SYMPAIS: Symbolic Adaptive Importance Sampling for Probabilistic Program Analysis

Imperial College London

UNIVERSITY OF OXFORD

Yicheng Luo, Antonio Filieri, Yuan Zhou

Imperial College London, University of Oxford

arXiv:2010.05050

Background

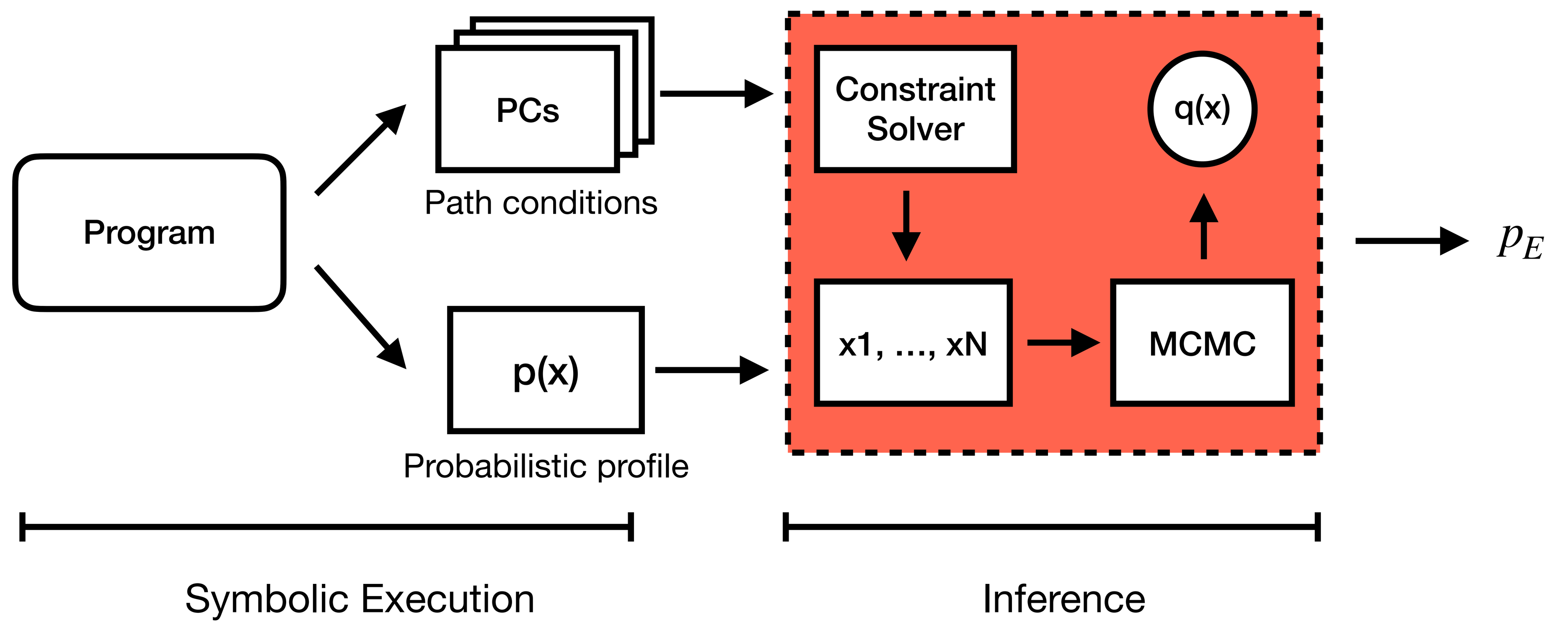

Figure: Overview of the probabilistic software analysis pipeline

- Compositional probabilistic quantification by analyzing symbolic path constraints (PC).
- individual constraints analyzed by Monte-Carlo based approaches.
- Require high-precision answers for rare events in mission-critical software; direct Monte Carlo not sample-efficient for rare events; stratified sampling (qCoral) inefficient in high dimensions.

Listing 1 Safety controller for a flying vehicle

```
// Assumptions
altitude = Normal(10000, 100);
headFlap, tailFlap =
  Normal([0, 0],
     [[0.2, 0.1], [0.1, 0.2]]);
// Program
if (altitude <= 9000) { ...
  if (
    Math.sin(headFlap * tailFlap)
      > 0.999
  ) { callSupervisor(); } ...
} else { callSupervisor(); }
```

Methods

- Combine symbolic execution with adaptive importance sampling.
- Use symbolic execution to identify path constraints and feasible initial solutions.
- Run PI-MAIS which uses MCMC to construct adaptive proposal distribution for importance sampling.

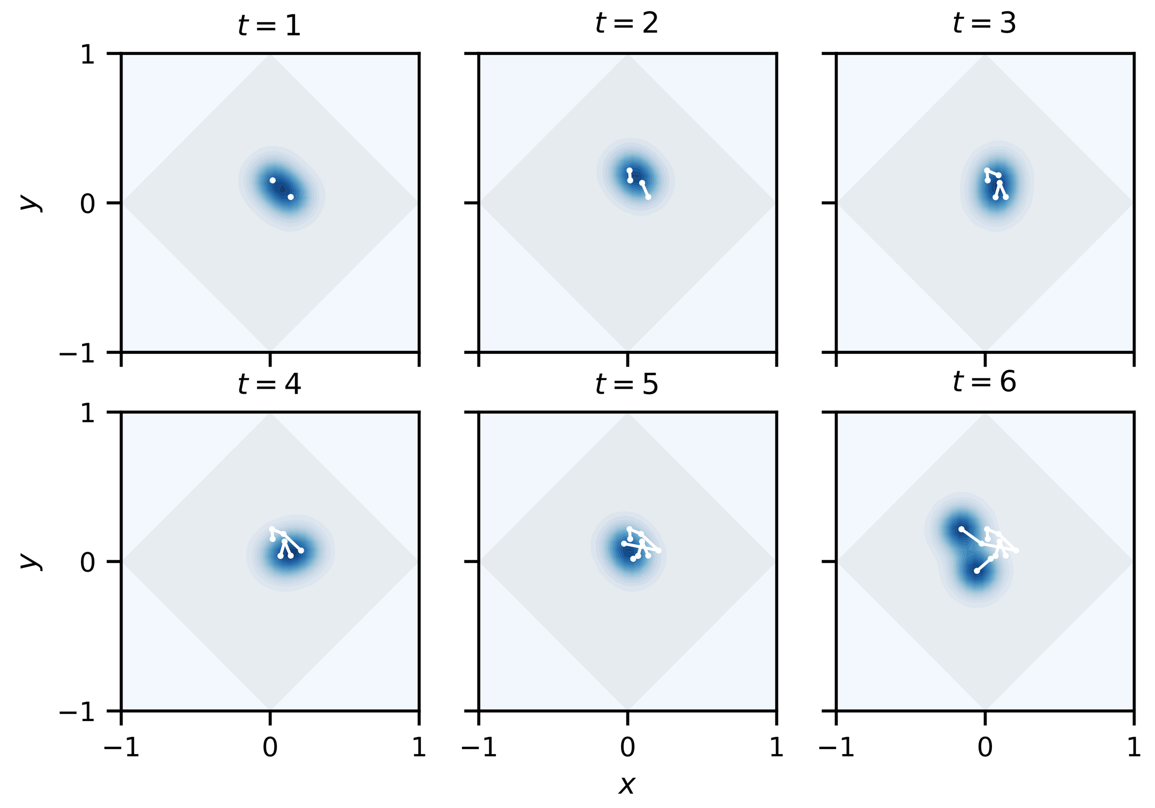

Figure: Graphical illustration of the SYMPAIS algorithm

Results

- Quantify the probability that samples drawn from $p(x) = \mathcal{N}(\mathbf{0}, \mathbf{1})$ are in the interior of a sphere, i.e., $\|x - \mathbf{1}\|^2 \leq 1$.
- SYMPAIS achieves similar performance for $d \geq 5$ and is orders-of-magnitude more efficient for $d \geq 8$.

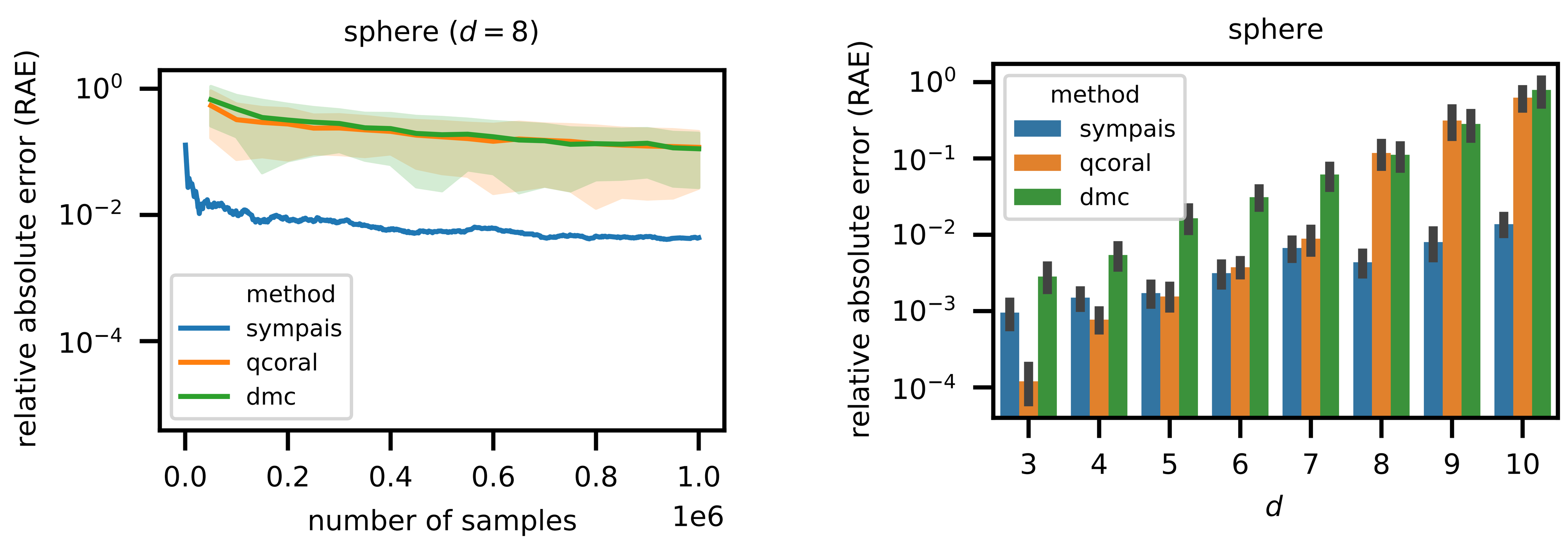

Figure: Performance comparison between SYMPAIS and other algorithms on the sphere task.

References

- M. Borges, A. Filieri, M. d'Amorim, C. S. Păsăreanu, and W. Visser, "Compositional solution space quantification for probabilistic software analysis," PLDI '14, pp. 123–132, 2014.
- L. Martino, V. Elvira, D. Luengo, and J. Corander, "Layered adaptive importance sampling," *Statistics and Computing*, vol. 27, no. 3, pp. 599–623, 2017.