Machine Teaching

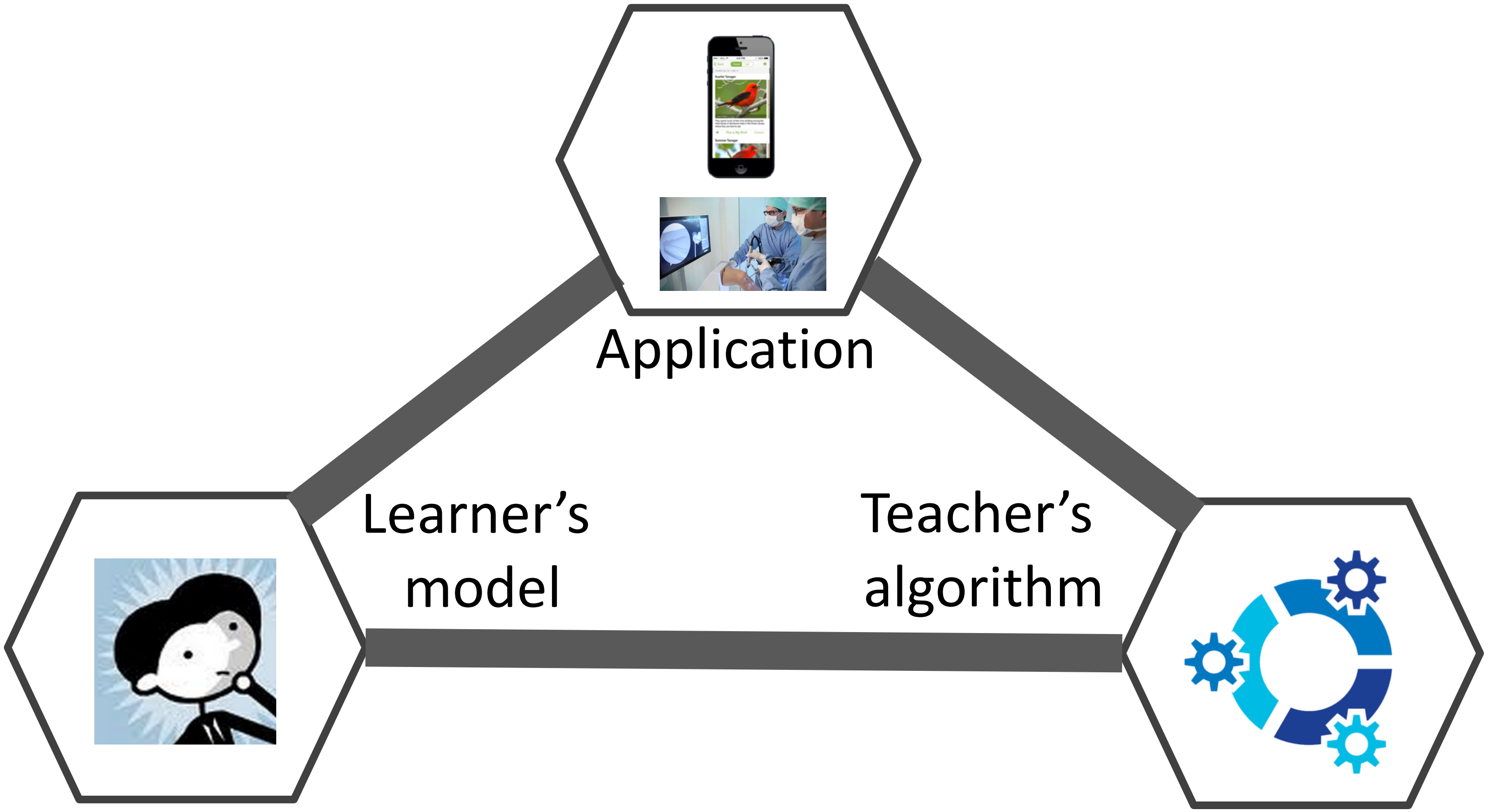
Adish Singla

CMMRS, August 2019

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

Machine Teaching: Key Components



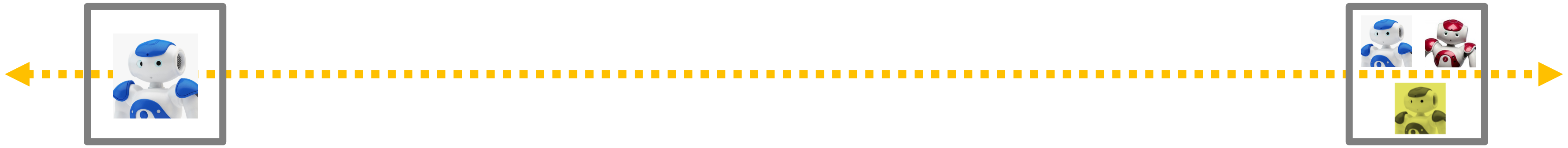
Machine Teaching: Problem Space

- Type and complexity of task

[arXiv ’19]

- Type and model of learning agent

- Teacher’s knowledge and observability



Applications: Language Learning

duolingo

memrise BETA
GROW YOUR MIND

- Over 300+ million students
- Based on **spaced repetition** of flash cards

Can we compute **optimal personalized schedule** of repetition?

Setup: Learning via Flashcards

- n: number of concepts (flashcards)
- T: total time learning steps

Teaching Interaction using Flashcards

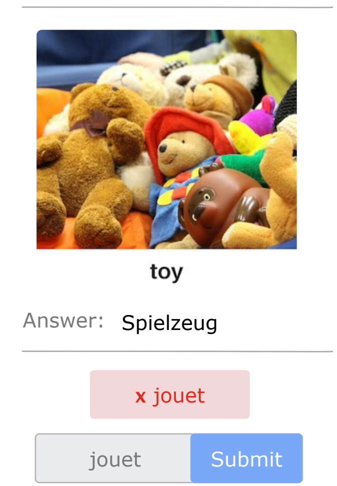
Interaction at time $t = 1, 2, \ldots T$

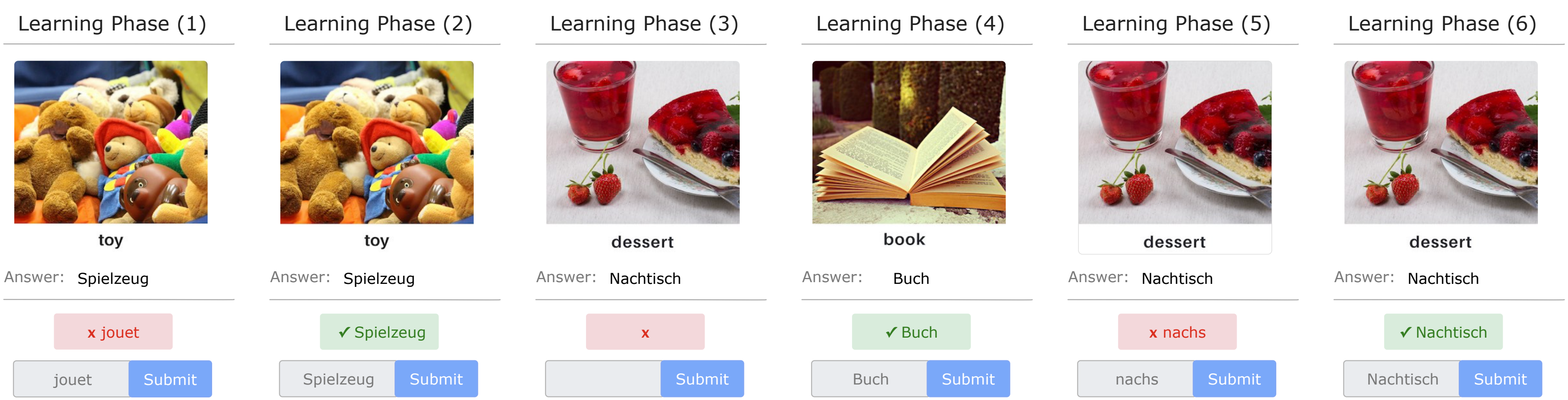
1. Teacher displays a flashcard $x_t \in \{1, 2, .., n\}$
2. Learner's recall is $y_t \in \{0, 1\}$
3. Teacher provides the correct answer

1

3

2





Research question: What is an optimal schedule of displaying cards?

Background on Teaching Policies

Example setup

- $n = 5$ concepts given by $\{a, b, c, d, e\}$
- $T = 20$

Random teaching policy

$$a \rightarrow b \rightarrow a \rightarrow e \rightarrow c \rightarrow d \rightarrow a \rightarrow d \rightarrow c \rightarrow a \rightarrow b \rightarrow e \rightarrow a \rightarrow b \rightarrow d \rightarrow e \rightarrow$$

Round-robin teaching policy

$$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow a \rightarrow$$

Key limitation: Schedule agnostic to learning process

Background on Teaching Policies

The Pimsleur method (1967)

- Used in mainstream language learning platforms
- Based on spaced repetition ideas
  - **Spacing effect:** practice should spread out over time
  - **Lag effect:** spacing between practices should gradually increase

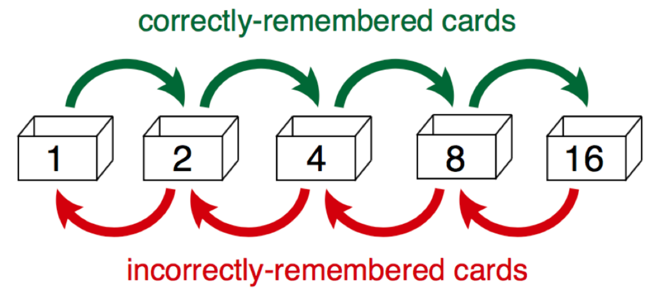
$$a \to b \to a \to b \to c \to a \to c \to b \to d \to c \to d \to a \to b \to d \to c \to e \to$$

Key limitation: Non-adaptive schedule ignores learner's responses

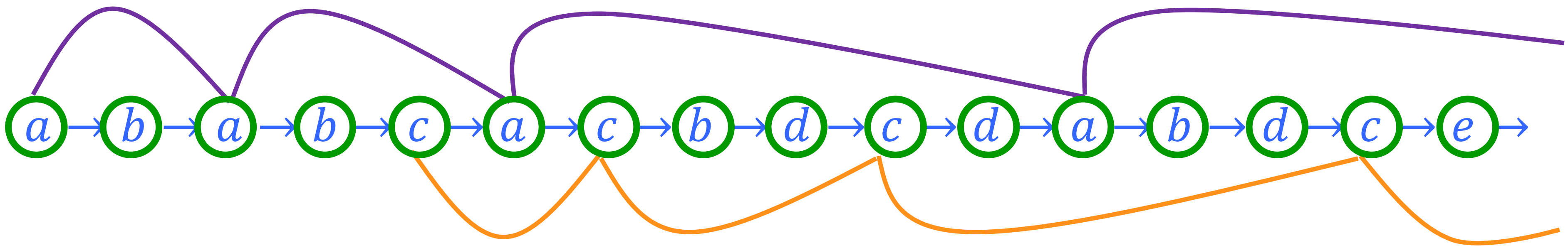
Background on Teaching Policies

The Leitner system (1972)

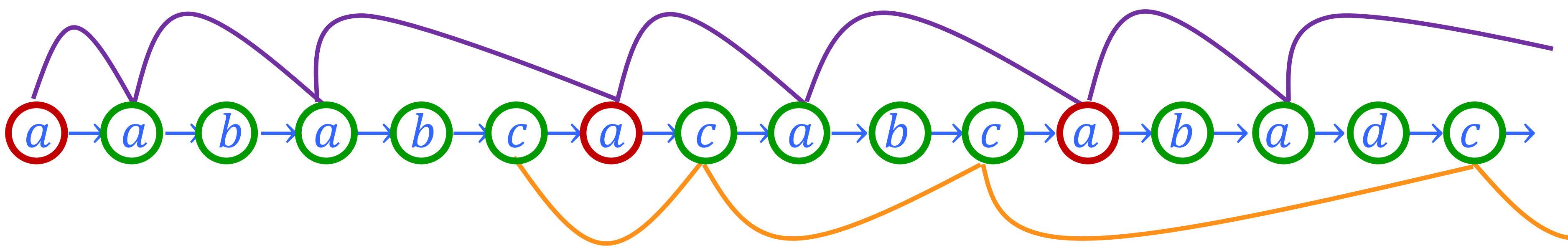
- Used by Duolingo in its first launch
- Adaptive spacing intervals



Schedule 1



Schedule 2



Key limitation: No guarantees on the optimality of the schedule

Learner: Modeling Memory & Responses

Half-life regression (HLR) model

- Introduced by [Settles, Meeder @ ACL'16]
- History up to time t given by $(x_{1:t}, y_{1:t})$
- For a concept x:
  - Last time step when concept x was taught is $l_t^x \in \{1, .., t\}$
  - Learner's mastery for concept x at time t is h_t^x

Recall probability in future under HLR model

- Probability to recall concept x at future time $\tau \in \{t+1, .., T\}$ is

$$g^x(\tau, (x_{1:t}, y_{1:t})) = 2^{-\left((\tau - l_t^x)/h_t^x\right)}$$

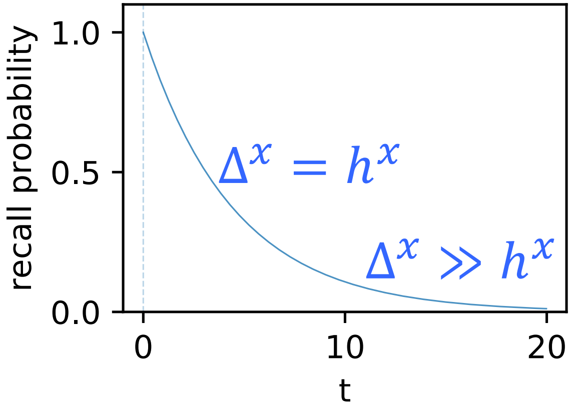
Learner: Modeling Memory & Responses

- Recall probability based on exponential forgetting curve

$$g^x(\tau, (x_{1:t}, y_{1:t})) = 2^{-\left(\frac{\Delta^x}{h^x}\right)}$$

Δ^x: time past since concept x was taught

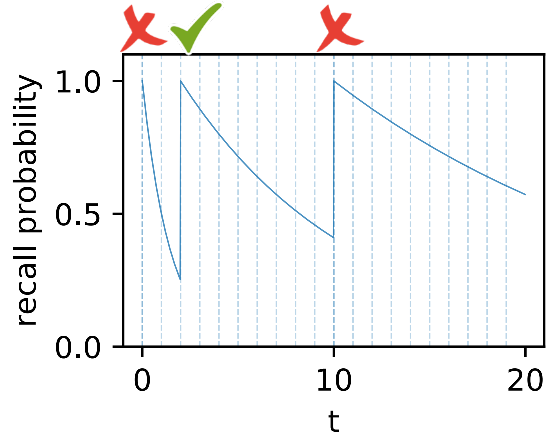
h^x: current "half-life" of concept x



- Half-life h^x changes when learner is taught concept x
- Changes parameterized by (a^x, b^x)

✓ $h^x \mathrel{+}= a^x$

✗ $h^x \mathrel{+}= b^x$

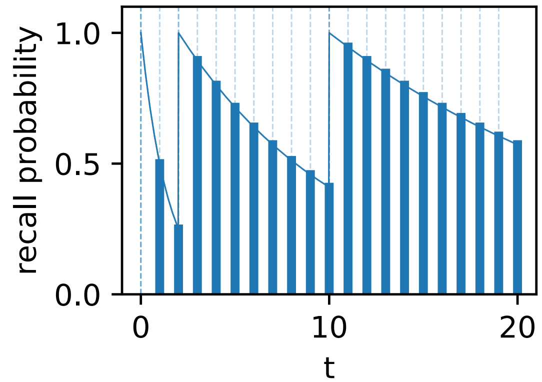


Teacher: Scheduling as Optimization

Teacher's objective function

- Given a sequence of concepts and observations $x_{1:T}, y_{1:T}$, we define

$$f(x_{1:T}, y_{1:T}) = \frac{1}{nT}\sum_{x=1}^{n}\underbrace{\sum_{t=1}^{T} g^{x}(t+1, (x_{1:t}, y_{1:t}))}_{\text{Area under the curve}}$$



Optimization problem

- Teaching policy $\pi: (x_{1:t-1}, y_{1:t-1}) \to \{1,2,..,n\}$
- Denote average utility of a policy π as $F(\pi) := \mathbb{E}_{(x,y)}\left[f(x^{\pi}_{1:T}, y^{\pi}_{1:T})\right]$
- Optimization problem is given by

$$\pi^* = \mathrm{argmax}_{\pi}\, F(\pi)$$

Teacher: Algorithm

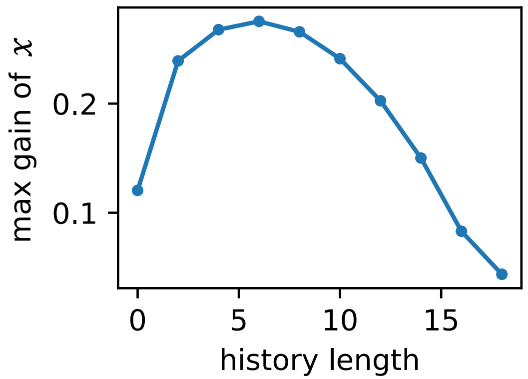
Adaptive greedy algorithm

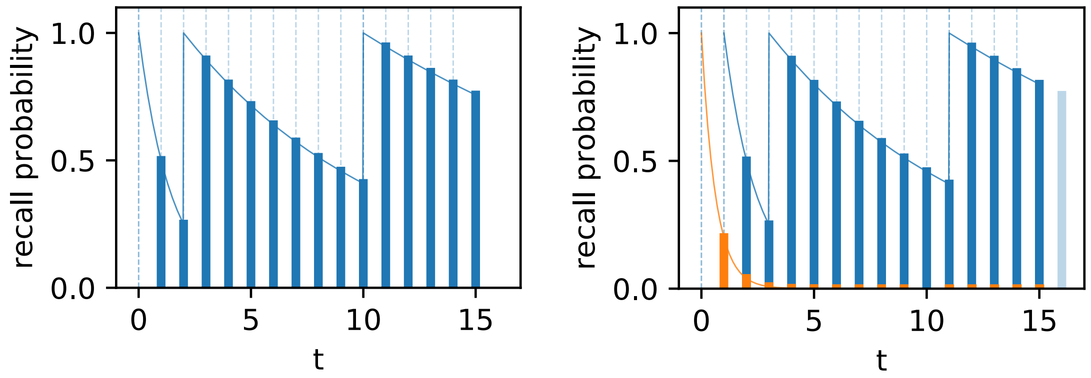
- for $t = 1, 2, \dots T$:
  - Select $x_t \leftarrow \operatorname{argmax}_x \mathbb{E}_{(y)}[f(x_{1:t-1} \oplus x, y_{1:t-1} \oplus y)] - f(x_{1:t-1}, y_{1:t-1})$
  - Observe learner's recall $y_t \in \{0, 1\}$
  - Update $x_{1:t} \leftarrow x_{1:t-1} \oplus x_t$; $y_{1:t} \leftarrow y_{1:t-1} \oplus y_t$

Teacher: Theoretical Guarantees

Characteristics of the problem

- Adaptive sequence optimization
- Non-submodular
  - Gain of a concept x can increase given longer history
  - Captured by submodularity ratio γ over sequences
- Post-fix non-monotone
  - $f(\textbf{orange} \oplus \textbf{blue}) < f(\textbf{blue})$
  - Captured by curvature ω over sequences





Teacher: Theoretical Guarantees

Guarantees for general case (any memory model)

- Utility of π^{gr} (greedy policy) compared to π^{opt} is given by

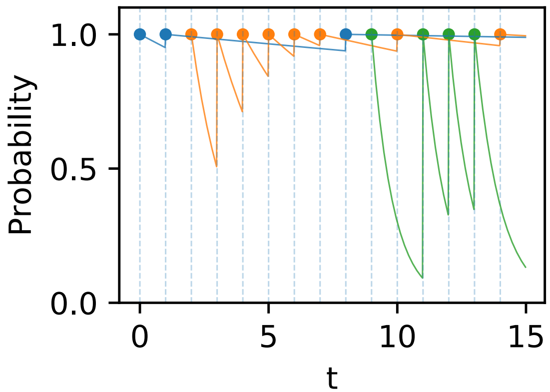
$$F(\pi^{\text{gr}}) \geq F(\pi^{\text{opt}}) \sum_{t=1}^{T} \left(\frac{\gamma_{T-t}}{T} \prod_{\tau=0}^{t-1} \left(1 - \frac{\omega_\tau \cdot \gamma_\tau}{T} \right) \right)$$

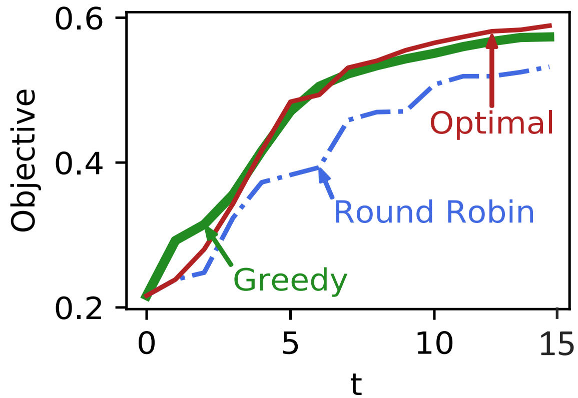
Theorem 1

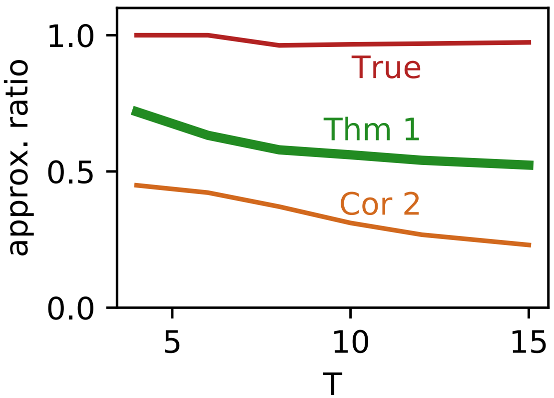
$$\geq F(\pi^{\text{opt}}) \frac{1}{\omega_{\max}} \left(1 - e^{-\omega_{\max} \cdot \gamma_{\min}} \right)$$

Corollary 2

- Illustration with $T=15$ and $n=3$ concepts using HLR model







Teacher: Theoretical Guarantees

Guarantees for the HLR model

- Consider the task of teaching n concepts where each concept is following an independent HLR model with the same parameters $(a^x = z, b^x = z) \forall\, x \in \{1,2,..,n\}$

Theorem 3: A sufficient condition for the algorithm to achieve a high utility of at least $(1 - \epsilon)$ is given by $\mathrm{T} \geq O\left(\frac{n^2 \cdot \exp(-z)}{\epsilon}\right)$.

Results: Simulated Learners

HLR learner model

- Equal proportion of two types of concepts
  - easy concepts with parameters $(a = 10, b = 5)$
  - difficult concepts with parameters $(a = 3, b = 1.5)$

Algorithms

- **RD**: Random, **RR**: Round-robin
- **LR**: Least-recall (generalization of Pimsleur and Leitner system)
- **GR**: Our algorithm

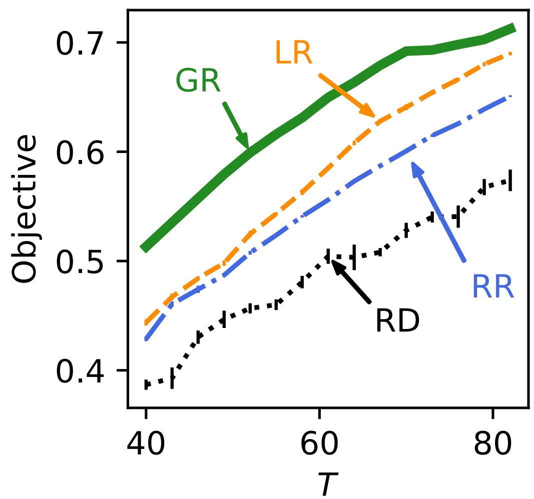
Performance metrics

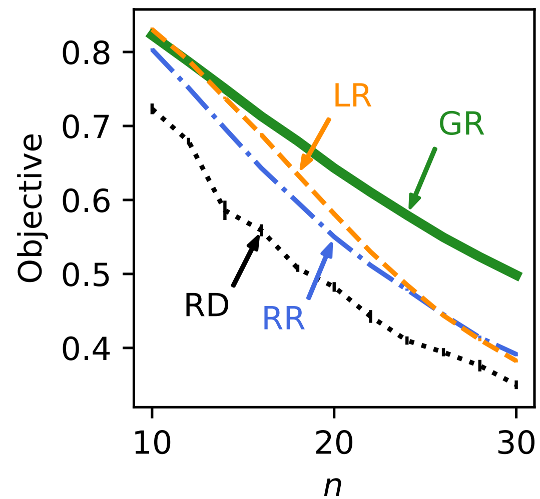
- Objective function value
- Recall in near future after finishing teaching (Recall at $\text{“}T + 10\text{”}$)

Results: Simulated Learners

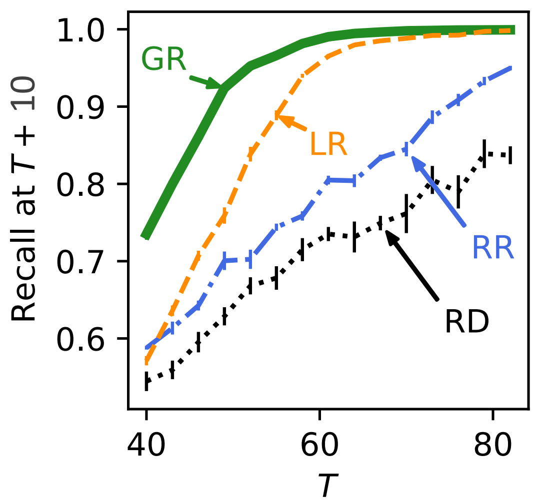
Varying $\boldsymbol{T}$ (fix $n = 20$) | **Varying $\boldsymbol{n}$ (fix $T = 60$)**

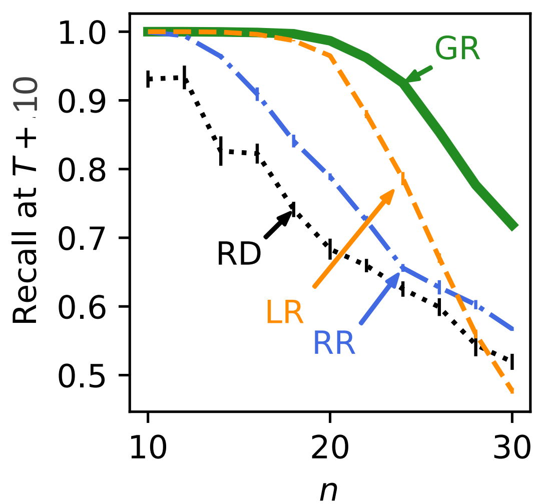
Objective value





Recall in future

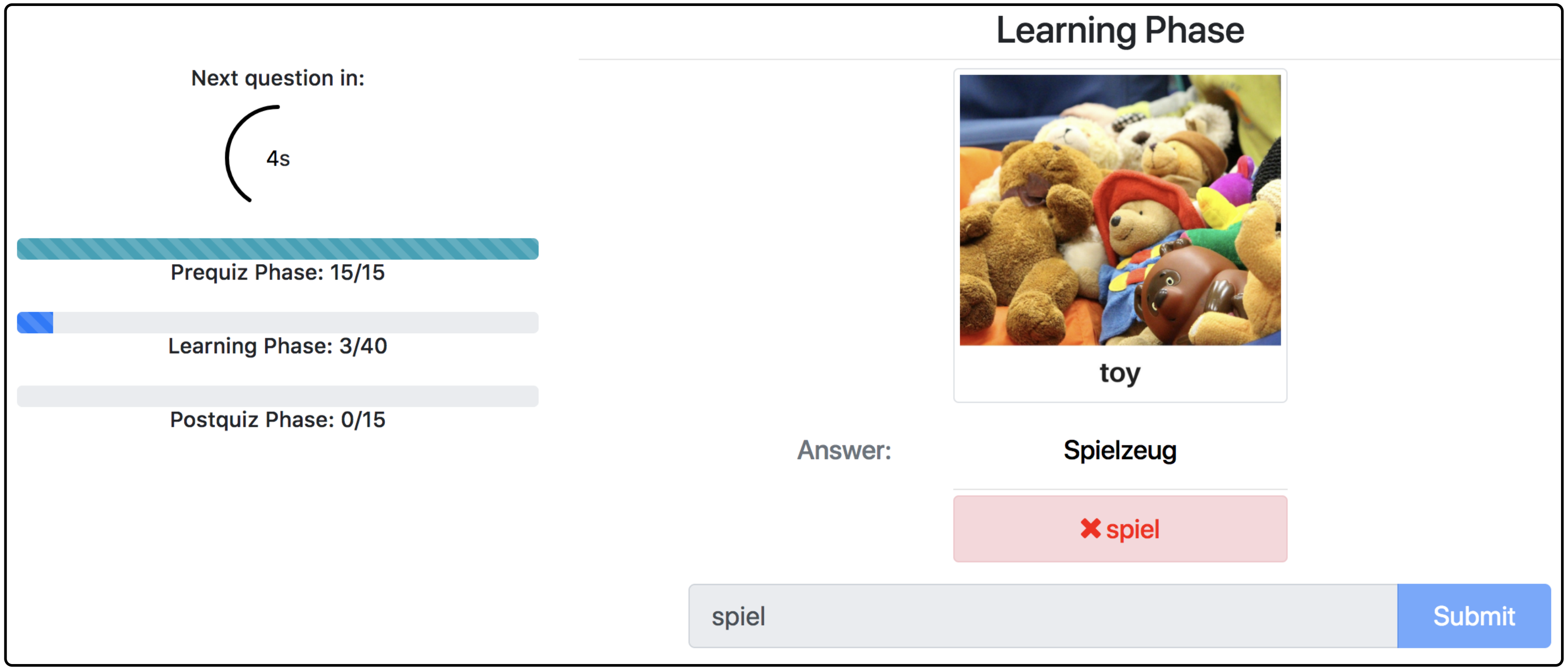




Results: Human Learners

Online learning platforms

- German vocabulary: https://www.teaching-german.cc/
- Species names: https://www.teaching-biodiversity.cc/



- Performance based on (**post-quiz** score – **pre-quiz** score)

Results (German): Human Learners

- 80 participants from a crowdsourcing platform (20 per algorithm)
- Dataset of 100 English-German word pairs
  - GR parameters: $(a = 6, b = 2)$ for all concepts
- $T = 40, n = 15$

	GR	LR	RR	RD
Avg. gain	**0.572**	0.487	0.462	0.467
p-value	-	0.0538	0.0155	0.0119

Results (Biodiversity): Human Learners

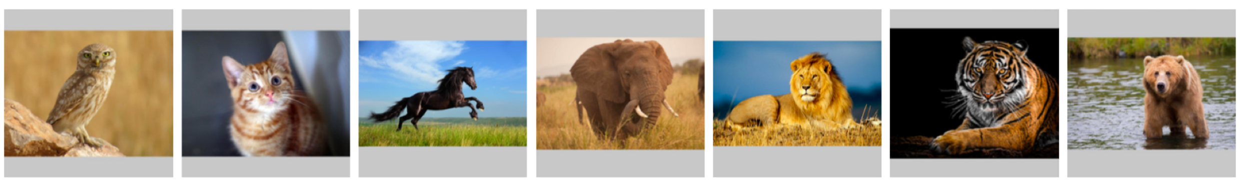
- 320 participants from a crowdsourcing platform (80 per algorithm)
- Dataset of 50 animal images of common and rare species
  - GR parameters: $(a = 10, b = 5)$ for common, $(a = 3, b = 1.5)$ for rare
- $T = 40, n = 15$

All species

	GR	LR	RR	RD
Avg. gain	**0.475**	0.411	0.390	0.251
p-value	-	0.0021	0.0001	0.0001

Rare species

	GR	LR	RR	RD
Avg. gain	**0.766**	0.668	0.601	0.396
p-value	-	0.0001	0.0001	0.0001

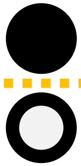


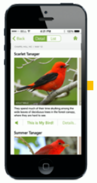
(a) Common: Owl, Cat, Horse, Elephant, Lion, Tiger, Bear

(b) Rare: Angwantibo, Olinguito, Axolotl, Ptarmigan, Patrijshond, Coelacanth, Pyrrhuloxia

Machine Teaching: Problem Space

- Type and complexity of task

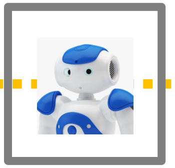




[arXiv ’19]

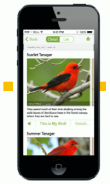
- Type and model of learning agent

- Teacher’s knowledge and observability



Machine Teaching: Problem Space

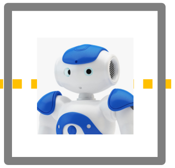
- Type and complexity of task



[NeurIPS ’18]
[IJCAI ’19]
[arXiv ’19]

- Type and model of learning agent

- Teacher’s knowledge and observability

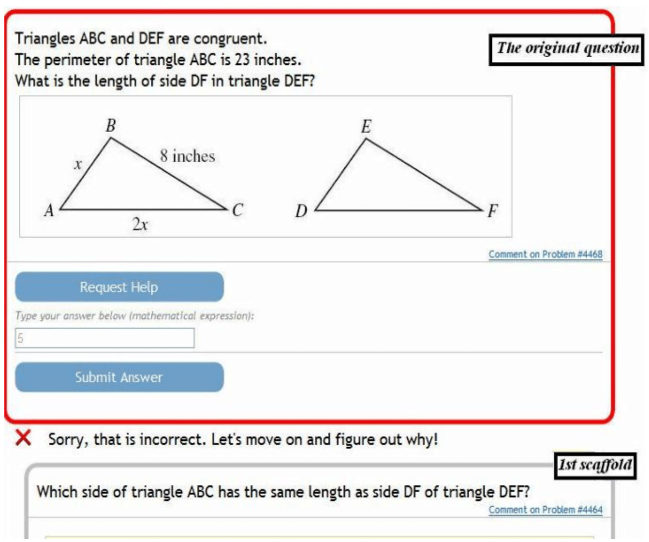


Applications: Training Simulators

VIRTAMED
WE SIMULATE REALITY

Key limitation: No automated or personalized curriculum of tasks

Applications: Skill Assessment and Practice



Key limitation: No automated or personalized curriculum of tasks

Sequential Decision Making: Ingredients

Key ingredients

- A sequence of actions with long term consequences
- Delayed feedback
  - Safely reaching the destination in time
  - Successfully solving the exercise
  - Winning or losing a game
- Main components
  - **Environment** representing the problem
  - **Student** is the learning agent taking actions
  - **Teacher** helping the student to learn faster

Sequential Decision Making: Environment

Markov Decision Process $M := (S, A, P, S_{init}, S_{end}, R)$

- S: states of the environment
- A: actions that can be taken by agent
- $P(s'|s, a)$: the transition of the environment when action is taken
- S_{init}: defines a set of initial states
- S_{end}: defines a set of terminal states
- $R(s, a)$: reward function

Sequential Decision Making: Policy

Agent's policy π

- $\pi(s) \rightarrow a$: A deterministic policy
- $\pi(s) \rightarrow P(a)$: A stochastic policy

Utility of a policy

- Expected total reward when executing a policy π is given by

$$U^{\pi} = \mathbb{E}_{P,\pi}\left[\sum_{\tau} R(s_{\tau}, a_{\tau})\right]$$

- Agent's goal is to learn an optimal policy

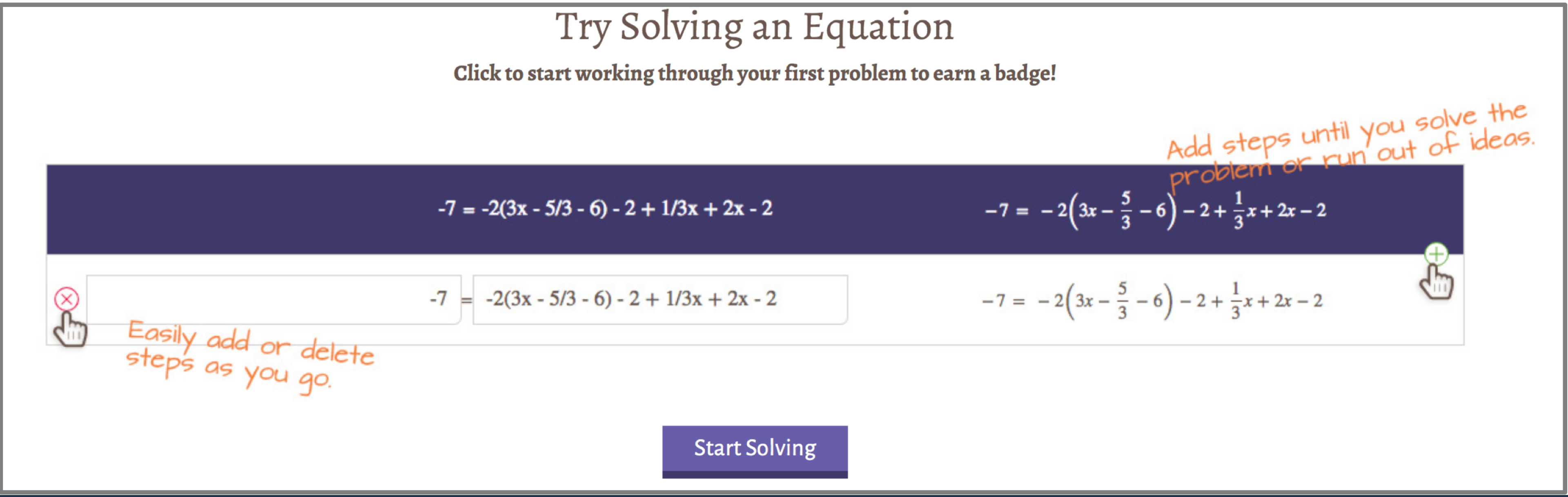
$$\pi^{*} = \operatorname{argmax}_{\pi} U^{\pi}$$

An Example: Car Driving Simulator

- State s represented by a feature vector $\phi(s)$
  (location, speed, acceleration, car-in-front, HOV, …)
- Action a could be discrete/continuous
  {left, straight, right, brake, speed+, speed-, …}
- Transition $P(s'|s, a)$ defines how world evolves
  (stochastic as it depends on other drivers in the environment)
- $R(s, a)$ defines immediate reward, e.g.,
  - 100 if $s \in S_{end}$
  - -1 if $s \notin S_{end}$
  - -10 if s represents ``accident”
- Policy π^* dictates how an agent should drive

An Example: Tutoring System for Algebra

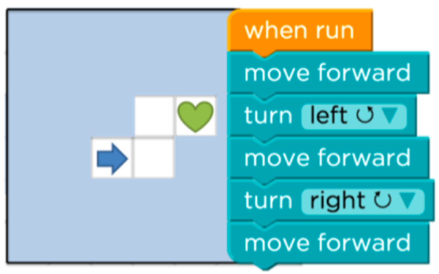
- State s represented by the current layout of variables
- Action a could be {move, combine, distribute, stop, …}
- Transition $P(s'|s,a)$ is deterministic
- $R(s,a)$ defines immediate reward, e.g.,
  - 100 if $s \in S_{end}$
  - -1 if $s \notin S_{end}$



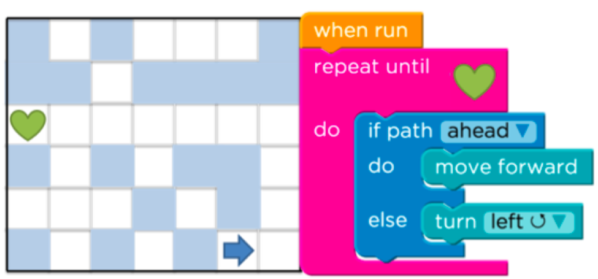
- State s could be represented by
  - raw source code
  - abstract syntax tree (AST)
  - execution behavior
  - …

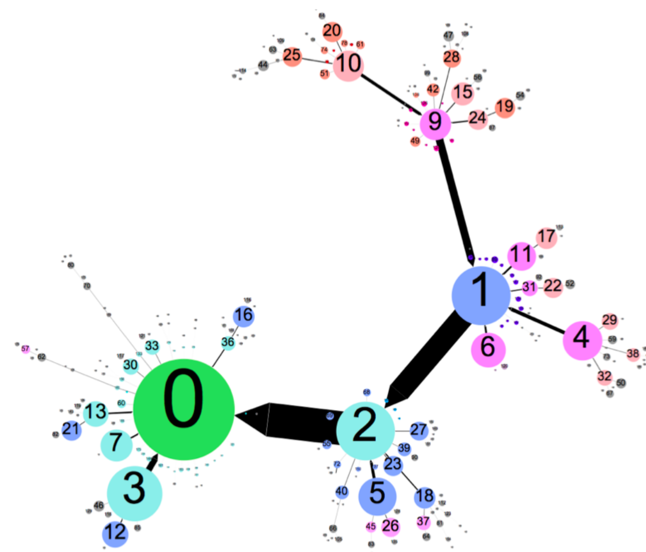
- Action a could be eligible updates (e.g., allowed by the interface)

HoC Problem 4



HoC Problem 18





Learning Settings: Reward Signals

- Standard setting in reinforcement learning (RL)
- P is known, R is known
  - Mode-based planning algorithms (e.g., Dynamic Programming)
- P, R are both unknown
  - Model-free learning algorithms (e.g., Q-learning)
- A wrong model of P is known
  - Algorithms with robustness and safety criteria

(Book) Reinforcement Learning: An Introduction [Barto and Sutton 2018]

Learning Settings: Demonstrations

- Learning via observing behavior of another agent
- Behavioral cloning
  - Direct policy learning from observed demonstrations
  - E.g., Dagger algorithm
- Inverse reinforcement learning (IRL)
  - Recover reward function explaining observed demonstrations
  - E.g., Maximum Causal Entropy algorithm (MCE-IRL)

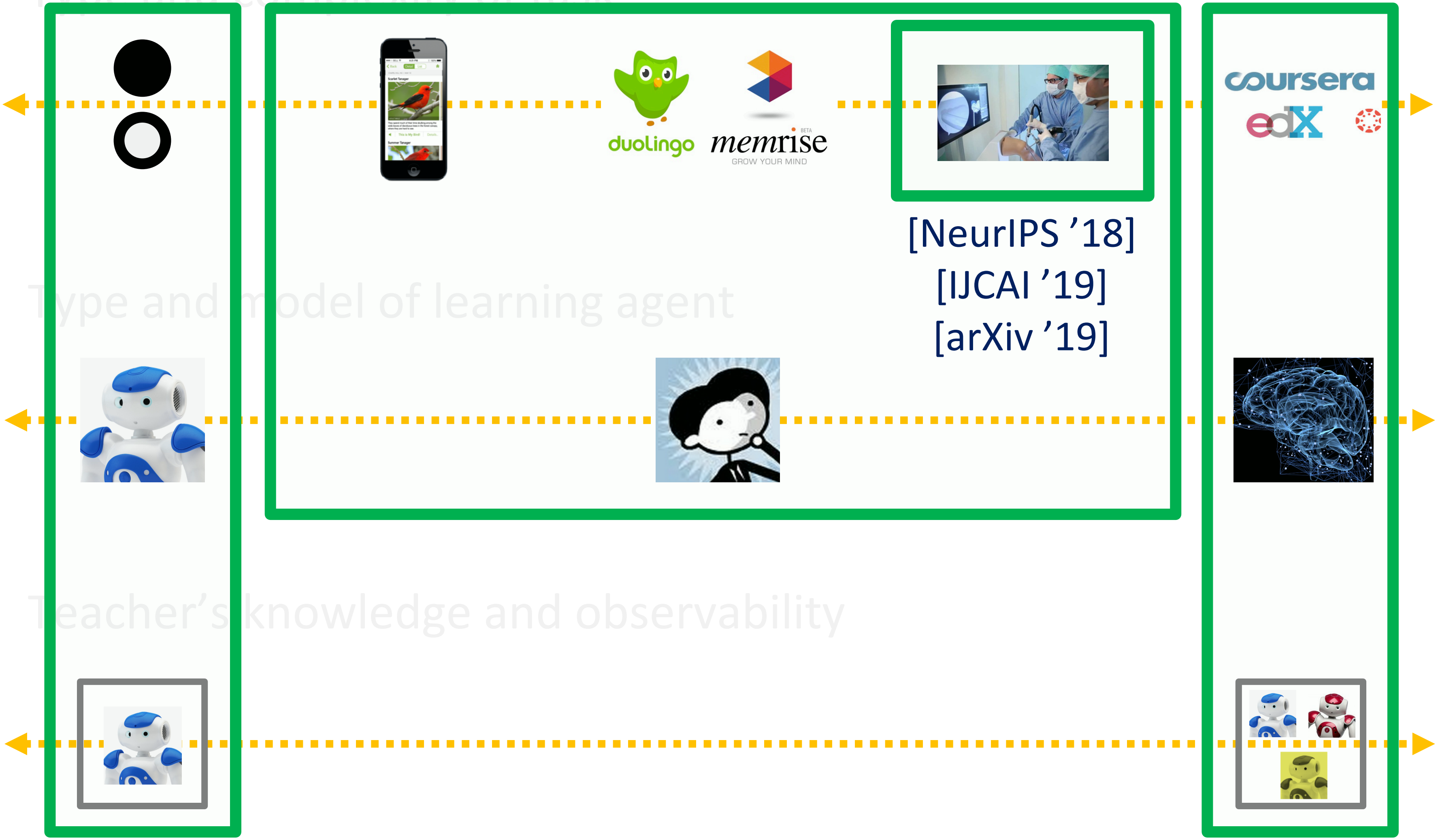
(Survey) An Algorithmic Perspective on Imitation Learning [Osa et al. 2018]

The Role of Teacher: Research Problems

Teaching via demonstrations		Teaching via reward signals
Optimizing curriculum of tasks	[IJCAI '19]	**Optimizing curriculum of tasks**
Optimizing sequence of demonstrations		**Denser rewards (e.g., defining sub-goals)**
Accounting for model mismatch	[NeurIPS '18] [arXiv '19]	**Providing advice (e.g., correcting errors)**

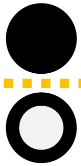
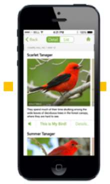
Machine Teaching: Problem Space

- Type and complexity of task
- Type and model of learning agent
- Teacher's knowledge and observability



Machine Teaching: Problem Space

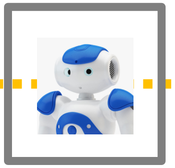
- Type and complexity of task


- Type and model of learning agent



- Teacher’s knowledge and observability



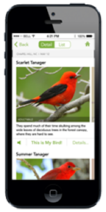

Machine Teaching: Applications

Online education via MOOCs

Skill assessment and practice

Language learning

Educational settings



Biodiversity monitoring

Training simulators

Machine Teaching Group @ MPI-SWS

- Webpage

  https://machineteaching.mpi-sws.org/

- Recent publications

  https://machineteaching.mpi-sws.org/publications.html

- Contact

  adishs@mpi-sws.org

- Slides

  https://machineteaching.mpi-sws.org/files/talks/cmmrs2019-machineteaching-day1.pdf
  https://machineteaching.mpi-sws.org/files/talks/cmmrs2019-machineteaching-day2.pdf
  https://machineteaching.mpi-sws.org/files/talks/cmmrs2019-machineteaching-day3.pdf