“The Tools Challenge: Rapid trail-and-error learning in physical problem solving[1]”

by

Kelsey R. Allen, Kevin A. Smith, & Joshua B. Tenenbaum,

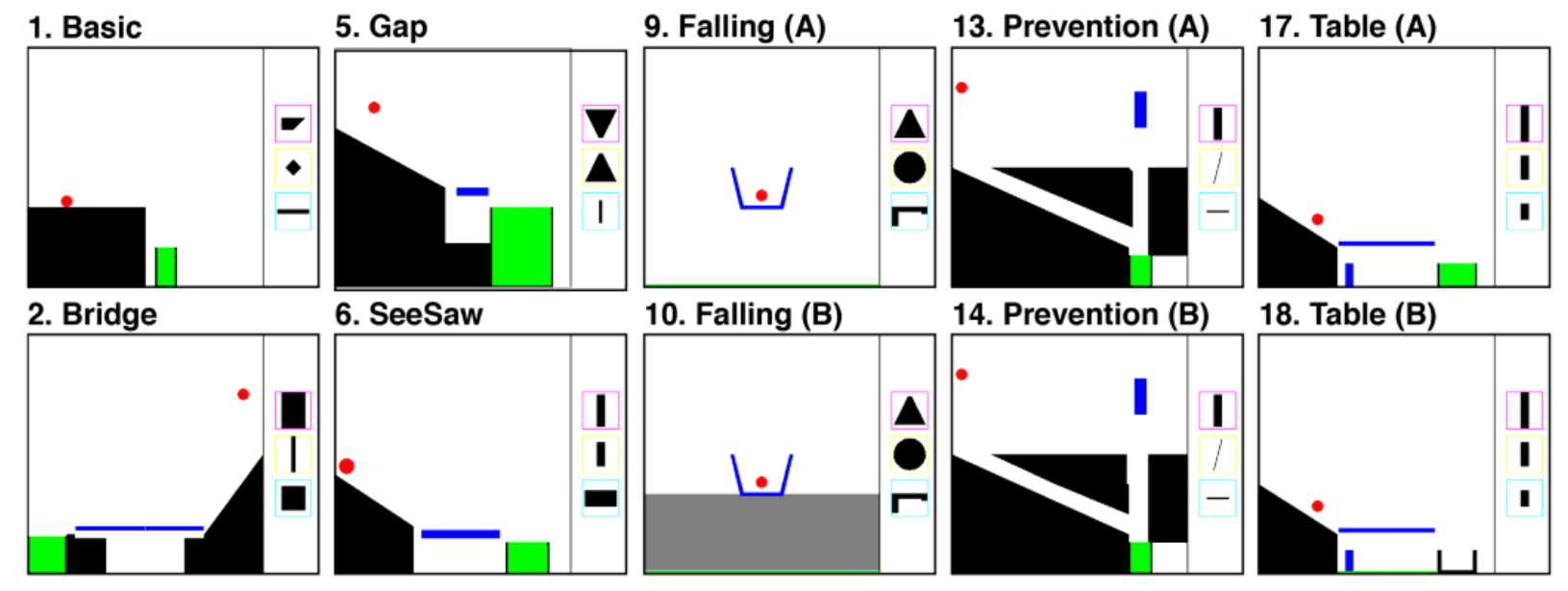
arXiv e-prints, 2019, arXiv:1907.09620

Presented by:
Gaurav Kumar

Index

- Task
- Architecture
- Results
- Discussion

Task

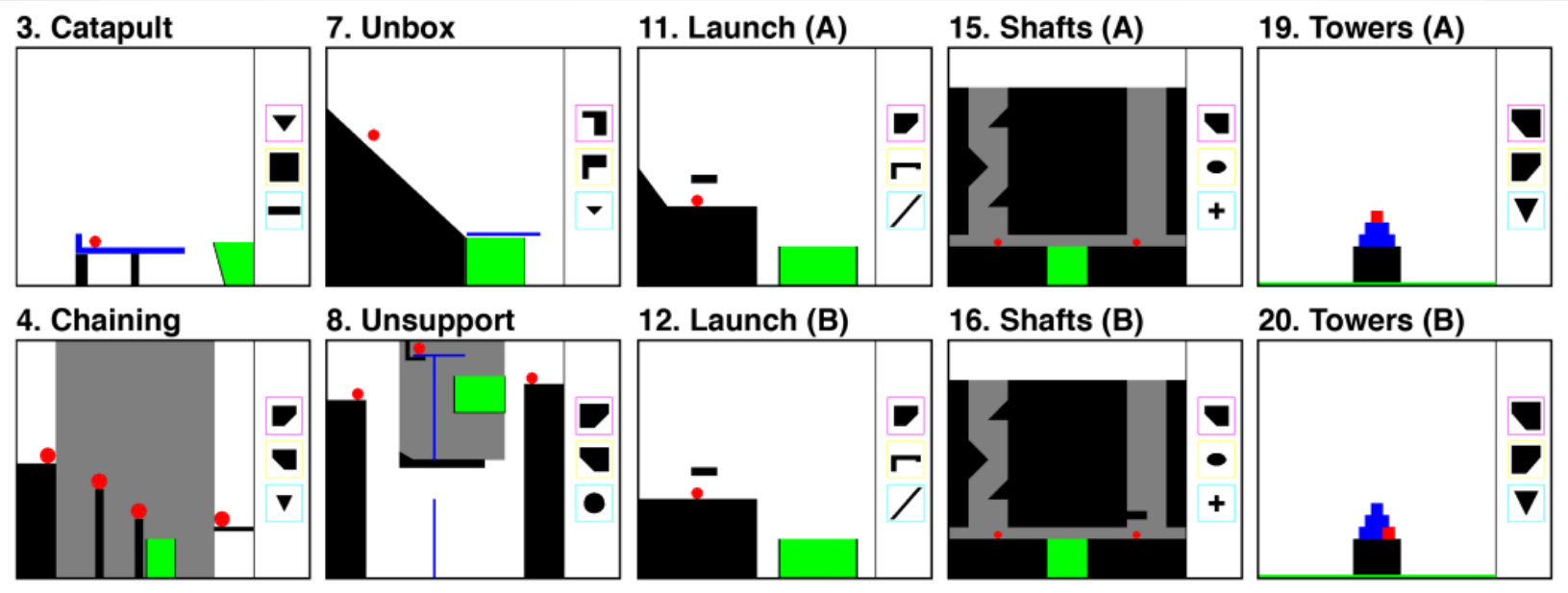


- Types of task: Multiple cognitive levels
- Dynamic Scene objects: Tools
- Action has two variables: Type of tool and location of tool
- Intuitive aspect:
  - Tool Type : nature of task
  - Vertical location : force
  - Horizontal location : nature of task

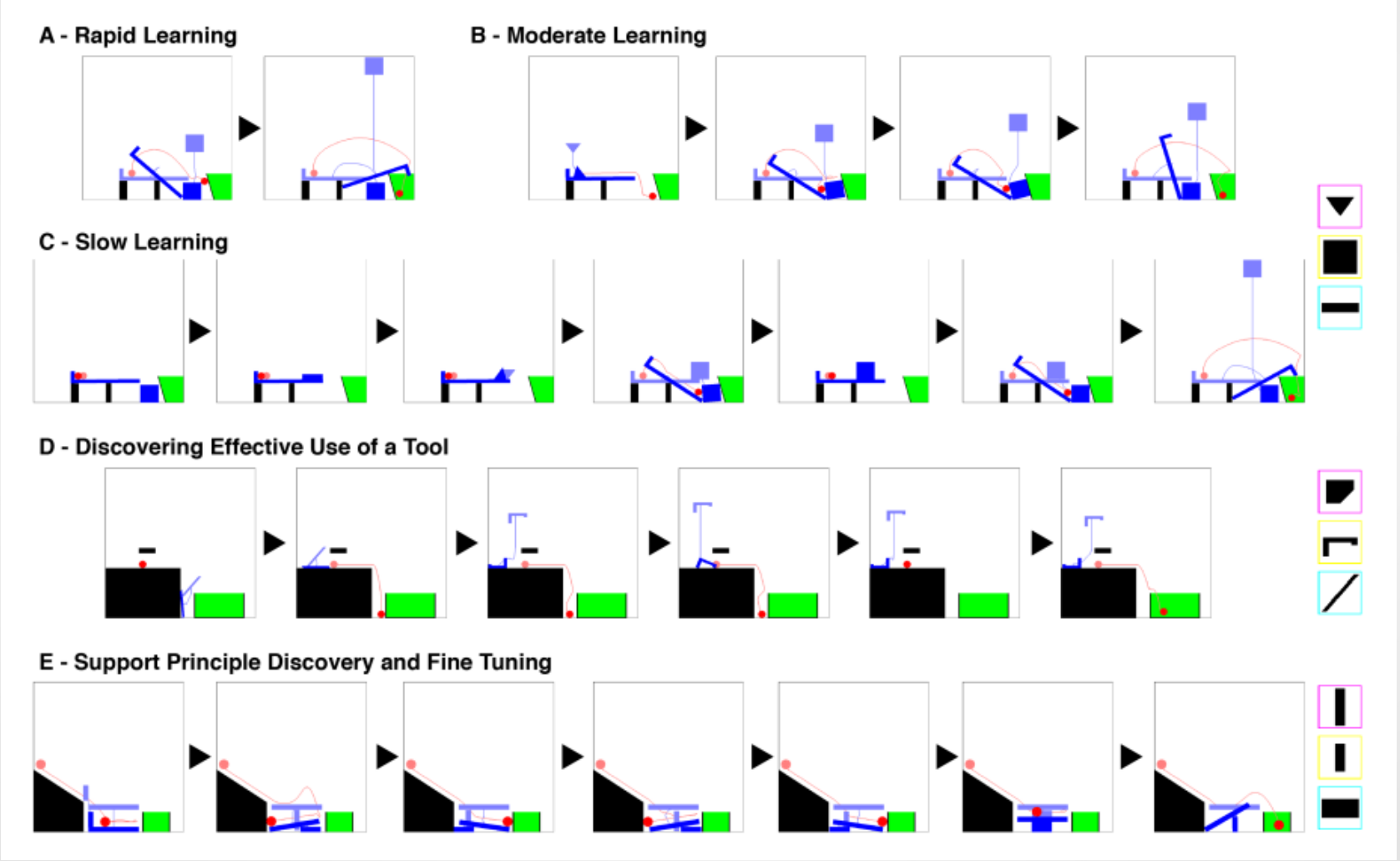
Salient Features (Phyre, Tools)

- Diversity of levels but physical dynamics is same
  - Assumption/Reasoning – Human expert comes with a noisy physics simulation engine and a rough idea of tools.
  - Use of Priors for location (Planner), tool (Red Circle) is uniform (although human have bias)
  - A noisy physics engine (Dreamer)
- Causal reasoning
  - Assumption/Reasoning – Human expert might get the location of tool usage right in the first shot but the tool used was wrong. Or the tool was right but the location needed adjustment.
  - Can this partial knowledge be transferred for the next trial ?
- Long horizon predictions and expectation of Few-shot trial-and-error learning
  - Search has elements of randomness, but within a plausible solution space
  - “Sample, Simulate, Update” (SSUP) framework
  - (Dream (SAMPLE) Multiple noisy trials from physics engine as per priors else random in epsilon-greedy way to SIMULATE and then UPDATE)
  - “This style of problem solving is a very structured sort of trial-and-error learning”

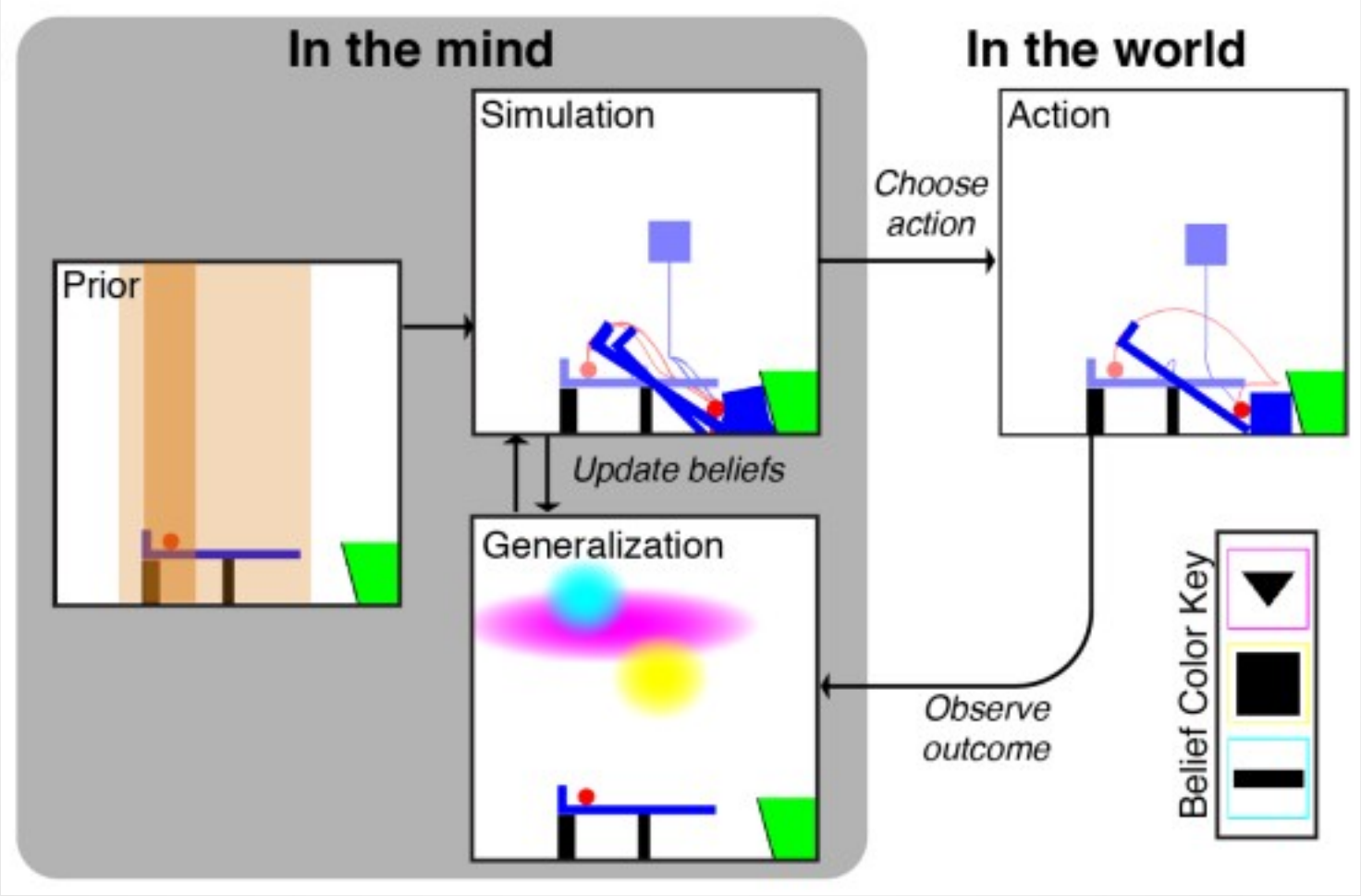
Task


3. Catapult
7. Unbox
11. Launch (A)
15. Shafts (A)
19. Towers (A)
4. Chaining
8. Unsupport
12. Launch (B)
16. Shafts (B)
20. Towers (B)

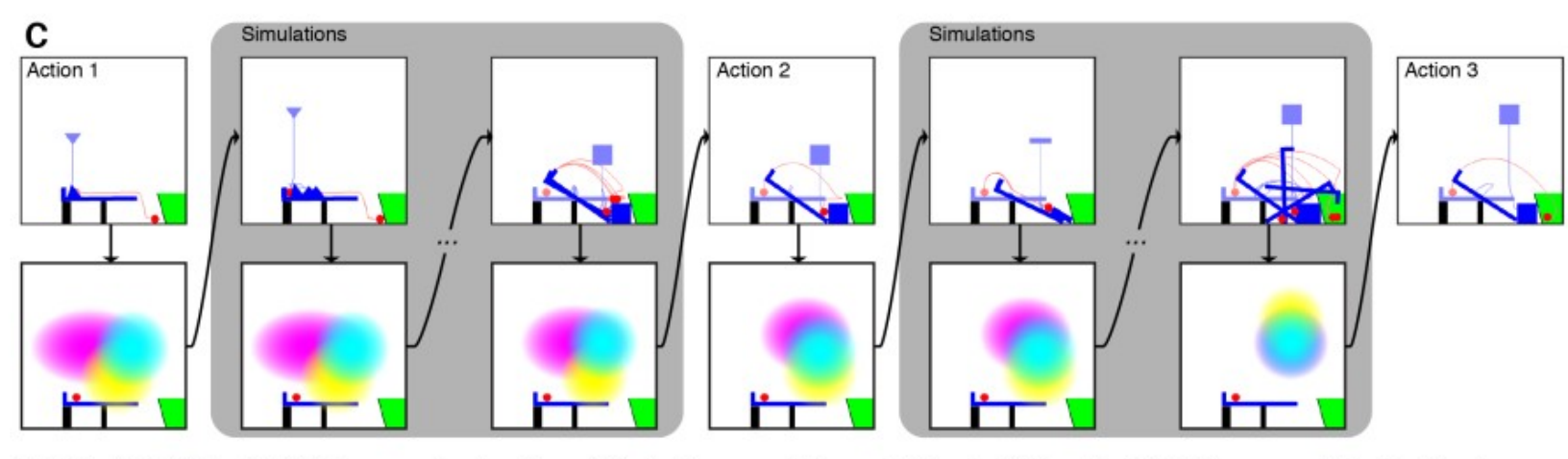
Human Expert



Architecture



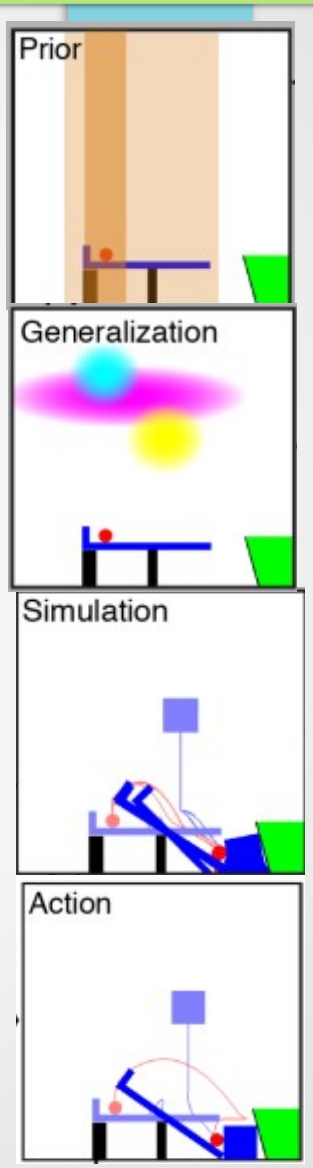
Pipeline



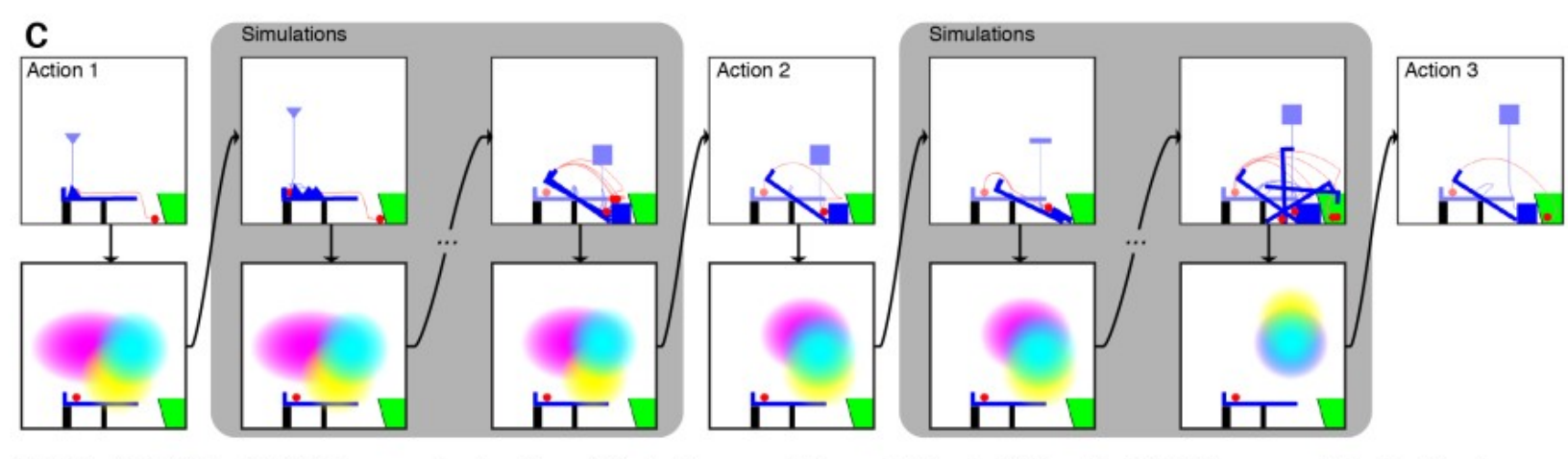
SSUP

- Sample from Object-based prior / Policy
- Simulate on a noisy physics engine
- Update from thoughts(Simulation outcome) and actions(Real outcome)

structured sort of trial-and-error learning

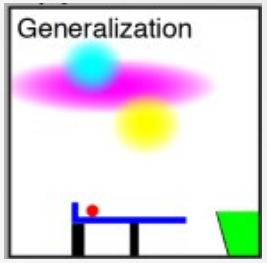


Pipeline



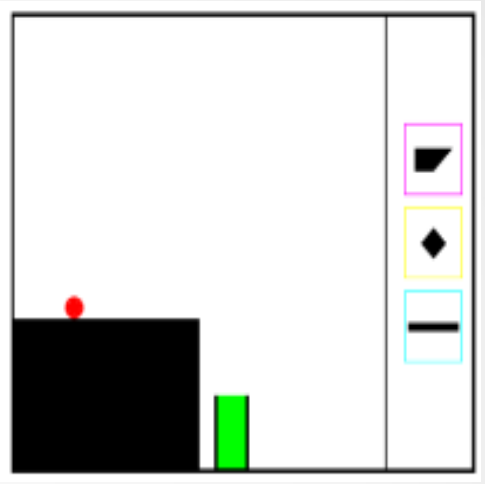
Update

- Learn : One Gaussian per Action
- How : Rapid-trial-and-error
- Update method : Policy-Gradient
- Loss : achieved minimum distance to goal



“This algorithm will shape the posterior beliefs around areas to place each tool which are expected to move target objects close to the goal, and therefore is likely to contain a solution”

Model

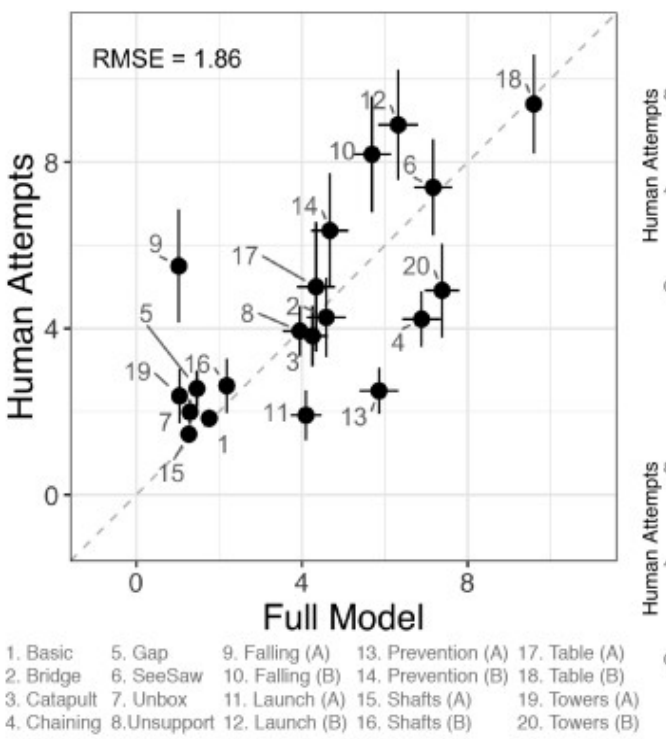


- Sample a dynamic object from scene objects
- Sample a position y relative to that object
  - using a Gaussian distribution parameterized with mean object y and standard deviation σy
  - Represents the size of the tool, as due to higher or lower height, higher or lower impact is induced
- Sample a position x relative to that object:
  - Compute the left and right edges of the bounding box for that object, BB left and BB right
  - Sample a value v uniformly between BB left − σ_x and BB right − σ_x.
  - If v < BB left or v > BB right, sample x from a normal centered on the edge of the bounding box with standard deviation σ_x.
  - Otherwise, x = v.

Algorithm S1 SSUP model for the Tools game

Sample n_{init} points from prior $\pi(s)$ for each tool
Simulate actions to get noisy rewards $\hat{r}$ using internal model
Initialize policy parameters θ using policy gradient on initial points
while not successful **do**
  Set $acting = False$
  With probability ε, sample action a from prior
  With probability $1 - \varepsilon$, sample action a from policy
  Estimate noisy reward $\hat{r}$ from internal model on action a
  if $r > T$ **then**
    Set $acting = True$
    Try action a in environment
  else if $i \geq n_{iters}$ **then**
    Set $acting = True$
    Try best action a^* simulated so far which has not yet been tried
  end if
  if acting **then**
    Observe r from environment on action a.
    If successful, exit.
    Simulate $\hat{r}$ assuming other two tool choices.
    Update policy based on all three estimates and actions.
  else
    Update policy using policy gradient
  end if
end while

Results



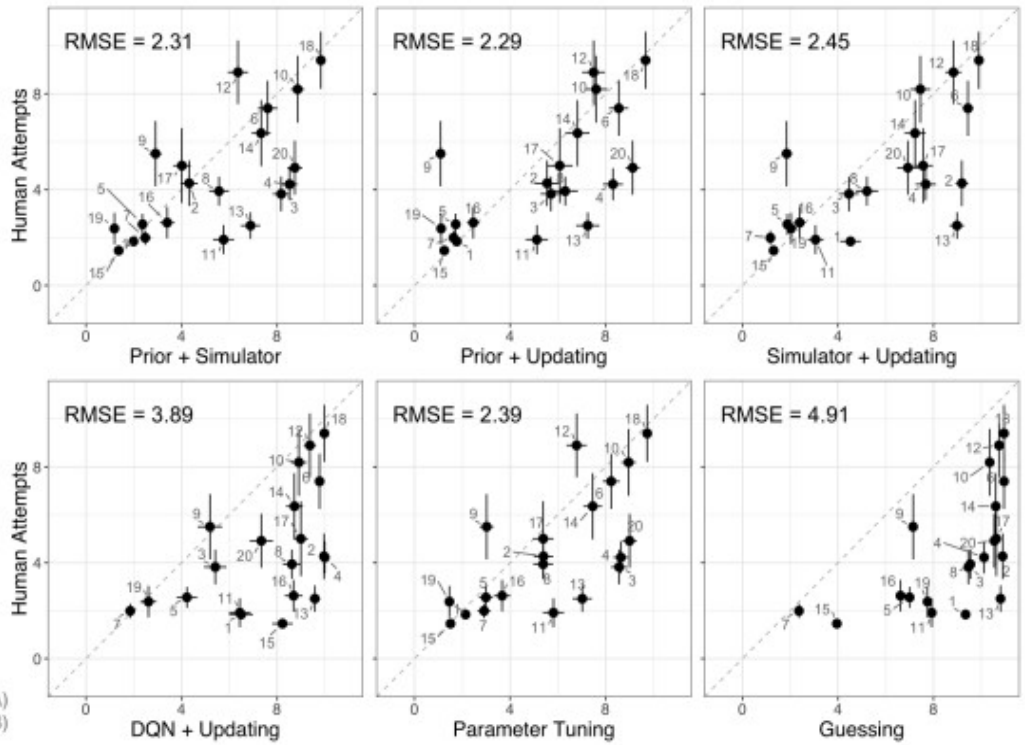


Figure 5: Comparison of average number of human participants' attempts for each level with average number of attempts for the full model (*left*) and six alternate models (see Section 3.2 for descriptions). Numbers correspond to the trials in Fig. 2. Bars indicate 95% confidence intervals on estimates of the means. The number of placements was capped at 10 for all models. If a model took more than 10 attempts on a particular level, it is considered unsolved. Model results are combined over 250 runs.

Notice Ablation in results

Results

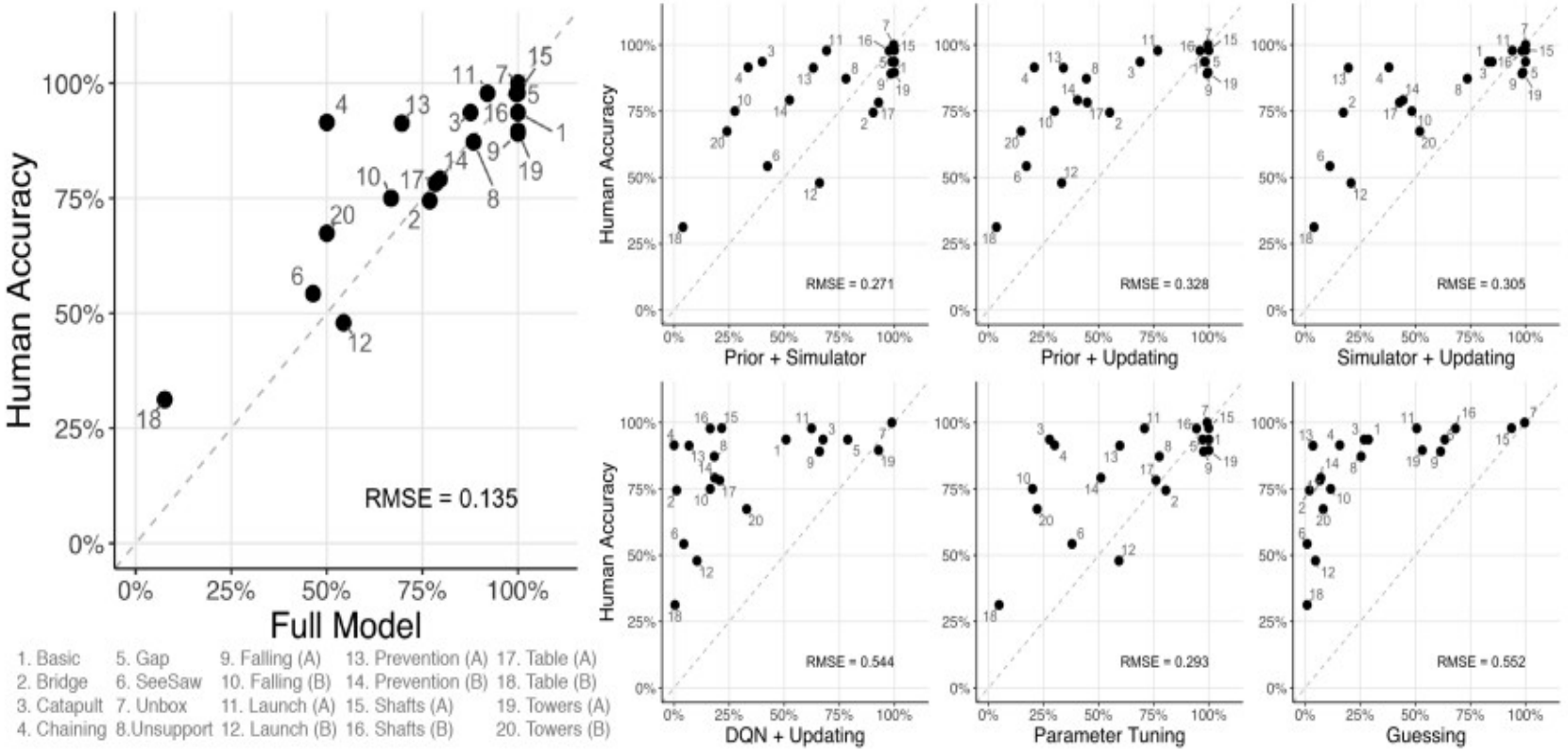


Figure 6: Comparison of human participants' accuracy on each trial versus the accuracy of all models. Numbers correspond to the trials in Fig. 2. DQN+Updating and Guessing perform badly across most levels.

Results

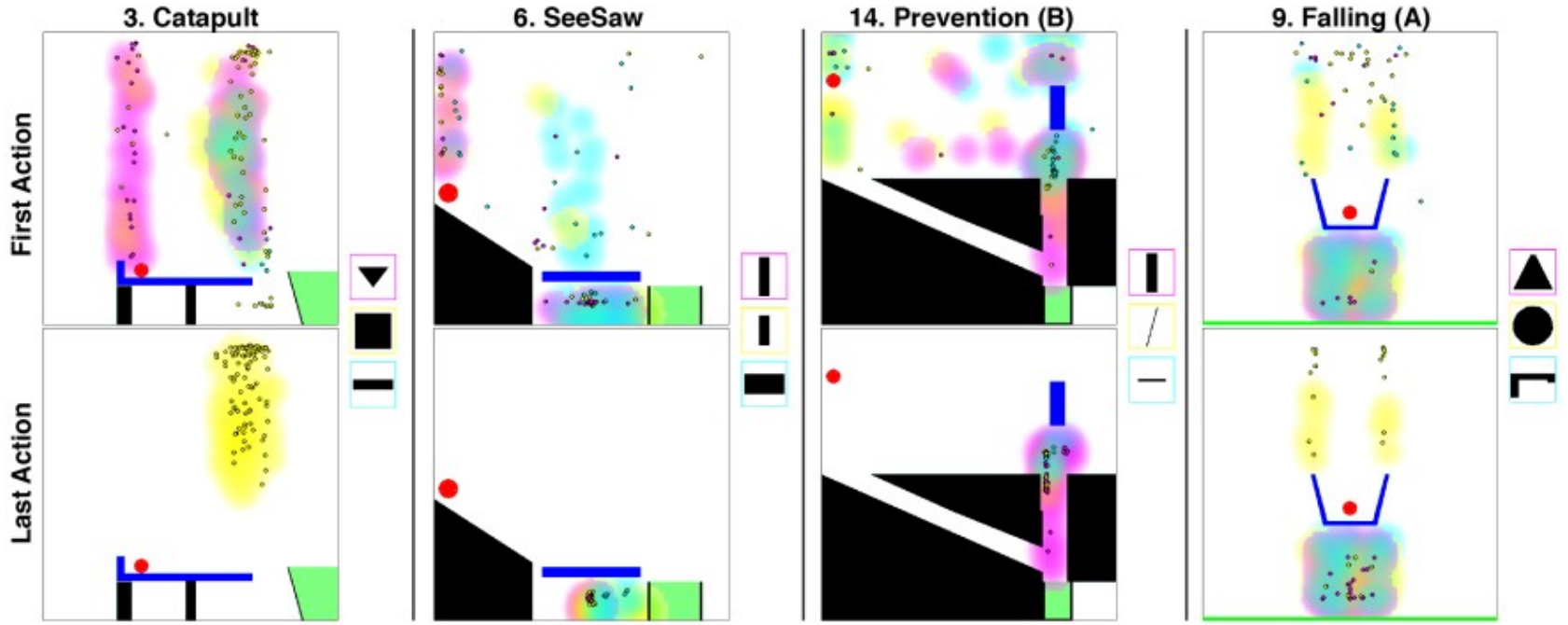


Figure 7: Distribution of predicted model actions (background) versus human actions (points) on the first attempts of the level (*top*) and the attempt used to solve the level (*bottom*) for a selection of four levels. Colors indicate the tool used, with the tools and associated colors shown to the right of each level. Across many levels that include using concepts such as launching, catapulting, and blocking, the model captures the diversity of where people initialize their search, as well as the types of solutions they eventually find. However, there are certain levels where people appear to have different prior beliefs than the model (Falling (A); *right*), which lead to different solution patterns.

Results

Model	Avg. Attempts	Attempt RMSE	Accuracy	Accuracy RMSE	Total ABCS
Human	4.48 [4.25, 4.66]	-	0.81	-	-
Full Model	4.24 [4.17, 4.32]	1.86 [1.66, 2.17]	0.77 [0.76, 0.78]	0.135 [0.121, 0.169]	2.21
Prior + Simulator	5.38 [5.32, 5.46]	2.31 [2.16, 2.59]	0.69 [0.68, 0.7]	0.271 [0.248, 0.299]	3.78
Prior + Updating	5.23 [5.14, 5.31]	2.29 [2.12, 2.58]	0.59 [0.58, 0.6]	0.328 [0.307, 0.353]	4.37
Simulator + Updating	5.55 [5.48, 5.62]	2.45 [2.29, 2.71]	0.61 [0.6, 0.62]	0.305 [0.288, 0.33]	4.05
DQN + Updating	7.52 [7.44, 7.61]	3.89 [3.76, 4.1]	0.34 [0.33, 0.35]	0.544 [0.527, 0.566]	7.95
Parameter Tuning	5.7 [5.64, 5.78]	2.39 [2.25, 2.65]	0.65 [0.64, 0.66]	0.293 [0.275, 0.323]	3.88
Guessing	8.88	4.91 [4.75, 5.1]	0.32	0.552 [0.531, 0.573]	8.04

Table 1: Comparisons with alternate models. Brackets indicate bootstrapped 95% confidence intervals on the estimate. 'Total ABCS' refers to the sum of the *Area Between Cumulative Solution* curves of participants and the model (see Fig. 8). 'Guessing' model placements and accuracy can be calculated exactly and therefore have no confidence intervals.

Thank You