Mitigating Goal Misgeneralization via Minimax Regret

Brought together by KASL — KRUEGER AI SAFETY LAB

Karim Abdel Sadek(=) University of California, Berkeley

Matthew Farrugia-Roberts(=) University of Oxford

Usman Anwar University of Cambridge

Hannah Erlebach University College London

Christian Schroeder de Witt University of Oxford

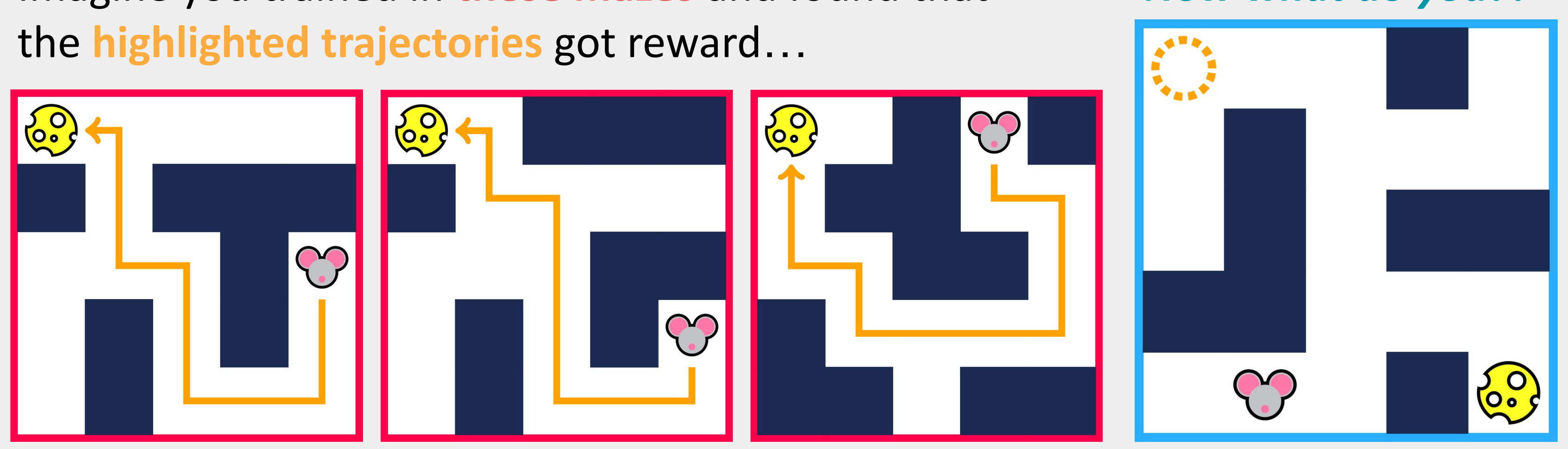
David Krueger Mila, University of Montreal

Michael Dennis Google DeepMind

Quiz: Are *YOU* susceptible to *goal misgeneralization??*

Training:
Imagine you trained in these mazes and found that the highlighted trajectories got reward…

Deployment:
Now what do you??

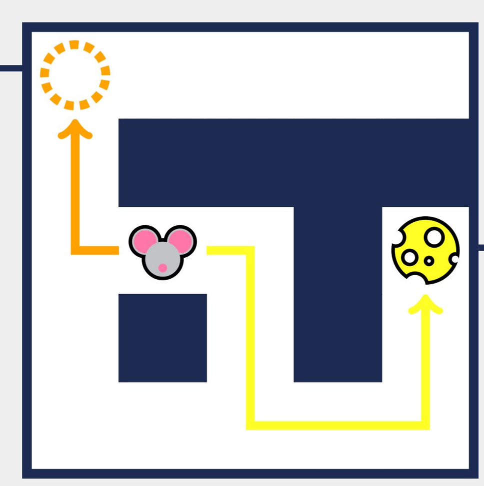


Goal misgeneralization: Learning a policy for a **proxy goal** instead of the **intended goal** from an **ambiguous training environment** distribution.

Problem Setting:
Proxy-Distinguishing Distribution Shift

1. Define goals…

Proxy goal: Get to the top-left corner of the maze (even if it is empty)

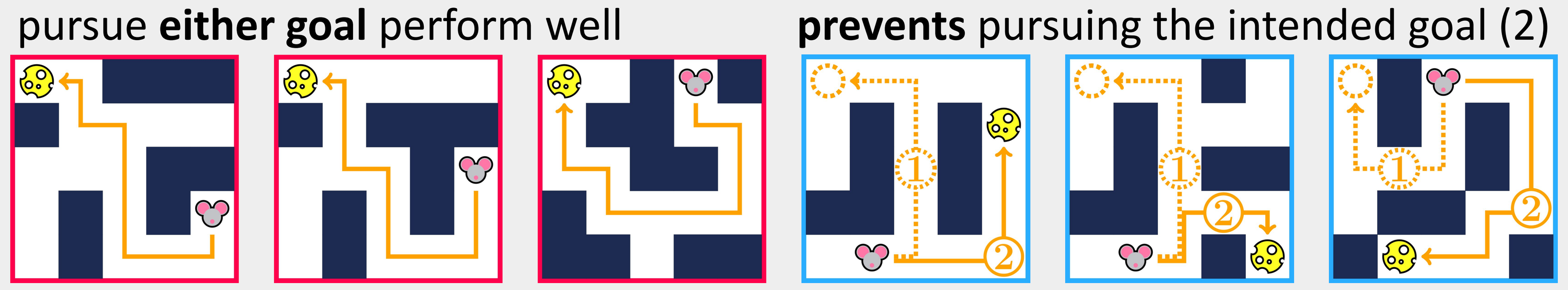


Intended goal: Get to the cheese, (no matter where it is in the maze)

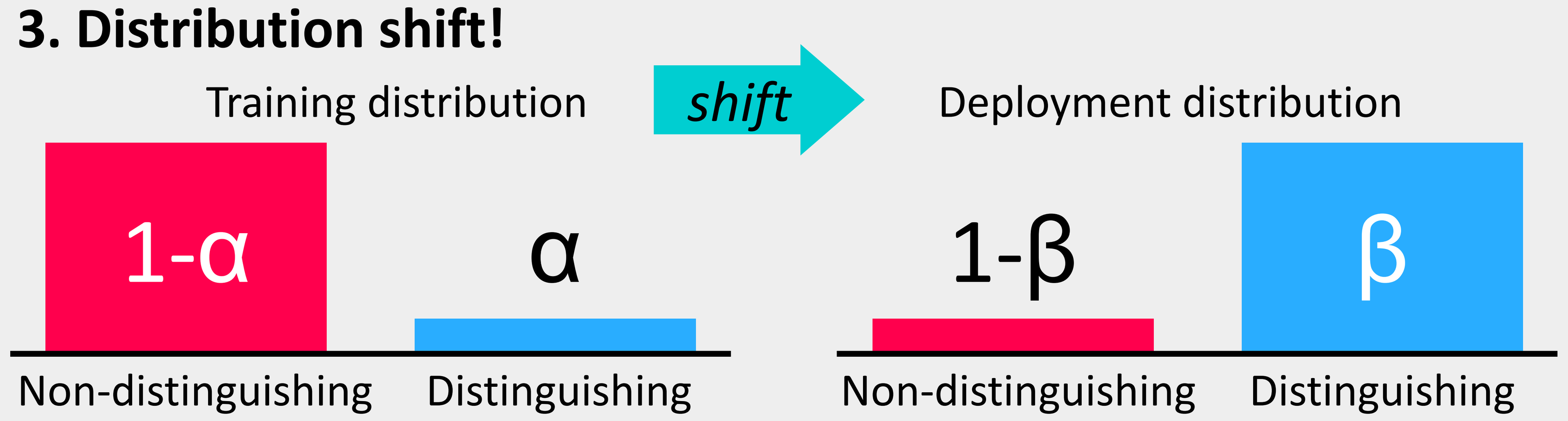
2. Classify levels…

Non-distinguishing levels: policies that pursue **either goal** perform well

Distinguishing levels: pursuing proxy (1) **prevents** pursuing the intended goal (2)



3. Distribution shift!



We show that training with

the *maximum expected value* objective is **susceptible** to goal misgeneralization!

Approximate **Maximum Expected Value (MEV)** objective:

$$\pi^{\mathrm{MEV}} \in \underset{\pi\in\Pi}{\text{arg-}\varepsilon\text{-max}}\,\mathrm{Value}(\pi, \Lambda^{\mathrm{Train}})$$

Approx. maximization within some threshold ε ≥ 0 of optimal policy

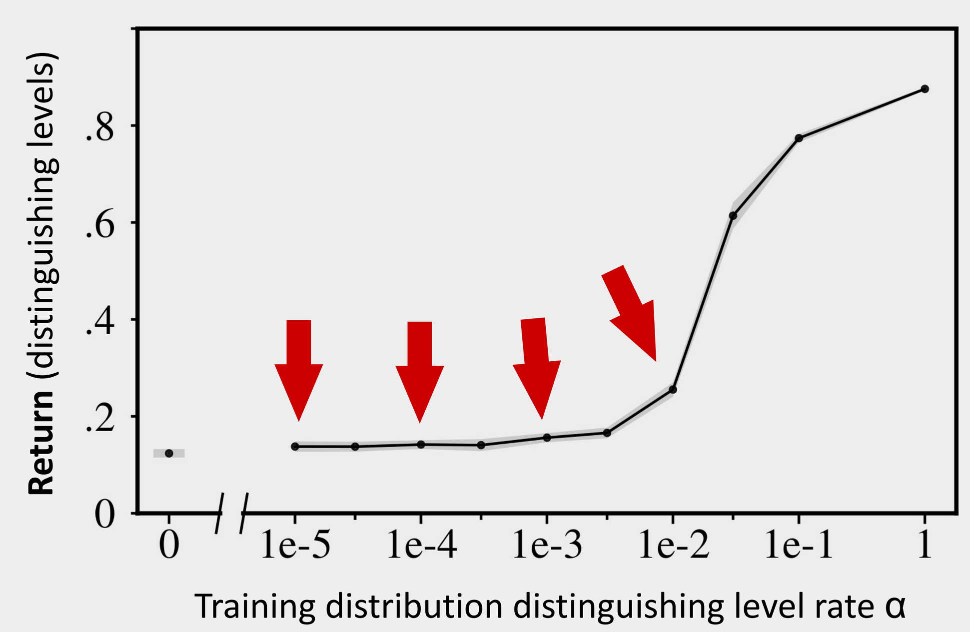
Expected return over some fixed level distribution

Training distribution of non-distinguishing / distinguishing levels

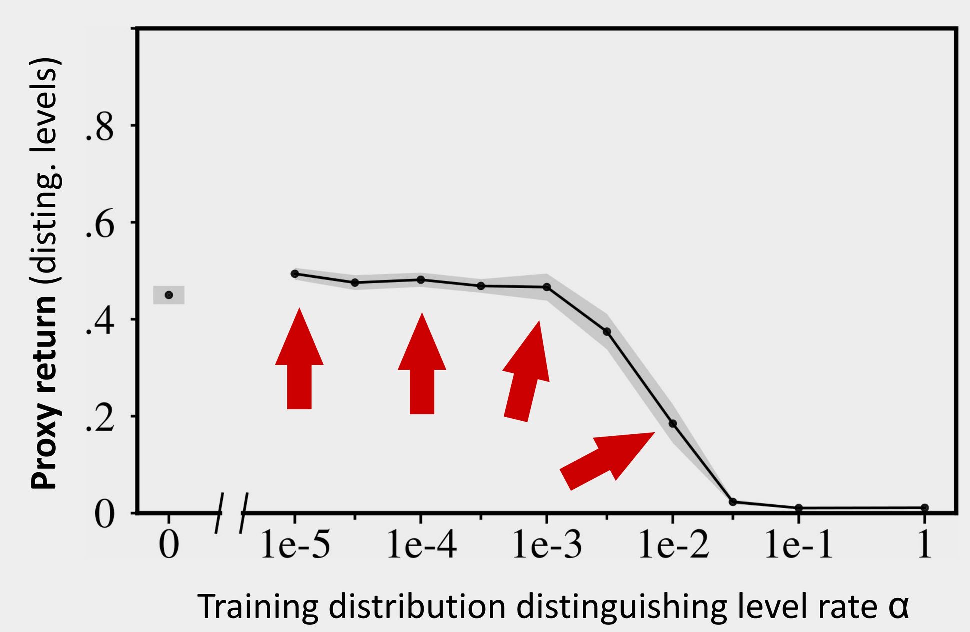
Theorem 1: If α ≤ ε, some MEV policies pursue the proxy goal:

$$\exists\pi^{\mathrm{MEV}};\ \pi^{\mathrm{MEV}} \in \underset{\pi\in\Pi}{\arg\max}\,\mathrm{ProxyValue}(\pi, \Lambda^{\mathrm{Deploy}}) \setminus \underset{\pi\in\Pi}{\text{arg-}\beta\text{-max}}\,\mathrm{Value}(\pi, \Lambda^{\mathrm{Deploy}})$$

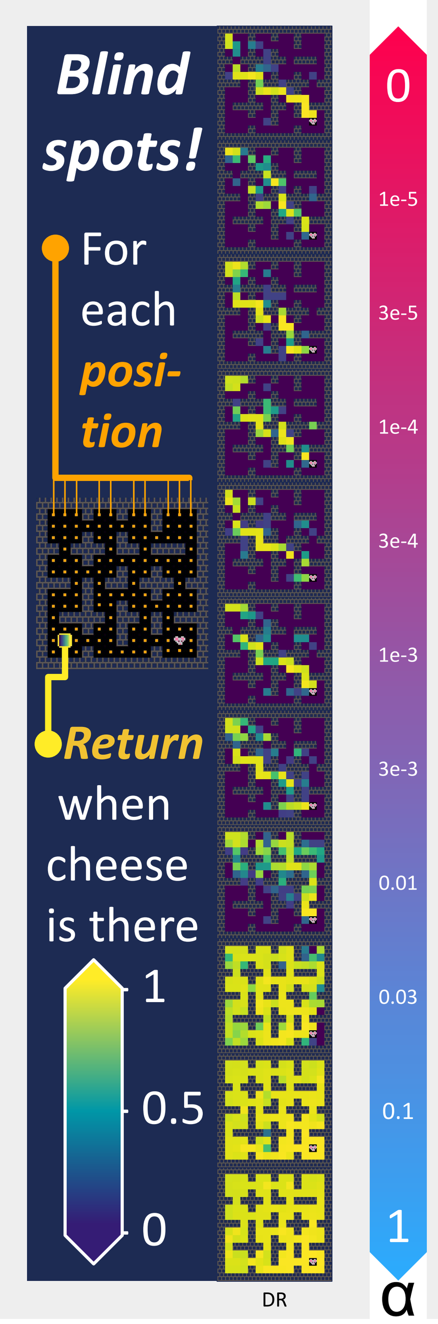
Experiments with *Domain Randomization:*
We train with **domain randomization** (implementing the MEV objective). We use training distributions with **varying α** (proportion of distinguishing levels).



When α < 0.03, domain randzn. learns a policy that **fails to pursue the intended goal** on distinguishing levels...



… instead the policy **pursues the proxy goal** on these levels, leading to **mis-generalization in deployment.**



On the other hand, training with

the *minimax expected regret* objective is **robust** to goal misgeneralization!

Approximate **MiniMax Expected Regret (MMER)** objective:

$$\pi^{\mathrm{MMER}} \in \underset{\pi\in\Pi}{\text{arg-}\varepsilon\text{-min}}\ \max_{\Lambda\in\Delta(\mathrm{lvl.})} \mathrm{Regret}(\pi, \Lambda)$$

Approx. minimization within some threshold ε ≥ 0 of optimal policy

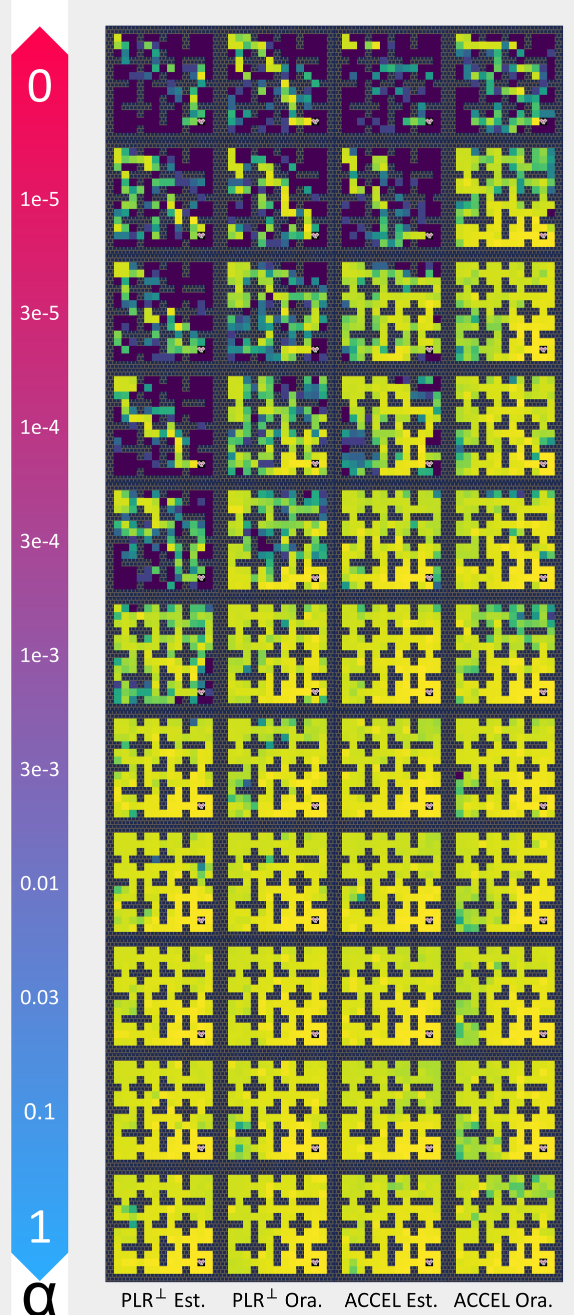
Inner maximization worst-case level distr. relative to policy

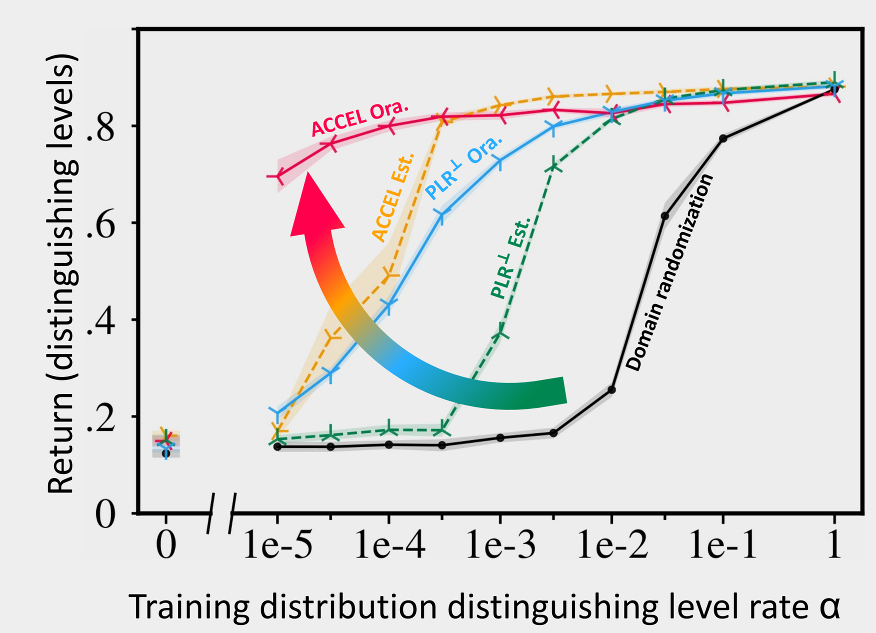
Expected Regret: $\mathrm{Value}(\pi^\star, \mathrm{level}) - \mathrm{Value}(\pi, \mathrm{level})$ averaged over level distribution

Theorem 2: All MMER policies pursue the intended goal:

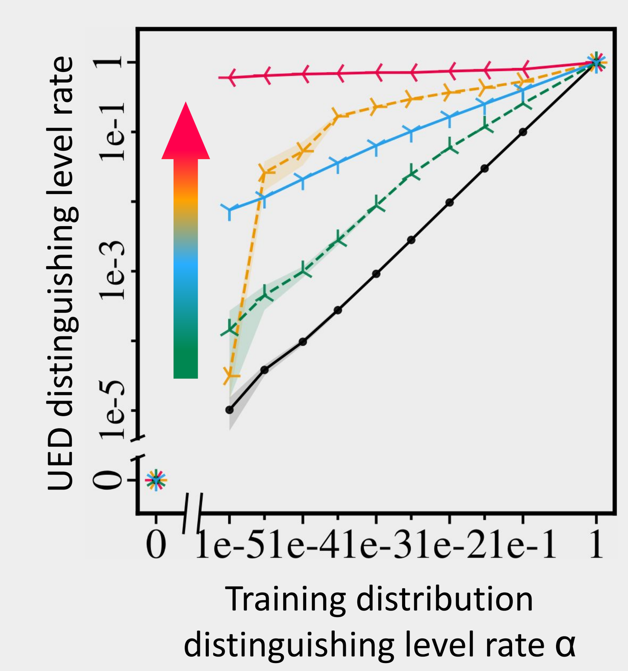
$$\forall\pi^{\mathrm{MMER}},\ \pi^{\mathrm{MMER}} \in \underset{\pi\in\Pi}{\text{arg-}\varepsilon\text{-max}}\ \mathrm{Value}(\pi, \Lambda^{\mathrm{Deploy}})$$

Experiments with *Unsupervised Environment Design:*
Train with **unsupervised environment design** (implementing MMER objective). We use four increasingly powerful adversarial designers and regret estimators.





UED policies **pursue the intended goal** at many low α where domain randomization policy pursued the proxy goal.



How? The adversary **finds high-regret distinguishing levels** and plays them more often than α.

See paper for…
+ theory details
+ more results
+ more environments
+ more methods