



NASimJax: A GPU-Accelerated Policy Learning Framework for Penetration Testing

Raphael Simon^{1,2}, José Carrasquel¹, Elli Makdis Antoun¹, Wim Mees¹,
Pieter Libin²

¹Royal Military Academy, Belgium ²Vrije Universiteit Brussel, Belgium

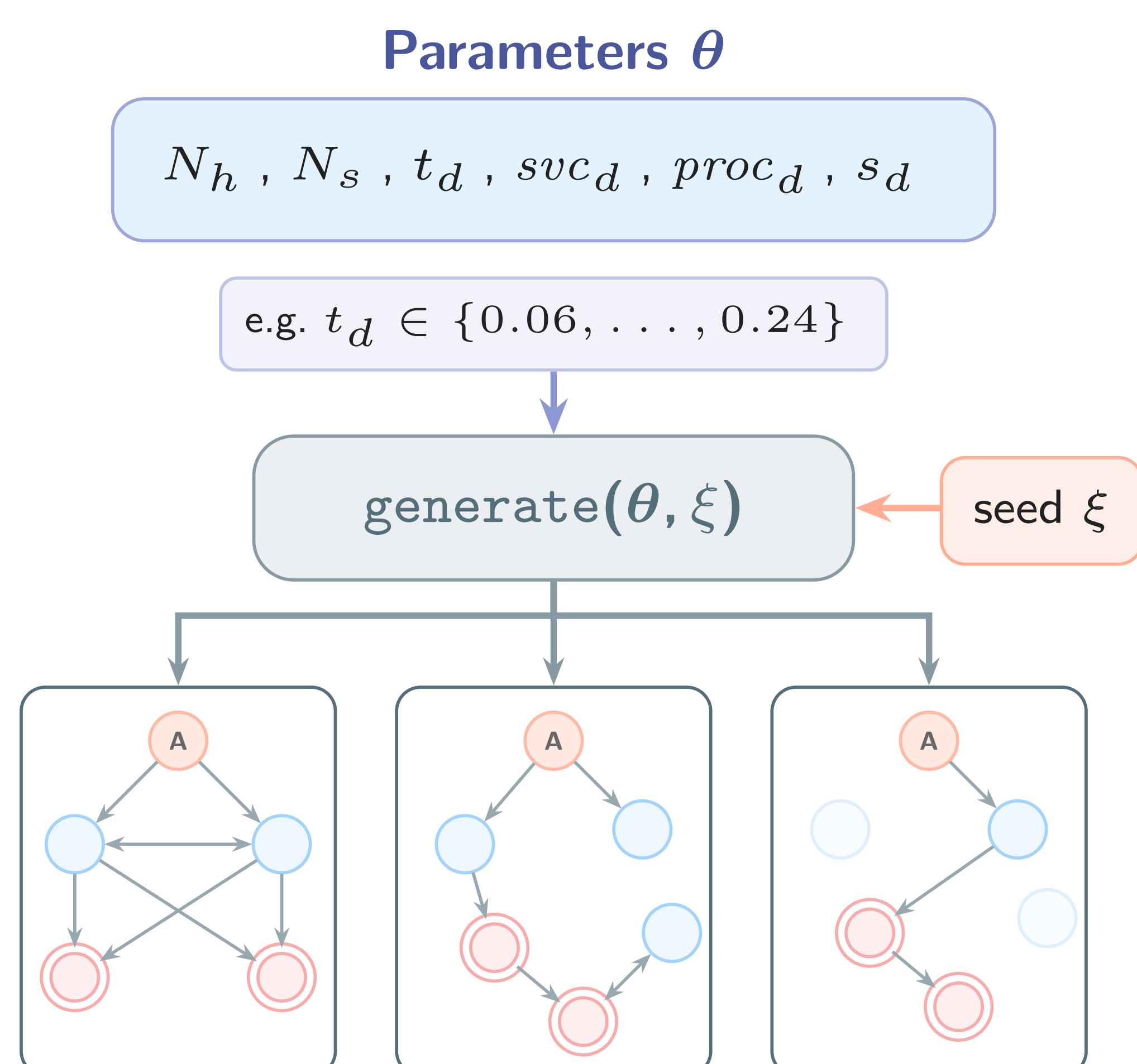


The Problem

RL-based penetration testing has a **generalization problem**: policies trained on narrow scenarios fail to transfer to new networks. Existing simulators don't help. They're **slow** (CPU-bound Python) and **rigid** (fixed or narrowly parameterized scenarios). Together, these make it infeasible to train across the structural diversity of real networks.

NASimJax

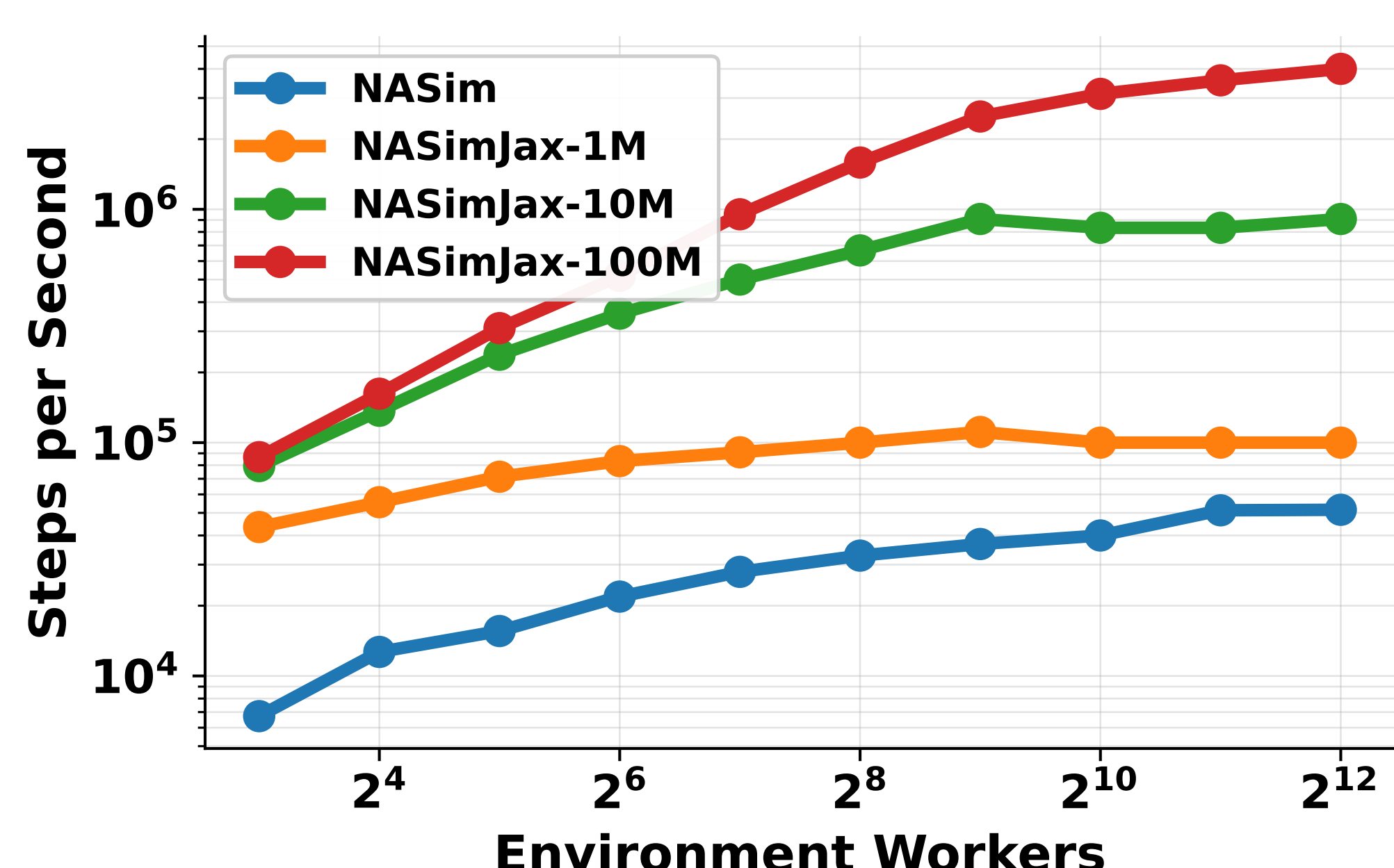
A **JAX-native research framework** that formulates penetration testing as a **Contextual POMDP** over procedurally generated networks. Each episode is conditioned on a network instance sampled from a configurable distribution, turning a fixed-scenario problem into a distributional one.



Key design choices:

- ▶ Parametric generator: topology density t_d , host count N_h , number of subnets N_s , vulnerability density ($svc_d, proc_d$), sensitive density s_d
- ▶ We guarantee **every generated network is solvable**
- ▶ Random topologies and variable active-host counts create an implicit difficulty curriculum

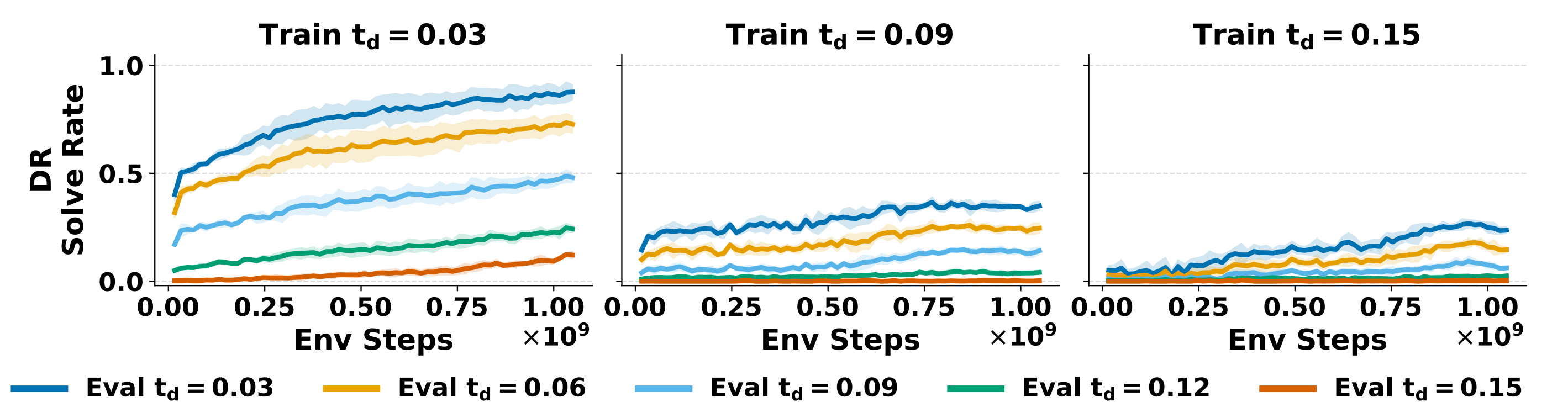
Up to 80× Faster Than Prior Simulators



Case Study: Generalization

We use NASimJax to answer a question prior simulators couldn't: **how does the training distribution shape zero-shot transfer?**

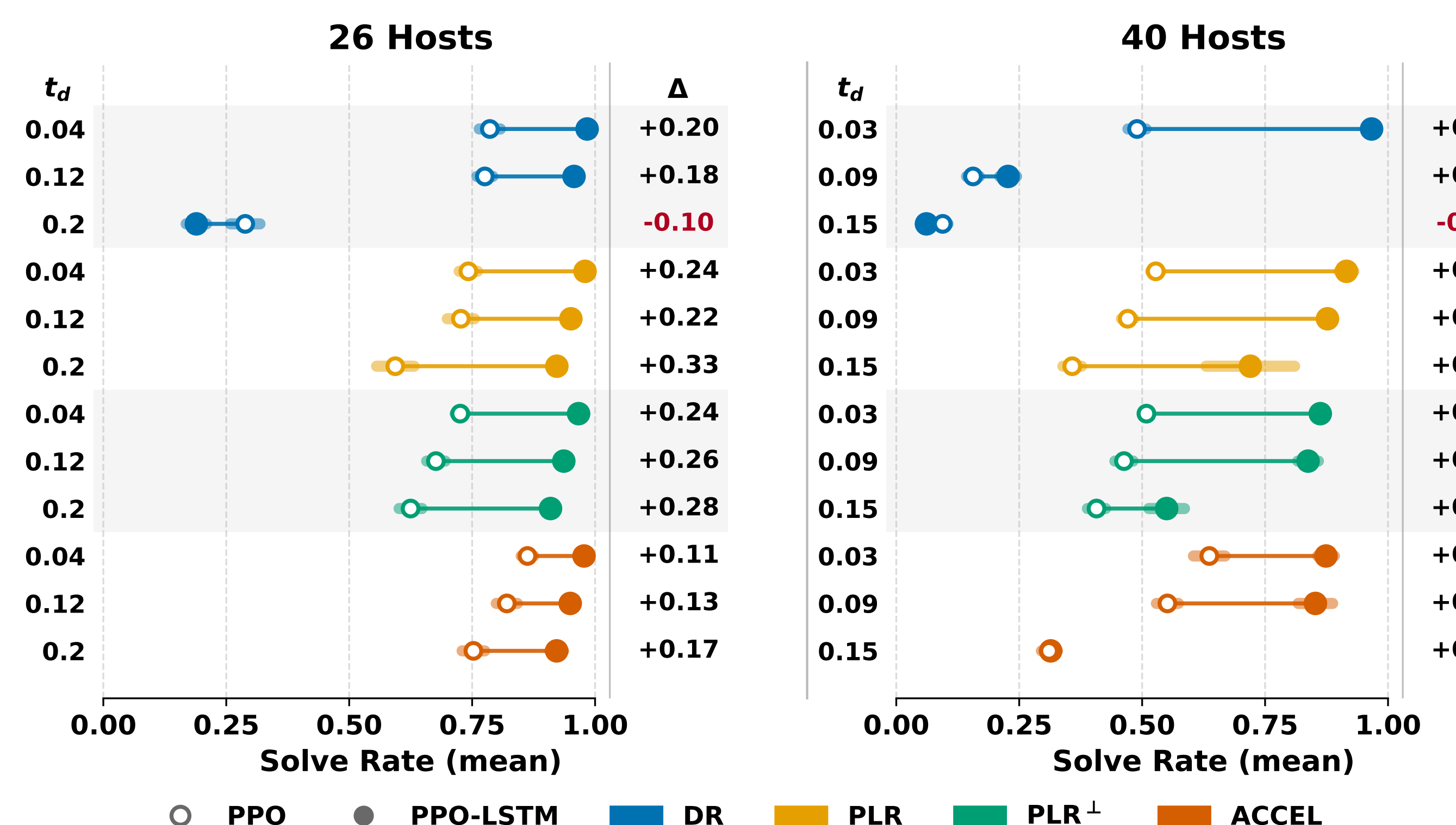
Train sparse, generalize denser



Finding: training on **sparser** topologies produces the best overall generalization. This holds even when evaluated on denser networks than seen during training. Sparse topologies expose the agent to shorter-horizon tasks early, building competence before encountering complex attack paths → **an implicit curriculum**.

Practical implication: train on computationally cheap and sparse networks, deploy on dense ones.

The effect is robust across algorithms



Finding: the curriculum effect holds across Unsupervised Environment Design methods. PLR, PLR⁺ and ACCEL leverage replay to construct an explicit version of the same effect, remaining competitive when trained at high t_d where DR collapses.

Future Directions

Beyond the findings shown here, NASimJax supports:

- ▶ Systematic study of **action-space scaling**
- ▶ **Large network distributional training**
- ▶ Controlled experiments over **topology, density, and vulnerability surface**
- ▶ **Architectural studies** (permutation invariance, recurrent policies, etc.)

NASimJax is configurable, guaranteed-solvable, and fast enough to facilitate large-scale experiments.

