Safe Reinforcement Learning

bridging theory and practice

Ming Jin
Assist. Prof. @ Virginia Tech

Shangding Gu
Postdoc. @ UC Berkeley

IJCAI 2024 Tutorial, August 5, 2024

Reinforcement Learning: Foundation of Intelligent Agents

Reinforcement Learning (RL)
an agent learns optimal policy through environment interaction and feedback

interactive

continuous agent-environment interaction

e.g., human-robot collaboration, control applications

feedback-driven

learning guided by environmental responses

e.g., RL w/ human feedback (RLHF) for LLMs, recommender systems

complex planning

sequential decision-making & adaptation

e.g., AlphaGo's strategic gameplay

The RL Mountain Climb: Scaling Peaks of Complexity

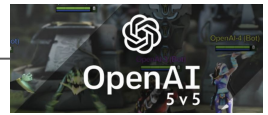
RL has evolved from simple board games to complex, real-time strategy games

…but is scaling complexity alone enough for real-world deployment?

complexity

2019

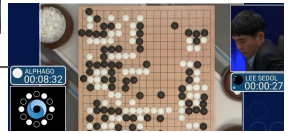
AlphaStar, achieving **grandmaster level** in **StarCraft II**



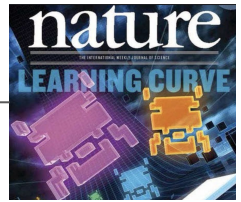
OpenAI Five in **DOTA 2**, a multiplayer strategy game

2017

2016



AlphaGo by DeepMind, defeating the world champion in Go



Deep Q-Networks (DQN) by DeepMind, applying deep learning to RL and outperforming humans on several **Atari games**

2013

1997

TD-Gammon, using temporal difference learning, achieving expert-level play

Samuel's checkers- playing program (1959)

1950s

Bridging the Safety Chasm

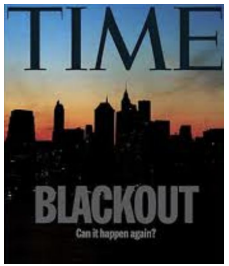
real-world application

simulation success

treatment safety → healthcare

no mechanical harm → robotics

grid stability
voltage regulation → power systems



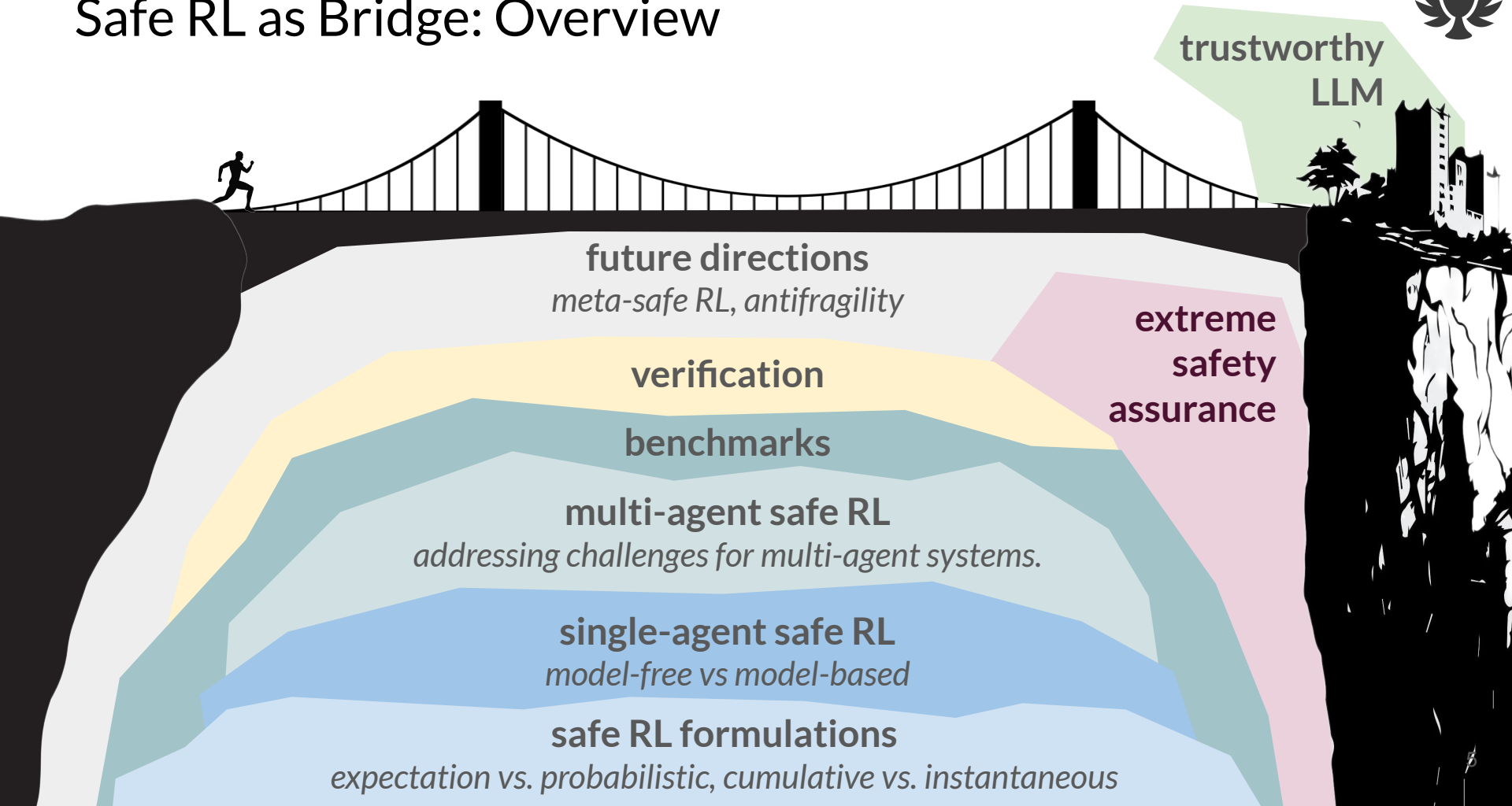
collision avoidance → autonomous driving

alignment, ethical → AI agent

Safety Definition

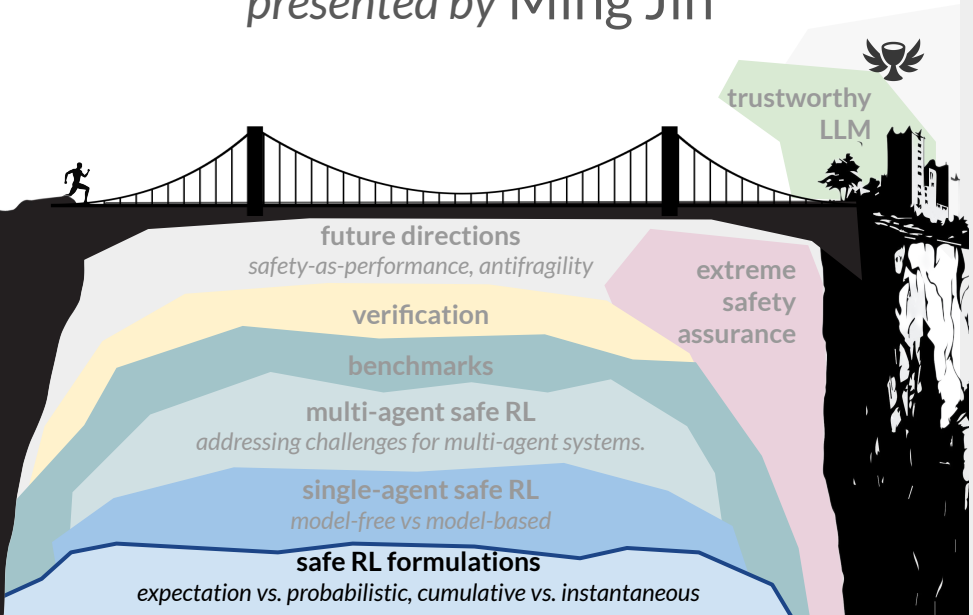
agents act without causing **harm**, maintaining **stability**, **reliability,** and **ethical standards** across diverse real-world environments.

Safe RL as Bridge: Overview



safe RL formulations

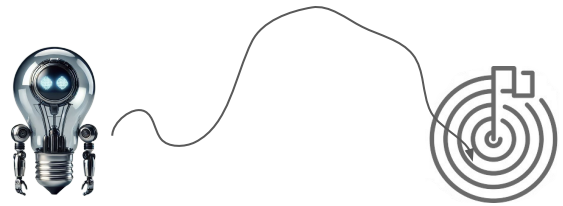
presented by Ming Jin



Key Objectives

- understand Constrained MDPs (CMDPs) and various safe RL formulations
- compare and contrast different safety constraint types (expectation-based vs. probability-based, instantaneous vs. cumulative)
- explore real-world applications and considerations of safe RL in energy systems and AI language models

Safe RL Formulations (1/7)



RL Goal

find the optimal policy that maximizes the **expected cumulative reward over time**:

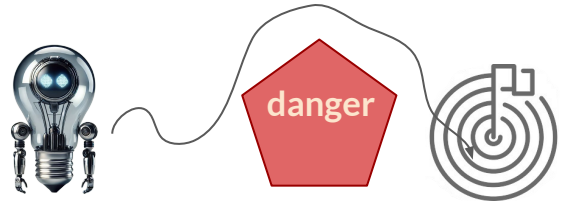
$$\max_{\pi} V_r^{\pi}(\rho) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t)\right]$$

MDPs (Markov Decision Processes):
standard framework for RL

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho)$$

- $\mathcal{S}$ **state space** (e.g., robot's position)
- $\mathcal{A}$ **action space** (e.g., move forward, turn left/right)
- $\mathcal{P}$ **transition function** (e.g., prob. of next position)
- r **reward function** (e.g., distance to goal)
- γ **discount factor** (between 0 to 1)
- ρ **initial state distribution**

Safe RL Formulations (2/7)



Safe RL general formulation

find the optimal policy that maximizes the expected cumulative reward over time while **ensuring safety**:

$$\max_{\pi} V_r^{\pi}(\rho) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t)\right]$$

subject to **safety constraint**

$$\pi \in \Pi_{\text{safe}}$$

CMDPs (Constrained MDPs):

standard framework for Safe RL

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho) \cup (c, \xi)$$

c **safety cost function** (e.g., proximity to obstacles)

ξ **safety threshold**

Expected Cumulative Cost Constraint:

Π_{safe} : all policies π such that

$$V_c^{\pi}(\rho) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t c(s_t, a_t)\right] \leq \xi$$

👍 mirrors reward structure, most common
❎ delayed feedback, no guarantee for trajectories

Expectation-based Constraints (3/7)

Safe RL general formulation

$$\max_{\pi} V_r^{\pi}(\rho) := \mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t)\right]$$

subject to **safety constraint**

$$\pi \in \Pi_{\text{safe}}$$

expected cumulative cost constraint

$$V_c^{\pi}(\rho) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t c(s_t, a_t)\right] \leq \xi$$

special case:

$$c(s,a) = \mathbb{I}(s \in S_{\text{unsafe}})$$

state constraint

$$\mathbb{E}_{\pi}\left[\sum_{t=0}^{H} \gamma^t \mathbb{I}(s_t \in S_{\text{unsafe}})\right] \leq \xi$$

👍focuses specifically on avoiding unsafe states
❎still cumulative, so delayed feedback

E.g., CISR [Turchetta et al., 2020]

address the **delayed feedback** issue

expected **instantaneous** safety

$$\mathbb{E}_{\pi}\left[c(s_t, a_t)\right] \leq \xi_t, \quad \forall t \in [H]$$

👍immmediate feedback on violations
❎may be overly conservative

E.g., Autonomous driving with changing speed limits, OptLayer [Pham et al., 2018]

Probability-based Constraints (4/7)

Markov's inequality $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$

*probability constraints are **stricter***

$$\mathbb{E}_\pi[c(s_t, a_t)] \leq \xi_t \Leftarrow \mathbb{P}_\pi[c(s_t, a_t) \leq \xi_t] = 1$$

*expectation as **conservative** approximation*

$$\mathbb{E}_\pi[c(s_t, a_t)] \leq \delta\xi_t \Rightarrow \mathbb{P}_\pi[c(s_t, a_t) \leq \xi_t] \geq 1 - \delta$$

expected cumulative cost constraint

$$V_c^\pi(\rho) = \mathbb{E}_\pi\left[\sum_{t=0}^{H} \gamma^t c(s_t, a_t)\right] \leq \xi$$

***Motivation:** move from average-case to worst-case safety, guarantee safety for every trajectory*

chance constraint on **cumulative** cost

$$\mathbb{P}_\pi\left[\sum_{t=0}^{H} \gamma^t c(s_t, a_t) \leq \xi\right] \geq 1 - \delta$$

👍safety for all trajectories
❎can be overly conservative, delayed feedback

E.g., Sauté RL [Sootla et al., 2022]

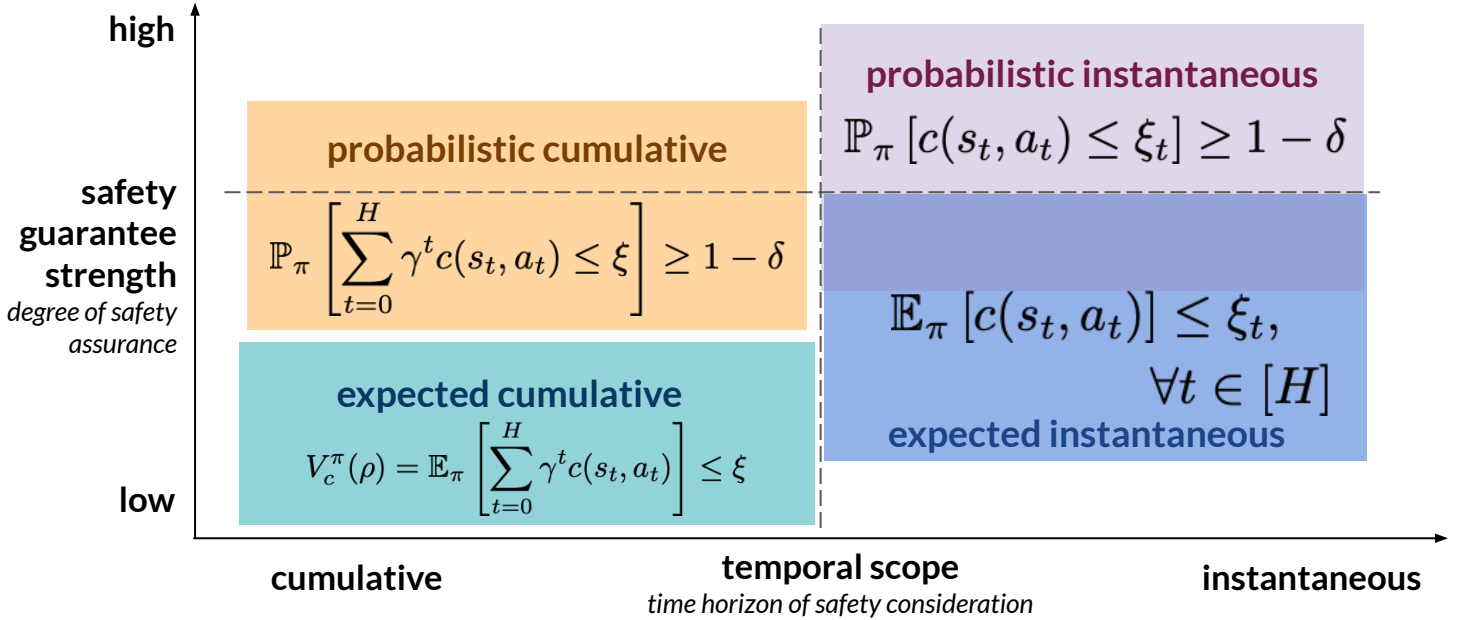
chance constraint on **immediate** cost

$$\mathbb{P}_\pi\left[c(s_t, a_t) \leq \xi_t\right] \geq 1 - \delta \quad \forall t \in [H]$$

👍immediate, strict safety guarantees
❎most conservative approach

E.g., MASE [Wachi et al., 2023]

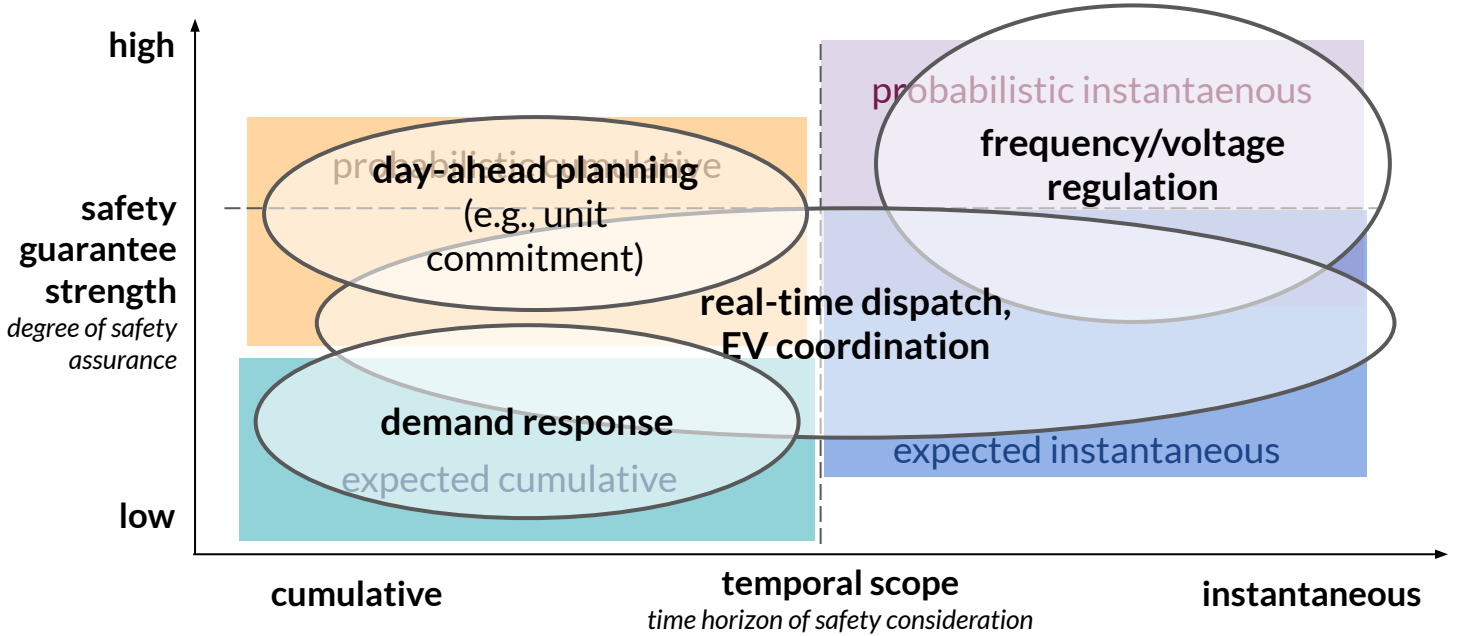
Safe RL Formulations: Comparison (5/7)



Key takeaways

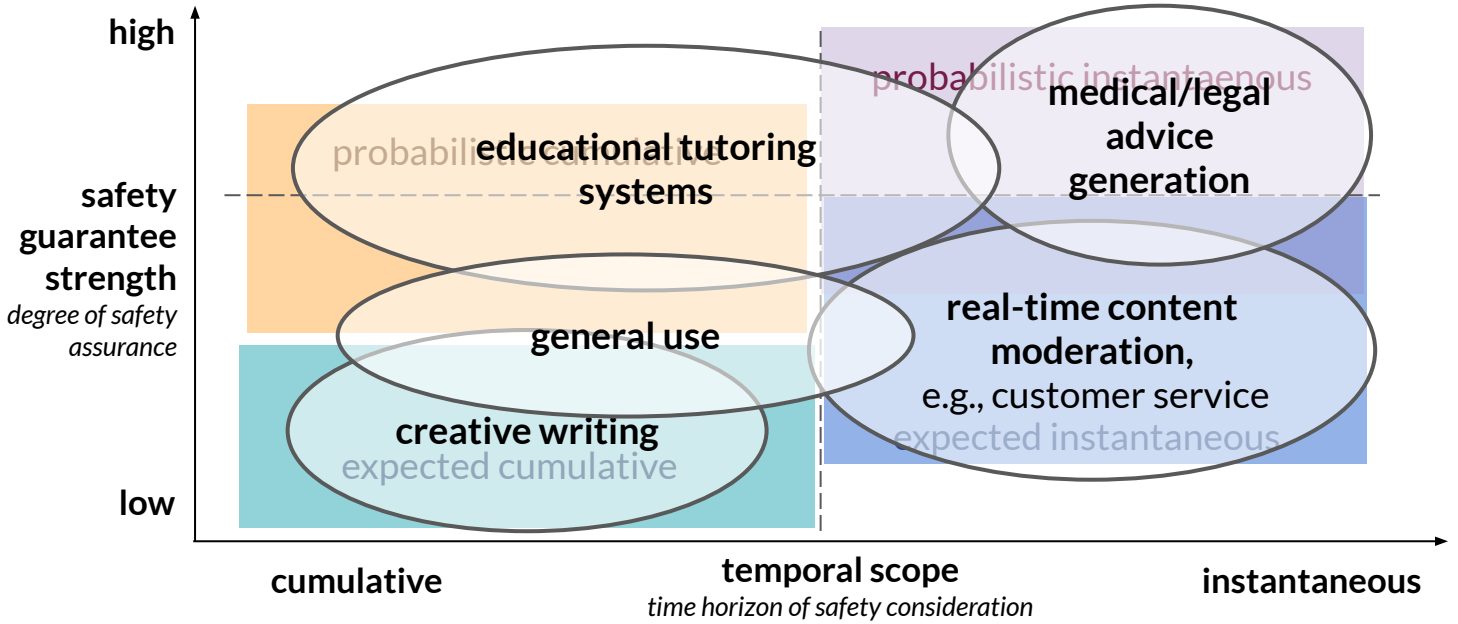
- Trade-off: stronger guarantees vs. computational complexity
- instantaneous constraints: Higher safety assurance, but potentially more conservative
- Choice depends on safety criticality and resources

Real-World Applications: Energy Grid (6/7)



- Clear distinction between instantaneous and cumulative constraints
- Safety criticality varies by grid function

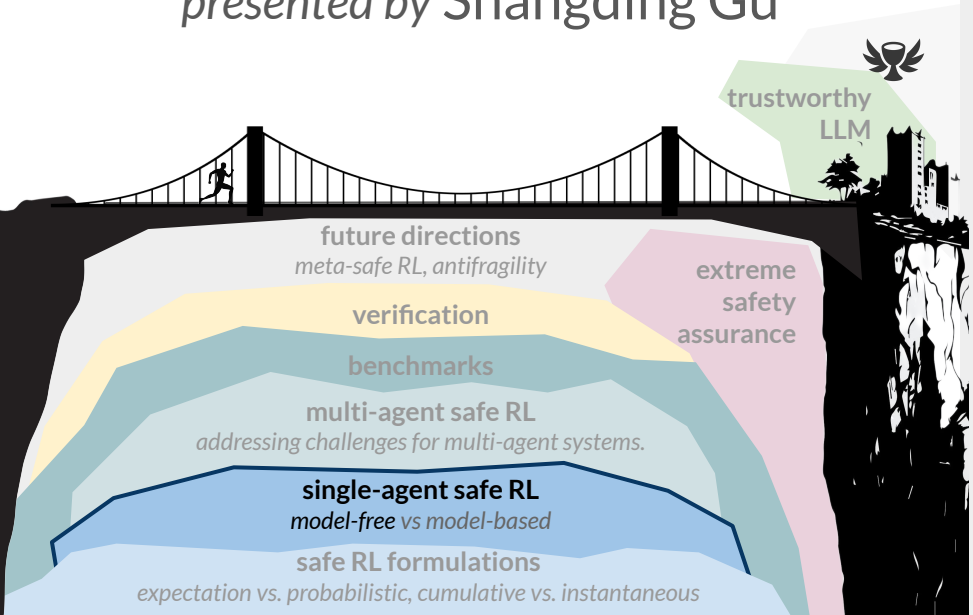
Real-World Applications: LLMs (7/7)



- Blend of instantaneous and cumulative safety considerations
- Challenge: Balancing immediate safety with long-term behavioral goals

safe model-free RL

presented by Shangding Gu

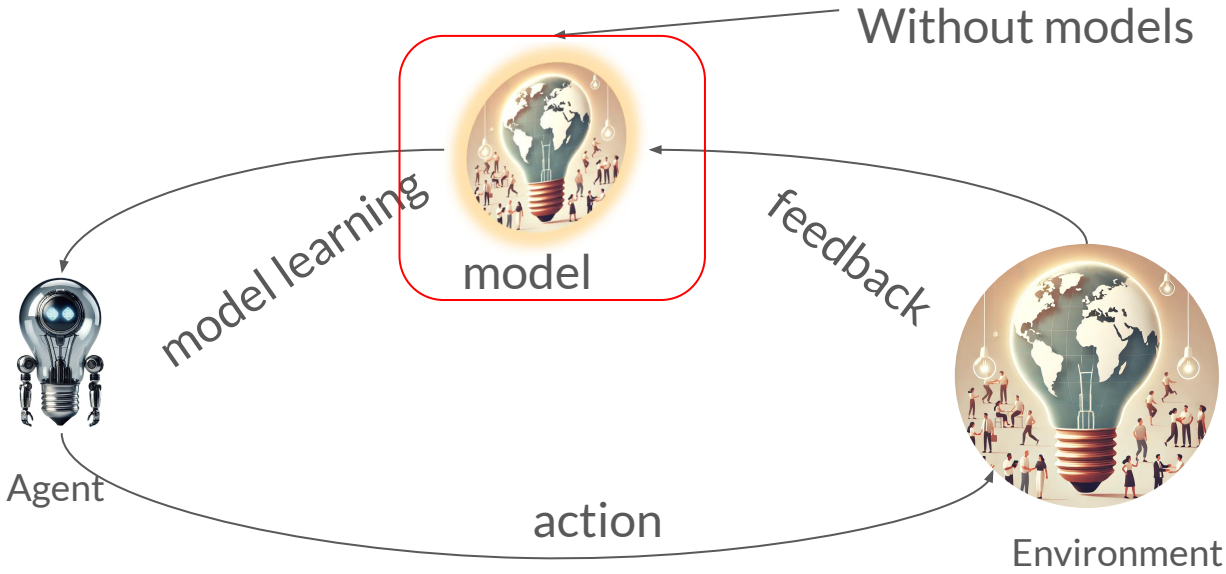


Key Objectives

- understand the framework of safe model-free reinforcement learning
- know the key ideas of primal-dual and primal based safe reinforcement learning
- explore popular safe model-free reinforcement learning algorithms

Model-free methods (Shangding)

- Primal-Dual Methods [Achiam et al., 2017]
- Primal Methods [Xu et al., 2021]
- Popular model-free safe RL methods: CPO [Achiam et al., 2017], MACPO [Gu et al., 2017], CRPO [Xu et al., 2021], etc.
- For each typical method we will introduce 3 SOTA baselines



Model-free methods

Overview:

When we do not know models, **we usually sample more to estimate reward and safety values during policy optimization.** Two popular safe RL solutions: primal-dual based methods and primal based methods.

The balance between reward and cost

Primal-Dual methods: Ensure safety by optimizing dual parameters.

Primal Methods: Directly optimize reward OR safety performance.

Model-free methods

Primal-dual based Methods:

$$(\pi_*, \lambda_*) = \arg\min_{\lambda} \max_{\pi} \{V_r^{\pi}(\rho) - \lambda(V_c^{\pi}(\rho) - \xi)\}$$

$$\pi_{t+1} = \arg\max_{\pi \in \Pi_0} \{V_r^{\pi} - \lambda_t(V_c^{\pi}(\pi) - \xi)\} \qquad \lambda_{t+1} = \lambda_t + \eta(\xi - V_c^{\pi}(\pi_t))_+$$

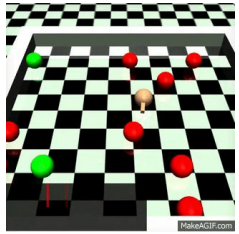
Features:
1. Easy to use.
2. Can be adapted for any RL algorithms.
3. Theoretical guarantees.

Need to further investigate:
1. There is a delay to optimize safety.
2. Sensitive to the initial points.
3. Need to fine tune dual parameters.

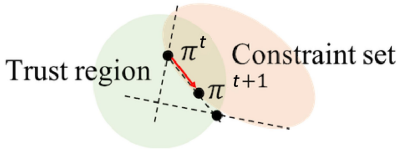
Model-free methods

Primal-Dual methods:

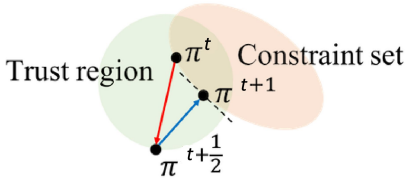
CPO [Achiam et al., 2017] optimize reward and cost as two surrogate models, its optimization is similar to TRPO [Schulman et al., 2015]



PCPO [Yang et al., 2020]



Update procedures for CPO:
CPO computes the update by simultaneously considering the trust region (light green) and the constraint set (light orange).
CPO becomes infeasible when these two sets do not intersect.



Update procedures for PCPO:
In step one (red arrow), PCPO follows the reward improvement direction in the trust region (light green).
In step two (blue arrow), PCPO projects the policy onto the constraint set (light orange).

Model-free methods

TRPO-Lag and **PPO-Lag** [Ray et al., 2019] share the same key idea as follows,

```
Returns:
    The advantage function combined with reward and cost.
"""
penalty = self._lagrange.lagrangian_multiplier.item()
return (adv_r - penalty * adv_c) / (1 + penalty)
```

PPO-Lagrangian method:

- easy to implement and use for safe learning;
- however, it hard to ensure learning safety and stability.

PPO-Lagrangian methods like other **primal-dual based methods**, have some issues, e.g.,

- depley safety optimization;
- sensitive to initial points;
- needs extra to fine-tune hyparameters.

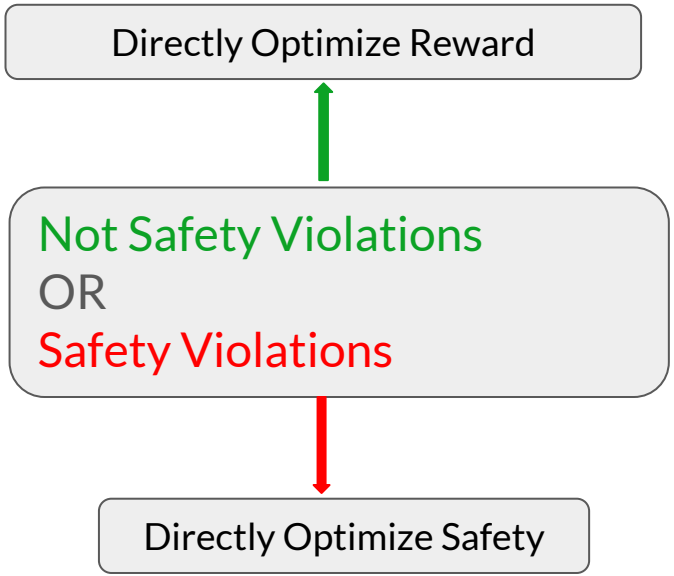
Primal based methods can alleviate these issues by directly optimize safety and reward performance.

Model-free methods

Primal based Methods: Only optimize $\max_{\pi} V_r^{\pi}(\rho)$ if no safety violation happens;
Only optimize $\min_{\pi} V_c^{\pi}(\rho)$ if safety violation happens.

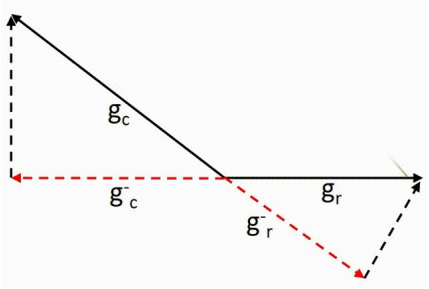
Features:
1. It is not sensitive to the initial points.
2. Do not need to fine tune the dual parameters.
3. Theoretical guarantees.

Need to further investigate:
1. For multiple constraints
2. Sample efficiency

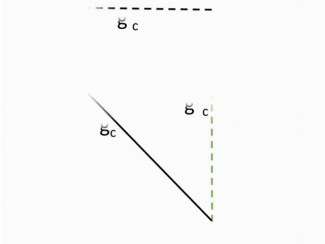


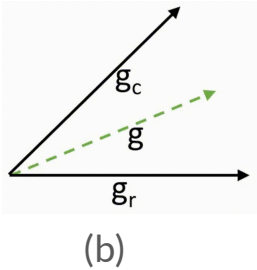
Model-free methods

Primal methods:



CRPO: hard switching via the evaluation of safety violation [Xu et al., 2021].



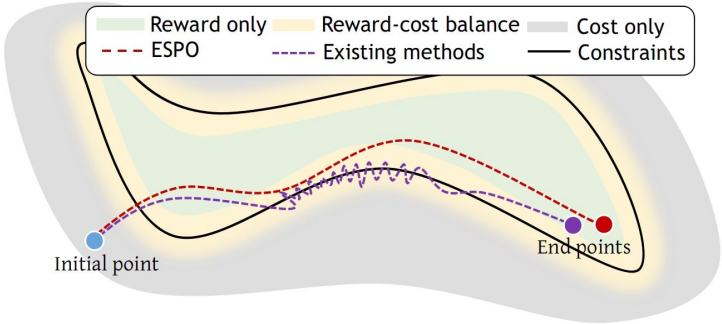


PCRPO: Soft switching through gradient manipulation [Gu et al., 2024a].

Model-free methods

Primal methods:

ESPO [Gu et al., 2024b]

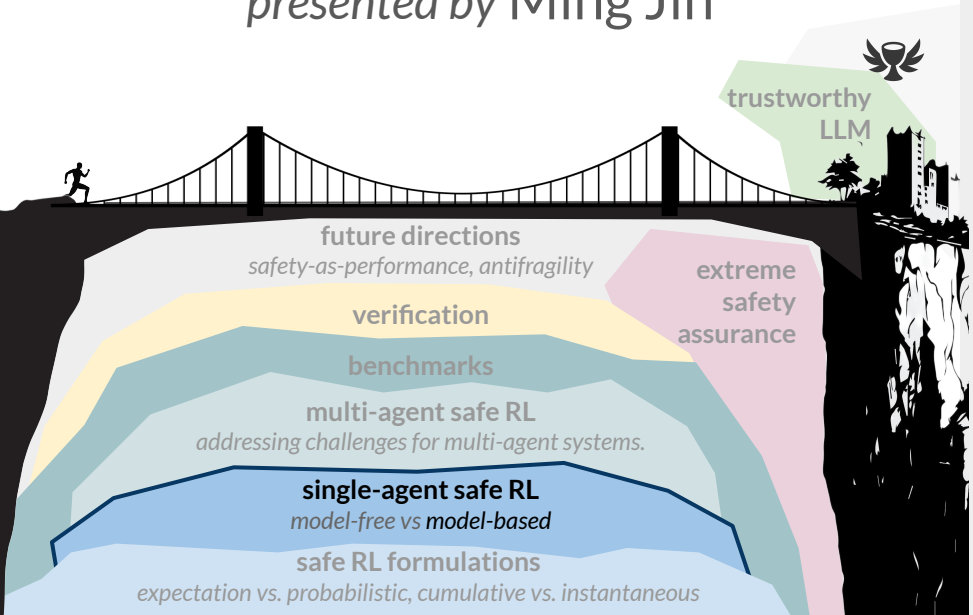


With the metric of gradient conflicts, dynamically adjust sample size during three stages:

- Safety Violations
- Soft Constraint Violations
- No Violations

safe model-based RL

presented by Ming Jin



Key Objectives

- understand the framework of safe model-based reinforcement learning
- explore control theoretic certificates for safe MBRL, including Lyapunov functions, barrier functions, and contraction metrics

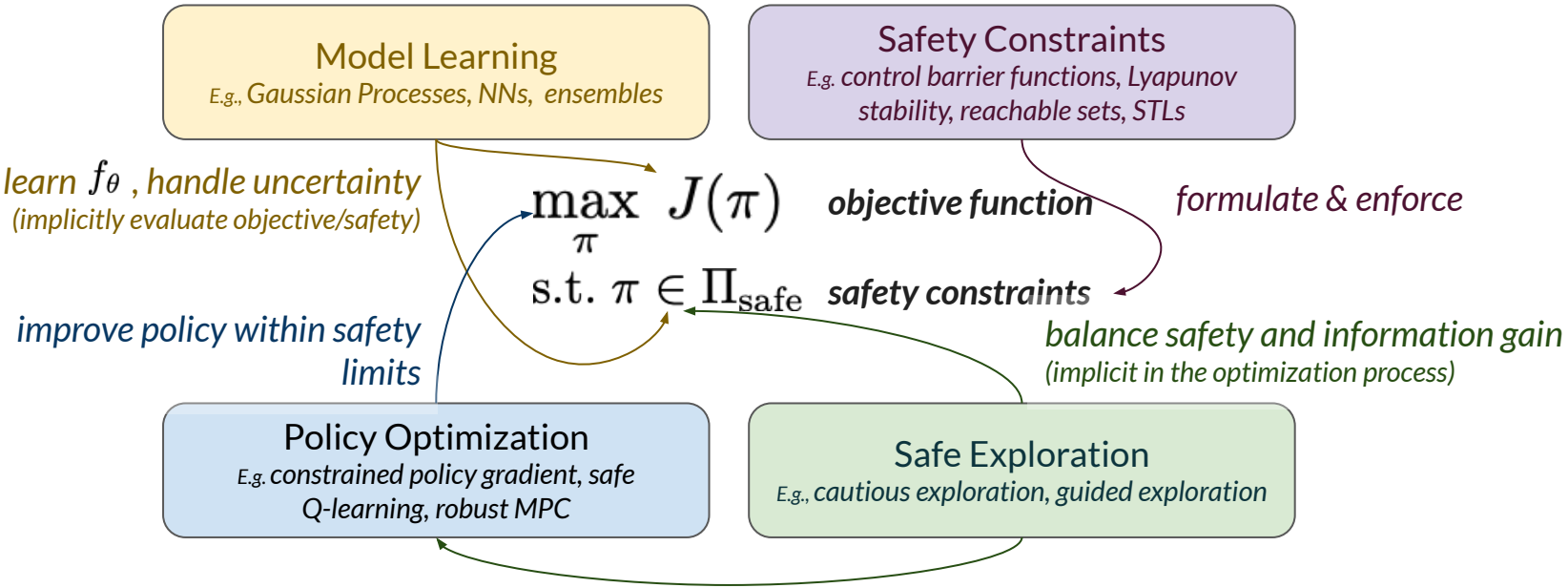
Overview of Safe Model-based RL (MBRL)

Safe MBRL: An approach combining model learning, RL, and safety constraints

model: $s_{t+1} = f_\theta(s_t, a_t, w_t)$

system dynamics

uncertainty/disturbances



Overview of Control Theoretic Notions

system dynamics (deterministic, fully observable): $s_{t+1} = f(s_t, a_t)$ control action: $a_t = \pi(s_t)$

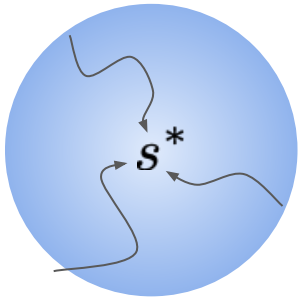
Lyapunov functions

♠ positive definite: $W(s) > 0$ for all $s \neq s^*, W(s^*) = 0$

♠ "dissipates energy": for $s \neq s^*$

$\Delta W(s) = W(f(s, \pi(s))) - W(s) < 0$

ensure system converges to a **desired equilibrium** s^*



Barrier functions

♠ safe set: $S_{\text{safe}} = \{s | h(s) \leq 0\}$

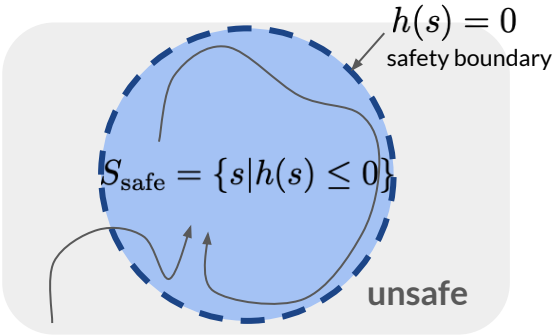
♠ "barrier" condition:

$h(f(s, \pi(s))) - h(s) \leq -\alpha(h(s))$

where $\alpha(\cdot)$ is a class K function (continuous, strictly increasing with $\alpha(0) = 0$)

(Ames, et al., 2019)

ensures system **stay within safe region** & keeps away from unsafe region at each step



Contraction metrics

♠ continuous, positive definite:

$M : \mathbb{R}^n \to \mathbb{R}^{n \times n}$

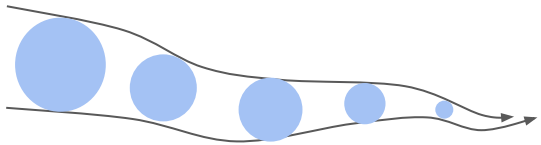
♠ "contraction" condition: for $0 < \beta < 1$,

$\delta s_{t+1}^\top M(s_{t+1}) \delta s_{t+1} \leq \beta \delta s_t^\top M(s_t) \delta s_t$

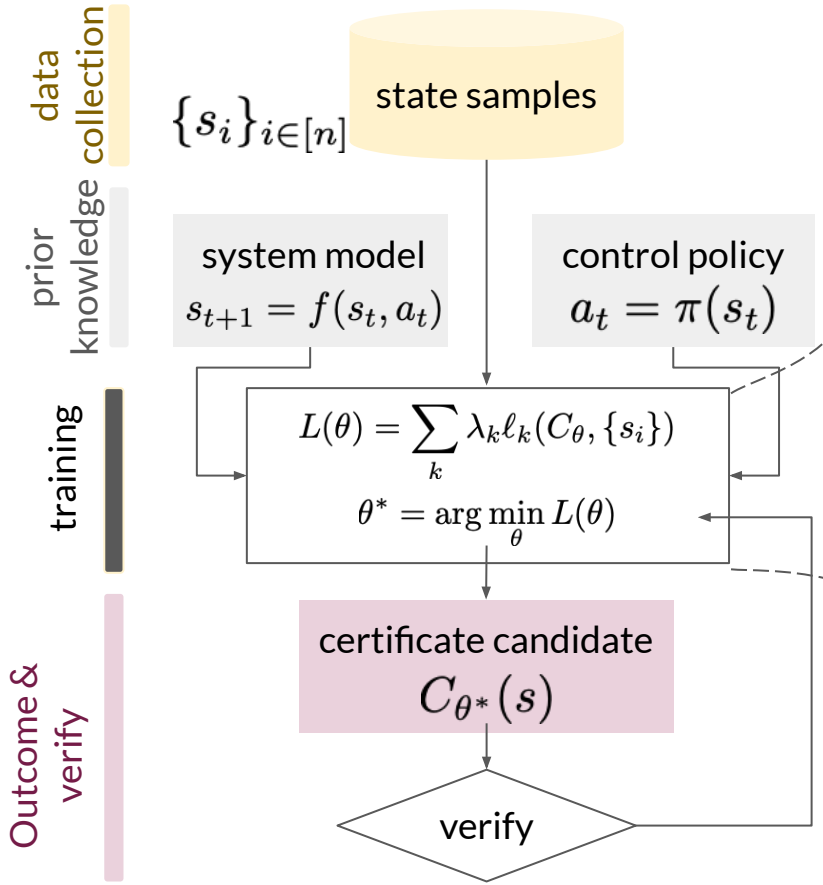
where δs is a virtual displacement between two nearby trajectories.

(Lohmiller & Slotine,, 1998)

guarantees exponential **convergence of nearby trajectories**



Neural Certificate for Control Policy



How to design loss function?

$$L(\theta) = \sum_k \lambda_k \ell_k(C_\theta, \{s_i\})$$

Key insight: convert conditions to loss

- ♠ inequality $a \le b$ → loss $[a-b]_+ := \max(0, a-b)$ or $([a-b]_+)^2$
- ♠ equality $a = b$ becomes loss term $(a-b)^2$

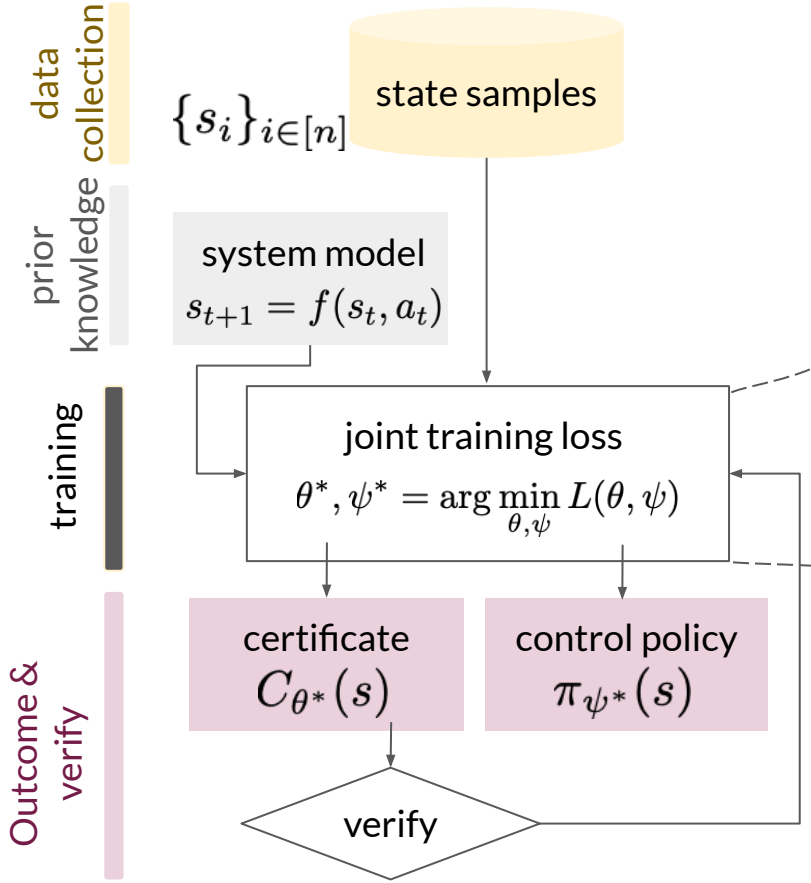
E.g., Learning-based neural Lyapunov loss

$$W(s^*) = 0 \qquad W(s) \ge 0 \qquad W(f(s,\pi(s))) - W(s) < 0$$

$$\lambda_1 W_\theta(s^*) + \lambda_2 \sum_i [-W_\theta(s_i)]_+ + \lambda_3 [W_\theta(f(s_i, a_i)) - W_\theta(s_i)]_+$$

Key Considerations:
1) sample coverage (focus on critical regions and safe set boundaries), **2)** balance loss terms (adjust weights λ_k for different conditions), **3)** add small margin for robustness, **4)** online update during execution [Qin et al, 2020]

Joint Neural Certificate and Policy Learning



How to design **joint** loss function?

$$L(\theta, \psi) = L_{\text{cer}}(\theta, \psi) + \lambda L_{\text{policy}}(\psi)$$

$L_{\text{cer}}(\theta, \psi)$: same as before, but with a_t replaced by $\pi_\psi(s_t)$

$$L_{\text{policy}}(\psi) = -V(\pi_\psi)$$

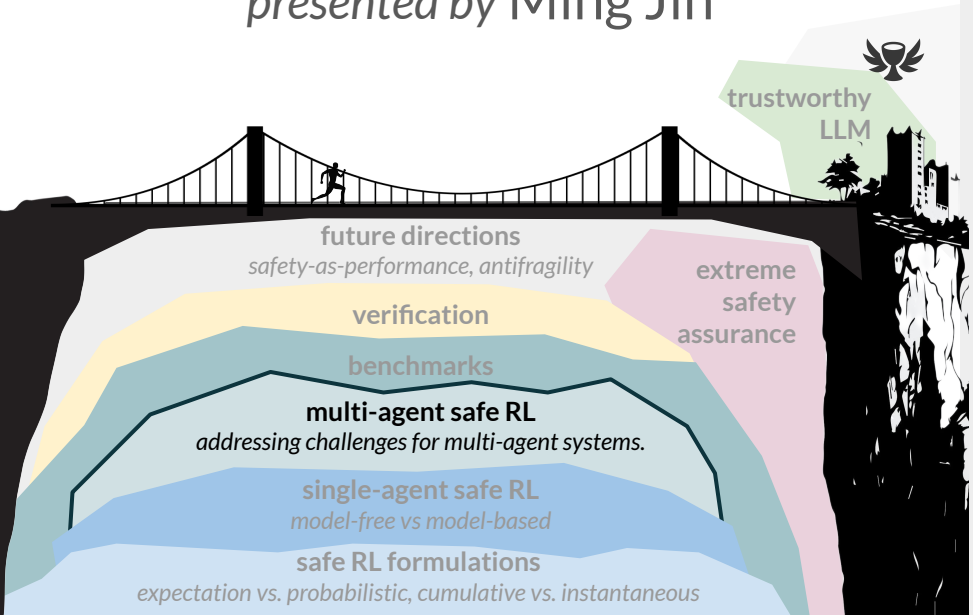
$$:= -\mathbb{E}\left[\sum_t \gamma^t r(s_t, \pi_\psi(s_t))\right]$$

- ♠ Optimize θ and ψ simultaneously
- ♠ Verify: learner-falsifier framework (Chang et al., 2019)

Neural Lyapunov functions: (Richards et al., 2018) (Chang et al., 2019) (Choi et al., 2020) (Taylor et al., 2019); **Learning barrier functions:** (Srinivasan et al., 2020), (Qin et al., 2021); **Learning contraction metrics:** (Singh et al., 2020); **Verification:**(Abate et al., 2021) (Yang et al., 2023); **Survey:** (Dawson et al., 2022)

safe multi-agent RL

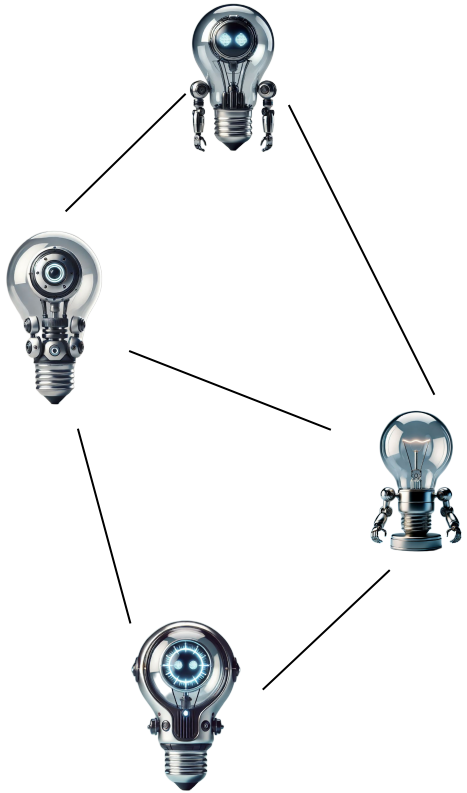
presented by Ming Jin

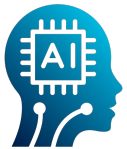


Key Objectives

- understand the three key challenges of MARL: perception, learning, and interaction challenges, and the approaches addresssing them
- examine the concept of safety in MARL, including global and local safety definitions
- analyze advanced techniques for safe MARL, including Lagrangian methods and control-theoretic approaches

Multi-Agent Systems





MDPs (Markov Decision Processes):
formalizing single-agent decisions

Goal: find the optimal policy that maximizes the **expected cumulative reward over time**.

$$\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

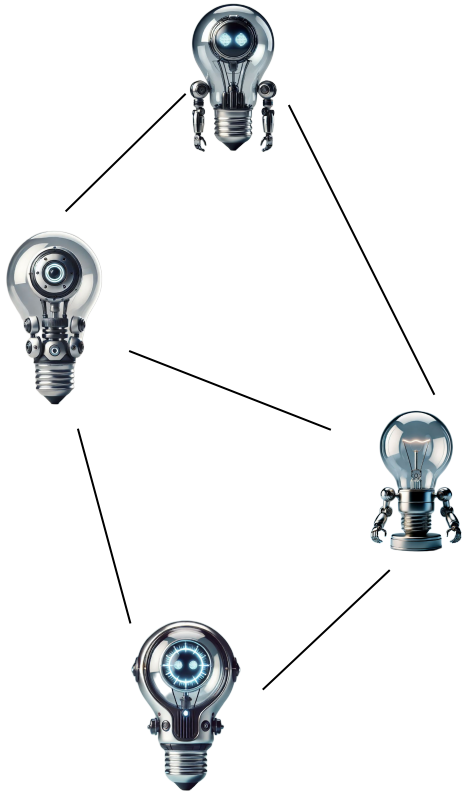
$\mathcal{S}$ **state:** the current snapshot of the system

$\mathcal{A}$ **action:** the choices the agent can make

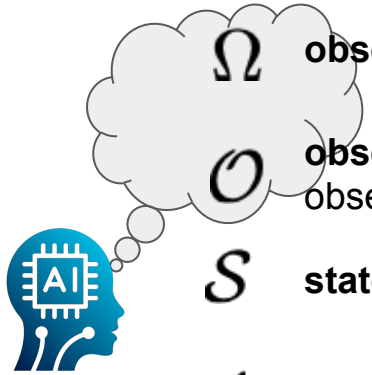
r **reward:** the feedback that guides learning

$\mathcal{P}$ **transition:** how actions change the state

Multi-Agent Systems



POMDPs (Partially Observable MDPs)
single-agent with *incomplete* information about states



Ω **observation:** the set of all possible measurements

$\mathcal{O}$ **observation probability:** the probability of observation given that the system state and action

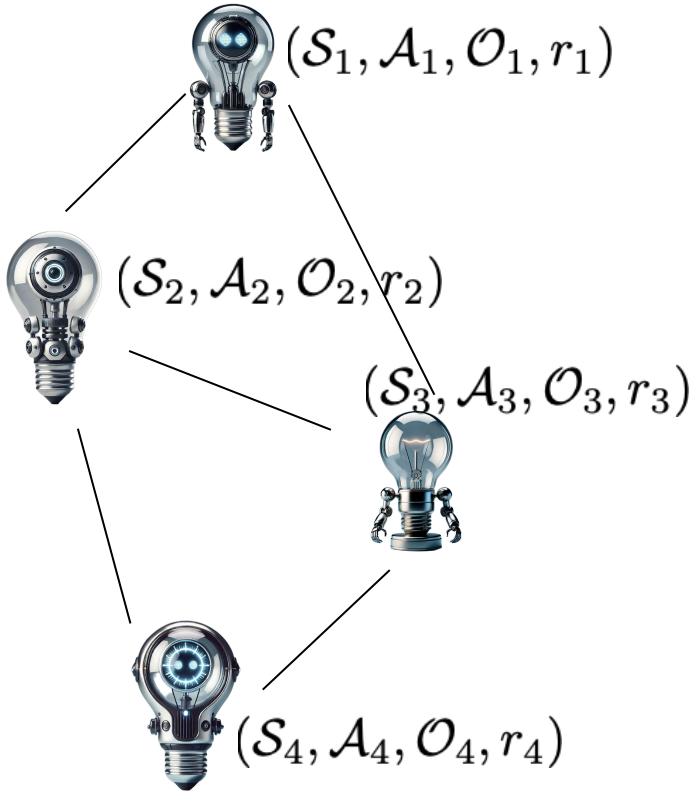
$\mathcal{S}$ **state:** the current snapshot of the system

$\mathcal{A}$ **action:** the choices the agent can make

r **reward:** the feedback that guides learning

$\mathcal{P}$ **transition:** how actions change the state

Multi-Agent Systems



Dec-POMDPs (Decentralized POMDPs)
a framework for modeling multi-agent systems where agents have limited information with **individual rewards**

Goal: find the optimal **joint** policy that maximizes the **expected cumulative reward over time**.

$$\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

$\mathcal{K}$ **set of agents**

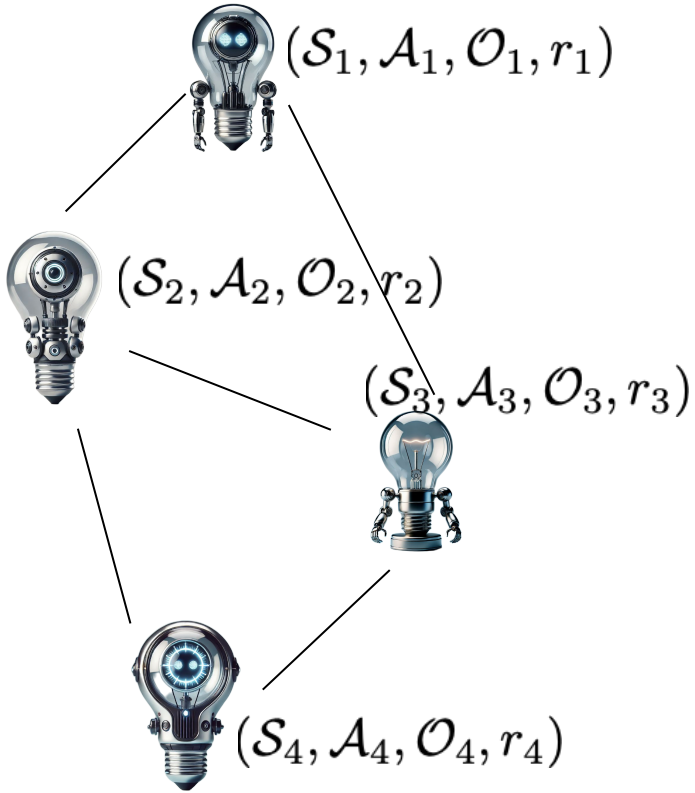
$$\mathcal{O} = \times_k \mathcal{O}_k$$
$$\mathcal{S} = \times_k \mathcal{S}_k$$
$$\mathcal{A} = \times_k \mathcal{A}_k$$

joint observation, state, and action spaces

$$(r_1, \ldots, r_K)$$

common reward for **fully cooperative** Dec-POMDP

Multi-Agent Reinforcement Learning (MARL)



Dec-POMDPs (Decentralized POMDPs)

a framework for modeling multi-agent systems where agents have limited information with **individual rewards**

Goal: find the optimal **joint** policy that maximizes the **expected cumulative reward over time**.

$$\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

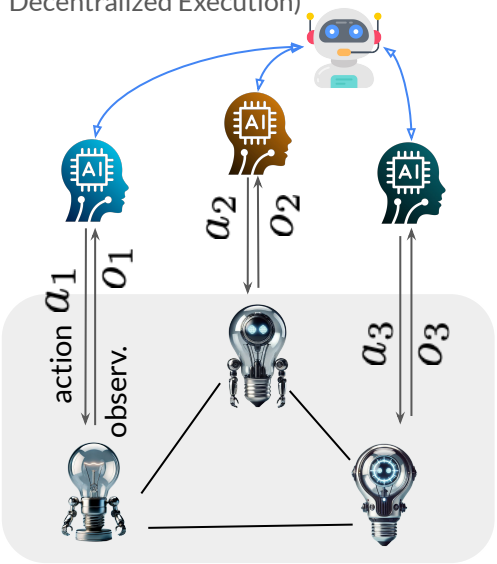
MARL: multiple agents learn simultaneously in shared environment for solving Dec-POMDP

e.g., **IQL** (Independent Q-Learning), **MAAC** (Multi-Agent Actor-Critic), **MADDPG** (Multi-Agent Deep Deterministic Policy Gradient), **QMIX**

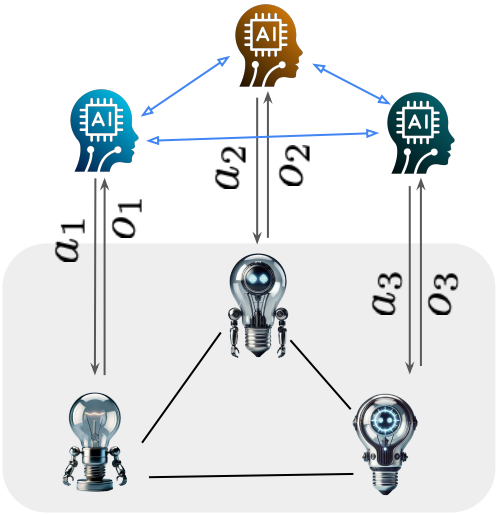
…enables these agents to learn and adapt to complex, dynamic environments

Training Regimes for MARL

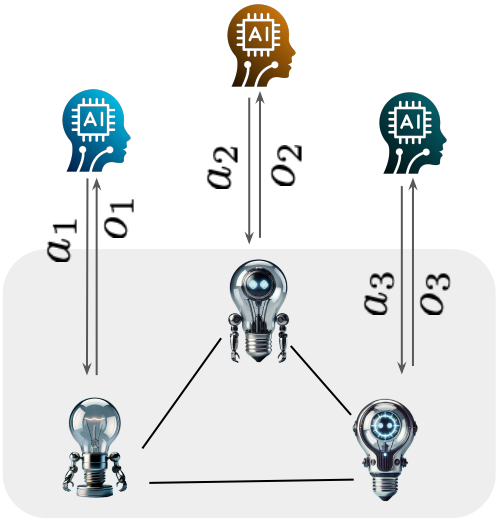
CTDE (**Centralized Training** with Decentralized Execution)



networked training



decentralized training



	CTDE	networked training	decentralized training
learning speed	high	medium	low
comm. require.	high	medium	low
scalability	low	medium	high

The MARL Maze: Navigating the Challenges

partial observability
limited information hinders decision-making

coordination
achieving cooperation with limited or no communication

category 1: ***perception challenge***

category 3: ***interaction challenge***

non-stationarity
environment changes due to concurrent learning

scalability
complexity increases exponentially with #agents

credit assignment
difficult to attribute individual contributions to rewards

exploration vs. exploitation
balancing learning new strategies with using known ones

category 2: ***learning challenge***

Tackling Perception Challenge (1/3)

 partial observability  non-stationarity

RNNs and memory-based approaches

maintains an internal memory of histories

integrates past information with current observations

helps capture latent state and predict future trends

e.g., Deep Recurrent Q-Net. [Hausknecht & Stone, 2015], Recurrent-MADDPG [Suttle et al., 2019]

+ PROS: relatively simple to implement, GRUs (e.g., Graph-MARL}, LSTMs (e.g., PowerNet)

— CONS: can be difficult to train, esp. for long sequences

Attention mechanisms

attends to relevant parts of observations

filters out noise and prioritizes critical information

e.g., MAAC (Multi-Actor-Attention-Critic) [Iqbal & Sha, 2019], Attention-based Optimizer for Continuous Control [Jiang & Lu, 2018]

+ PROS: efficient for large state spaces and long-range dependencies

— CONS: can be computationally expensive or over-flexible

Agent modeling and belief states

explicitly model other agents' behaviors or maintain beliefs about the true state of the environment

e.g., DRON [He et al., 2016], Bayes-Adaptive POMDP [Ross et al., 2011]

+ PROS: theoretically sound, effective for systems with a few agents

— CONS: may not scale well, sensitive to modeling assumptions

Tackling Perception Challenge (2/3)

 partial observability  non-stationarity

Multi-agent communication

allows agents to share information

can lead to emergent team strategies

e.g., Deep CommNet [Sukhbaatar et al., 2016], DIAL (Differentiable Inter-Agent Learning) [Foerster et al., 2016]

+ PROS: can significantly improve performance in cooperative tasks

— CONS: learned protocols may not be interpretable

CTDE (Central. Training with Decentral. Execution)

leverages global information during training

agents can learn to coordinate more effectively

e.g., MADDPG and variants

+ PROS: can stabilize training

— CONS: may not fully capture decentralized dynamics

Experience Replay Modifications

store and replay sequences of observations to help with temporal dependencies

e.g., Importance Sampling [Foerster et al., 2017], Multi-agent Fingerprints [Foerster et al., 2017]

+ PROS: helps stabilize learning and sample efficiency, can combine with other methods

— CONS: may slow down learning in rapidly changing environments

Tackling Perception Challenge (3/3)


partial observability


non-stationarity

	Scalability *ability to handle increasing # of agents*	**Comp. Complex.** *resource requirements for training & execution*	**Sample Efficiency** *learning from limited interactions*	**Interpretability** *ease of explaining the learned policies/behaviors*
Memory-based (RNNs)	◐		⃠	
Attention mechanisms		⃠	⃠	◐
Agent modeling/belief state	⃠		◐	◐
Communication	◐		◐	
CTDE	●	◐	●	
Experience replay mod.	◐	◐	●	

⃠ limitation ◐ good ● excellent

Key takeaways:

- most address both partial observability and non-stationarity, but no single "best" method
- consider tradeoffs based on specific problems (e.g., #agents, available data/compute)
- hybrid approaches (combine CTDE with other methods) can be effective.

Enhancing Learning Efficiency (1/3)


credit assignment


explore vs. exploit

…leveraging shared methods for synergistic improvement

1

CTDE
(Central. Training w/ Decentral. Execution)

more accurate individual contributions and coordinate exploration strategies

e.g., **central critic:** MADDPG, COMA with Exploration Curriculum [Wang et al., 2020]
central latent space: MAVEN (Multi-Agent Variational Exploration) [Mahajan et al., 2019]

2

Attention mechanism

focus on relevant information, weighing contributions and guiding exploration

e.g., MAAC use attention-based critic to implicitly assign credits

3

Communi cation

facilitate the sharing of reward information and coordinated exploration

e.g., MADDPG-M (MADDPG with Multi-Agent Coordinated Exploration) [Chu et al., 2020] uses communication specifically for coordinating exploration

Enhancing Learning Efficiency (2/3)

credit assignment

explore vs. exploit

Credit Assignment Approaches

Difference rewards/Shapley value

quantifying Individual Impact

by comparing the overall reward with and without their actions

e.g., Wonderful Life Utility [Wolpert & Tumer, 2002], Difference Rewards [Tumer & Agogino, 2007], Shapley Value based, e.g., SQDDPG [Wang et al., 2020]

PROS: clear, intuitive, theoretically sound, effective for cooperative tasks

CONS: not easily scalable, may struggle with interdependent tasks.

Value decomposition

decomposes the team's value function into individual agent value functions

e.g., VDN (Value Decomposition Networks) [Sunehag et al., 2018], QMIX [Rashid et al., 2018], MAAC [Lowe et al., 2017]

PROS: scalable, allows for decentralized execution, flexible (QMIX)

CONS: assumes additivity (VDN) or monotonicity (QMIX), may struggle with non-linear interactions

Enhancing Learning Efficiency (3/3)


credit assignment


explore vs. exploit

Exploration Approaches

Diversity-driven exploration/ parameter space exploration (PSN)

promote diverse behaviors and actions within and across agents

e.g., NoisyNet adapted for MARL (adds noise to weights) [Plappert et al., 2018], Population-Based Training with Diversity [Jaderberg et al., 2019]

\+ PROS: PSN useful for continuous action spaces, avoid premature convergence

— CONS: requires diversity metric, can be computationally demanding.

Safe exploration

explore while adhering to safety constraints

e.g., Safe MARL with Safety Constraints [Castellini et al., 2021]

\+ PROS: enables learning in high-stakes, risk-sensitive environments, provides safety guarantees during training

— CONS: can limit exploration space, may increase computational complexity due to constraint checking

Addressing Scalability Challenge (1/2)


scalability


coordination

1

Graph neural networks

model agent interactions using graphs

PROS naturally models spatial relationships, scales to different numbers of agents

e.g., DGN (Graph Convolutional RL) [Jiang et al., 2020], MAGNet (Multi-Agent Graph Network) [Malysheva et al., 2019], MAGEC [Goeckner et al., 2024]

CONS can be computationally expensive

2

Hierarchical methods

use multi-level structures for coordination

agents can learn to coordinate more effectively

PROS enables multi-level coordination

e.g., HAMA (Hierarchical Multiagent Framework) [Tang et al., 2018], FEU (Feudal Ensemble Update) [Ahuja et al., 2019]

CONS designing effective hierarchies can be challenging

3

Mean field methods

approximate interactions with many agents using average effects

PROS extremely scalable to large numbers of agents, computationally efficient

e.g., Mean Field MARL [Yang et al., 2018]

CONS may oversimplify scenarios w/ heterogeneous agents

Addressing Scalability Challenge (2/2)


scalability


coordination

1

Decomposition

value decomp. factored MDPs

Exploit problem structure to factor state and action spaces or decompose value functions

e.g., VDN, QMIX, Factored-Value MARL
[Bargiacchi et al., 2018]

2

Attention mechanism

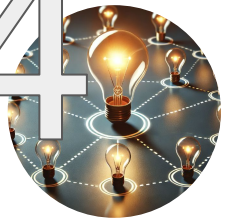
selectively focus on relevant agents or information.

e.g., MAAC (attention in critic), ATOC (attentional communication)

3

Communi cation

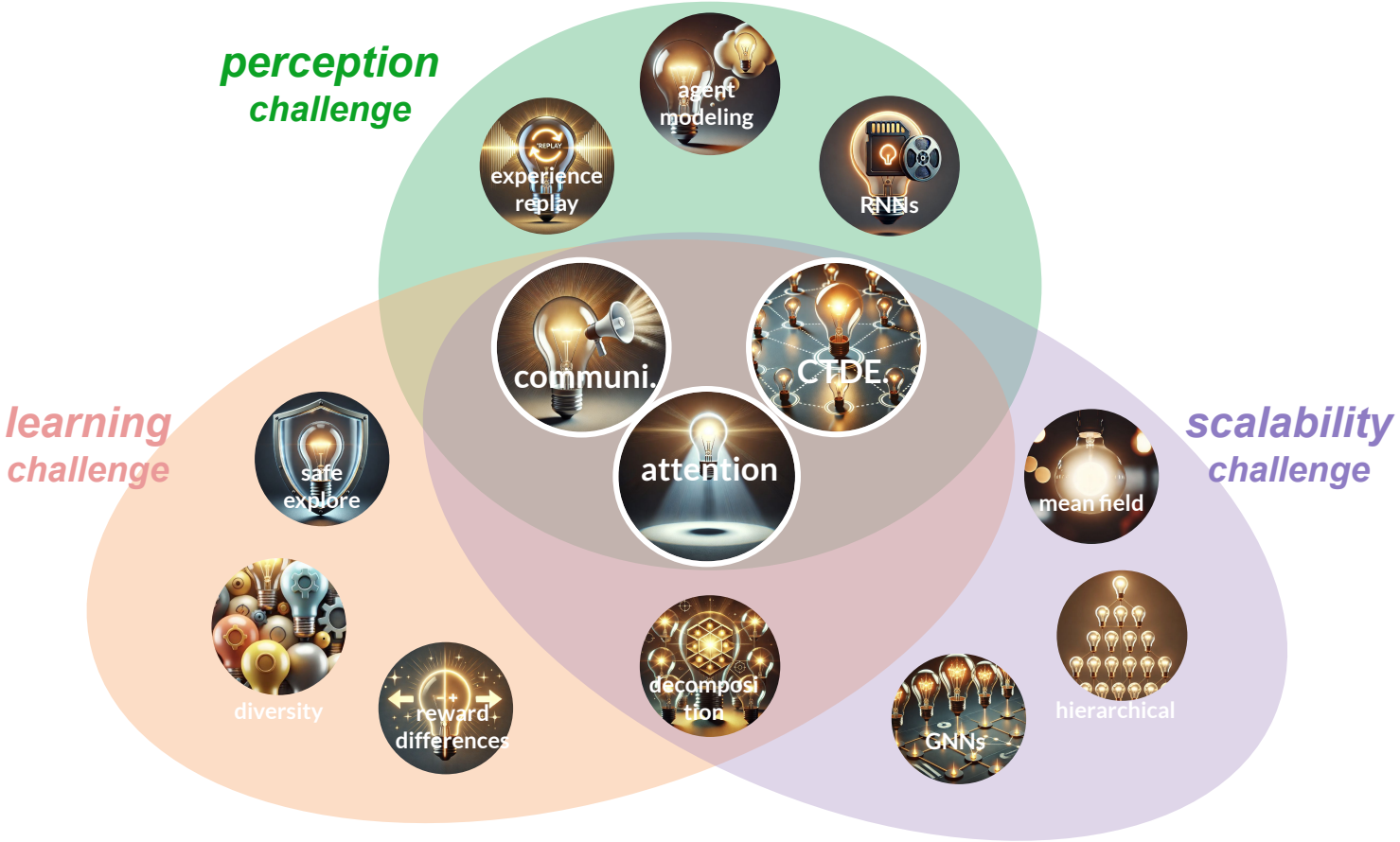
4



CTDE
(Central. Training w/ Decentral. Execution)

e.g., parameter sharing [Mu et al., 2023] [Sun and Lu, 2024]

MARL Methods: A Multi-Purpose Toolkit



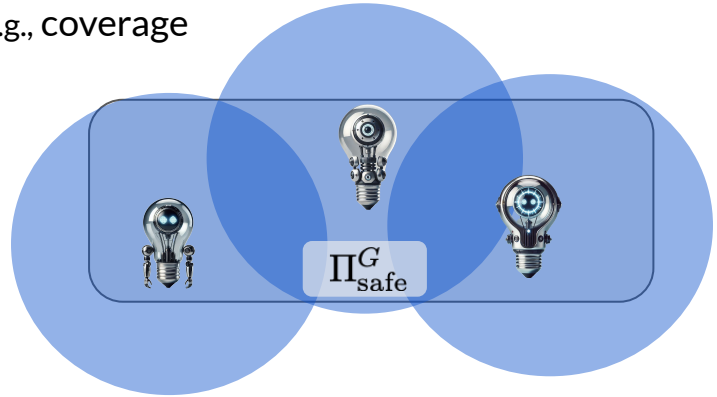
Multi-Agent Safety Definitions

Global safety

treats entire multi-agent system as one entity

joint policy constraint: $\pi = (\pi_1, ..., \pi_K) \in \Pi^G_{\text{safe}}$

E.g., coverage

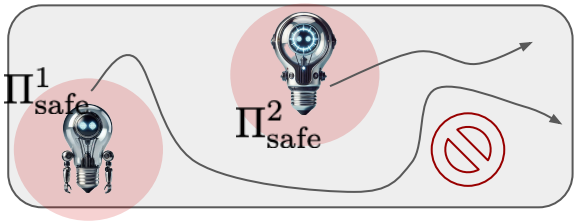


Local safety

each agent has its own safety constraint

individual constraints: $\pi_k \in \Pi^k_{\text{safe}}, \quad \forall k$

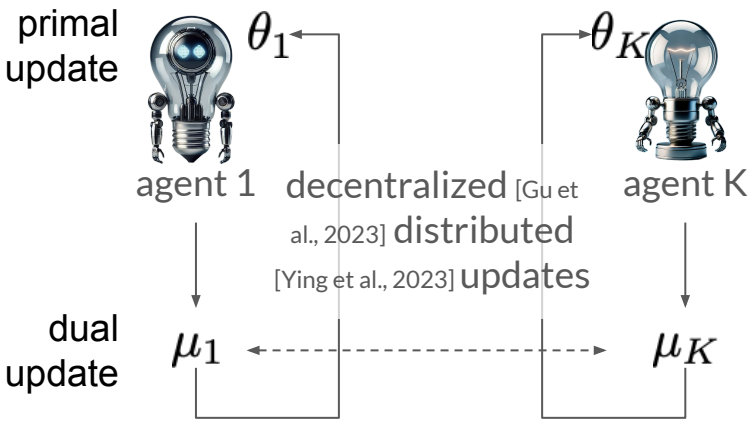
E.g., collision avoidance



♠ coverage vs collision avoidance: environmental monitoring, warehouse robotics, agricultural drones

♠ **smart grid:** global (system stability) vs local (voltage limits)

Lagrangian Methods for Safe MARL



safe MARL formulation with **general utilities**

$$\max_{\theta:=\{\theta_k\}_{k\in\mathcal{K}}} F(\theta) = \sum_k f_k(\lambda^{\theta_k})$$

$$\text{s.t. } G_k(\theta) = g_k(\lambda^{\theta_k}) \le 0 \quad \forall k \in \mathcal{K}$$

joint policy parameters

local occupancy measure of agent k

local objective/constraint

*general utilities f_k and g_k can be arbitrary (potentially nonlinear) functions for imitation learning, exploration.

*standard cumulative rewards/costs:

$f_k(\lambda^{\pi_k}) = \langle r_k, \lambda^{\pi_k} \rangle$ (where r_k is the local reward vector)

Lagrangian formulation w/ primal-dual updates

$$\max_{\theta} \min_{\mu \ge 0} L(\theta, \mu) = F(\theta) + \sum_k \mu_k G_k(\theta)$$

- primal update: each agent updates local policy θ_k
- dual updates: centralized [Gu et al., 2023] or decentralized (using shadow rewards and m-hop policy truncation) [Ying et al., 2023]

Distributed Optimization for Safe MARL

- use distributed optimization for constrained policy optimization
- incorporate safety constraints as coupling constraints across agents

CBF-induced central filter

$$\min_{u_1,\dots,u_K} \sum_{k=1}^{K} \frac{1}{2}\|u_k - u_{\text{nom},k}\|^2$$

$$\text{s.t.} \sum_{k=1}^{K} A_k u_k = c$$

♠ introduce auxiliary y_k variable for each agent k
♠ local exchange of auxiliary variables

auxiliary variable decomposition

for each agent k:

$$\min_{u_k} \|u_k - u_{\text{nom},k}\|^2$$

$$\text{s.t. } A_k u_k + \sum_{j \in N_k} (y_k - y_j) = c$$

? distributed decomposition, e.g., ADMM (Lagrangian-based)

key challenge: maintain safety during **every iteration** (violation-free/feasible algorithms)

safety at every iteration: sum of local constraints across all agents equals the global constraint:

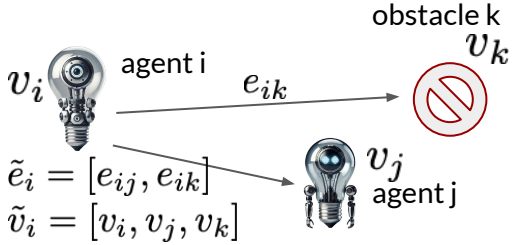
$$\sum_{k=1}^{K} \sum_{j \in N_k} (y_k - y_j) = 0$$

Tan and Dimarogonas (2022), Bai et al. (2024)

future directions: integration with deep RL, robust formulations for uncertain dynamics

Graph CBFs for MAS

- learning-based approach using Graph Neural Networks
- handles general nonlinear dynamics: $\dot{x}_k = f_k(x_k, u_k)$
- key advantages:
  - scalable to large numbers of agents
  - handles variable number of neighbors
  - distinguishes between agent types
  - generalizes to unseen scenarios



Graph CBF conditions

$h(\tilde{e}_i, \tilde{v}_i) > 0$ for all safe states

$h(\tilde{e}_i, \tilde{v}_i) < 0$ for all unsafe states

$h(\tilde{e}_i, \tilde{v}_i) \geq -\alpha(h(\tilde{e}_i, \tilde{v}_i))$ for all safe states

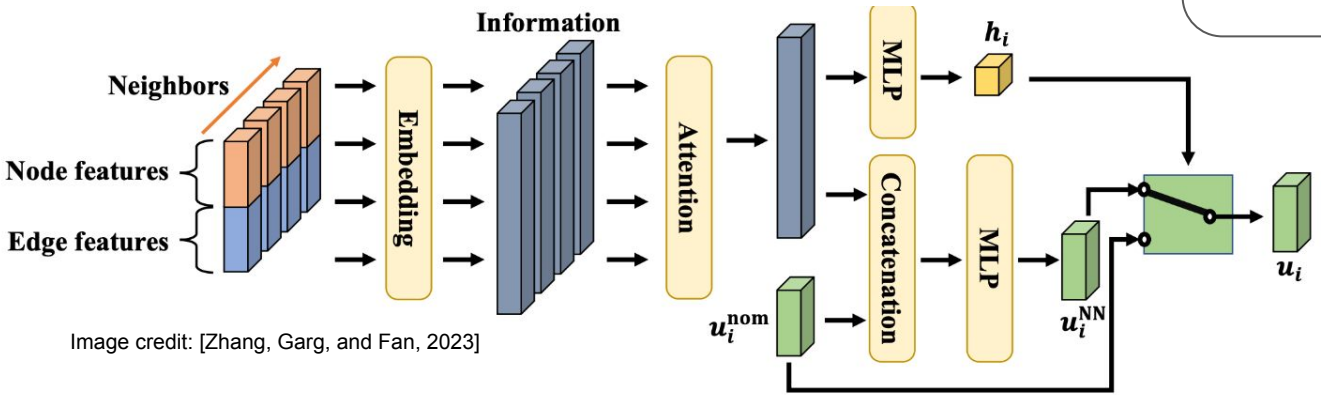


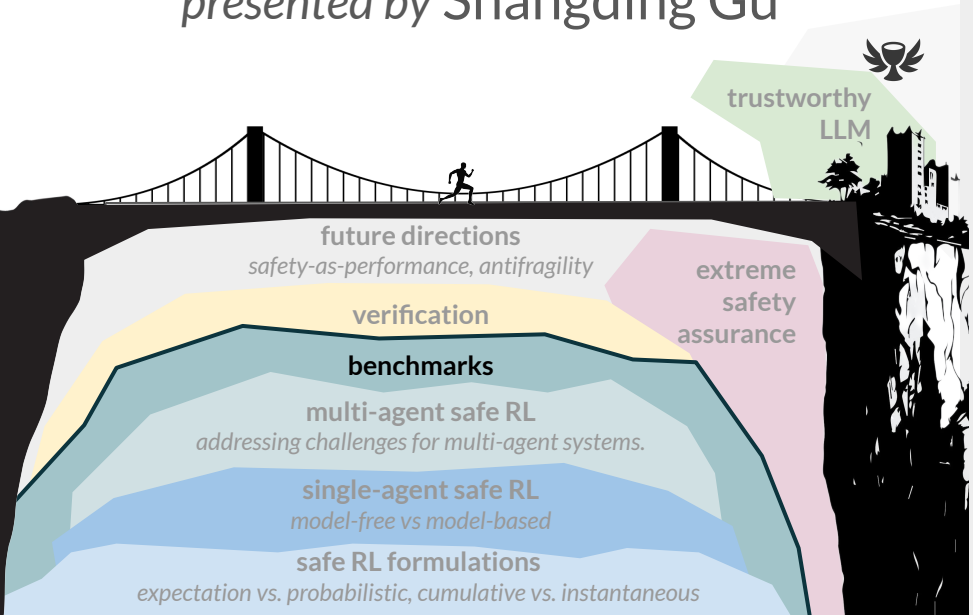
Image credit: [Zhang, Garg, and Fan, 2023]

Trained with only 16 agents, GCBF can still maintain more than 50% success rates for >1000 agents when other methods completely fail.

See also: [Tolstaya et al., 2021], [Li et al., 2020], [Yu et al., 2023], [Blumenkamp et al., 2022], [Jia et al., 2022], [Ji et al., 2021]

benchmarks

presented by Shangding Gu



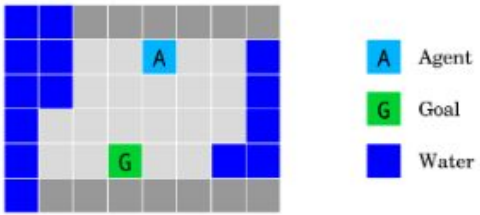
Key Objectives

- Introduce safe reinforcement learning benchmarks: mujoco, robot manipulation and isaac gym.
- Introduce safe robust reinforcement learning benchmarks.

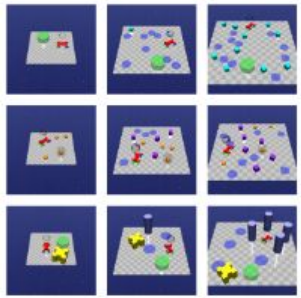
Safe Single Agent RL Benchmarks:

- AI Safety Gridworlds: discrete environments.
- Safety Gym: support continuous space.
- Safe Control Gym: consider symbolic constraints.
- Safety Gymnasium: extent safety gym to new mujoco and gym API.

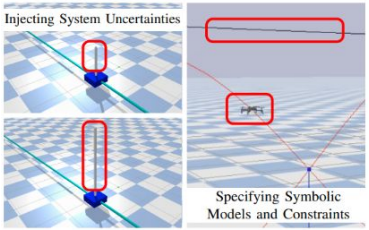
Scan the code to access the source



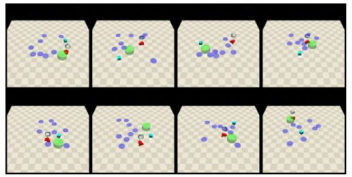
AI Safety Gridworlds



Safety Gym



Safe Control Gym

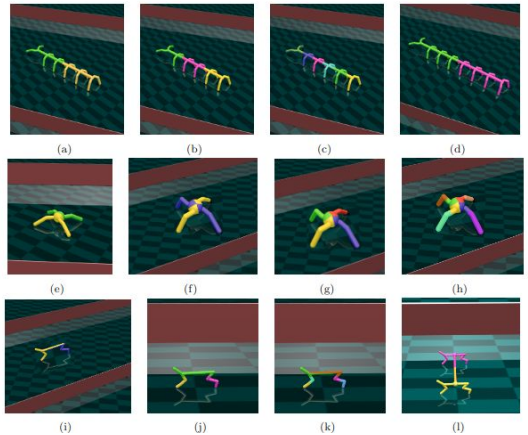


Safety Gymnasium

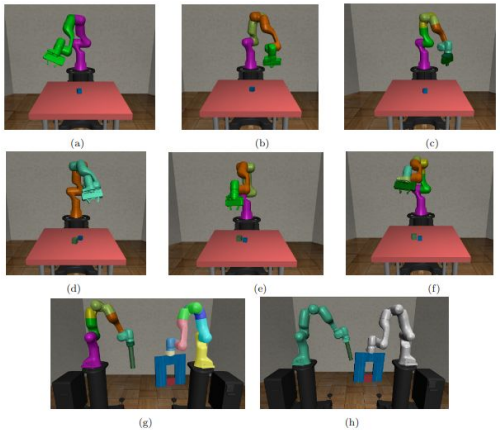
Safe Multi-Agent RL Benchmarks:

- Example tasks in Safe Multi-Agent Environments.
- Body parts of different colours are controlled by different agents.
- Avoiding crashing into unsafe red areas.

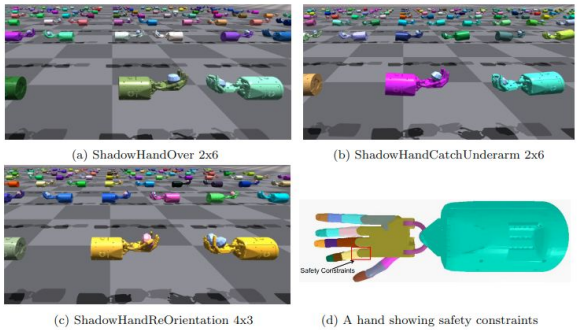
Scan the code to access the source



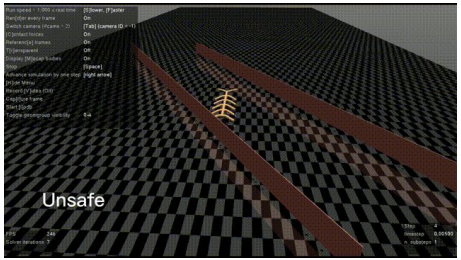
Safe Multi-Agent Mujoco Benchmarks

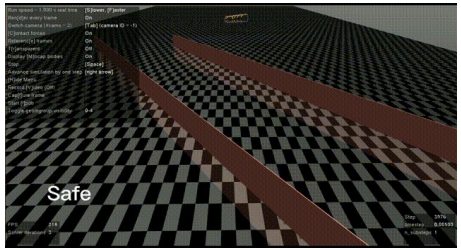


Safe Multi-Agent Robosuite Benchmarks

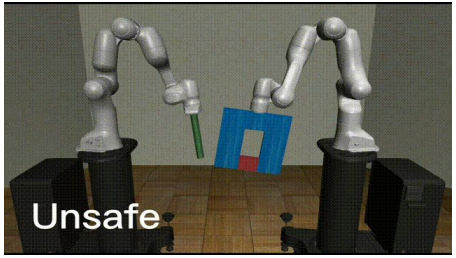


Safe multi-agent Isaac Gym Benchmarks

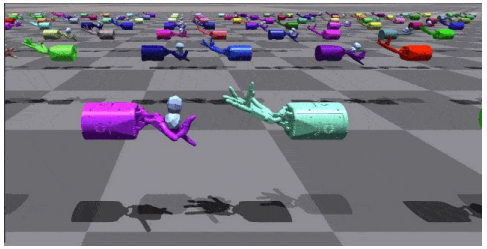


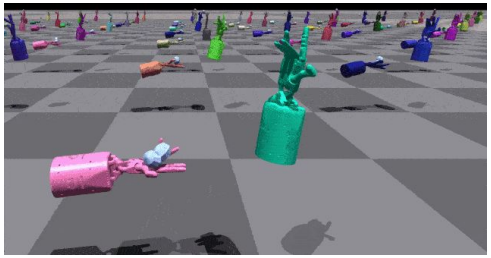


Safe Multi-Agent Mujoco Benchmark



Safe Multi-Agent Robosuite Benchmarks





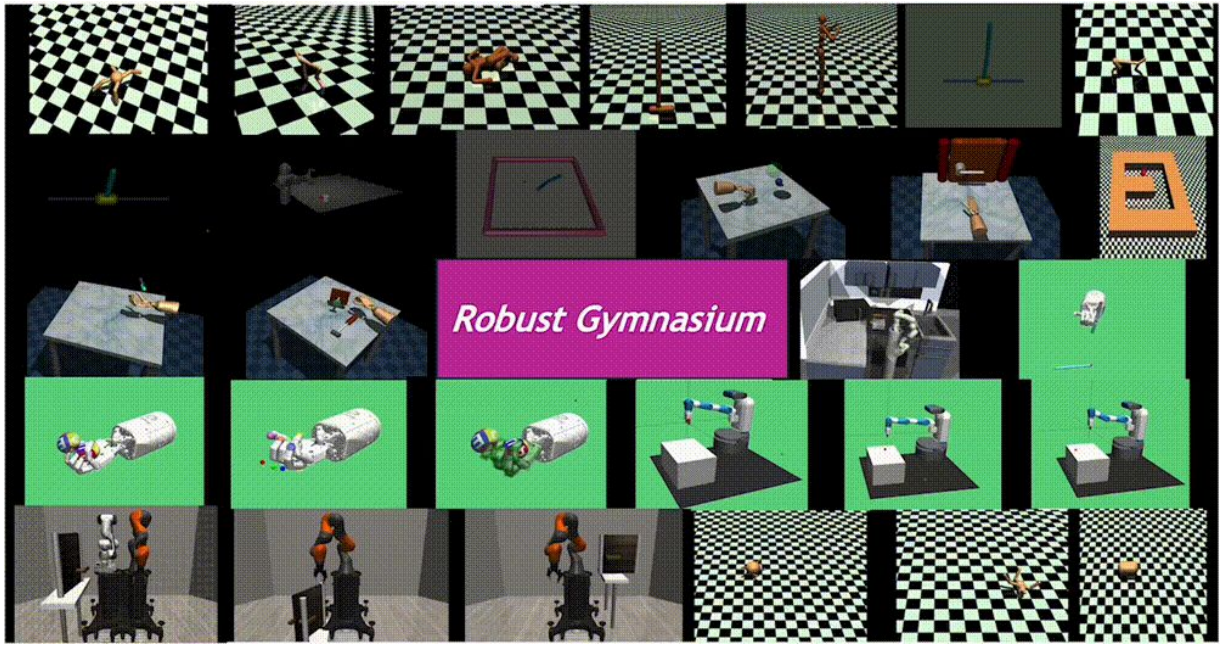
Safe multi-agent Isaac Gym environments

🔥 **Benchmark Features:**

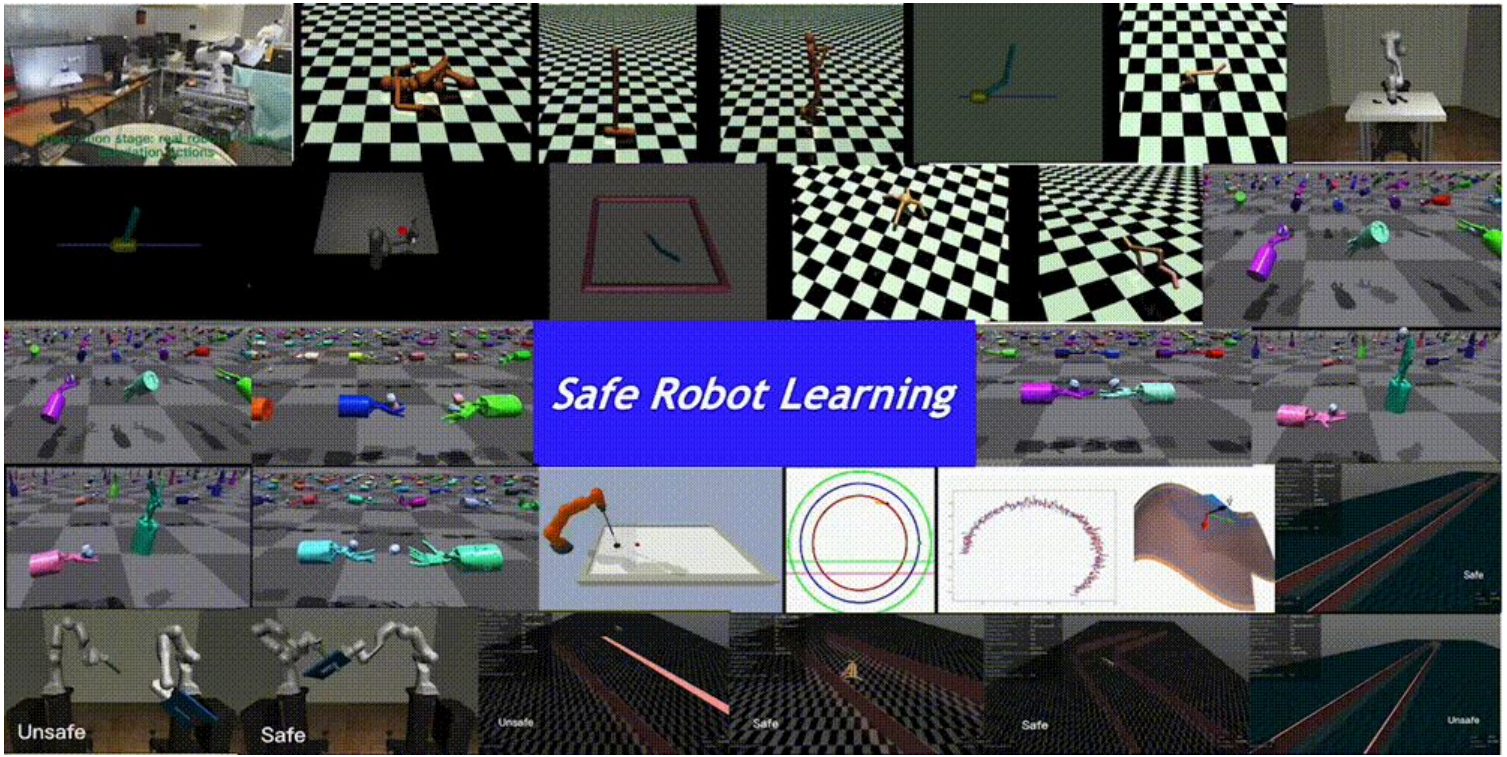
- High Modularity
- Task Coverage
- High Compatibility
- Support Vectorized Environments
- Support for New Gym API
- LLMs Guide Robust Learning

🔥 **Benchmark Tasks:**

- Robust MuJoCo Tasks
- Robust Box2D Tasks
- Robust Robot Manipulation Tasks
- Robust Safety Tasks
- Robust Android Hand Tasks
- Robust Dexterous Tasks
- Robust Fetch Manipulation Tasks
- Robust Robot Kitchen Tasks
- Robust Maze Tasks
- Robust Multi-Agent Tasks



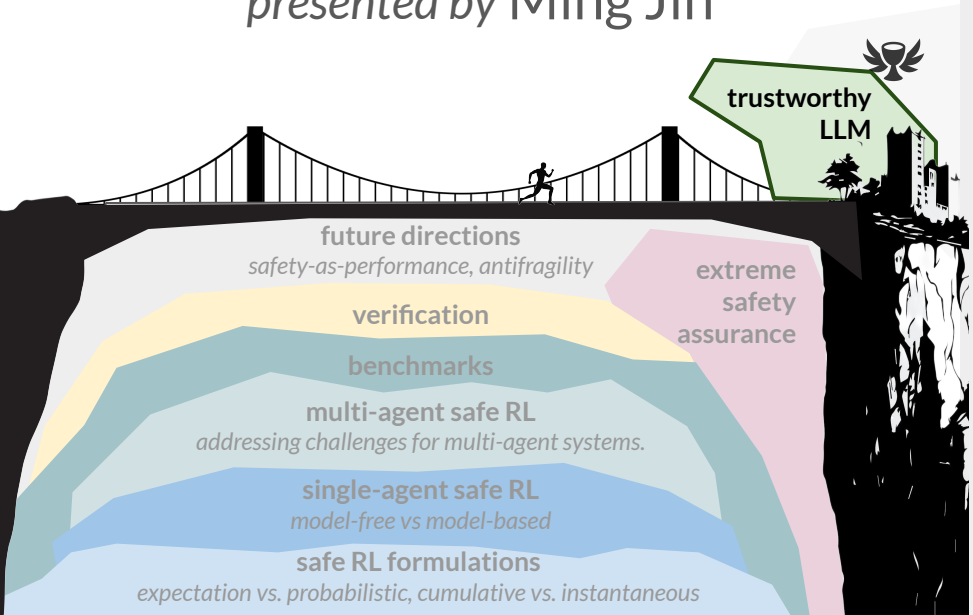
We released open resources for safe robust RL:
https://github.com/SafeRL-Lab/Robust-Gymnasium



We released open resources for safe RL: https://github.com/chauncygu

towards trustworthy LLM

presented by Ming Jin

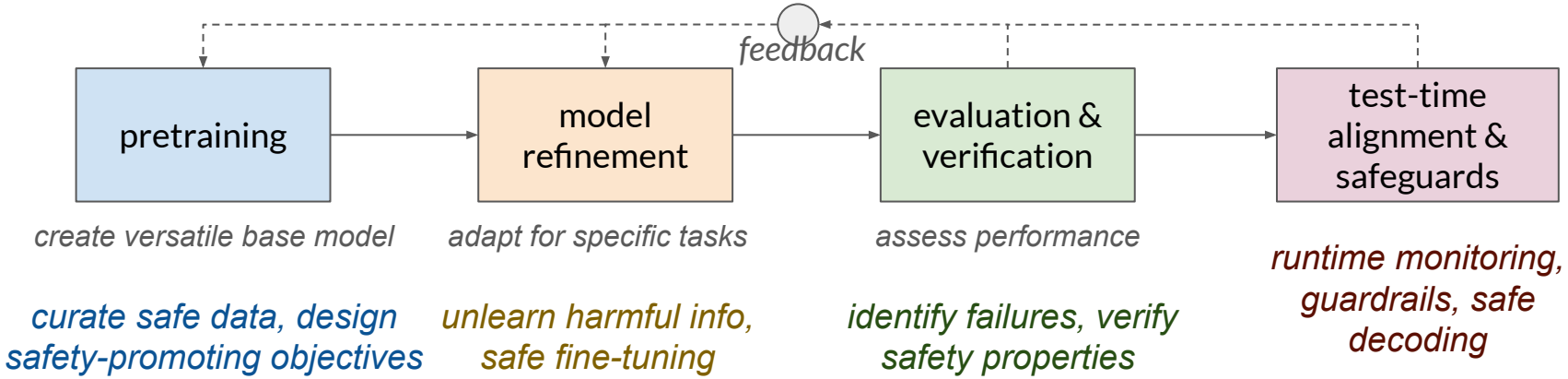


Key Objectives

- understand the lifecycle approach to LLM safety and alignment
- explore Safe RL applications:
  - balancing helpfulness and harmlessness in LLMs
  - falsification and evaluation
  - test-time alignment

LLM Safety and Alignment Lifecycle

ensuring that LLMs behave safely and consistent with human values

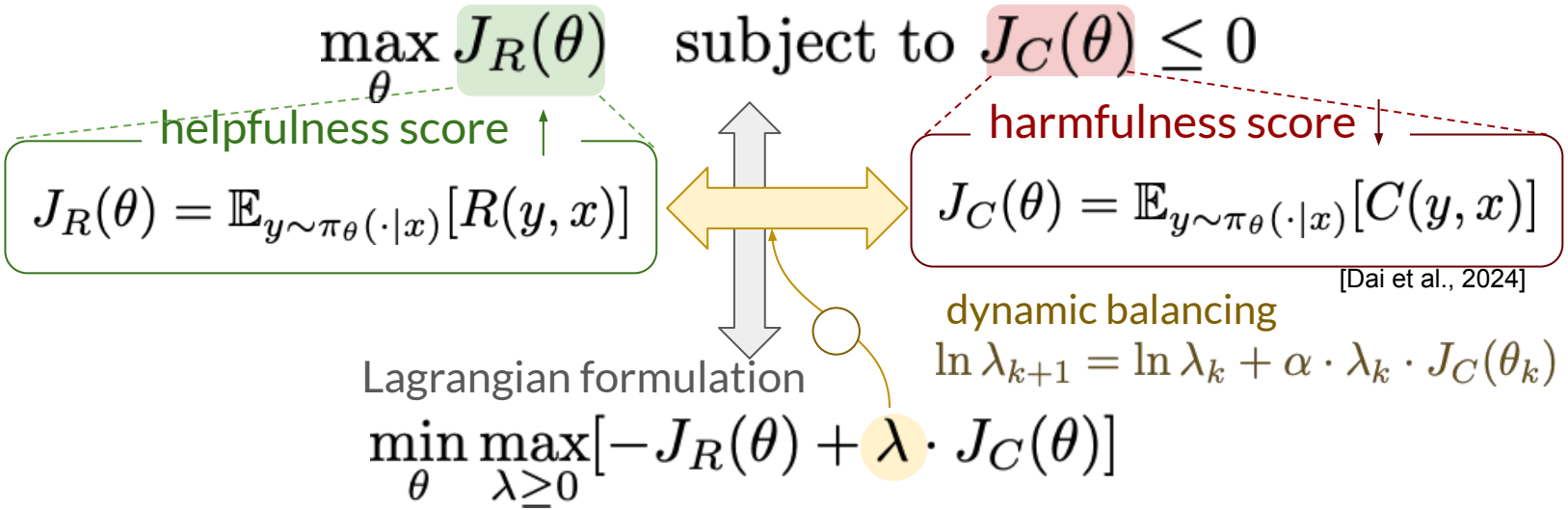


Key challenges

- balancing safety and capabilities
- scalability of safety techniques
- preventing safety degradation
- efficient runtime safety

Case Study: Safe RL with Human Feedback

Key idea: **decoupling** of helpfulness and harmlessness objectives:



♠dynamic balance between objectives ♠harm mitigation while maintaining performance

Approaches for LLM Falsification and Evaluation

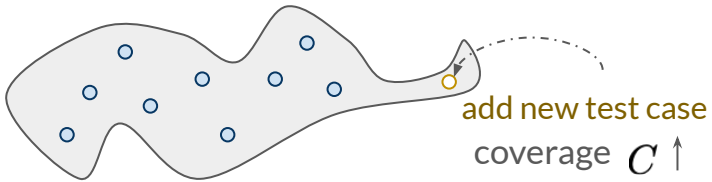
falsification and evaluation are crucial to ensure LLM safety and trustworthiness

coverage-guided techniques

♠ **input space coverage:** define distance metric $d(x, x')$ between inputs x and x'

♠ **behavioral coverage:** model LLM as Markov chain, measure state transition probabilities [Huang et al., 2021]

♠ use **coverage metrics** C to guide generation



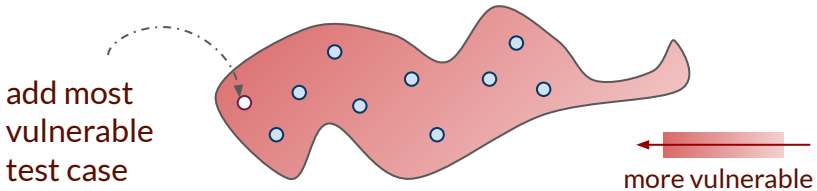
$$d(x, x') \leq \kappa \Rightarrow P(|f(x) - f(x')| \geq \epsilon) \leq \delta$$

provides probabilistic robustness guarantees for the LLM's behavior (assuming LLM response f is Lipschitz continuous)

adversarial testing/red teaming

♠ **prompt injection attacks:** craft prompts to hijack intended behavior [Perez and Ribeiro, 2022] [Zou et al., 2023]

♠ **semantic perturbations:** generate text inputs with small, semantically-preserving changes [Jia et al., 2017] [Wang and Bansal, 2018]



$$\max_x L(f(x)) + \lambda \Delta C(x)$$

hyperparam.

find the test input hat maximizes the vulnerability of the LLM $L(f(x))$ and increases the coverage of the test suite $\Delta C(x)$

future directions:

- develop coverage metrics and test case generation algorithms *tailored to LLMs*
- adapt probabilistic evaluation techniques to handle *text-based, non-deterministic* LLMs

Adversarial Testing for Blackbox LLMs

GCG (Greedy Coordinate Gradient)

$$\max_x L(f(x))$$

- **prompt injection** to induce harmful content generation
- greedy + gradient-based optimization
- universal transferability

[Zou et al., 2023]

connection

Learning-to-Optimize

see survey [Chen et al., 2022]

Problem 1

⋮

Problem N

meta-learning

general solver $\pi_\theta($ prob. $)$

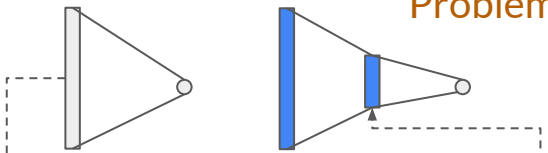
new prob.

$\hat{z}$

meta-learn a strategy that quickly adapt to new instances [Andrychowicz et al., 2016] [Metz et al., 2022]

*Meta-learned manifold random search for **blackbox** functions* [Sel et al., 2023]

$$f_\psi(\cdot) = h(r(\cdot;\psi_r);\psi_h)$$



search space is reduced to low-dimensional manifold

Problem 1

Problem 2

Problem N

identifies the low-dimensional manifold shared by all problems

meta-solver $\pi_\theta($ prob. $)$

test-time adaptation + optimization

new problem

Test-Time Alignment Methods for LLMs

improve LLM outputs without modifying model weights, adapting behavior during inference

1 prompting

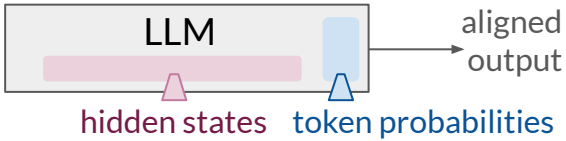
guide LLM through input text/prefix crafting

♠ blend instructions with in-context examples [Askell et al., 2021] [Lin et al., 2023] [Zhang et al., 2023]

♠ complex reasoning, e.g., constitutional AI [Bai et al., 2022b], "skin in the game" [Sel et al., 2024]

2 steering/intervention

intervene in generation process to enforce constraints

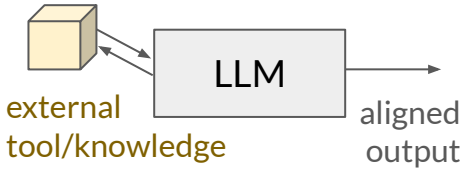


♠ **representation engineering:** add steering vectors to the representation space [Zou et al., 2023], attribute classifier [Dathathri et al., 2020], attention head outputs [Li et al., 2023], dynamic hidden state manipulation [Kong et al., 2024]

♠ **guided decoding**: reward-guided token selection [Khanov et al., 2024], prefix-based reward prediction [Mudgal et al., 2023]

3 external tool/ knowledge augmentation

incorporate external tool/ info during generation

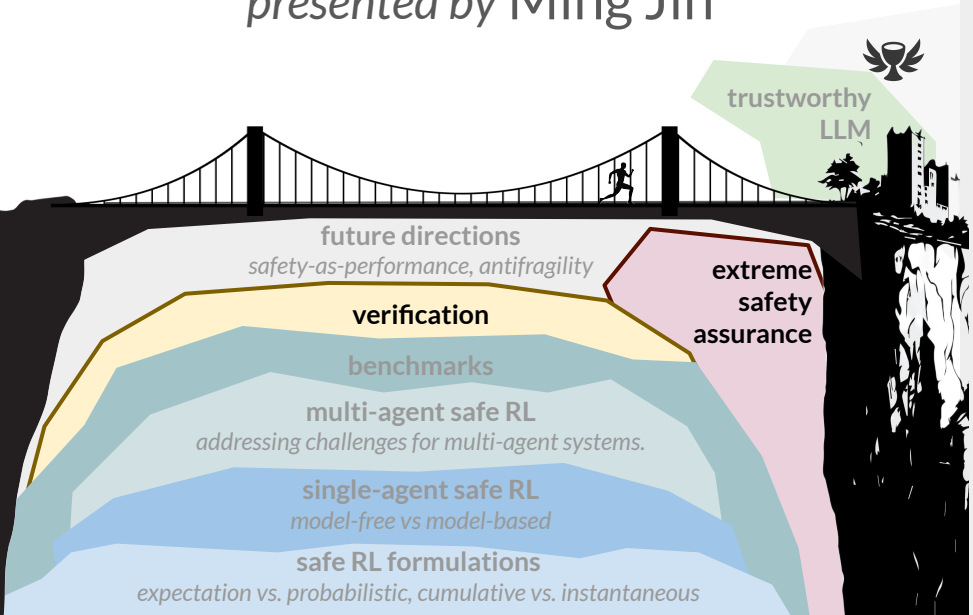


♠ **retrieval-augmented generation**: web browser [Guu et al., 2020] [Nakano et al., 2021], search engine [Menick et al., 2022], document database [Izacard et al., 2023]

♠ tools/APIs: ToolFormer [Schick et al., 2024], HuggingGPT [Shen et al., 2024], ToolLLM [Qin et al., 2024]

extreme safety assurance

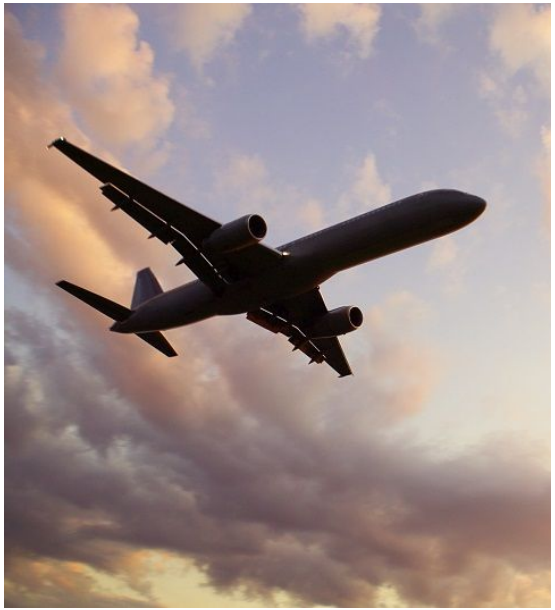
presented by Ming Jin



Key Objectives

- understand the extreme safety requirements in critical systems and the unique challenges they pose for RL
- explore Run Time Assurance (RTA) architectures for ensuring safety in complex, unverified systems
- examine various approaches to deep RL verification, e.g., programmatic, symbolic, and formal methods
- analyze the holistic view of DRL verification and future directions for achieving ultra-low failure rates in safety-critical domains

Extreme Safety Requirements in Critical Systems



10^-9 failure rate
in commercial aviation
"extremely improbable" by FAA
one failure per billion flight hours

[FAA Advisory Circular 25.1309-1A]

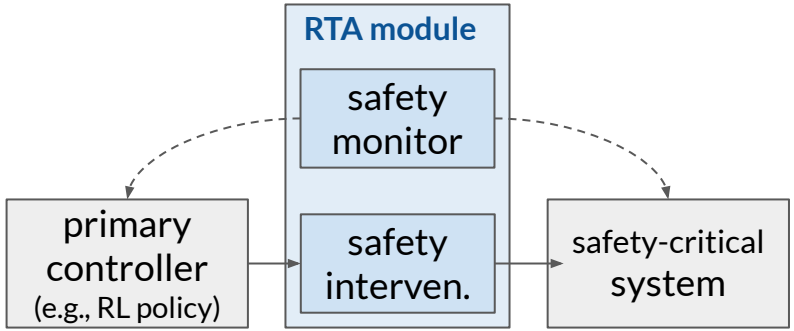
challenges specific to RL:

- exploration-exploitation dilemma
- distributional shift
- need for robust generalization

Run Time Assurance (RTA) Approaches to Safety

Key ideas:

- online **monitoring** and **intervention**
- safety filter between controller and system
- enables use of complex, unverified controllers (e.g., RL policies)



Key properties:

innocuity
no unintended negative impacts

viability
maintains future safe options

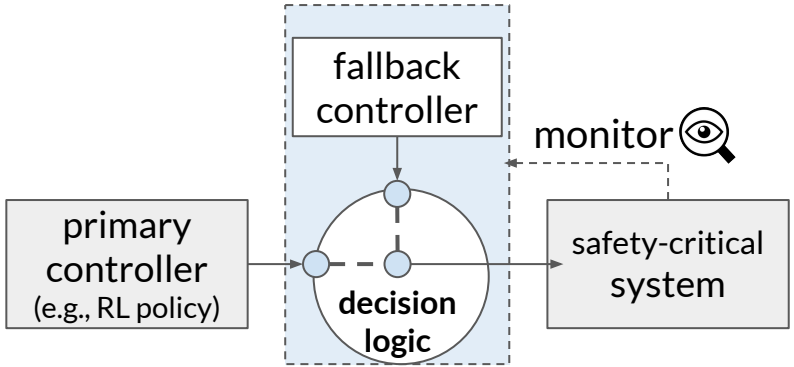
nuisance free
min. unnecessary interventions

Key industry applications: F-16 Automatic Ground Collision Avoidance System (Auto GCAS) [Swihart, et al., 2011], Mobileye's Responsibility-Sensitive Safety (RSS) model for autonomous vehicles [Shalev-Shwartz, et al., 2017]

RTA Architectures for Safe Learning Systems

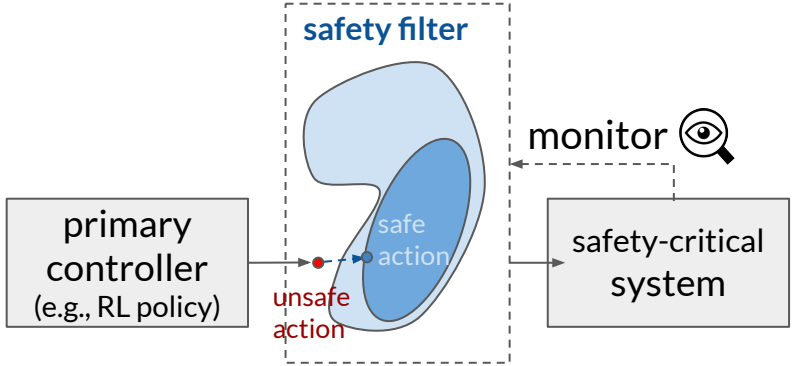
Simplex Architecture:

- **Switching** between learner and verified policy
- **Separation** of performance and safety
- Safe exploration through fallback mechanism



Safety Filtering:

- **Continuous** modification of control inputs
- **Integration** of safety into learning process
- Minimally invasive interventions



Applications: auto GCAS [Swihart et al., 2011], collision avoidance for high speed quadrotor navigation [Lopez and How, 2017], swarm robotics (Robotarium) [Pickem et al., 2017], lane keeping [Enache et al., 2010] [Xu et al., 2017] [Hu et al., 2019], exoskeletons [Gurriet et al., 2019]

see also [Table 1, Hobbs et al., 2023] for a survey on RTA applications

Overview of DRL Verification

- a critical component in a larger safety assurance strategy
- complementary to extensive testing, redundancy, and fail-safe mechanisms

unique challenges

sequential decision-making scalability

handling stochasticity

balancing performance & safety

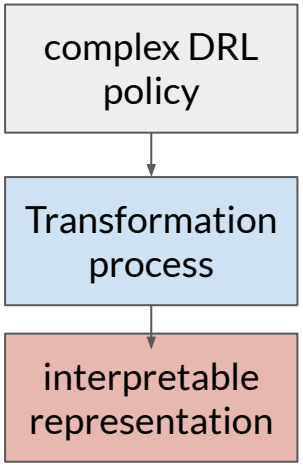
verify emergent behaviors

key considerations

- specifying properties vs. proving functional correctness
- managing large state spaces through abstraction
- probabilistic verification for stochastic systems
- compositional methods for complex systems

DRL Verification (1/4): Programmatic/Symbolic Transformation

Key idea: transform complex DRL policies into interpretable and verifiable forms



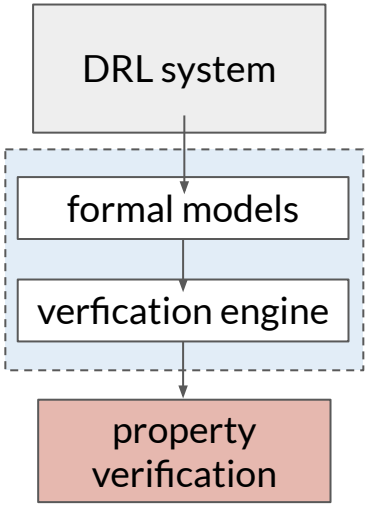
policy extraction (e.g., VIPER [Bastani et al., 2018])	**domain-specific language** (DSL) (e.g., PIRL [Verma et al., 2018])	**neurosymbolic approach** (e.g., REVEL [Anderson et al., 2020])
imitation learning black-box to white-box conversion	program synthesis domain knowledge integration	hybrid learning and reasoning combining neural and symbolic components
decision tree	DSL program	neurosymbolic model

Key aspects of transformation: • abstraction • knowledge transfer • iterative refinement • trade-off management • domain knowledge integration

Benefits: ♠ improved interpretability ♠ easier formal verification ♠ enhanced maintainability ♠ bridging AI and traditional software engineering

DRL Verification (2/4): Formal Methods-Based Verification

Key idea: apply rigorous techniques to prove properties of DRL systems



constraint-based:
a) SMT solving (e.g., Reluplex [Katz et al., 2017])
b) abstract Interpretation [Jin et al., 2021]

represent NNs as linear constraints

SMT solvers (e.g., Z3), abstract domain operations (e.g., interval arithmetics)

robustness to perturbations, reachability analysis

state exploration:
model checking (whiRL 2.0 [Amir et al., 2021], MOSAIC [Bacci and Parker, 2020])

state transition systems (e.g., MDPs)

symbolic model checkers (e.g., PRISM), bounded model checking w/ k-induction

temporal logic properties (e.g., LTL, PCTL), probabilistic safety properties

pros: ♠ provides strong mathematical guarantees ♠ can handle complex properties ♠ scalable to larger state spaces (with abstractions)

cons: • can be computationally intensive • may require expertise in formal methods • abstraction may lead to over-approximation

DRL Verification (3/4): Other Methods

1 control-theoretic approaches

analyis based on control theory and dynamical systems

e.g., Verisig (transforms DNN controllers into hybrid systems) [Ivanov et al., 2019], CBFs, IQC robust control [Gu et al., 2022]

PROS: leverages established mathematical frameworks

CONS: may be conservative for highly nonlinear systems

2 specification engineering

methods to express and refine properties to be verified in DRL systems

e.g., behavioral properties - encode general, rational behaviors expected from DRL systems [Corsi et al., 2021]

PROS: improves precision and relevance of verification

CONS: requires domain expertise to define appropriate properties

3 multi-agent verification

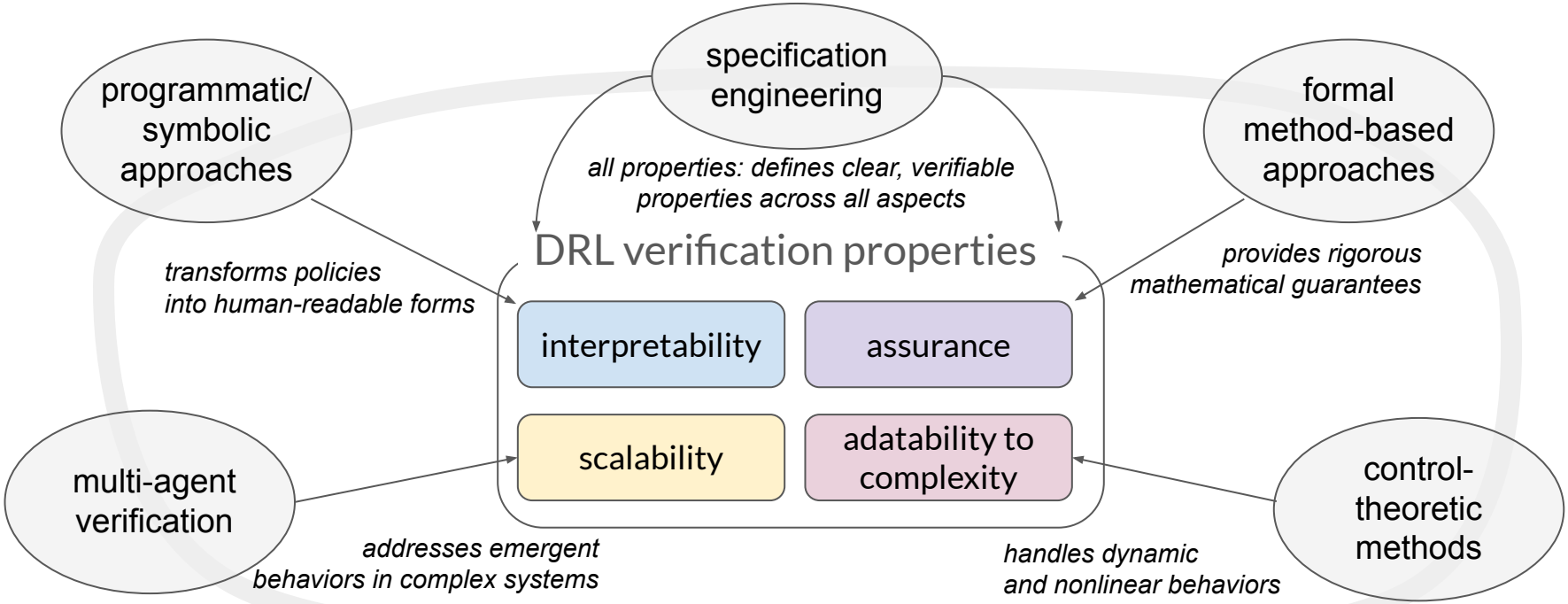
extend verification to systems with multiple interacting agents

e.g., compositional abstractions of individual agents [El Mqirmi et al., 2021]

PROS: verify emergent behaviors, coordination properties

CONS: scalability/computation challenge

DRL Verification (4/4): Holistic View and Future Directions

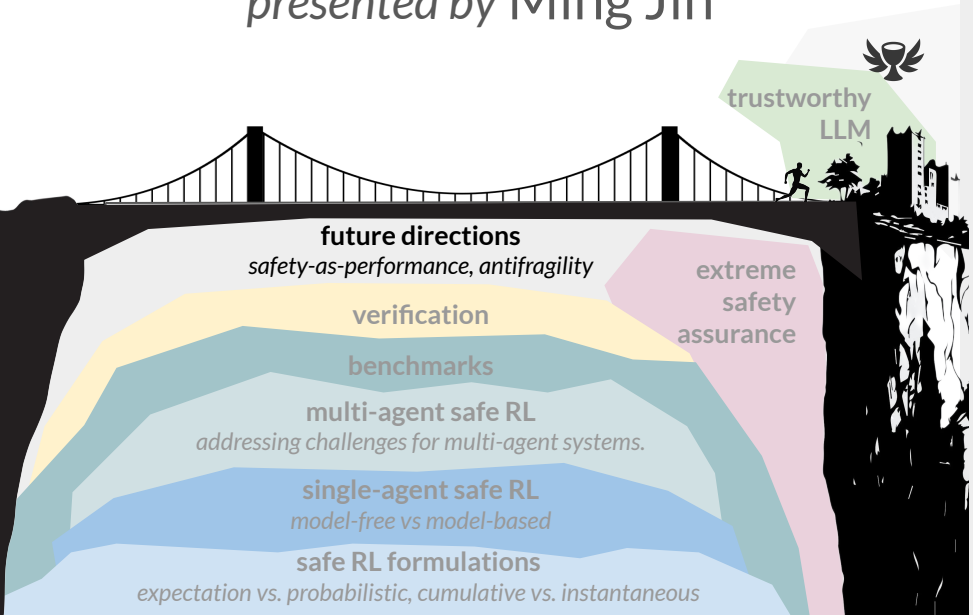


Future Directions: towards ultra-low failure rates (e.g., <10^-9)

- integrating AI/ML techniques into verification processes
- hybrid approaches for more comprehensive verification
- improving scalability for real-time verification and multi-agent and large-scale systems

challenges & future directions

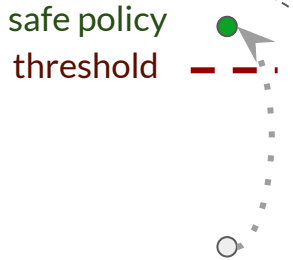
presented by Ming Jin



Future Research Directions

- From constraint satisfaction to performance optimization in open-world environments
- Advancing adaptive safety in dynamic scenarios
- Antifragility in AI: Developing systems that improve through exposure to challenges and uncertainties

Reimagining Safety in RL

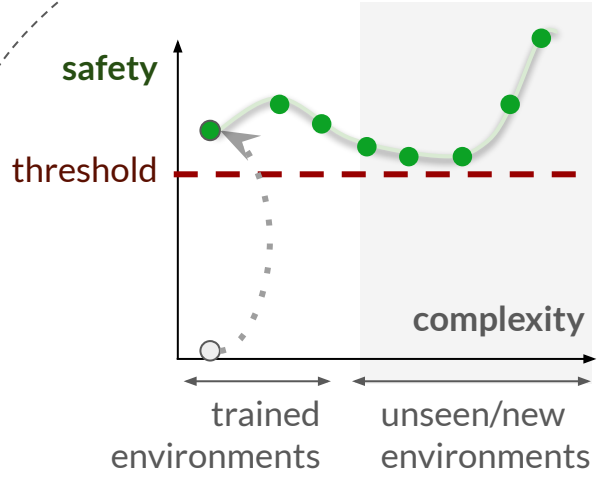


safety as constraint
in trained
environments

nonstationarity
adapting to constantly
changing environments

unknown unknowns
handling unforeseen
events

beyond robustness
thriving in adverse conditions
and safe in black swan events

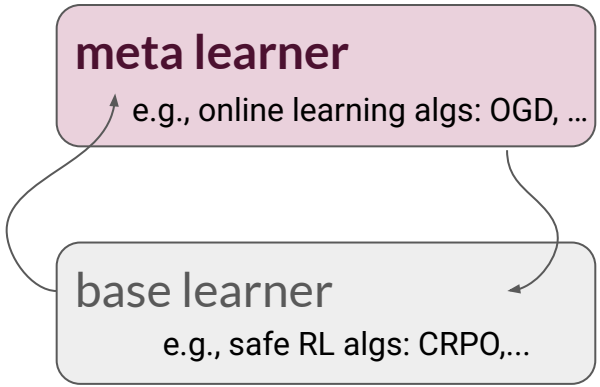


safety as performance
across complex, unknown,
extreme environments

closed world safety → **open world safety**

Case Study: Meta-Safe RL

Goal: to handle non-stationary environments and unknown scenarios by treating safety as a performance objective.



oversees/adapts base learner strategies

performs standard safe RL tasks

adaptive safety bound

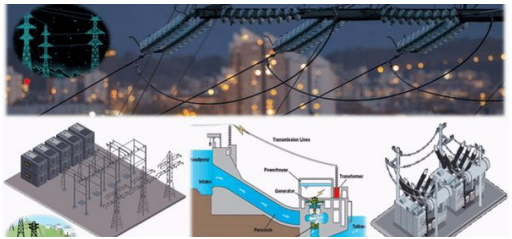
$$\mathcal{O}\left(\frac{\text{Env. variations}}{\sqrt{\#\text{ steps}}\#\text{ past CMDPs}}\right)$$

the more challenges (# CMDPs) encountered, the faster to be safe in future situations (reduced suboptimality gap)

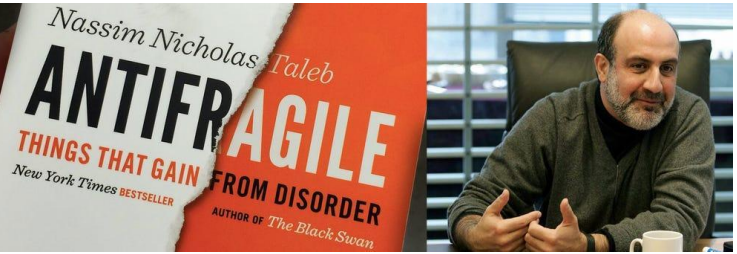
[Khattar et al., 2023]

♠**adaptive safety** mechanisms ♠**hierarchical structure** ♠first theoretical guarantee

critical load restoration to quickly restore the grid in blackouts [ul Abdeen et al., 2024], future applications in robotics, healthcare, autonomous driving, financial markets.

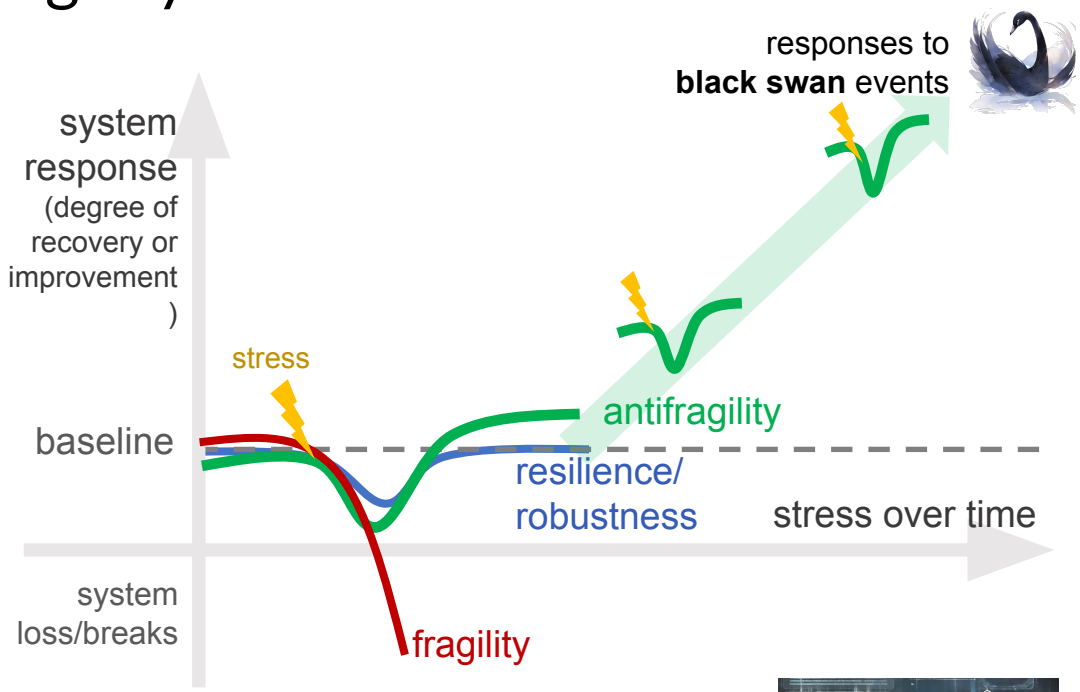


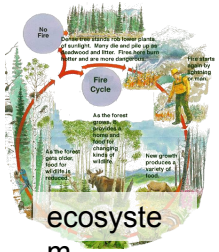
The Revelation of Antifragility



> some things **benefit from shocks**; they **thrive** and **grow** when exposed to **volatility, randomness, disorder,** and **stressors** and *love* adventure, risk, and uncertainty.

antifragility is *beyond resilience or robustness*. the resilient resists shocks and stays the same; the **antifragile gets better**.





immune
system

Towards Antifragile AI

Preparing for Black Swans 🦢:
The Antifragility Imperative for Machine Learning

Ming Jin*

*these research directions highlight the path towards creating **unusually safe** and **antifragile** AI systems capable of handling Black Swan events.*

nonstationary online learning

adapting to changing environments

improving with volatility

partially observable MDPs

capturing hidden factors and uncertainties

inferring and adapting to unobserved dynamics

safe exploration & control

learning under safety constraints

expanding boundaries safely and purposefully

continual/ lifelong learning

building on past knowledge, balancing stability and plasticity

adapting over time

meta-learning

learning to learn, adapting learning strategies

rapid adaptation

foundation models

on-the-fly adaptation via in-context learning

leveraging memory and retrieval for test-time adaptation

quality-diversity & multi-obj. learning

maintaining diverse solutions, handling conflicts

dynamic repertoire of strategies

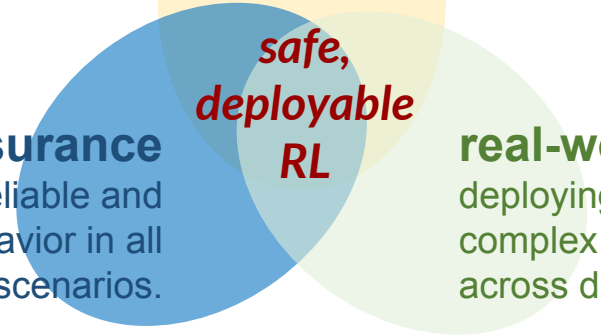
adversarial ML

revealing vulnerabilities, ensuring robustness to extremes

strengthening through “vaccination”

The Future of Safe RL

Reinforcement Learning
enabling AI to learn optimal decision-making through interaction & feedback



safety assurance
guaranteeing reliable and responsible AI behavior in all scenarios.

real-world impact
deploying AI to solve complex challenges across diverse industries.

call to action: envision, innovate, collaborate, and shape the future of safe, intelligent systems.

Questions? Contact Ming Jin (jinming@vt.edu)

Safe RL Seminar

Acknowledgements:

NSF amazon Commonwealth Cyber Initiative C3.ai 4-VA U.S. DEPARTMENT OF ENERGY Deloitte.

Image: DALL-E 3