

# Creating Artificial Intelligence from the World

Haeone Lee



How can we create “artificial” intelligence (AI) from the world?

# Definition of Intelligence

# Can we define intelligence?

“A very general mental capability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience” (a)

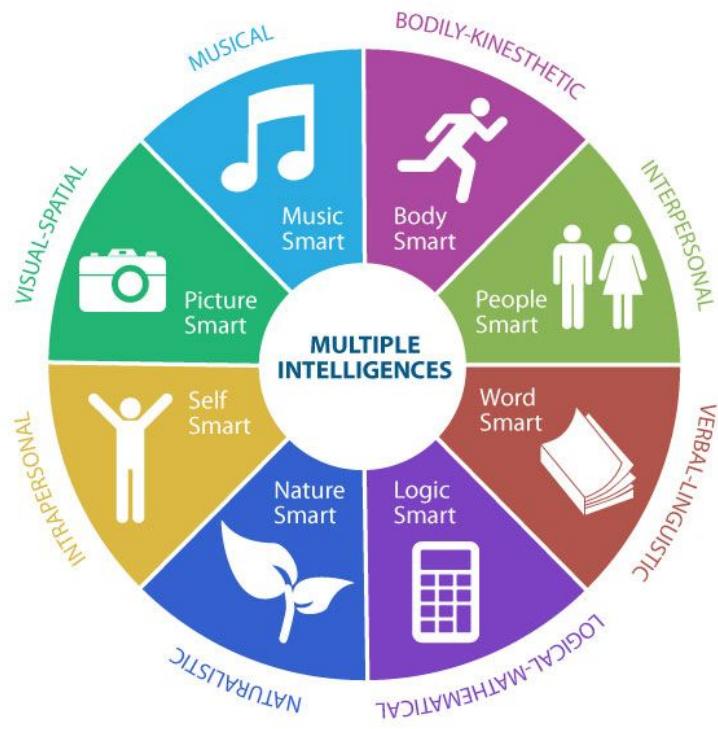
Mainstream Science on Intelligence

“a set of skills of problem solving” (b)

“To act purposefully, to think rationally, and to deal effectively with his environment” (c)



**There are many different definitions & kinds**



# Can we define intelligence?

“A very general mental capability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience” (a)

Mainstream Science on Intelligence

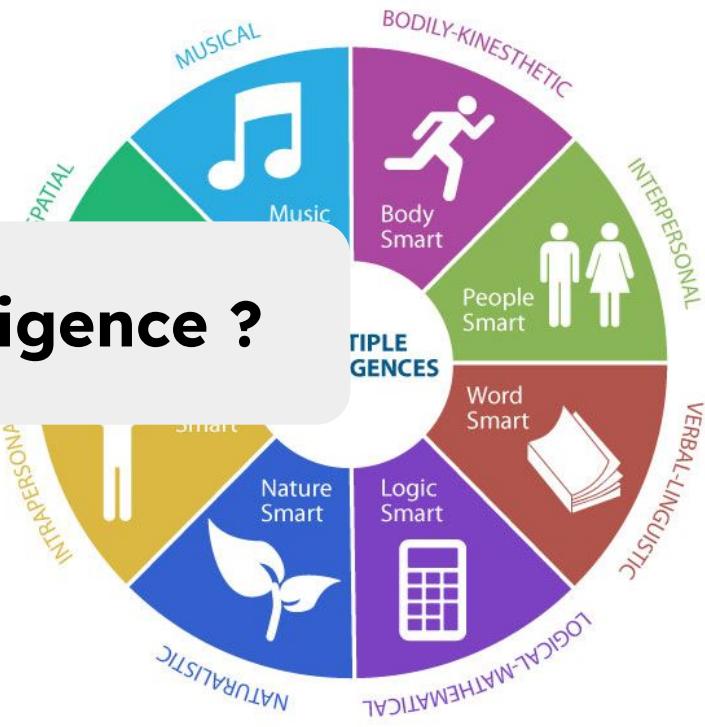
“a set of skills of problem solving

“To act purposefully, to think and to feel with his environment” (c)



## Why we need intelligence ?

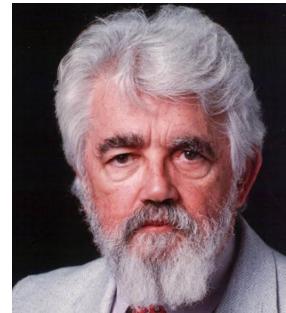
There are many different definitions & kinds



# Intelligence to achieve goals

“Intelligence is the computational part of the ability to achieve goals in the world” (a)

John McCarthy



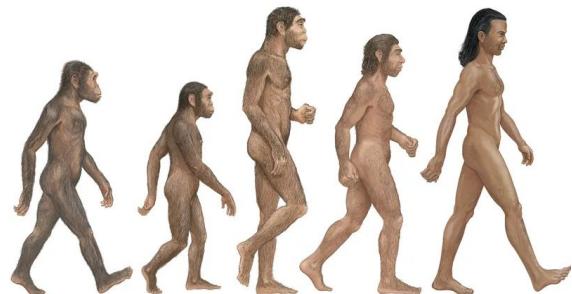
**The goals are**



To win the game

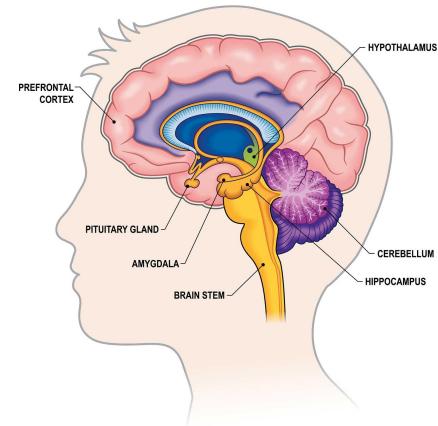
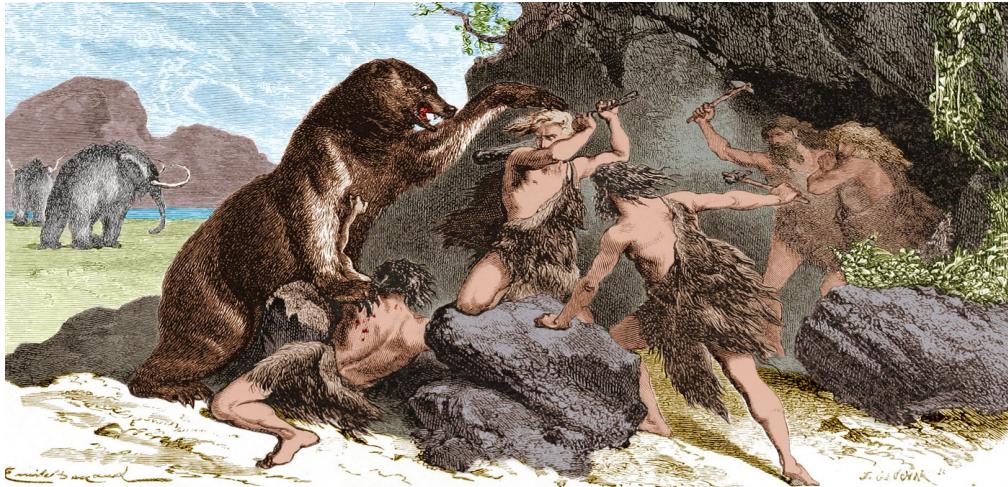


To collect more food



To survive & breed

# This leads to many complex capabilities!



## Human develops

- **Perception** to identify objects (enemy vs colleague, edible vs inedible food, different tools)
- **Language** and **social** intelligence to manipulate other agents in a favored way
- **Physical** and **tool** intelligence to manipulate the world
- **Artistic** intelligence to attract the opponent sex

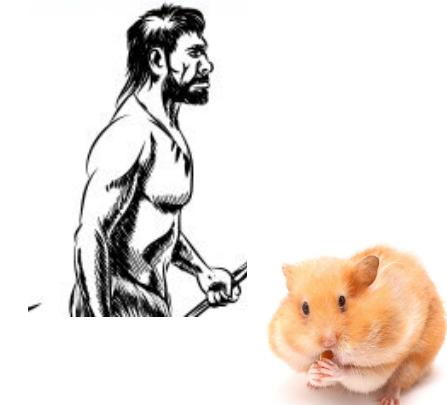
# Key properties of intelligence



VS



More intelligent



- The more complex the world, the more complex intelligence is needed
- Physical (motor control) intelligence is very important in our world and should not be neglected
- More computational capacity leads to better intelligence

# Intelligence as decision making

- To achieve goals, good decision makings are necessary
- = Intelligence is to make good decision

Examples: which word to say next, place to go, muscle to move

At the  
lowest level  
decisions

“We have a brain for one reason and one reason only – and that’s to produce adaptable and complex movements. Movement is the only way we have affecting the world around us” (a)

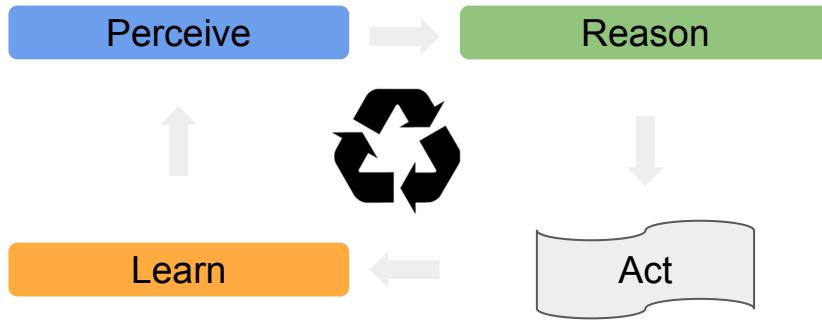
Daniel Wolpert



Note: decisions are often hierarchical, high level involving low-level decision  
E.g., place to go → way to get there → muscle to move,

# Intelligence as decision making

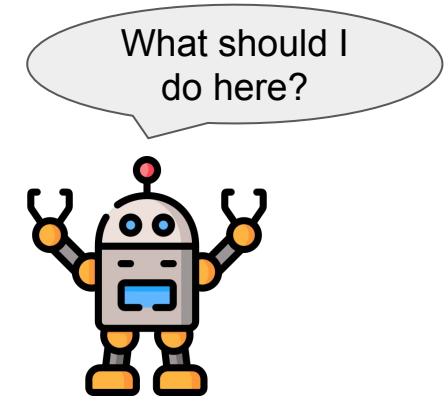
Decision making can be formulated as **Perception**, **Reasoning**, and **Learning**



- **Perception:** process the information of the world in a favored way
- **Reason:** outputs the decision from the perception
- **Learn:** observe the outcome and correct the policy

**My definition** - perception, reasoning, and learning are the intelligence!

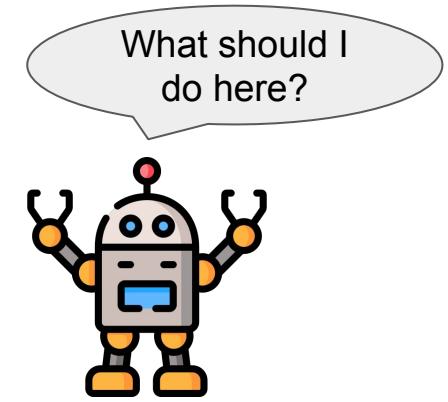
# World as **decision making process**



Worlds: real world, games, simulation, etc.

- Worlds can be modelled as **decision making process** given a specific goal
- Given the current state, AI outputs action and transitions into the next state

# World as POMDP



Since all the components in the world cannot be observed at the same time, this becomes POMDP!

$$M = (S, A, T, \Omega, O, \mathcal{G}, p_0) \quad \text{where}$$

$\mathcal{G}$  all the possible set of goals

$p_0$  initial state distribution

$\Omega$  emission probability

# Intelligence in different worlds

- Beyond real worlds, we can define many different worlds and create intelligence out of it

## Text world

defined by a set of tokens (words)

What we want:



Hi! I'm traveling to Hawaii next week. Can you recommend an activity to me that I will enjoy?



Of course! To start off, would you only like activities for a particular island?



Yes, I will mainly just be staying in Maui.



Great! There are plenty of activities to do there. Would you consider yourself very active?

## POMDP

**State:** sentence

**Action:** next token

**Next State:** sentence + token

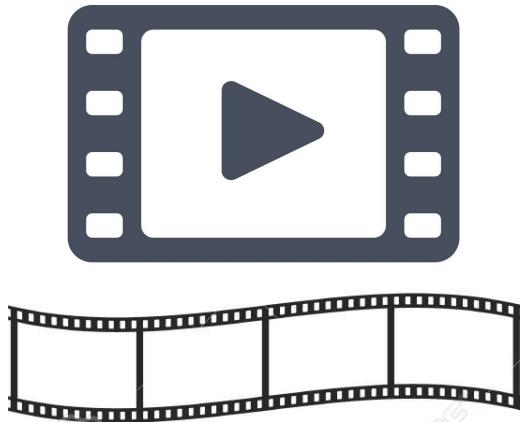
## Goals to pursue

- Generate plausible text
- Answer the question
- Manipulate others (e.g. to buy the product)

# Intelligence in different worlds

- Beyond real worlds, we can define many different worlds and create intelligence out of it

**Video world**  
defined by a set of videos



## POMDP

**State:** frame

**Action:** next frame

**Next State:** next frame

## Goals to pursue

- E.g., to move to the specific frame
- To develop good perception

# Intelligence in different worlds

- Beyond real worlds, we can define many different worlds and create intelligence out of it

## Dataset world defined by a set of data



## POMDP

**State:** data

**Action:** label or next data

**Next State:** next data

## Goals to pursue

- To output correct label
- To develop good perception

**Trials of creating intelligence**

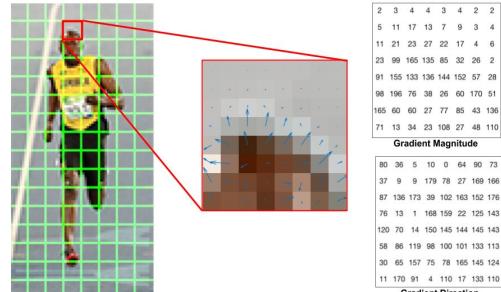
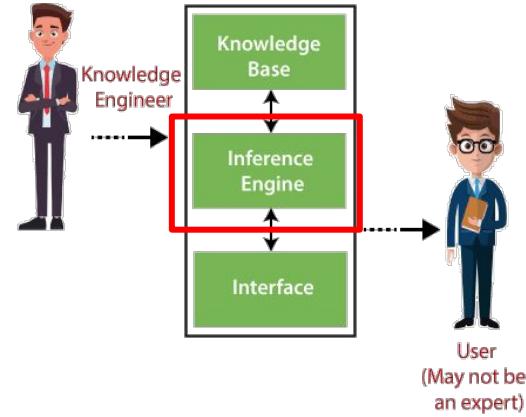
# Engineering approach

Decision making procedure is **manually** designed by human. E.g.,

- 1) knowledge-based system (reasoning)
- 2) feature engineering (perception)

## Shortcomings

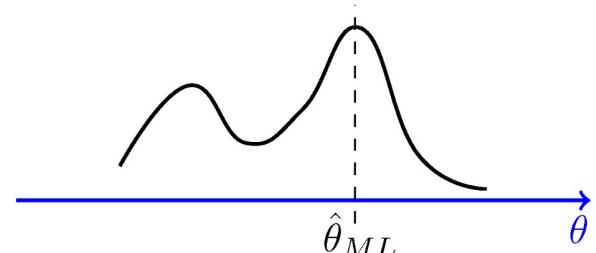
- Large manual effort required
- Injects human inductive bias which could be wrong or not generalizable
- Hard to scale at complicated problems



# Data-modelling approach

Can we automatically extract decision-making rule from the dataset?

$$\tau = (o_0, a_0, o_1, a_1, \dots) \sim M(\pi_g)$$



$\pi_g(a | s)$  **Policy** that pursues a specific goal g (e.g., human correctly labeling the data)

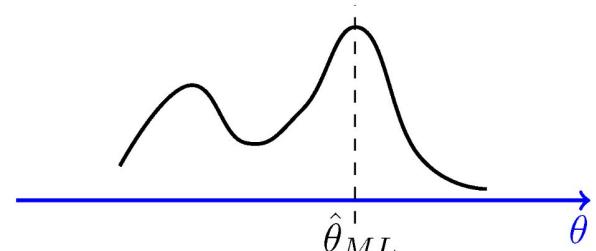
- Modelling the resulting data distribution can develop the intelligence for achieving goal g

Reasoning	Perception	Learning
$P(A   S)$	$P(S), P(\tau) \dots$	$P(\tau)$

# Data-modelling approach

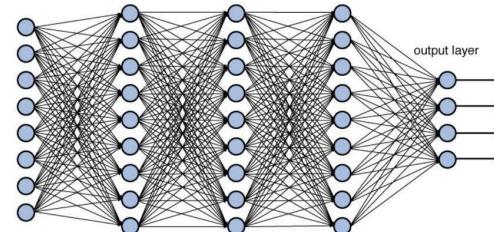
Can we automatically extract decision-making rule from the dataset?

$$\tau = (o_0, a_0, o_1, a_1, \dots) \sim M(\pi_g)$$

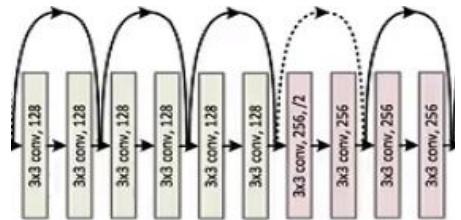


$\pi_g(a | s)$  **Policy** that pursues a specific goal g (e.g., human correctly labeling the data)

- Modelling the resulting data distribution can develop the intelligence for achieving goal g
- DNN can be used for a model due to its
  - a. excellent generalization capability
  - b. good capacity



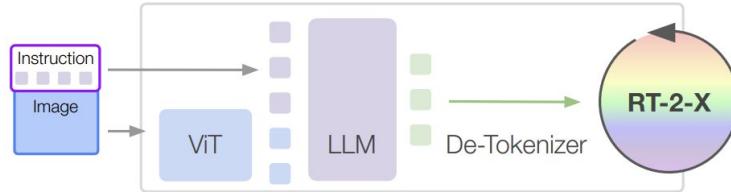
# Data-modelling approach: Reasoning



ResNet[1]

**World:** dataset world (ImageNet)

**Policy:** human to label the data correctly



Robotic transformer[2, 3, 4]

**World:** real world with robot

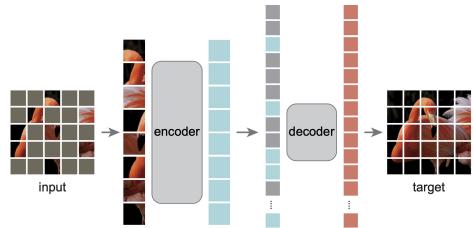
**Policy:** human manipulating robot to achieve the given task

- Outputs the decision given the state
- Perception is trained end-to-end by neural network
- Includes traditional supervised-learning & behavior cloning

$$p(A | S)$$

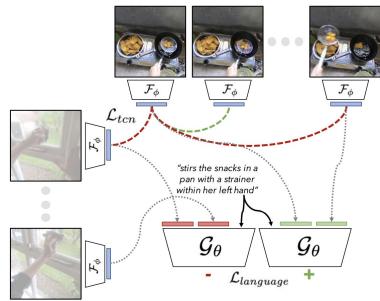
Reasoning

# Data-modelling approach: Perception



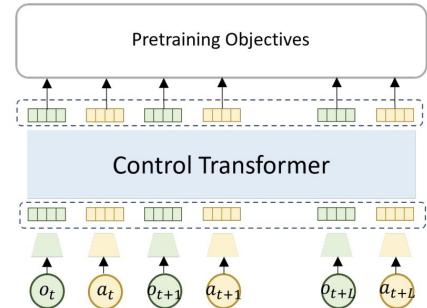
MAE[1]

Reconstruct the image from the masked image



R3M[2]

TCL in video world with text alignment



PASTA[3]

Next token prediction in low-level action trajectory

- Ground the reasoning by providing good expression of the state
- Includes unsupervised learning & representation learning

$$P(S), P(S' | S), P(\tau)$$

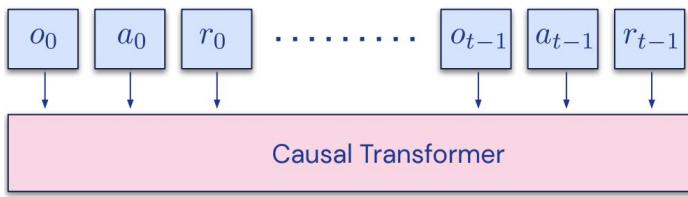
Perception

# General: Trajectory modeling



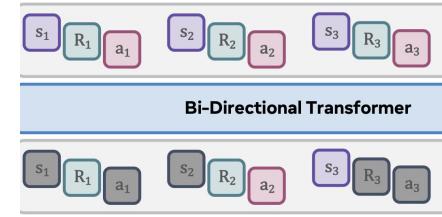
GPT 1~3[1]

NTP from the text  
generated by human policy



Algorithm distillation[2]

Models improving trajectory by  
RL policy with different goals



MTM[3]

Predict tokens that are  
masked in trajectory

- Can function all the perception, reasoning, and learning
- Learning can be implemented by modelling improving trajectories

$$p(\tau)$$

Trajectory modelling

# Is data modelling enough?



showed success

- Data collection is **burdensome** → internet scraping
- Generalizability is limited to the **dataset support** → use large amount of data
- Still, the performance to the goal is **upper-bounded** by the dataset
- Cannot **adapt** to the changes in the world
  - e.g., covariate shift, label shift

Reasoning	Perception	Learning
$P(A   S)$	$P(S), P(\tau)...$	$P(\tau)$

# What's the way to go?



All of what we mean by **goals and purposes** can be well thought of as **maximization** of the expected value of the cumulative sum of a **received scalar signal**

[Richard Sutton](#)



Intelligence, and all of its associated abilities, can be understood as subserving the maximization of reward

[David Silver et al.](#)

# What's the way to go?

- Can we convert our goal to the reward?
- Can we directly optimize such reward and observe the resulting intelligence?



## REINFORCEMENT LEARNING

All of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal

[Richard Sutton](#)

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \dots} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t) \right]$$

---

Reward is enough

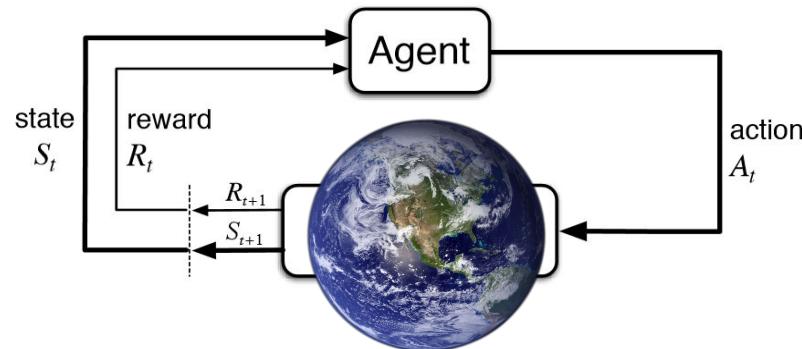
David Silver\*, Satinder Singh, Doina Precup, Richard S. Sutton

---

# Reinforcement learning

## Good

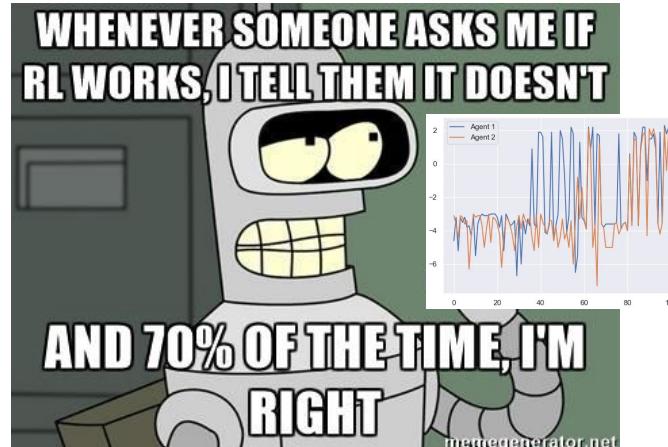
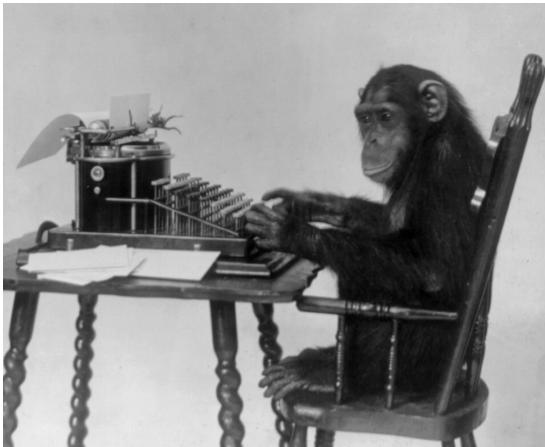
- Might lead to **superhuman** intelligence
- Learning can be done **autonomously** given only the reward
- Can **adapt** to the changing world by interaction and is trained life-long
- **Analogous** to human learning by reinforcement in psychology



# Challenges

But..

- How can the agent explore the world well to find good decisions – because there are so many options (“monkey typing hamlet” problem)
- RL has instability and low sample efficiency issue



# Using prior knowledge

**How can the agent explore the world well to find good options**

- NOTE: still, good options are very few!
- Prior knowledge can guide agent to the good options



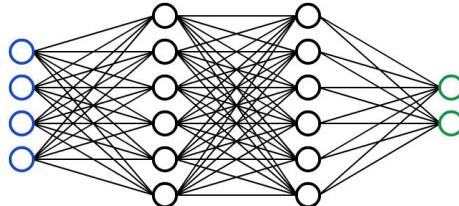
Baby deer instinctively know that it has to run away



Baby moves to door when instructed 'open the door' not going to other direction

# Using prior knowledge

## Examples



### Static dataset

- Action trajectory
- Video
- Image
- Text

### Pretrained model

- Policy
- Value
- Trajectory model
- Perception

### Large common sense model

- LLM
- VLM

There are so many kinds. What's the best way to utilize those ?

# Applications of prior knowledge in RL

**Policy prior** - what is generally useful to do here

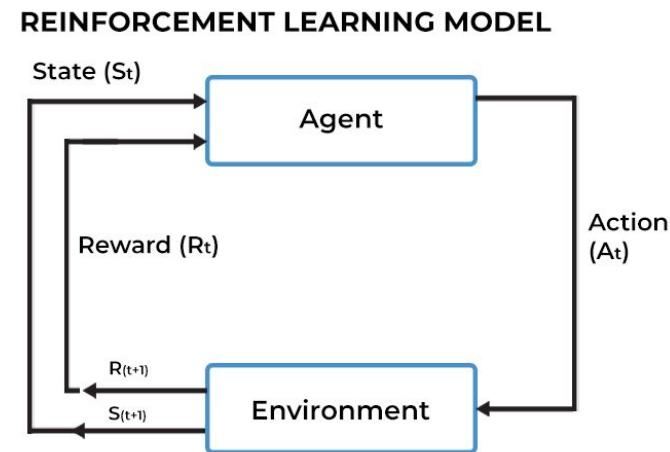
**Value/Reward prior** - how generally good my behavior is

**Dynamics** - how the world works

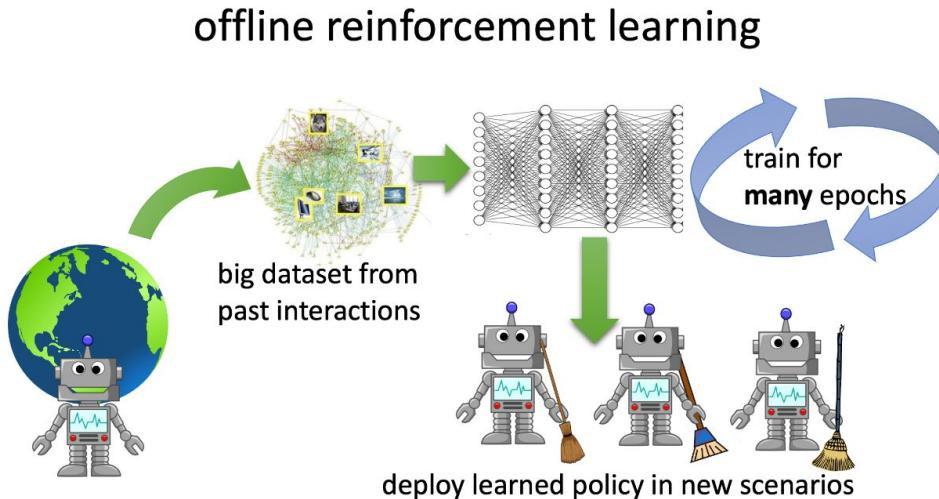
**Perception** - how to process the information

Reward and dynamics are normally given in RL

- Still, prior knowledge can be required to design the reward function
- Dynamics might be used for model-based RL



# Using static prior: offline RL



= Data-driven optimization

- Algorithms are designed to be conservative outside of the support

$$\arg \min_Q - \mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a} \sim \hat{\pi}_\beta(\mathbf{a}|\mathbf{s})} [Q(\mathbf{s}, \mathbf{a})]$$

$$+ \frac{1}{2} \mathbb{E}_{\mathbf{s}, \mathbf{a}, \mathbf{s}' \sim \mathcal{D}} \left[ (Q(\mathbf{s}, \mathbf{a}) - \hat{\mathcal{B}}^\pi \hat{Q}^k(\mathbf{s}, \mathbf{a}))^2 \right]$$

- Since data is limited, learned model ( $Q$ , policy) might be inaccurate outside of dataset support

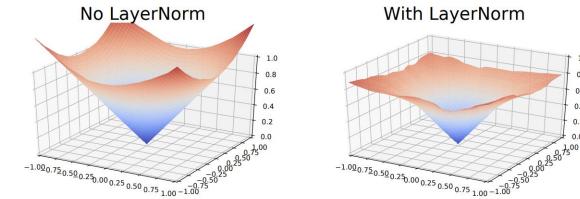
# RL hits different now

**RL has instability and low sample efficiency issue**

→ Algorithm has got a lot better nowadays. This may not be an issue anymore

use Q-functions = recycle old data via replay buffer

Algorithms: SAC (but also TD3, REDQ, DroQ, all good...)



- Techniques such as layernorm, clipped Q learning, and ensemble greatly improves stability[1, 2]
- Sample efficiency is improved by employing priors and using off-policy RL with high replay ratio

# Example of intelligences by RL

# Text world: maximize human preference

**Policy goal:** to generate the text that human prefers

**Policy prior:** (supervised) LLM

**Value prior:** reward → since it is initialized by reward function

**Reward prior:** LLM → since reward uses LLM for initialization

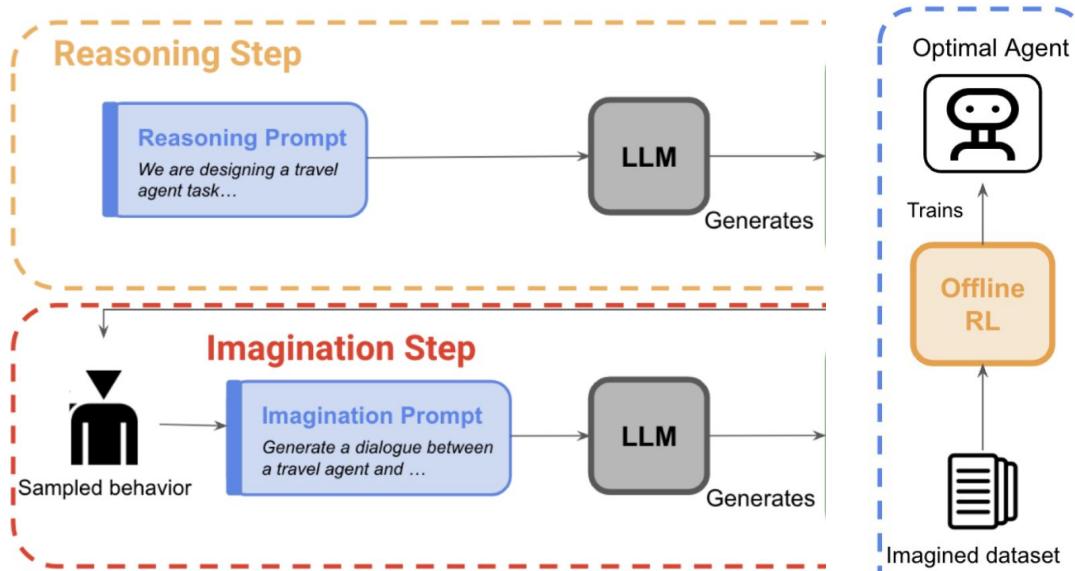


$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r(x, y_w) - r(x, y_l))] \quad \text{Reward objective (preference)}$$

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [r_\phi(x, y)] - \beta \mathbb{D}_{\text{KL}} [\pi_\theta(y|x) || \pi_{\text{ref}}(y|x)] \quad \text{Policy objective}$$

- Train a reward model by human preference on policy samples (or use existing critique models)
- Optimize the reward by RL with conservatism to prevent over-optimization

# Text world: achieve the goal in conversation



**Policy Goal:** to achieve intended conversation outcome  
**Policy prior:** LLM  
**Value prior:** -  
**Reward prior:** -  
**Dynamics prior:** LLM

- Generate the successful and failed dialogue given the specific conversation outcome
- Use success/failure as a reward and optimize by offline RL(IQL) from the data

# Video world: learn good perception

- Train the value function with the goal of reaching the specific frame
- Use the representation of value function in the downstream tasks (value learning, policy learning, etc.)

$V(\phi(o); \phi(g))$  probability of reaching g by optimal goal-reaching policy from o

$V_\theta(s, s_+, z) = \phi(s)^\top T(z) \psi(s_+)$  probability of reaching  $s_+$  while acting according to z



**Policy goal:** to achieve the given goal frame

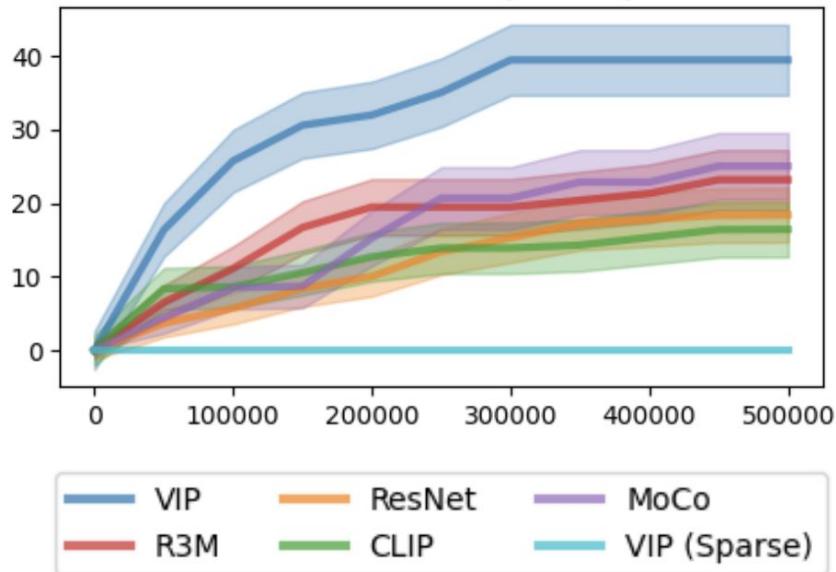
**Policy prior:** static dataset

**Value prior:** -

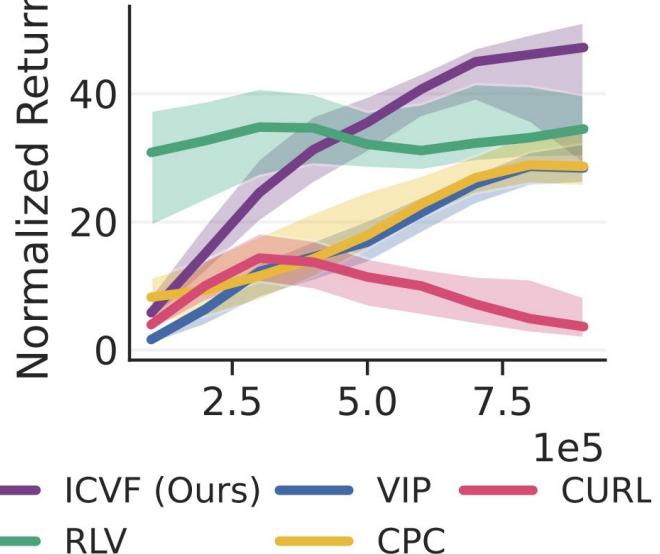
**Reward prior:** -

# Video world: learn good perception

Online RL (NPG)



Downstream Performance



- Representation of learned by RL (value function) shows superb performance compared to the ones by dataset modelling

# Image world: learn good perception

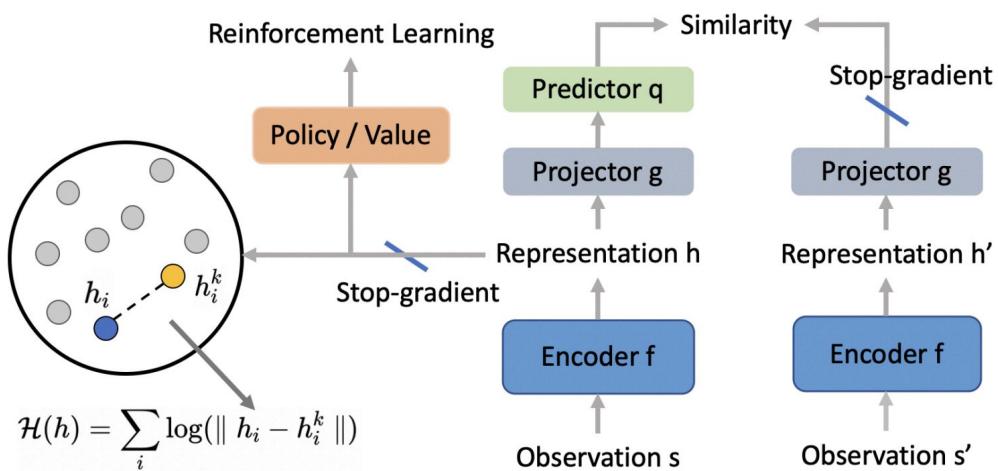
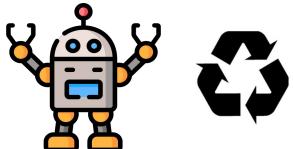
- Generate the diverse images by exploring the latent of generative models
- Apply contrastive learning for the images
- Such diversity of data leads to robust representation learning

**Policy goal: to maximize state entropy of the state**

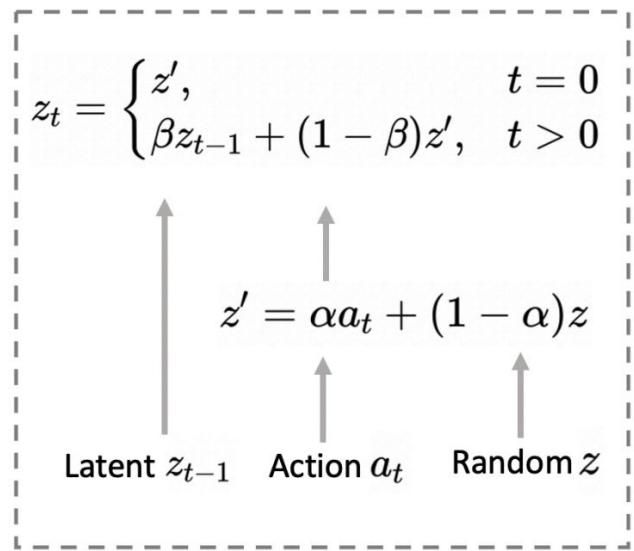
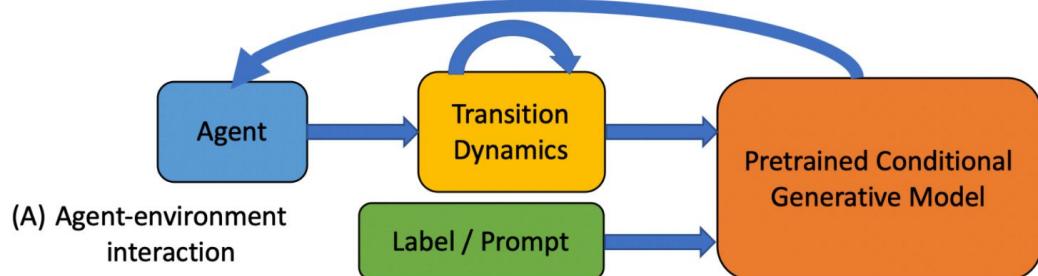
Policy prior: -

Value prior: -

Reward prior: -



# Image world: learn good perception



## Building image world

- First, learn the smooth latent for the dataset by generative modelling (StyleGAN)
- Define the action as delta direction on the latent space
- Transition dynamics is the interpolation between action and current latent

# Real world: achieve the instructed task

- Apply offline RL in robotic trajectories
  - a. autonomously collected data including failures
  - b. expert demos
- RL on mixed quality dataset is better than BC on expert data only

**Policy goal:** to complete the instructed task

**Policy prior:** static dataset

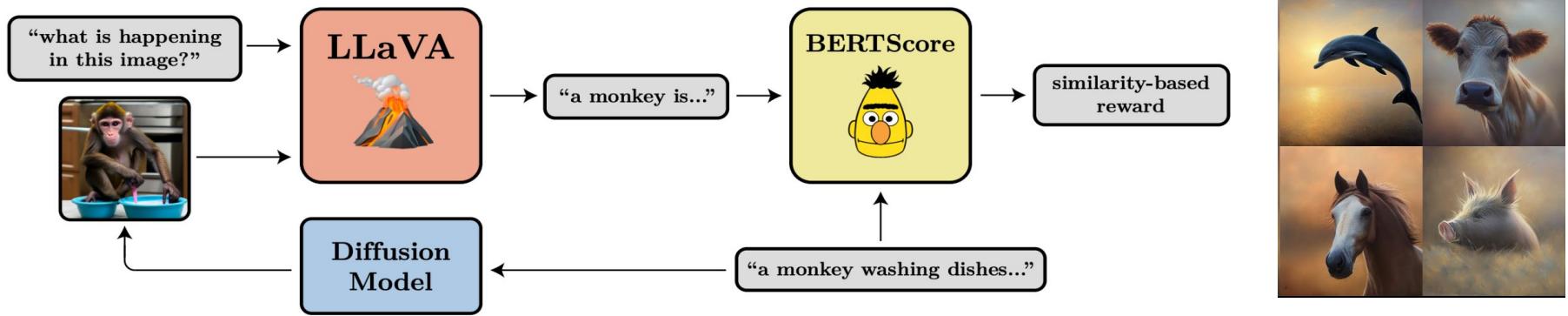
**Value prior:** -

**Reward prior:** -

Task category	Q-T	DT	IQL	RT-1
drawer pick and place	64%	49%	11 %	17%
open and close drawer	33%	11%	11%	0%
move object near target	71%	40%	60%	58%
Average success rate	<b>56%</b>	33%	27%	25%



# Pixel world: generate instructed image



**Policy goal:** to output image satisfying the conditions

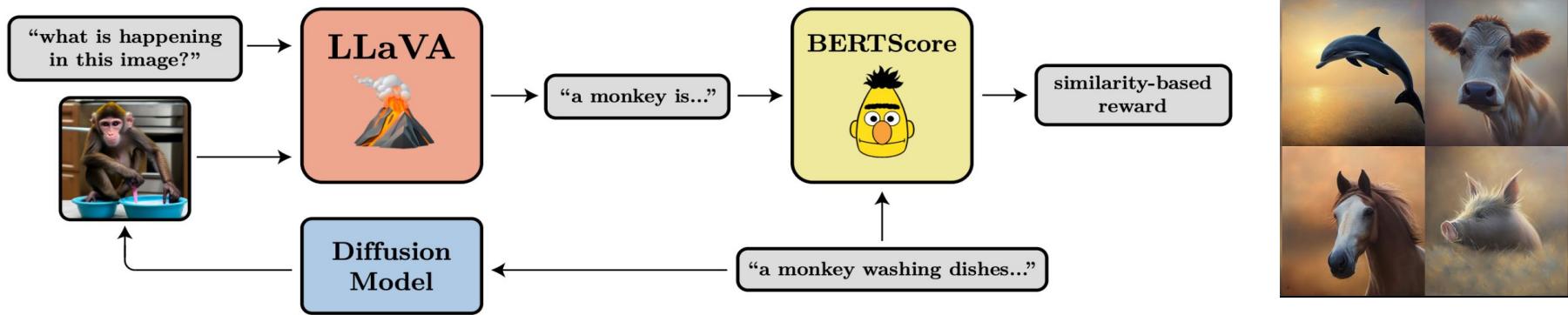
**Policy prior:** diffusion model

**Value prior:** -

**Reward prior:** vlm

- Models the diffusion step as decision making in MDP
- Use the off-the-shelf reward model
- Objective: compressibility, aesthetic quality, prompt alignment

# Pixel world: generate instructed image



$$R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{x}_0, \mathbf{c}) & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi(\mathbf{a}_t \mid \mathbf{s}_t) \triangleq p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$$

$$\rho_0(\mathbf{s}_0) \triangleq (p(\mathbf{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I}))$$

- Policy is modelled by one-step denoising
- Initial state is normal dist w/ sampled condition
- Trained by REINFORCE or PPO



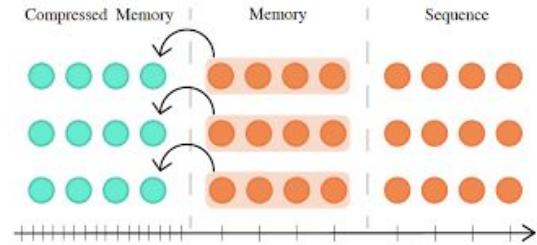
# Things to do

# Things left to be done

Many important capabilities are lacking in current methods

## Memory

- Memory is necessary in many worlds due to **partial observability**
- Still, most methods use memoryless policies



## Long-horizon

- Discount factor might limit the horizon of policy by multiplication
- Bellman **bootstrapping** might be unstable at very long-horizon



# Things left to be done

## Autonomous training & deployment

- Many works assume **episodic resets** during training and deployment



## Inductive bias

- Lots of works employ human inductive bias such as **reward engineering**
- This could limit the generalizability and capacity of the models



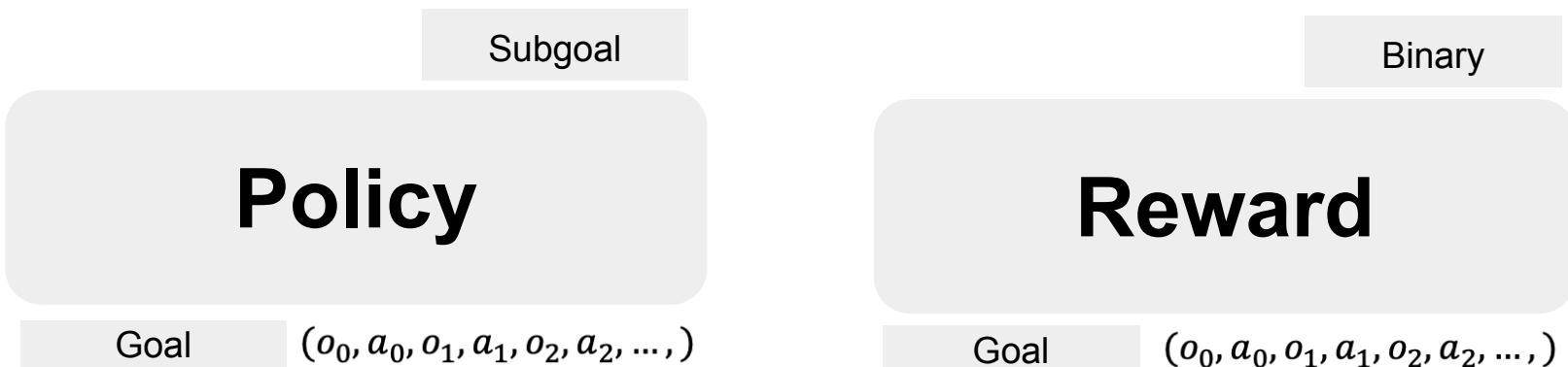
## Use of all the prior knowledge

- Most works use limited amount of prior knowledge

# Proposed structure

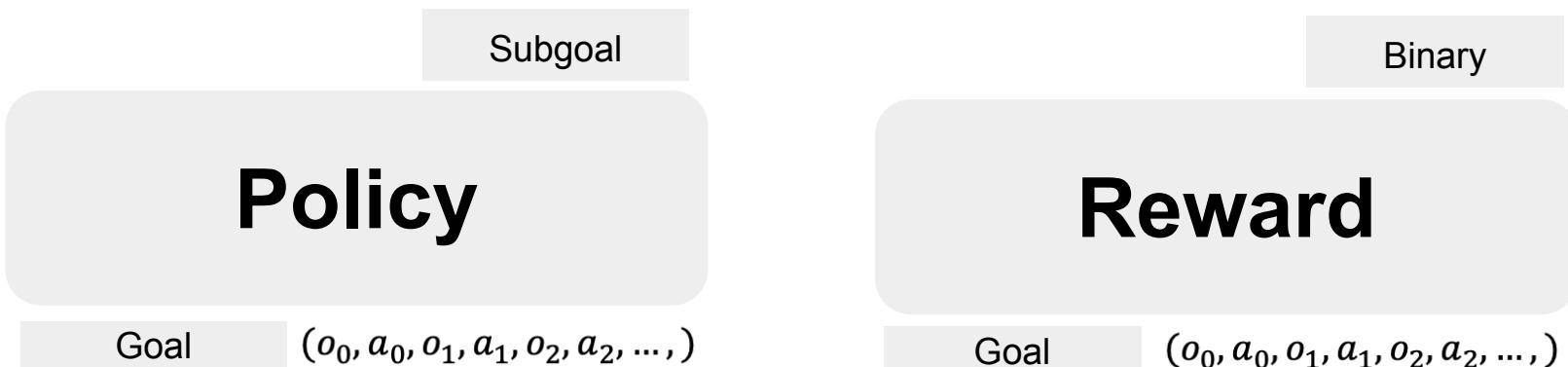
## Abstraction helps

- We can abstract low-level actions into higher-level subgoals
- This helps to plan in long-horizon by turning multiple actions into the single subgoal
- Similarly, abstracted states can help efficient memorizing in long-term



# Proposed structure

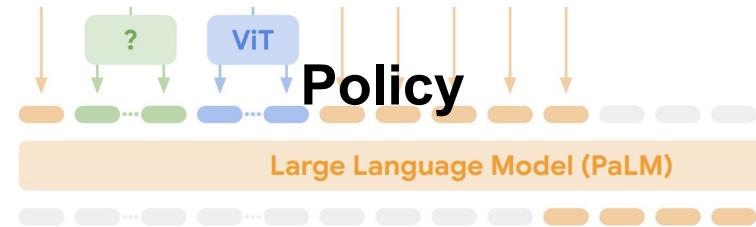
- Policy: given goal and **historical** input, outputs subgoal
- Reward: given goal, outputs the probability that the goal is **achieved**
- Policy can form hierarchy by recursively querying subgoals
- Using recursive memory architectures (e.g., RMT[1], MAMBA[2]) practically enables infinite context



# Using pretrained models for policy & reward

## Policy

- Use common sense model.
- Use models trained by dataset modelling (e.g., goal-conditioned BC)
- Action: subgoals in different abstraction levels



**Policy:** Given (state), what is the subgoal for AI do to achieve (goal)?

**Reward:** Given (goal), does (trajectory) achieve it?

Common sense model prompting

## Reward

- Common sense model
- Alignment model (e.g., CLIP) between goal and trajectory

# Training policy from static data

Use offline goal-conditioned RL to the trajectory dataset (k-steps long subgoal)

$$Q^*(s_t, g, goal) \leftarrow R(s_t, g, goal) + \gamma \max_{\hat{g}(g)} Q^*(s_{t+k}, \hat{g}, goal)$$

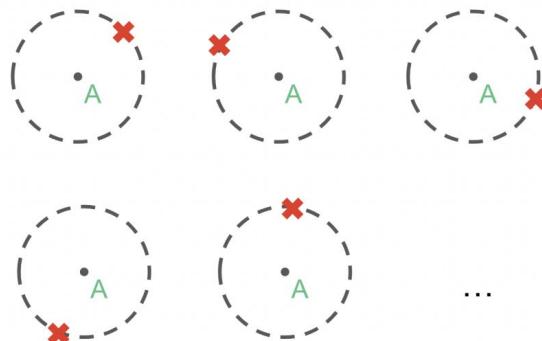
$$\mathbb{E}_{\tilde{a}_t \sim p_{\tilde{\mathcal{A}}} } [P^\pi(s_T \neq g | s_t, \tilde{a}_t)] = 1 \quad \text{Apply conservatism for unseen actions and goals[1]}$$

- Policy and reward function can be initialized by pre-trained models
- Goals and subgoals are randomly sampled from trajectory
- In addition, trajectory can be labelled by the reward function

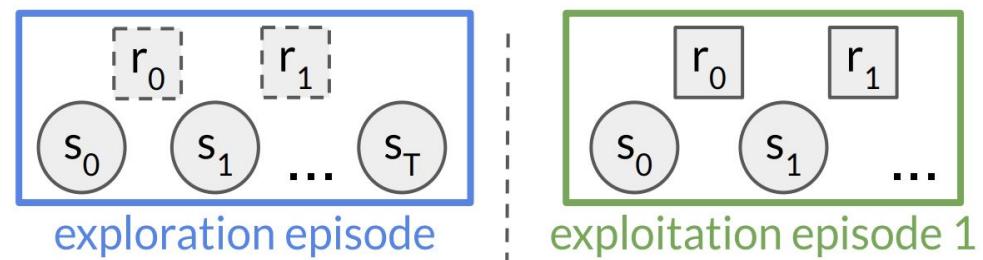


# Meta reinforcement learning of policy

- Policy can **meta-learned** by training from diverse environments
- This is equivalent to learning **bayes adaptive optimal policy** from the distribution of environments
- It will provide the optimal exploration and exploitation strategy



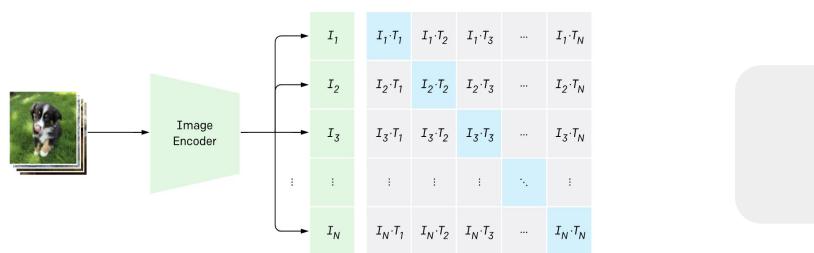
E.g., open the door by pulling (fails) → memorize such information → try pushing



Training in many worlds (different way of opening doors)

# Training reward

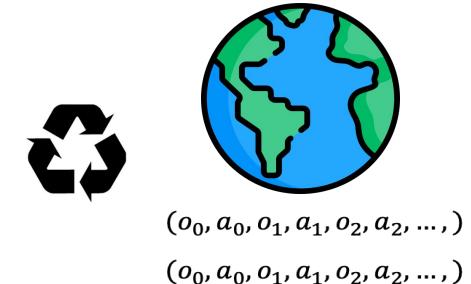
- **Fine-tune** pretrained models by contrastive learning (CLIP) or instruction-tuning method using (trajectory, goal) paired datasets
- Reward function can also be trained from the **online** data, which is challenging since it is unlabeled
- One could use label bootstrapping technique, similar to SSL[3]



Reward

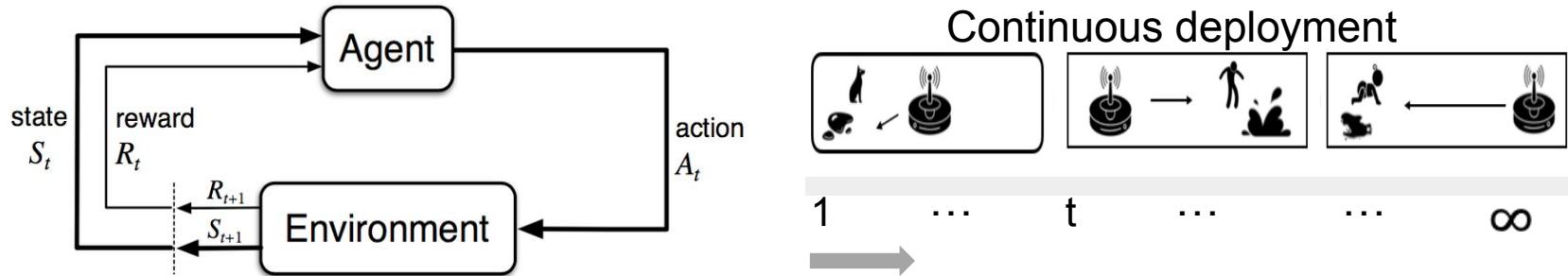
Online bootstrapping

Similarity between (goal) and (trajectory)



...

# Online reinforcement learning

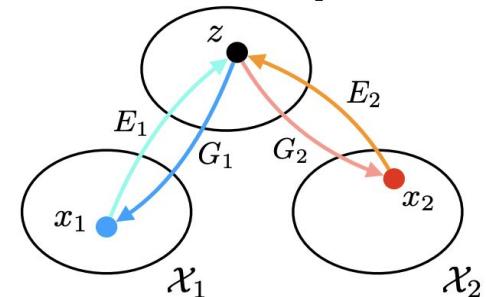


- We run online RL to the target environment with policy and reward initialized by priors
- Reward can be given by the reward function for the goals pursued by the policy + hindsight re-labelling
- Efficient exploration is enabled by priors and meta-learning by memory
- **Human instruction** can be included in the state and followed by agent

# Use all the prior knowledge

- How to leverage text data to help robotic learning?
- Information transfer between different data is needed

Assume two trajectories (A) and (B) and policy learned there as  $\langle A \rangle$ ,  $\langle B \rangle$



**Hierarchical transfer:** subgoals from  $\langle B \rangle$  can be followed by  $\langle A \rangle$  or vice-versa

**Policy transfer:**  $\langle B \rangle$  /  $\langle B \rangle$  can be a policy prior for  $\langle A \rangle$

**Perception transfer:** training in  $\langle B \rangle$  can provide a perception prior for  $\langle A \rangle$

**Semantic transfer:** training generalizes between data that has the same semantics

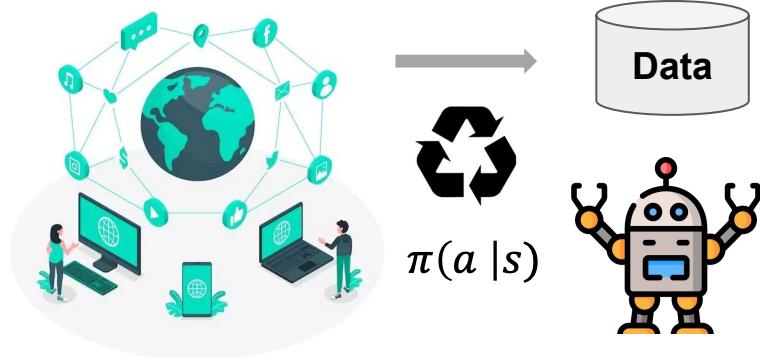
## Ex) information transfer to learn robotic policy

Training / Type	Perception	Policy	Hierarchical
RL	ICVF(v), VIP(v)	Cal-QL(a)[1]	-
SL / USL	MAE(i), R3M(v)	Demo-guided RL (a)[2]	SayCan(t)[3]

(i), (v), (a), and (t) denotes image, video, low-level action, and text dataset, the origin of information.

- Semantic transfer can be employed in addition (e.g., two trajectories are policy transferable after changing modality by semantic transfer)
- E.g., pixel goal generalizes to the text goal[4]

# Automatic data collection

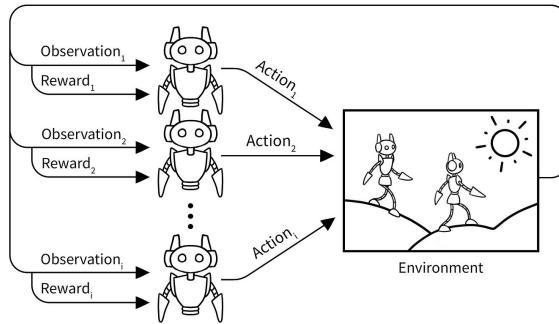


## Shortcomings of using **static** data

- **Limited** amount and diversity
- Not adaptive to the **changing** world
- Not **consider** to model's performance

- Quality data is important to train reward function
- We propose to use **dataset collection policy** in arbitrary world (e.g., Internet)
- This can be trained by RL to maximize **information gain**
- Heuristic objective such as model uncertainty about data could be employed (cf. similar RL exploration objective[1, 2])

# Social: multi-agent RL

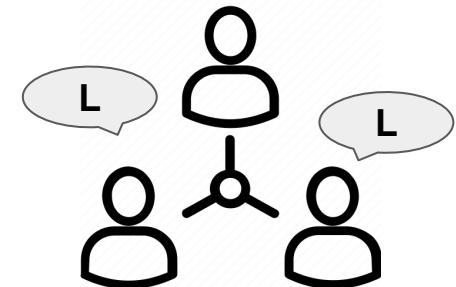


Multi-agent RL

emerges  
→



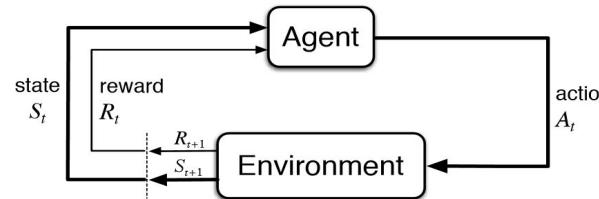
**imitation**: to mimic others' better performance



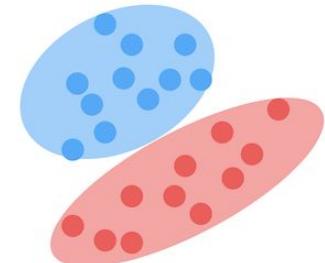
**language**: to cooperate to achieve the goal

- Multiple intelligences can interact to each other in deployment,
- This makes the world more complicated, promoting the emergence of more complex intelligence
- Language and imitation behavior is expected to emerge[1, 2]

# Summary



Offline goal-conditioned RL



Static data



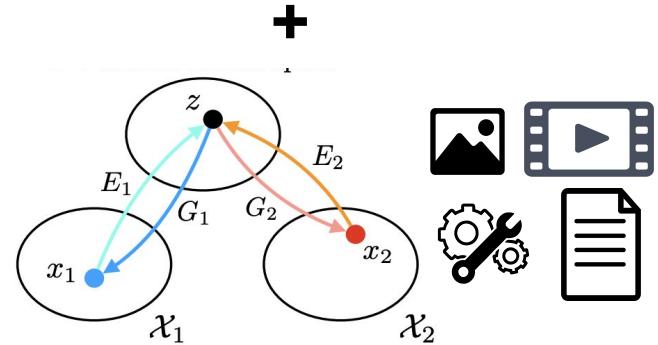
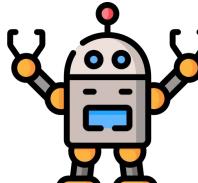
World



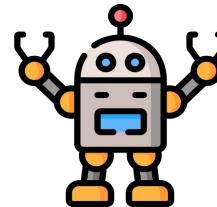
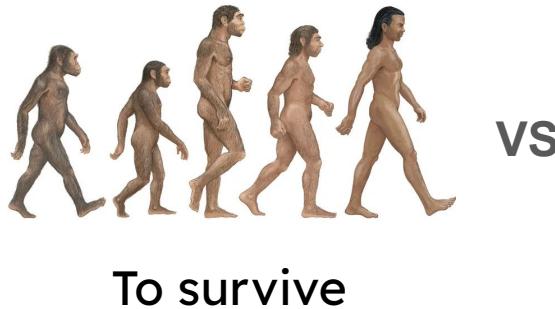
Dataset world

Online reinforcement learning

$(S, A, T, \Omega, O)$



# Toward ‘helpful’ intelligence



**What should be the highest level of goal for AI?**

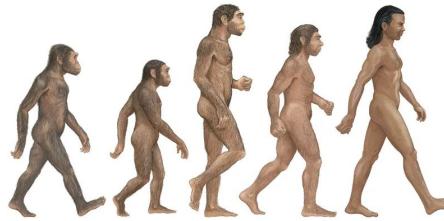
The intelligence should be **helpful** for human

Many capabilities of AI are **subset** of “being helpful”

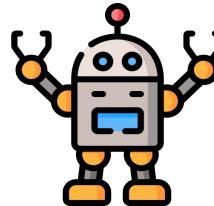
- AI plays the game well to entertain human
- answers the questions to assist human
- explores the world to gain knowledge for doing future tasks

Just like many **capabilities** of human is for survival

# My target: AGI in real world



vs



To survive

To be helpful for human



(AGI def.) “An intelligence that **maximizes** its **helpfulness** objective in a world which has an **infinite** number of **tasks** that are sufficiently **different** and identified as **productive** by humans.”

- We can use our **framework** to maximize “helpfulness”
- My **ultimate goal** is creating such AGI, especially in the real world.

# Projects

# My roadmap

Improving goal-conditioned RL

Offline goal-driven reinforcement learning in video, text

- Hierarchical policy
- Long-horizon
- Stitching
- Conservatism

- Reward function bootstrapping
- Efficient memory by RL
- Information transfer
- Adapting different memory architectures

Real world robotics

Multi-agent RL

AGI in real world

- Active training by RL (data collection)
- Autonomous practice & adaptation
- Multi-modality (sensor)
- Explainability

- Language & imitation emergence

- Apply all the methods

# Improving algorithm: learning hierarchical policy

Q function can be interpreted as probability of achieving (goal) by (action)

$$P(s_T = g | s_t, a_t) = Q^*(s_t, a_t, g)$$

(prob to achieve goal) = (prob to achieve subgoal) x (prob to achieve goal by subgoal)

$$Q^*(s_t, g^1, goal) \leftarrow \max_g Q^*(s_t, g, g^1) \{ R(s_t, g^1, goal) + \gamma \max_{\hat{g}(g^1)} Q^*(s_{t+k}, \hat{g}, goal) \}$$

- This end-to-end learns the Q function with hierarchy where the high-level subgoals are grounded to the low-level
- $g(g')$  denotes the set of subgoals that has the horizon length of  $g'$

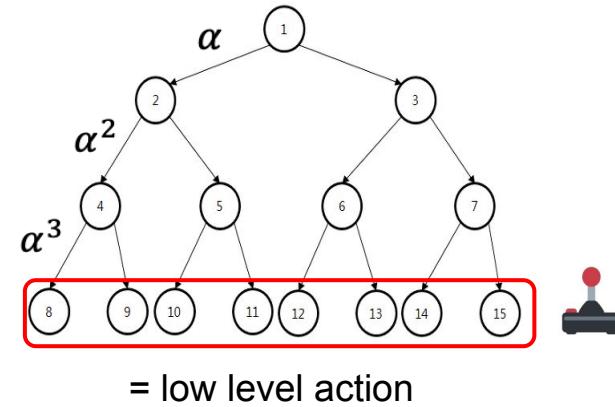
# Improving algorithm: learning hierarchical policy

- Policy should output the low-level action at the end
- To prevent excessive number of querying subgoals, we discount each query by multiplying alpha

$$Q^*(s_t, g^1, goal) \leftarrow \alpha \left( \max_g Q^*(s_t, g, g^1) \right) \{ R(s_t, g^1, goal) + \gamma \max_{\hat{g}(g^1)} Q^*(s_{t+k}, \hat{g}, goal) \}$$

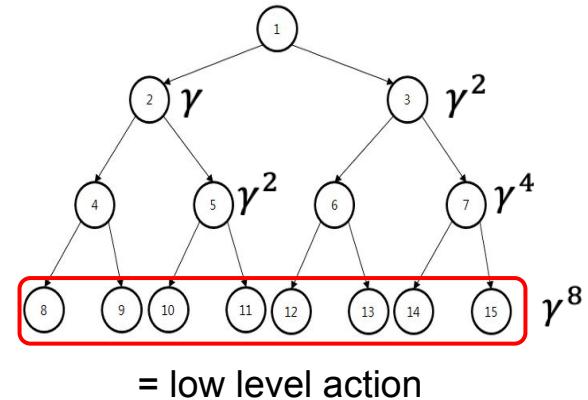
$$\begin{cases} \max_g Q_H^*(s_t, g, g^1) = 1 & \text{if } g^1 = \text{lowest level} \\ \alpha = 1 & \end{cases}$$

Best possible refinement steps may also be limited to n



# Improving algorithm: dealing with long-horizon

- Discounting factor limits the possible horizon of policy. Ex)  $(0.99)^{500} = 0.007$
- To address this, we can multiply a single gamma after the subgoal completion



$$Q^*(s_t, g, goal) \leftarrow R(s_t, g, goal) + \gamma \max_{\hat{g}(g)} Q^*(s_{t+k}, \hat{g}, goal)$$

- Grounding is done by (prob to achieve goal) = (prob to achieve subgoal) x (prob to achieve goal by subgoal)

I.e.  $Q_H^*(s_t, g^1, goal) \leftarrow \alpha \max_g Q_H^*(s_t, g, g^1) Q^*(s_t, g^1, goal)$

# Offline RL in video

- Learning in video data is promising
  - since large amount of data can be leveraged this way
- Downstream policy(e.g., robot) can utilize the video policy by information transfers
  - **H**: follow the next frame as a subgoal
  - **S**: From consequent frames, extract action by inverse dynamics
  - **P**: Use video policy representation



**State**: frame

**Action**: next frame

**Next State**: next frame



$$a = I(s, s')$$

Policy prior: video prediction model

# Offline RL in text

- offline RL can be employed in the text domain  
→ in contrast to data modelling approach (LLM)
- Goal and actions might be randomly sampled from the bunch of text
- Semantic transfer is crucial for good abstraction

**LLM can aid the training by the role of**

- (1) Dynamics (e.g., to generate dialogue)
- (2) Reward
- (3) Policy prior

$$\tau_1 = (o_0, a_0, o_1, a_1, o_2, a_2, o_3, a_3, o_4, a_4, o_5, a_5, o_6, a_6, o_7, a_7, o_8, a_8, \dots, )$$

State                      Action                      Goal

