IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION

2027 SEOUL

Beyond Policy Alignment

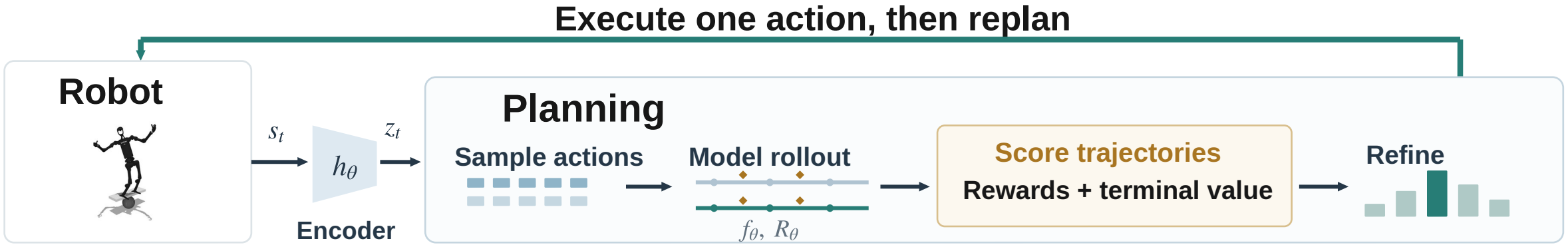
Closing the Planning–Learning Loop for Robot Control with Learned World Models

Anonymous Authors

https://icra27-pl-mpc.github.io/

TD-MPC: planning with a learned world model

Model predictive control: optimize a short action sequence.



World model

Predicts action outcomes

Critic

Estimates future return

Policy

Proposes candidate actions

Execute → collect experience → learn → replan

Studied problem: beyond policy alignment

Policy alignment trains the policy toward planner behavior.

1. **Critic supervision**

   Later rewards reach earlier values through repeated updates.

2. **Terminal-value evaluation**

   Uncertain values can change trajectory ranking.

3. **Planner-to-policy transfer**

   Planner experience varies in quality.

PL-MPC addresses all three interfaces.

Closing the planning–learning loop

PL-MPC · Planning–Learning MPC

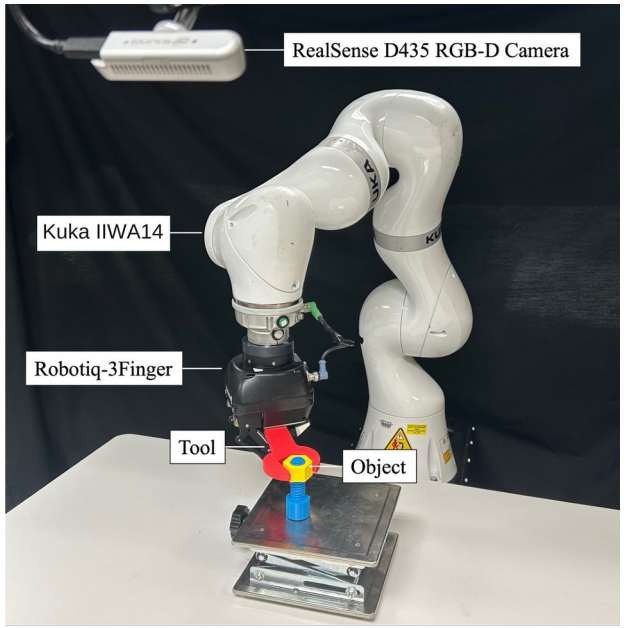
Studied Problem

Policy alignment addresses only one part of the planning–learning loop.

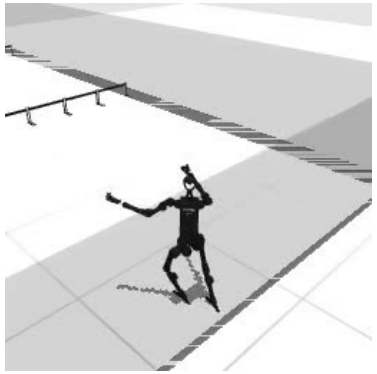
Our Solution

Unchanged world-model architecture and MPPI optimizer.

1. **MTD · Hybrid multi-step TD targets**
   More observed rewards before bootstrapping.
2. **ATE · Adaptive terminal estimates**
   Penalize terminal values using critic disagreement.
3. **RAD · Return-weighted actor distillation**
   Weight imitation by realized episode return.



balance-hard



hurdle

Humanoid control and zero-shot sim-to-real transfer

Result balance-hard: 98 → 387 TAR hurdle: 199 → 466 TAR Robot: 61.3% → 74.2%

vs. TD-M(PC)^2 · TAR: Total Average Return · Robot: observed success on training size (31 trials/method).

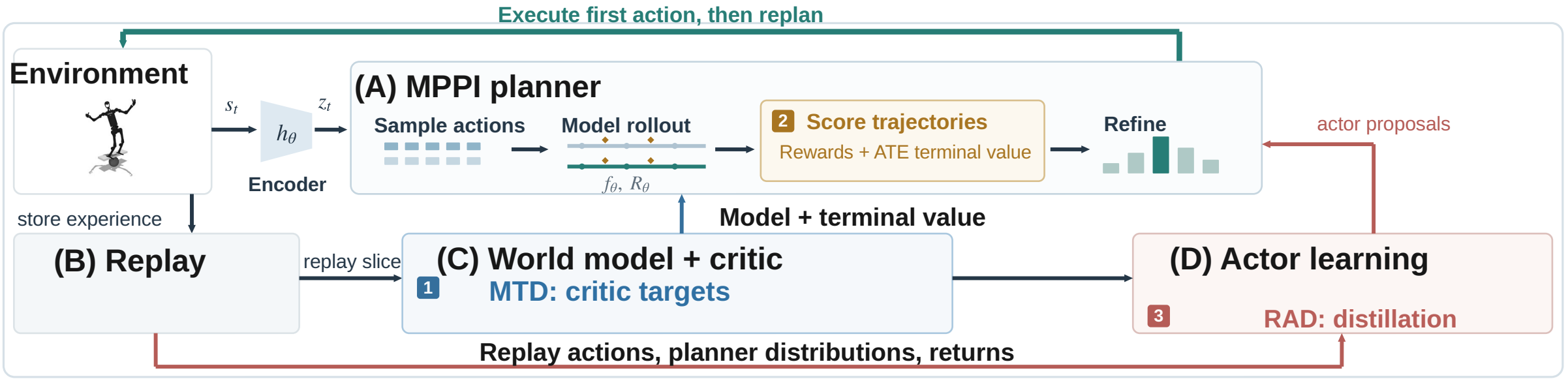
Related work

Policy learning, value learning, and terminal-value evaluation

Method	Policy learning	TD targets	Terminal value
TD-M(PC)²	Planner constraint	One-step TD	Q bootstrap
BMPC	Expert imitation + lazy reanalysis	One-step TD (model-based)	Value bootstrap
BOOM	Q-weighted imitation	One-step TD	Q bootstrap
PO-MPC	KL-regularized RL + adaptive prior	One-step TD	Q bootstrap
PL-MPC (Ours)	**Constraint + episode-return weighting**	**Hybrid multi-step TD (replay + model)**	**Q bootstrap − disagreement penalty**

PL-MPC architecture

Plan → execute → learn → replan



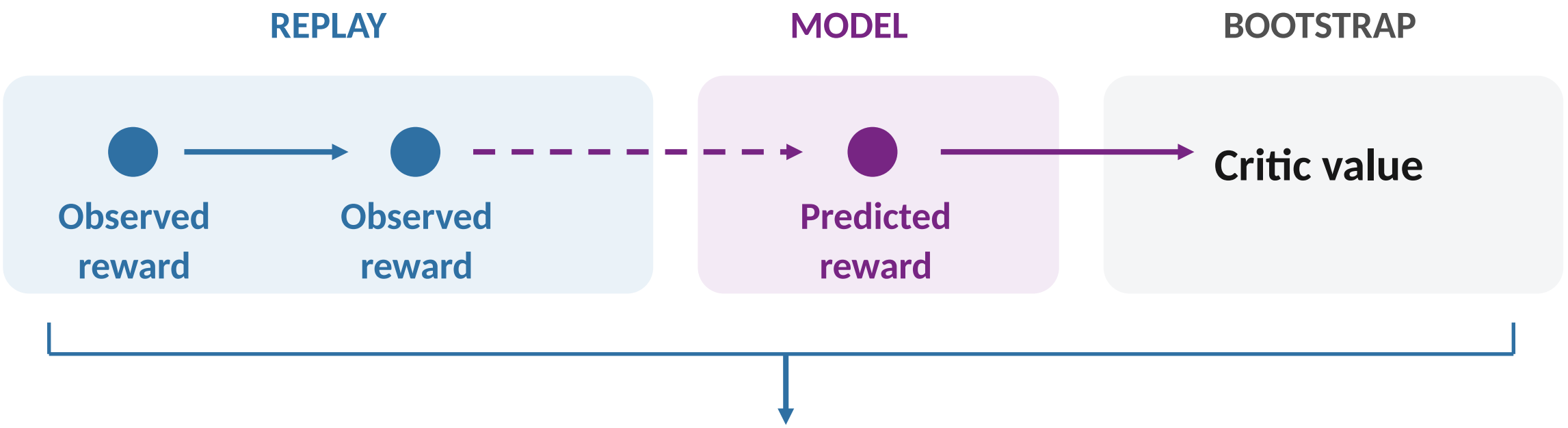
1 Critic supervision

2 Terminal-value evaluation

3 Planner-to-policy transfer

Hybrid multi-step temporal-difference targets

Let later rewards teach earlier decisions.



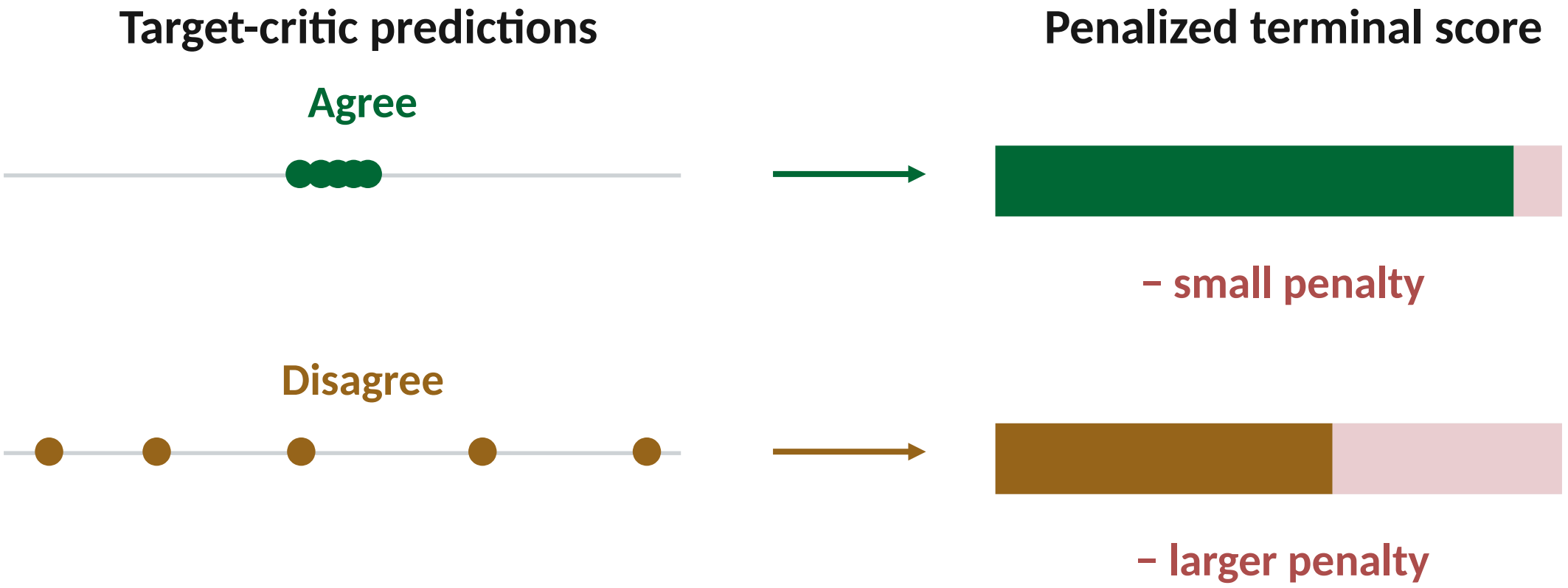
One richer target for an earlier critic update

Use observed rewards where available; predict only the missing tail.

Example: a three-step target near the end of a replay slice. Rewards and bootstrap are discounted.

Adaptive terminal estimates

Be cautious when target critics disagree.

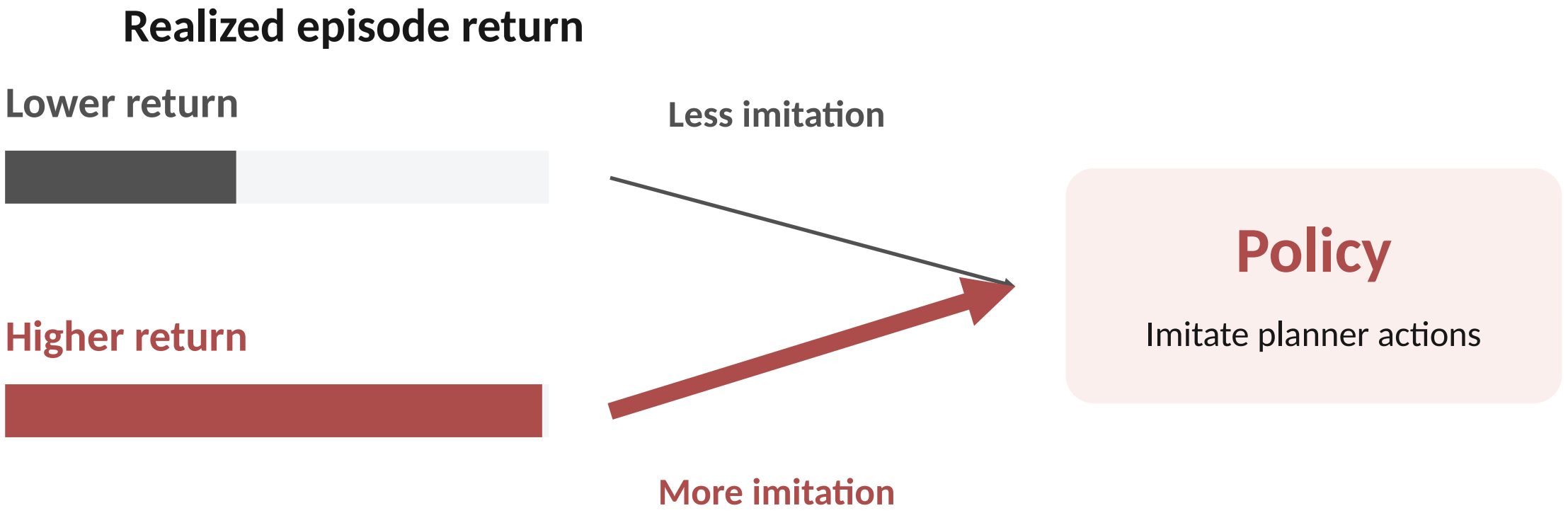


More disagreement → less influence on planning

Conceptual illustration. Disagreement is an uncertainty proxy; only the planner's terminal score is penalized.

Return-weighted actor distillation

Learn more from higher-return episodes.



Fixed warmup → enable return-weighted imitation

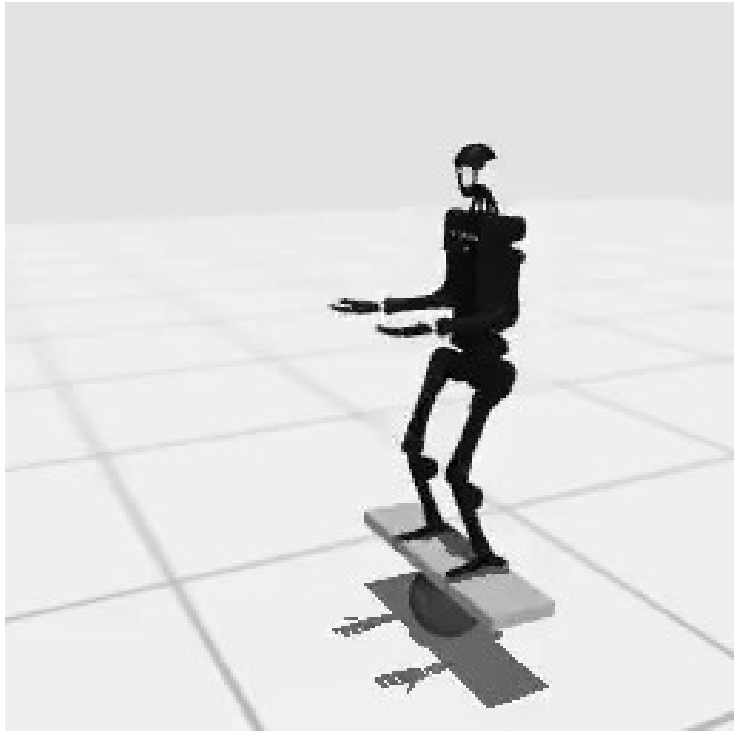
Weights use observed episode returns. Weighted imitation is added to the existing actor objective.

HumanoidBench: balance-hard

5× speed

Maintaining balance for the full episode

PL-MPC

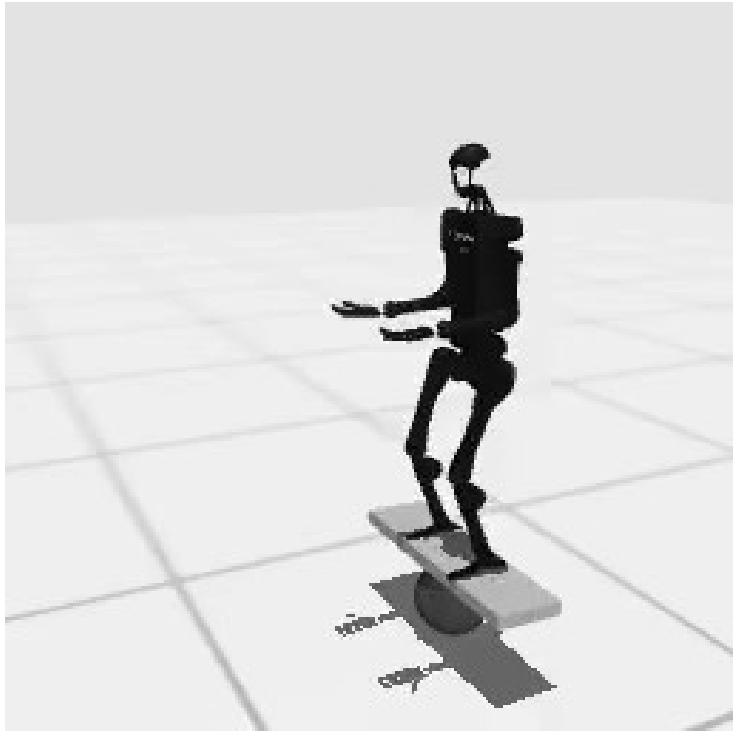


Full episode

Best-seed success: 50.9%

TAR: 387 ± 255

TD-M(PC)²



Falls

Best-seed success: 0%

TAR: 98 ± 18

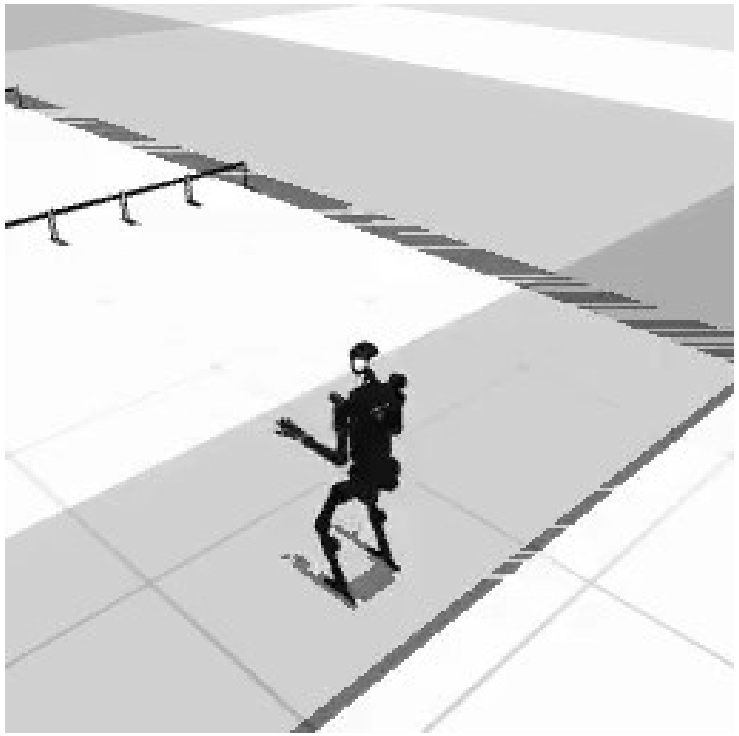
Success: best of 3 seeds · TAR: mean ± SD over 3 seeds · Representative rollouts · Playback 5×

HumanoidBench: hurdle

5× speed

Crossing successive hurdles

PL-MPC

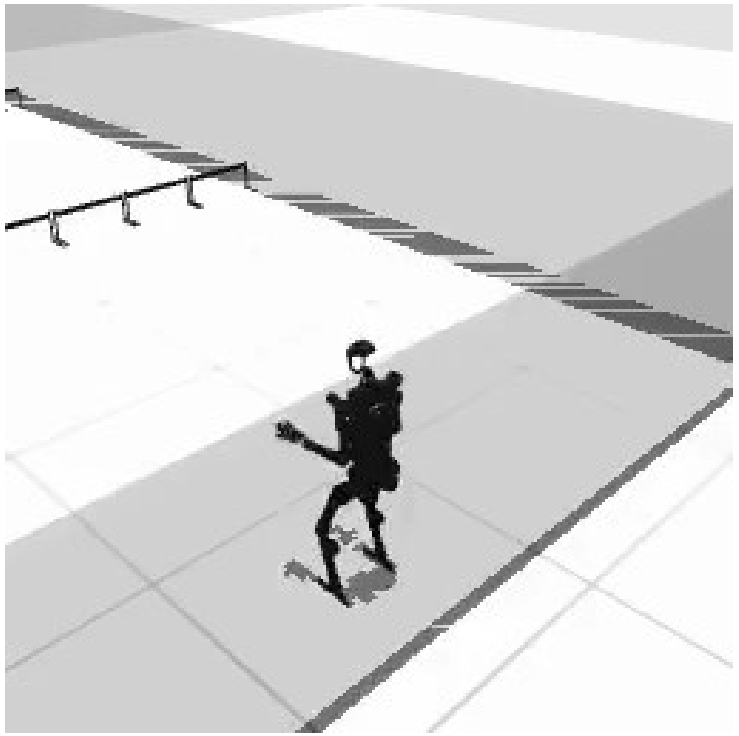


Full episode

Best-seed success: 58.0%

TAR: 466 ± 200

TD-M(PC)²



Falls

Best-seed success: 0%

TAR: 199 ± 13

Success: best of 3 seeds · TAR: mean ± SD over 3 seeds · Representative rollouts · Playback 5×

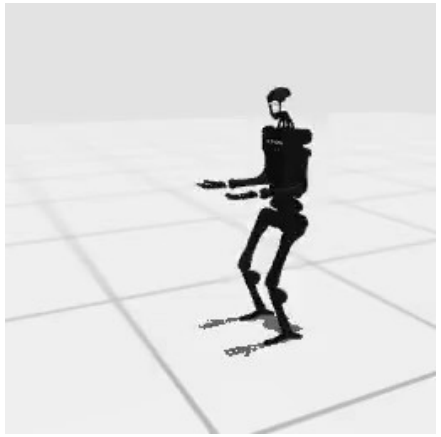
Other HumanoidBench tasks

7× speed

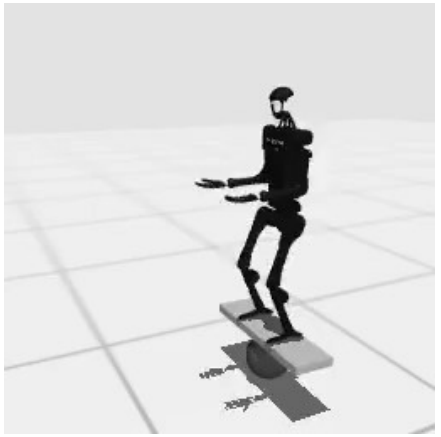
Competent performance on other tasks

run

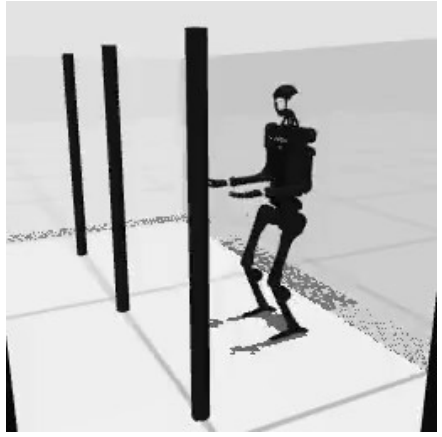
crawl



balance-simple

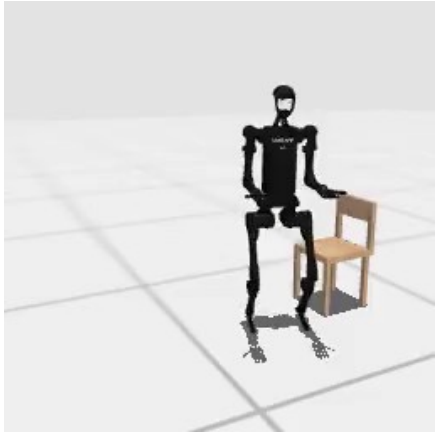


pole



slide

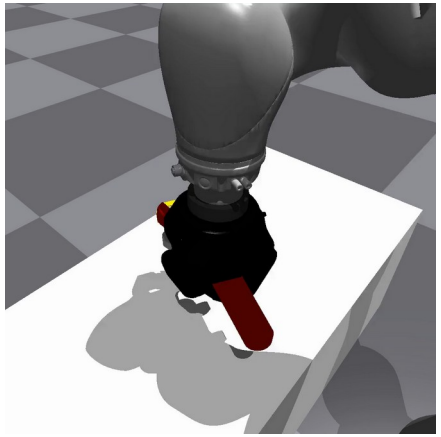
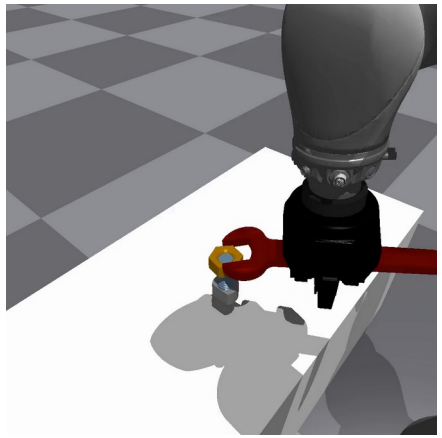
sit-hard



Wrench–nut alignment in simulation

3× speed

Isaac Gym · Training size 5 · Three starting configurations

	Scene 1	Scene 3	Scene 4
PL-MPC	 Success	 Success	 Success
TD-M(PC)²	 Timeout	 Timeout	 Success

Selected scene pairs from the project page · Labels describe these clips · Playback 3×

Zero-shot wrench–nut alignment

4× speed

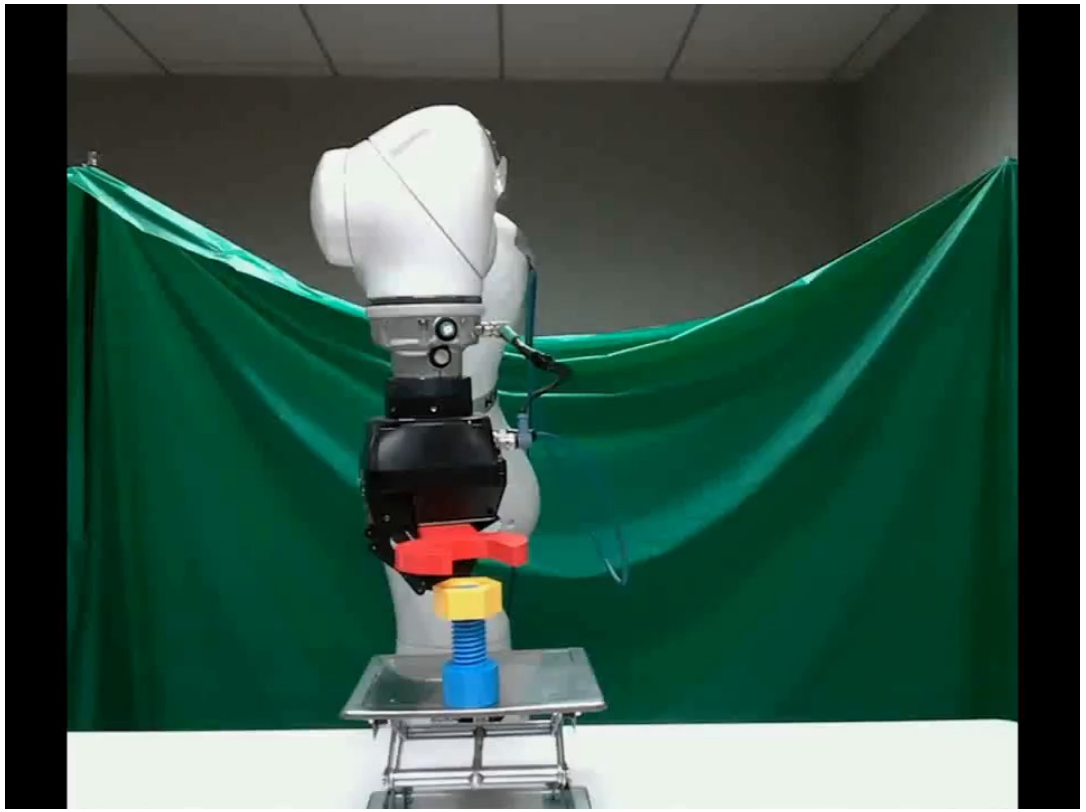
Simulation-trained world model · Online planning on the real robot

PL-MPC

Success

74.2% · 23/31

TD-M(PC)²



Timeout

61.3% · 19/31

Training size · Observed success, 31 trials/method · Representative trials · Playback 4×

Additional real-robot demonstrations

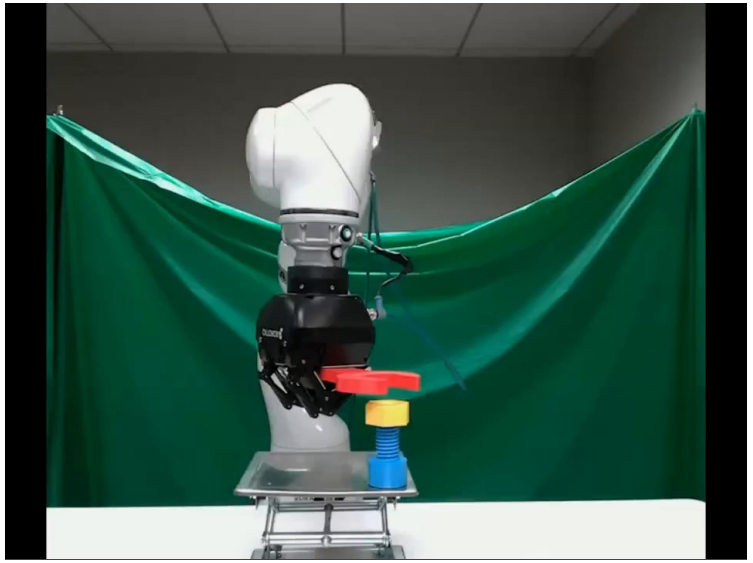
4× speed

PL-MPC · Zero-shot sim-to-real · No real-world fine-tuning

Training size 5 · 46 mm

Success

Training size 5 · 46 mm



Success

Unseen size 3 · 36 mm

Success

Representative successful PL-MPC trials · Aggregate rates on adjacent slides · Playback 4×

Transfer to unseen object sizes

6× speed

Size 3 · 36 mm

PL-MPC

Success

80.0%

12/15 trials

TD-M(PC)²

Timeout

73.3%

11/15 trials

Size 1 · 30 mm

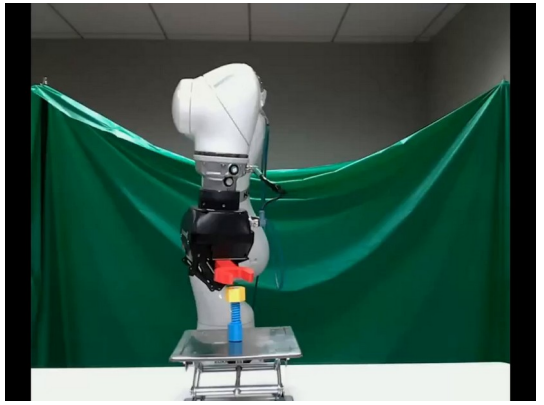
PL-MPC

Success

33.3%

5/15 trials

TD-M(PC)²



Success

20.0%

3/15 trials

Observed success · One trained checkpoint/method · Playback 6×

Summary: closing the planning–learning loop

PL-MPC · Planning–Learning MPC

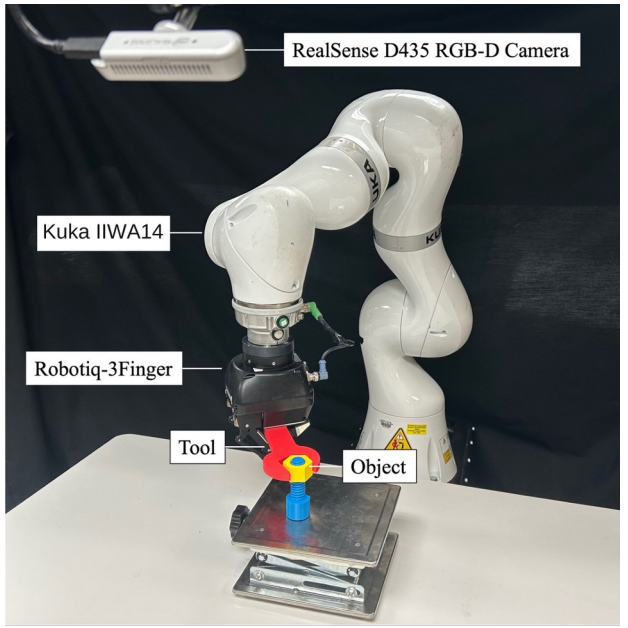
Studied Problem

Policy alignment addresses only one part of the planning–learning loop.

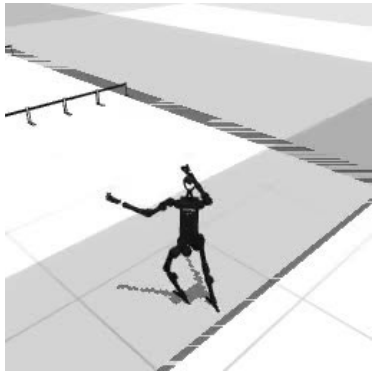
Our Solution

PL-MPC · A policy-constrained TD-MPC extension.

1. **MTD · Hybrid multi-step TD targets**
   More observed rewards before bootstrapping.
2. **ATE · Adaptive terminal estimates**
   Penalize terminal values using critic disagreement.
3. **RAD · Return-weighted actor distillation**
   Weight imitation by realized episode return.



balance-hard



hurdle

https://icra27-pl-mpc.github.io/

Result **balance-hard: 98 → 387 TAR** **hurdle: 199 → 466 TAR** **Robot: 61.3% → 74.2%**

vs. TD-M(PC)^2 · TAR: Total Average Return · Robot: observed success on training size (31 trials/method).