A Stochastic Optimal Control Perspective on Affect-Sensitive Teaching

Jacob Whitehill[1,2]
Javier Movellan[1,2]

[1]University of California, San Diego (UCSD)
[2]Machine Perception Technologies (www.mptec.com)

Automated teaching machines

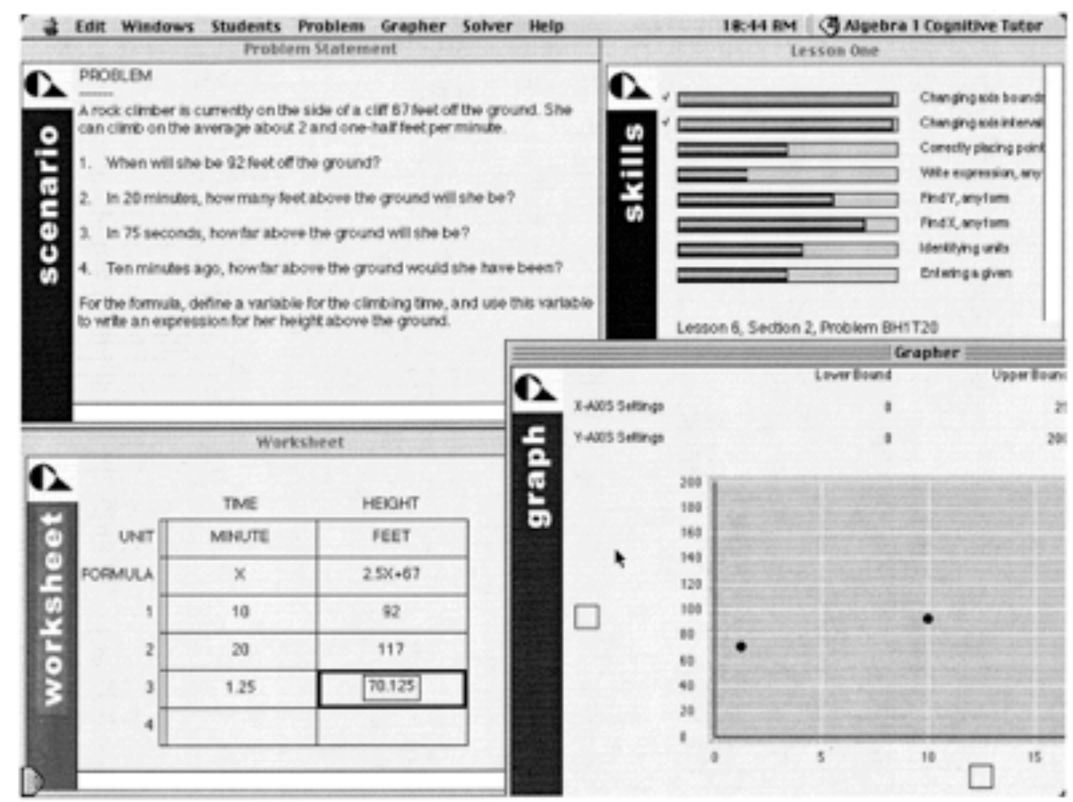
- Automated teaching machines, a.k.a. intelligent tutoring systems (ITS), offer the ability to personalize instruction to the individual student.
- ITS offer some of the benefits of 1-on-1 human tutoring at a fraction of the cost.

History of automated teaching

- Automated teaching has a 50+ year history:
  - 1960s-70s: Stanford researchers (e.g., Atkinson) applied control theory to optimize the learning process for "flashcard"-style vocabulary learning.

History of automated teaching

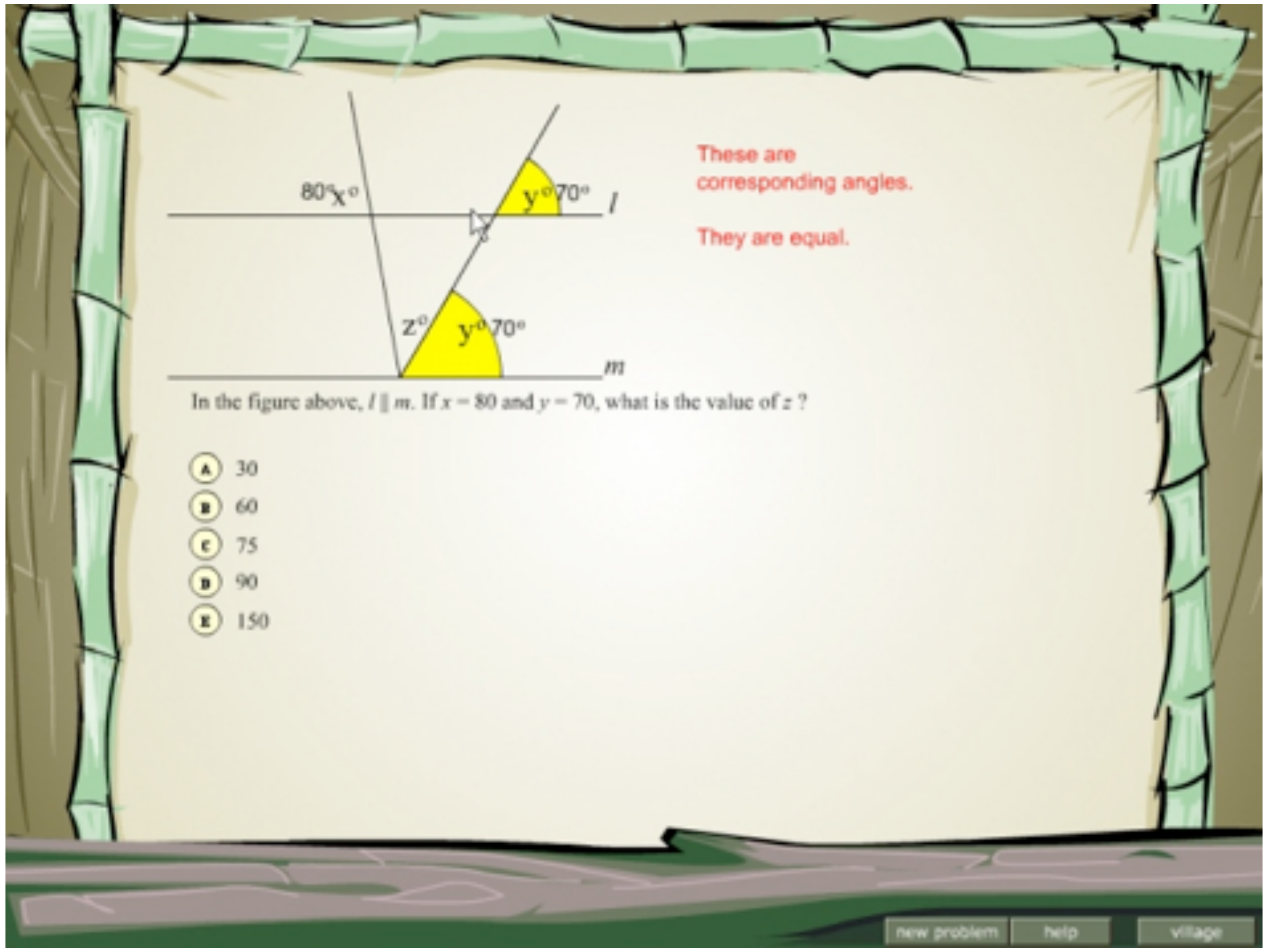
- Automated teaching has a 50+ year history:
  - 1980s-90s: John Anderson at CMU started the "cognitive tutor" movement to teach complex skills, e.g.:
    - Algebra
    - Geometry
    - Computer programming

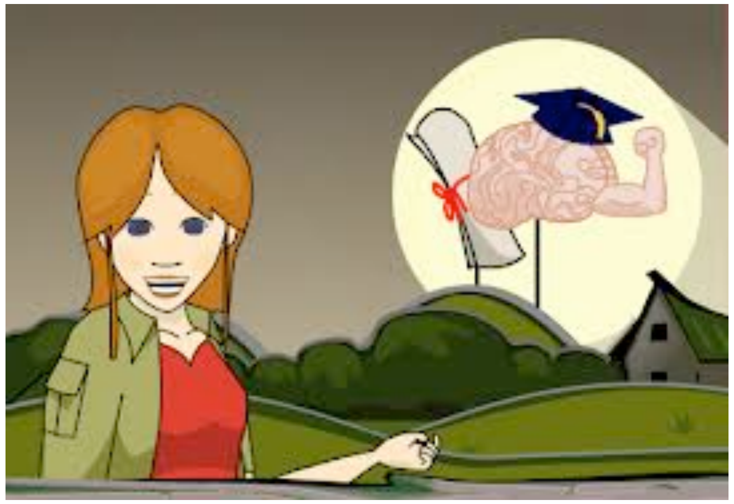


Algebra Cognitive Tutor

History of automated teaching

- Automated teaching has a 50+ year history:
  - 2000s-present: cognitive tutors were enhanced with more sophisticated graphics and sound.
    - Applications of reinforcement learning to ITS.





Wayang Outpost math tutor

Limited sensors

- Over their 50+ year history, one notable feature about ITS is the **limited sensors** they use, usually consisting of:
  - Keyboard
  - Mouse
  - Touch screen

Sensors

- In contrast, human tutors consider the student's:
  - Speech
  - Body posture
  - Facial expression

Sensors

- In contrast, human tutors consider the student's:
  - Speech
  - Body posture
  - Facial expression
- It is possible that automated tutors could become more effective if they used richer sensory information.

Affect-sensitive automated teachers

- A hot topic in the ITS community is **affect-sensitive** automated teaching systems.
- "Affect-sensitive": use rich sensors to sense and respond to the student's affective state.
- "Affective state":
  - Student's motivation, engagement, frustration, confusion, boredom, etc.

Affect-sensitive automated teachers

- Developing an affect-sensitive ITS can be divided into 2 computational problems:
  - **Perception**: how to recognize affective states automatically using affective sensors.
    - E.g., how to map image pixels from a webcam into a estimate of the student's engagement.

Affect-sensitive automated teachers

- Developing an affect-sensitive ITS can be divided into 2 computational problems:
  - **Perception**: how to recognize affective states automatically using affective sensors.
    - E.g., how to map image pixels from a webcam into a estimate of the student's engagement.
  - **Control**: how to use affective state estimates to teach more effectively.

Perception problem

- Tremendous progress has been made in machine learning & vision during last 15 years.
  - Real-time automatic face detectors are commonplace.
  - Facial expression recognition is starting to become practical.

Control problem

- Much less research has addressed how students' affective state estimates should influence the teacher's decisions.

Control problem

- Much less research has addressed how students' affective state estimates should influence the teacher's decisions.
- Thus far, the approaches have been **rule-based**:
    - *If* student looks frustrated, *then*:
      Say: "That was frustrating. Let's move to something easier."

      (Wayang Outpost Tutor -- Woolf, et al. 2009)

Control problem

- So far there is little empirical evidence that affect-sensitivity is beneficial.
- Comparison of affect-sensitive to affect-blind computer literacy tutor (“AutoTutor”):

	Learning gains	
	Aff.-Sens.	Aff.-Blind
Day 1	0.249	0.389
Day 2	0.407	0.377

Affect-sensitive tutor was *less effective* on day 1.

D’Mello, et al. 2010

Control problem

- Even if rules can be devised for a few scenarios, it is unlikely that this approach will scale up:
  - Multiple sensors, high bandwidth, varying timescales, etc.

Control problem

- Even if rules can be devised for a few scenarios, it is unlikely that this approach will scale up:
  - Multiple sensors, high bandwidth, varying timescales, etc.
- Instead, a formal computational framework for decision-making may be useful.

Stochastic optimal control

- Stochastic optimal control (SOC) theory may provide such a framework.
- SOC provides:
  - Mathematics to define teaching as an optimization problem.
  - Computational tools to solve the optimization problem.

Stochastic optimal control

- SOC has well-known computational difficulties:
  - Finding exact solutions to SOC problems is usually intractable.
  - More research is needed on how to find *approximately* optimal control policies for automated teaching problems.
  - Since the 1960s, a variety of machine learning and reinforcement learning methods have been developed for finding approximately optimal solutions.

SOC-based ITS

- In this talk, I will describe one approach to building an ITS for language acquisition using approximate methods from SOC.
  - Our work draws inspiration from Rafferty, Brunskill, Griffiths, and Shafto (2011).
- I also describe how an SOC-based automated teacher naturally uses affective observations when they are available.
  - No ad-hoc rules are necessary.

Teaching word meanings from visual examples

ontbyt

Teaching word meanings from visual examples

ontbyt

Teaching word meanings from visual examples

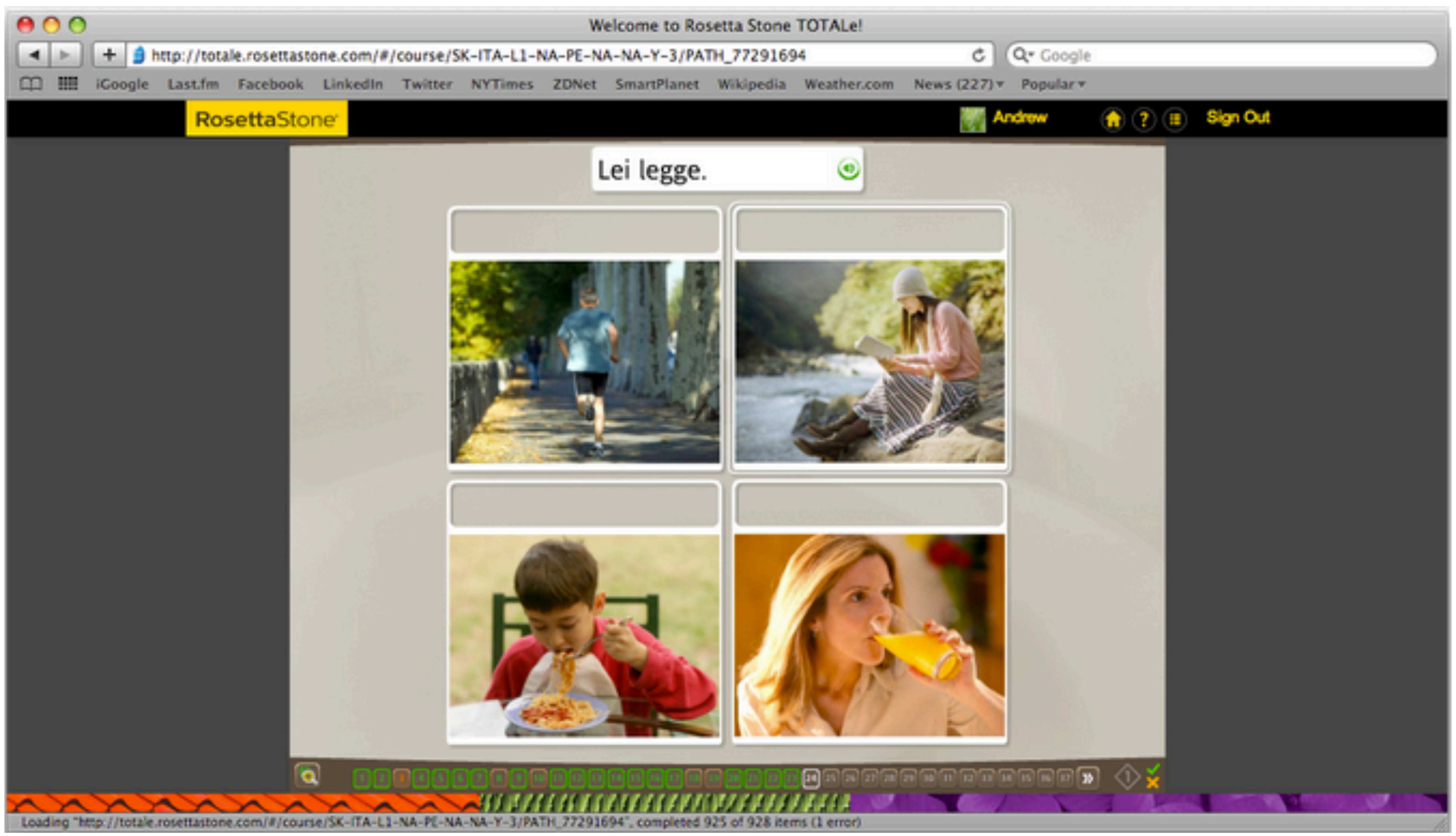
ontbyt

Teaching word meanings from visual examples

ontbyt (breakfast)

Teaching word meanings from visual examples

- This is the learning approach used in Rosetta Stone language software.



Teaching task

- We wish to teach the meanings of a set of words.
- Each word can mean any one of a set of concepts.
- We have a set of example images.
- At each timestep *t*, the automated teacher can:
  - *Teach* word *j* using image *k*
  - *Ask* student a question about word *j*
  - Give the student a *test* on all the words in the set
- Teacher's goal: help student pass the test as quickly as possible.

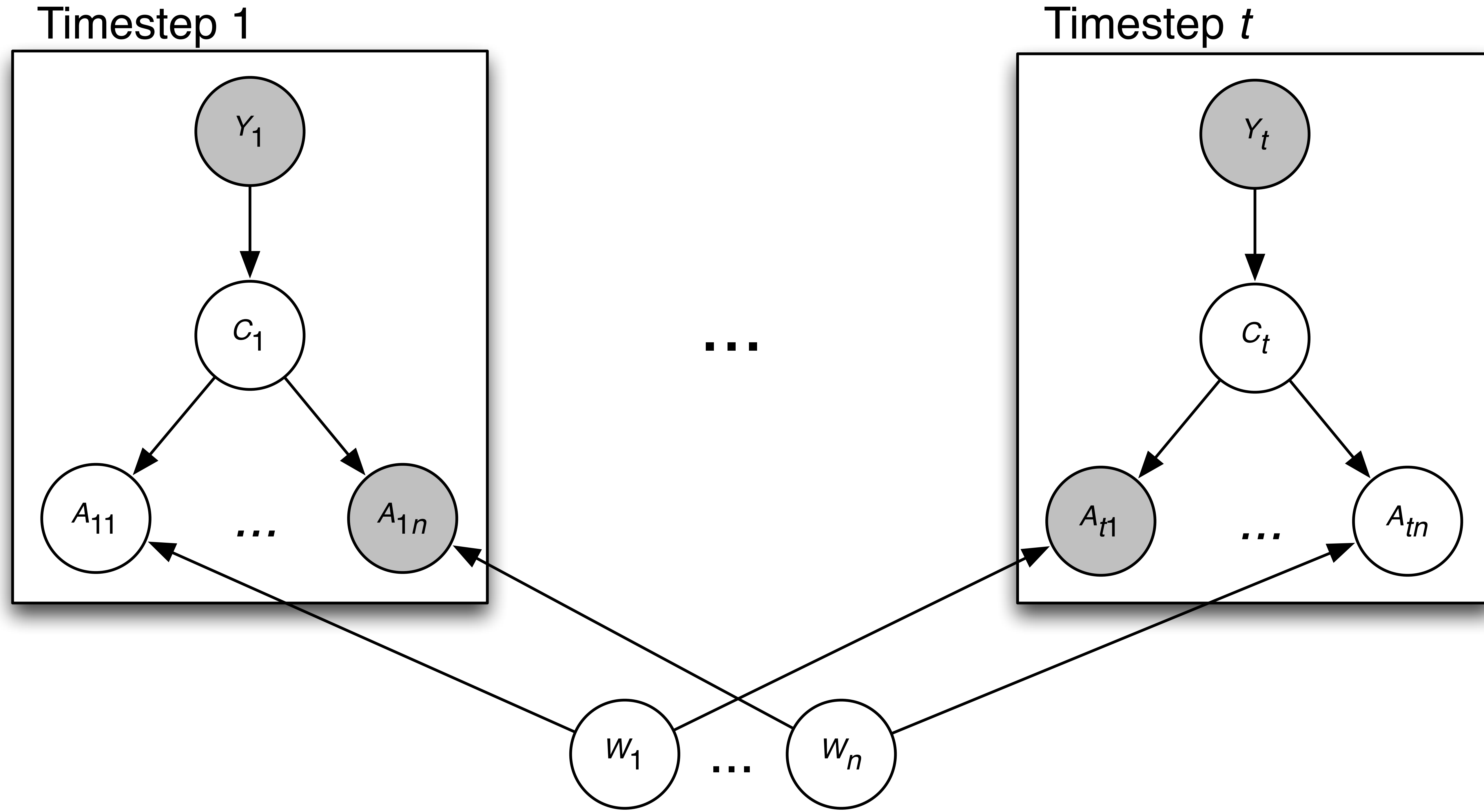
Teaching task as SOC problem

- We pose this teaching task as a SOC problem.
- We use **model-based control**:
  - We develop probabilistic models of how the student *learns*, and how she *responds* to questions asked by the teacher.
  - We collect data of human students to estimate model parameters.
  - Once model is learned, we can optimize the automated teacher using simulation.

Student model

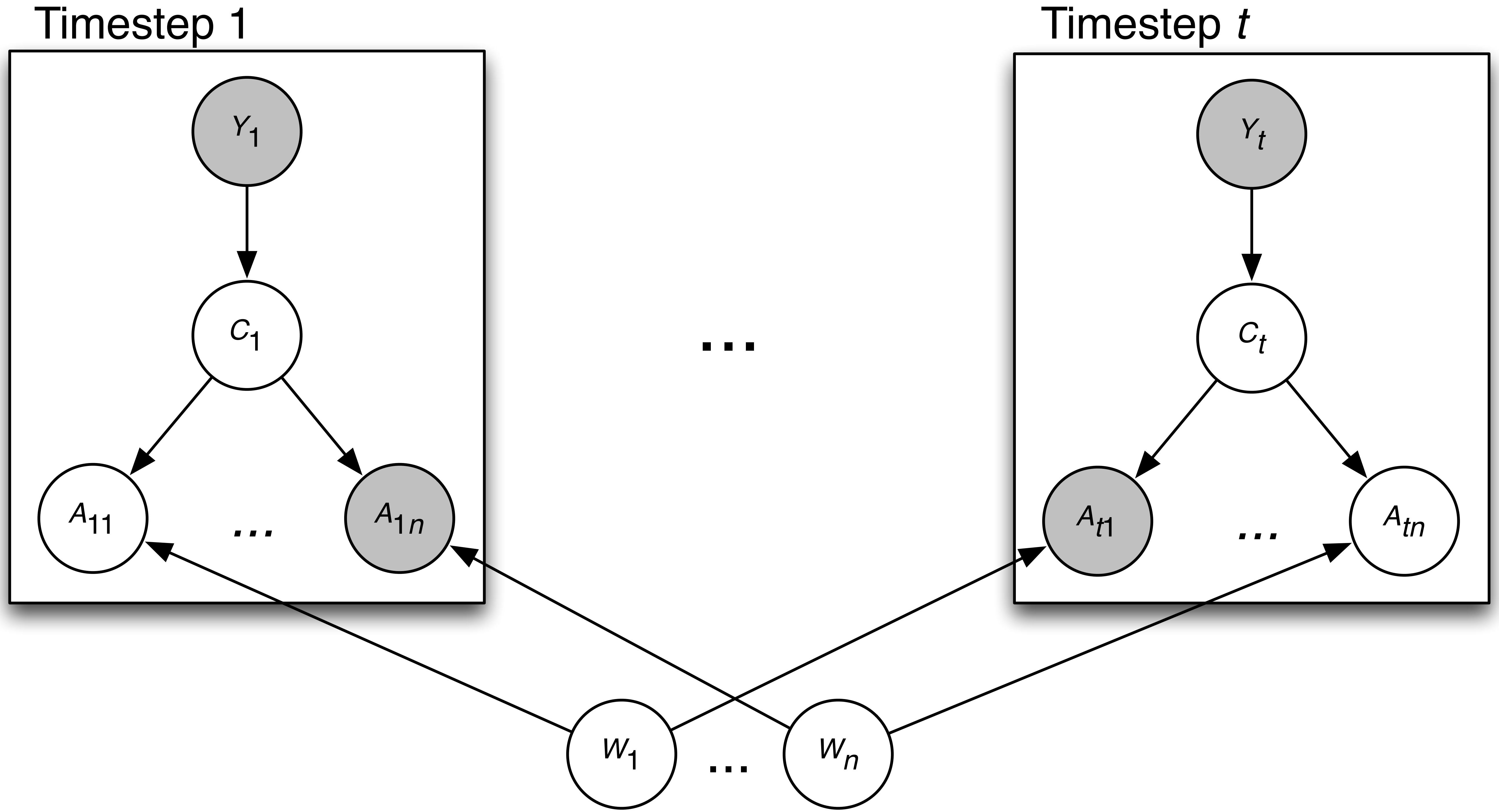
- We model the student as a **Bayesian learner**, in the manner of Nelson, Tenenbaum and Movellan (2007) for concept learning and Rafferty, et al. (2011) for concept teaching.
  - Reduces amount of data needed to fit the model.

Student model



Student has a belief $P(c \mid y)$ about what concept the teacher was trying to convey with the image.

Student model



After *t* timesteps the student updates her belief:

$$m_{tj} \doteq P(w_j | \; y_{1:t}, a_{1q_1}, \ldots, a_{tq_t})$$

Student inference

- Since a perfectly Bayesian learner is unrealistic (Nelson and Cottrell 2007), we "soften" the model by introducing a "belief update strength" variable $\beta_t \in (0,1]$:
- β_t specifies how much the student updates her belief at time t.
- β_t may be related to the student's level of "engagement" in the learning task.

Student responses

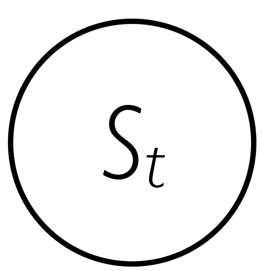
- For *ask* and *test* actions:
  - If student is asked to define the meaning of word *j*, she responds using probability matching according to m_{tj}.
  - Probability matching is a popular response model in psychology (e.g., Movellan and McClelland, 2000).

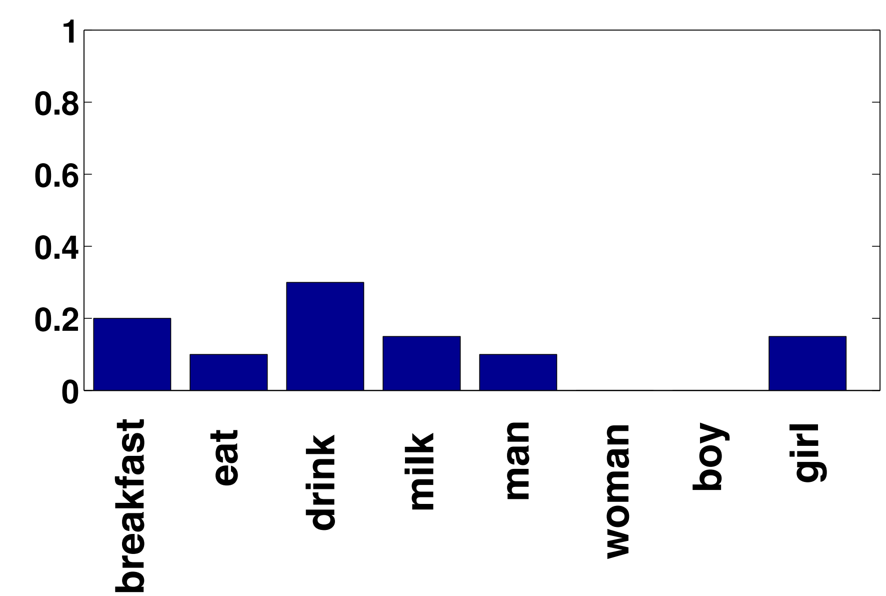
Teacher model

- Let us now consider the problem from the *automated teacher*'s perspective...

Problem formulation using SOC

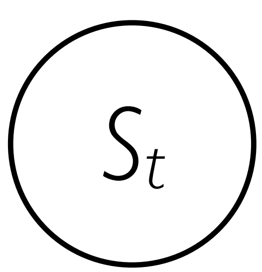
- **State** S_t:
  - Student's knowledge m_t of the words' meanings as well as the belief update strength β_t.

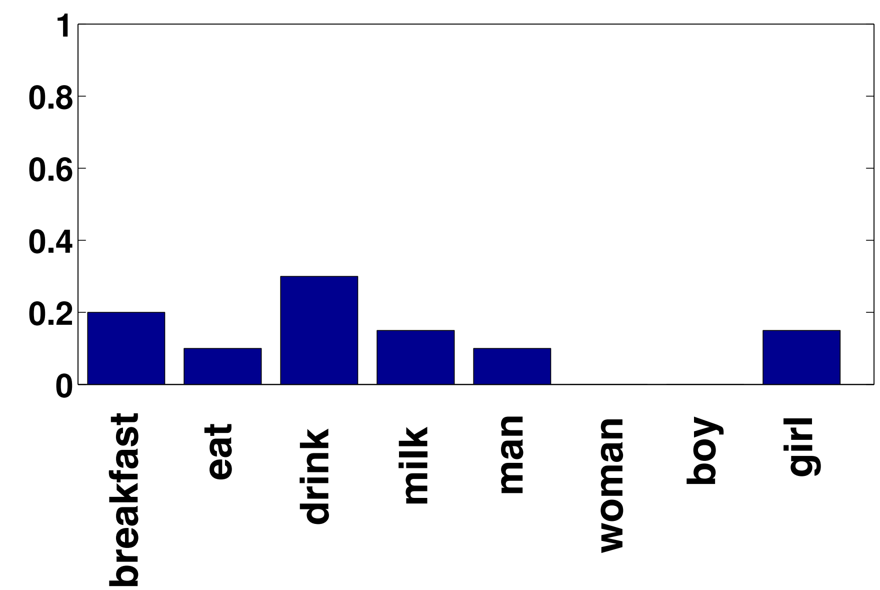




Problem formulation using SOC

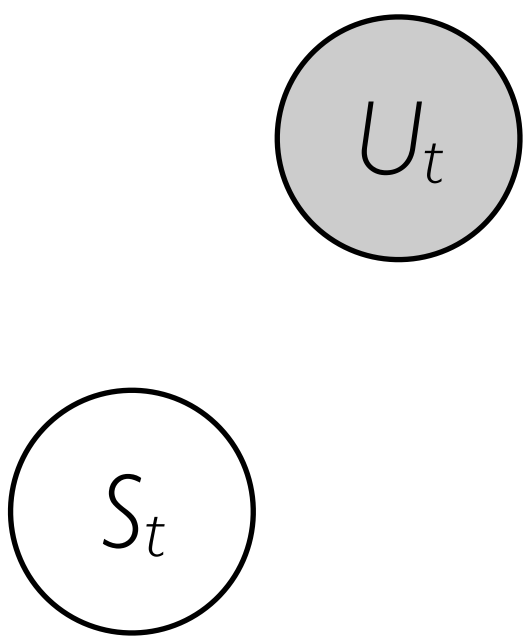
- **State** S_t:
  - The state is assumed to be “hidden” from the teacher because the state is inside the student’s brain.





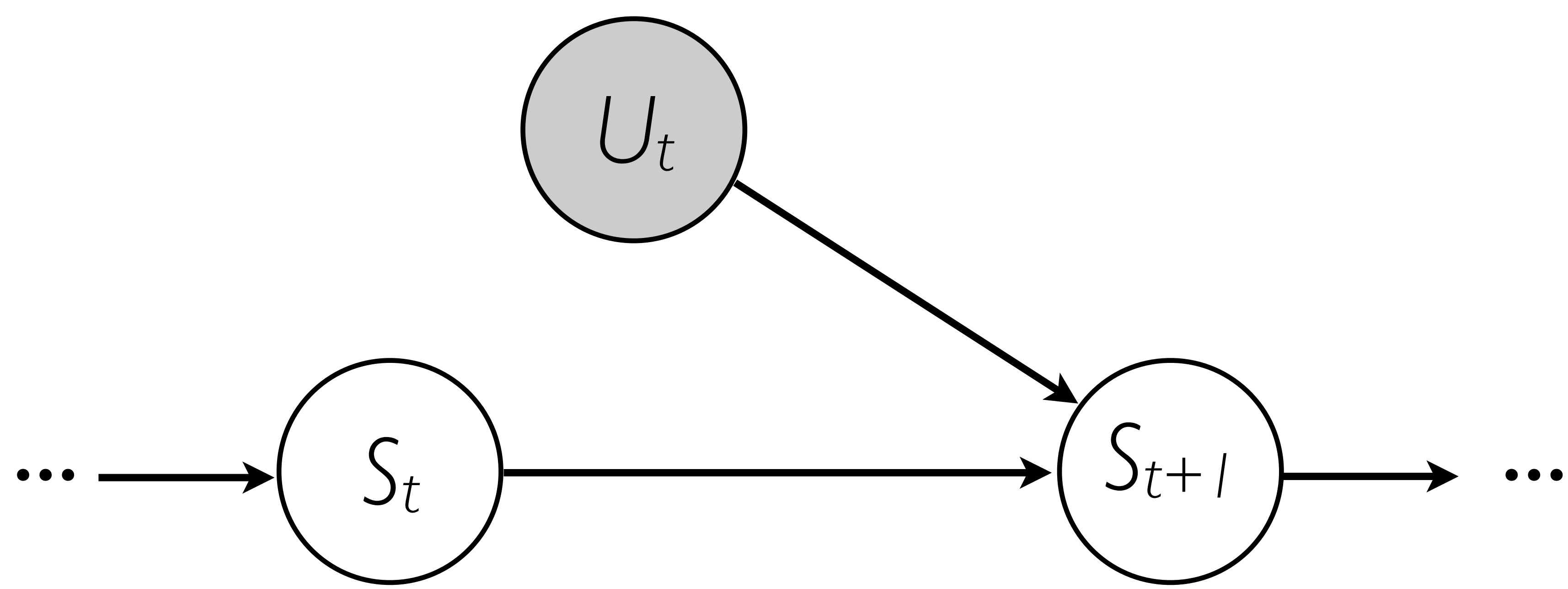
Problem formulation using SOC

- **Action** U_t:
  - Teach word *j* with image *k*
  - Ask word *j*
  - Test



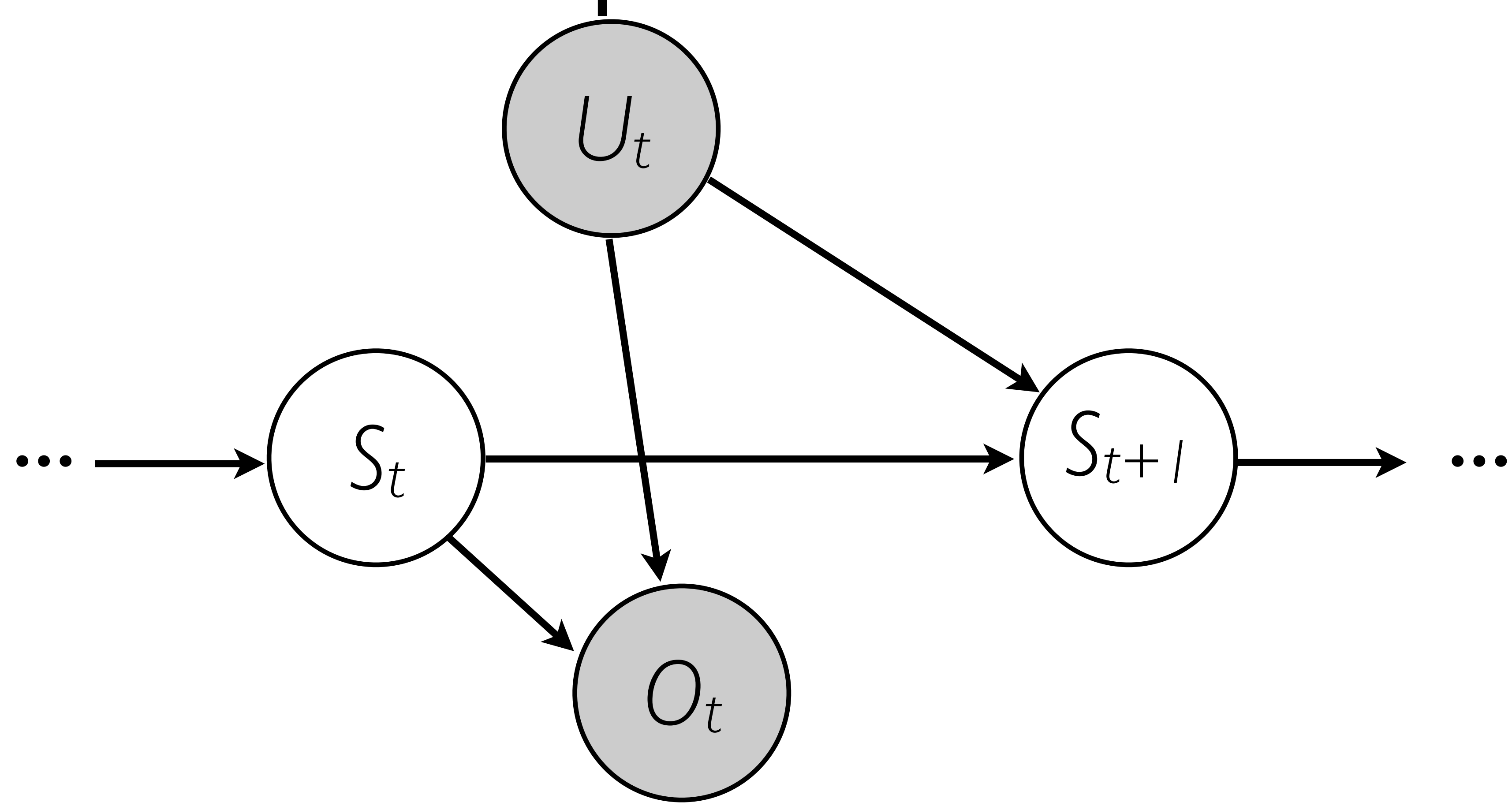
Problem formulation using SOC

- **Action** U_t:
  - U_t and S_t jointly determine the student's next state S_{t+1} according to the *transition dynamics* given by the student learning model.



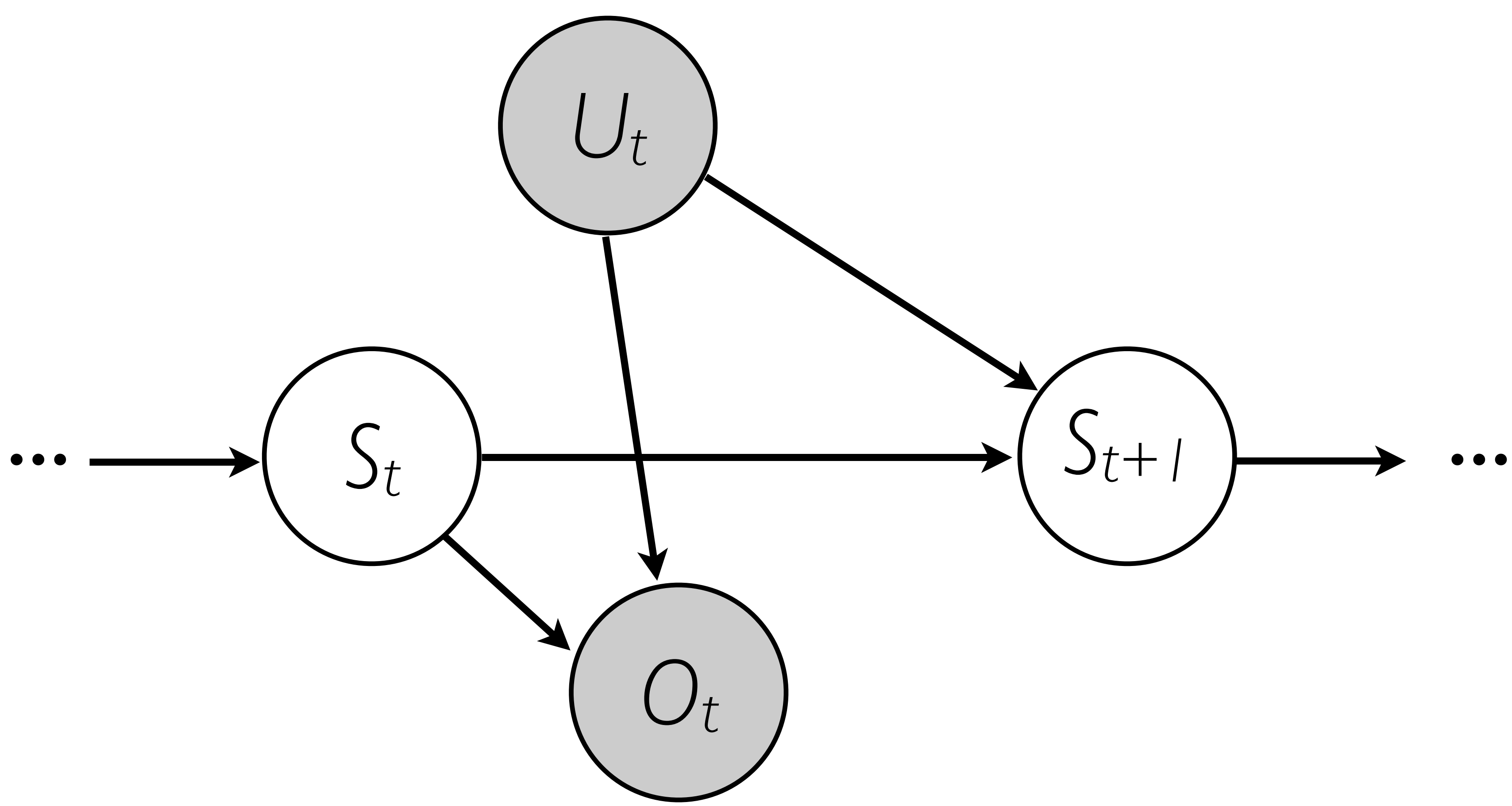
Problem formulation using SOC

- **Observation** O_t:
  - When the teacher asks a question, it receives a response ("observation") from the student.
  - O_t is determined by S_t and U_t according to the student response model.



Problem formulation using SOC

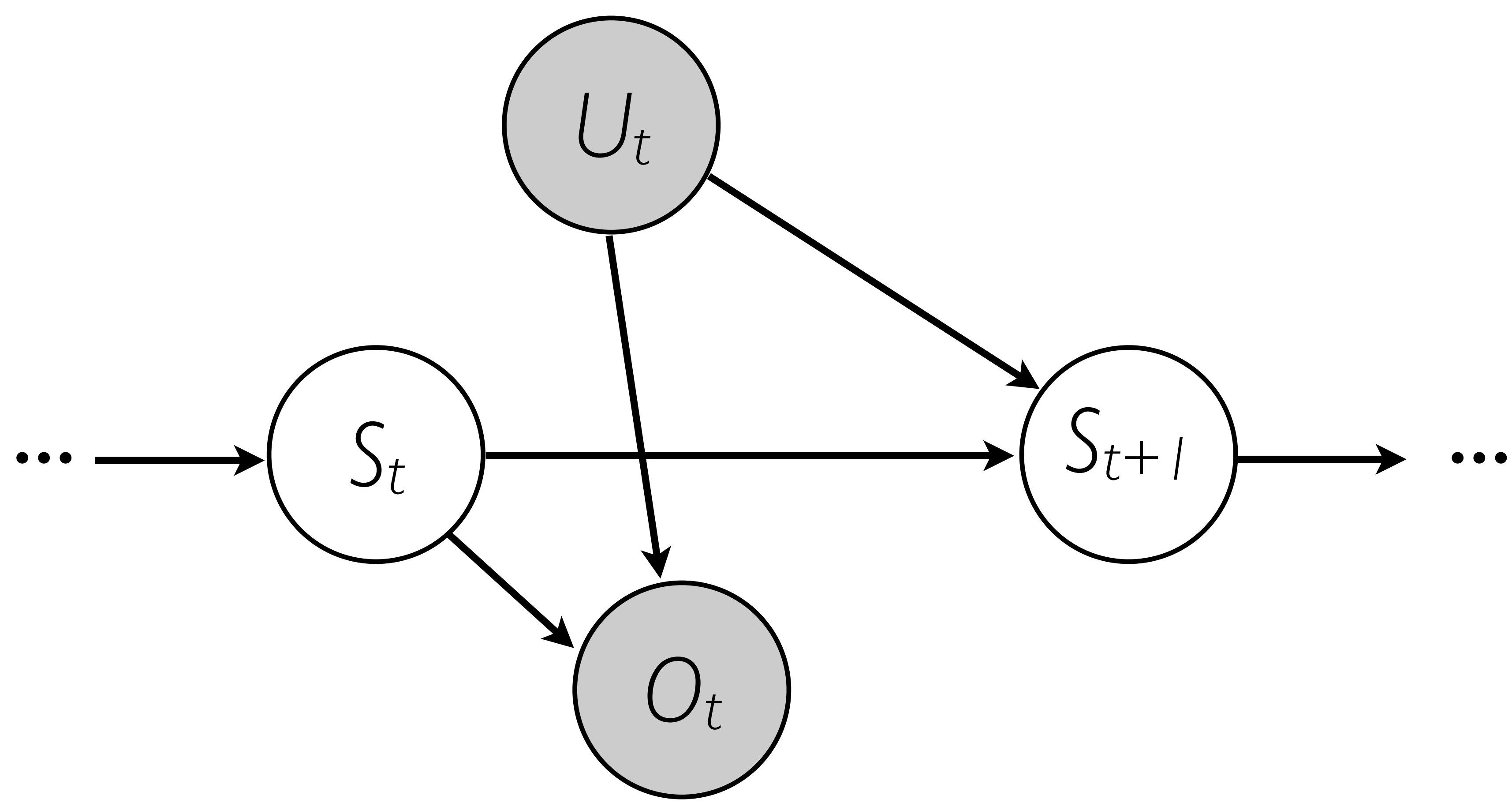
- **Belief** B_t:
  - The teacher maintains a belief $b_t \doteq P(s_t \mid o_{1:t-1}, u_{1:t-1})$ over the student's state given the history of actions and observations up to time *t*.



Problem formulation using SOC

- **Belief** *B_t*: update from time *t* to time *t*+1:

$$P(s_{t+1} \mid o_{1:t}, u_{1:t})$$
$$\propto \int P(s_{t+1} \mid s_t, u_t) P(o_t \mid s_t, u_t) P(s_t \mid o_{1:t-1}, u_{1:t-1}) ds_t$$

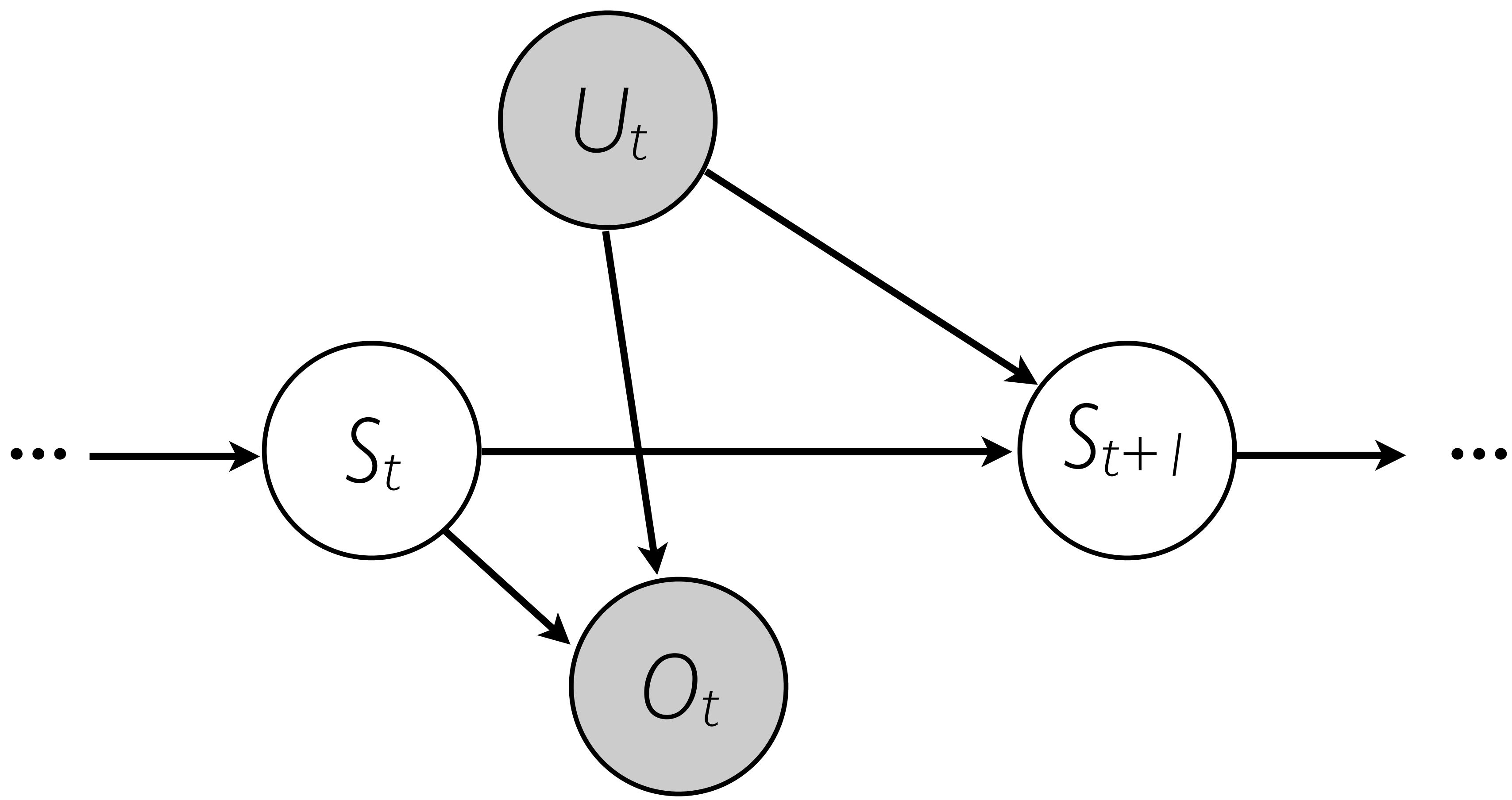


Problem formulation using SOC

- **Belief** B_t: update from time t to time t+1:

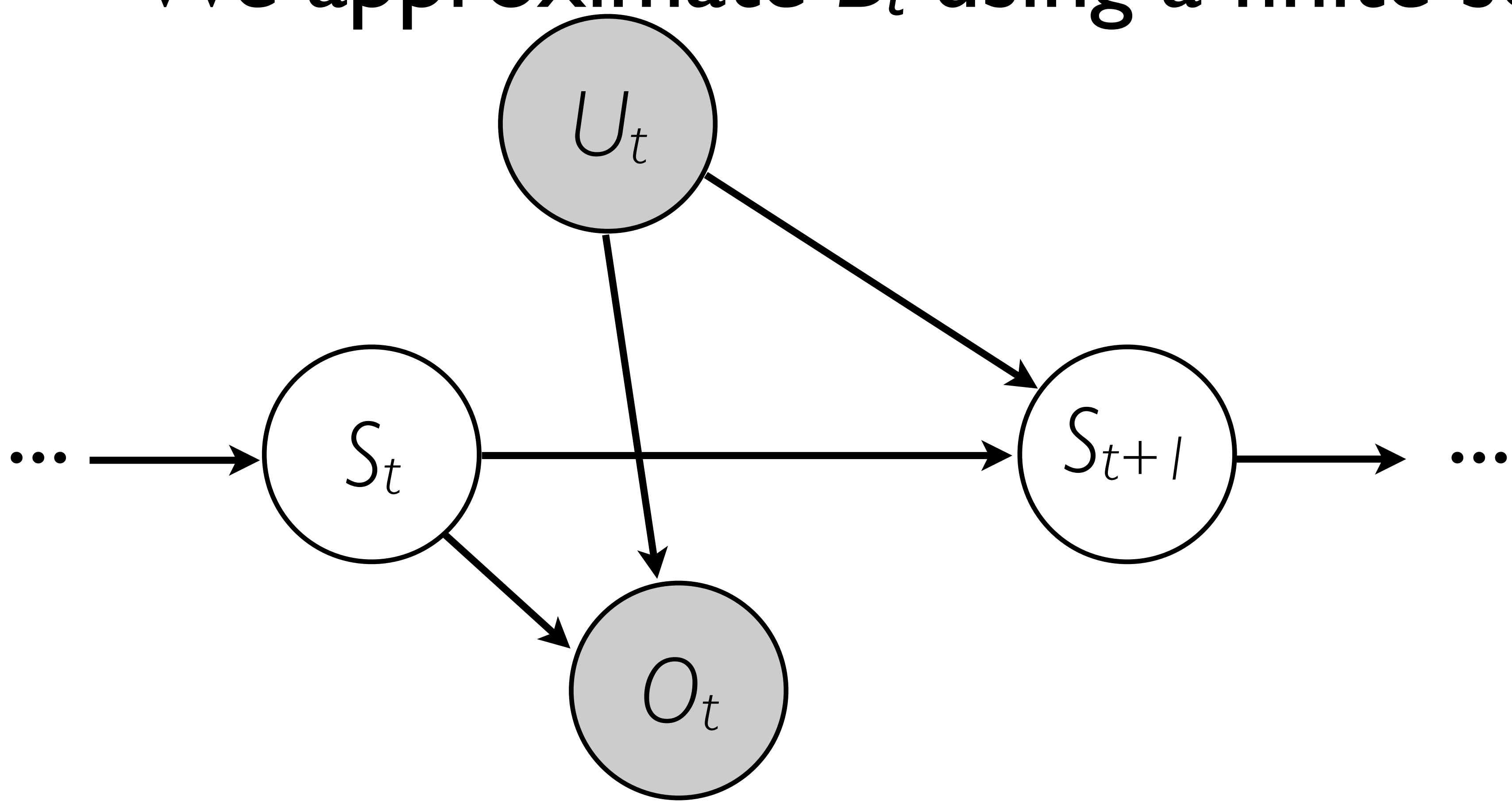
$$P(s_{t+1} \mid o_{1:t}, u_{1:t})$$ Posterior belief

$$\propto \int \underbrace{P(s_{t+1} \mid s_t, u_t)}_{\text{Student learning dynamics}} \underbrace{P(o_t \mid s_t, u_t)}_{\text{Student response likelihood}} \underbrace{P(s_t \mid o_{1:t-1}, u_{1:t-1})}_{\text{Prior belief}} ds_t$$



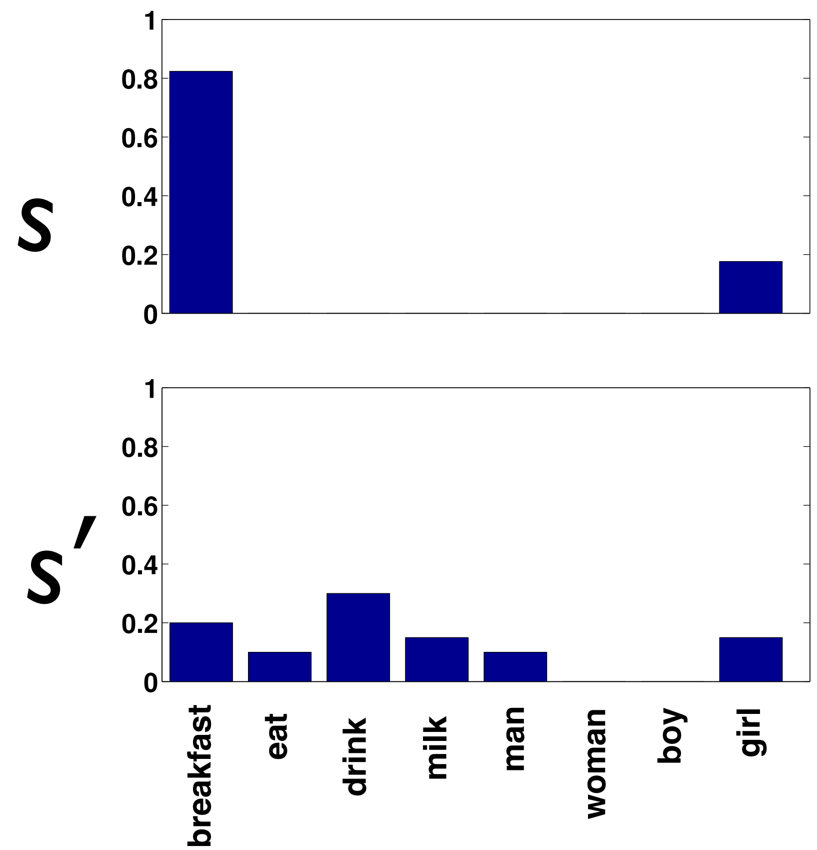
Problem formulation using SOC

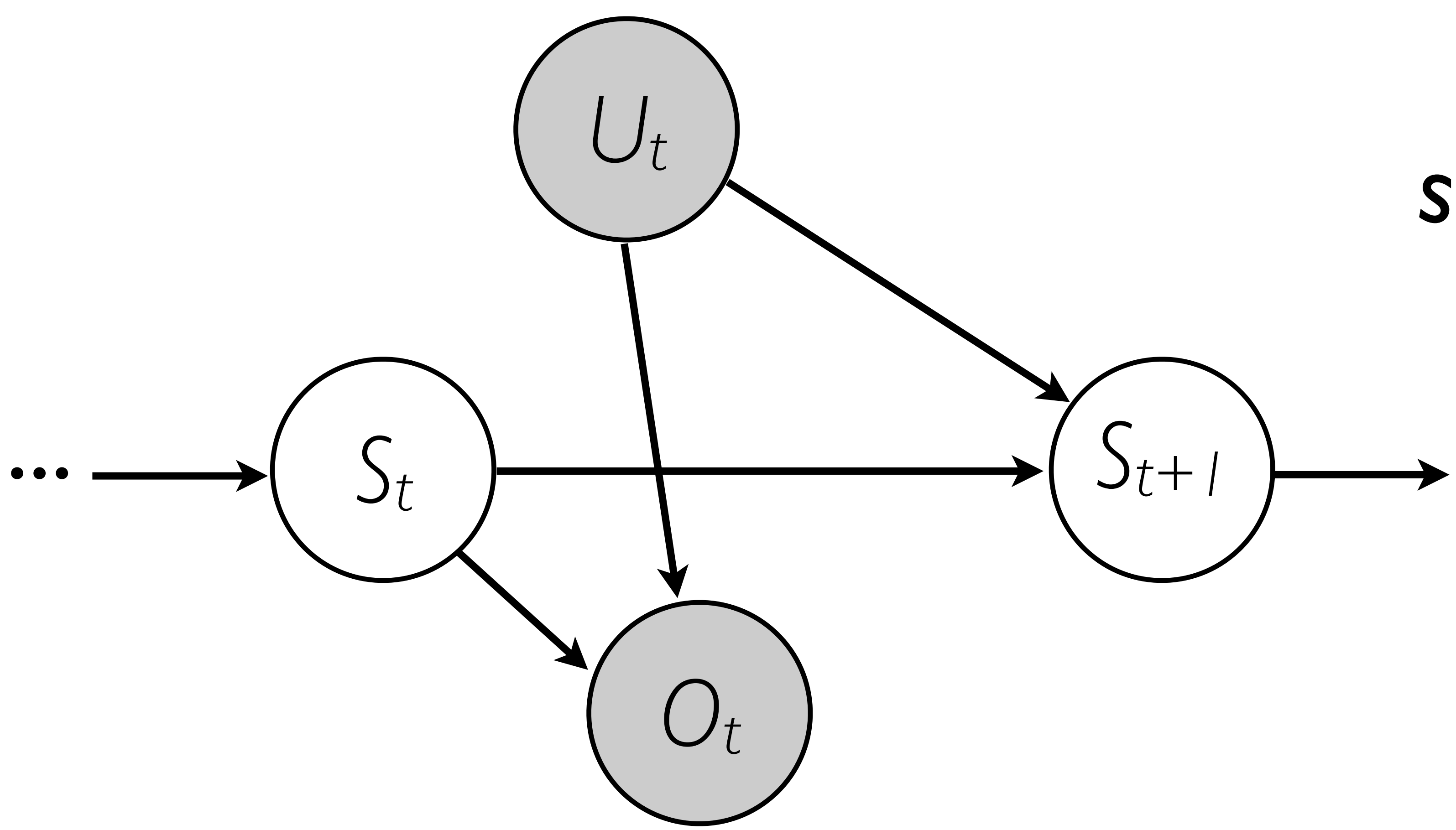
- **Belief** B_t:
  - Since S_t itself is a probability distribution, B_t is a probability distribution over probability distributions.
  - We approximate B_t using a finite set of particles.



Problem formulation using SOC

- **Reward function** *r(s,u)*:
    - Teacher may prefer certain states, or certain state+action combinations, over others.





Problem formulation using SOC

- **Control policy** π:
    - The teacher chooses its action at time t according to the control policy π.
    - π maps the teacher's belief b_t about what the student knows, into an action u_t.

Problem formulation using SOC

- **Control policy** π:
  - Different policies are better than others, as expressed by their *value V*:

$$V(\pi) \doteq E\left[\sum_{t=1}^{\tau} r(S_t, U_t) \mid \pi\right]$$

where τ is the length of the teaching session, measured in # of teacher's actions.

Problem formulation using SOC

- **Control policy** π:
  - Different policies are better than others, as expressed by their *value V*:

$$V(\pi) \doteq E\left[\sum_{t=1}^{\tau} r(S_t, U_t) \mid \pi\right]$$

  - An *optimal policy* π* is a policy that maximizes *V*:

$$\pi^* \doteq \arg\max_{\pi} V(\pi)$$

Computing policies

- Finding π^* exactly is intractable.
- Instead, we find an *approximately* optimal policy using *policy gradient* to maximize $V(\pi)$ in simulation using the student model.

Experiment

- We created a vocabulary of 10 words from an artificial language:

Word	**Meaning**
duzetuzi	man
fota	woman
nokidono	boy
mininami	girl
pipesu	dog
mekizo	cat
xisaxepe	bird
botazi	rabbit
koto	eat
notesabi	drink

Experiment

- We collected a set of images from Google Image Search:

Experiment

- To estimate student model parameters as well as time costs of each action (*teach*, *ask*, *test*), we collected data from human subjects.
- Given the student model and time costs, we used policy gradient to compute π so as to minimize the expected time the student needs to pass the test.
  - This control policy constitutes the "SOCTeacher".

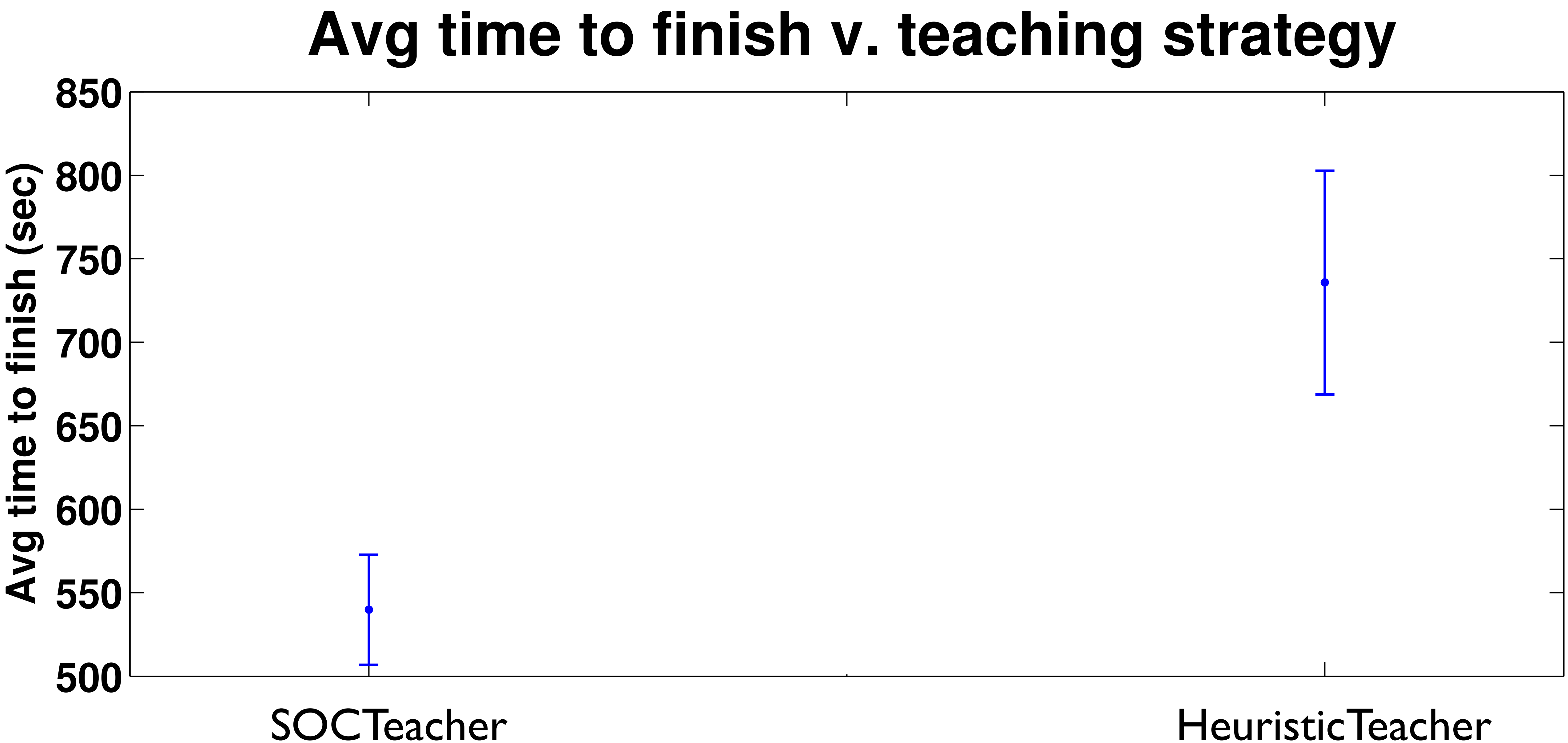
Experiment

- We conducted an experiment on 90 subjects from the Amazon Mechanical Turk.
- Dependent variable: time to pass the test.

Experimental conditions

1. SOCTeacher
2. HeuristicTeacher
   - Select a word randomly at each round, and teach it using an image sampled according to P(c | y).
   - Test every p rounds (p was optimized in simulation).

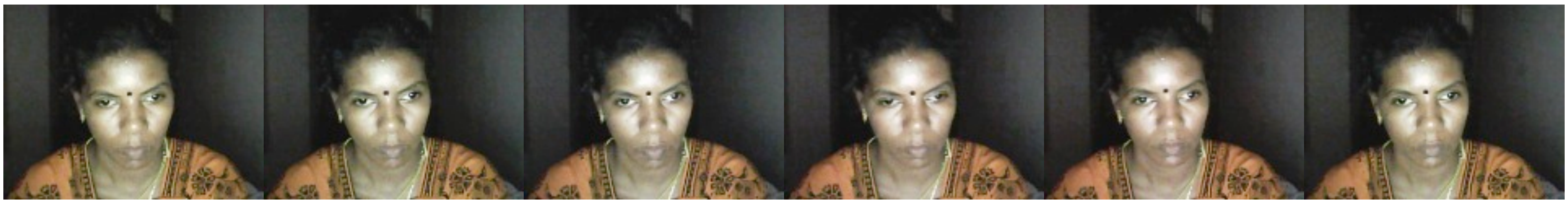
Results

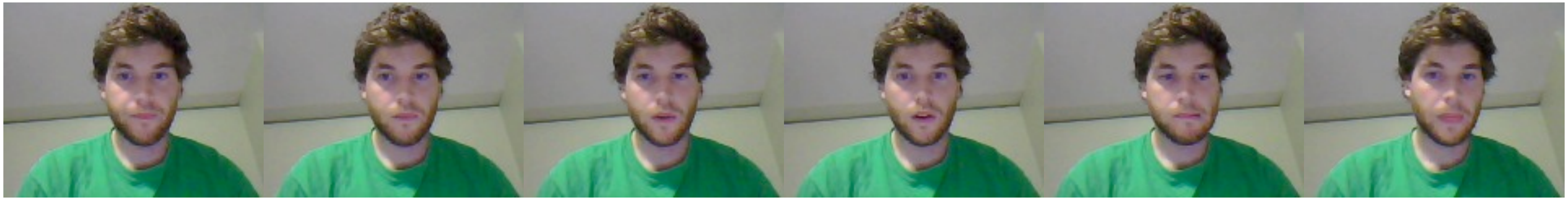


TimeCost(SOCTeacher) is 24% less than TimeCost(HeuristicTeacher) ($p < 0.01$).

Affect while learning

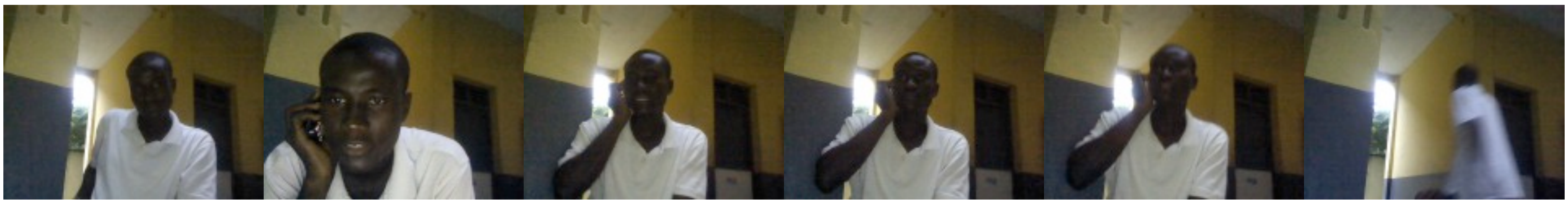
- In pilot exploration of students' affect, we found that students were usually engaged in the task.

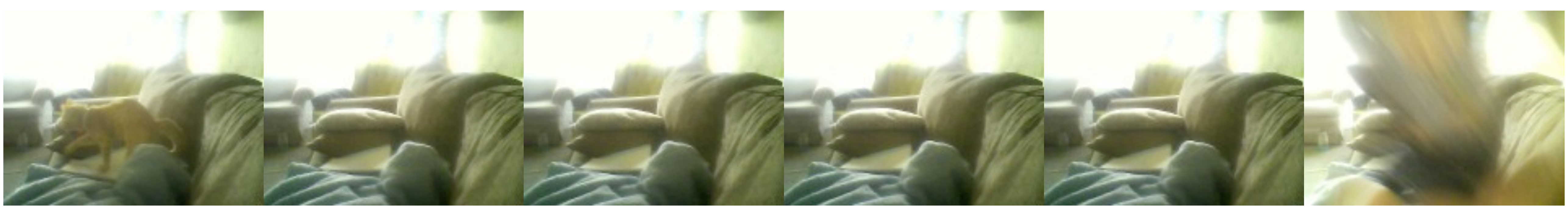




Affect while learning

- There were, however, occasional moments of non-engagement.





How affect could be used

- Suppose that the student's face image z_t is correlated with the student's belief update strength β_t according to $P(z_t \mid \beta_t)$:
- How can this "affective sensor" measurement be used to teach better?

How affect could be used

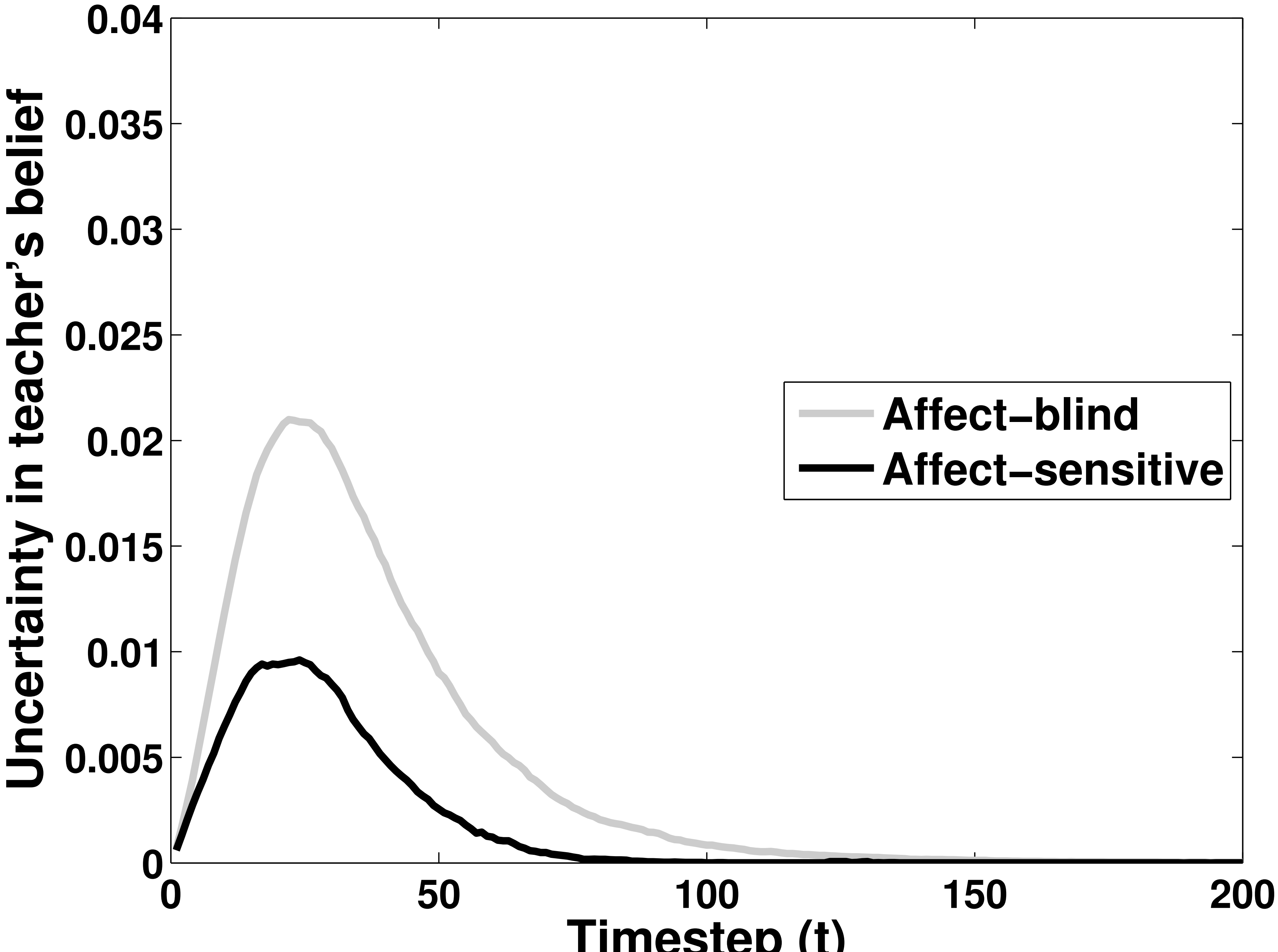
- In an SOC-based automated teacher, the teacher's belief update simply gains an additional term:

$$P(s_{t+1} \mid o_{1:t}, u_{1:t}) \\ \propto \int P(s_{t+1} \mid s_t, u_t) P(o_t \mid s_t, u_t) \underbrace{P(z_t \mid \beta_t)}_{\text{Affective observation}} P(s_t \mid o_{1:t-1}, u_{1:t-1}) ds_t$$

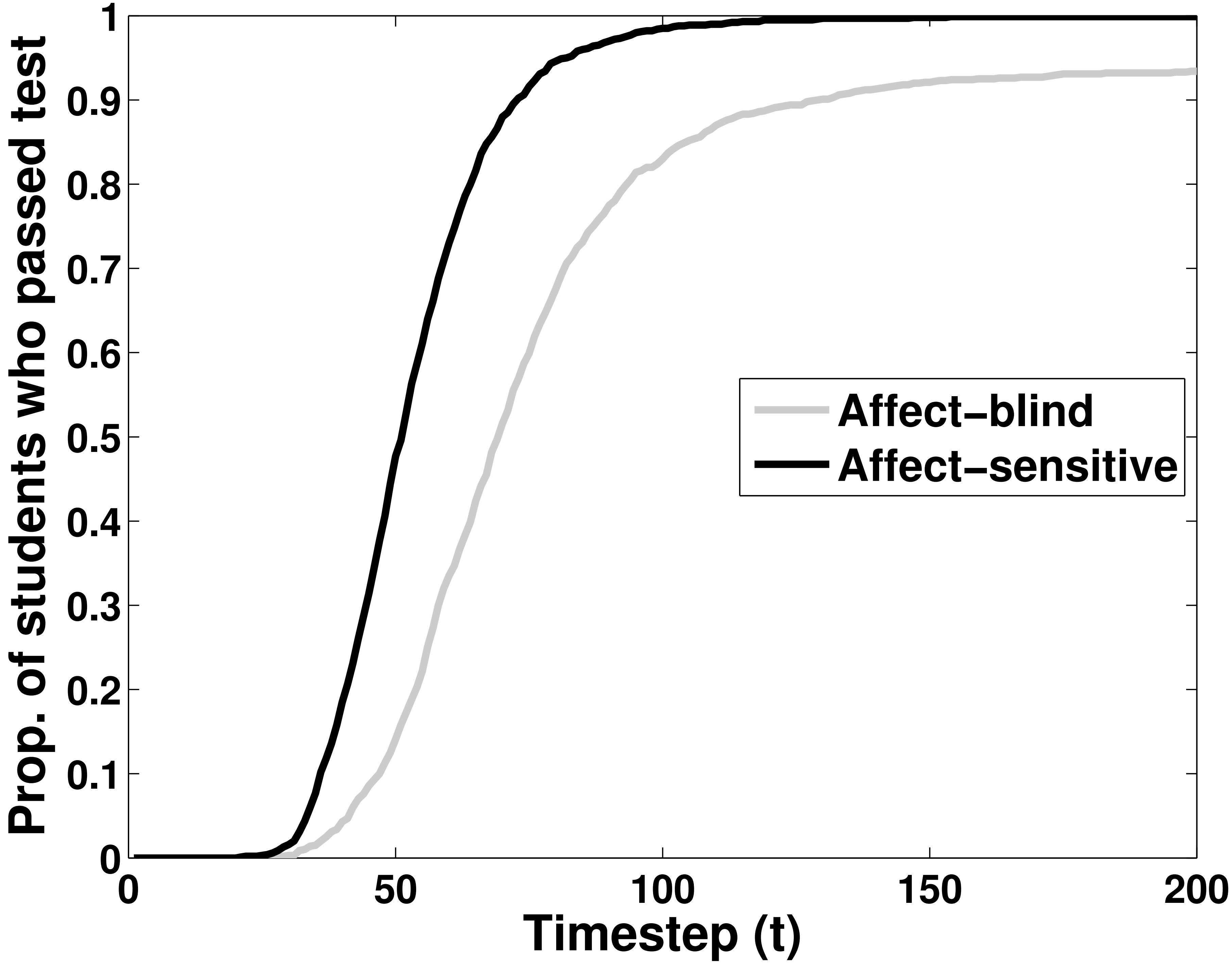
- The "affective observation" greatly constrains the teacher's belief of the student's knowledge.

- Amended belief update emerges naturally from probability theory -- no need for ad-hoc rules.

Incorporating affect: simulation



Incorporating affect: simulation



Summary

- While stochastic optimal control brings with it significant computational challenges, approximate solution methods can be used to create practical ITS.
- SOC provides a principled method of incorporating affective sensor readings into the teaching process.

Thank you