A Multifidelity Sim-to-Real Pipeline for Verifiable and Compositional Reinforcement Learning

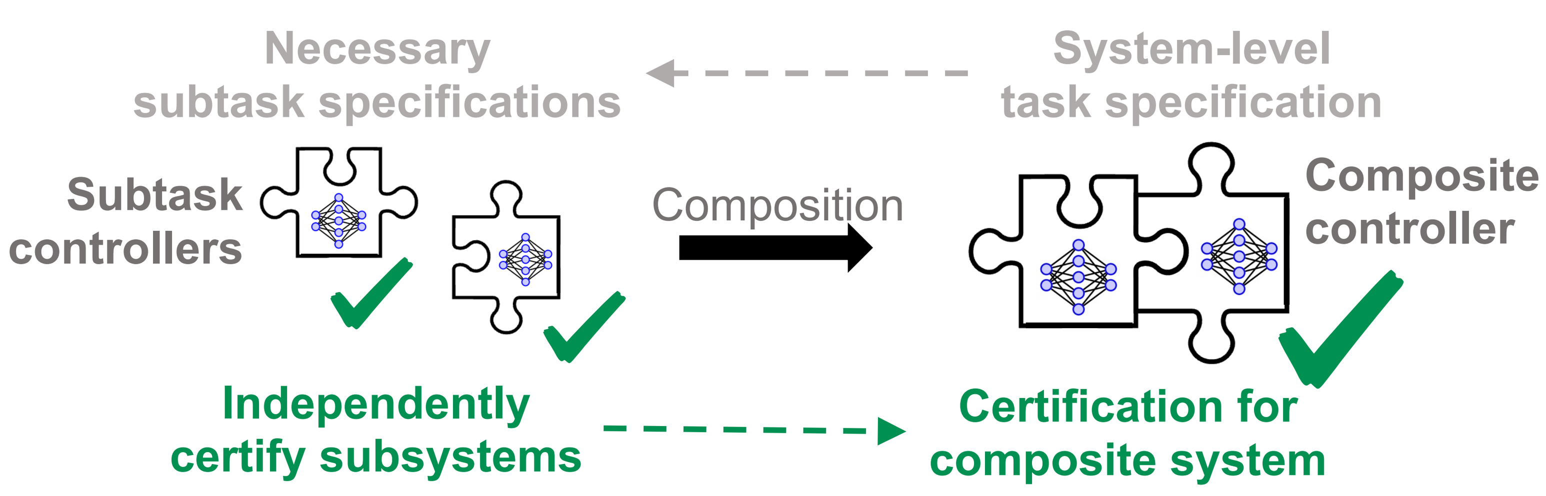
Cyrus Neary[1], Christian Ellis[2], Aryaman Singh Samyal[1], Craig Lennon[3], and Ufuk Topcu[1]

[1] The University of Texas at Austin, [2] The University of Massachusetts Dartmouth, [3] U.S. Army Research Lab

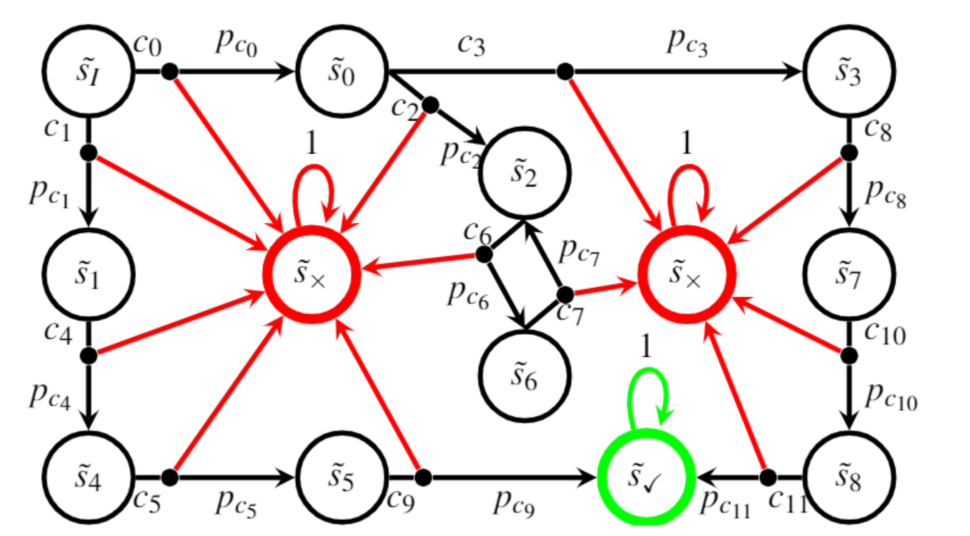
The central question.
How can we deploy **learning-based** control policies on **real-world hardware** whose behaviors can be **modeled and empirically verified against specifications**?

A summary of the approach: A framework to **compose** and **model the outcomes** of RL-based systems.

1. Construct a hierarchical ***high-level model*** to decompose tasks into subtasks.
2. Train and test subtask policies in a multifidelity sim-to-real pipeline.
3. Iteratively refine the high-level model.



The high-level model



States defined by subtask entrance and exit conditions.

Actions represent subtasks.

Transition probabilities represent likelihood of subtask success.

Failure states represent failure of any subtask.

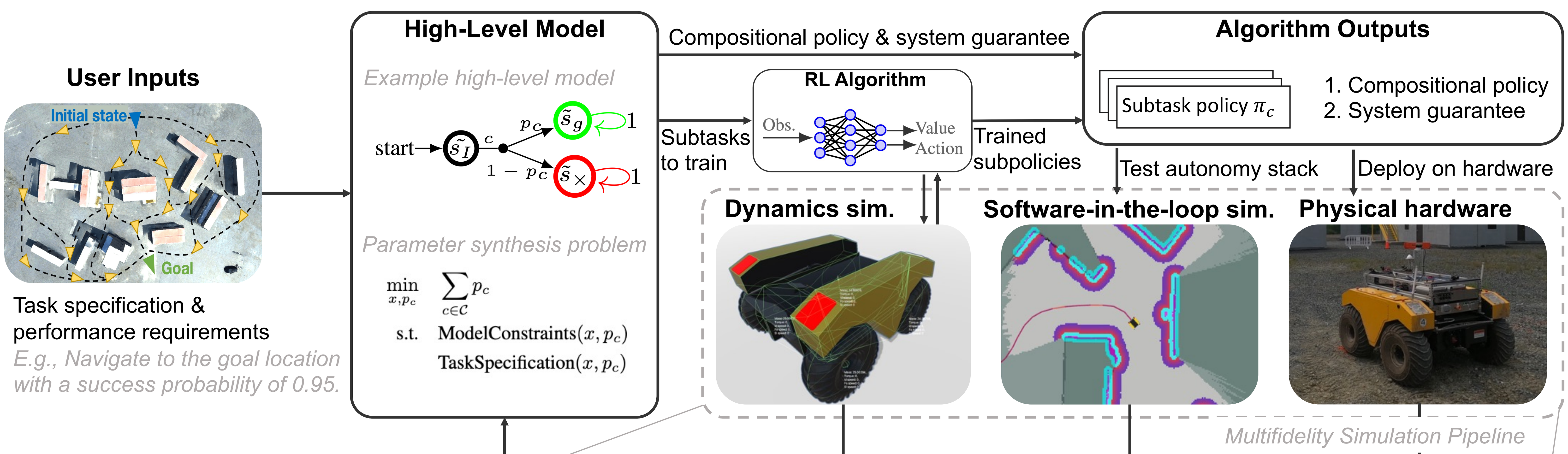
Success states represent overall task success.

Relating the high-level model to the environment

If [HLM Transition Probability p_c] $\leq$ [Success probability of subsystem c], $\forall c$

Then $Prob$ {Reaching $\tilde{s}_\checkmark$ in the HLM} $\leq$ $Prob$ {Compositional system completes its task in training environment.}

Iterative and compositional reinforcement learning within a multifidelity sim-to-real pipeline



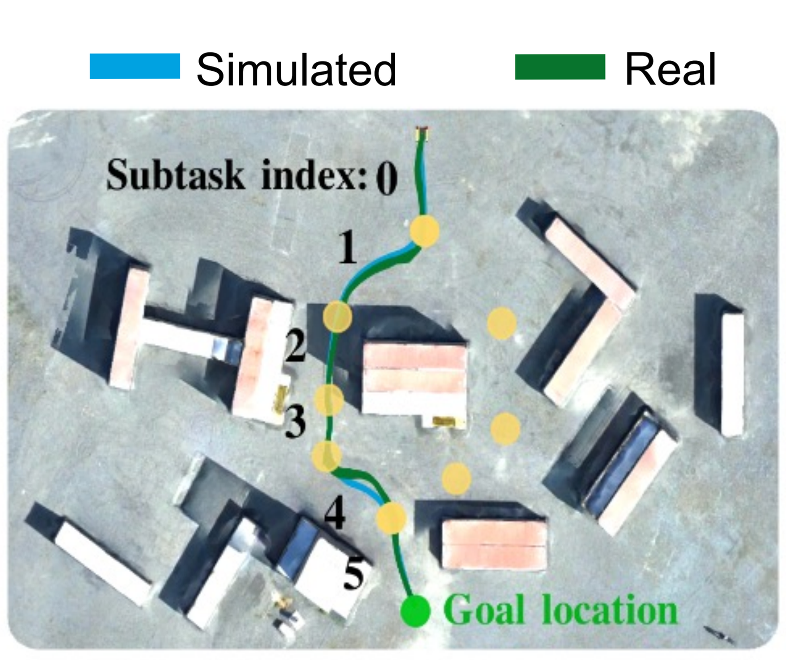
The HLM parameter synthesis problem corresponds to the **automatic decomposition** of task specifications into subtask specifications.

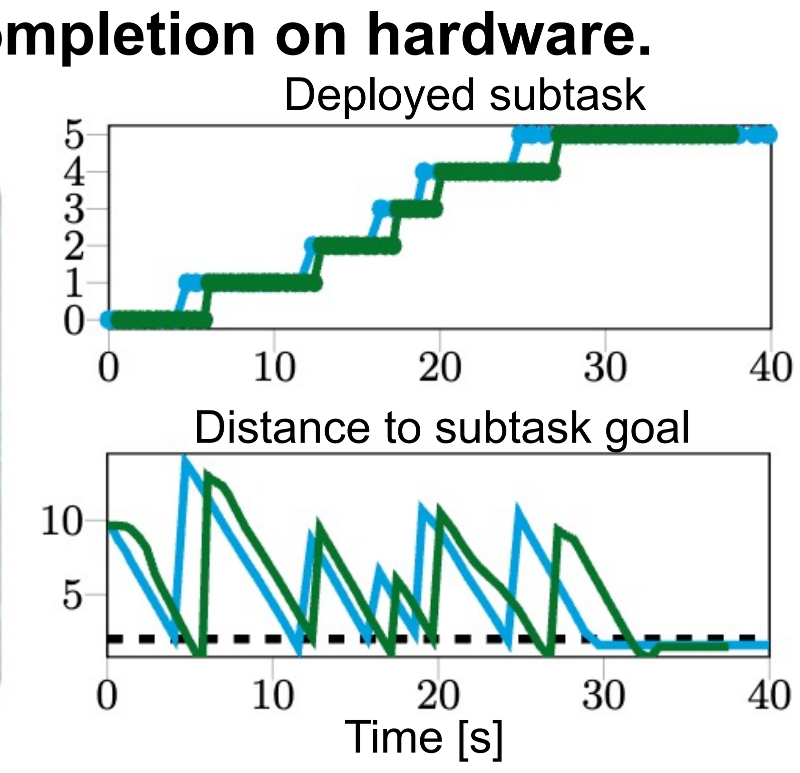
Dynamics sim.
- Physics-only simulation built in Unity.
- Fast simulation suitable for RL training.

Software-in-the-loop sim.
- Simulate policy interaction with ARL autonomy stack.
- Real-time simulation limits scale of data collection.

Physical hardware
- Policy maps observations directly to velocities.
- Limited opportunity for evaluations on hardware.

Compositional RL systems trained in simulation lead to successful task completion on hardware.



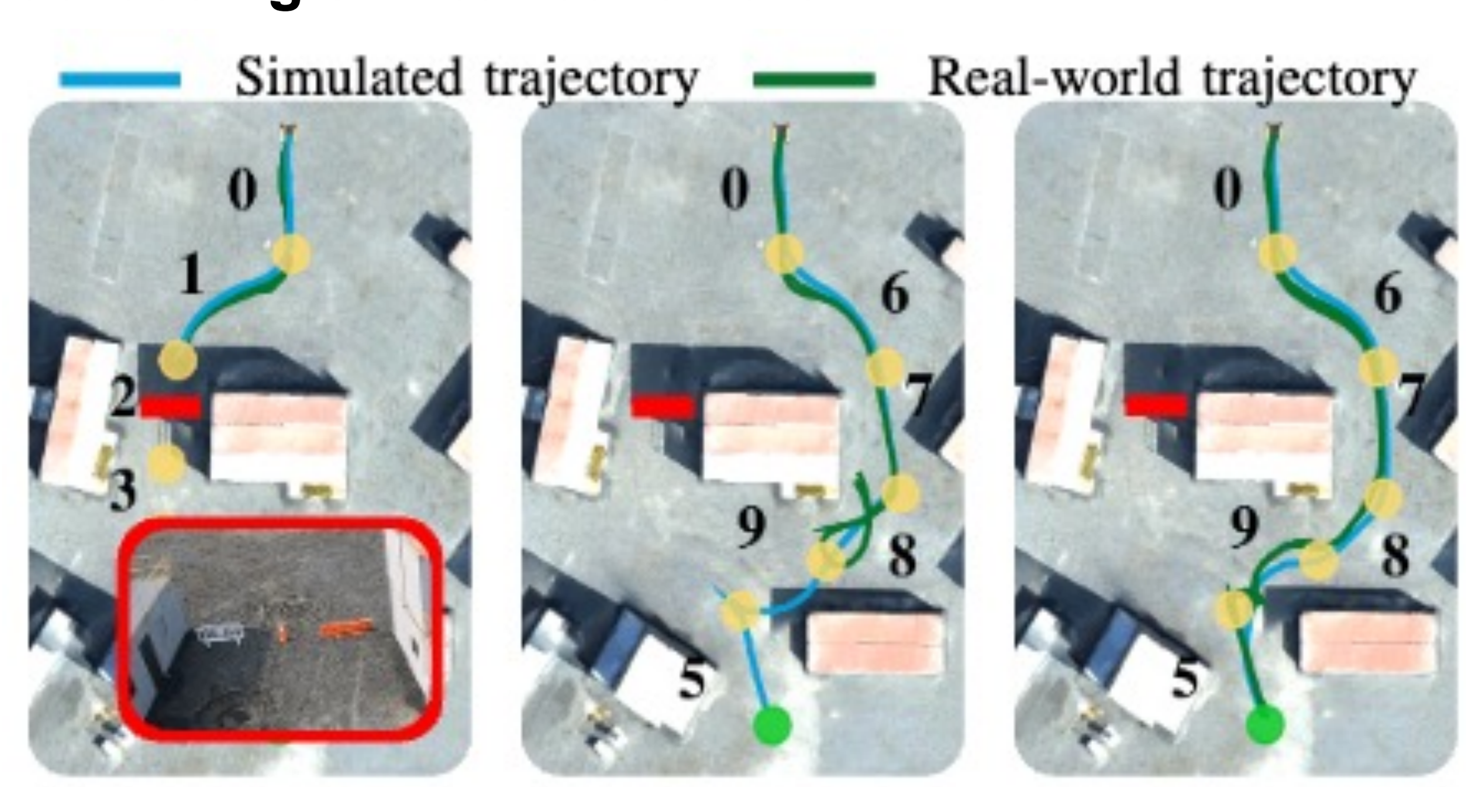


The framework automatically selects underperforming subtask policies for further training.

Subtask Success Probabilities

Subtask	0	1	2	3	4	5
Required	1.0	0.98	1.0	1.0	0.95	0.97
Estimated	1.0	0.98	1.0	1.0	0.90	0.97

The framework **automatically adapts to environment changes** and it simplifies the process of **targeting and addressing sim-to-real errors.**



An obstacle blocks the previously planned path. | Automatic replanning initially fails subtask 8 on physical hardware. | Targeted re-training of specific subtasks leads to task success.

Presentation session: AI-Enabled Robotics 1. NTG6 - 13:30 to 15:00.

Contact: cneary@utexas.edu