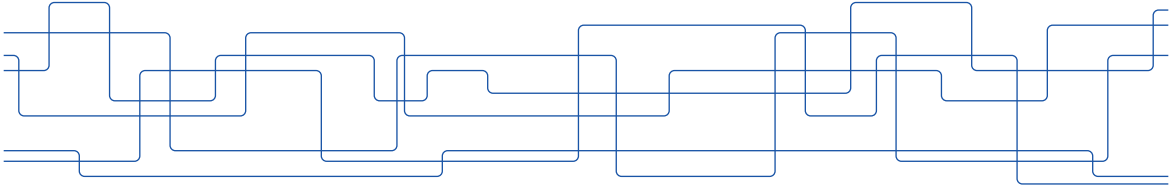
Reinforcement Learning

PhD Course FDD3359 – 2022

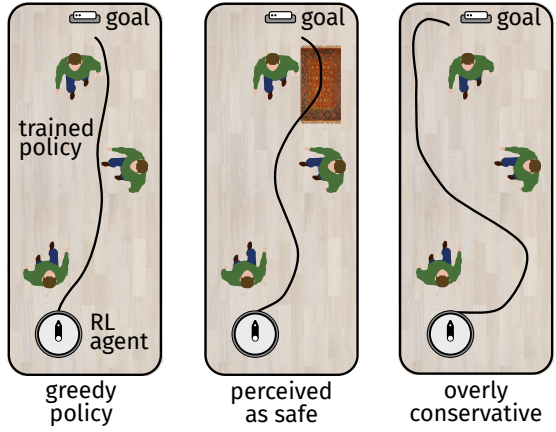
—Correcting learned robot policies with human feedback—

Chris Pek



Do learned policies really do what we want?

- Deployment environments differ from simulation
- Policies might not be *perceived as safe*
- Policies may *fail* due to environmental changes



How can we leverage human feedback to correct policies during/after training?

Learning outcomes

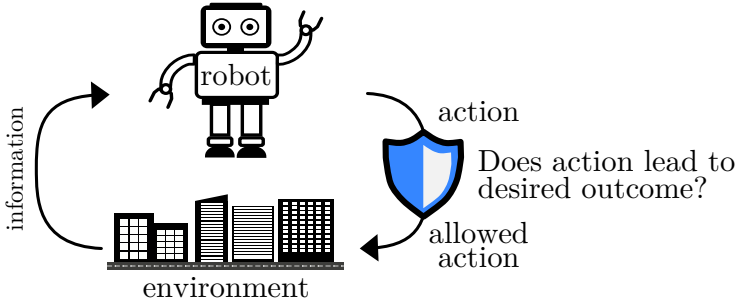
By the end of this session, you will be able to

1. explain and define human-in-the-loop reinforcement learning;
2. understand how to leverage human feedback;
3. explain basic algorithms and ideas for human-in-the-loop reinforcement learning;
4. understand the challenges of using human feedback;
5. formulate human-in-the-loop setups for different RL applications;
6. apply shielding with human feedback to various problems;
7. analyse and evaluate how you can leverage human feedback in your RL applications.

Human-in-the-loop policy corrections

- More on human-in-the-loop learning [Akalin and Loutfi, 2021]:
  - Learning from demonstrations
  - Interactive reinforcement learning
  - Preference learning
  - ...
- Today we focus on correcting policies before and after training:
  - Marta, D., Pek, C., Tumova, J., and Leite, I. Human-feedback shield synthesis for perceived safety in deep reinforcement learning. *IEEE Robotics and Automation Letters*, 7(1):406–413, 2021.
  - Van Waveren, S., Pek, C., Tumova, J., and Leite, I. Correct me if I'm wrong: Using non-experts to repair reinforcement learning policies. In *Proc. of the ACM/IEEE Conf. on Human-Robot Interacton*, pages 493–501, 2022.

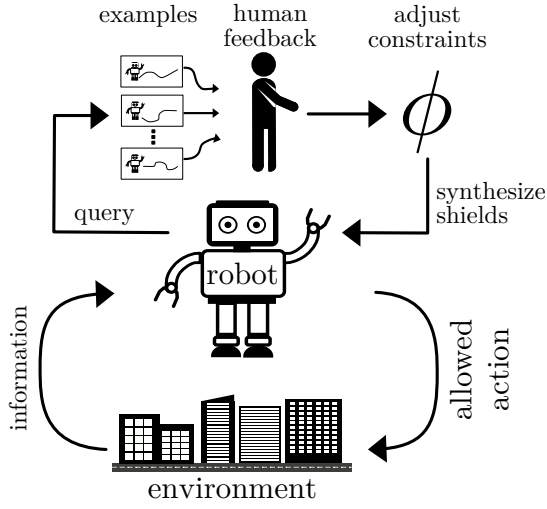
Shielding to constrain reinforcement learning



- Robots learn through interactions with their environment
- Actions may lead to undesired outcomes
- Shields constrain learning

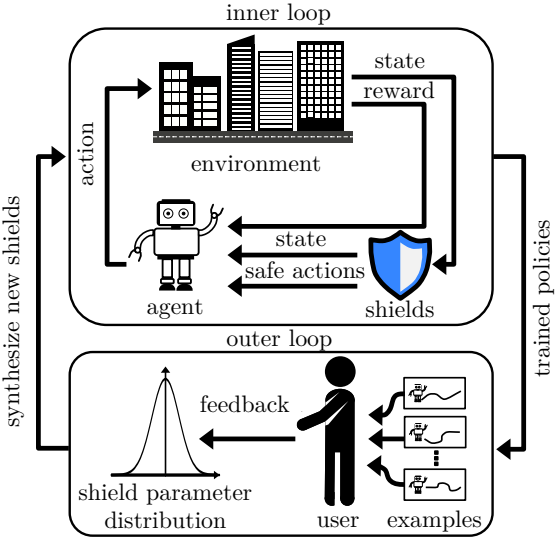
Correcting learned robot policies with human feedback

- Idea: humans can provide additional knowledge
- Query humans for feedback on policy
- Constrain learning based on human feedback
- Minimize number of queries, expressive queries/feedback



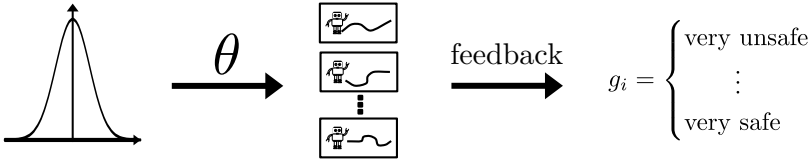
Human-feedback shield synthesis

- Parameterize shields: $\mathrm{shield}(\theta)$
- Inner loop trains policy
- Outer loop updates θ
- Repeat until convergence of θ



Shield and feedback distributions

- Model shield parameter with distribution f_s
- Model human parameter guess with distribution f_h
- Feedback samples, e.g., $g_i \in \mathcal{H} = \{\text{very unsafe}, \ldots, \text{fine}, \ldots, \text{very safe}\}$
- Advantage: higher robustness against uncertain feedback



Updating human distribution through emperical data

- Mapping of human feedback to machine interpretable data:

$$\text{map}(g_j) = \begin{cases} \mu_h - \frac{|\mathcal{H}|\sigma}{2} & \text{if } g_j = \text{very unsafe}, \\ \vdots & \vdots \\ \mu_h & \text{if } g_j = \text{fine}, \\ \vdots & \vdots \\ \mu_h + \frac{|\mathcal{H}|\sigma}{2} & \text{if } g_j = \text{very safe}. \end{cases}$$

- Update distribution f_h with new mean ${}^u\mu_h$ and variance ${}^u\sigma_h^2$:

$${}^u\mu_h = \frac{1}{N_{\text{user}}} \sum_{j=1}^{N_{\text{user}}} \text{map}(g_j)$$

$${}^u\sigma_h^2 = \max\left(\frac{1}{N_{\text{user}}} \sum_{j=1}^{N_{\text{user}}} \left(\text{map}(g_j) - ({}^u\mu_h^2)\right), \sigma_{\min}^2 \right)$$

Updating shield distribution through Bayesian inference

- Update f_s through Bayesian inference from human distribution f_h
- Set $\mathcal{G}$ of samples from human distribution
- Likelihood of true parameter distribution:

$$p(\mathcal{G}|\mu, \sigma^2) = \prod_{j=1}^{|\mathcal{G}|} p(x_j|\mu, \sigma^2), x_j \in \mathcal{G}$$

- Bayesian update of f_s through posterior:

$$\hat{p}_\theta(\hat{\mu}_\theta|\mathcal{G}, \hat{\sigma}_\theta^2) \propto p(\mathcal{G}|\mu, \sigma^2) p_\theta(\mu|\mu_\theta, \sigma_\theta^2)$$

Updating shield distribution through Bayesian inference

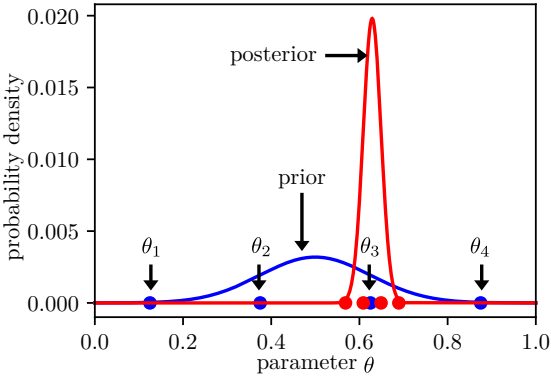
- Closed form solution for Gaussian distributions:

$$\hat{\sigma}_\theta^2 = \frac{1}{\frac{n}{\sigma} + \frac{1}{\sigma_\theta^2}},$$

$$\hat{\mu}_\theta = \hat{\sigma}_\theta^2 \left(\frac{\mu_\theta}{\sigma_\theta^2} + \frac{n\bar{x}}{\sigma^2} \right),$$

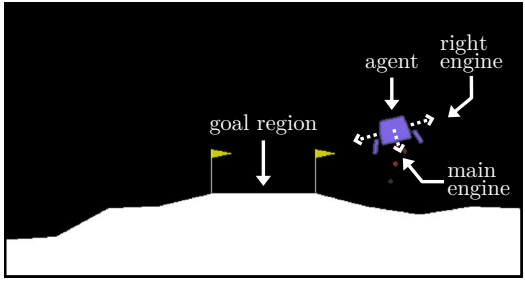
where $n = |\mathcal{G}|$ and $\bar{x}$ is the mean of the samples in $\mathcal{G}$.

- Train until KL-divergence sufficiently small



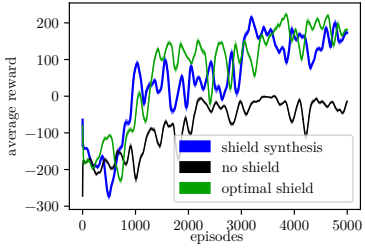
Example: safe Lunar Landing

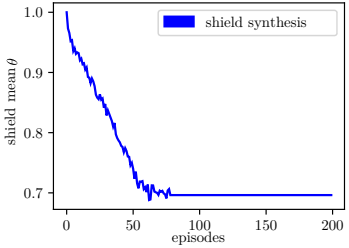
- Safety is not overusing main engine, i.e., usage $> 85\%$
- Shield limits main engine use: $a \in [-\theta, \theta], \theta \in [0, 1]$
- Simulated humans

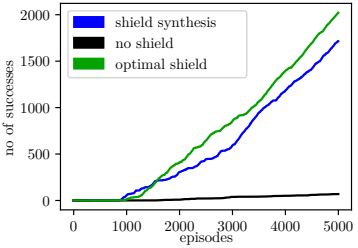


Example: safe Lunar Landing

- Parameter convergences to optimal parameter
- Using human knowledge and shielding improves learning

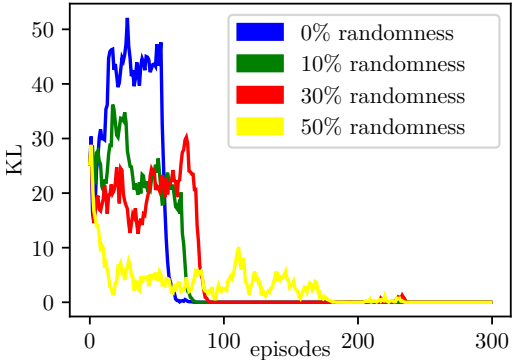






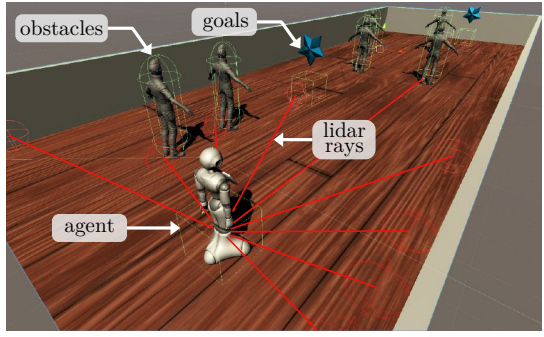
Example: safe Lunar Landing

- Increasing uncertainty in human feedback
- Parameter still convergences
- Convergence requires more episodes



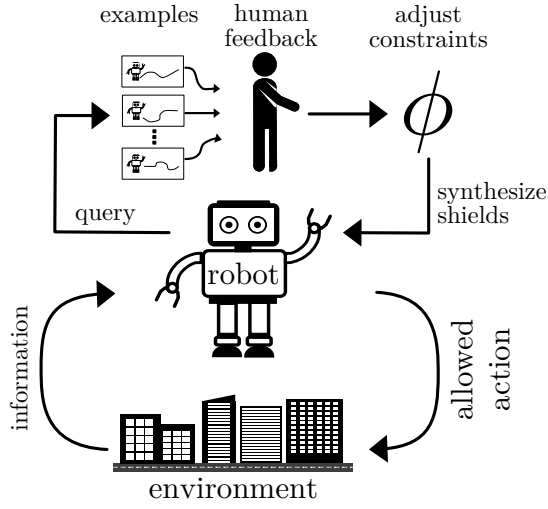
Example: safe social navigation

- Reach goal while not running into humans
- Shield constrains minimum distance to humans
- Parameter θ defines minimum distance



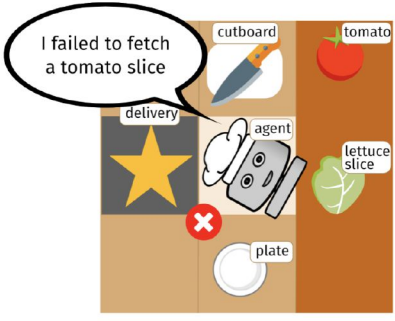
Correcting learned robot policies with human feedback

- Idea: humans can provide additional knowledge
- Query humans for feedback on policy
- Constrain learning based on human feedback

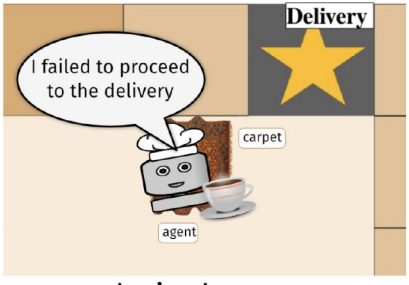


Deploying policies in real environments

- Policies may fail in the deployment environment due to environmental changes
- Can we leverage human feedback to correct policies even after deployment?



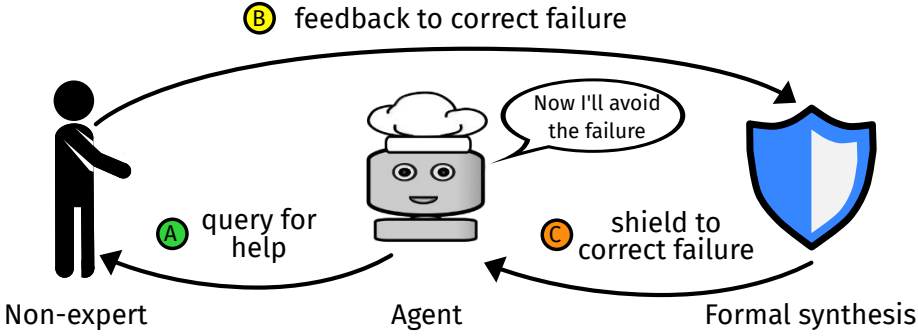
missing item



undesired outcome

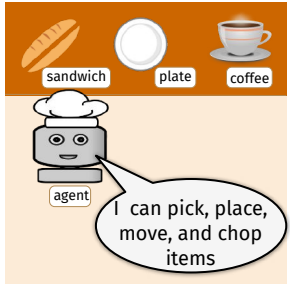
Correcting failures with human feedback and shielding

- Idea: robot queries for help when encountering a failure



Correcting failures with human feedback and shielding

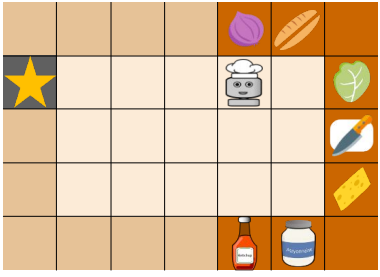
- High-level decision making tasks
- Common daily tasks
- Interpretable actions $\mathcal{A}$ and states $\mathcal{S}$
- Non-experts can provide feedback



high level actions in common daily tasks

Failure queries

- Robot needs to send meaningful queries
- Failure trace and environment overview

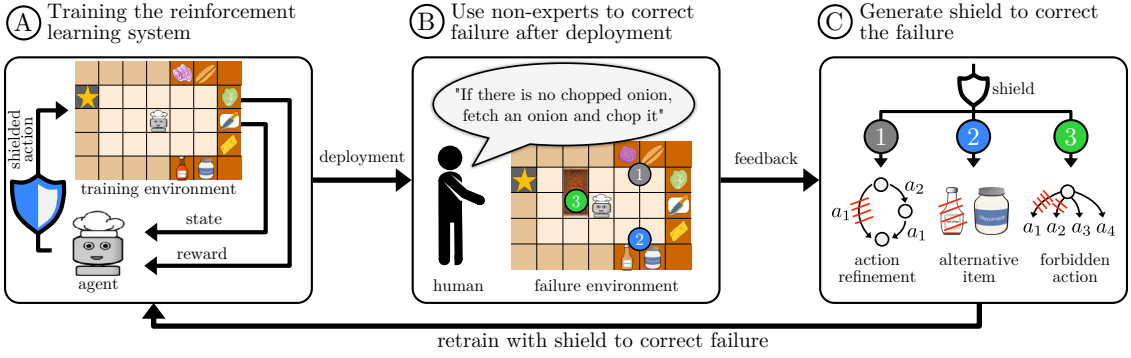


I executed the following actions until the failure:
1) Fetching sliced bread
2) Fetching ketchup
3) Fetching salad slice
4) Fetching chopped onions

I failed to complete my task. I tried to `fetch a chopped onion`, but failed.

What could I do to avoid this failure and complete the task in the future?

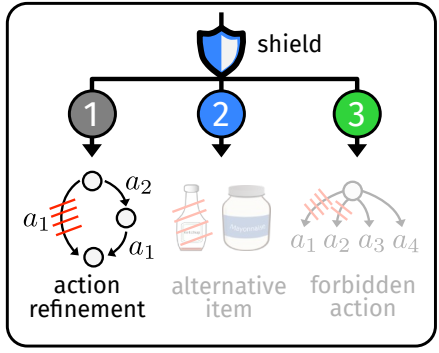
Failure correction shield synthesis



Action refinement

- State transition $\delta(s_t, a) = s_{t+1}$
- Desired states $\mathcal{S}_{\text{desired}}$ without failure
- Feedback $c(s, a)$ for alternative action
- Refine action when undesired outcome

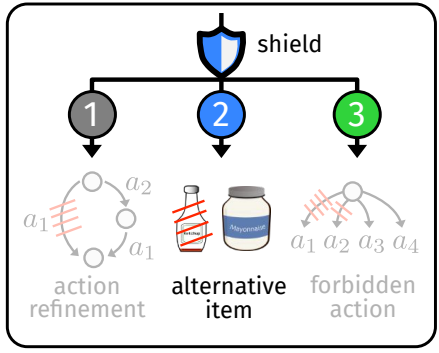
$$\text{refine}_s(a) = \begin{cases} a, & \text{if } \delta(s,a) \in \mathcal{S}_{\text{desired}} \\ c(s,a), & \text{if } \delta(s,a) \notin \mathcal{S}_{\text{desired}} \end{cases}$$



Alternative item

- Action $a^{<p>}$ manipulates item p
- Original item p_O not available
- Non-expert suggests alternative p_A
- Shield changes action parameter

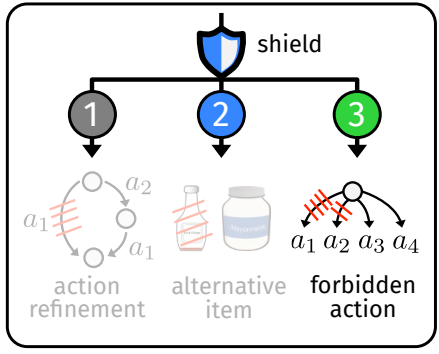
$$\mathrm{alt}(a^{<p_O>}) = \begin{cases} a^{<p_O>}, & \text{if } p_O \text{ in Env} \\ a^{<p_A>}, & \text{if } p_O \text{ not in Env} \end{cases}$$



Forbidden actions

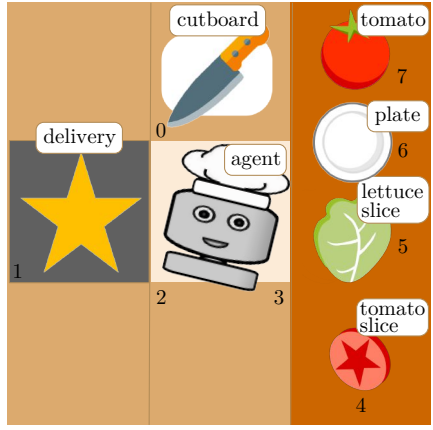
- Action may not be desired any more
- Feedback labels forbidden actions
- Desired states $\mathcal{S}_{\text{desired}}$ without failure
- Shield removes forbidden actions

$$\mathcal{A}_{\text{allowed}}(s) = \{a \in \mathcal{A} \mid \delta(s, a) \in \mathcal{S}_{\text{desired}}\}$$



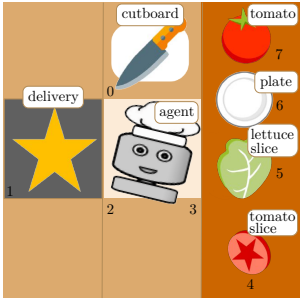
Example: kitchen environment

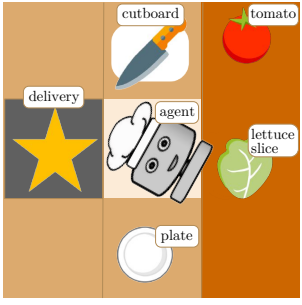
- Overcooked-AI based environment [Wang et al., 2020]
- Agent is tasked to prepare/deliver certain dishes
- Grid environment
- Actions such as turn, fetch, chop, and deliver an item



Example: making a salad

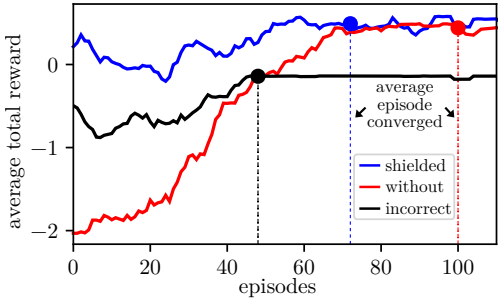
- Failure: chopped tomatoes are not available any more
- Non-expert suggests to chop tomatoes in such cases
- Action refinement





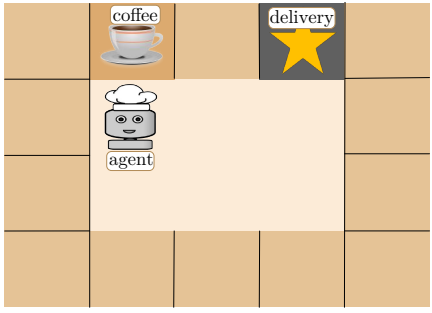
Example: making a salad

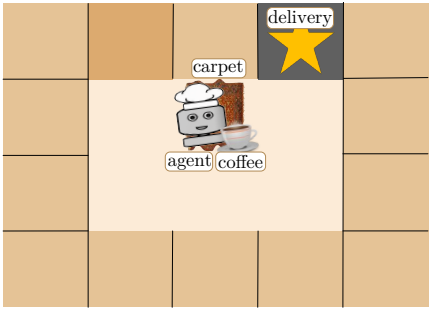
- Shielding always corrects failure
- Faster convergence
- Check feedback quality



Example: deliver coffee

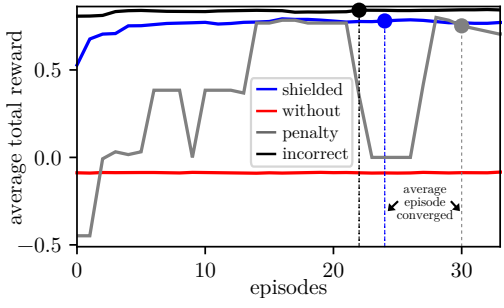
- Failure: environment contains carpets and agent gets stuck
- Non-expert suggests to never move to carpets
- Forbidden actions





Example: deliver coffee

- Shielding always corrects failure
- Without shield never successful
- Requires reward engineering to solve task
- Incorrect feedback might result in non-generalizable solutions



Conclusions

How can we leverage human feedback to correct policies during/after training?

- Human feedback can provide additional knowledge
- Different ways to integrate humans into learning
- Speed up learning, improving acceptability, and correcting policies
- Consider number and type of queries, type of feedback
- Update parameters of shields during learning
- Automatic and verifiable failure correction after deployment

References I


Akalin, N. and Loutfi, A. (2021).
Reinforcement learning approaches in social robotics.
Sensors, 21(4):1292.


Marta, D., Pek, C., Tumova, J., and Leite, I. (2021).
Human-feedback shield synthesis for perceived safety in deep reinforcement learning.
IEEE Robotics and Automation Letters, 7(1):406–413.


Van Waveren, S., Pek, C., Tumova, J., and Leite, I. (2022).
Correct me if I'm wrong: Using non-experts to repair reinforcement learning policies.
In *Proc. of the ACM/IEEE Conf. on Human-Robot Interacton*, pages 493–501.


Wang, R. E., Wu, S. A., Evans, J. A., Tenenbaum, J. B., Parkes, D. C., and Kleiman-Weiner, M. (2020).
Too many cooks: Coordinating multi-agent collaboration through inverse planning.
arXiv preprint arXiv:2003.11778.