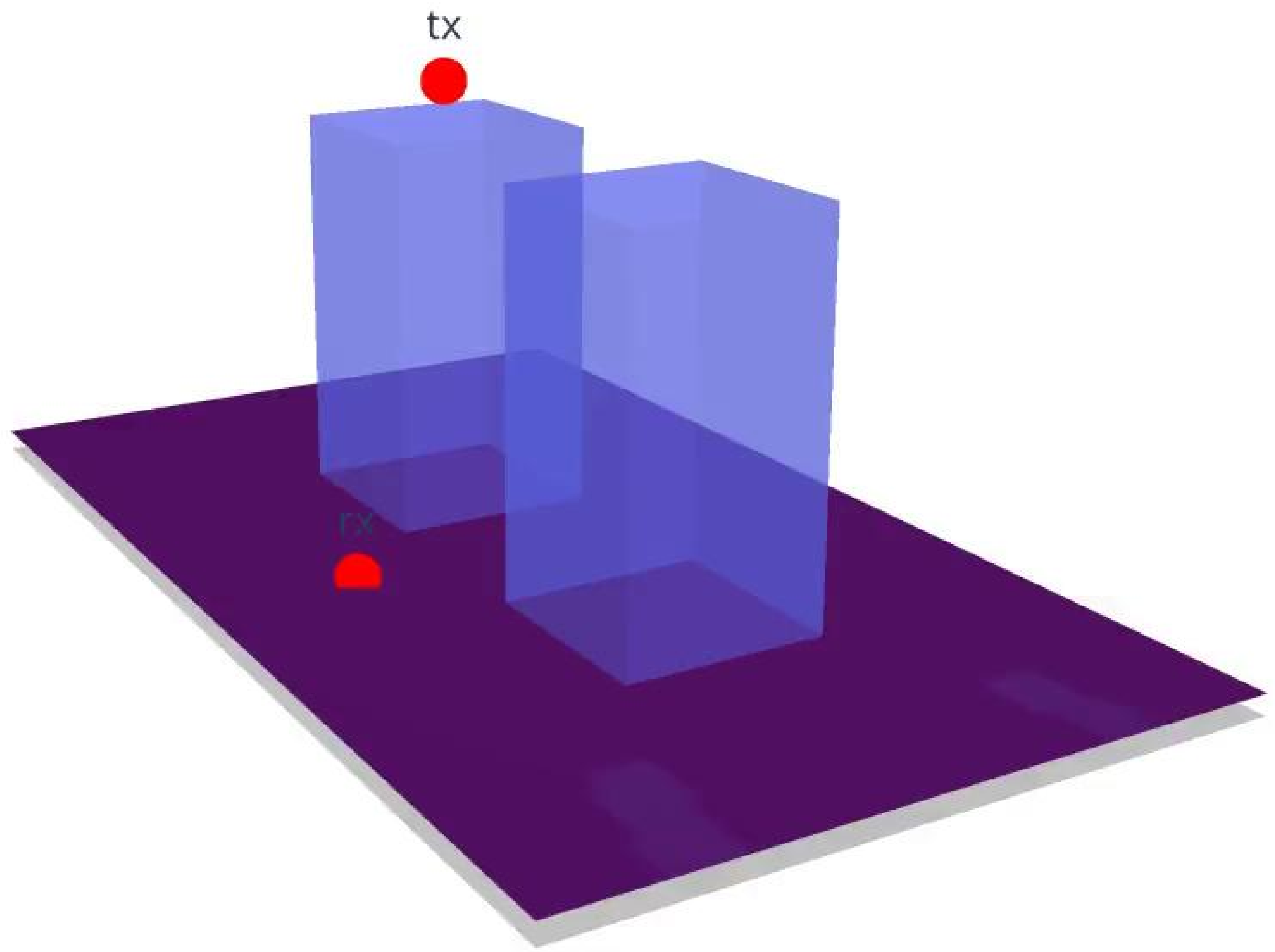
Differentiable Ray Tracing



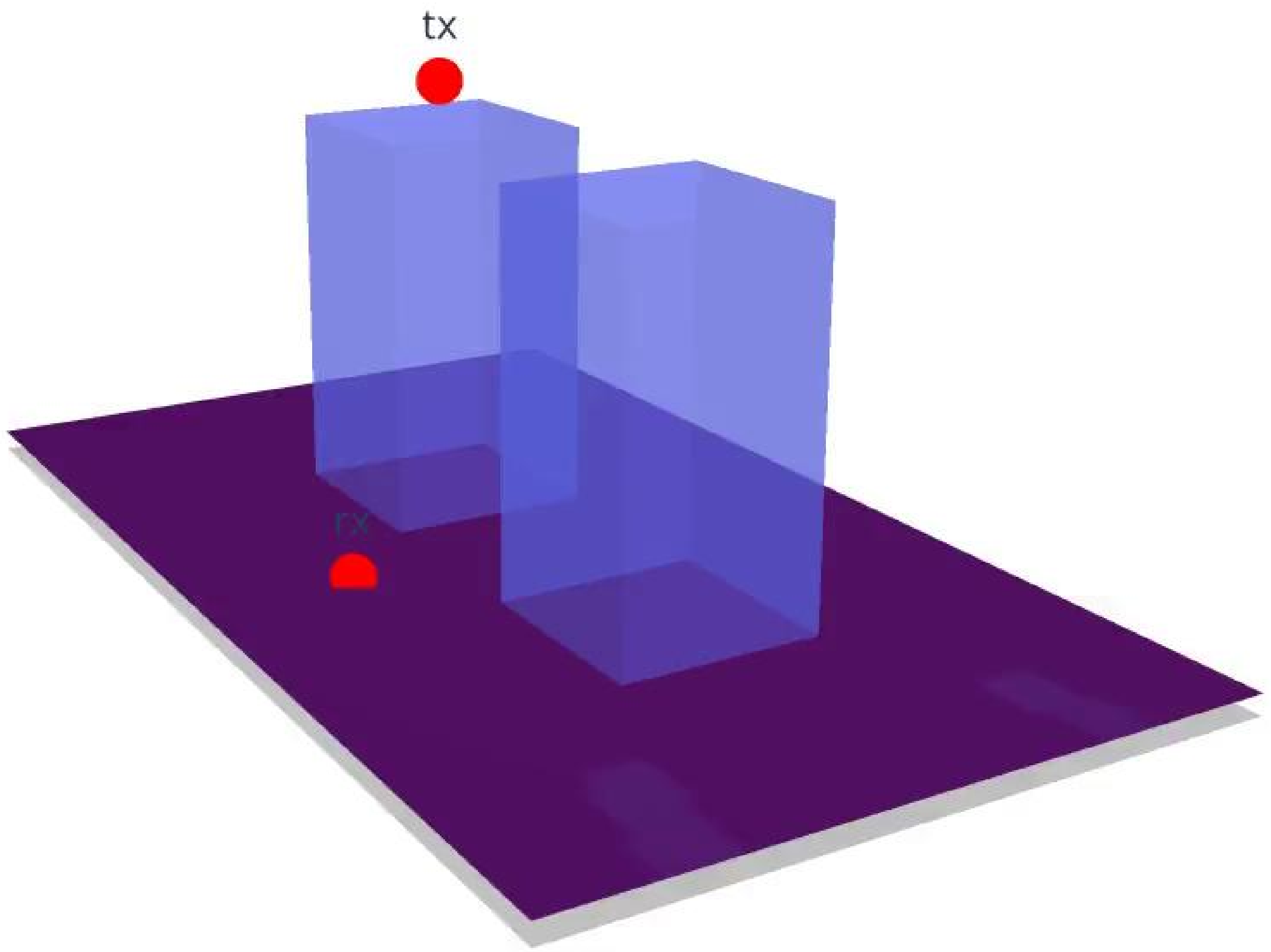
Differentiable Ray Tracing



Differentiable Ray Tracing



Differentiable Ray Tracing



One solution: differentiability.

Differentiable Ray Tracing

Sionna RT: Differentiable Ray Tracing for Radio Propagation Modeling

Jakob Hoydis, Fayçal Aït Aoudia, Sebastian Cammerer, Merlin Nimier-David, Nikolaus Binder, Guillermo Marcus, and Alexander Keller

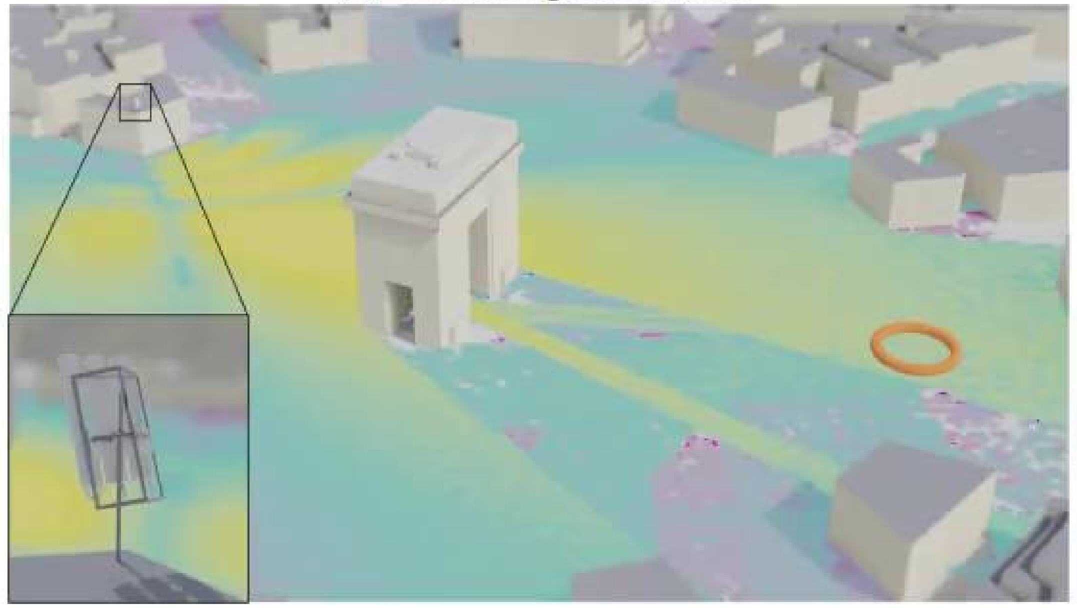
***Abstract*—Sionna™ is a GPU-accelerated open-source library for link-level simulations based on TensorFlow. Since release v0.14 it integrates a differentiable ray tracer (RT) for the simulation of radio wave propagation. This unique feature allows for the computation of gradients of the channel impulse response and other related quantities with respect to many system and environment parameters, such as material properties, antenna patterns, array geometries, as well as transmitter and receiver orientations and positions. In this paper, we outline the key components of Sionna RT and showcase example applications such as learning radio materials and optimizing transmitter orientations by gradient descent. While classic ray tracing is a crucial tool for 6G research topics like reconfigurable intelligent surfaces, integrated sensing and communications, as well as user localization, differentiable ray tracing is a key enabler for many novel and exciting research directions, for example, digital twins.**

Fig. 1: One of Sionna RT's example scenes. Data from [25].

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing

(a) Before optimization

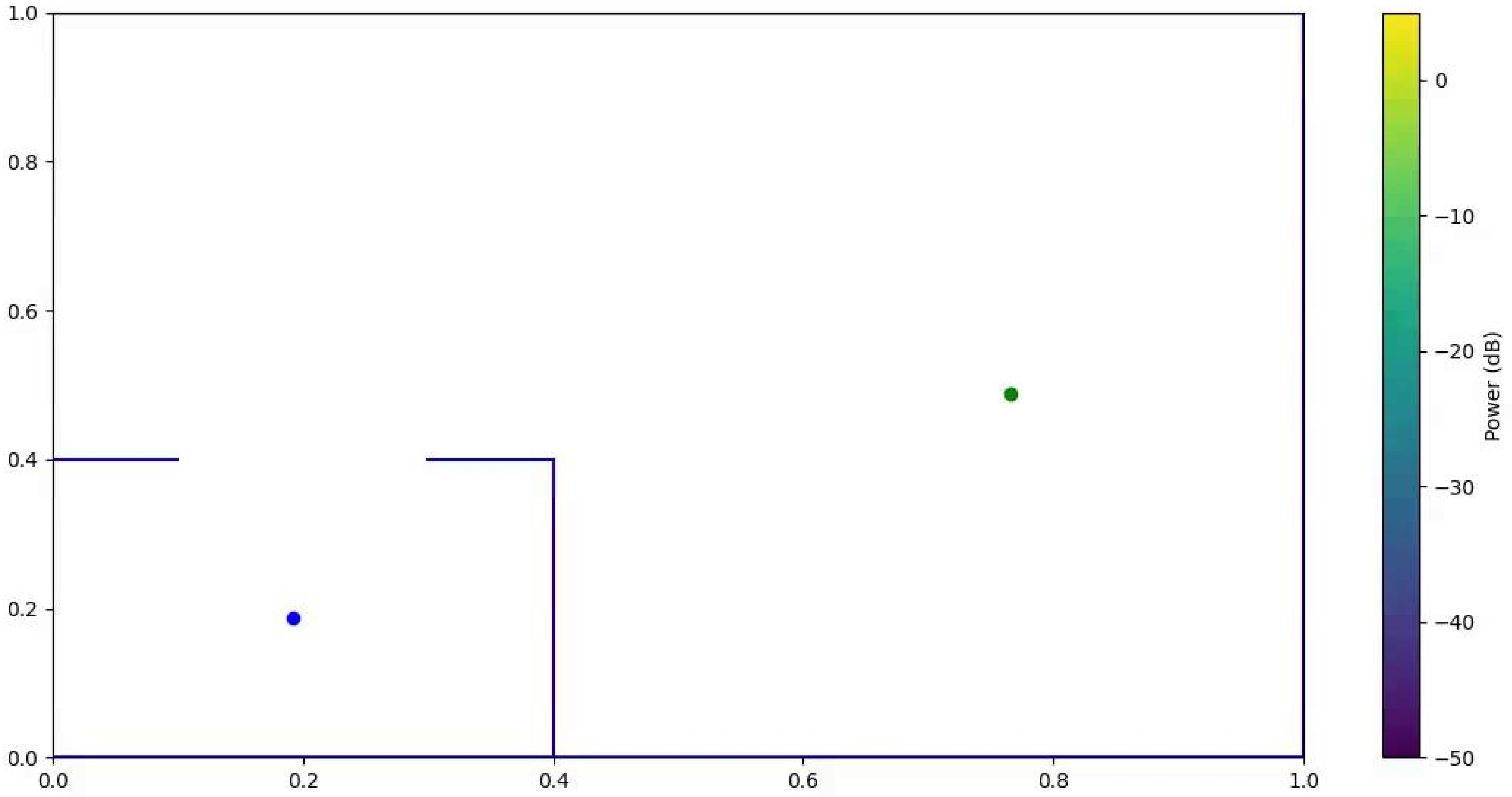


(b) After optimization

Fig. 5: Gradient-based optimization of the orientation of a transmitter (see the inset) with respect to the average received power within a small region of the scene (orange ring).

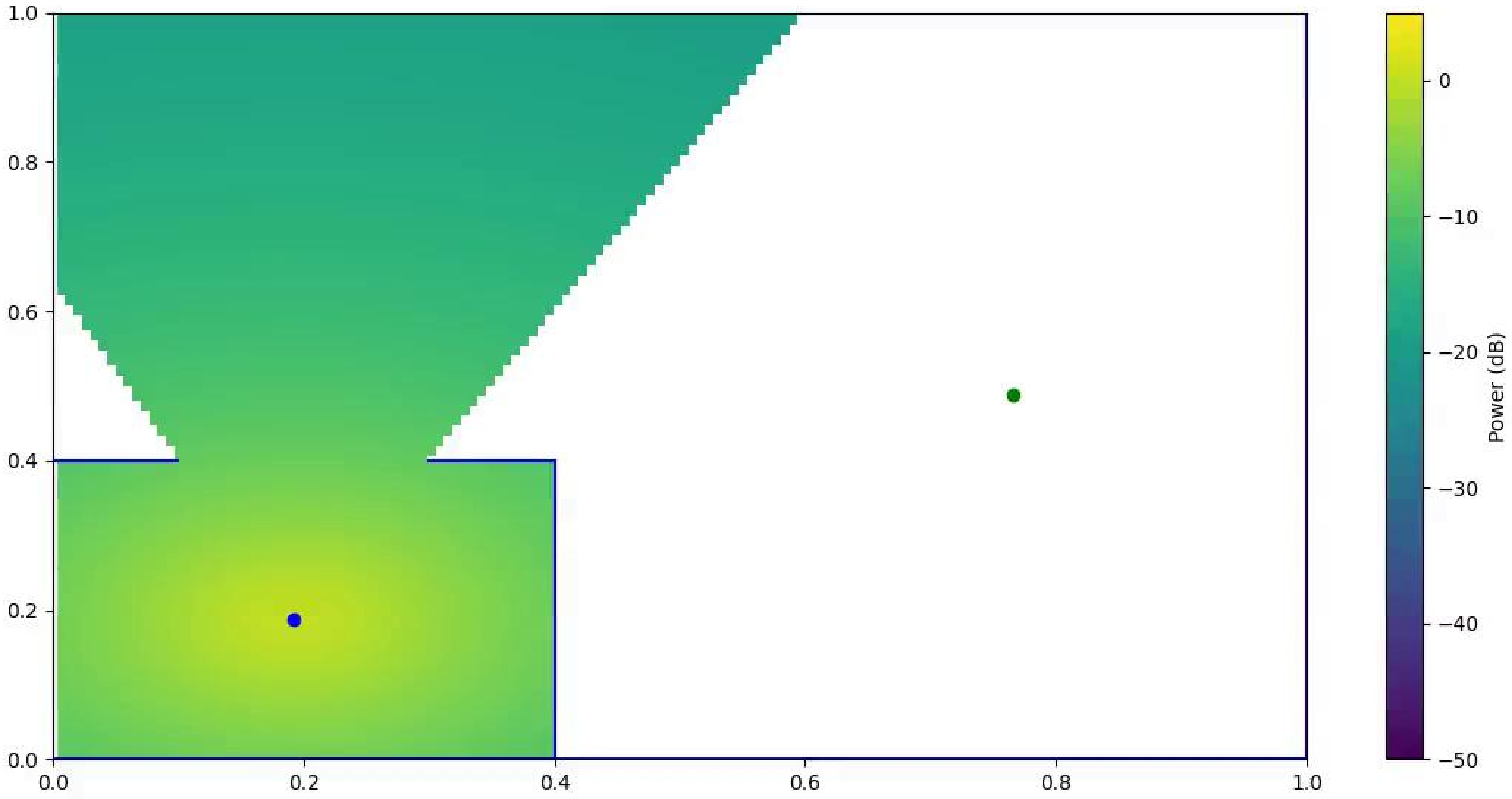
Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing



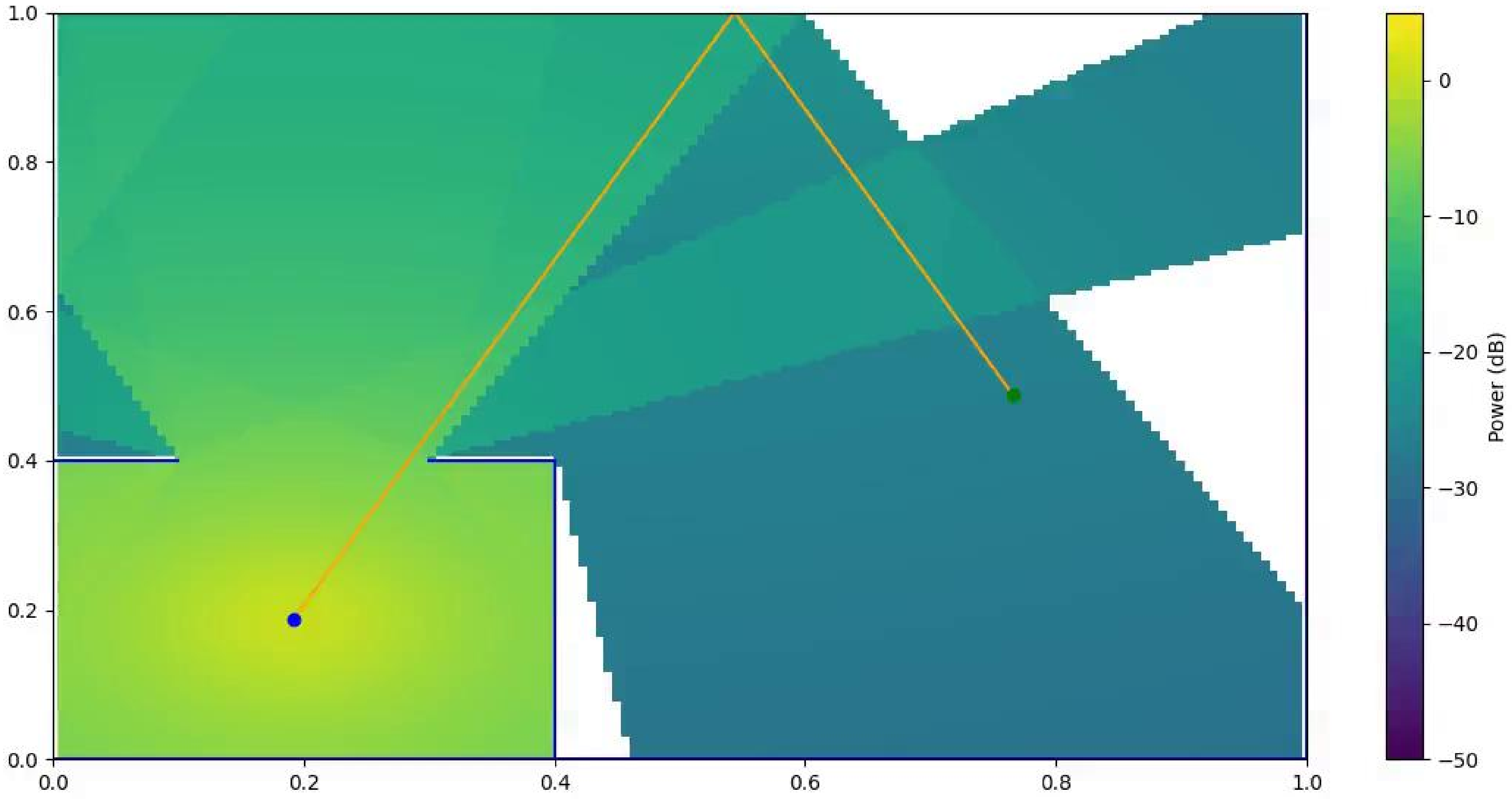
Challenge: number of paths.

Differentiable Ray Tracing



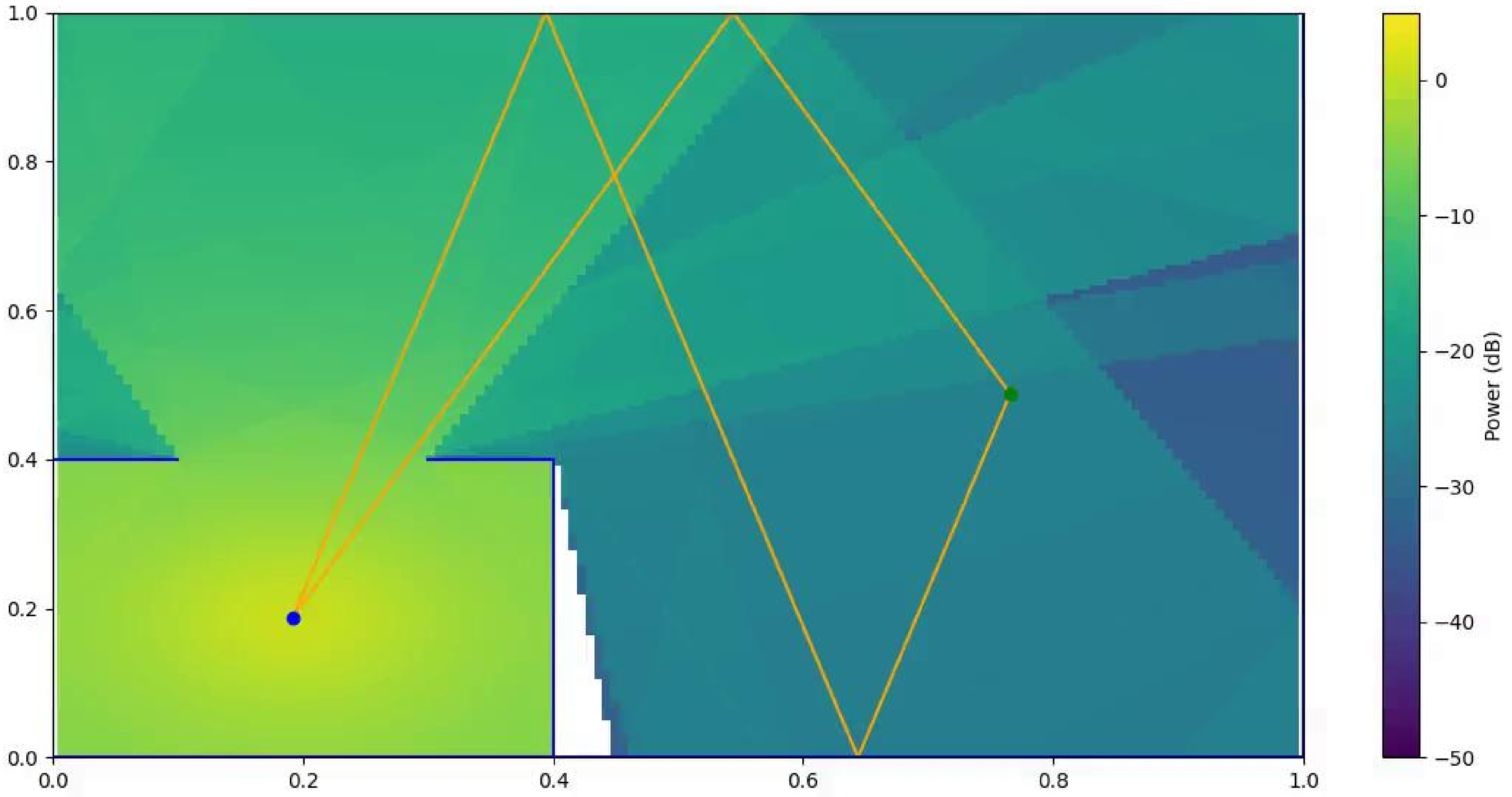
Challenge: number of paths.

Differentiable Ray Tracing



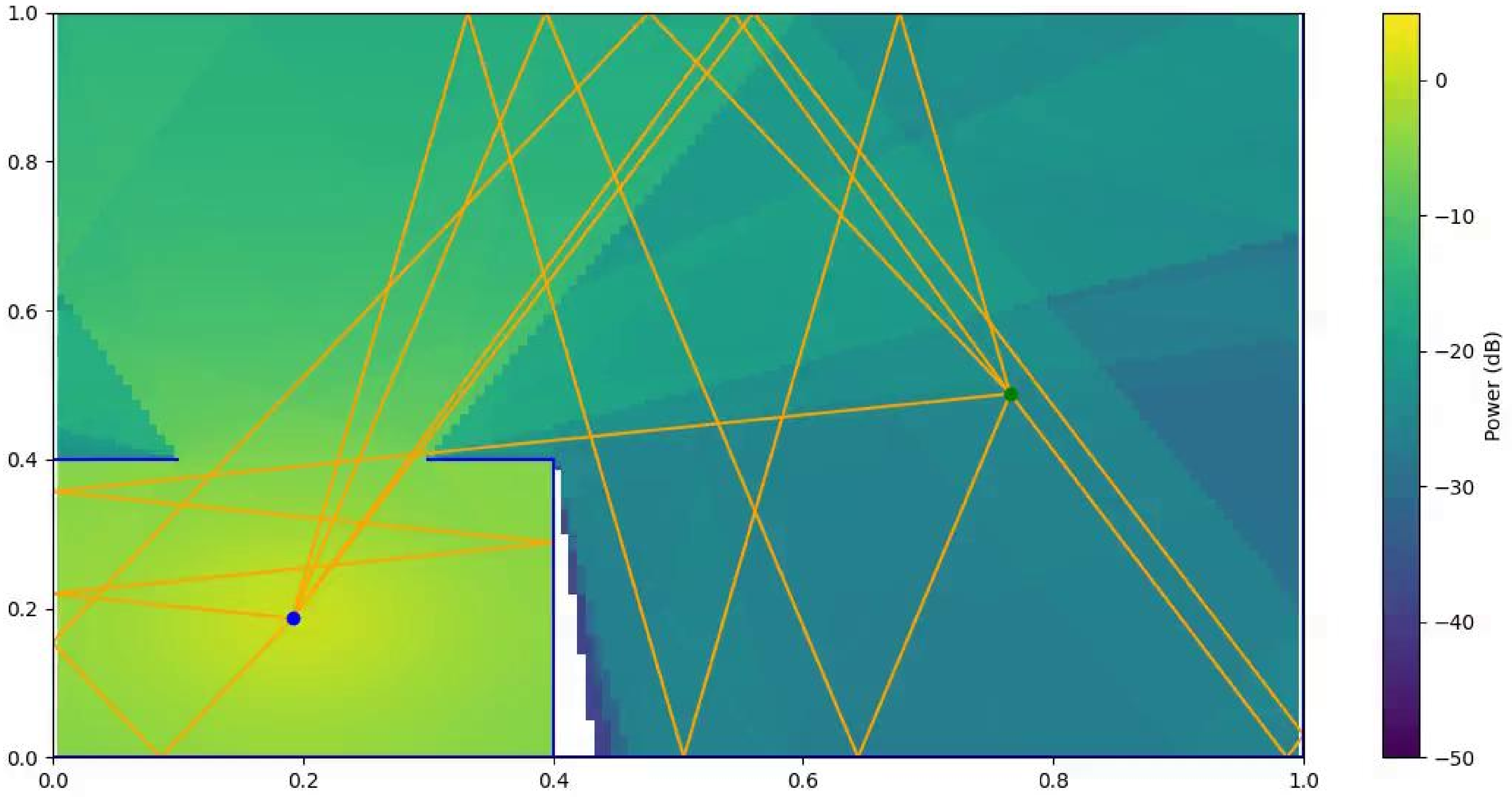
Challenge: number of paths.

Differentiable Ray Tracing



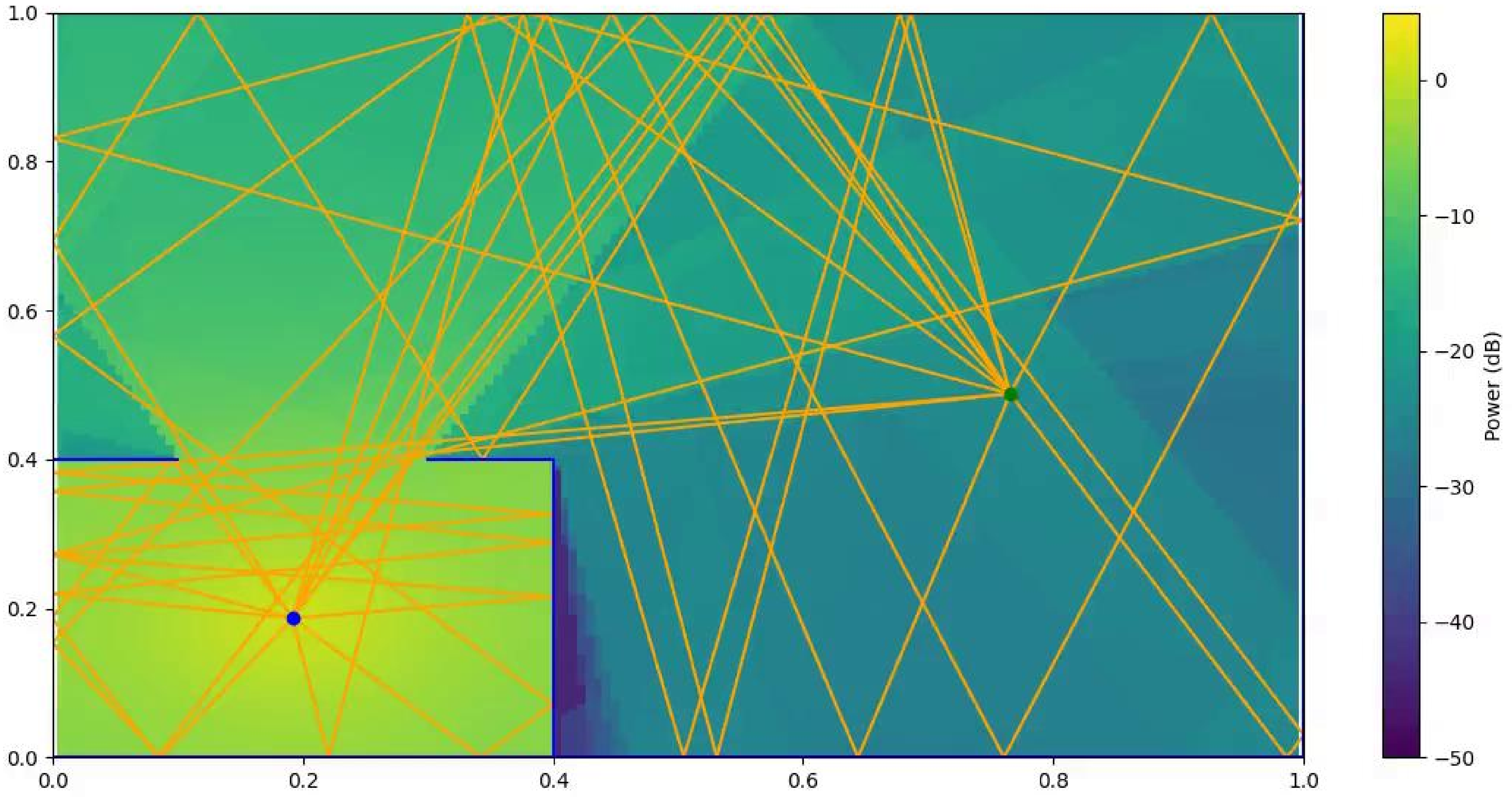
Challenge: number of paths.

Differentiable Ray Tracing



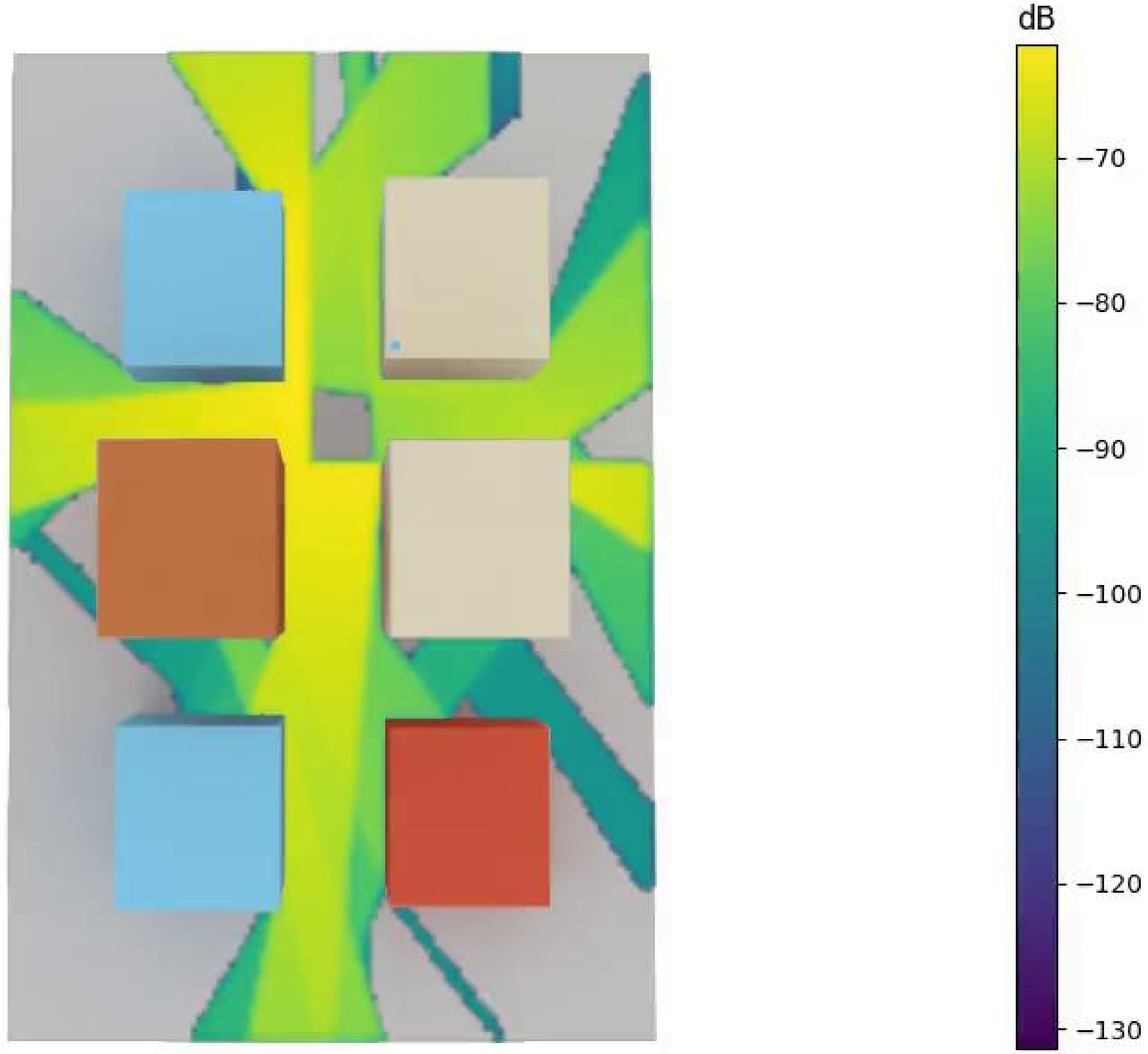
Challenge: number of paths.

Differentiable Ray Tracing



Challenge: number of paths.

Differentiable Ray Tracing

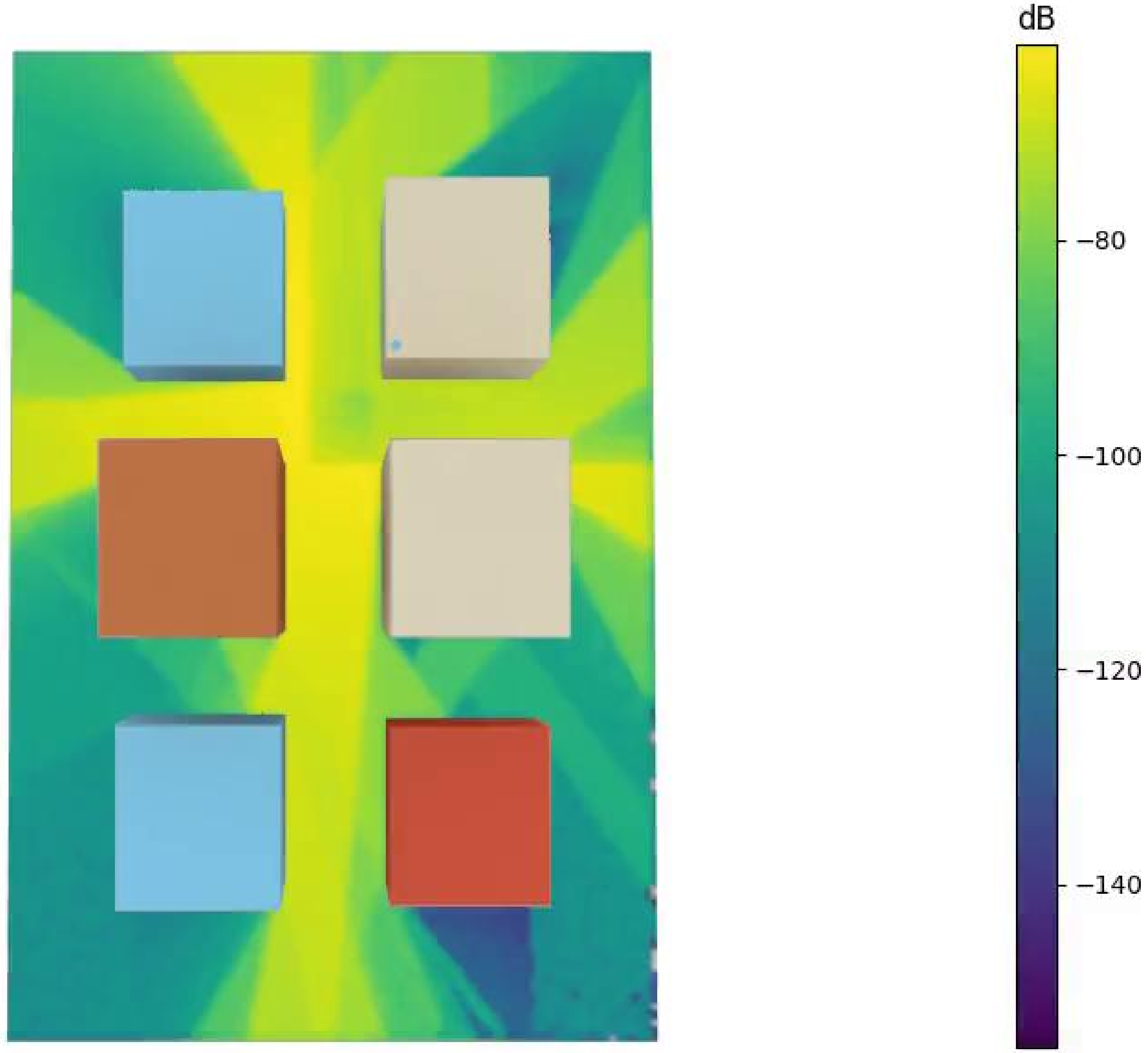


LOS + reflection

Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing

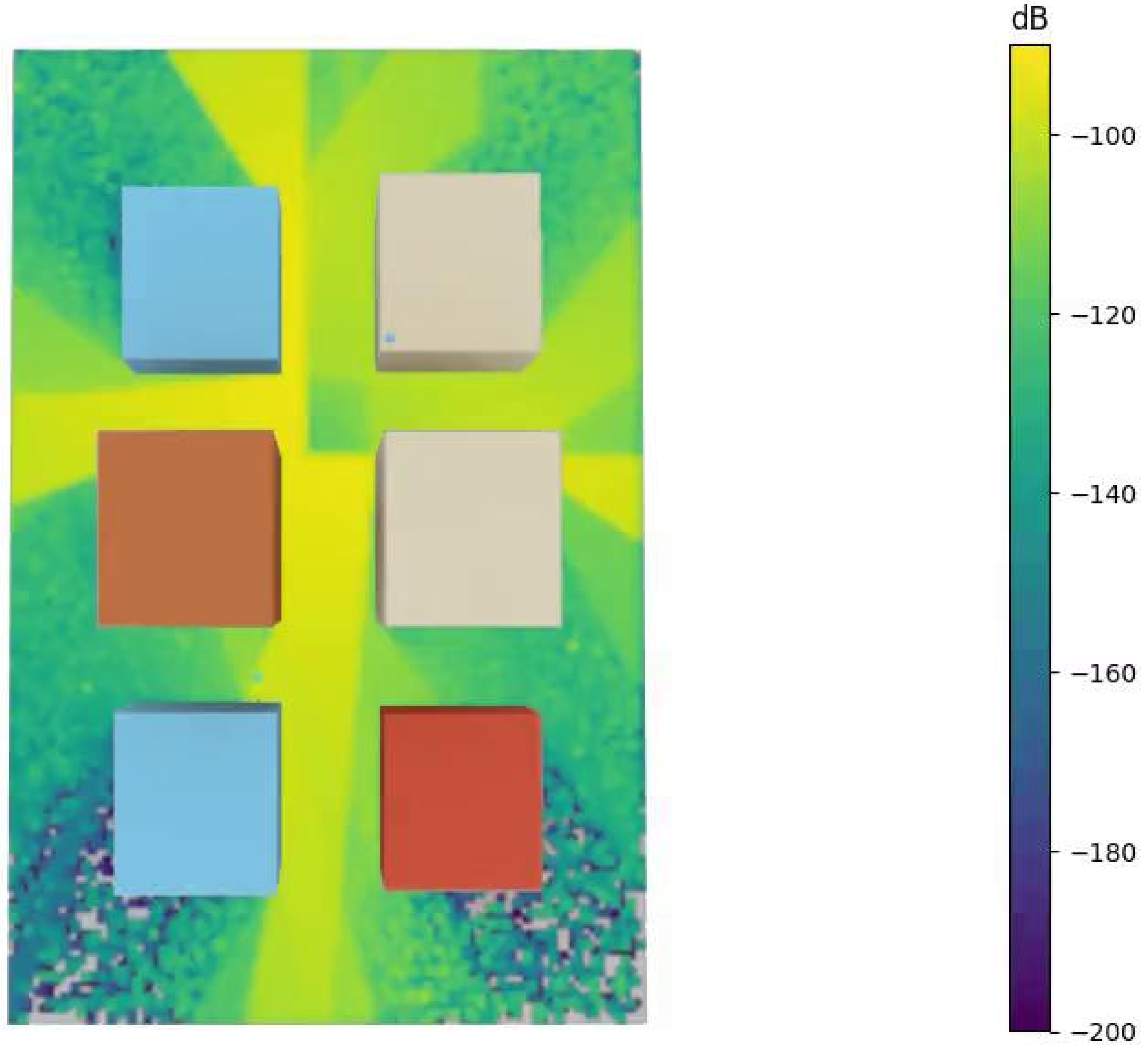


LOS + reflection + diffraction

Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing

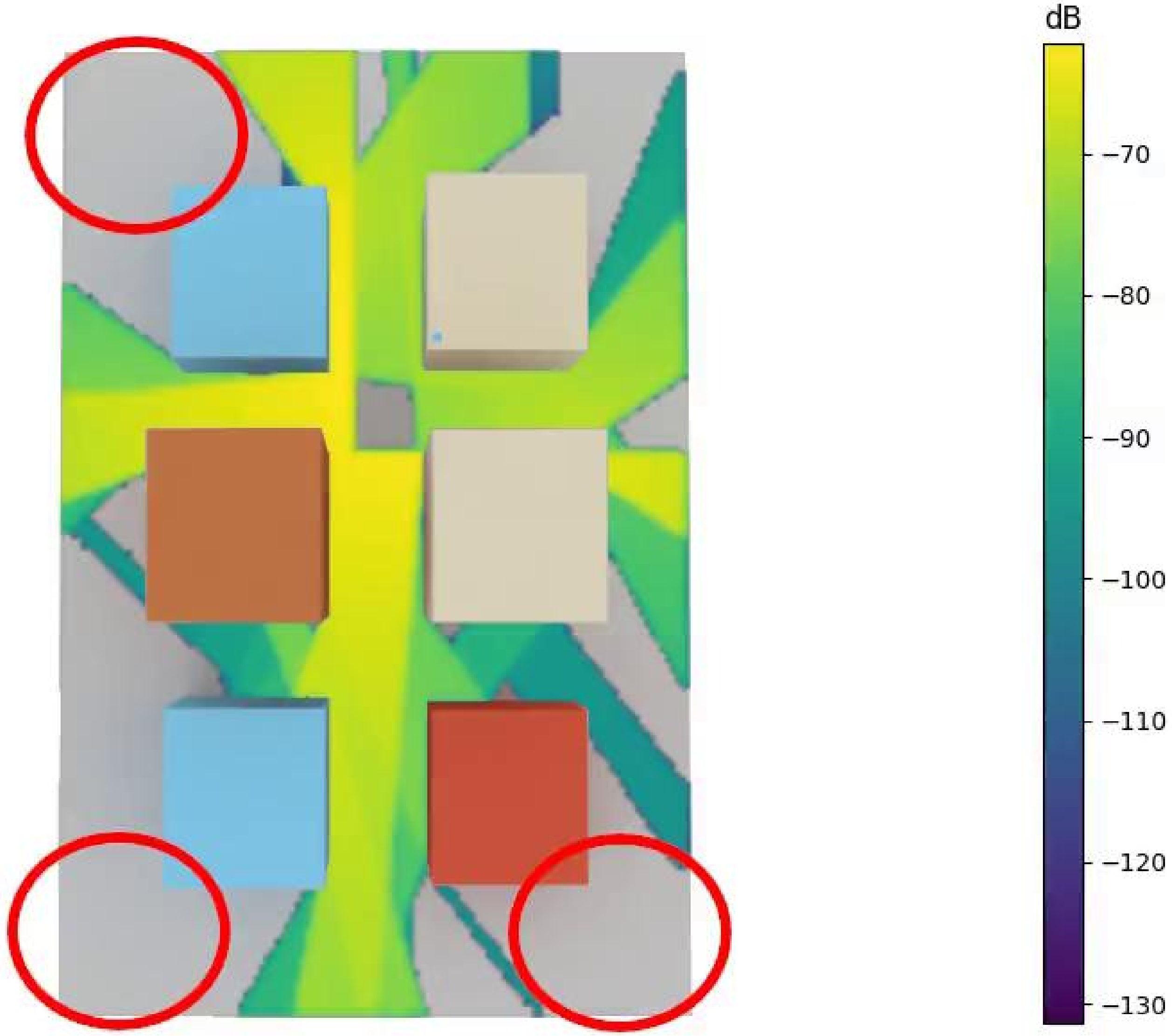


LOS + reflection + scattering

Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing



LOS + reflection

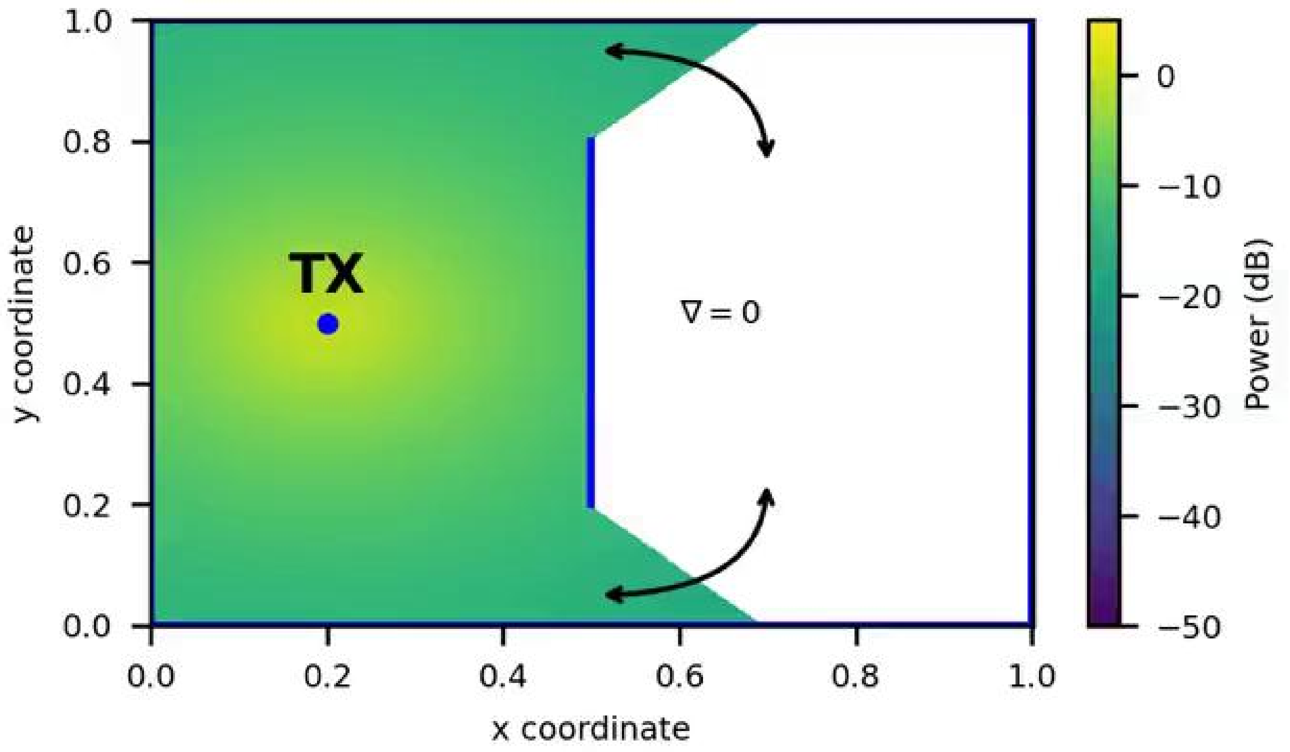
Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Present work: discontinuity smoothing

- Zero-gradient and discontinuity issues;
- Smoothing technique;
- Optimization example.

Present work: discontinuity smoothing



$$\theta(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise,} \end{cases}$$

Present work: discontinuity smoothing

$$\lim_{\alpha \to \infty} s(x;\alpha) = \theta(x)$$

[C1] $\lim_{x\to-\infty} s(x;\alpha) = 0$ and $\lim_{x\to+\infty} s(x;\alpha) = 1$;

[C2] $s(\cdot;\alpha)$ is monotonically increasing;

[C3] $s(0;\alpha) = \frac{1}{2}$;

[C4] and $s(x;\alpha) - s(0;\alpha) = s(0;\alpha) - s(-x;\alpha)$.

Present work: discontinuity smoothing

$$s(x;\alpha) = s(\alpha x). \tag{1}$$

The sigmoid is defined with a real-valued exponential

$$\text{sigmoid}(x;\alpha) = \frac{1}{1+e^{-\alpha x}}, \tag{2}$$

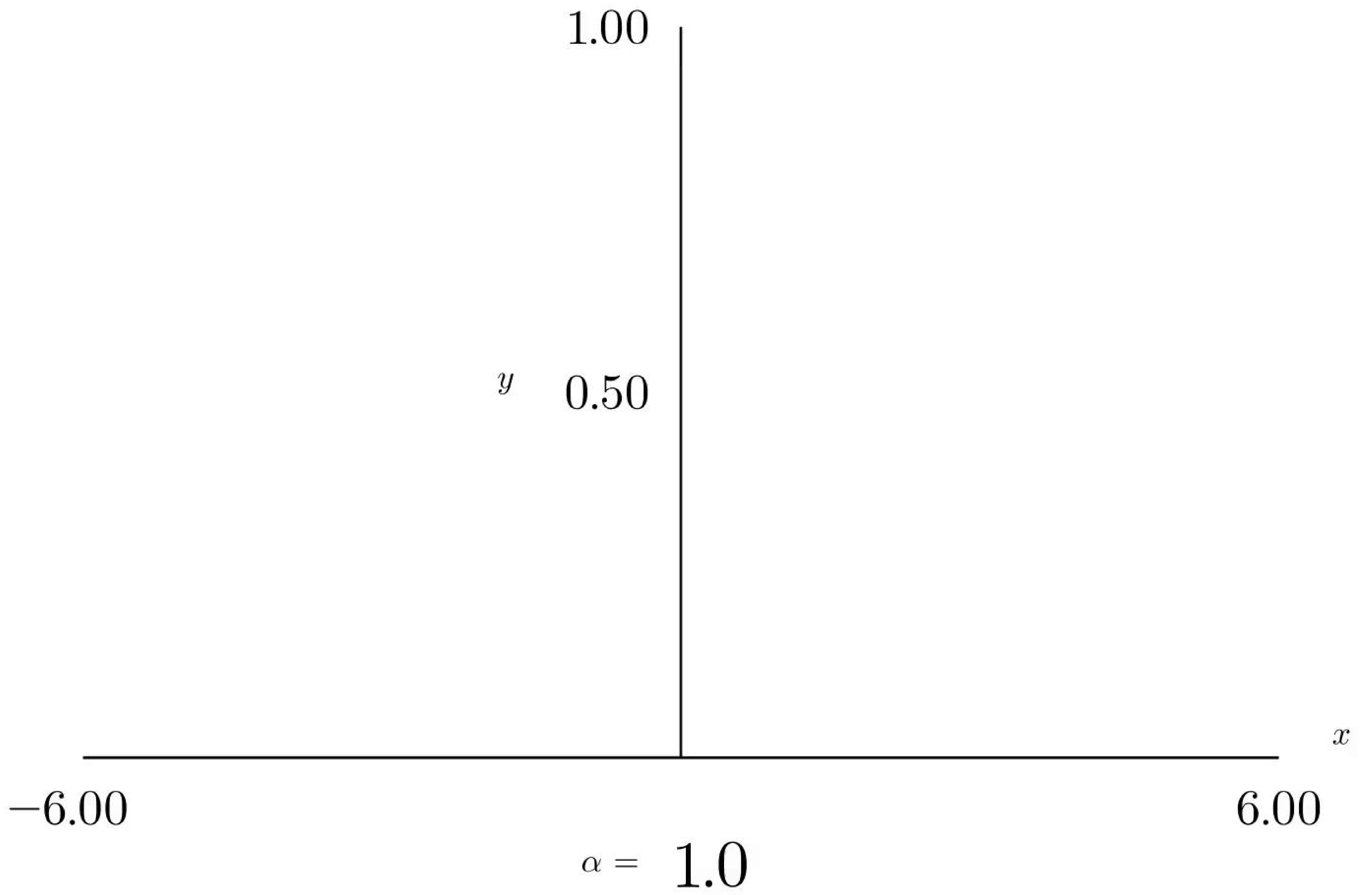
and the hard sigmoid is the piecewise linear function defined by

$$\text{hard sigmoid}(x;\alpha) = \frac{\text{relu6}(\alpha x + 3)}{6}, \tag{3}$$

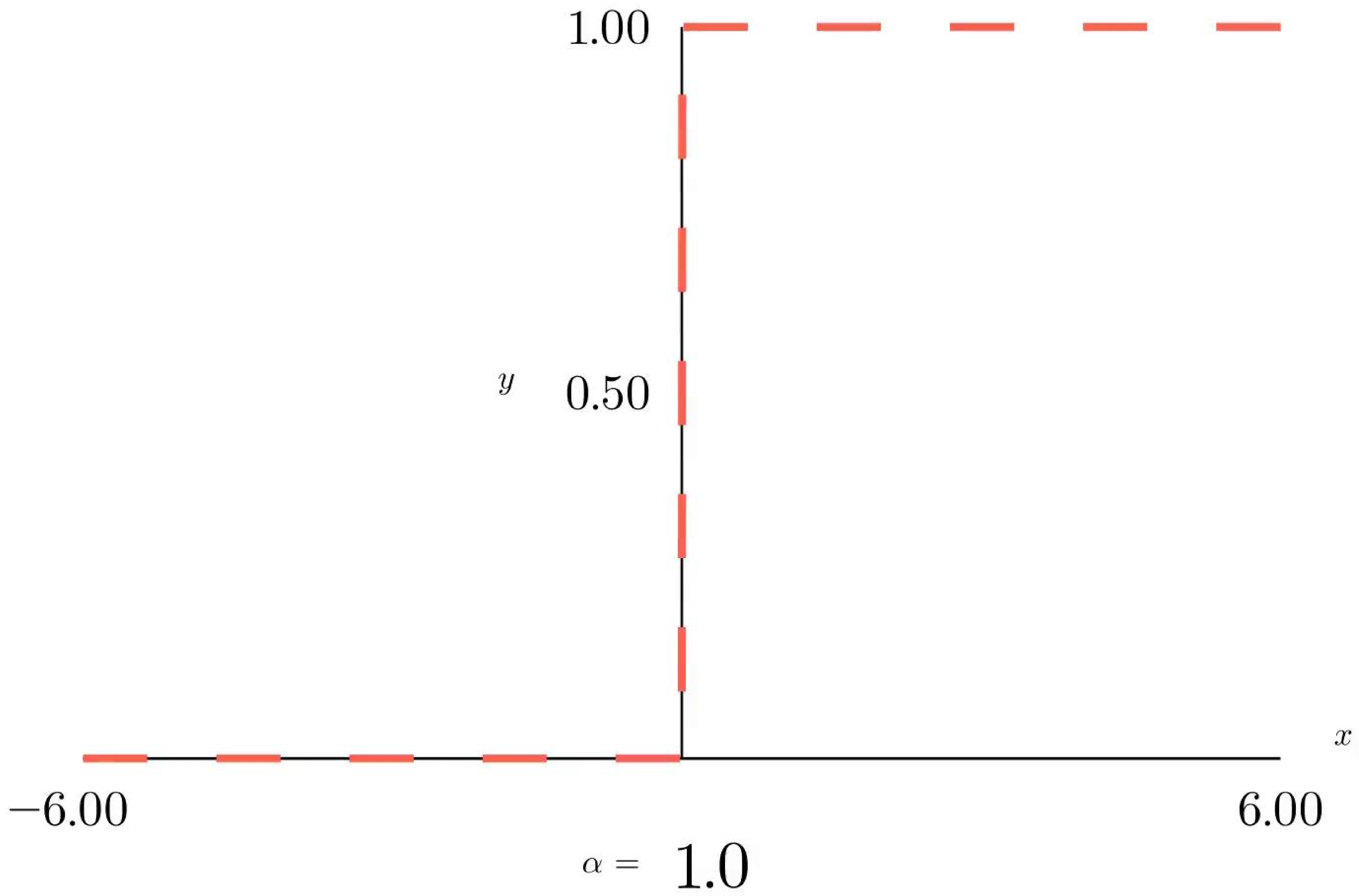
where

$$\text{relu6}(x) = \min(\max(0,x),6). \tag{4}$$

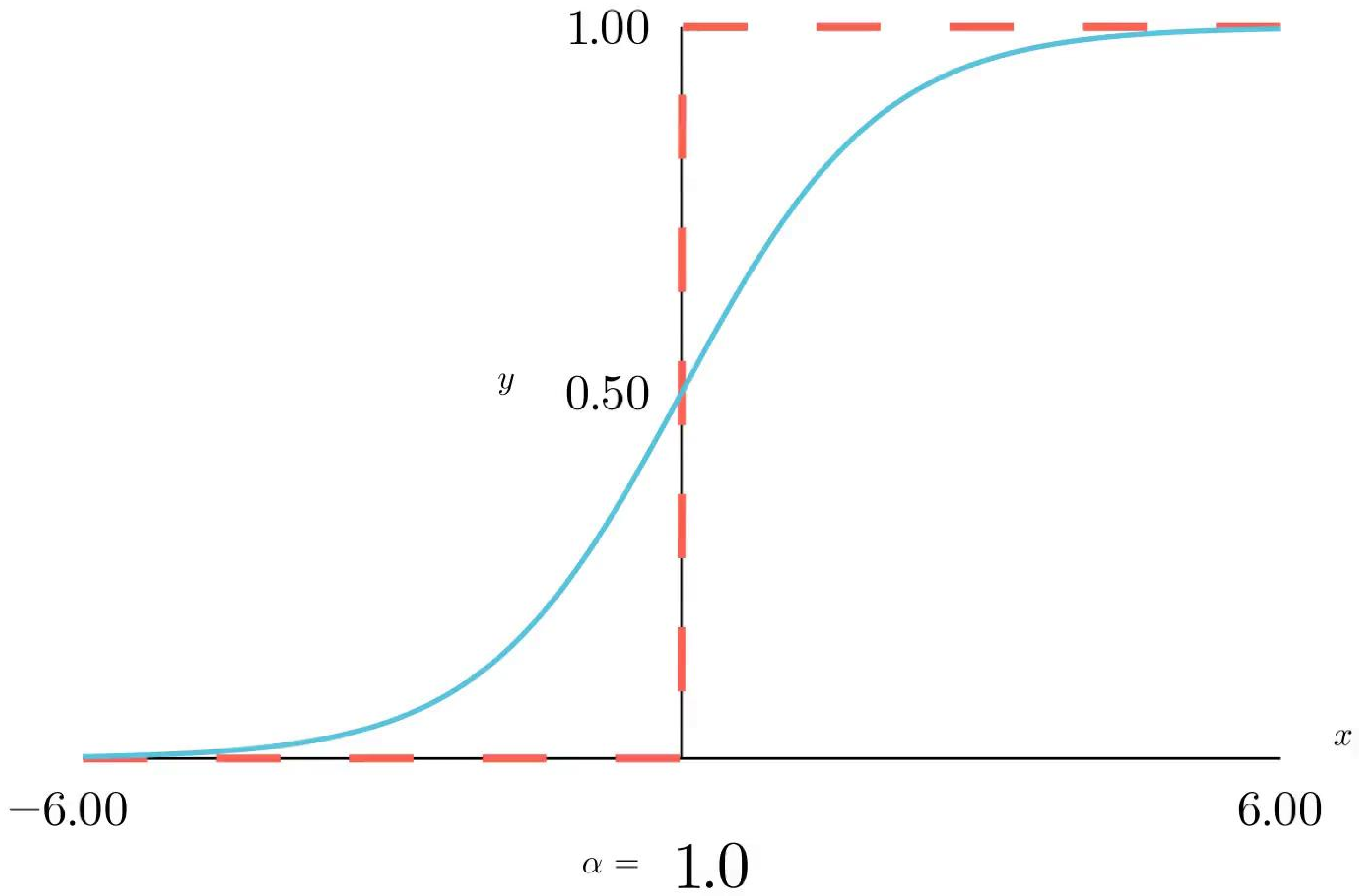
Present work: discontinuity smoothing



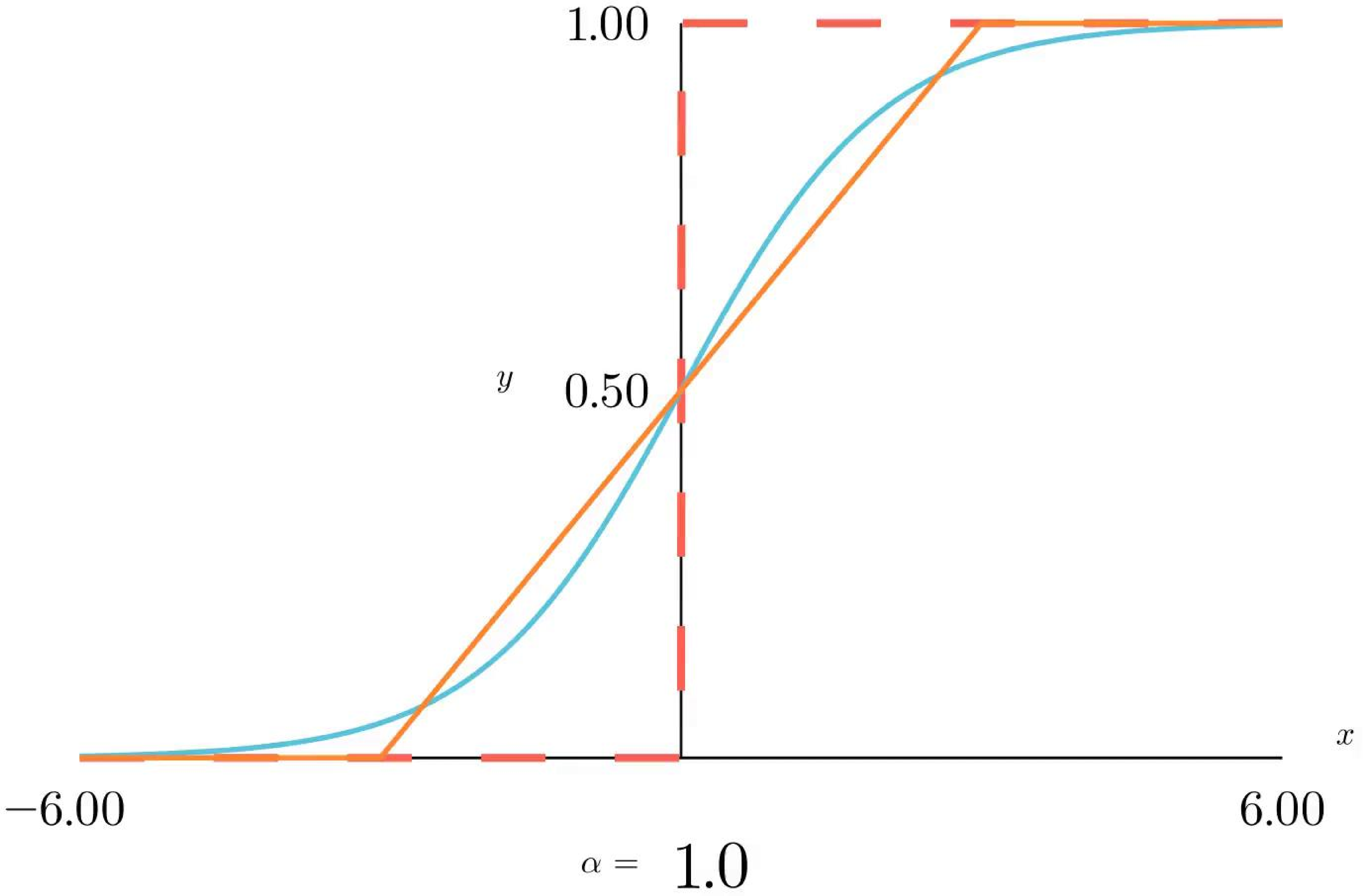
Present work: discontinuity smoothing



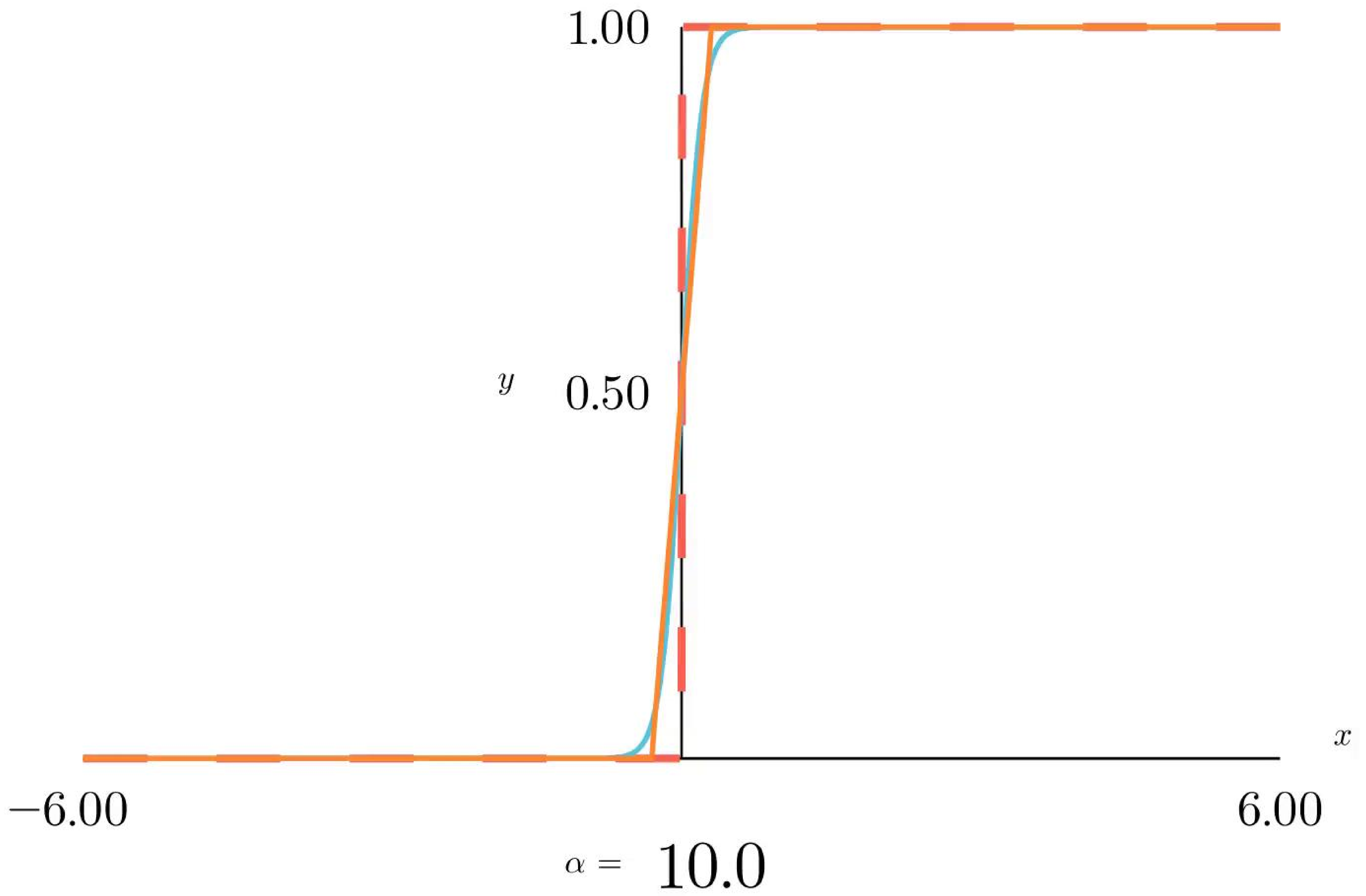
Present work: discontinuity smoothing



Present work: discontinuity smoothing



Present work: discontinuity smoothing

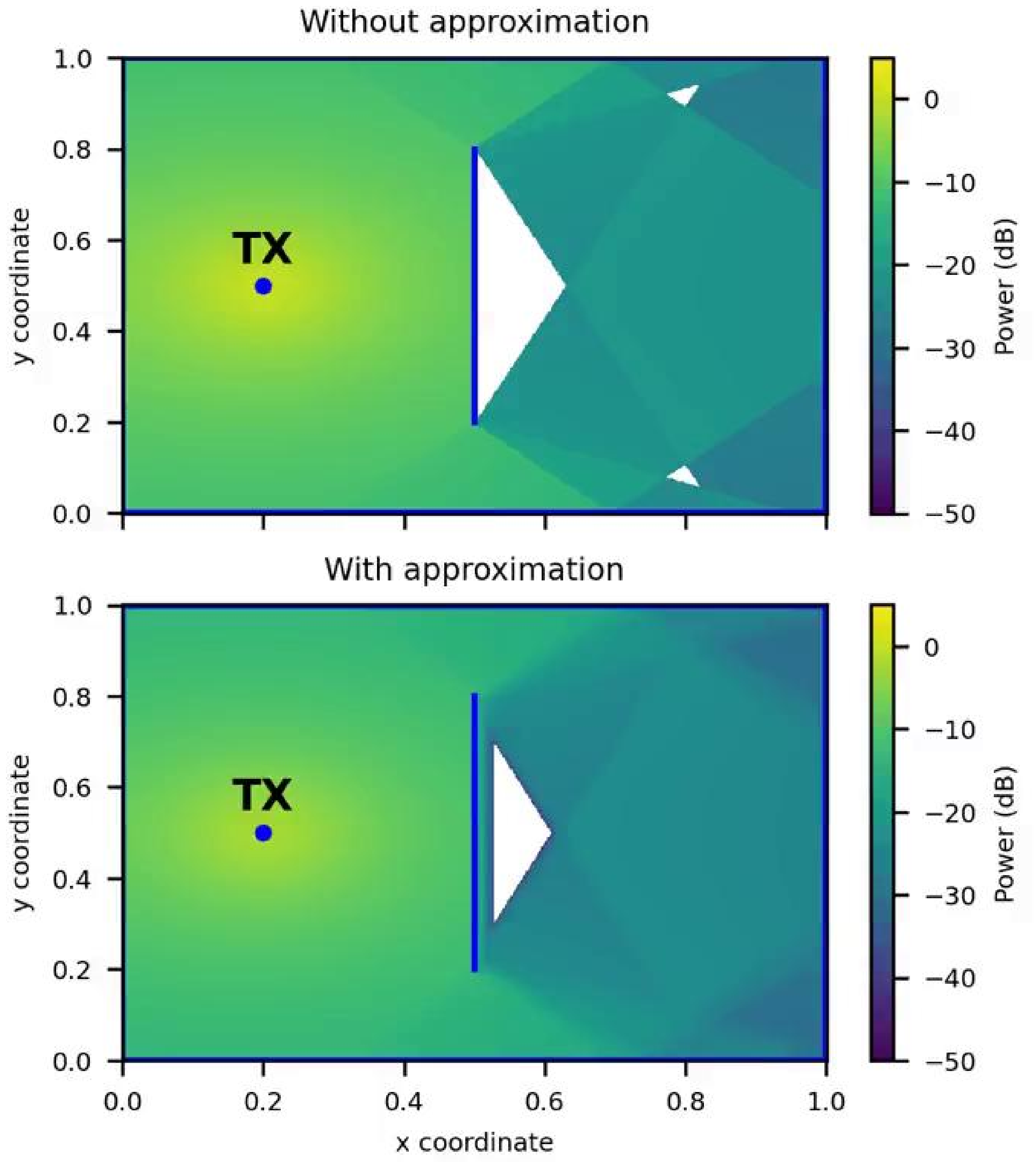


Present work: discontinuity smoothing

$$\vec{E}(x,y) = \sum_{\mathcal{P}\in\mathcal{S}} V(\mathcal{P})\left(\bar{C}(\mathcal{P}) \cdot \vec{E}(\mathcal{P}_1) \right)$$

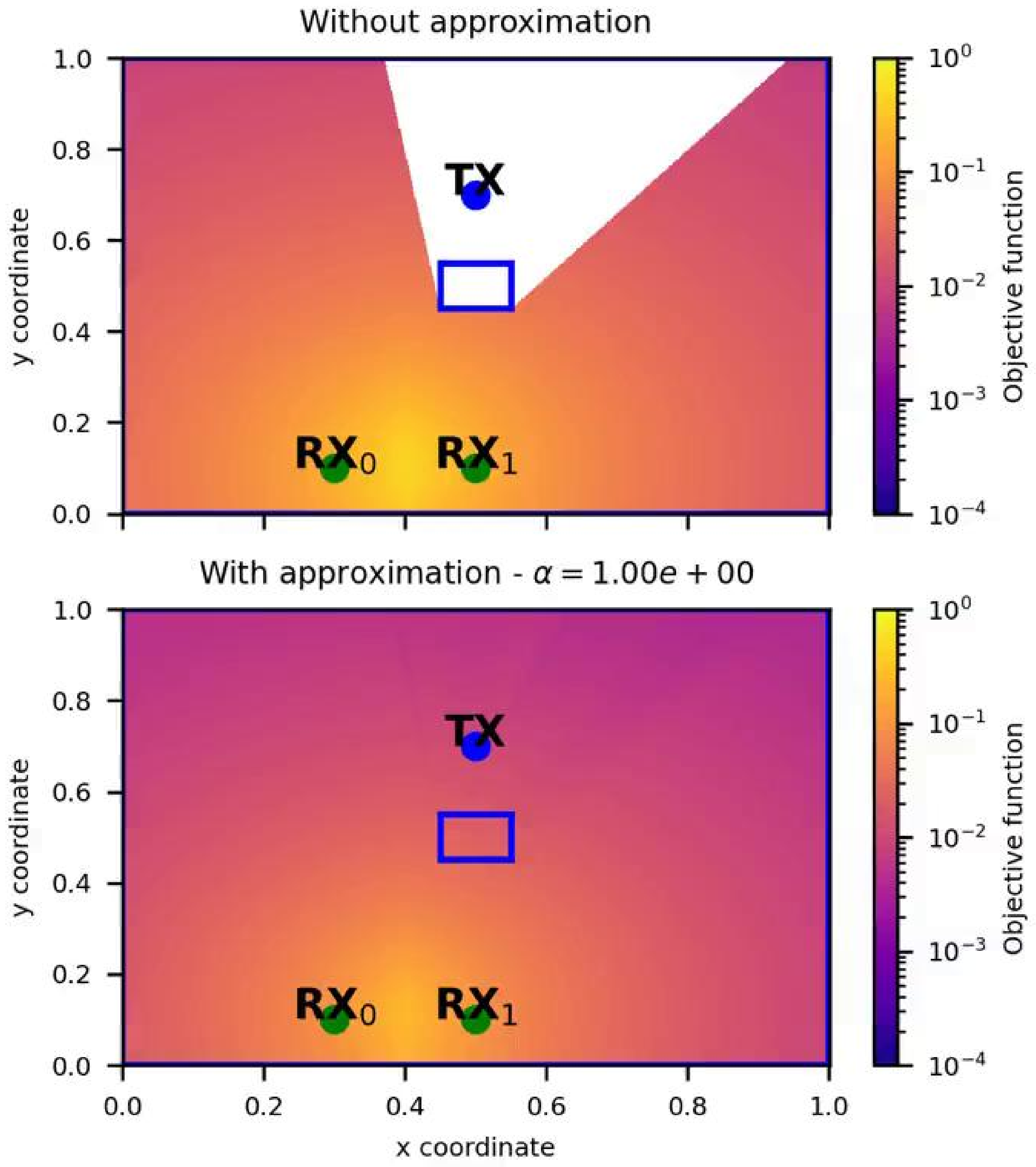
$$P(x,y) \approx \sum_{\mathcal{P}\in\mathcal{S}} P_{\mathcal{P}}(x,y)$$

(incoherently added)

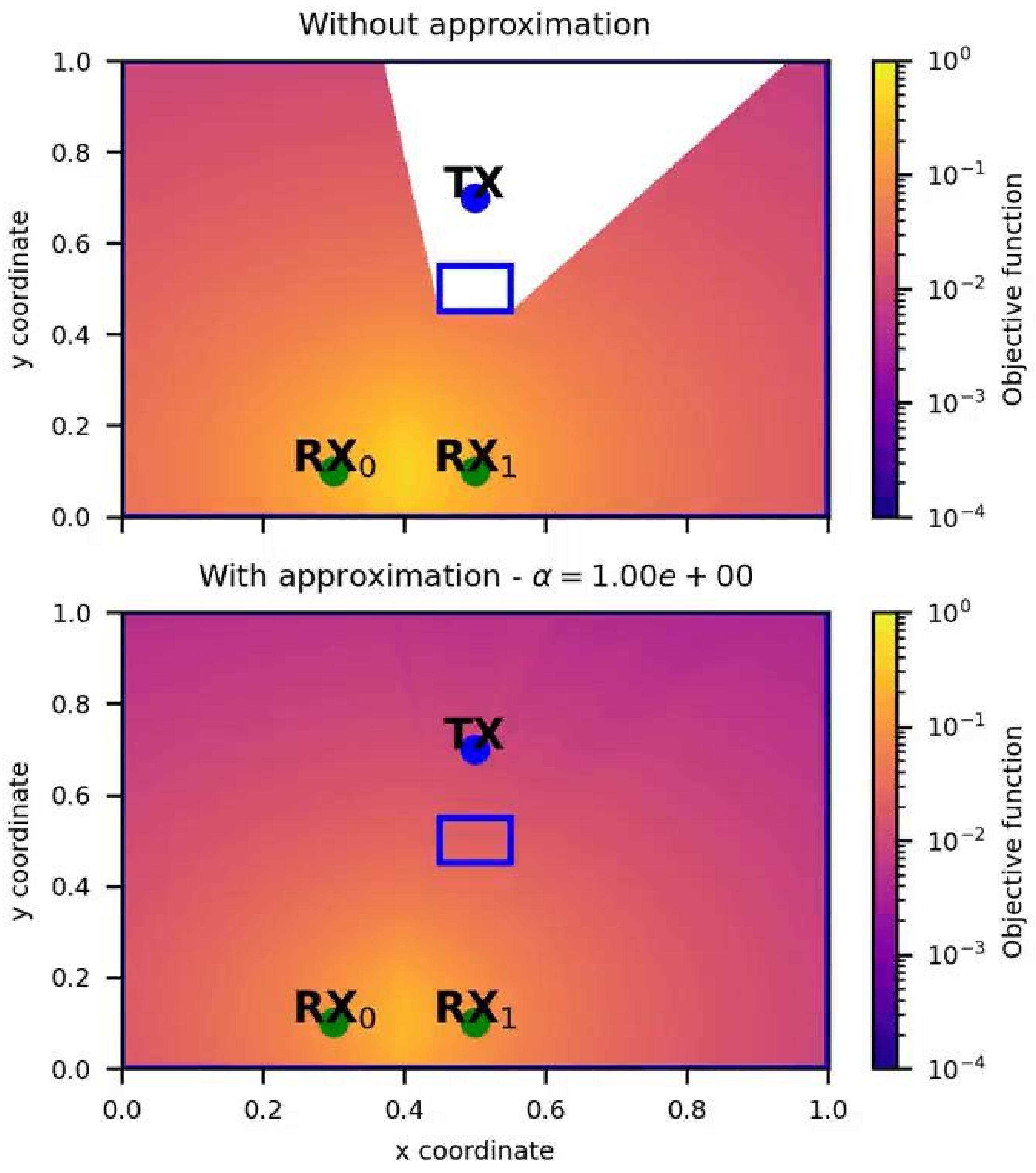


Present work: discontinuity smoothing

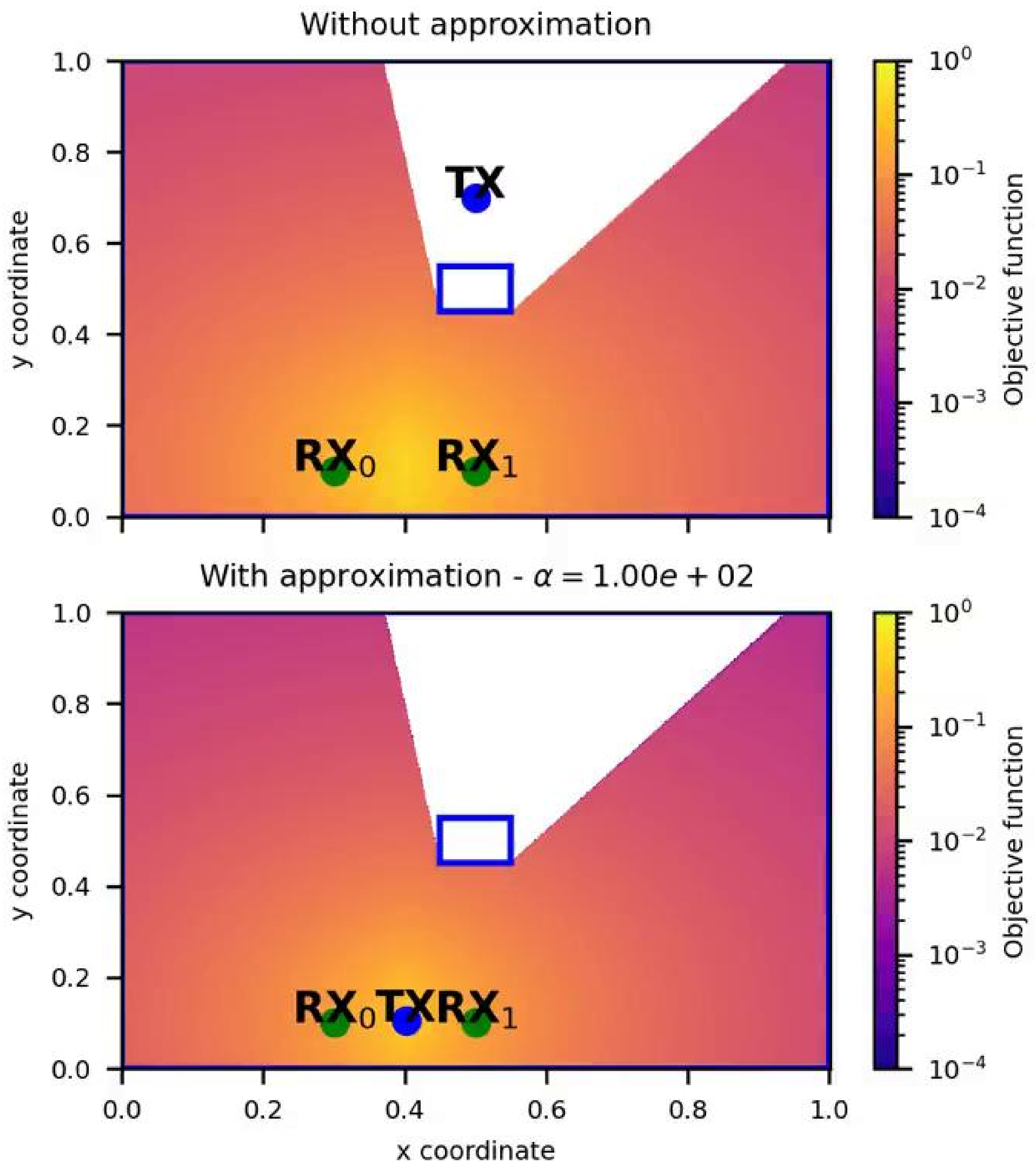
$$\mathcal{F}(x, y) = \min\left(P_{\mathrm{RX}_0}(x, y), P_{\mathrm{RX}_1}(x, y)\right)$$



Present work: discontinuity smoothing

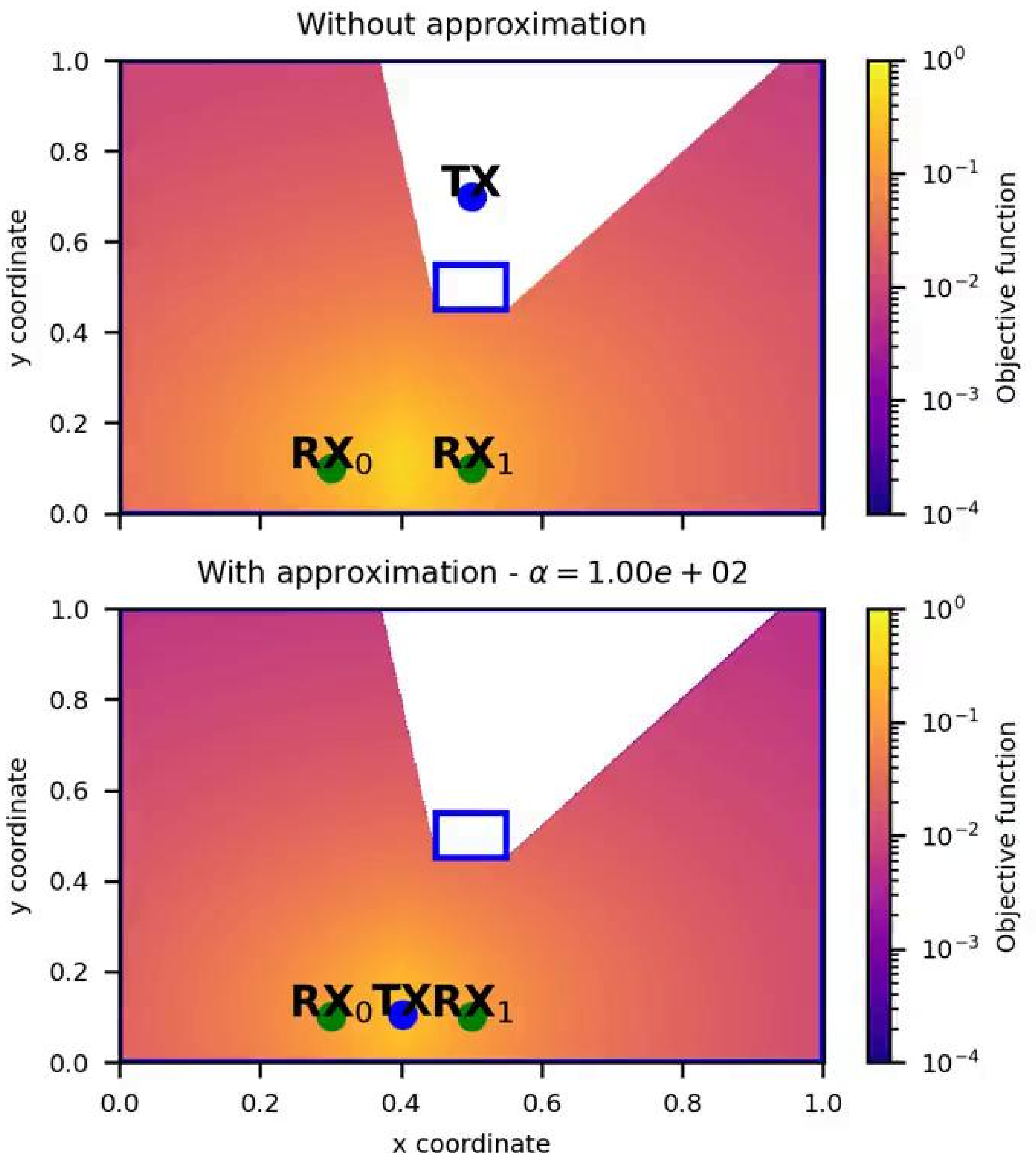


Present work: discontinuity smoothing



Present work: discontinuity smoothing

- Convergence success rate x 1.5 ~ 2;
- Success rate w/ respect to no approx.: 92% to 98%.



Future

- Trade-off of smoothing vs many minimizations;
- Where to apply smoothing;
- Physical model behind smoothing (e.g., diffraction);
- 3D scenes at city-scales (DiffeRT).

jeertmans/DiffeRT2d

jeertmans/DiffeRT

Thanks for listening!