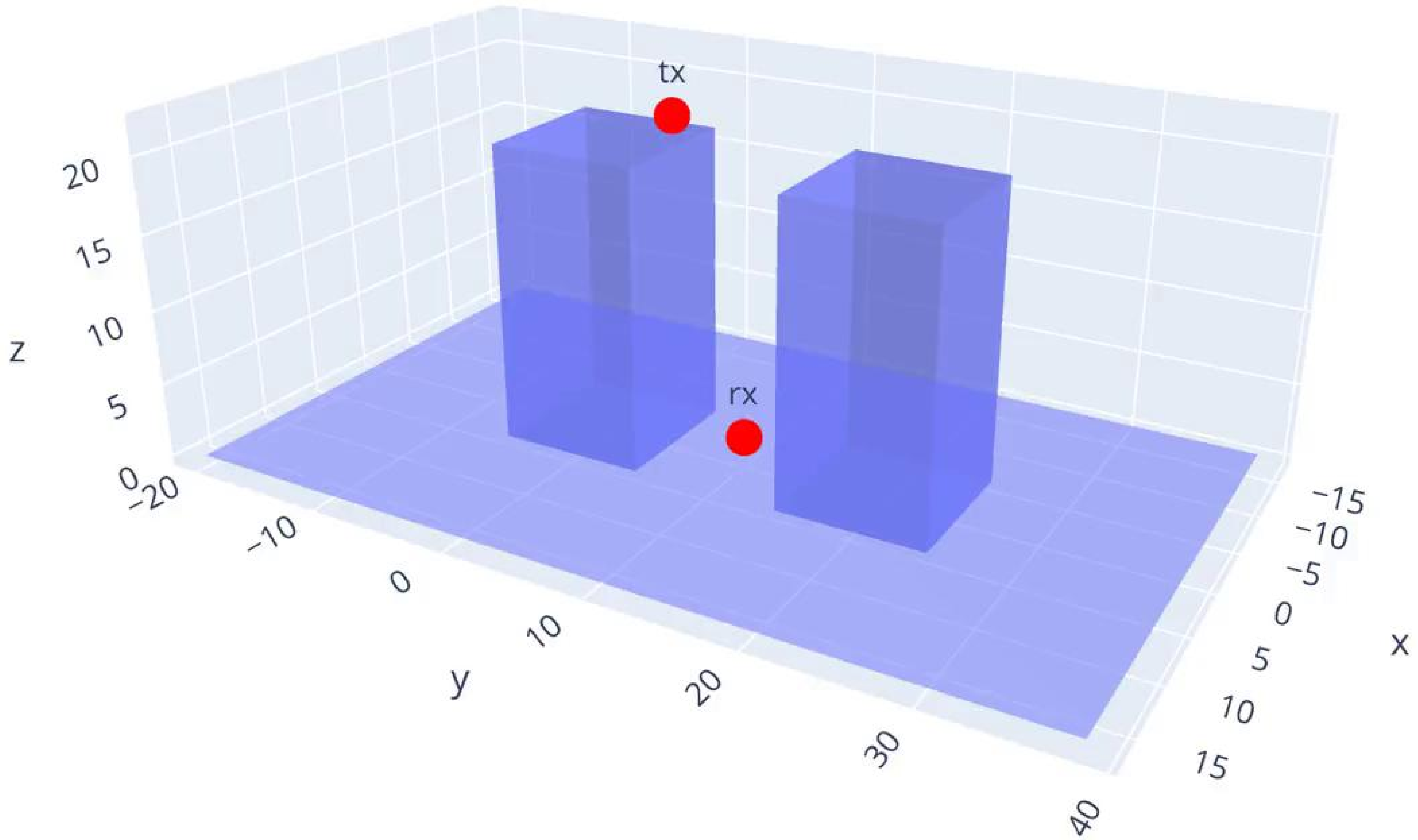
Fully Differentiable Ray Tracing via Discontinuity Smoothing for Radio Network Optimization

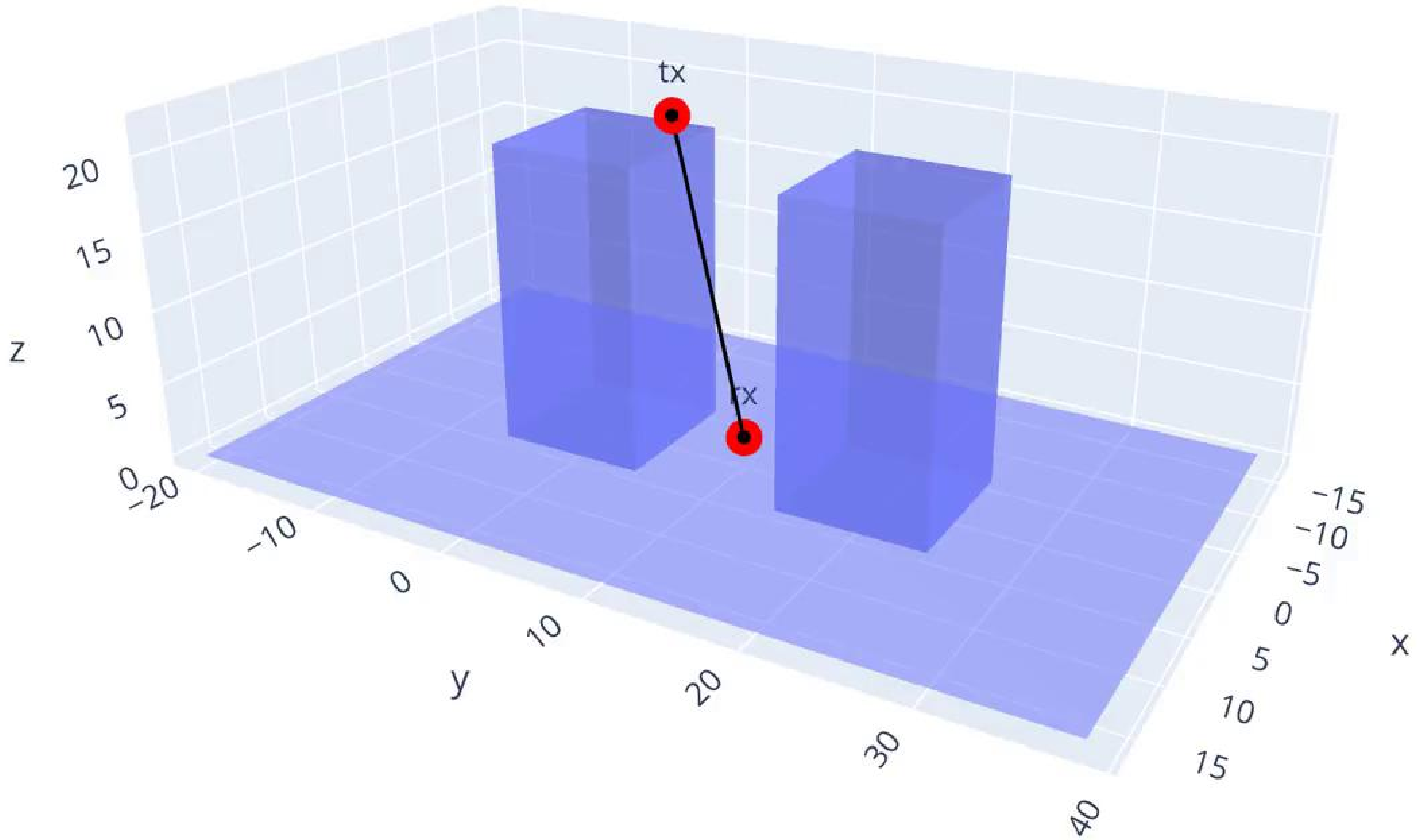
Jérome Eertmans - March 18th 2024

Differentiable Ray Tracing

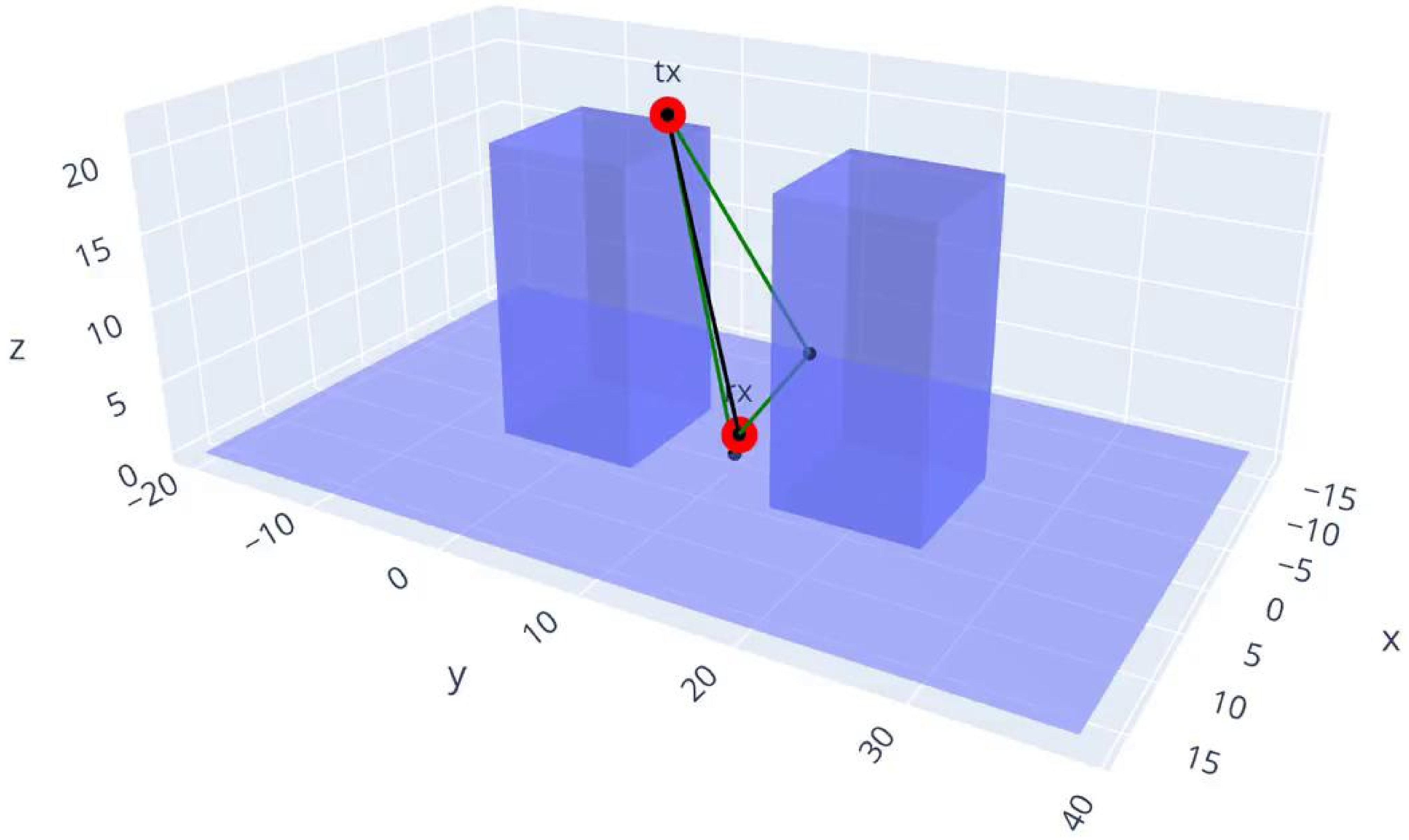
Differentiable Ray Tracing



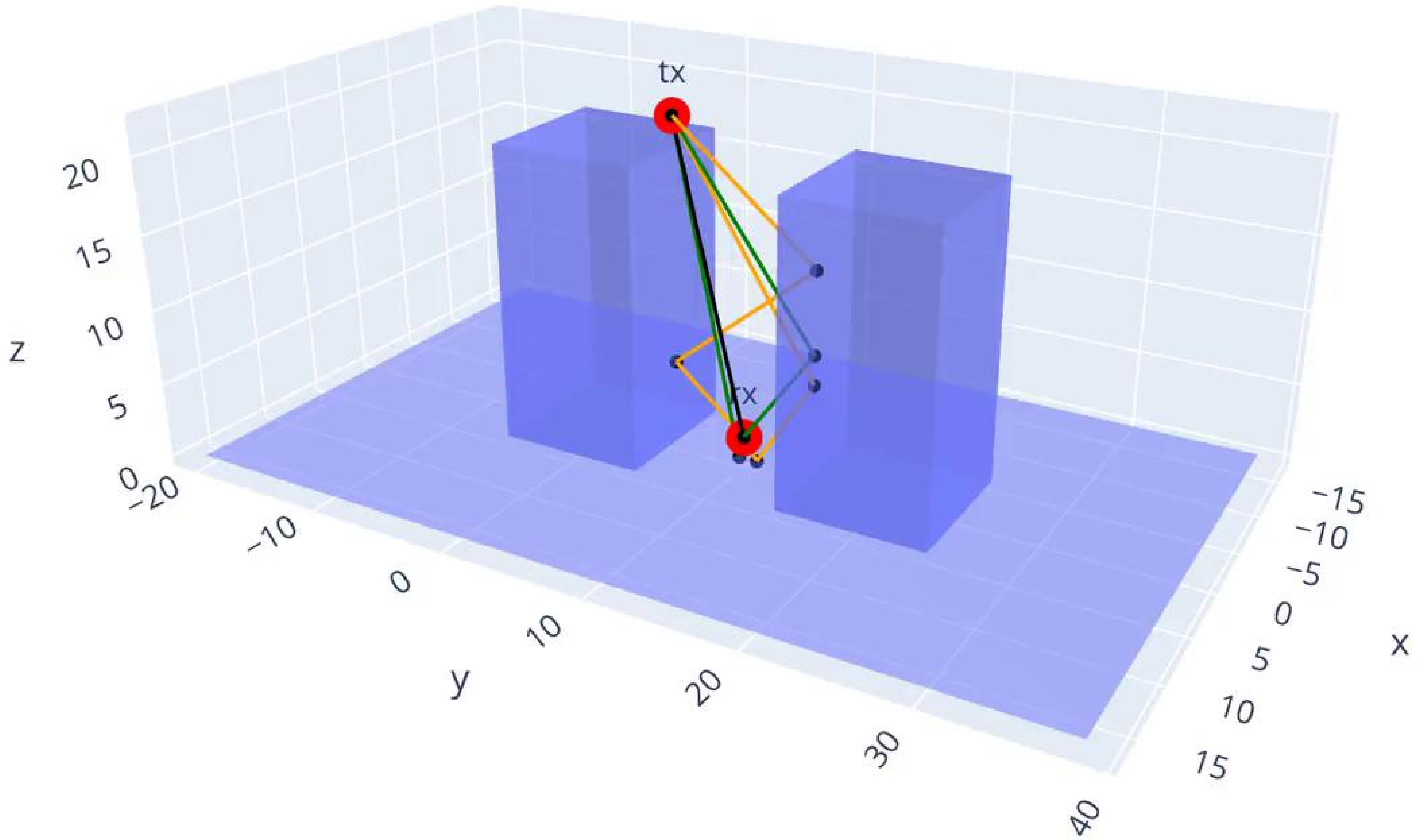
Differentiable Ray Tracing



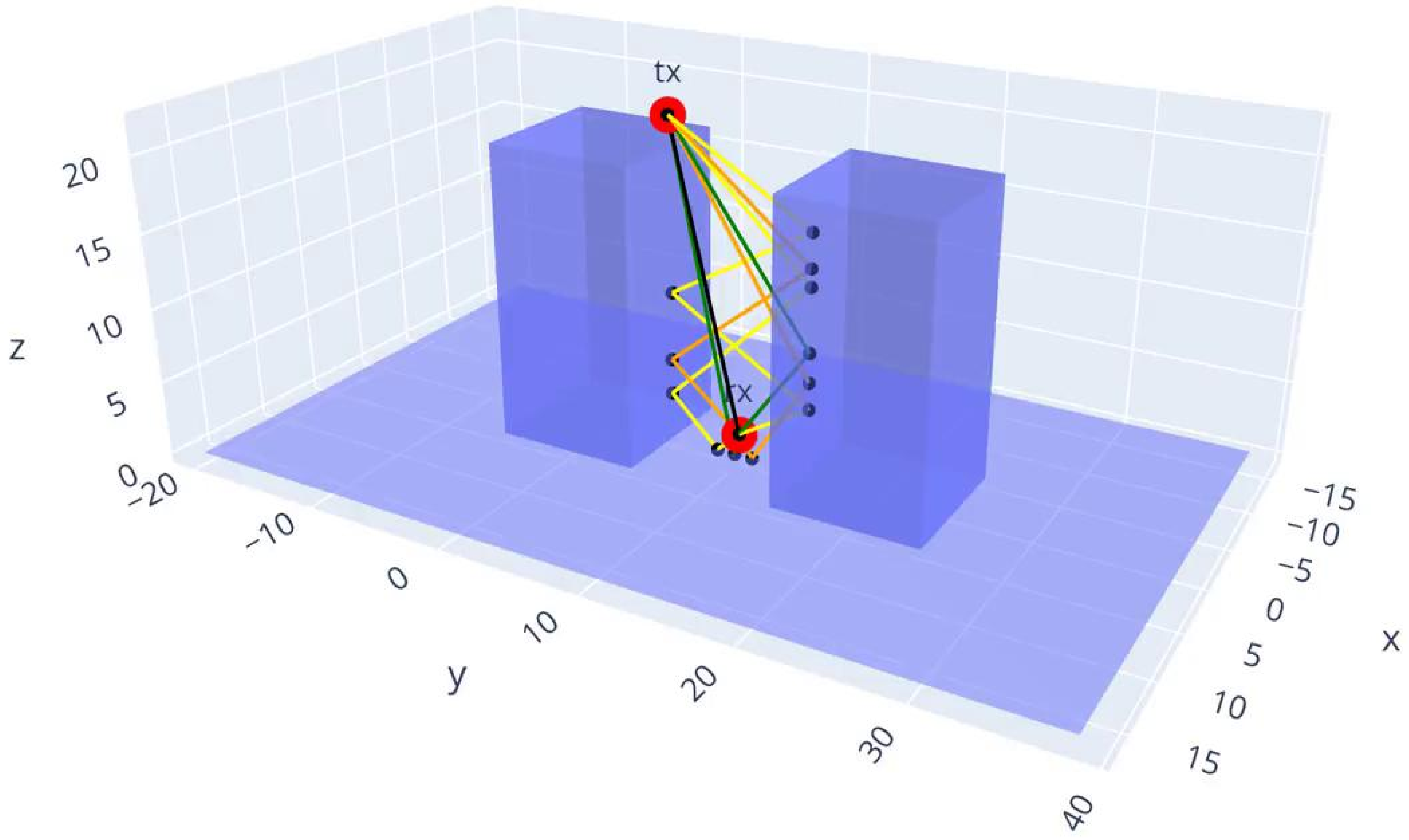
Differentiable Ray Tracing



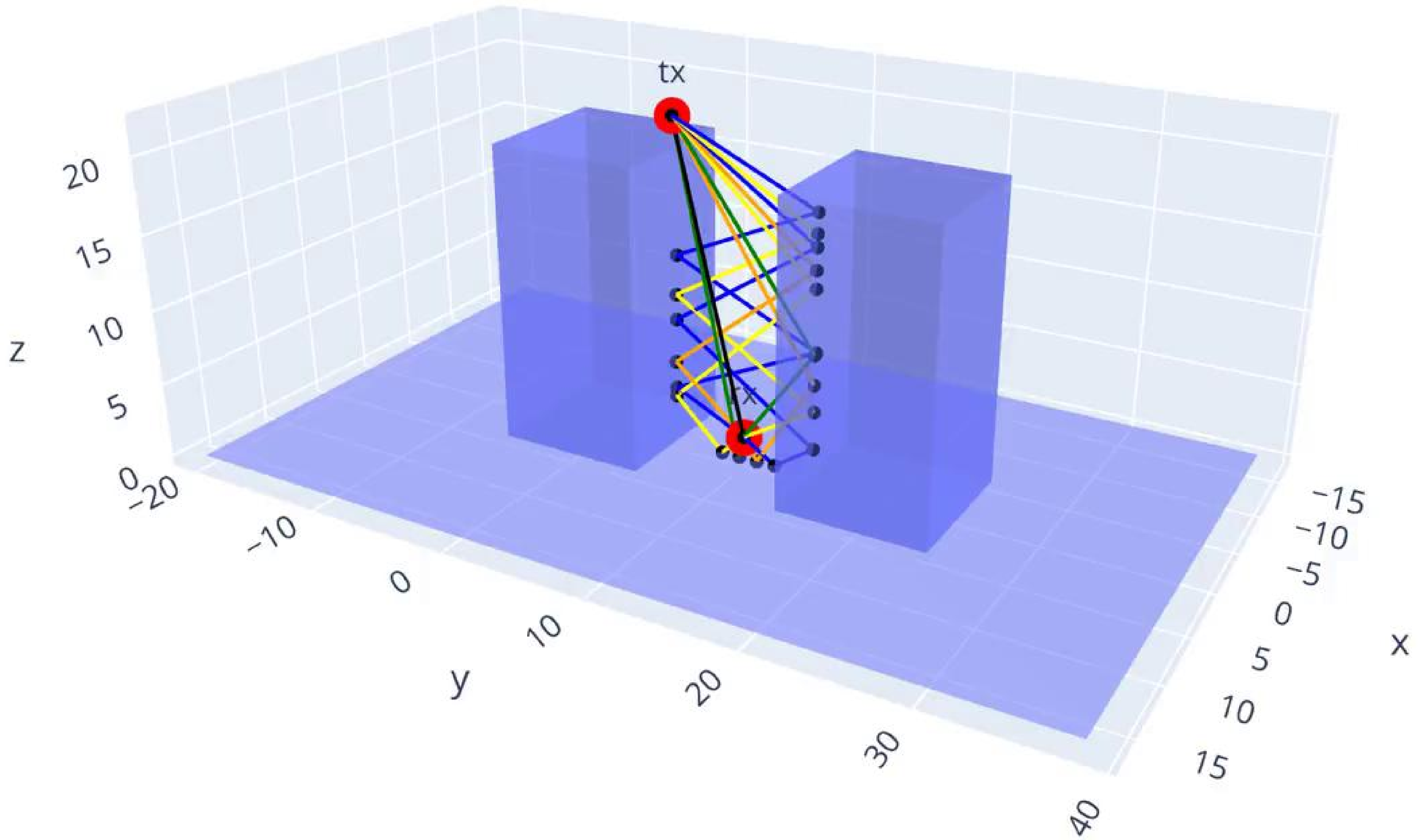
Differentiable Ray Tracing



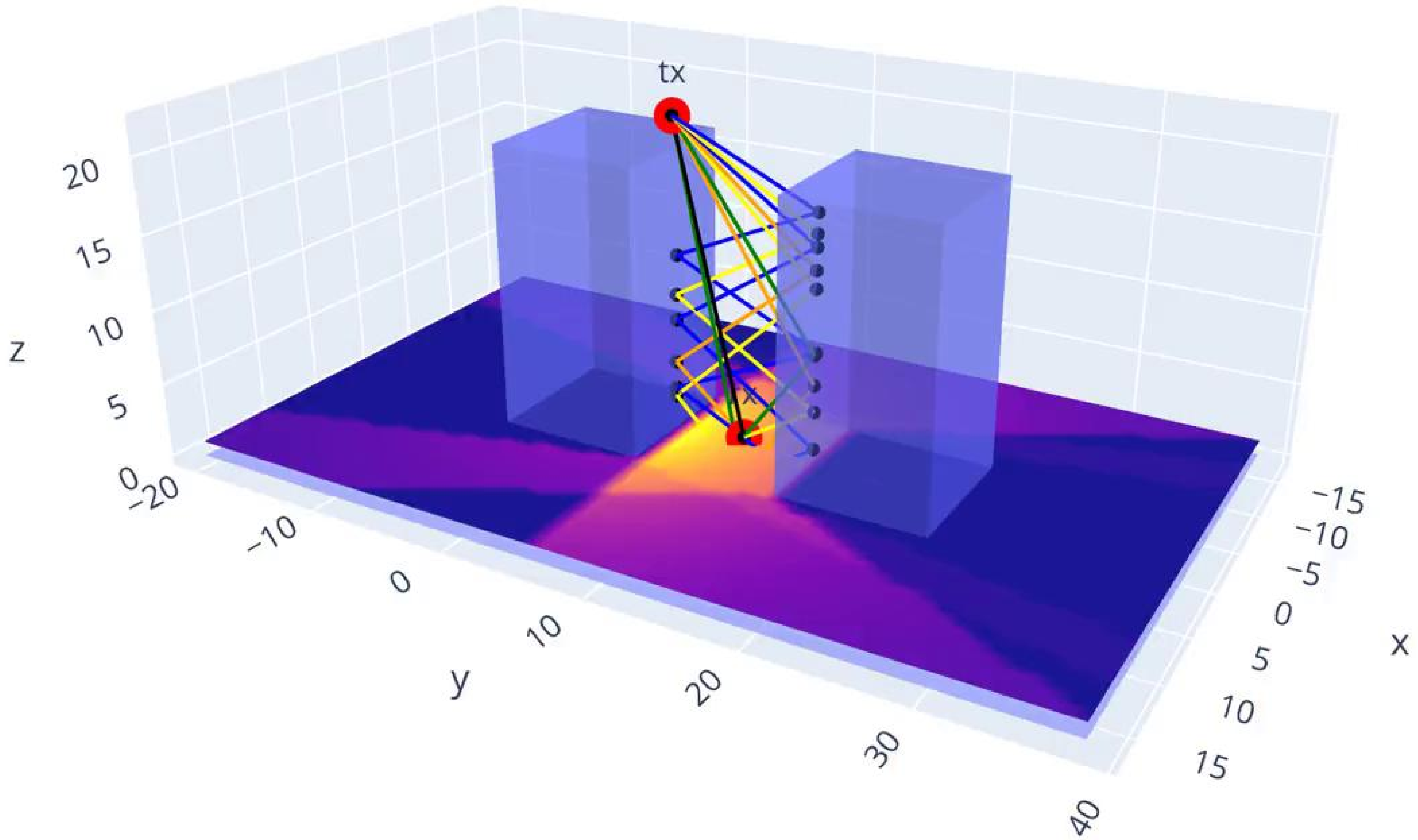
Differentiable Ray Tracing



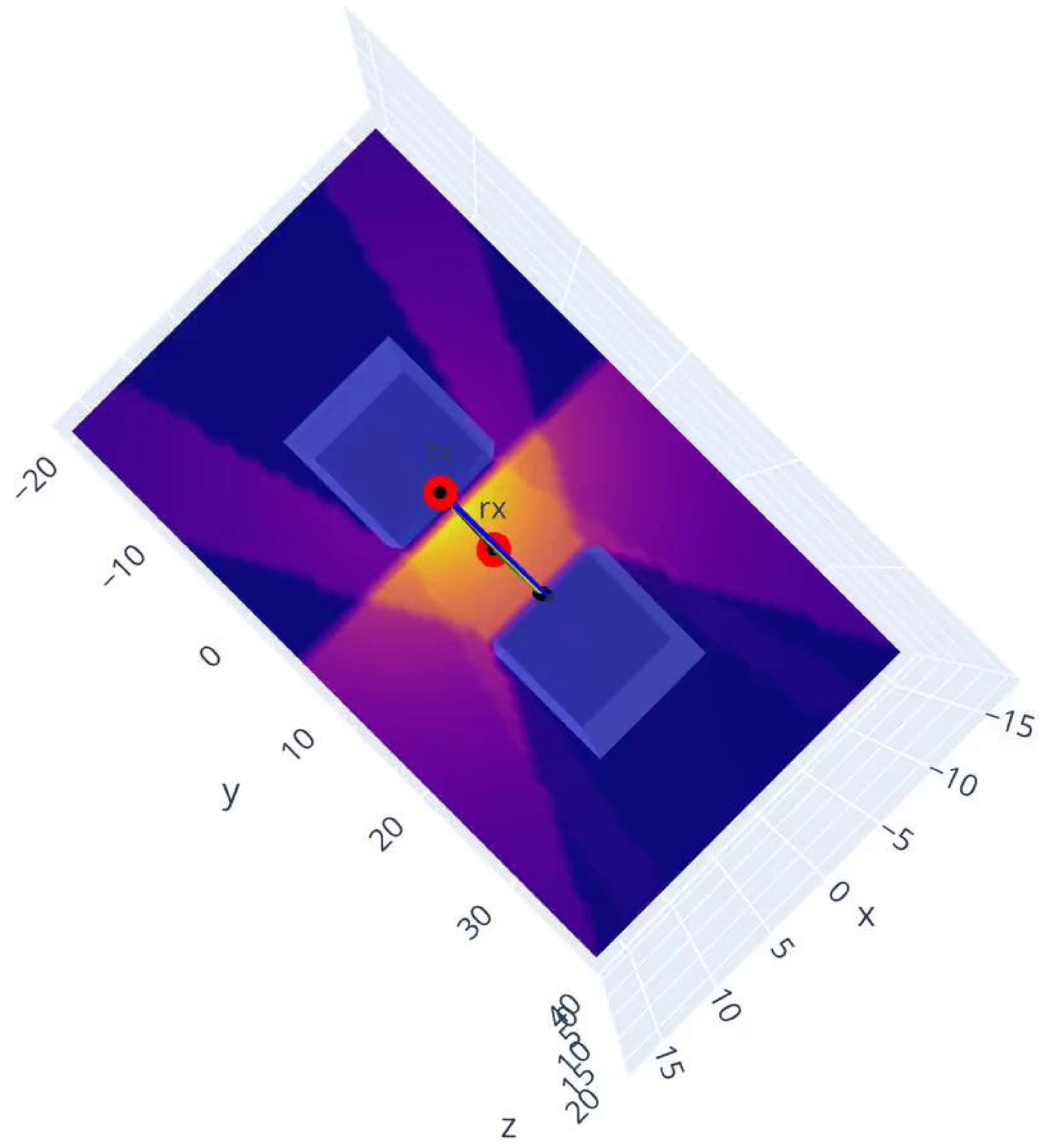
Differentiable Ray Tracing



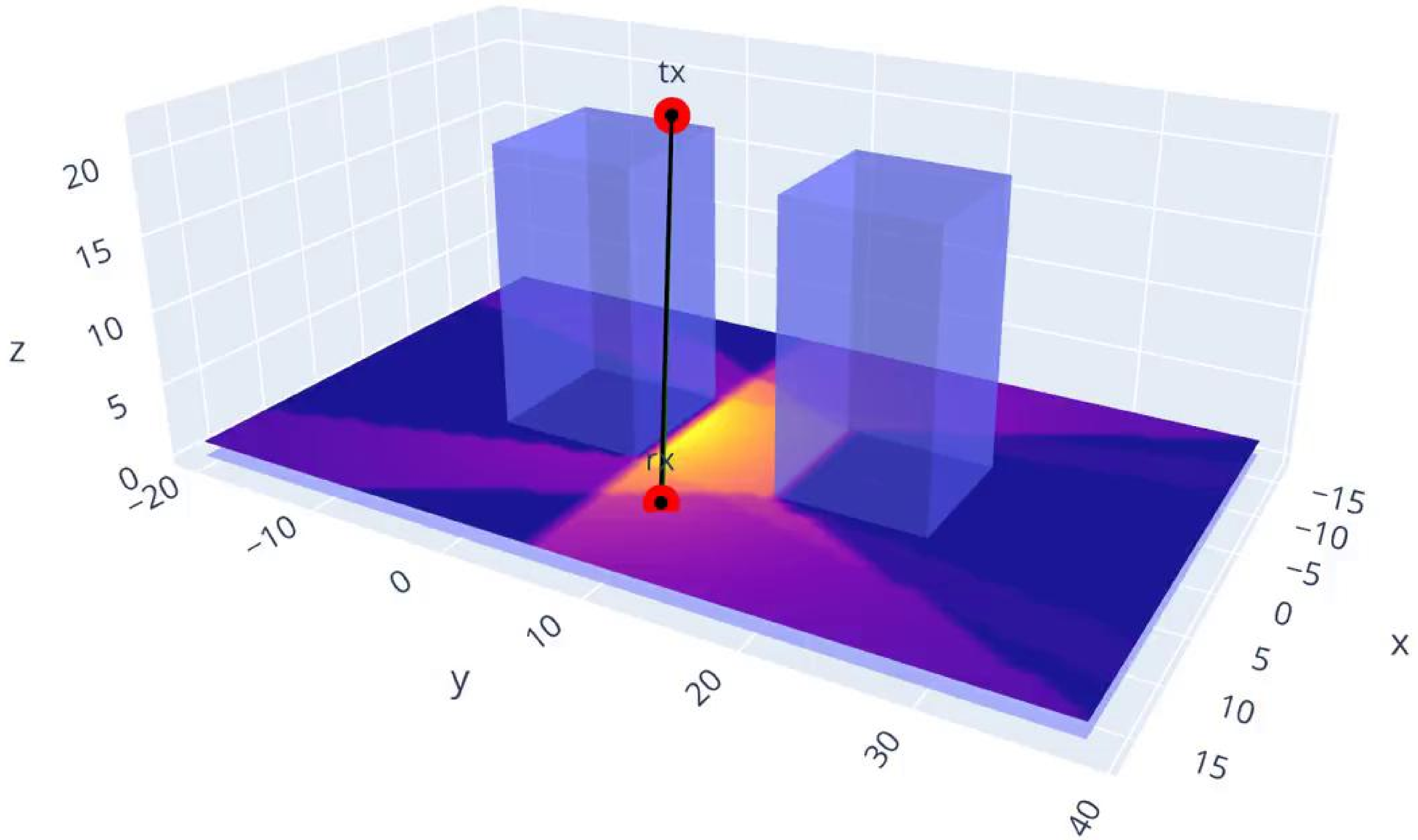
Differentiable Ray Tracing



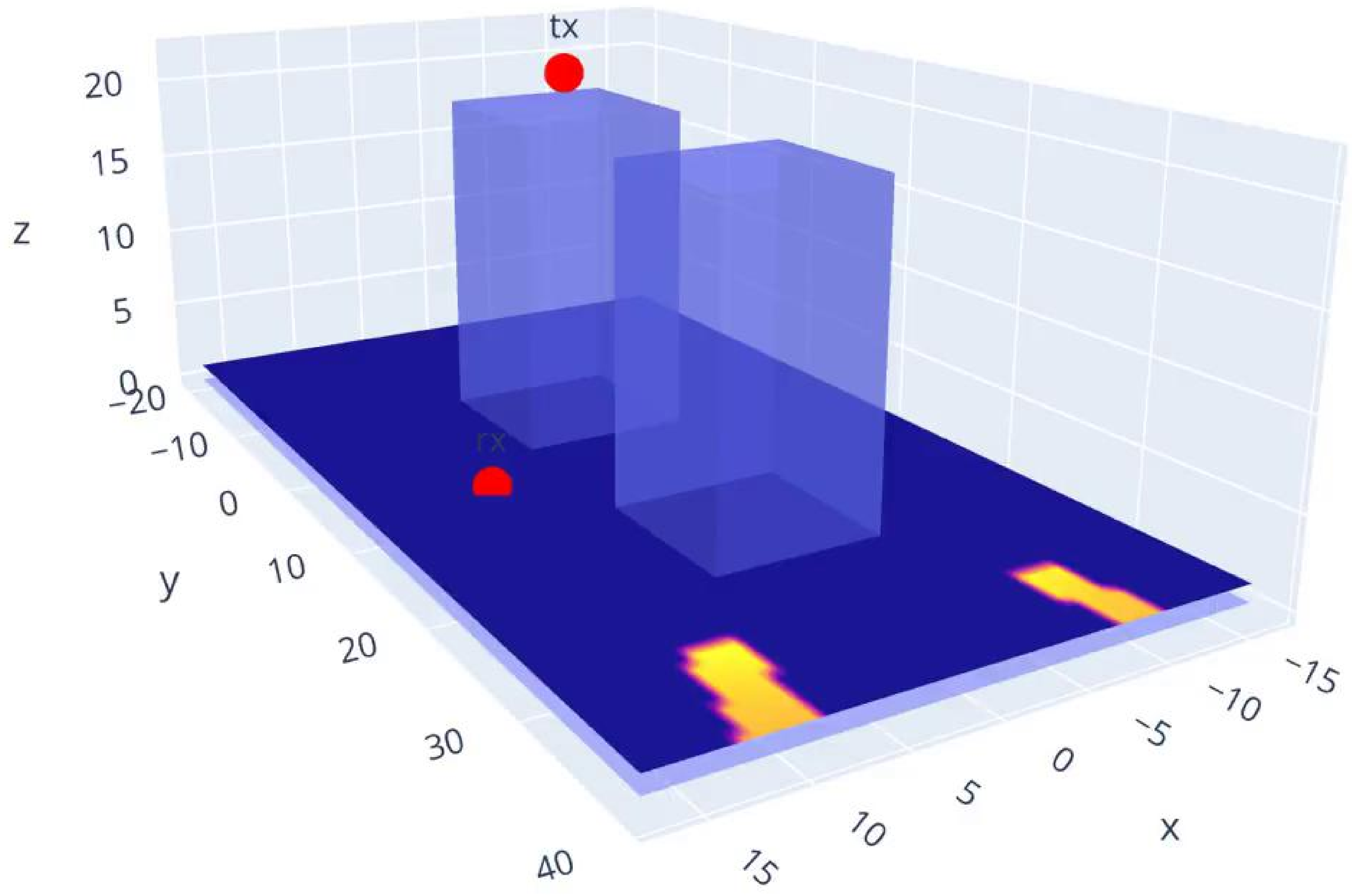
Differentiable Ray Tracing



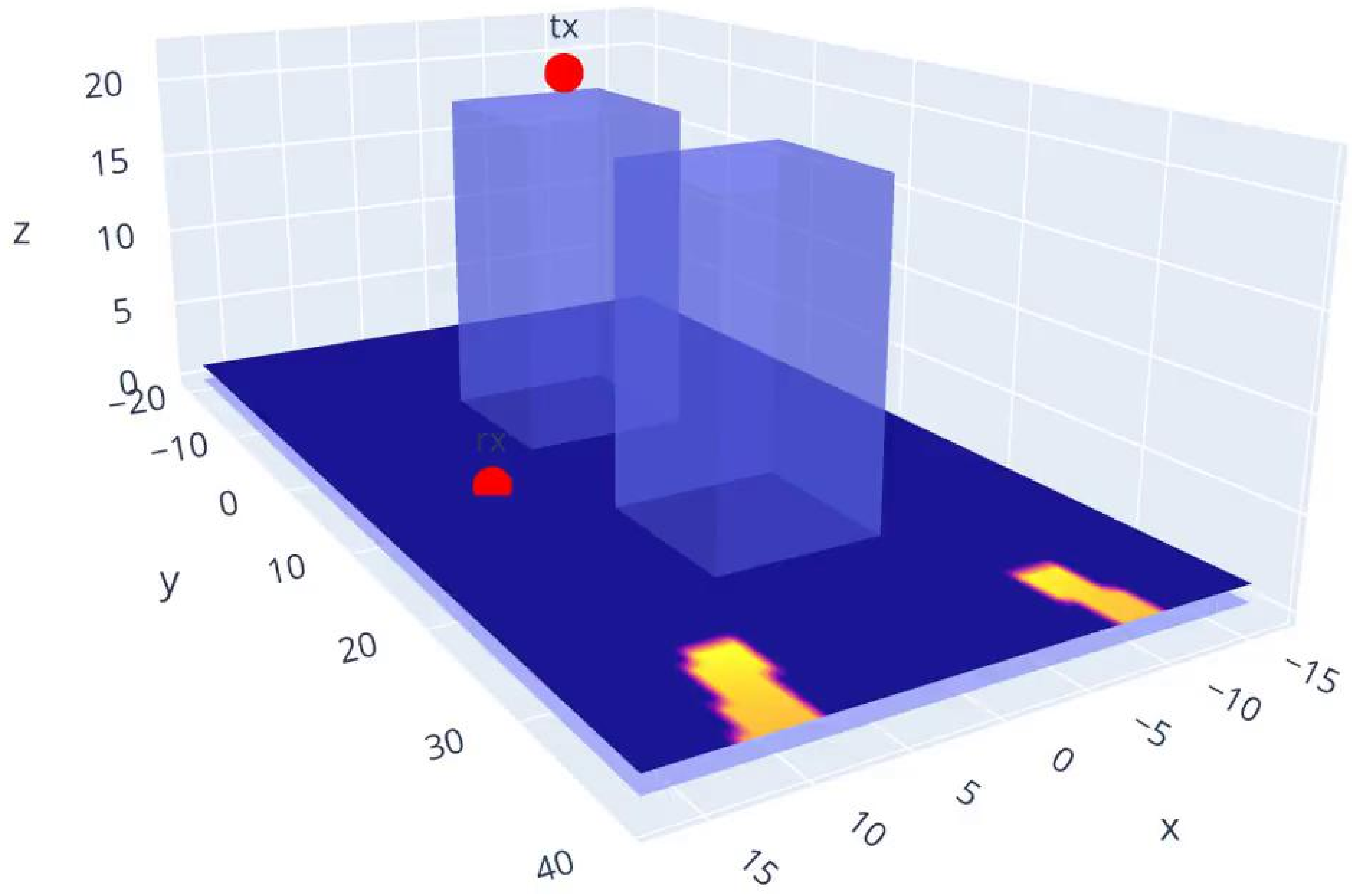
Differentiable Ray Tracing



Differentiable Ray Tracing



Differentiable Ray Tracing



One solution: differentiability.

```python
#
import numpy as np


def g(x):
    return np.cos(x)


def f(x):
    return x * g(2 * x) + 1


# df = ?
```

```python
import jax
import jax.numpy as jnp


def g(x):
    return jnp.cos(x)


def f(x):
    return x * g(2 * x) + 1


df = jax.grad(f)
```

Differentiability with auto-diff is easy!

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

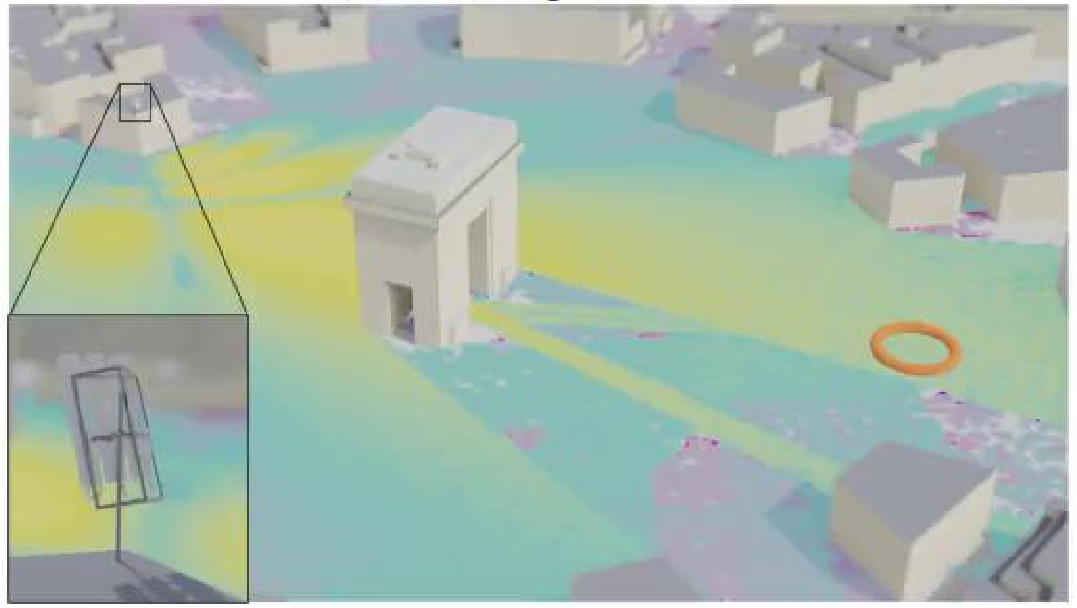
***Abstract*—Sionna™ is a GPU-accelerated open-source library for link-level simulations based on TensorFlow. Since release v0.14 it integrates a differentiable ray tracer (RT) for the simulation of radio wave propagation. This unique feature allows for the computation of gradients of the channel impulse 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

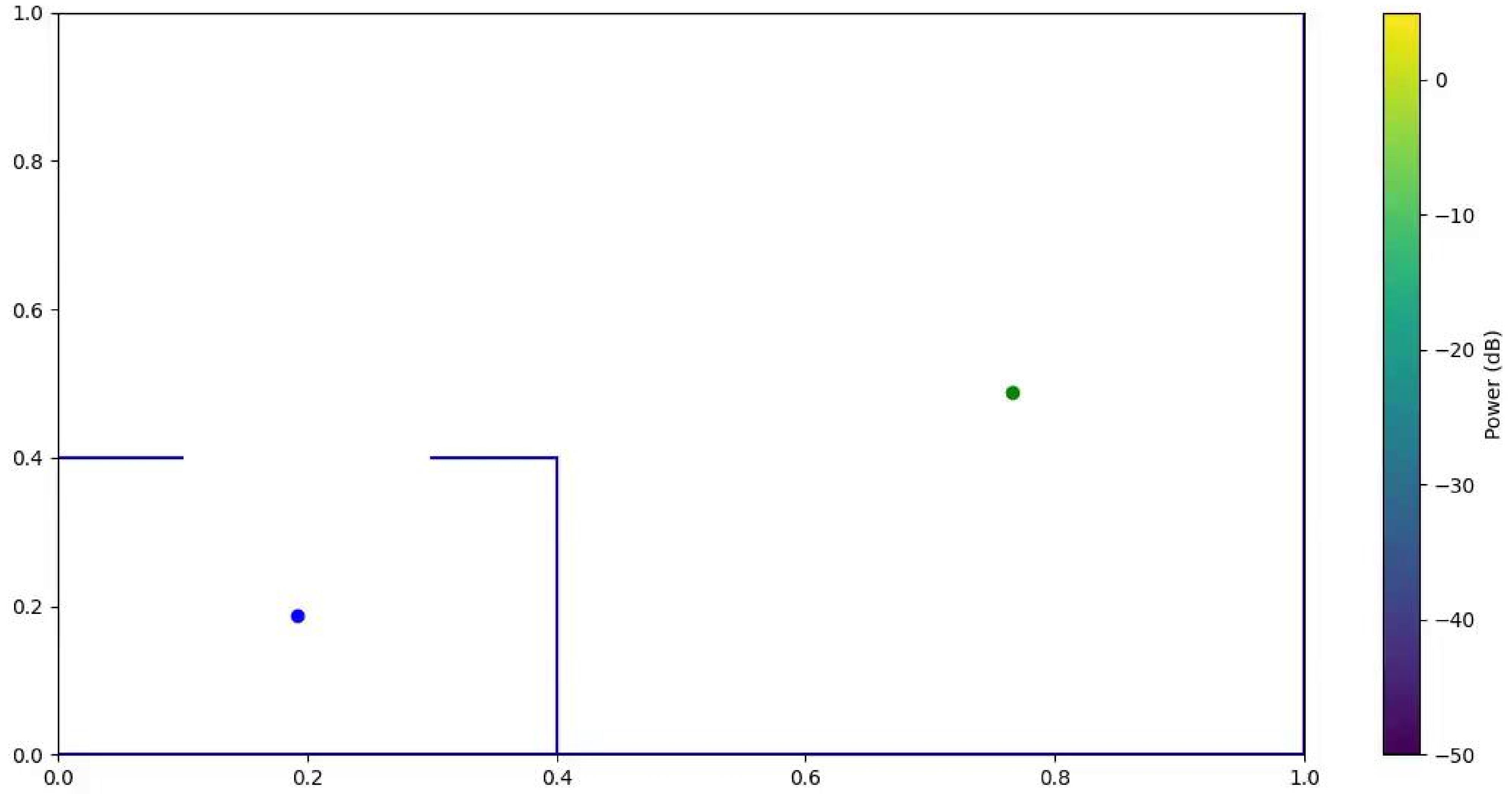


(b) After optimization

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).

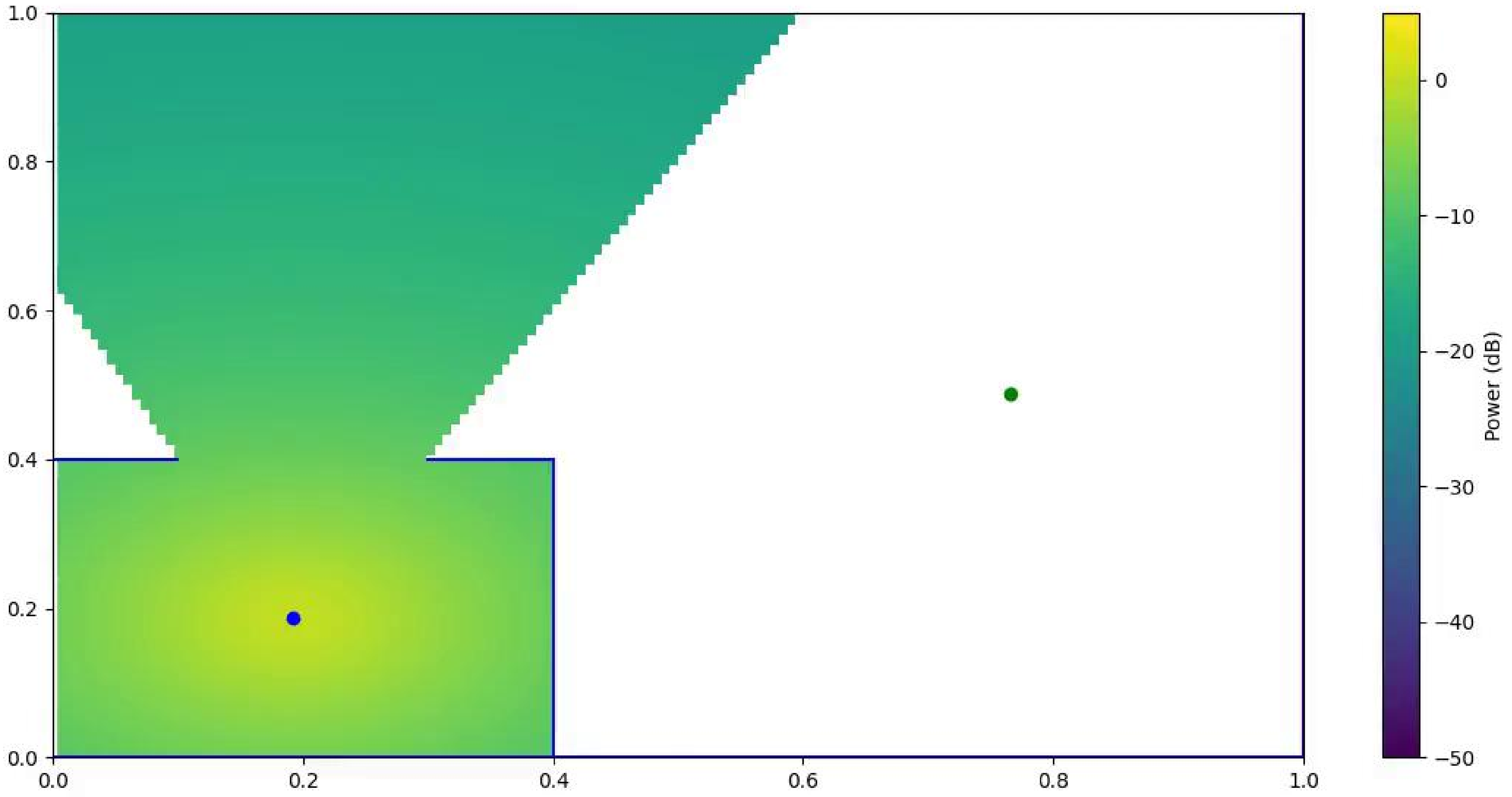
Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing



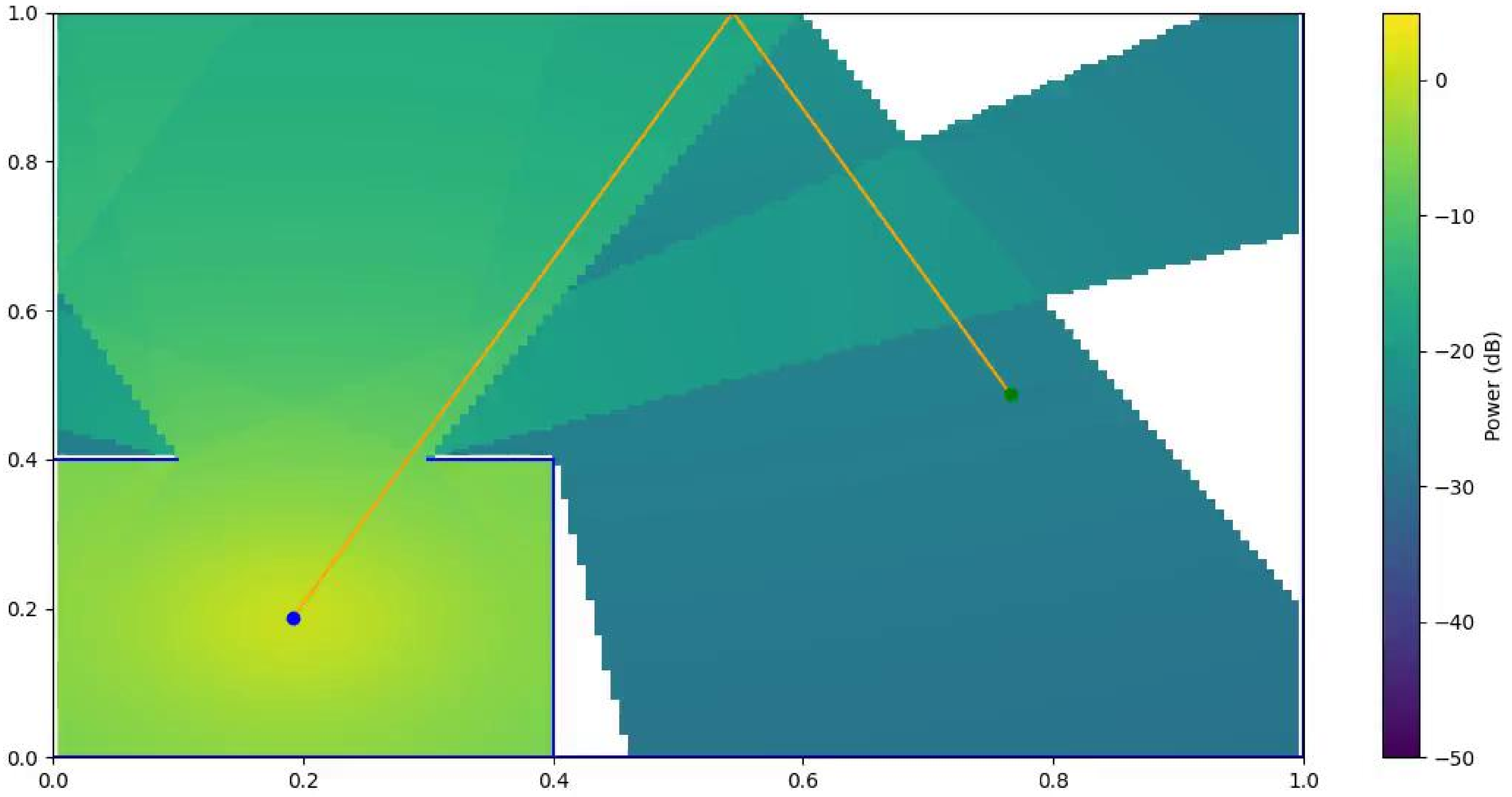
Challenge: number of paths.

Differentiable Ray Tracing



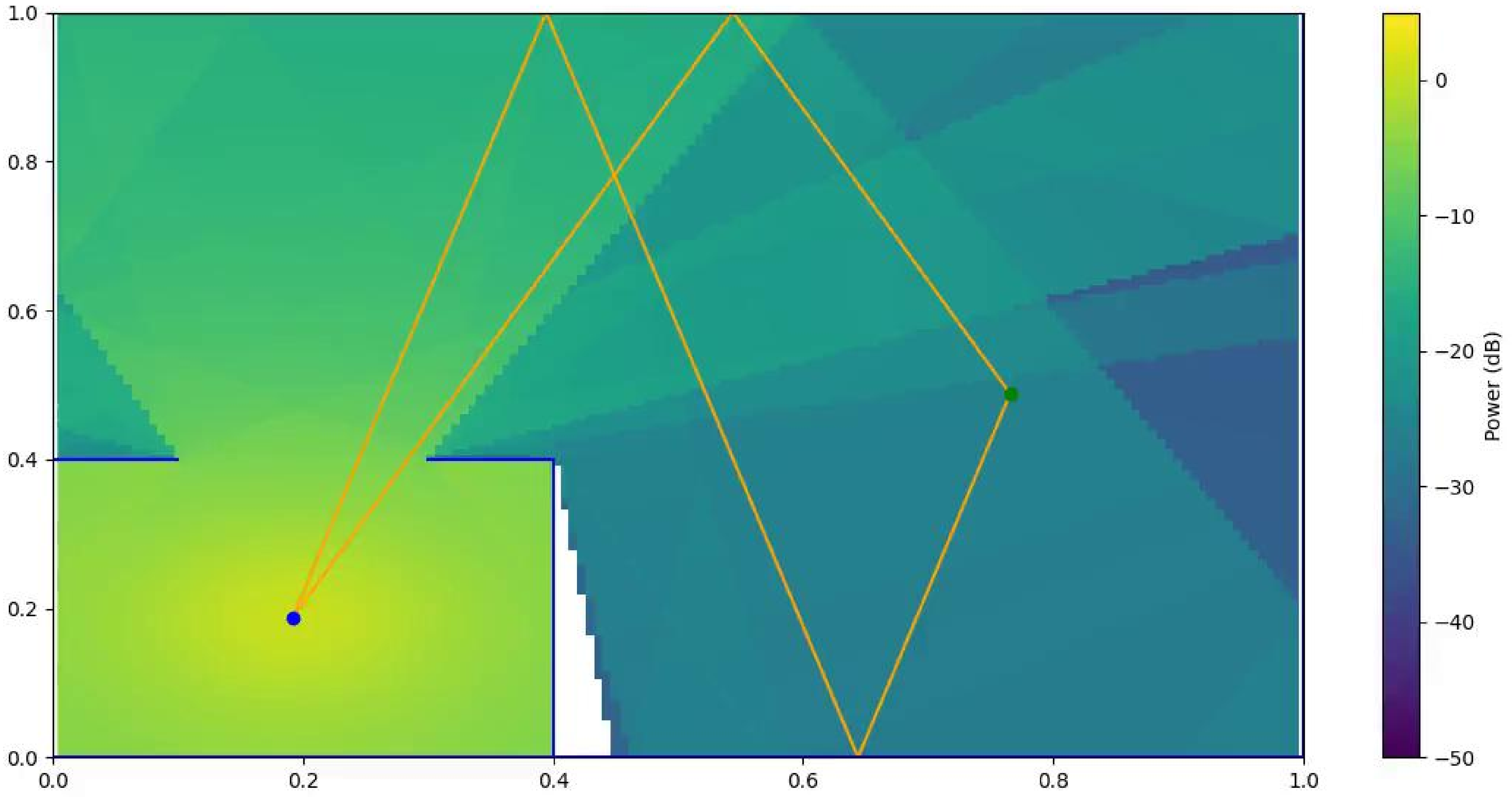
Challenge: number of paths.

Differentiable Ray Tracing



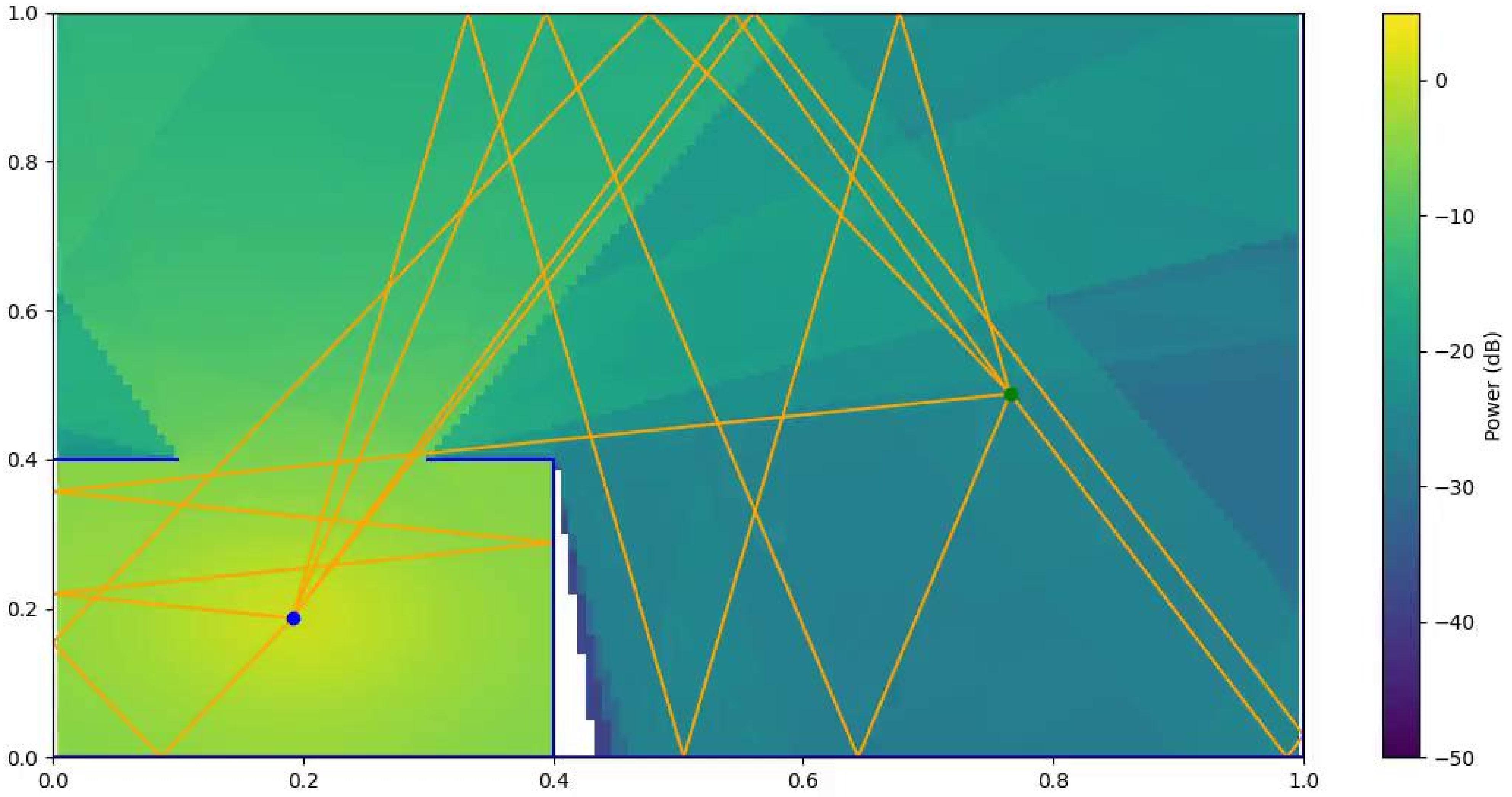
Challenge: number of paths.

Differentiable Ray Tracing



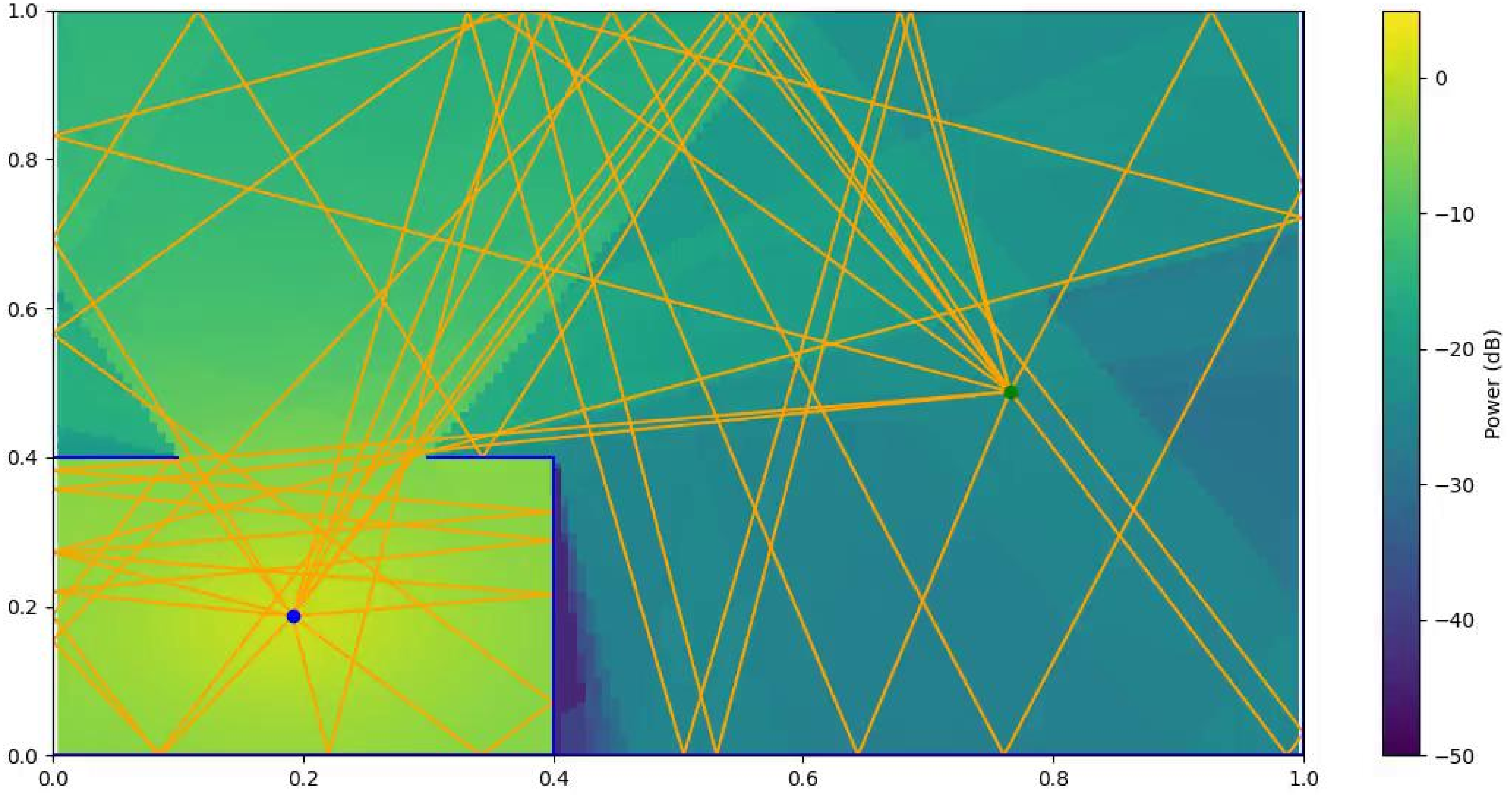
Challenge: number of paths.

Differentiable Ray Tracing



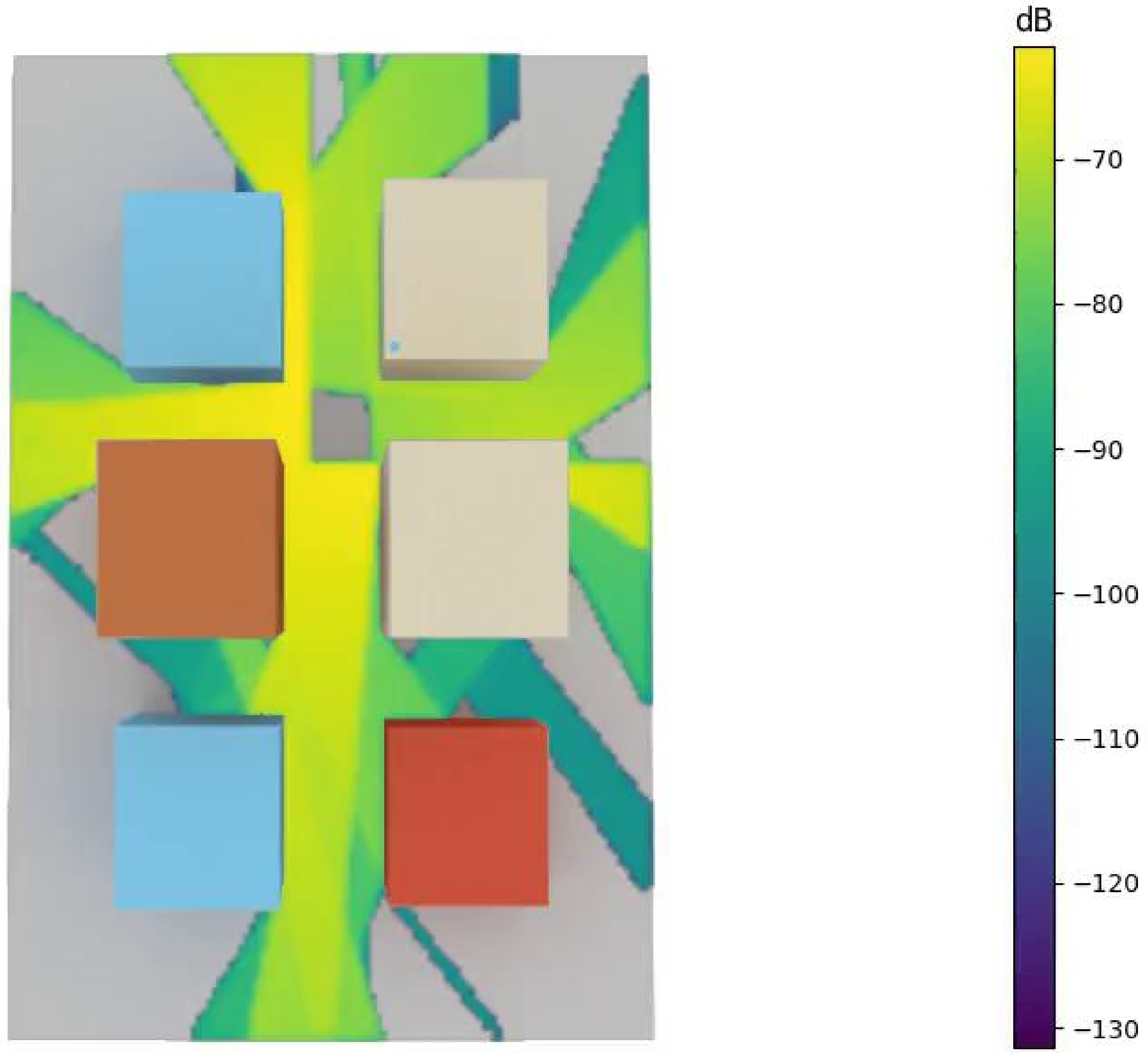
Challenge: number of paths.

Differentiable Ray Tracing



Challenge: number of paths.

Differentiable Ray Tracing

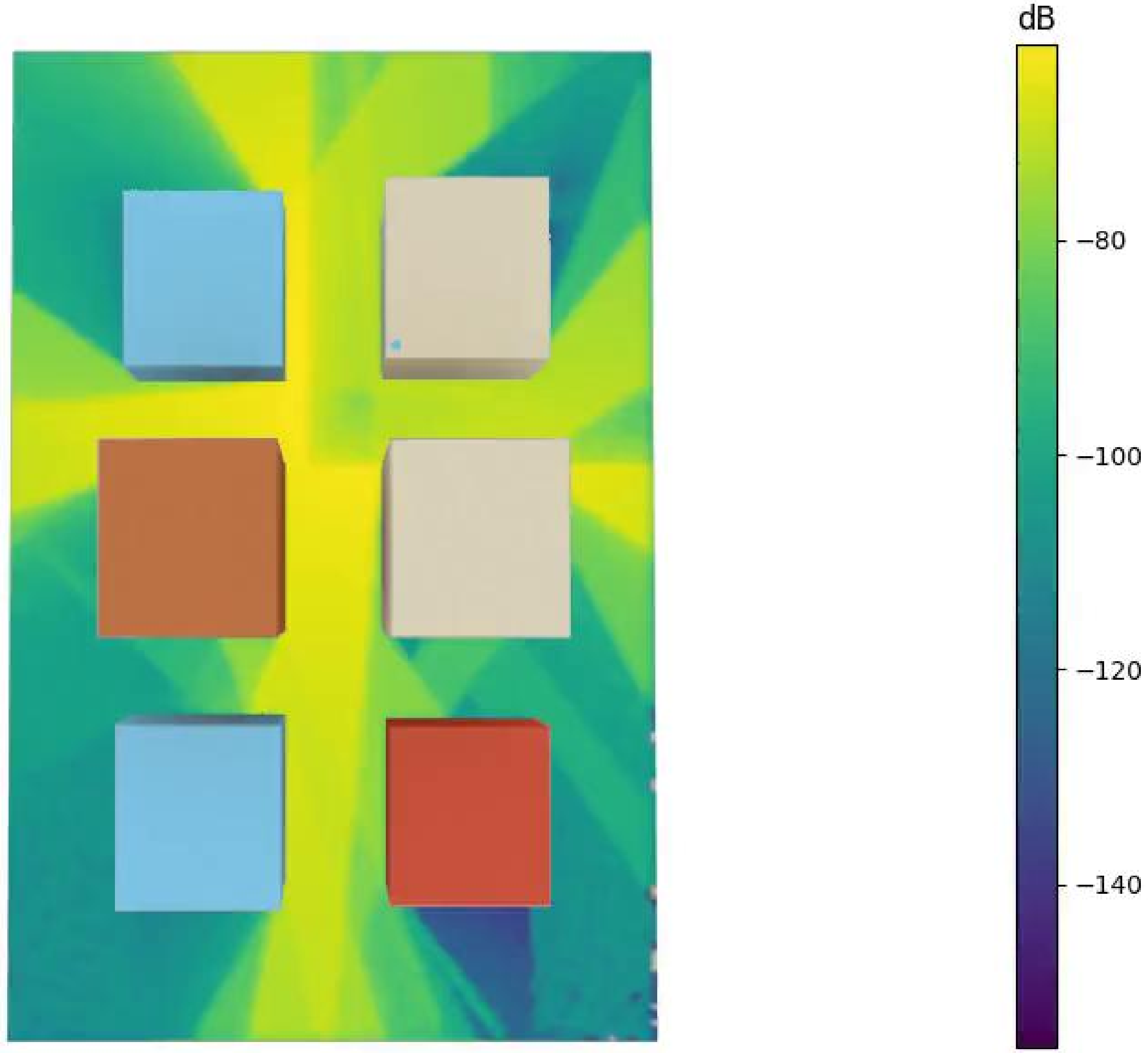


LOS + reflection

Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing

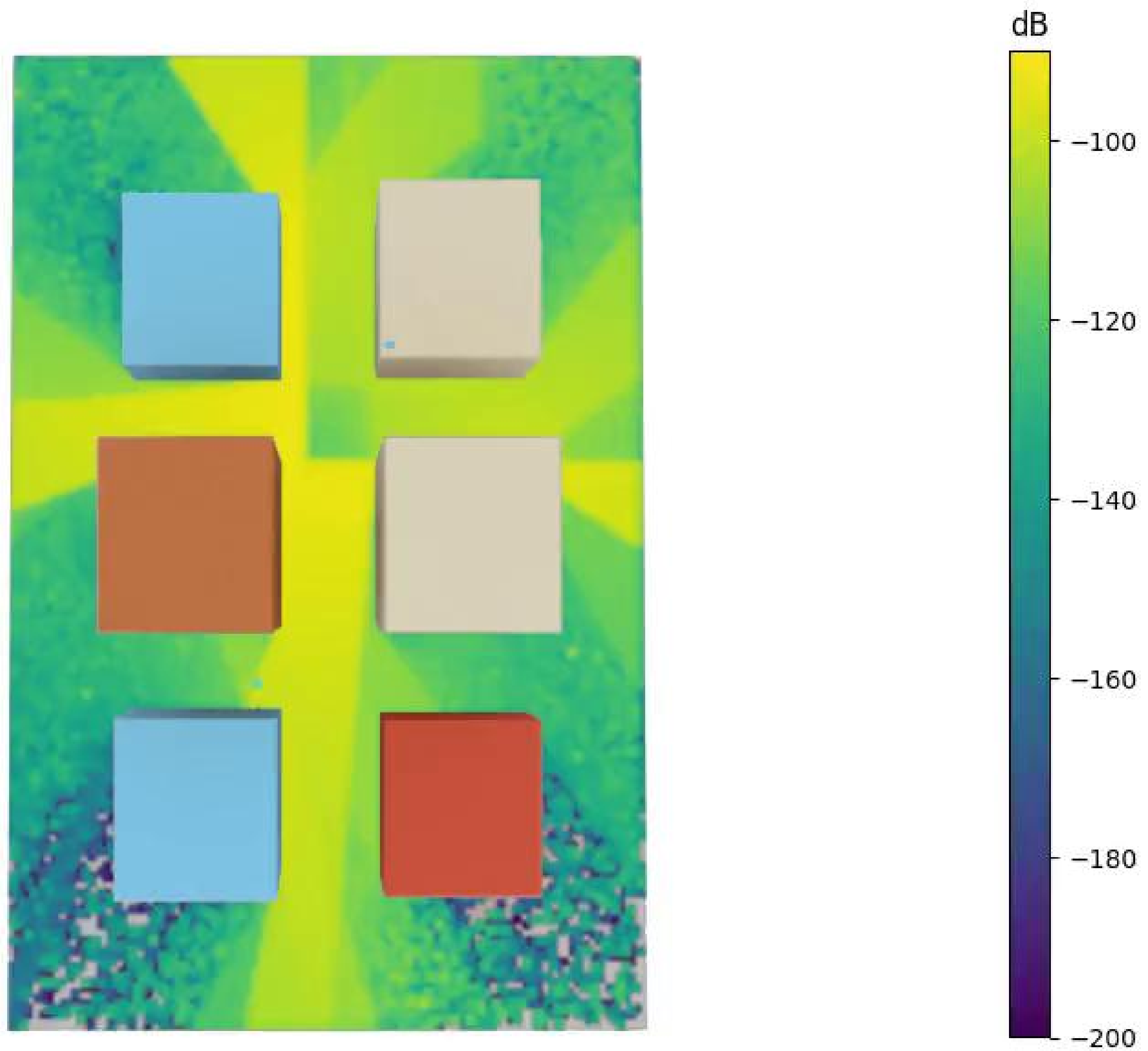


LOS + reflection + diffraction

Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

Differentiable Ray Tracing



LOS + reflection + scattering

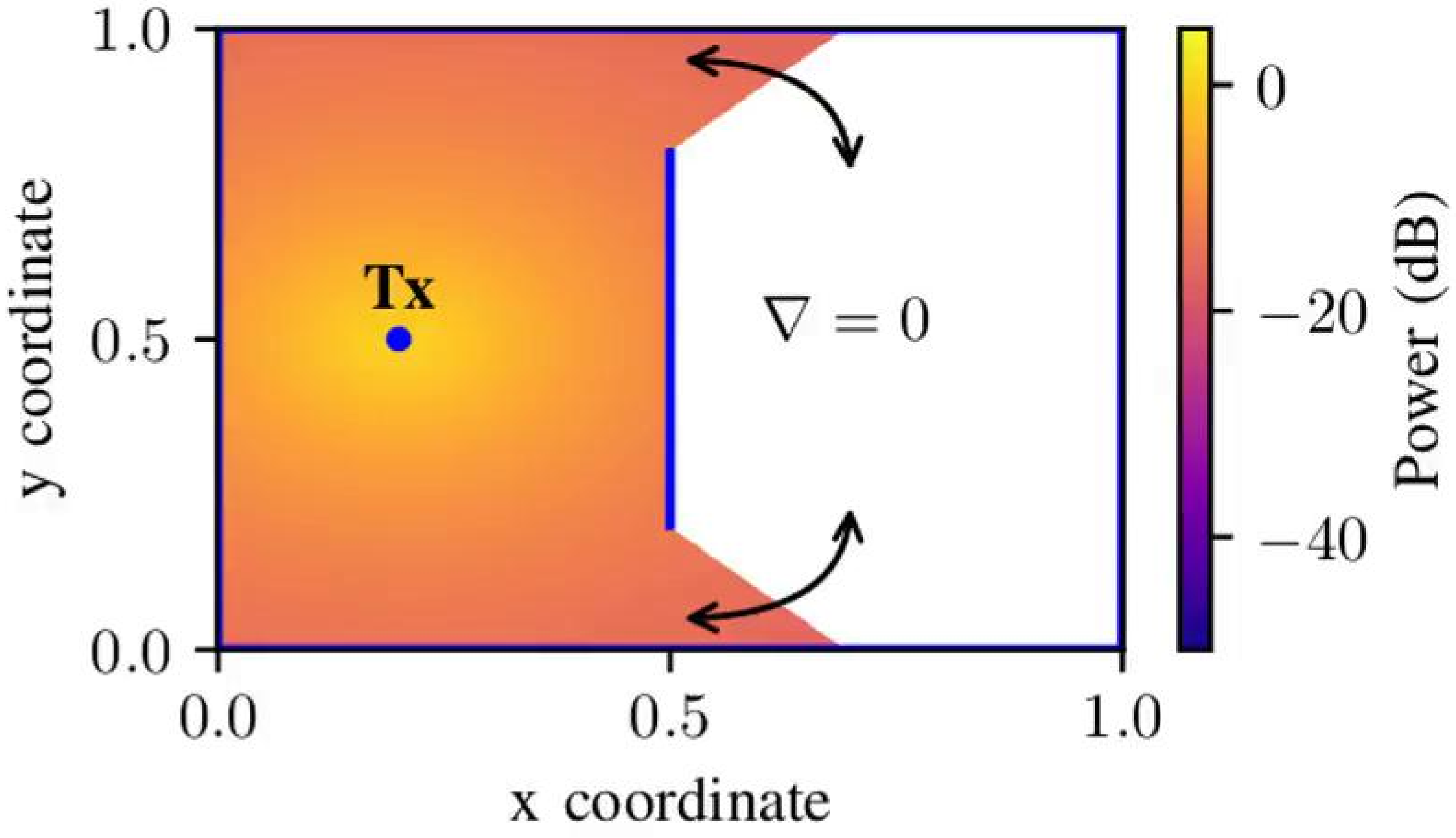
Challenge: coverage vs order and types.

Credits: Sionna authors, Nvidia.

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

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

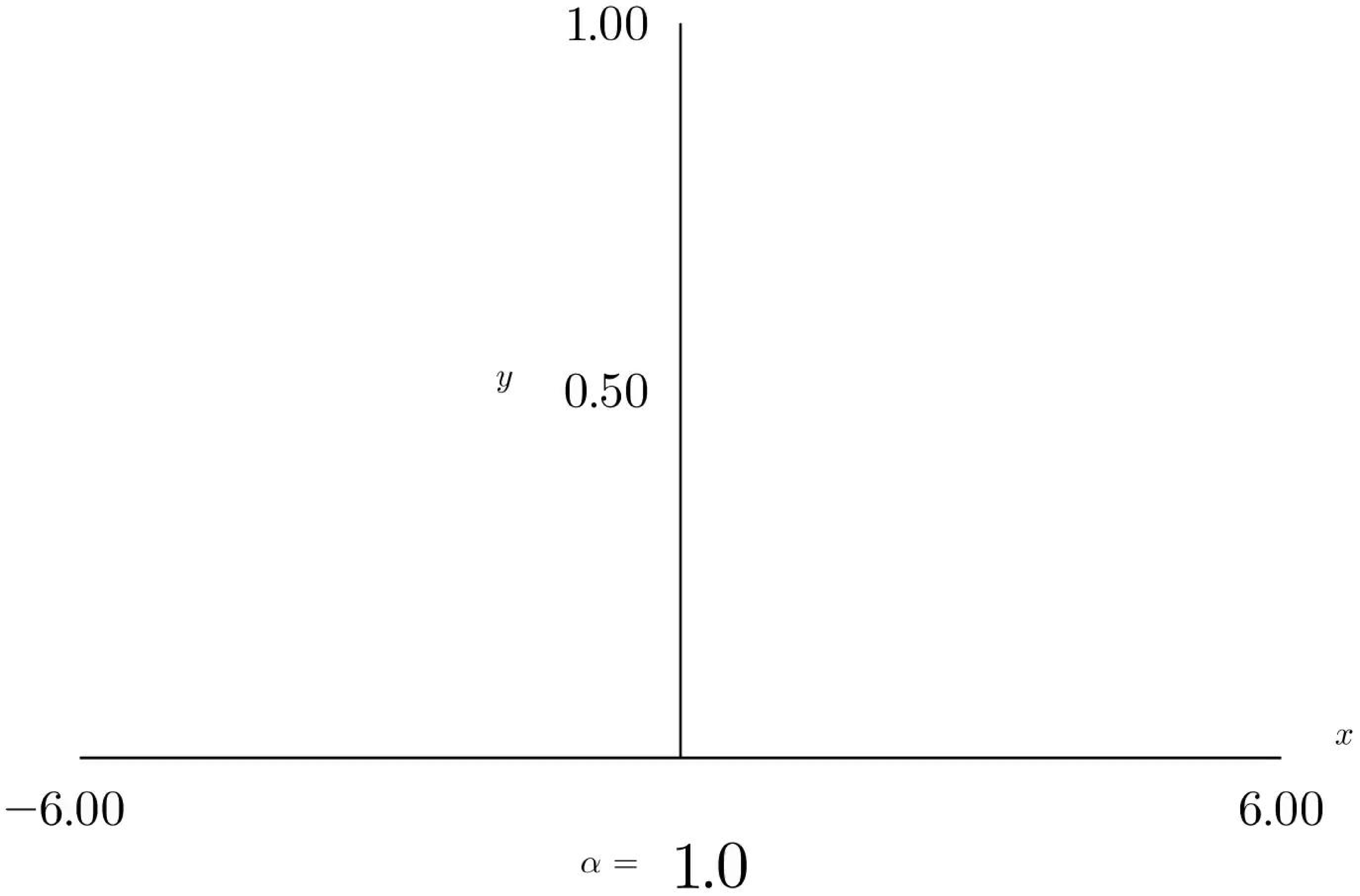
and the hard sigmoid is the piecewise linear function defined by

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

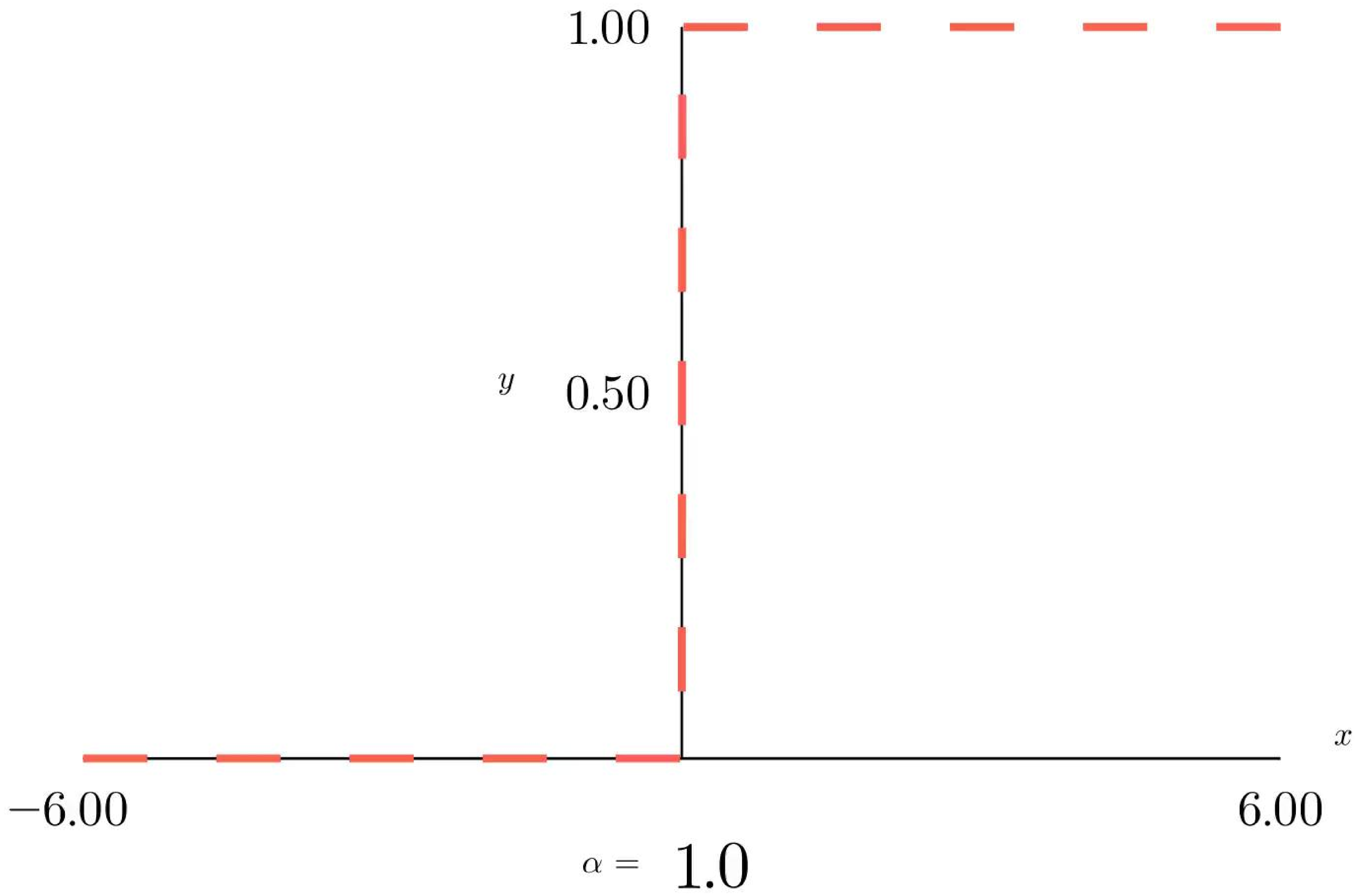
where

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

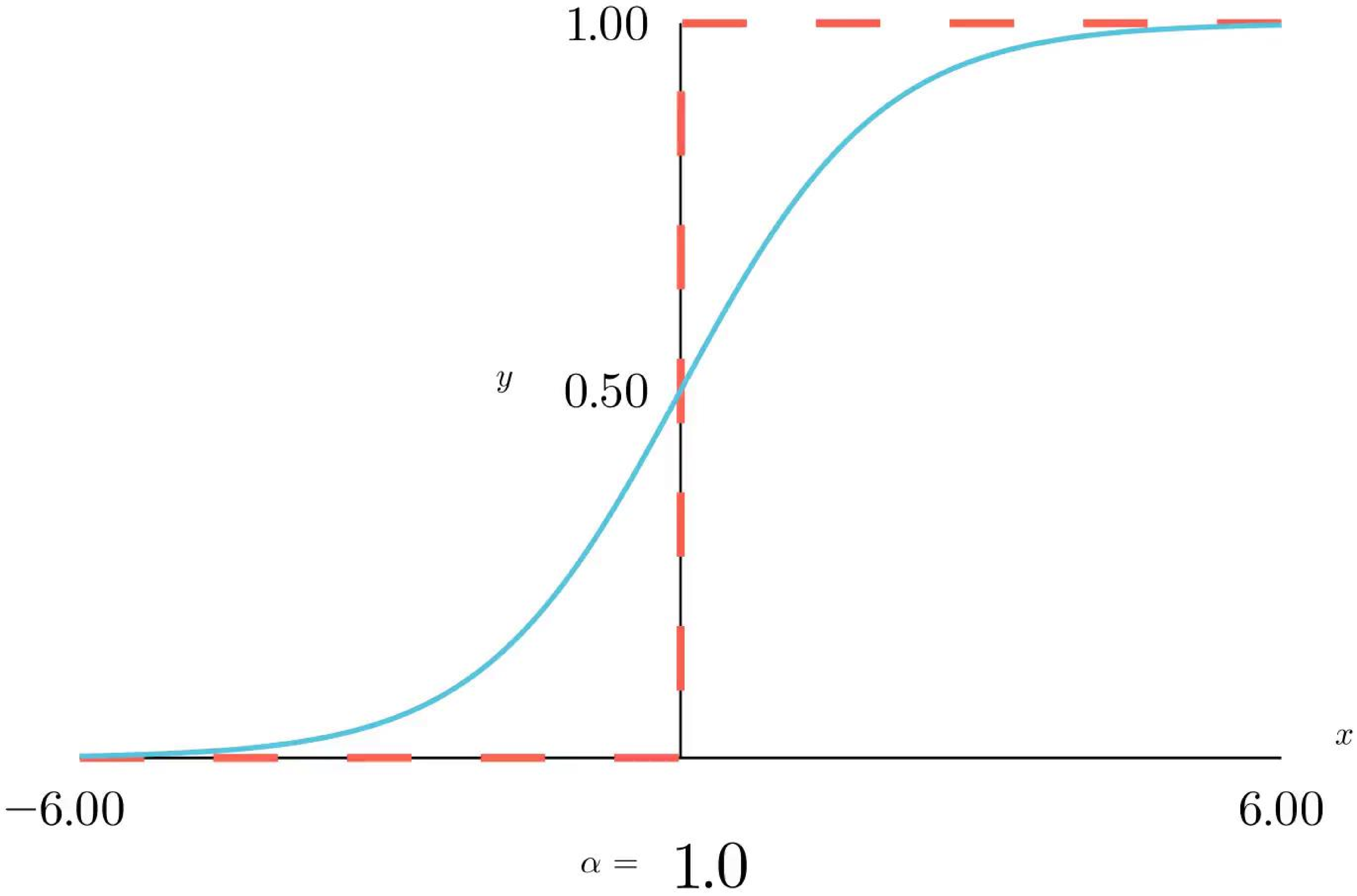
Present work: discontinuity smoothing



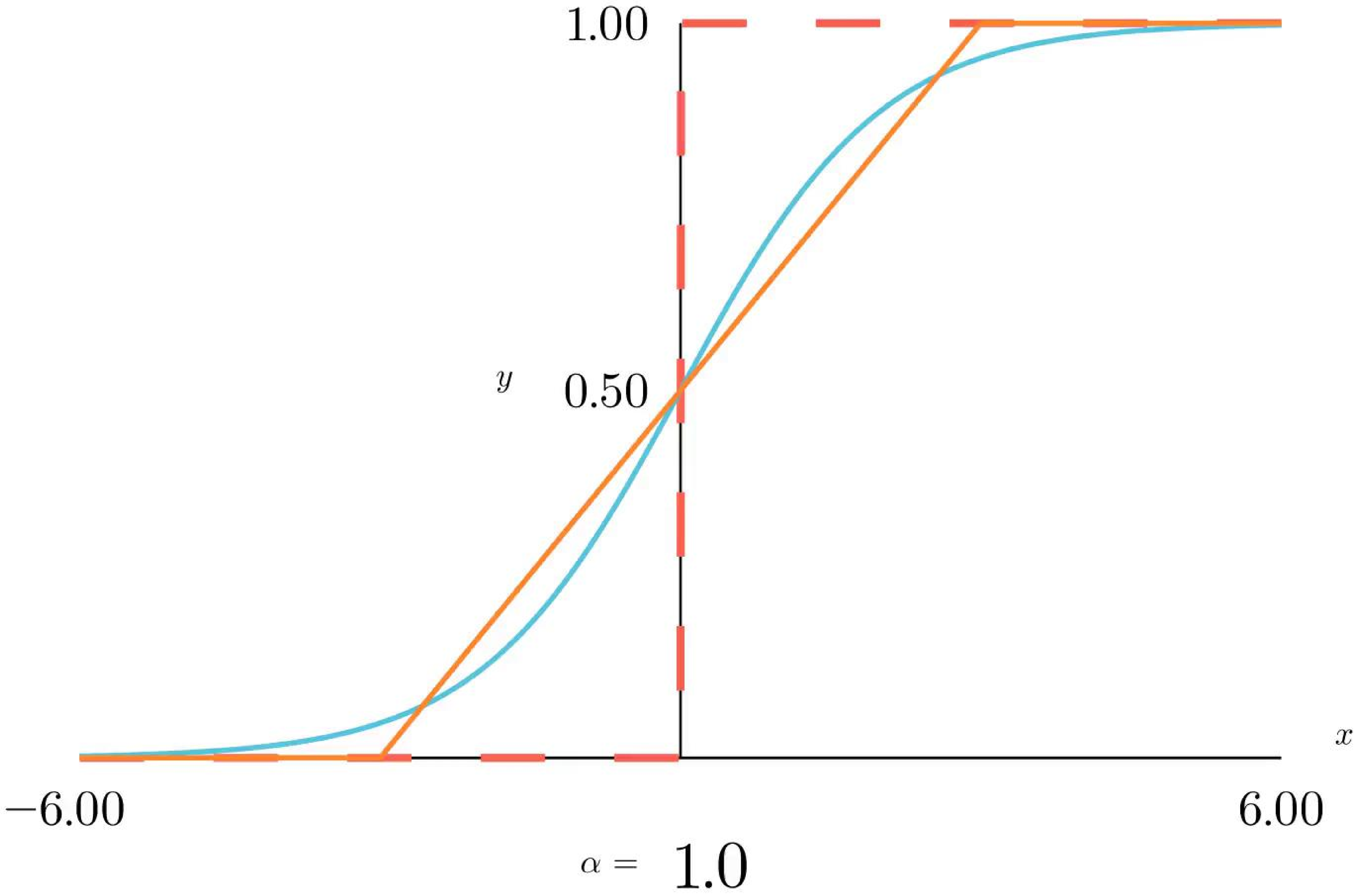
Present work: discontinuity smoothing



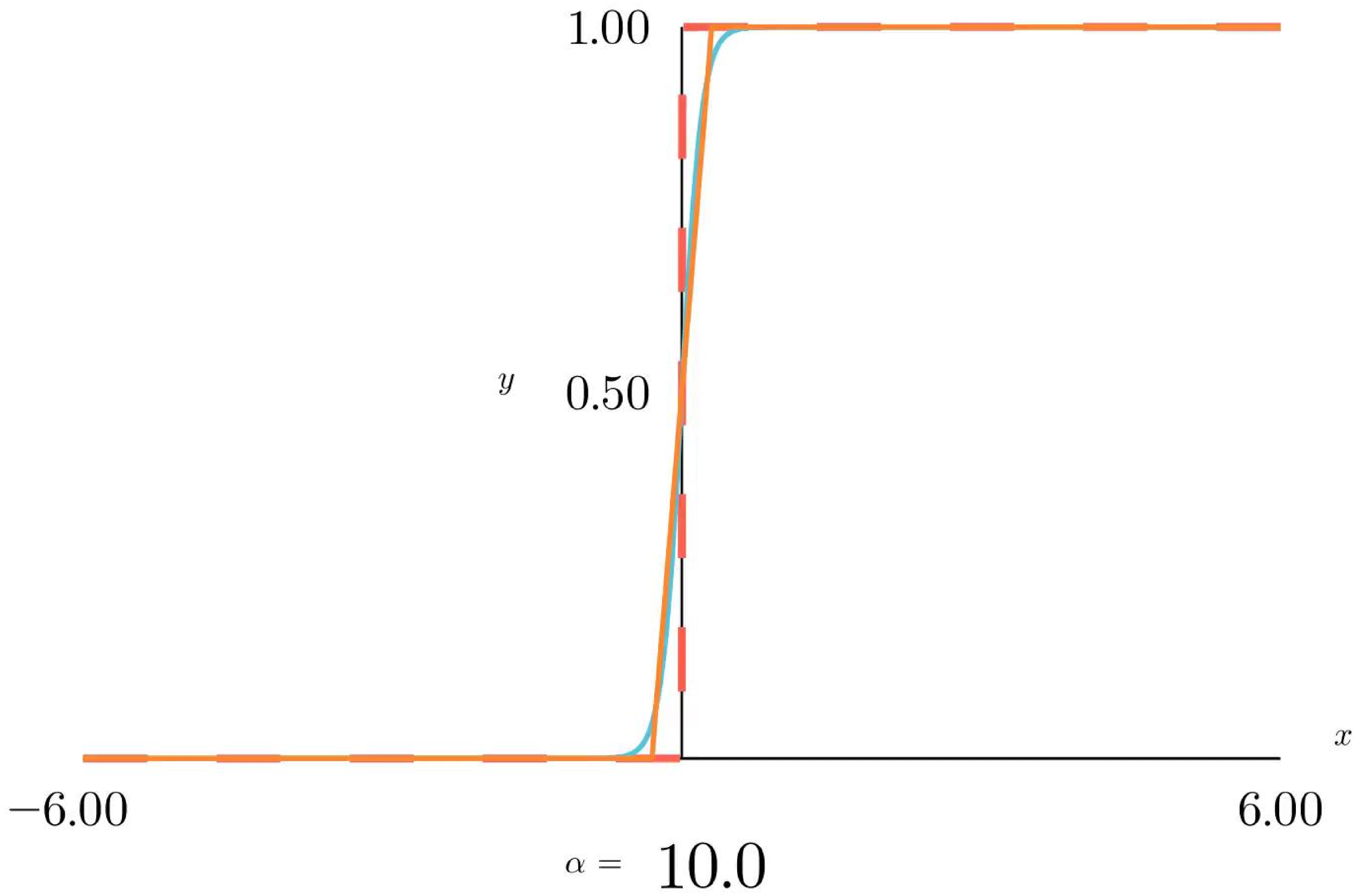
Present work: discontinuity smoothing



Present work: discontinuity smoothing

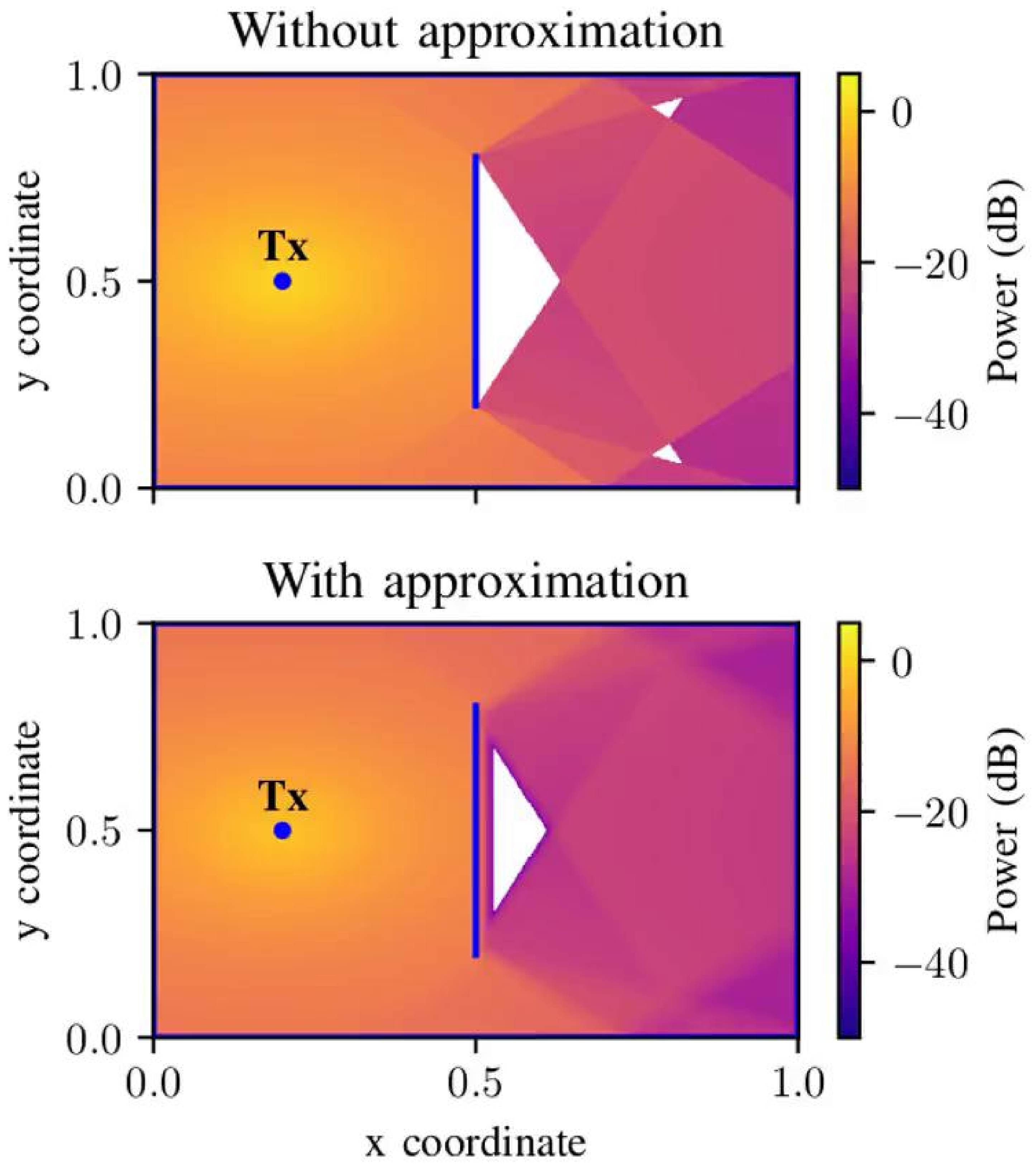


Present work: discontinuity smoothing



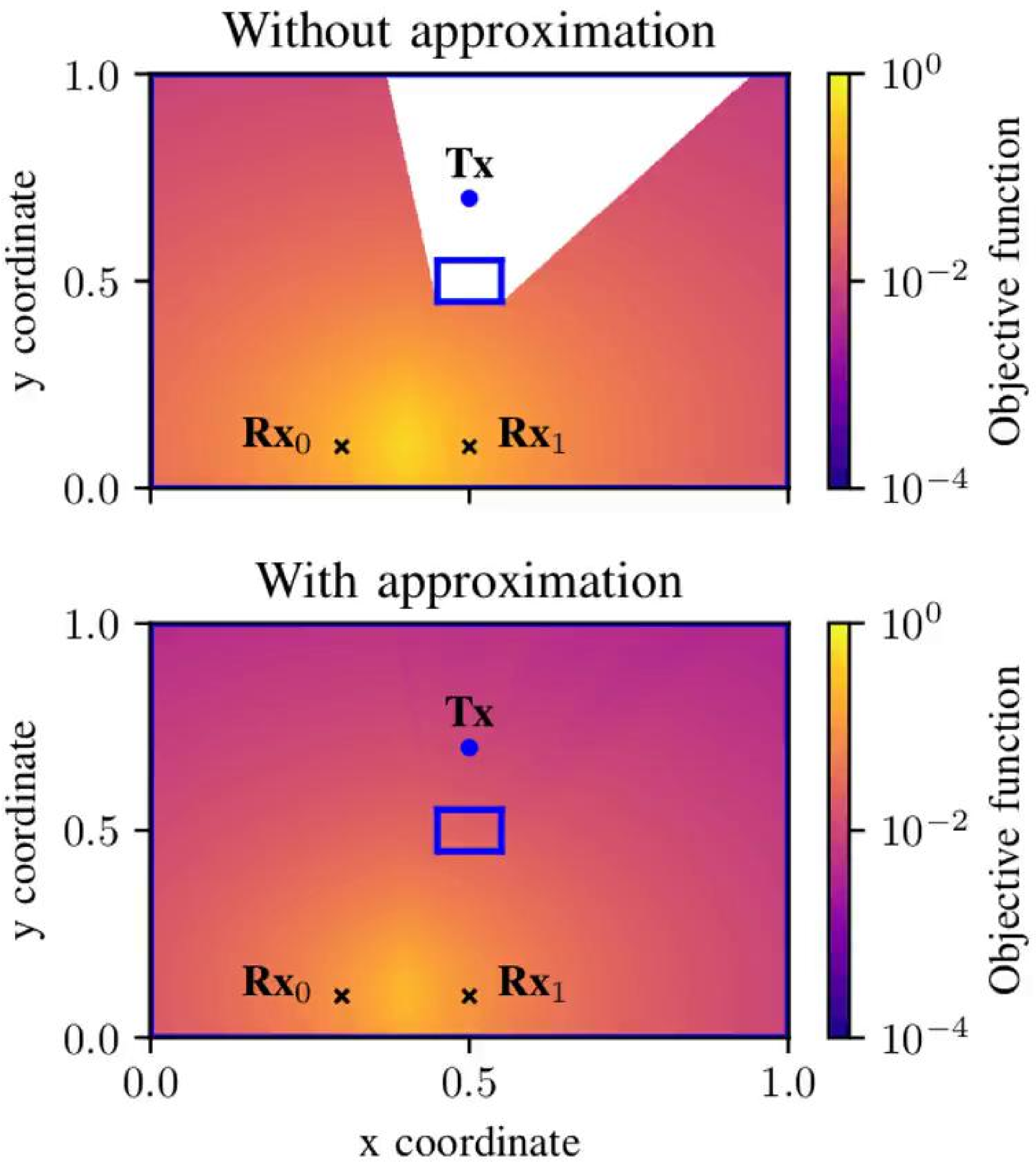
Present work: discontinuity smoothing

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

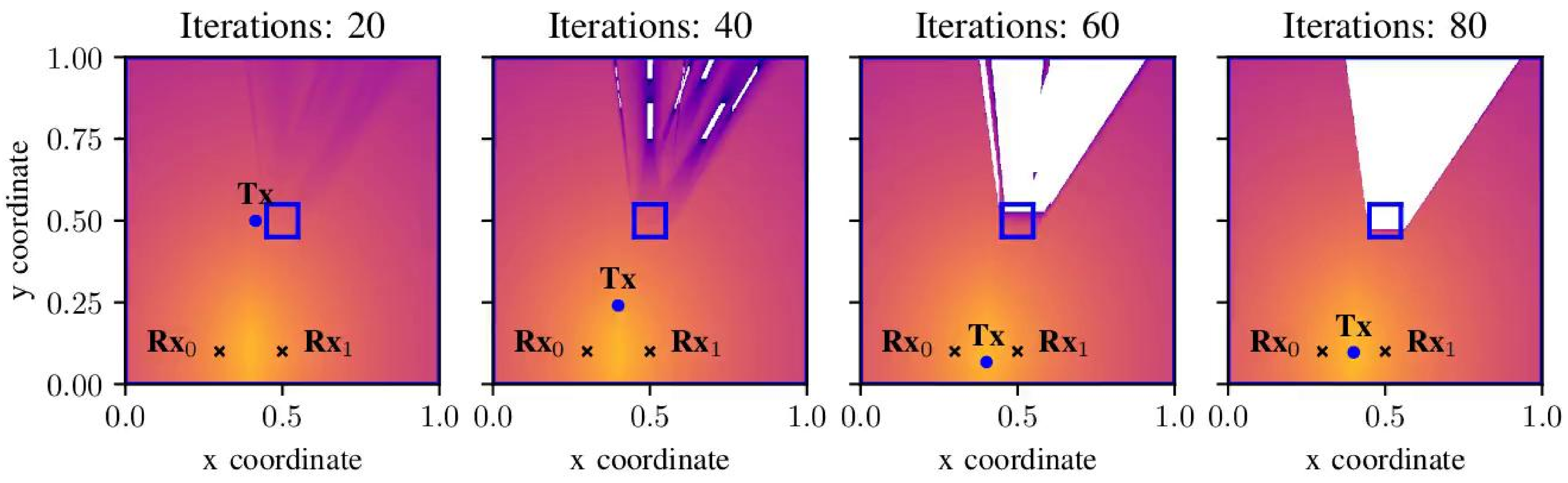


Present work: discontinuity smoothing

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



Present work: discontinuity smoothing



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