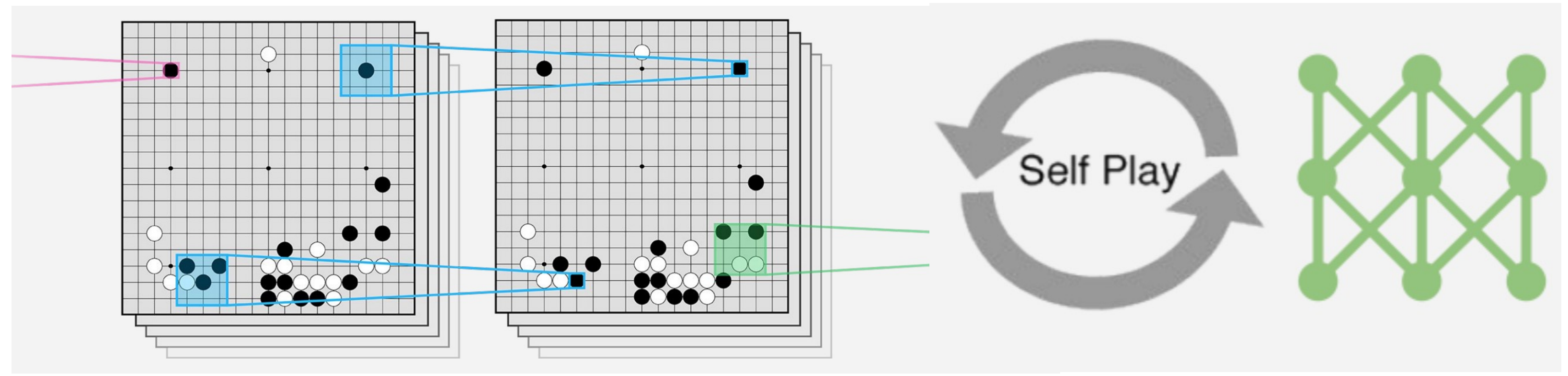
Heterogeneous Adversarial Play in Interactive Environments

Manjie Xu, Xinyi Yang, Jiayu Zhan, Wei Liang, Chi Zhang, Yixin Zhu

Peking University, Beijing Institute of Technology

Background:

Self-play constitutes a fundamental paradigm for autonomous skill acquisition, whereby agents iteratively enhance their capabilities through self-directed environmental exploration:



AlphaGo is trained through supervised learning on human expert games followed by reinforcement learning through self-play.

Core Problem:

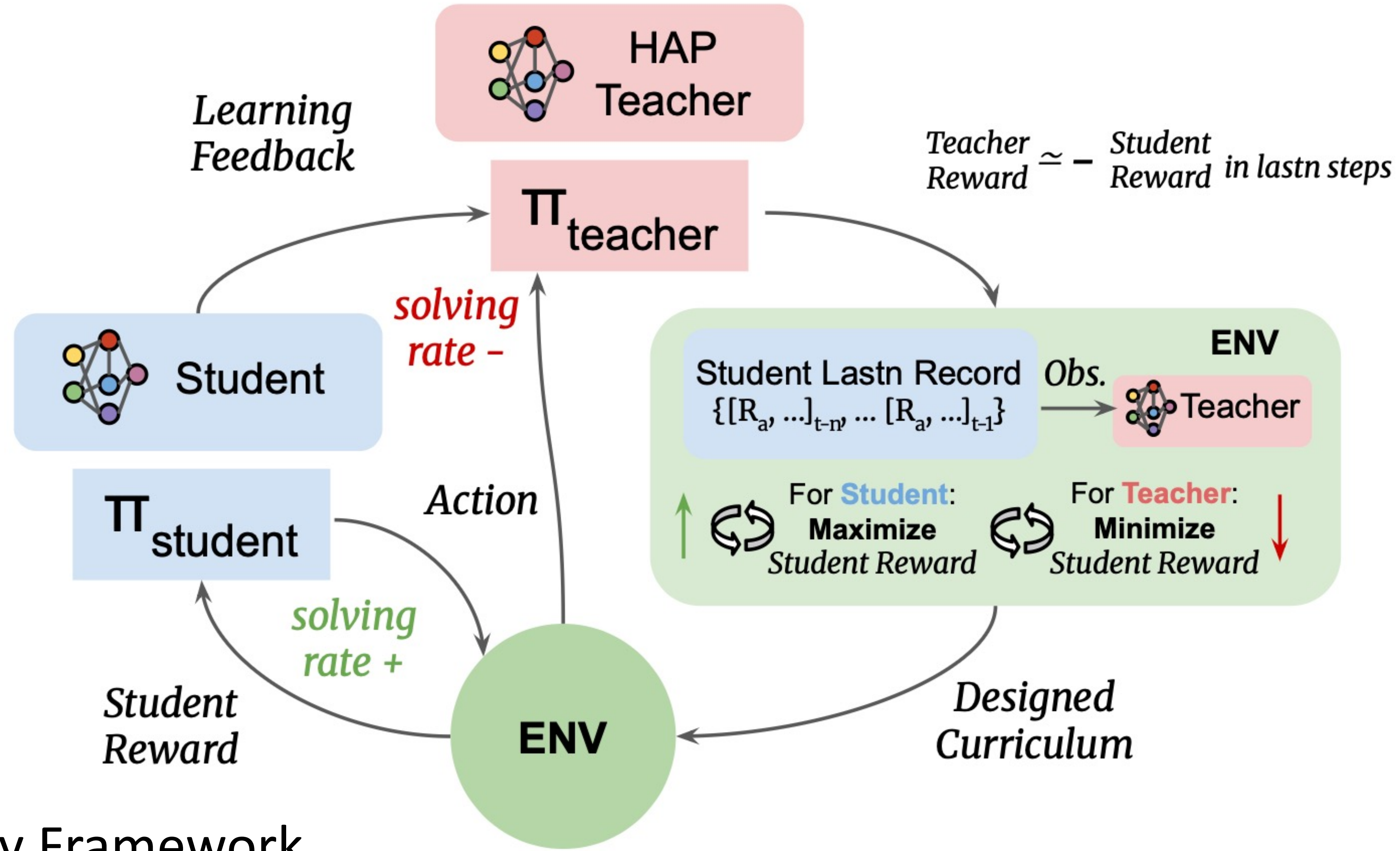
Self-play needs to solve both evaluation and curriculum generation simultaneously while the agent is learning:

- Challenge 1: Position Evaluation
  - How do we know if the current state is good or bad?
  - No clear reward signal until the very end
- Challenge 2: Curriculum Design
  - What tasks should we practice next?
  - Generate experiences actually help learning

Algorithm 1: Heterogeneous Adversarial Play (HAP) Training Loop

```
Data: Initial $\theta$, $\phi$; learning rates $\alpha$, $\beta$
while not converged do
    ;/* 1. Teacher's Adversarial Task Selection:
        Generate task distribution: $p_\phi(C) \propto \exp(\phi)$;
        ;/* Minimization strategy: Sample task $C \sim p_\phi(C)$ to challenge current $\pi$
    ;/* 2. Student's Policy Maximization:
        Execute $\pi(a|s, C; \theta)$, collect trajectory $\tau$;
        Compute reward signal: $R(\tau; C) = \sum_{t=0}^{H} \gamma^t r_t$;
        Update $\theta$ to maximize returns:;
            $\theta \leftarrow \theta + \alpha \nabla_\theta \mathbb{E}_\tau [R(\tau; C)]$;
    ;/* 3. Teacher's Adversarial Update:
        Update $\phi$ to minimize student success:;
            $\phi \leftarrow \phi - \beta \nabla_\phi \mathbb{E}_C [R(\tau; C)]$;
        where $\nabla_\phi J_{\text{teacher}} = -\mathbb{E}_C [\nabla_\phi \log p_\phi(C) \cdot R(\tau; C)]$;
end
```

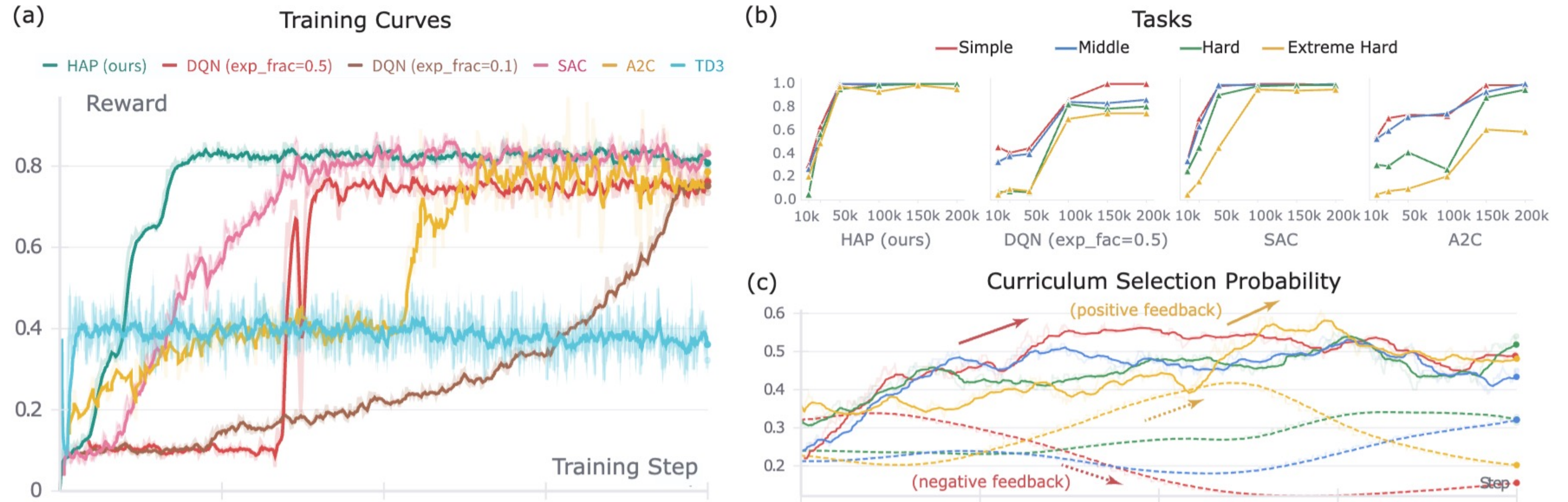
In the proposed HAP framework, the student's goal is to maximize its expected reward by improving its policy, while the teacher tries to minimize this reward by presenting more challenging tasks. Training alternates between the teacher picking tasks, the student attempting them, and both updating their strategies.


HAP Teacher
Learning Feedback
π teacher
Teacher Reward ≃ – Student Reward in lastn steps
solving rate –
Student
ENV
Student Lastn Record
{[R_a, ...]_{t-n}, ... [R_a, ...]_{t-1}}
Obs.
Teacher
Action
π student
For Student: Maximize Student Reward
For Teacher: Minimize Student Reward
solving rate +
Student Reward
ENV
Designed Curriculum

Key Framework

HAP extends automatic curriculum learning through adversarial co-evolution. The teacher learns to generate challenging but solvable tasks that maximize student learning, while the student adapts to solve the teacher's evolving problem proposals.

We discover the advantages of HAP through an intuitive experiment:

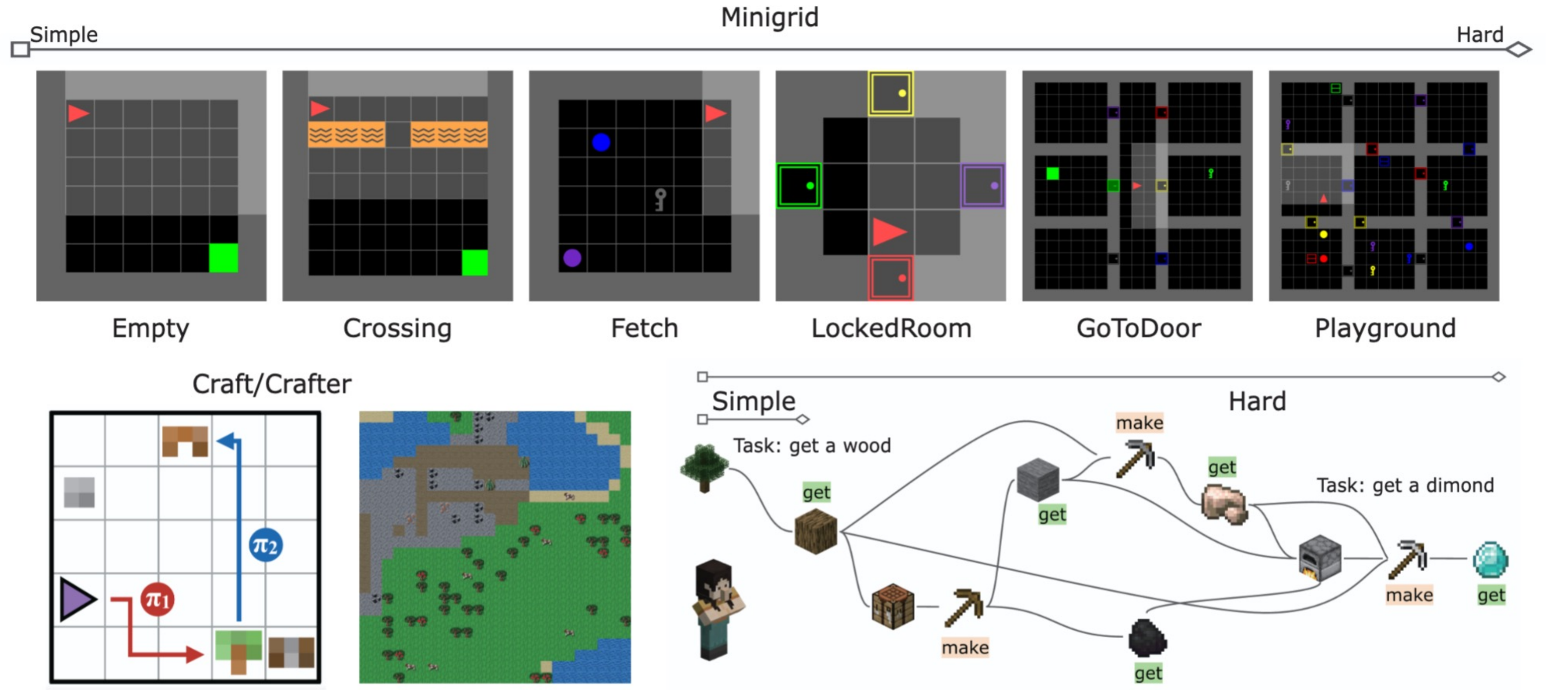


Agents use symbolic maps to reach destinations across four tasks of increasing difficulty. A simple teacher policy samples tasks from a learnable distribution. Results show our method converges faster—within about 35000 steps—and achieves higher overall rewards than all baselines with the same policy network. While other methods eventually learn easy tasks, they struggle on harder ones, likely due to overfitting or forgetting.

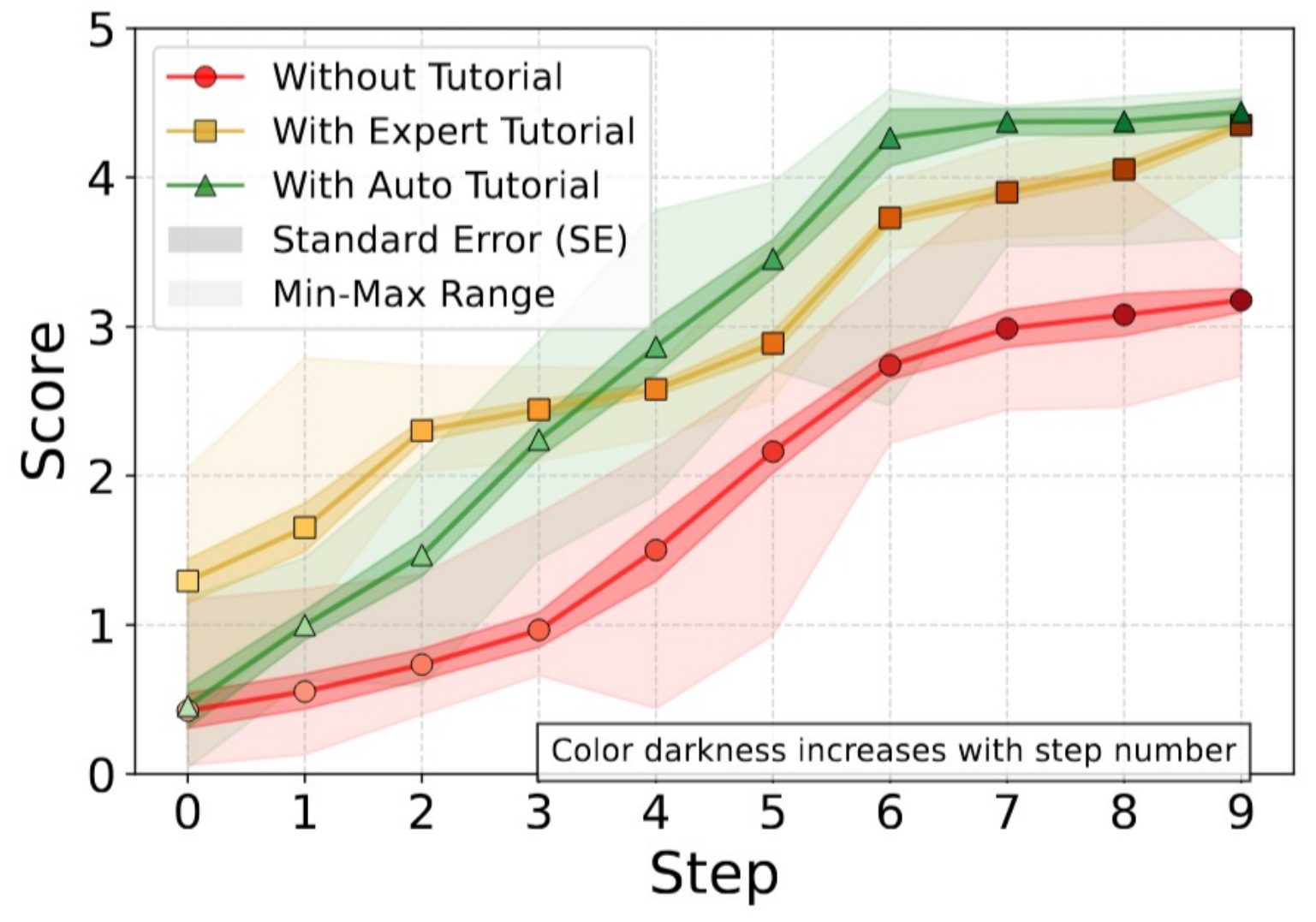
HAP benefits from two key design elements:

- a positive feedback loop, where the teacher increases the sampling probability of a task the learner fails often, thereby accelerating the learner's acquisition in a specific skill
- a negative feedback mechanism, which lowers the sampling probability for tasks the learner has already mastered

Testing in Complex Multi-task Scenes



Env	DQN	A2C	PPO	SAC	TD3	DreamerV3	TSCL	EXP3	HAP	Human
Minigrid										
Easy	**0.98**	0.94	0.88	0.97	0.95	0.96	0.96	0.97	0.92	1.00
Middle	0.24	0.25	0.22	0.27	0.26	0.34	0.21	0.24	**0.46**	0.78
Hard	0.00	0.00	0.00	0.13	0.08	0.18	0.16	0.18	**0.20**	0.46
General	0.407	0.397	0.367	0.457	0.43	0.493	0.443	0.463	**0.527**	0.747
CRAFT										
Easy	0.78	0.84	0.87	0.87	0.86	0.89	**0.94**	0.91	0.88	0.94
Middle	0.26	0.45	0.48	0.42	0.42	0.55	0.24	0.56	**0.63**	0.86
Hard	0.02	0.14	0.12	0.15	0.14	0.27	0.03	0.24	**0.31**	0.66
General	0.278	0.415	0.426	0.413	0.407	0.516	0.307	0.513	0.562	0.802
Crafter										
Easy	0.61	0.79	**0.94**	0.91	0.84	0.91	0.82	0.87	0.91	0.99
Middle	0.28	0.37	0.67	0.47	0.39	0.66	0.45	0.58	**0.68**	0.82
Hard	0.00	0.00	0.47	0.22	0.29	0.52	0.00	0.02	**0.58**	0.74
General	0.297	0.387	0.693	0.533	0.507	0.697	0.423	0.49	**0.723**	0.85



Across these environments, HAP method consistently outperforms other algorithms on most tasks. In our human study, we show that While humans showed greater improvement within a single step when provided with expert step-by-step tutorials, the HAP framework offered more flexible, adaptive curricula tailored to individual behaviors.

Take-away

- In HAP, distinct teacher and student networks co-adapt dynamically, enabling automated curriculum design without predefined task hierarchies or symmetric architectures.
- HAP's emergent curricula mirror human pedagogical strategies while offering personalized adaptation that surpasses fixed, expert-designed teaching approaches.