Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning

Kelsey R. Allen[a,b,1,2], **Kevin A. Smith**[a,b,1], **and Joshua B. Tenenbaum**[a,b]

[a]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139; and [b]Center for Brains, Minds, and Machines, Cambridge, MA 02139

Edited by Nikolaus Kriegeskorte, Columbia University, New York, NY, and accepted by Editorial Board Member Dale Purves April 11, 2020 (received for review November 1, 2019)

Many animals, and an increasing number of artificial agents, display sophisticated capabilities to perceive and manipulate objects. But human beings remain distinctive in their capacity for flexible, creative tool use—using objects in new ways to act on the world, achieve a goal, or solve a problem. To study this type of general physical problem solving, we introduce the Virtual Tools game. In this game, people solve a large range of challenging physical puzzles in just a handful of attempts. We propose that the flexibility of human physical problem solving rests on an ability to imagine the effects of hypothesized actions, while the efficiency of human search arises from rich action priors which are updated via observations of the world. We instantiate these components in the "sample, simulate, update" (SSUP) model and show that it captures human performance across 30 levels of the Virtual Tools game. More broadly, this model provides a mechanism for explaining how people condense general physical knowledge into actionable, task-specific plans to achieve flexible and efficient physical problem solving.

intuitive physics | physical problem solving | tool use

While trying to set up a tent on a camping trip, you realize that the ground is too hard for the tent stakes, and you have no hammer. What would you do? You might look around for a suitable hammer substitute, passing over objects like pinecones or water bottles in favor of a graspable rock. And if that rock failed to drive in the stakes at first, you might try a different grip or search for a heavier rock. Most likely, you would need only a handful of attempts before you found an approach that works. Determining how to pound in tent stakes without a hammer is an example of the flexibility and efficiency of more general physical problem solving. It requires a causal understanding of how the physics of the world work and sophisticated abilities for inference and learning to construct plans that solve a novel problem. Consider how, when faced with the tent stake challenge, we do not choose an object at random; we choose a rock because we believe we know how we could use it to generate sufficient force on the stake. And if we find that the first rock fails, we again search around for a solution, but use the knowledge of our failures to guide our future search. This style of problem solving is a very structured sort of trial-and-error learning: Our search has elements of randomness, but within a plausible solution space, such that the goal can often be reached very quickly.

Here we study the cognitive and computational underpinnings of flexible tool use. While human tool use relies on a number of cognitive systems—for instance, knowing how to grasp and manipulate an object or understanding how a particular tool is typically used—here we focus on "mechanical reasoning," or the ability to spontaneously repurpose objects in our environment to accomplish a novel goal (3–5).

We target this mechanical reasoning because it is the type of tool use that is quintessentially human. While other animals can manipulate objects to achieve their aims, only a few species of birds and primates have been observed to spontaneously use objects in novel ways, and we often view these activities as some of the most "human-like" forms of animal cognition (e.g., Fig. 1 *A* and *B*) (6). Similarly, while artificial intelligence (AI) systems have become increasingly adept at perceiving and manipulating objects, none perform the sort of rapid mechanical reasoning that people do. Some artificial agents learn to use tools from expert demonstrations (7), which limits their flexibility. Others learn from thousands of years of simulated experience (8), which is significantly longer than required for people. Still others can reason about mechanical functions of arbitrary objects but require perfect physical knowledge of the environment (9), which is unavailable in real-world scenarios. In contrast, even young humans are capable tool users: By the age of 4 years they can quickly choose an appropriate object and determine how to use it to solve a novel task (e.g., picking a hooked rather than a straight pipe cleaner to retrieve an object from a narrow tube; Fig. 1*C*) (10).

What are the cognitive systems that let us use tools so flexibly and accomplish our goals so rapidly? It has been suggested that mechanical reasoning relies on mental simulation, which lets us predict how our actions will cause changes in the world (3). This general-purpose simulation is a necessary component that supports our ability to reason about objects in novel environments, but by itself cannot explain how we make and update our plans so quickly. We propose that another key to rapid tool use is knowing what sorts of actions to even consider—both from an initial understanding of what actions are useful and by updating this belief from observing the outcome of our actions, in simulation and in reality.

This paper makes two contributions. First, we introduce the Virtual Tools game, which presents a suite of physical problem-solving challenges and allows for precise, quantifiable comparisons between human and machine agents. Second, we present

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, "Brain Produces Mind by Modeling," held May 1–3, 2019, at the Arnold and Mabel Beckman Center of the National Academies of Sciences and Engineering in Irvine, CA. NAS colloquia began in 1991 and have been published in PNAS since 1995. From February 2001 through May 2019, colloquia were supported by a generous gift from The Dame Jillian and Dr. Arthur M. Sackler Foundation for the Arts, Sciences, & Humanities, in memory of Dame Sackler's husband, Arthur M. Sackler. The complete program and video recordings of most presentations are available on the NAS website at http://www.nasonline.org/brain-produces-mind-by.

Author contributions: K.R.A., K.A.S., and J.B.T. designed research; K.R.A. and K.A.S. performed research; K.R.A. and K.A.S. analyzed data; and K.R.A., K.A.S., and J.B.T. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. N.K. is a guest editor invited by the Editorial Board.

Data deposition: Access to the Virtual Tools game and all data including human and model placements is provided at GitHub, https://k-r-allen.github.io/tool-games/.

[1]K.R.A. and K.A.S. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: krallen@mit.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.1912341117/-/DCSupplemental.

First published November 23, 2020.

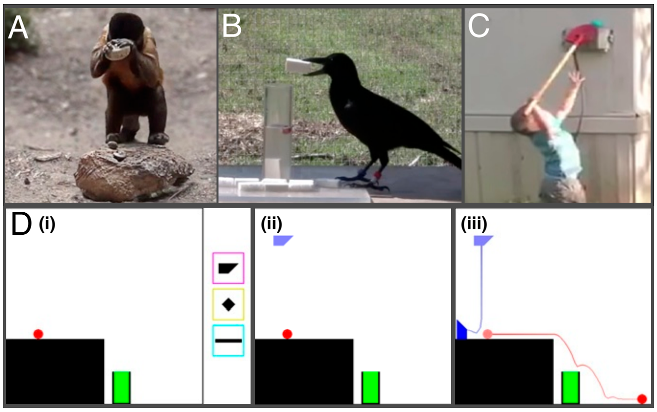


Fig. 1. Examples of using objects to achieve a goal. (*A*) Bearded capuchin monkey opening a cashew nut with an appropriately sized stone. Reprinted from ref. 1, which is licensed under CC BY 4.0. (*B*) New Caledonian crow using heavy blocks to raise the water level in a tube to retrieve food (2). (*C*) Toddler using a shovel to reach a ball. Image credit: YouTube/Funny Vines (http://youtu.be/hwrNQ93-568?t=198). (*D*) One illustrative trial in the Virtual Tools game (https://k-r-allen.github.io/tool-games/). (*D*, *i*) The player must get the red object into the green goal using one of the three tools. (*D*, *ii*) The player chooses a tool and where to place it. (*D*, *iii*) Physics is turned "on" and the tool interacts with other objects. The action results in a near miss.

a minimal model of flexible tool use, called "sample, simulate, update" (SSUP). This model is built around an efficient albeit noisy simulation engine that allows the model to act flexibly across a wide variety of physical tasks. To solve problems rapidly, the SSUP model contains rich knowledge about the world in the form of a structured prior on candidate tools and actions likely to solve the problem, which allows it to limit its simulations to promising candidates. It further learns from its simulations and from observing the outcome of its own actions to update its beliefs about what those promising candidates should be. Across 30 Virtual Tools levels in two experiments, we show that an instantiation of the SSUP model captures the relative difficulties of different levels for human players, the particular actions performed to attempt to solve each level, and how the solution rates for each level evolve.

The Virtual Tools Game Inspired by human tool use, as well as mobile physics games (11), we propose the Virtual Tools game as a platform for investigating the priors, representations, and planning and learning algorithms used in physical problem solving (https://k-r-allen.github.io/tool-games/). This game asks players to place one of several objects ("tools") into a two-dimensional (2D) dynamic physical environment to achieve a goal: getting a red object into a green region (Fig. 1*D*). This goal is the same for every level, but what is required to achieve it varies greatly. Once a single tool is placed, the physics of the world are enabled so that players see the effect of the action they took. If the goal is not achieved, players can "reset" the world to its original state and try again; they are limited to a single action on each attempt. We designed 30 levels—20 for the original experiment (Fig. 2) and 10 for a validation experiment (see Fig. 7*A*)—to test concepts such as "launching," "blocking," and "supporting." Of the first 20 levels, 12 were constructed in six "matched pairs" which incorporated small differences in the goals or objects in the scene to test whether subtle differences in stimuli would lead to observable differences in behavior.

The Virtual Tools game presents particular challenges that we believe underlie the kinds of reasoning required for rapid physical problem solving more generally. First, there is a diversity of tasks that require different strategies and physical concepts to solve, but employ shared physical dynamics that approximate the real world. Second, the game requires long-horizon causal reasoning. Since players can interact with the game only by placing a single object, they must be able to reason about the complex cause and effect relationships of their action long into the future when they can no longer intervene. Finally, the game elicits rapid trial-and-error learning in humans. Human players do not generally solve levels on their first attempt, but also generally do not require more than 5 to 10 attempts to succeed. People demonstrate a wide range of problem-solving behaviors, including "aha" insights where they suddenly discover the right idea for how to solve a particular task, as well as incremental trial-and-error strategy refinement. Fig. 3 demonstrates how this occurs in practice, showing four different examples of participants learning rapidly or slowly and

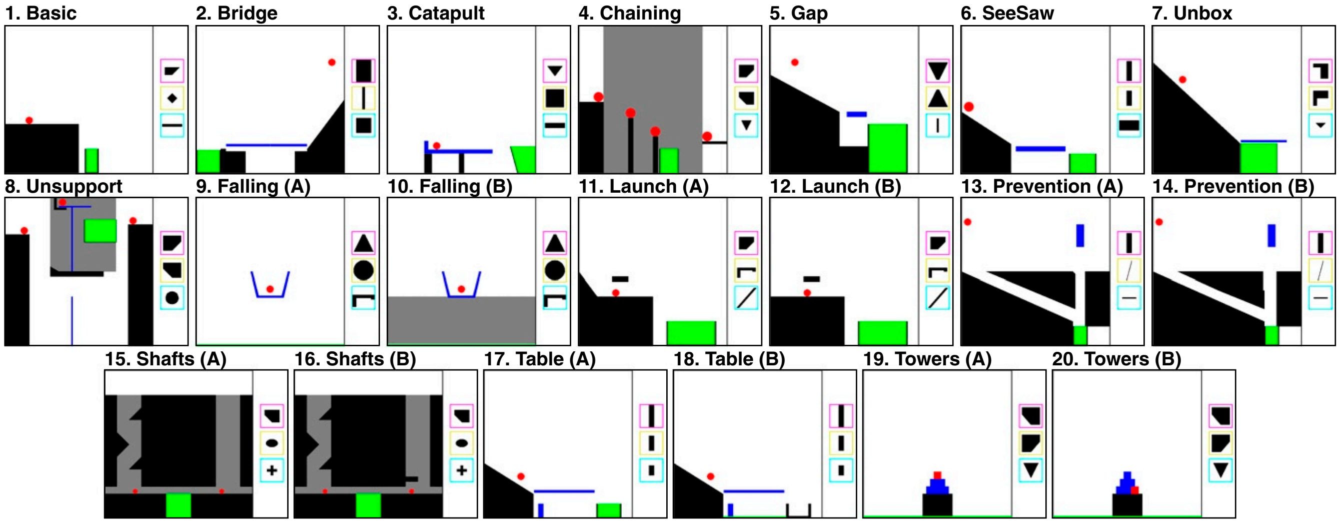


Fig. 2. Twenty levels used in the Virtual Tools game. Players choose one of three tools (shown to the right of each level) to place in the scene to get a red object into the green goal area. Black objects are fixed, while blue objects also move; gray regions are prohibited for tool placement. Levels denoted with A/B labels are matched pairs.

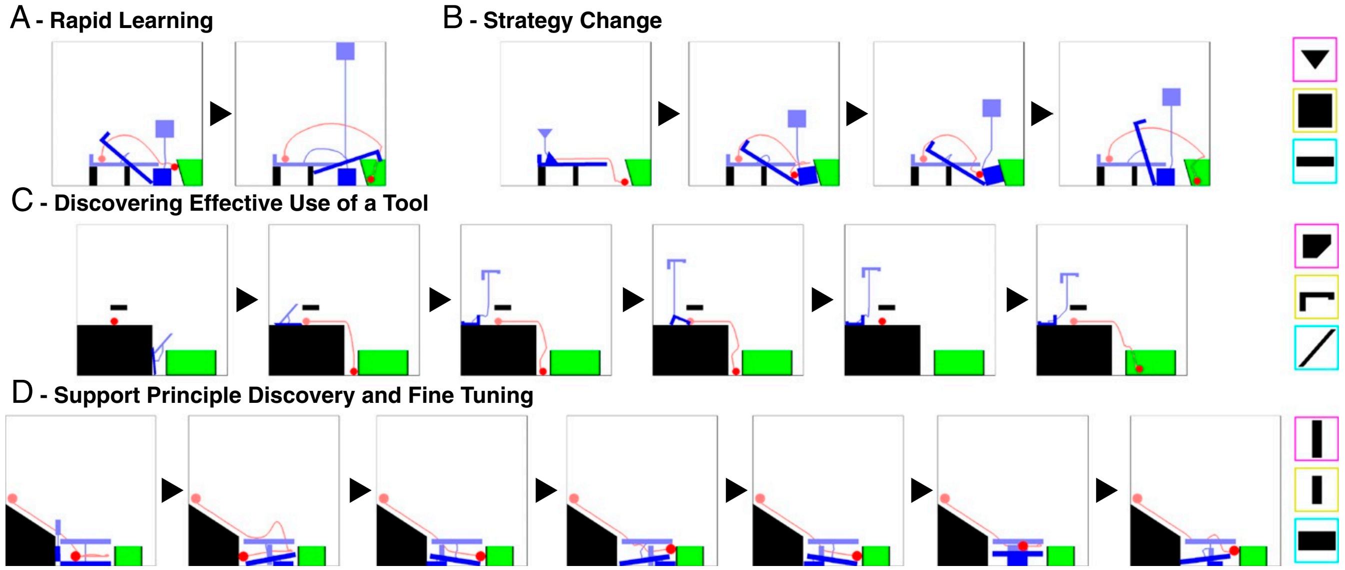

A - Rapid Learning
B - Strategy Change
C - Discovering Effective Use of a Tool
D - Support Principle Discovery and Fine Tuning

Fig. 3. Examples of participants' behavior on three levels, representative of rapid trial-and-error learning: Initial plans are structured around objects, followed by exploring to identify more promising strategies and then refining actions until success. Objects start as shown by light blue/red outlines and follow paths traced out by colored lines. Possible tool choices are shown at *Right*. (*A*) In the Catapult level, a useful strategy is often identified immediately and rapidly fine-tuned. (*B*) Other participants first try an unsuccessful strategy but then switch to a more viable strategy and refine it. (*C*) The Launch (*B*) level is designed to prevent obvious solutions. This participant may have initially believed the ball would start rolling and attempted to use a tool as a bridge. When this failed, the participant realized the need to launch the ball but discovered only after several trials how to use a tool in a nonobvious way to accomplish this, via a hooking motion around the blocking ledge. The participant then took several more trials to fine-tune this action. (*D*) In the SeeSaw level, a participant realized on the second attempt the platform must be supported for the ball to roll across and then tried different ways of making this happen.

discovering different ways to use the tools across a variety of levels.

SSUP Model We consider the components required to capture both the flexibility and efficiency of human tool use. We propose that people achieve flexibility through an internal mental model that allows them to imagine the effects of actions they may have never tried before ("simulate"). However, a mental model alone is not sufficient—there are far too many possible actions that could be simulated, many of which are uninformative and unlikely to achieve a specific goal. Some mechanism for guiding an internal search is necessary to focus on useful parts of the hypothesis space. We therefore propose people use structured, object-oriented priors ("sample") and a rapid belief updating mechanism ("update") to guide the search toward promising hypotheses. We formalize human tool use with these components in the SSUP model (Fig. 4*A*).

SSUP is inspired by the theory of "problem solving as search" (12), as well as Dyna and other model-based policy optimization methods (13, 14). Crucially, we posit that structured priors and physical simulators must already be in place to solve problems as rapidly as people; thus unlike most model-based policy optimization methods, we do not perform online updates of the dynamics model.

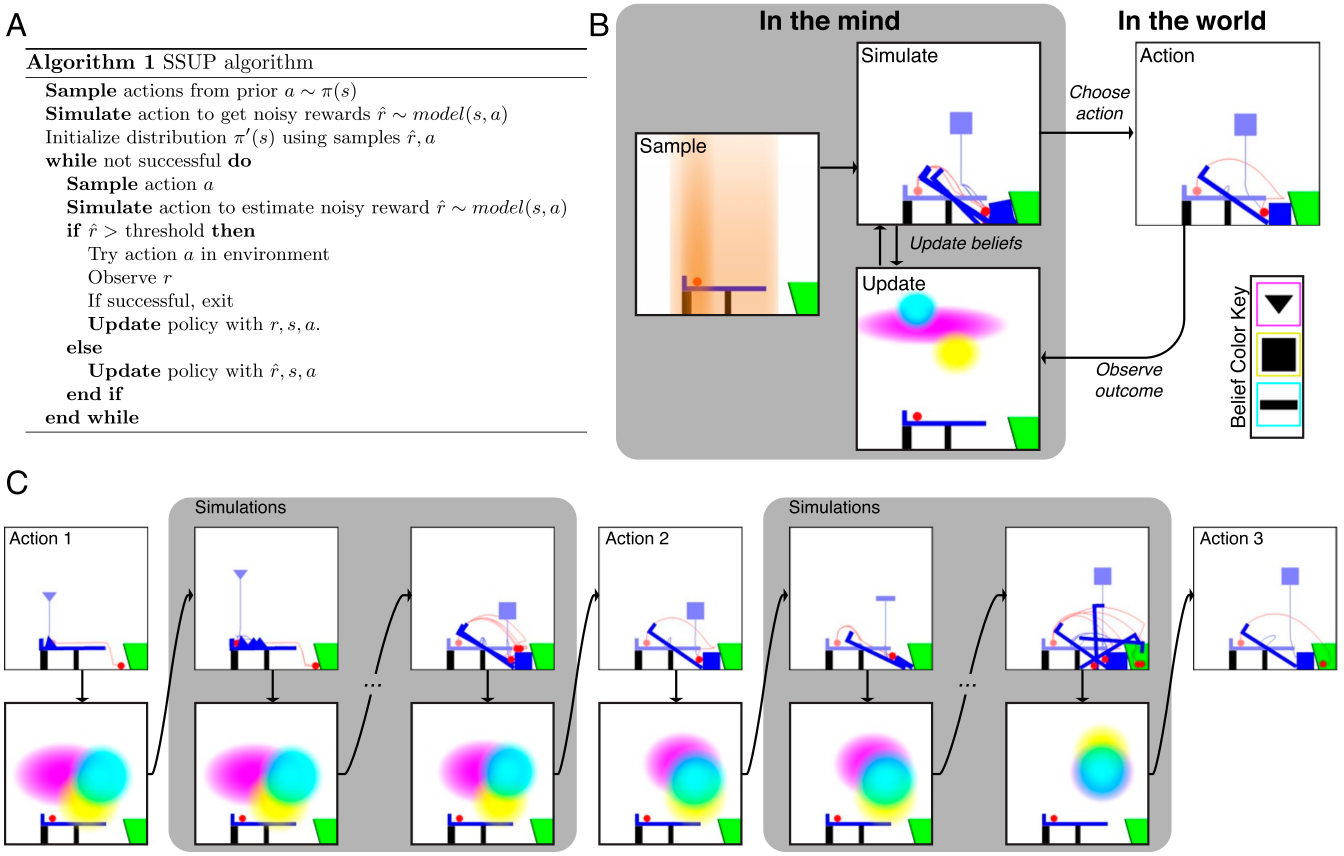
We emphasize that we view SSUP as a general modeling framework for physical problem solving and present here only one instance of that framework: the minimal model (described below, with more detail in *SI Appendix*, section S2) that we think is needed to capture basic human behavior in the Virtual Tools game. In the *Discussion* we highlight ways the model will need to be improved in future work, as well as aspects of physical reasoning that rely on a richer set of cognitive systems going beyond the framework presented here.

Sample: Object-Based Prior. At a minimum, the actions we should consider to achieve any goal should have the potential to impact our environment. We therefore incorporate an object-based prior for sampling actions. Specifically, the model selects one of the movable objects in the scene and then chooses an *x* coordinate in an area that extends slightly beyond the width of the object and a *y* coordinate either above or below that object (Fig. 4*B*: sample). For tool choice, we assume participants are equally likely to choose any of the three tools since all tools in the game were designed to be unfamiliar to participants. Samples from this distribution are used to initialize search.

Simulate: A Noisy Physics Engine. To determine which sampled actions are worth trying in the world, we assume people use an "intuitive physics engine" (15) to flexibly imagine the effects of their actions. This engine is able to simulate the world forward in time with approximately correct but stochastic dynamics (16, 17). Determining the effect of a proposed action therefore involves applying that action to one's mental representation and using the intuitive physics engine to posit the range of ways that action might cause the world to unfold (18, 19). Here we implement simulation using a game physics engine with noisy dynamics. People characteristically have noisy predictions of how collisions will resolve (16), and so for simplicity we assume uncertainty about outcomes is driven only by noise in those collisions (the direction and amount of force that is applied between two colliding objects).*

Since the internal model is imperfect, to evaluate an action we produce a small number of stochastic simulations (n_{sims}, set here at four) to form a set of hypotheses about the outcome. To formalize how good an outcome is (the reward of a given action),

*We also considered models with additional sources of physics model uncertainty added but found that the additional parameters did not improve model fit, so we do not analyze those models here.



```
Algorithm 1 SSUP algorithm
Sample actions from prior $a \sim \pi(s)$
Simulate action to get noisy rewards $\hat{r} \sim model(s,a)$
Initialize distribution $\pi'(s)$ using samples $\hat{r}, a$
while not successful do
  Sample action $a$
  Simulate action to estimate noisy reward $\hat{r} \sim model(s,a)$
  if $\hat{r} >$ threshold then
    Try action $a$ in environment
    Observe $r$
    If successful, exit
    Update policy with $r, s, a$.
  else
    Update policy with $\hat{r}, s, a$
  end if
end while
```

Fig. 4. (*A*) The SSUP algorithm. (*B*) A diagram of the model for the Virtual Tools game. It incorporates an object-based prior, a simulation engine for filtering proposals, and an update module that suggests new proposals based on observations "in the mind" and from actions taken in the world. (*C*) Illustration of the policy π' evolving while attempting a level. Colored patches represent the Gaussian policy for each tool as indicated by the Belief Color Key.

we borrow an idea from the causal reasoning literature for how people conceptualize "almost" succeeding (20). Almost succeeding is not a function of the absolute distance an action moved you toward your goal, but instead how much of a difference that action made. To capture this, the minimum distance between the green goal area and any of the red goal objects is recorded; these values are averaged across the simulations and normalized by the minimum distance that would have been achieved if no tool had been added. The reward used in SSUP is 1 minus the normalized distance, so that closer objects lead to higher reward.

Once the model finds a good enough action (formalized as the average reward being above some threshold), it takes that action "in the world." Additionally, to model time limits for thinking, if the model considers more than T different action proposals without acting (set here at five), it takes the best action it has imagined so far. We evaluate the effect of all parameter choices in a sensitivity analysis (*SI Appendix*, Fig. S1).

Update: Learning from Thoughts and Actions. So far, we have described a way of intelligently initializing a search to avoid considering actions that will not be useful. But what if the prior still presents an intractably large space of possible actions?

To tackle this, we incorporate an update mechanism that learns from both simulated and real experience to guide future search toward more promising regions of the hypothesis space (21). This is formally defined as a Gaussian mixture model policy over the three tools and their positions, $\pi'(s)$, which represents the model's belief about high-value actions for each tool. $\pi'(s)$ is initialized with samples from the object-oriented prior and updated using a simple policy gradient algorithm (22). 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 are therefore likely to contain a solution. Such an update strategy is useful when it finds high-value actions that are nearby successful actions, but may also get stuck in local optima where a successful action does not exist. We therefore use a standard technique from reinforcement learning: epsilon-greedy exploration. With epsilon-greedy exploration, potential actions are sampled from the policy $100 - \epsilon\%$ of the time and from the prior $\epsilon\%$ of the time. Note that this exploration is used only for proposing internal simulations; model actions are chosen based on the set of simulation outcomes. This is akin to thinking of something new, instead of focusing on an existing strategy.

Results

We analyze human performance on the first 20 levels of the Virtual Tools game and compare humans to the SSUP model and alternates, including SSUP models with ablations and two alternate learning baselines. We show that the full SSUP model best captures human performance. Access to the game and all data including human and model placements is provided at https://k-r-allen.github.io/tool-games/.

Human Results. Experiments were approved by the Massachusetts Institute of Technology Committee on the Use of Humans as Experimental Subjects under protocol 0812003014. Participants were notified of their rights before the experiment, were free to terminate participation at any time by closing the browser window, and were compensated monetarily for their time.

We recruited 94 participants through Amazon Mechanical Turk and asked each participant to solve 14 levels: all 8 of the unmatched levels and one variation of each of the 6 matched pairs (randomly selected).

Participants could choose to move on once a problem was solved or after 2 min had passed. See *SI Appendix*, section S1 for further details.

The variation in difficulty between levels of the game was substantial. Participants showed an average solution rate of 81% (SD = 19%), with the range covering 31% for the hardest level to 100% for the easiest. Similarly, participants took an average of 4.5 actions (SD = 2.5) for each level, with a range from 1.5 to 9.4 average attempts. Even within trials, there was a large amount of heterogeneity in the number of actions participants used to solve the level. This would be expected with "rapid trial-and-error" learning: Participants who initially tried a promising action would solve the puzzle quickly, while others explored different actions before happening on promising ones (e.g., Fig. 3).

Behavior differed across all six matched level pairs. We study whether these subtle differences do indeed affect behavior, even without feedback on the first action, by asking whether we can identify which level variant each action came from. We find these actions are differentiable across matched levels in "Shafts," "Prevention," "Launch," and "Table" on the first attempt, but not "Falling" or "Towers" (see *SI Appendix*, Fig. S11 and section S6A for details). However, participants required a different number of actions to solve every level (all $ts > 2.7$, $ps < 0.01$). This suggests that people are paying attention to subtle differences in the scene or goal to choose their actions.

Model Results. We investigate several metrics for comparing the models to human data. First, we look at how quickly and how often each model solves each level and whether that matches participants. This is measured as the correlation and root-mean-square error (RMSE) between the average number of participant attempts for each level and the average number of model attempts for each level and the correlation and RMSE between human and model solution rates. The SSUP model explains the patterns of human behavior across the different levels well (*SI Appendix*, Table S2). It uses a similar number of attempts on each level ($r = 0.71$; 95% CI = $[0.62, 0.76]$; mean empirical attempts across all levels, 4.48; mean model attempts, 4.24; Fig. 5*A*) and achieves similar accuracy ($r = 0.86$; 95% CI = $[0.76, 0.89]$; Fig. 5*B*).

Across many levels, the SSUP model not only achieves the same overall solution rate as people, but also approaches it at the same rate. We measure this by looking at cumulative solution rates—over all participants or model runs, what proportion solved each level within X placements—and find that people and the model often demonstrate similar solution profiles (Fig. 6*A*; see *SI Appendix*, section S6B for quantitative comparison).

We can look in more detail at how the model accomplishes this by comparing both the first actions that people and the model take and the actions that both take to solve a level (Fig. 6*B*). Like our human participants, the model takes significantly different actions on the first attempt between matched level pairs (*SI Appendix*, section S6A). More generally, both people and the model will often begin with a variety of plausible actions (e.g., Catapult). In some cases, both will attempt initial actions that have very little impact on the scene [e.g., SeeSaw and Prevention (B)]; this could be because people cannot think of any useful actions and so decide to try something, similar to how the model can exceed its simulation threshold. However, in other cases, the model's initial predictions diverge from people, and this leads to a different pattern of search and solutions. For instance, in Falling (A), the model quickly finds that placing an object under the container will reliably tip the ball onto the ground, but people are biased to drop an object from above. Because of this, the model often rapidly solves the level with an object below, whereas a proportion of participants find a way to flip the container from above; this discrepancy can also be seen in the comparison of number of attempts before the solution, where the model finds a solution quickly, while people take a good deal longer (Fig. 5*A*). For comparisons of the first and last actions across all levels, see *SI Appendix*, Fig. S11.

Model Comparisons on Virtual Tools. We compare the full SSUP model against a set of six alternate models. Three models investigate the contribution of each SSUP component by removing the prior, simulation, or updating individually. Two models propose alternate solution methods: learning better world models rather than learning over actions (parameter tuning) or replacing the prior and simulator with a learned proposal mechanism (Deep Q Network [DQN, ref. 23] + updating). The parameter tuning alternate model uses inference to learn object densities, frictions, and elasticities from observed trajectories. The learned proposal mechanism corresponds to a model-free deep reinforcement learning agent (23) which is trained on a set of 4,500 randomly generated levels of the game (*SI Appendix*, section S5)

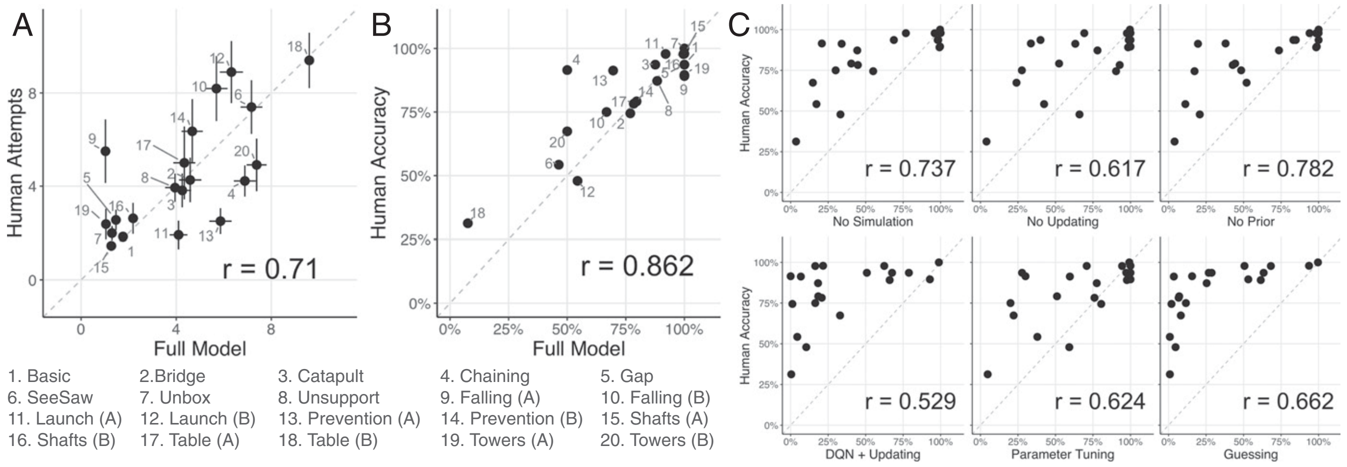


Fig. 5. (*A*) Comparison of average number of human participants' attempts for each level with average number of attempts for the SSUP model. Bars indicate 95% confidence intervals on estimates of the means. (*B*) Comparison of human participants' accuracy on each trial versus the accuracy of the SSUP model. (*C*) Comparison of human participants' accuracy to all alternate models. Numbers correspond to the trials in Fig. 2.

and then updated online for each of the 20 testing levels using the same mechanism as SSUP. This model has substantially more experience with the environment than other models and serves as a test of whether model-free methods can make use of this experience to learn generalizable policies that can guide rapid learning. Finally, we compare to a "guessing" baseline for performance if an agent were to simply place tools randomly. See Fig. 5*C* and *SI Appendix*, Table S2 for these comparisons.

Eliminating any of the three SSUP components causes a significant decrease in performance (measured as deviation between empirical and model cumulative solution curves; all bootstrapped $ps < 0.0001$; see *SI Appendix*, section S6B and Fig. S6 for further details). The reduced models typically require more attempts to solve levels because they are searching in the wrong area of the action space (no prior), attempting actions that have no chance of being successful (no simulation), or do not guide search toward more promising areas (no updating).

DQN + updating performs worst of all plausible alternate models, using the most actions and solving levels at a rate barely over chance. Because this is equivalent to the no simulation model with a different prior, its poor performance suggests that generalized action policies cannot easily be learned from repeatedly playing similar levels (*SI Appendix*, section S5).

Because the parameter tuning model is equivalent to the no updating model except that the properties of the dynamics model can be learned in parameter tuning, comparing those two models allows us to test whether we need to assume that people are learning the dynamics of the world in this game. The fact that both models perform roughly equivalently (Fig. 5*C*) suggests that we do not need this assumption here.

Finally, we quantified how well each model captured the particular actions people took. Due to heterogeneity in participants' responses, we were unable to cleanly differentiate models' performance except to find that the DQN + updating model underperformed the rest (*SI Appendix*, section S6C). However, no model reached the theoretical noise ceiling, suggesting components of the SSUP framework could be improved to better explain participants' actions (*Discussion*).

Validation on Novel Levels. We conducted a second experiment to test whether the models generalize to novel levels and physical concepts without tuning hyperparameters. For this experiment, we created 10 new levels: 6 novel level types and 4 variants of the originals (Fig 7*A*), testing an independent sample of 50 participants on all levels. The 6 novel level types were designed to test new physical strategies, including balancing, breaking, and removing objects from a ball's path. All other experimental details were identical to the main experiment.

Without tuning any model parameters, we find a good correspondence between human and model solution rates (Fig. 7*B*) and a strong correlation between the model's performance and human performance across number of placements (Fig. 7*C*, $r = 0.85$) and accuracy (Fig. 7*D*, $r = 0.95$). Similar to the main experiment, we find a decrement in performance if the prior or simulation is removed or for the DQN + updating model (all bootstrapped $ps < 0.0001$; *SI Appendix*, Fig. S7). However, while numerically worse, we do not find a reliable difference if the update mechanism is removed ($p = 0.055$) or swapped for model learning ($p = 0.346$), suggesting that the particular reward function or update procedure might be less applicable to these levels (*SI Appendix*, section S6B).

Discussion

We introduce the Virtual Tools game for investigating flexible physical problem solving in humans and machines and show that human behavior on this challenge expresses a wide variety of

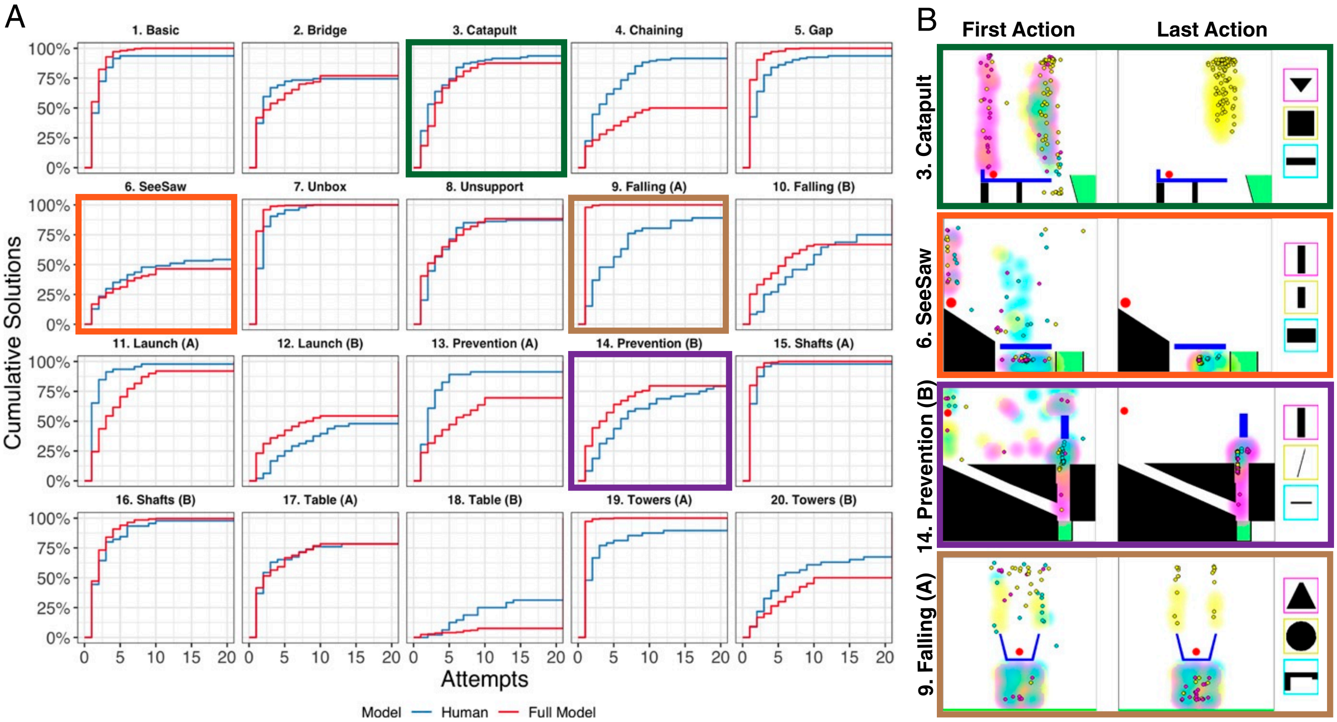


Fig. 6. (*A*) Cumulative solution rate over number of placements for participants vs. the SSUP model. (*B*) Distribution of model actions (background) versus human actions (points) on the first and last attempts of the level for a selection of four levels. The distribution of model actions is estimated based on fitting a kernel density estimate to the actions taken by the model across 250 simulations. Colors indicate the tool used, with the tools and associated colors shown at *Right* of each level. In most levels, the SSUP model captures the evolution of participants' solutions well, including the particular actions chosen; in the few cases that it differs, there is no alternative model that systematically explains these differences.

trial-and-error problem-solving strategies. We also introduce a model for human physical problem solving: sample, simulate, update. The model presumes that to solve these physics problems, people rely on an internal model of how the world works. Learning in this game therefore involves condensing this vast world knowledge to rapidly learn how to act in each instance, using a structured trial-and-error search.

Model Limitations. Although the SSUP model we used solves many of the levels of the Virtual Tools game in a human-like way, we believe that this is still only a first approximation to the rich set of cognitive processes that people bring to the task. In particular, there are at least two ways in which the model is insufficient: its reliance on very simple priors and its planning and generalizing only in the forward direction.

We can see the limits of the object-based prior in the Falling (A) level (Fig. 5*B*): People are much less likely to consider placing an object underneath the container to tip it over. Instead, many people try to tip it over from above, even though this is more difficult. In this way, people's priors over strategies are context specific, which causes them to be slower than the model in this level. In other cases, this context specificity is helpful: For instance, in the hypothetical level shown in Fig. 8*A*, there is a hole that one of the tools fits suspiciously perfectly into. Many people notice this coincidence quickly, but because the model cannot assess how tools might fit into the environment without running a simulation, it succeeds only 10% of the time. In future work, a more complex prior could be instantiated in the SSUP framework, but it remains an open question how people might form these context-specific priors or how they might be shaped over time via experience.

People show greater flexibility than our model in the ability to work backward from the goal state to find more easily solvable subgoals (24). In the hypothetical level in Fig. 8*B*, the catapult is finicky, which means that most catapulting actions will not make it over the barrier and therefore will never hit the ball on the left. Instead, the easiest way to increase the objective function is by the incorrect strategy of knocking the ball on the right to get close to the goal, and therefore the model solves the level only 8% of the time. Working backward to set the first subgoal of launching the ball over the barrier would prevent getting stuck with knocking the ball as a local minimum. From an engineering standpoint, creating subgoals is natural with discrete problem spaces (12), but it is less clear how these might be discovered in the continuous action space of the Virtual Tools game.

Related Cognitive Systems. There is an extensive body of research into the cognitive systems that underlie the use of real-world tools, including understanding how to manipulate them and knowing their typical uses (e.g., refs. 3, 4, 10, and 25). Here our focus was on "mechanical knowledge" of tools: how to use objects in novel situations. However, in real-world tool use, these systems work together with motor planning and semantic knowledge of tools. Future work can focus on these links, such as how novel tools become familiar or how our motor limits constrain the plans we might consider.

The Virtual Tools game presents a problem-solving task that blends facets of prior work, but encompasses a novel challenge. To rapidly solve these problems requires good prior knowledge of the dynamics—unlike complex problem solving in which the dynamics are learned in an evolving situation (26)—and further iteration once a promising solution is considered—unlike the "aha" moment that leads immediately to a solution in insight problem solving (27, 28). Unlike in traditional model-based or model-free reinforcement learning, in this task people bring rich models of the world that they can quickly tailor to specific, novel problems.

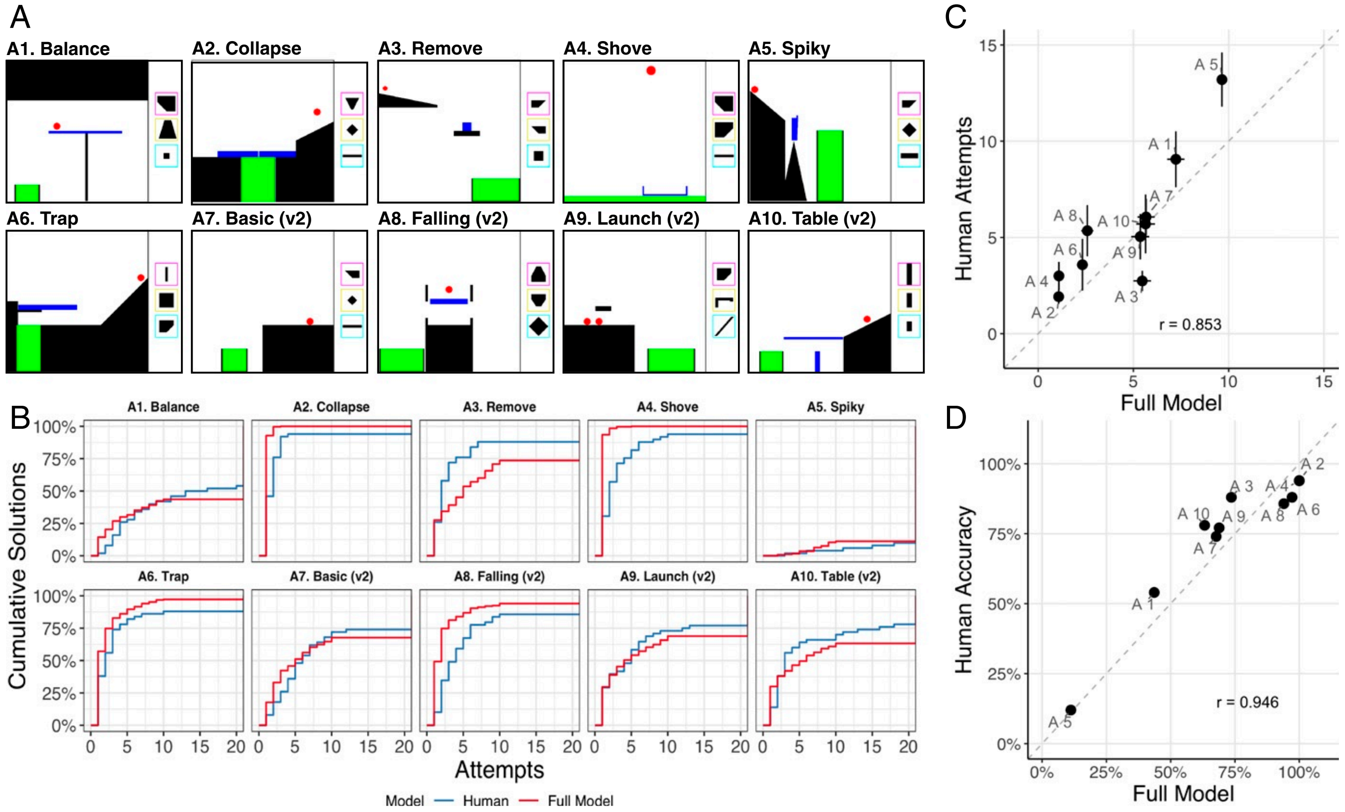


Fig. 7. Results on 10 additional trials. (*A*) Trials used for the second experiment. (*B*) The cumulative solution rate for participants and the SSUP model. (*C*) Comparison of the number of human and model actions by trial. (*D*) Comparison of human and model accuracy on each trial.

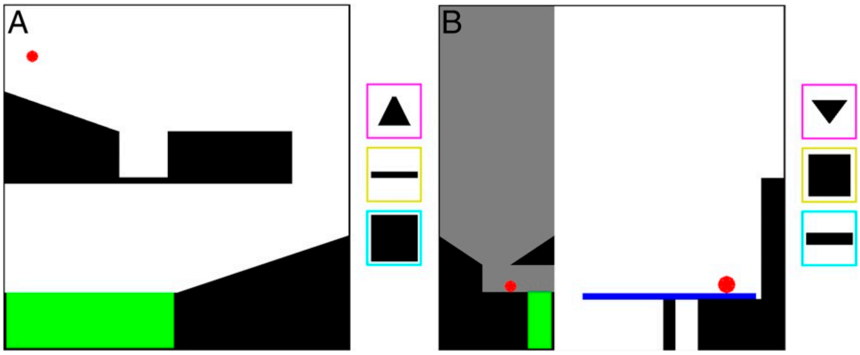


Fig. 8. Two problems that demonstrate limitations of the current model. (*A*) A "suspicious coincidence" that one tool fits perfectly in the hole. (*B*) Creating a "subgoal" to launch the ball onto the other side is useful.

Distilling rich world knowledge to useful task knowledge is necessary for any agent interacting with a complex world. One proposal for how this occurs is "learning by thinking" (29): translating knowledge from one source (internal models of physics) to another, more specific instantiation (a mapping between actions and outcomes on this particular level). We show how SSUP instantiates one example of learning by thinking: by training a policy with data from an internal model. Evidence for this sort of knowledge transfer has been found in people (30, 31), but has focused on simpler discrete settings in which the model and policy are jointly learned.

Virtual Tools as an AI Challenge. In preliminary experiments with model-free reinforcement learning approaches (23), we found limited generalization with inefficient learning across almost all of the Virtual Tools levels (*SI Appendix*, section S5) despite significant experience with related levels.

Based on our human experiments, we believe that model-based approaches will be required to be able to play games like Virtual Tools. Such approaches are becoming increasingly popular in machine learning (32), especially when combined with "learning-to-learn" techniques that can learn to adapt quickly to new tasks (33, 34). Learning these models remains challenging, but approaches that incorporate added structure have excelled in recent years (35, 36). Within the AI and robotics communities, model-based methods are already popular (9, 37, 38). Remaining challenges include how to learn accurate enough models that can be used with raw sensor data (39) and how to handle dynamic environments.

Virtual Tools adds to a growing set of environments that test artificial agents' abilities to predict and reason using physics, such as the concurrently developed physical reasoning (PHYRE) benchmark (40) and others (41–43). In contrast, our focus is on providing problems that people find challenging but intuitive, where solutions are nonobvious and do not rely on precise knowledge of world dynamics. By contributing human data to compare artificial and biological intelligence, we hope to provide a testbed for more human-like artificial agents.

Future Empirical Directions. This work provides an initial foray into formalizing the computational and empirical underpinnings of flexible tool use, but there remains much to study. For instance, we do not find evidence that people learn more about the world, perhaps because there is little benefit to additional precision here. But there are cases where learning the dynamics is clearly helpful (e.g., discovering that an object is abnormally heavy or glued down), and we would expect people to update their physical beliefs in these cases. When and in what ways people update their internal models to support planning is an important area of study.

Children can discover how to use existing objects earlier than they can make novel tools (10), suggesting that tool creation is more challenging than tool use. Yet it is the ability to make and then pass on novel tools that is theorized to drive human culture (44). It is therefore important to understand not just how people use tools, but also how they develop and transmit them, which we can study by expanding the action space of the Virtual Tools game.

Conclusion

Understanding how to flexibly use tools to accomplish our goals is a basic and central cognitive capability. In the Virtual Tools game, we find that people efficiently use tools to solve a wide variety of physical problems. We can explain this rapid trial-and-error learning with the three components of the SSUP framework: rich prior world knowledge, simulation of hypothetical actions, and the ability to learn from both simulations and observed actions. We hope this empirical domain and modeling framework can provide the foundations for future research on this quintessentially human trait: using, making, and reasoning about tools and more generally shaping the physical world to our ends.

ACKNOWLEDGMENTS. We thank Leslie Kaelbling, Roger Levy, Eric Schulz, Jessica Hamrick, and Tom Silver for helpful comments and Mario Belledonne for help with the parameter tuning model. This work was supported by National Science Foundation Science Technology Center Award CCF-1231216; Office of Naval Research Multidisciplinary University Research Initiative (ONR MURI) N00014-13-1-0333; and research grants from ONR, Honda, and Mitsubishi Electric.

1. L. V. Luncz *et al.*, Wild capuchin monkeys adjust stone tools according to changing nut properties. *Sci. Rep.* **6**, 33089 (2016).
2. S. A. Jelbert, A. H. Taylor, L. G. Cheke, N. S. Clayton, R. D. Gray, Using the Aesop's fable paradigm to investigate causal understanding of water displacement by New Caledonian crows. *PloS One* **9**, e92895 (2014).
3. F. Osiurak, A. Badets, Tool use and affordance: Manipulation-based versus reasoning-based approaches. *Psychol. Rev.* **123**, 534–568 (2016).
4. G. A. Orban, F. Caruana, The neural basis of human tool use. *Front. Psychol.* **5**, 310 (2014).
5. G. Goldenberg, J. Spatt, The neural basis of tool use. *Brain* **132**, 1645–1655 (2009).
6. R. W. Shumaker, K. R. Walkup, B. B. Beck, *Animal Tool Behavior: The Use and Manufacture of Tools by Animals* (JHU Press, 2011).
7. A. Xie, F. Ebert, S. Levine, C. Finn, "Improvisation through physical understanding: Using novel objects as tools with visual foresight" in *Robotics: Science and Systems*, A. Bicchi, H. Kress-Gazit, S. Hutchinson, Eds. (RSS Foundation, 2019).
8. B. Baker *et al.*, *Emergent Tool Use from Multi-Agent Autocurricula* (ICLR, 2019).
9. M. Toussaint, K. R. Allen, K. A. Smith, J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning" in *Robotics: Science and Systems*, H. Kress-Gazi, S. Srinivasa, T. Howard, N. Atanasov, Eds. (RSS Foundation, 2018).
10. S. R. Beck, I. A. Apperly, J. Chappell, C. Guthrie, N. Cutting, Making tools isn't child's play. *Cognition* **119**, 301–306 (2011).
11. Orbital Nine Games, Brain it on. https://brainitongame.com/ (2015). Accessed 11 May 2020.

12. A. Newell, H. A. Simon, *Human Problem Solving* (Prentice-Hall, Oxford, England, 1972).
13. R. S. Sutton, Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.* **2**, 160–163 (1991).
14. M. Deisenroth, G. Neumann, J. Peters, A survey on policy search for robotics. *Found. Trends Rob.* **2**, 1–142 (2013).
15. P. W. Battaglia, J. B. Hamrick, J. B. Tenenbaum, Simulation as an engine of physical scene understanding. *Proc. Natl. Acad. Sci. U.S.A.* **110**, 18327–18332 (2013).
16. K. A. Smith, E. Vul, Sources of uncertainty in intuitive physics. *TopiCS* **5**, 185–199 (2013).
17. A. N. Sanborn, V. K. Mansinghka, T. L. Griffiths, Reconciling intuitive physics and Newtonian mechanics for colliding objects. *Psychol. Rev.* **120**, 411–437 (2013).
18. K. J. W. Craik, *The Nature of Explanation* (CUP Archive, 1943).
19. I. Dasgupta, K. A. Smith, E. Schulz, J. B. Tenenbaum, S. J. Gershman, "Learning to act by integrating mental simulations and physical experiments" in *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*, C. Kalish, M. Rau, J. Zhu, T. T. Rogers, Eds. (Cognitive Science Society, 2018), pp. 275–280.
20. T. Gerstenberg, N. D. Goodman, D. A. Lagnado, J. B. Tenenbaum, "How, whether, why: Causal judgments as counterfactual contrasts" in *37th Annual Meeting of the Cognitive Science Society*, D.C. Noelle, R. Dale, A.S. Warlaumont, J. Yoshimi, T. Matlock, C.D. Jennings, P.P. Maglio, Eds. (Cognitive Science Society, 2015), pp. 782–787.
21. K. Juechems, C. Summerfield, Where does value come from? *Trends. Cogn. Sci.* **23**, 836–850 (2019).
22. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).
23. V. Mnih *et al.*, Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
24. J. R. Anderson, Problem solving and learning. *Am. Psychol.* **48**, 35–44 (1993).
25. K. Vaesen, The cognitive bases of human tool use. *Behav. Brain Sci.* **35**, 203–218 (2012).
26. P. A. Frensch, J. Funke, *Complex Problem Solving: The European Perspective* (Psychology Press, 1995).
27. Y. Chu, J. N. MacGregor, Human performance on insight problem solving: A review. *J. Probl. Solving* **3**, 6 (2011).
28. M. L. Gick, K. J. Holyoak, Analogical problem solving. *Cognit. Psychol.* **12**, 306–355 (1980).
29. T. Lombrozo, "Learning by thinking" in science and in everyday life" in *The Scientific Imagination*, P. Godfrey-Smith, A. Levy, Eds. (Oxford University Press, New York, NY, 2018), pp. 230–249.
30. S. J. Gershman, A. B. Markman, A. Ross Otto, Retrospective revaluation in sequential decision making: A tale of two systems. *J. Exp. Psychol. Gen.* **143**, 182–194 (2014).
31. S. J. Gershman, J. Zhou, C. Kommers, Imaginative reinforcement learning: Computational principles and neural mechanisms. *J. Cognit. Neurosci.* **29**, 2103–2113 (2017).
32. T. Weber *et al.*, *Imagination-Augmented Agents for Deep Reinforcement Learning* (NeurIPS, 2017).
33. C. Finn, P. Abbeel, S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks" in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup, Y. W. Teh, Eds. (JMLR.org, 2017), vol. 70, pp. 1126–1135.
34. J. Schmidhuber, J. Zhao, N. N. Schraudolph, "Reinforcement learning with self-modifying policies" in *Learning to Learn* (Springer, 1998), pp. 293–309.
35. M. B. Chang, T. Ullman, A. Torralba, J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics" in *International Conference on Learning Representations*, M. Ranzato, H. Larochelle, O. Vinyals, T. Sainath, Eds. (ICLR, 2016).
36. D. Jayaraman, F. Ebert, A. A. Efros, S. Levine, "Time-agnostic prediction: Predicting predictable video frames" in *International Conference on Learning Representations*, A. Rush, S. Levine, K. Livescu, S. Mohamed, Eds. (ICLR, 2018).
37. L. P. Kaelbling, T. Lozano-Pérez, "Hierarchical task and motion planning in the now" in *IEEE International Conference on Robotics and Automation*, Y. F. Zheng, Ed. (IEEE, 2011), pp. 1470–1477.
38. C. E. Garcia, D. M. Prett, M. Morari, Model predictive control: Theory and practice—A survey. *Automatica* **25**, 335–348 (1989).
39. O. Kroemer, S. Niekum, G. Konidaris, A review of robot learning for manipulation: Challenges, representations, and algorithms. arXiv:1907.03146 (6 July 2019).
40. A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, R. Girshick, *Phyre: A New Benchmark for Physical Reasoning* (NeurIPS, 2019).
41. S. Wenke, D. Saunders, M. Qiu, J. Fleming, Reasoning and generalization in rl: A tool use perspective. arXiv:1907.02050 (3 July 2019).
42. V. Bapst *et al.*, *Structured Agents for Physical Construction* (ICML, 2019).
43. X. Ge, J. H. Lee, J. Renz, P. Zhang, "Hole in one: Using qualitative reasoning for solving hard physical puzzle problems" in *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*, G. A. Kaminka, M. Fox, P. Bouquet, E. Hullermeier, V. Dignum, Eds. (IOS Press, 2016), pp. 1762–1763.
44. M. Tomasello, *The Cultural Origins of Human Cognition* (Harvard University Press, 1999).