

MINDFUL Seminar Series

3/11/2022

Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey

Cédric Colas, Tristan Karch, Oliver Sigaud, and Pierre-Yves Oudeyer

<https://arxiv.org/abs/2012.09830>

Presented by Drew Buck

University of Missouri

Department of Electrical Engineering and Computer Science



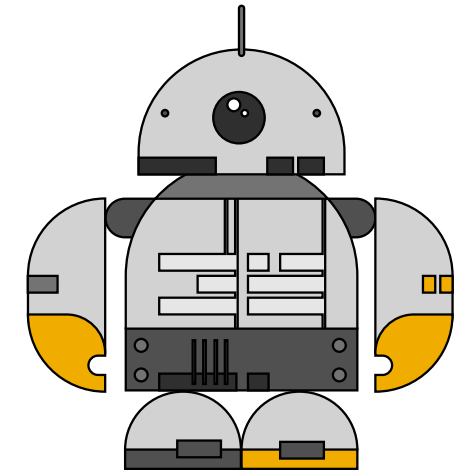
Intro

- **What is an autotelic agent?**
 - From Greek: *auto* (self) and *telos* (end, goal)
 - Agents that generate their own goals
- **Why do I care?**
 - We often model AI/ML as optimization problems
 - Minimize loss, maximize reward, search for solution, etc.
 - How to choose the goal/objective?
 - Humans demonstrate lifelong open-ended learning
 - Learn how to crawl, ask questions, interact with peers, etc.
 - Invent and pursue their own problems
 - Can we build artificial agents that do this?



Developmental RL

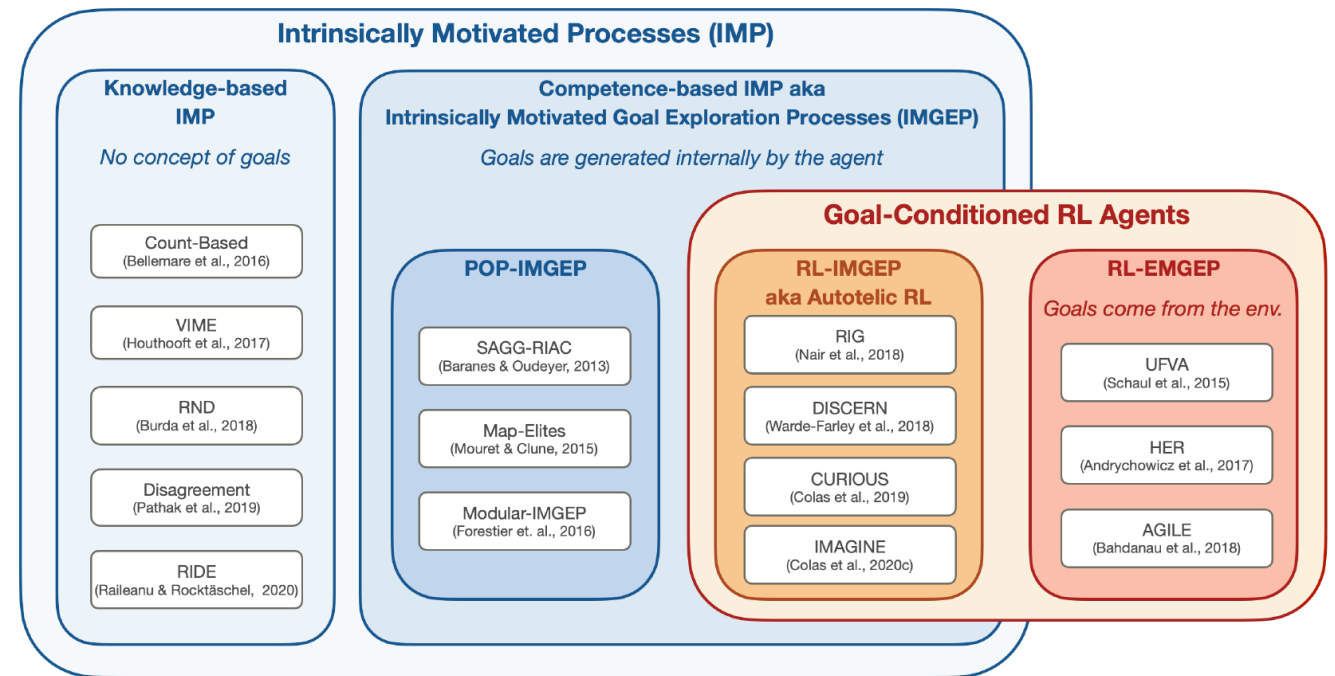
- **Developmental Reinforcement Learning**
 - Convergence of developmental robotics and RL
 - Developmental Robotics
 - Intelligence should be physically embodied
 - Modeled after children learning
 - Intrinsically motivated to explore, discover, learn
 - Often rely on population-based methods
 - Reinforcement Learning (RL)
 - Agents learn behavior through interaction
 - Seek to maximize experienced reward
 - No specific question; set of methods





Intrinsic Motivation

- **Two main types:**
 - **Knowledge-based IMs**
 - Try to predict future states
 - Reward prediction errors, experiencing dissonance, novelty, surprise, information gain, etc.
 - May be used as auxiliary reward to encourage exploration
 - **Competence-based IMs**
 - Solve self-generated problems
 - Need to represent, select, and master goals
 - Organize the acquisition of skills





The RL Problem

- Typically framed as Markov Decision Processes (MDPs)

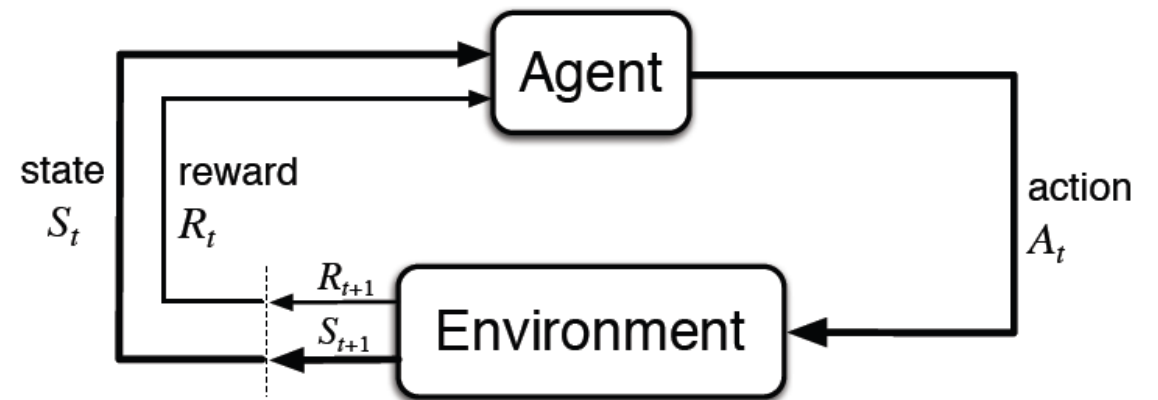
- $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, R\}$

- Environment and agent defined by $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0\}$

- Current state: $s \in \mathcal{S}$
- Initial state distribution: ρ_0
- Agent action: $a \in \mathcal{A}$
- State transition: $\mathcal{T}(s'|s, a)$

- Objective defined by reward R

- Reward given by transition: $R(s, a, s')$
- Maximize cumulative reward
 - $R_{\text{tot}} = \sum_{i=t}^{\infty} \gamma^{i-t} R(s_{i-1}, a_i, s_i)$



From Sutton & Barto (2018)



Goals

■ Defining goals:

■ In psychological research:

- “A goal is a cognitive representation of a future object that the organism is committed to approach or avoid” (Elliot & Fryer, 2008)

■ For RL agents need:

- 1) A compact representation of a goal
- 2) A way to assess progress toward the goal

Generalized definition of the goal construct for RL:

- **Goal:** a $g = (z_g, R_g)$ pair where z_g is a compact *goal parameterization* or *goal embedding* and R_g is a *goal-achievement* function.
- **Goal-achievement function:** $R_g(\cdot) = R_G(\cdot | z_g)$ where R_G is a goal-conditioned reward function.



Multi-Goal RL

- **Make an MDP handle multiple goals**
 - Replace reward function R with distribution \mathcal{R}_G
 - $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, \mathcal{R}_G\}$
- Not the same as multi-task RL, where other components $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0)$ can change
- Multi-goal RL is a particular case of multi-task RL where only the reward function changes
 - In the standard problem, \mathcal{R}_G is pre-defined by the experimenter



Goal-Conditioned Policy

- **RL agents operate according to a policy**
 - Maps states to actions
 - $a = \pi(s)$
 - Or probability of selecting action a when in state s
 - $\pi(a|s)$
- **For multi-goal RL,**
 - $\Pi: \mathcal{S} \times \mathcal{Z}_G \rightarrow \mathcal{A}$
 - \mathcal{Z}_G is the space of goal embeddings with goal space \mathcal{G}
 - Strategies:
 - Can pick a policy π from meta-policy Π with a one-hot goal embedding z_g
 - Hindsight learning: *what is the goal for which a given trajectory is optimal?*



Skill Acquisition

- **The Intrinsically Motivated Skills Acquisition Problem**
 - Agent operates in an open-ended environment
 - Needs to acquire a repertoire of skills
 - Skill is defined as a goal embedding z_g and the policy to reach it Π_g
 - Repertoire of skills is a set of goals \mathcal{G} with goal conditioned policy $\Pi_{\mathcal{G}}$
 - Reward-free MDP
 - $\mathcal{M} = \{S, \mathcal{A}, \mathcal{T}, \rho_0\}$
 - Agents (like children) must be autotelic
 - Learn to represent, generate, pursue, and master their own goals



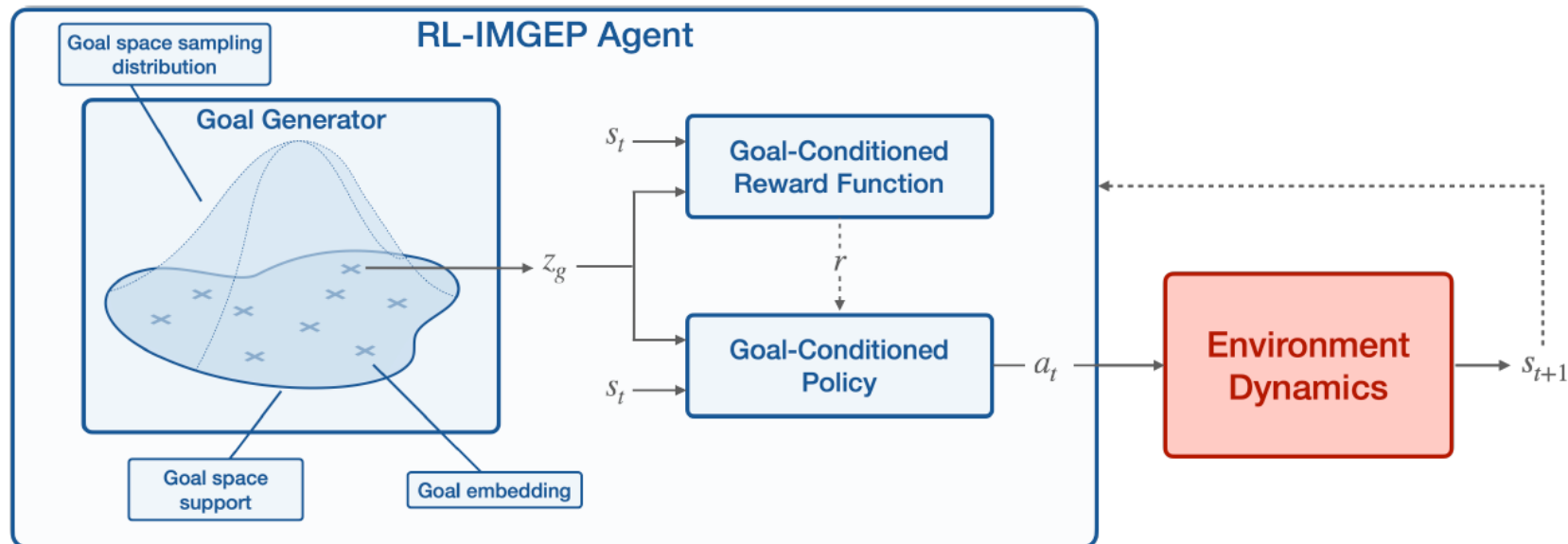
Evaluating RL-IMGEP Agents

- **How to evaluate competency of an RL-IMGEP agent?**
 - (RL-based intrinsically motivated goal exploration process)
 - *“If you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid.” – Einstein*
 - Measure exploration
 - Entropy, state coverage, interesting interactions, ...
 - Measure generalization
 - Hold out target goals from training, test on these; experimenter bias, ...
 - Measure transfer learning
 - RL-IMGEP as pre-training to bootstrap agent; eval agent on downstream task, ...
 - Open the black-box
 - Goal distribution, goal embeddings, learning trajectories, ...
 - Measure robustness
 - Large environments, distractors, non-stationary, ...



RL-IMGEP Agent

- **RL-IMGEP agents need to learn:**
 - 1) To represent goals g by compact embeddings z_g
 - 2) To represent the goal space $\mathcal{Z}_G = \{z_g\}_{g \in \mathcal{G}}$
 - 3) A goal distribution to sample goals $\mathcal{D}(z_g)$
 - 4) A goal-conditioned reward function \mathcal{R}_G



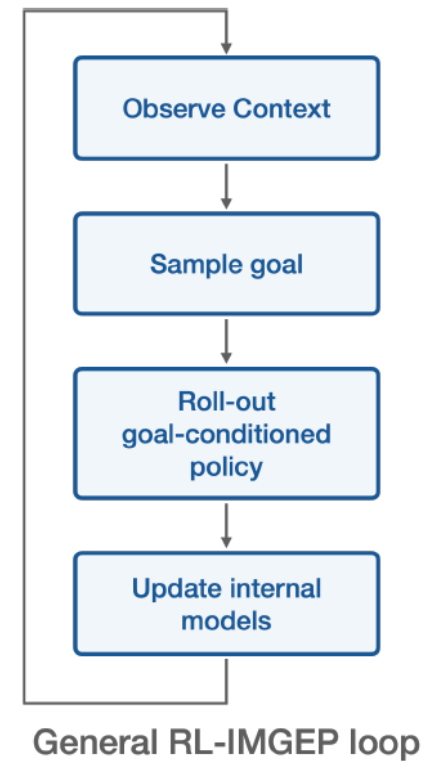


Training an Autotelic Agent

Algorithm 1 Autotelic Agent with RL-IMGEP

Require: environment \mathcal{E}

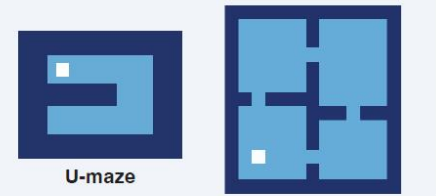
- 1: **Initialize** empty memory \mathcal{M} ,
 - 2: goal-conditioned policy $\Pi_{\mathcal{G}}$, goal-conditioned reward $R_{\mathcal{G}}$,
 - 3: goal space $\mathcal{Z}_{\mathcal{G}}$, goal sampling policy GS .
 - 4: **loop**
 - ▷ *Observe context*
 - 5: Get initial state: $s_0 \leftarrow \mathcal{E}.reset()$
 - ▷ *Sample goal*
 - 6: Sample goal embedding $z_g = GS(s_0, \mathcal{Z}_{\mathcal{G}})$.
 - ▷ *Roll-out goal-conditioned policy*
 - 7: Execute a roll-out with $\Pi_g = \Pi_{\mathcal{G}}(\cdot | z_g)$
 - 8: Store collected transitions $\tau = (s, a, s')$ in \mathcal{M} .
 - ▷ *Update internal models*
 - 9: Sample a batch of B transitions: $\mathcal{M} \sim \{(s, a, s')\}_B$.
 - 10: Perform Hindsight Relabelling $\{(s, a, s', z_g)\}_B$.
 - 11: Compute internal rewards $r = R_{\mathcal{G}}(s, a, s' | z_g)$.
 - 12: Update policy $\Pi_{\mathcal{G}}$ via RL on $\{(s, a, s', z_g, r)\}_B$.
 - 13: Update goal representations $\mathcal{Z}_{\mathcal{G}}$.
 - 14: Update goal-conditioned reward function $R_{\mathcal{G}}$.
 - 15: Update goal sampling policy GS .
 - 16: **return** $\Pi_{\mathcal{G}}, R_{\mathcal{G}}, \mathcal{Z}_{\mathcal{G}}$
-





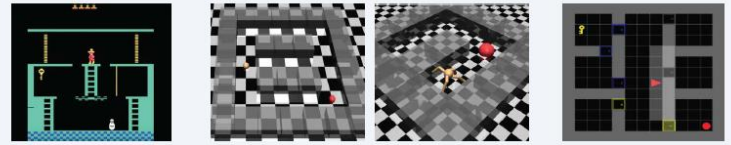
Typology of Goal Representations

Toy Environments



U-maze
Four rooms

Hard-Exploration Environments

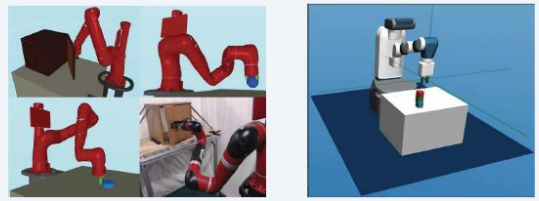


Montezuma's Revenge
DISCERN (Wardle-Farley et al. 2018)
GO-EXPLORE (Ecoffet et al. 2020)
AGENT57 (Badia et al. 2020b)

Ant Maze
GOALGAN (Fiorenza et al. 2018)

MiniGrid
AMIGO (Campero et al. 2018)

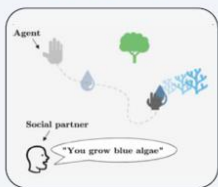
Object Manipulation Environments



Robot-arm
SKEW-FIT (Pong et al. 2019)


Fetch
CURIOUS (Colas et al. 2019)
RIG (Nair et al. 2018)
DESCTR (Akakazia et al. 2020)

Interactive Environment with Language Supervision



Playground
IMAGINE (Colas et al. 2020c)

Procedurally Generated Environments



XLand
XLAND OEL (Team et al. 2021)

Task Env Sulte
SLIDE (Fang et al. 2021)

Examples of environments in autotelic RL approaches.



Multiple Objectives

- **Goals as choices between multiple objectives**
 - Goals can be expressed as a list of different objectives the agent can choose from.
 - **Embedding:**
 - z_g is a one-hot encoding of the current objective among N available
 - $z_g^i = (\mathbf{1}_{j=i})_{j=[1..N]}$
 - **Reward function:**
 - N distinct reward functions
 - $R_g(\cdot) = R_i(\cdot)$ if $z_g = z_g^i$



Target Features

- **Goals as target features of states**
 - Goals can be expressed as target features of the state the agent desires to achieve.
 - **Embedding:**
 - State representation function φ maps state space to embedding space
 - $\mathcal{Z} = \varphi(\mathcal{S})$
 - Goal embeddings z_g are target points in \mathcal{Z}
 - Target block coordinates, agent positions, image-based goals
 - **Reward function:**
 - Reward function R_G is based on a distance metric D
 - E.g., $R_g = R_G(s|z_g) = -\alpha \times D(\varphi(s), z_g)$
 - Sparse: $R_G(s|z_g) = 1$ if $D(\varphi(s), z_g) < \epsilon$, 0 otherwise



Abstract Binary Problems

- **Goals as abstract binary problems**
 - Goals can be expressed as a set of constraints such that these constraints are either verified or not (binary goal achievement).
 - **Embedding:**
 - Finite embedding space
 - Language-based predicates
 - *“Sort the objects by size”*
 - *“Open the yellow door after you open a purple door”*
 - *“See opponent while holding a yellow pyramid or while yellow sphere is not on a green floor”*
 - **Reward function:**
 - Reward function $R_G(s|z_g)$ is determined based on if the state s verifies the goal semantics (positive reward) or not (null reward)



Multi-Objective Balance

■ Goals as a Multi-Objective Balance

- Goals can be expressed as a parameterization of a particular mixture of multiple objectives that should be maximized.
- **Embedding:**
 - Goals are sets of weights that balance the different objectives
 - $z_g = (\beta_i)_{i=[1..N]}$, where β_i is the weight for objective i for N objectives
- **Reward function:**
 - Reward is a convex combination of the objectives
 - $R_g(s) = \sum_{i=1}^N \beta_g^i R^i(s)$, where $z_g = \beta = \beta_i^g \mid_{i \in [1..N]}$ is the set of weights



Learning Goal Representations

- **How to learn goal representations?**
 - **Assuming pre-defined goal representation**
 - Given as part of the problem definition
 - E.g., go to location, combinatorial state space, ...
 - **Learning goal embeddings**
 - Language-based approaches, generative models of states, ...
 - **Learning the reward function**
 - Goal-conditioned reward function, empowerment, ...
 - **Learning the support of the goal distribution**
 - Option framework, bottleneck states, in vs. out of distribution goals, ...



How to Prioritize Goal Selection

- **Autotelic agents need to select their own goals**
- **Automatic Curriculum Learning for Goal Selection**
 - Some goals are trivial, others impossible
 - Organize goal sampling to maximize long-term performance improvement
 - **Intermediate or uniform difficulty**
 - Should we focus on goals of intermediate difficulty or sample goals of all levels of difficulty uniformly?
 - **Novelty – diversity**
 - Maximize empowerment (choose goals that give agent most control)
 - Select goals in sparse areas of goal space or uniformly distributed?
 - **Medium-term learning progress**
 - Recognize and pursue goals where the agent can make progress
 - Avoid goals that are currently too easy, hard, or impossible
- **Hierarchical Reinforcement Learning for Goal Sequencing**
 - Decompose tasks with long-term dependencies into smaller sub-tasks



RL-IMGEP Approaches

Approach	Goal Type	Goal Rep.	Reward Function	Goal sampling strategy
RL-IMGEPs that assume goal embeddings and reward functions				
(Fournier et al., 2018)	Target features (+tolerance)	Pre-def	Pre-def	LP-Based
HAC (Levy et al., 2018)	Target features	Pre-def	Pre-def	HRL
HIRO (Nachum et al., 2018)	Target features	Pre-def	Pre-def	HRL
CURIOUS (Colas et al., 2019)	Target features	Pre-def	Pre-def	LP-based
CLIC (Fournier et al., 2019)	Target features	Pre-def	Pre-def	LP-based
CWYC (Blaes et al., 2019)	Target features	Pre-def	Pre-def	LP-based
GO-EXPLORE (Ecoffet et al., 2020)	Target features	Pre-def	Pre-def	Novelty
NGU (Badia et al., 2020b)	Objectives balance	Pre-def	Pre-def	Uniform
AGENT 57 (Badia et al., 2020a)	Objectives balance	Pre-def	Pre-def	Meta-learned
DECSTR (Akakzia et al., 2020)	Binary problem	Pre-def	Pre-def	LP-based
SLIDE (Fang et al., 2021)	Skill index	Pre-def	Pre-def	Novelty (PCG)
XLAND OEL (Team et al., 2021)	Binary problem	Pre-def	Pre-def	Intermediate difficulty
RL-IMGEPs that learn their goal embedding and assume reward functions				
RIG (Nair et al., 2018)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
GOALGAN (Florensa et al., 2018)	Target features	Pre-def + GAN	Pre-def	Intermediate difficulty
(Florensa et al., 2019)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
SKEW-FIT (Pong et al., 2019)	Target features (images)	Learned (VAE)	Pre-def	Diversity
SETTER-SOLVER (Racanière et al., 2019)	Target features (images)	Learned (Gen. model)	Pre-def	Uniform difficulty
MEGA (Pitis et al., 2020)	Target features (images)	Learned (VAE)	Pre-def	Novelty
CC-RIG (Nair et al., 2020)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
AMIGO (Campero et al., 2020)	Target features (images)	Learned (with policy)	Pre-def	Adversarial
GRIMGEP (Kovač et al., 2020)	Target features (images)	Learned (with policy)	Pre-def	Diversity and ALP
Full RL-IMGEPs				
DISCERN (Warde-Farley et al., 2018)	Target features (images)	Learned (with policy)	Learned (similarity)	Diversity
DIAYN (Eysenbach et al., 2018)	Discrete skills	Learned (with policy)	Learned (discriminability)	Uniform
(Hartikainen et al., 2019)	Target features (images)	Learned (with policy)	Learned (distance)	Intermediate difficulty
(Venkattaramanujam et al., 2019)	Target features (images)	Learned (with policy)	Learned (distance)	Intermediate difficulty
IMAGINE (Colas et al., 2020c)	Binary problem (language)	Learned (with reward)	Learned	Uniform + Diversity
VGRL (Choi et al., 2021)	Target features	Learned	Learned	Empowerment

Table 1: **A classification of autotelic RL-IMGEP approaches.** Autotelic approaches require agents to sample their own goals. The proposed classification groups algorithms depending on their degree of autonomy: 1) RL-IMGEPs that rely on pre-defined goal representations (embeddings and reward functions); 2) RL-IMGEPs that rely on pre-defined reward functions but learn goal embeddings and 3) RL-IMGEPs that learn complete goal representations (embeddings and reward functions). For each algorithm, we report the type of goals being pursued (see Section 4), whether goal embeddings are learned (Section 5), whether reward functions are learned (Section 5.3) and how goals are sampled (Section 6). We mark in bold algorithms that use a developmental approaches and explicitly pursue the intrinsically motivated skills acquisition problem.



Open Challenges

- **Challenge #1: Targeting a Greater Diversity of Goals**
 - Time extended goals
 - E.g., *“knock three times”, “get the blue ball that was on the table yesterday, then roll it towards me.”*
 - Learning goals
 - E.g., *“I’m going to learn about knitting so I can knit a pullover to my friend for his birthday.”*
 - Goals as optimization under selected constraints
 - E.g., maximize a metric (walking speed) within constraints (maintain power consumption below a given threshold).
 - Meta-diversity of goals
 - Different types of goal representations; hierarchical goal space, ...



Open Challenges (cont.)

- **Challenge #2: Learning to Represent Diverse Goals**
 - Often limited to pre-existing goal embeddings or reward functions
 - Methods that learn autonomously tend to be restricted to specific domains
- **Challenge #3: Imagining Creative Goals**
 - Sampling goals outside of the modeled goal distribution
- **Challenge #4: Composing Skills for Better Generalization**
 - Transfer knowledge between skills; infer and compose new skills
- **Challenge #5: Leveraging Socio-Cultural Environments**
 - Learning from social interaction
 - How to make agents that are both autonomous and teachable?



Discussion and Conclusion

- **Developmental Reinforcement Learning**
 - Intersection of developmental robotics and RL
 - Intrinsically motivated skills acquisition problem
 - Autotelic agents that can learn to represent, generate, and achieve their own goals
- **Categorization of the goal construct**
 - Compact pairing of the goal representation and goal achievement function
 - Goal-conditioned RL approaches
- **Learning Agent as a Curious Scientist**
 - Build hypotheses about the world and explore it to find out if they are true
 - Challenge itself to learn about and interact with the world to grow skills and knowledge
 - Guided by curiosity; decide its own agenda
 - Immersed in socio-cultural environment like humans