Learning to Sample Ray Paths for Faster Point-to-Point Ray Tracing

Authors: **Jérome Eertmans, Nicola Di Cicco, Claude Oestges, Laurent Jacques, Enrico Maria Vitucci, Vittorio Degli Esposti**

June 2024

COST INTERACT meeting, Helsinki

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	Ray Launching (RL)
Complexity		
Accuracy		
Best for		

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	Ray Launching (RL)
Complexity	Exponential	
Accuracy		
Best for		

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	**Ray Launching (RL)**
Complexity	Exponential	Linear*
Accuracy		
Best for		

*: Depends on the number of rays launched.

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	**Ray Launching (RL)**
Complexity	Exponential	Linear*
Accuracy	Excellent	
Best for		

*: Depends on the number of rays launched.

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	**Ray Launching (RL)**
Complexity	Exponential	Linear*
Accuracy	Excellent	Good*
Best for		

*: Depends on the number of rays launched.

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	**Ray Launching (RL)**
Complexity	Exponential	Linear*
Accuracy	Excellent	Good*
Best for	P2P scenarios	

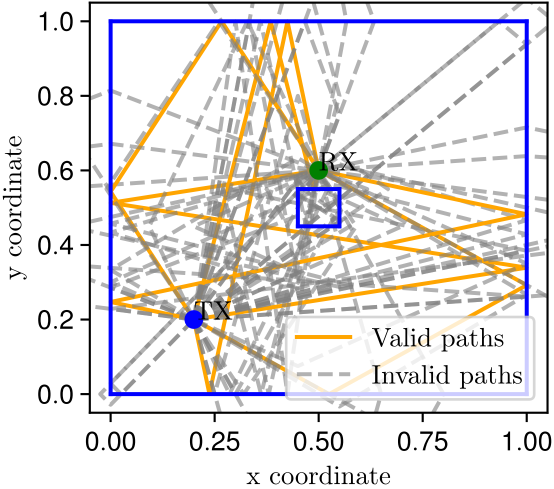
*: Depends on the number of rays launched.

Ray Tracing for Radio Propagation

	Point-to-Point (P2P) Ray Tracing (RT)	**Ray Launching (RL)**
Complexity	Exponential	Linear*
Accuracy	Excellent	Good*
Best for	P2P scenarios	Coverage map

*: Depends on the number of rays launched.

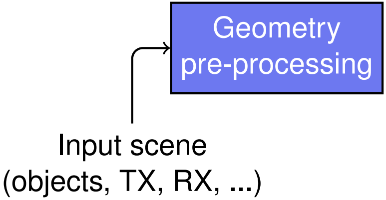
The curse of P2P RT



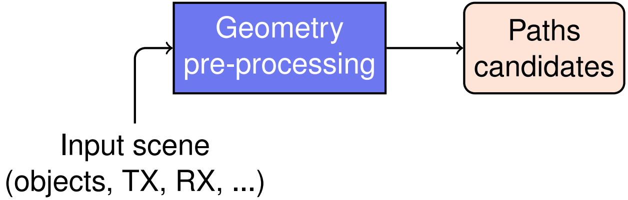
The typical P2P RT pipeline

Input scene
(objects, TX, RX, ...)

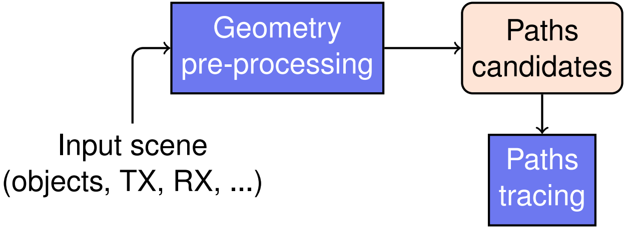
The typical P2P RT pipeline



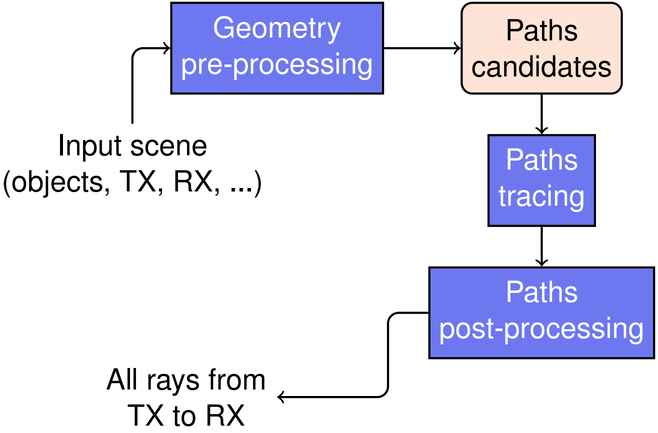
The typical P2P RT pipeline



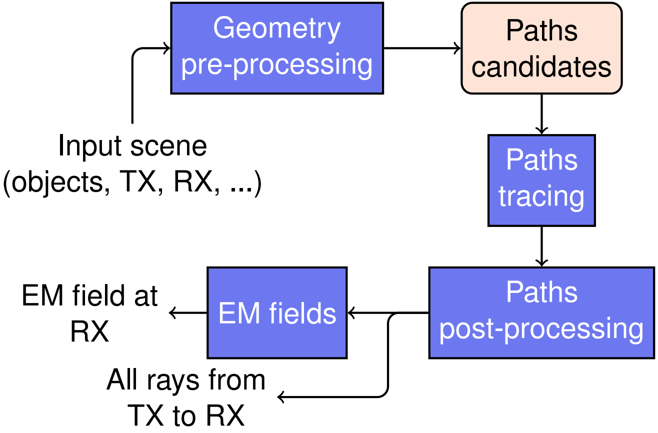
The typical P2P RT pipeline



The typical P2P RT pipeline



The typical P2P RT pipeline



Circumventing the exponential complexity

There are three principal countermeasures to this fundamental problem:

Circumventing the exponential complexity

There are three principal countermeasures to this fundamental problem:

1. resort to RL;

Circumventing the exponential complexity

There are three principal countermeasures to this fundamental problem:

1. resort to RL;
2. use a heuristic approach to reduce the number of path candidates (e.g., the *Fibonacci* method used in SionnaRT);

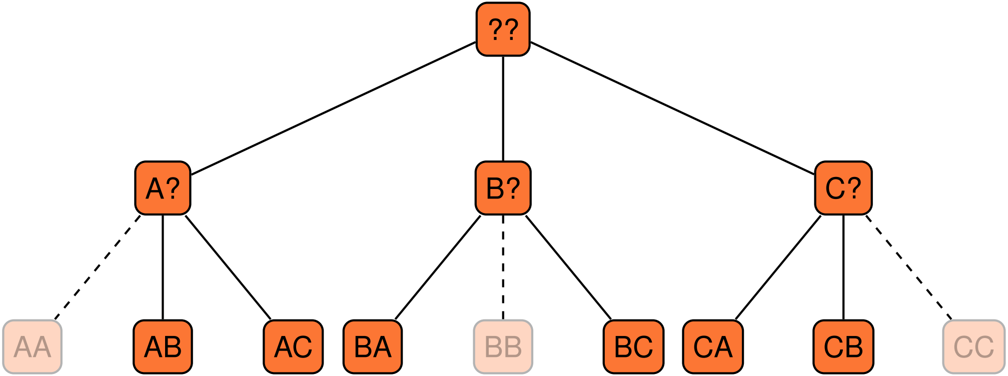
Circumventing the exponential complexity

There are three principal countermeasures to this fundamental problem:

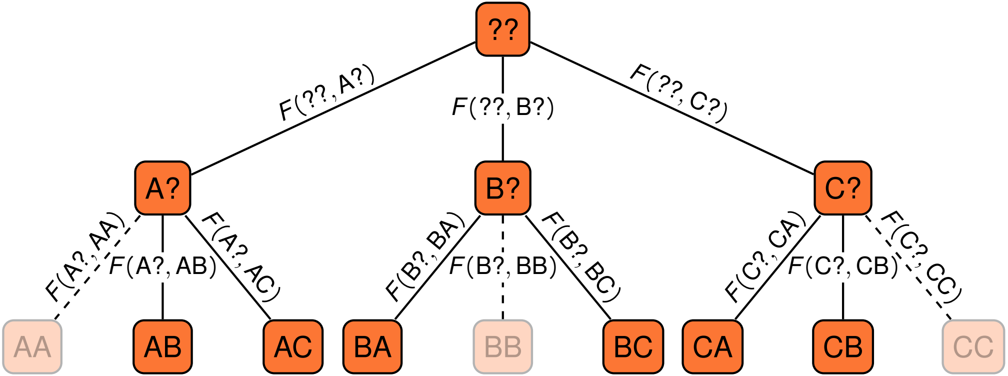
1. resort to RL;
2. use a heuristic approach to reduce the number of path candidates (e.g., the *Fibonacci* method used in SionnaRT);
3. pre-compute some kind of visibility tree.

Table of contents

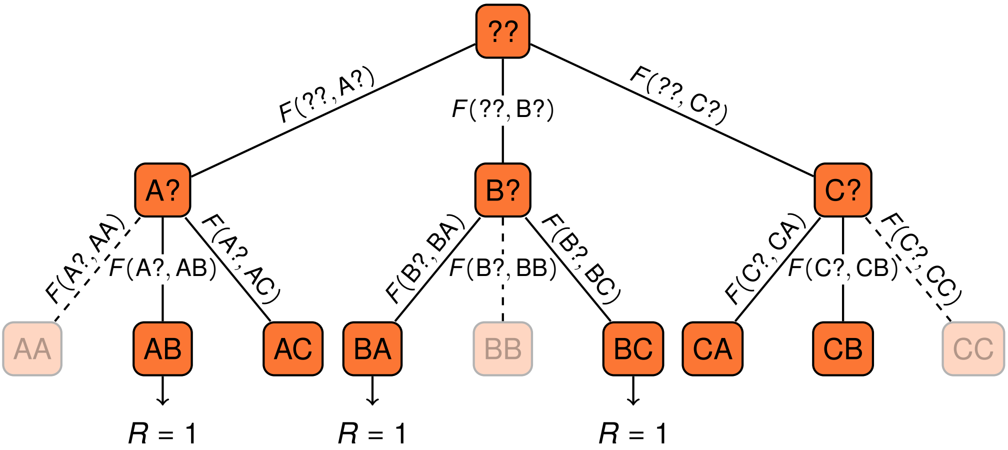
How path candidates are generated



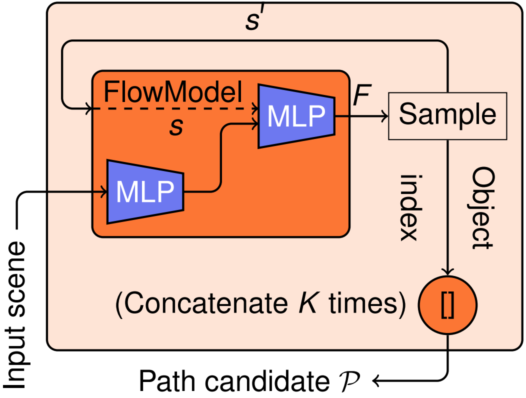
A GFlowNet-like architecture



A GFlowNet-like architecture



A GFlowNet-like architecture



A GFlowNet-like architecture

Our model:

A GFlowNet-like architecture

Our model:

1. does **not** learn a specific scene;

A GFlowNet-like architecture

Our model:

1. does **not** learn a specific scene;
2. can accommodate scenes of **arbitrary size**;

A GFlowNet-like architecture

Our model:

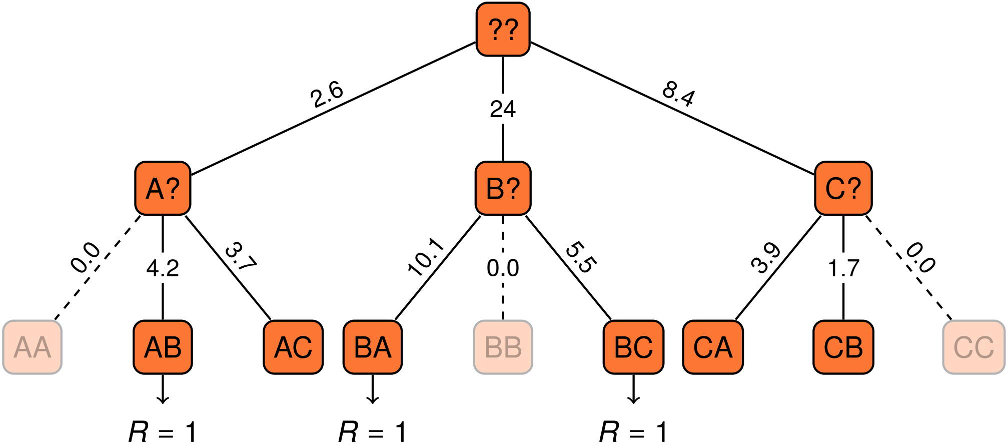
1. does **not** learn a specific scene;
2. can accommodate scenes of **arbitrary size**;
3. generates **one** path candidate at a time;

A GFlowNet-like architecture

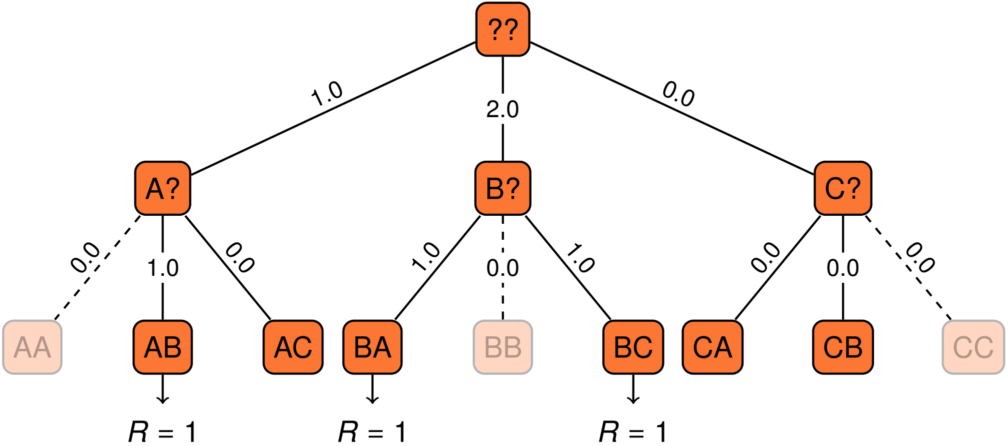
Our model:

1. does **not** learn a specific scene;
2. can accommodate scenes of **arbitrary size**;
3. generates **one** path candidate at a time;
4. and **aims at accelerating RT**, not replacing it.

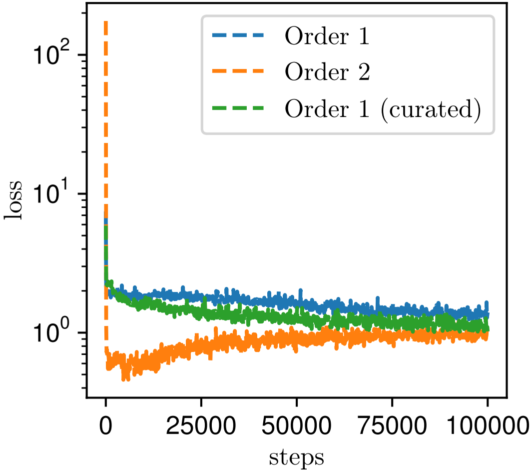
How we train our model - before



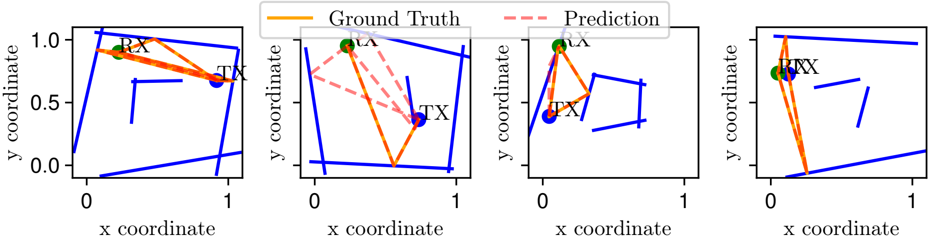
How we train our model - after



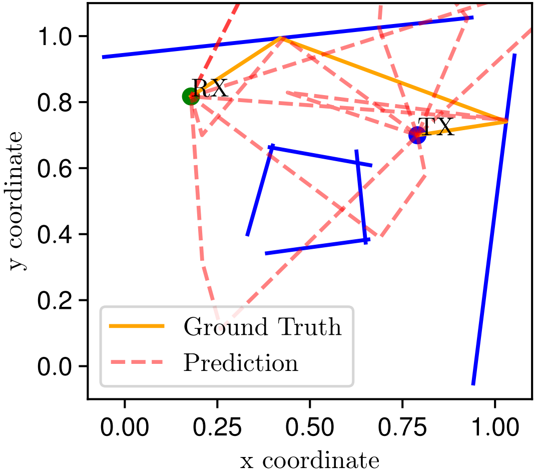
Learning reflection paths in 2D scenes



Predicting 1st order path candidates



Predicting 2nd order path candidates



Main contributions and future work

Our model offers:

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;
- promising results on 1st order path (esp. on generalization over scenes);

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;
- promising results on 1st order path (esp. on generalization over scenes);
- and a controllable complexity (like RL).

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;
- promising results on 1st order path (esp. on generalization over scenes);
- and a controllable complexity (like RL).

Several avenues for improvement:

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;
- promising results on 1st order path (esp. on generalization over scenes);
- and a controllable complexity (like RL).

Several avenues for improvement:

- the study of sparse reward functions, or smoothed ones using our recent work (EuCAP 2024);

Main contributions and future work

Our model offers:

- a novel Machine Learning approach to RT;
- promising results on 1st order path (esp. on generalization over scenes);
- and a controllable complexity (like RL).

Several avenues for improvement:

- the study of sparse reward functions, or smoothed ones using our recent work (EuCAP 2024);
- and the general shaping of the model (layers size, etc.).

Thanks

Any questions?

GFlowNet constraints

To achieve this, the model must be trained to respect the following fundamental properties:

1. Each edge in the search graph must be assigned a positive *flow*, $F(s, s') > 0$, where s is the parent state and s' is the child state;
2. Flow conservation between ingoing and outgoing edges must be ensured:

$$\forall s', F(s, s') = R(s') + \sum_{s''} F(s', s''), \tag{1}$$

   that is, the sum of output flows, $F(s', s'')$, must be equal to the input flow, $F(s, s')$, minus the reward;

GFlowNet constraints (continued)

3. The probability of choosing state s' given state s must be defined as

$$p(s'|s) = \frac{F(s,s')}{\sum_{s''} F(s,s'')}, \tag{2}$$

that is, the probability of traversing an edge in the search graph is equal to its flow value normalized over all outgoing edges.

Loss function

For training our model, we minimize the GFlowNets loss function, which rewrites (1) as a mean squared error:

$$L(s') = \left(F(s, s') - R(s') - \sum_{s''} F(s', s'') \right)^2 . \tag{3}$$