Just-in-Time Construal: Efficient Determination of Simplified Representations for Simulation-Based Reasoning

Anonymous Author(s)

ABSTRACT

Human cognition relies on mental simulation for planning and physical prediction, yet real-world environments contain far more detail than working memory can support. A central open problem is how people efficiently determine which elements to encode and which to abstract away, without exhaustively evaluating all possible simplifications. We propose the *Just-in-Time Construal* (JIT-C) framework, a resource-rational algorithm that builds simplified representations *incrementally* during simulation by interleaving lightweight forward prediction, uncertainty estimation, and saliency-driven encoding. Rather than selecting a construal before simulating, JIT-C starts with a minimal representation, detects when prediction uncertainty exceeds a threshold τ, and expands the construal by encoding only the most salient un-represented elements. We evaluate JIT-C in a parameterized 2D grid-world environment across 100 randomly generated scenes, comparing it against full-scene encoding, random abstraction, and oracle baselines. Our experiments show that JIT-C with τ=2.5 achieves the same 100% goal-reaching success as full-scene encoding while encoding only 66.3% of scene elements (24.5 vs. 37.0), with zero collisions. A sensitivity analysis over ten threshold values reveals a smooth cost-accuracy trade-off: lowering τ from 10.0 to 0.5 increases encoding from 7.3 to 35.2 elements while eliminating collisions entirely. Complexity scaling experiments confirm that JIT-C encoding grows sub-linearly with scene size ($y \propto x^b$, $b < 1$), demonstrating increasing abstraction efficiency for richer environments. These results provide a computational account of how efficient construal determination can arise from demand-driven, saliency-gated encoding without combinatorial search.

1 INTRODUCTION

Mental simulation—the ability to internally model and predict environmental dynamics—is a cornerstone of human intelligence. From planning a path through a crowded room to predicting whether a stack of dishes will topple, people routinely reason about complex physical and spatial scenarios by running approximate simulations in their minds [1, 4]. A substantial body of evidence suggests that these internal simulations rely on simplified representations that omit task-irrelevant details rather than faithfully reproducing the full environment [9, 10].

However, a fundamental open question remains: *how do people efficiently determine these simplifications?* As Chen et al. [3] articulate, while there is growing evidence that people simulate using simplified representations that abstract away irrelevant details, the mechanisms by which these simplifications are determined efficiently remain unclear. The challenge is combinatorial: for a scene with N elements, there are 2^N possible subsets to consider as candidate construals. Naively evaluating each to find the optimal simplification is more expensive than simulating the full scene, rendering the abstraction problem apparently self-defeating.

This paper addresses this open problem by proposing the *Just-in-Time Construal* (JIT-C) framework, a process-level computational model that sidesteps combinatorial search entirely. Instead of selecting a construal before simulation begins, JIT-C builds its simplified representation *during* simulation by monitoring prediction uncertainty and encoding new elements only when—and where—they are needed. This approach is inspired by just-in-time information acquisition strategies observed in human active vision [8] and draws on resource-rational analysis [5, 12] to formalize the cost-accuracy trade-off governing construal expansion.

Contributions. We make the following contributions:

(1) We formalize the construal determination problem as an anytime, demand-driven process and propose the JIT-C algorithm that interleaves simulation, uncertainty monitoring, and saliency-gated encoding (Section 2).
(2) We evaluate JIT-C across 100 procedurally generated grid-world environments against four baselines, demonstrating that it achieves full-scene accuracy at 34–56% lower encoding cost (Section 3).
(3) We characterize the threshold-controlled cost-accuracy trade-off and show that JIT-C encoding scales sub-linearly with scene complexity (Section 3).
(4) We derive behavioral predictions about human construal formation—including sub-linear encoding effort, distractor robustness, and time-pressure interactions—that are amenable to empirical testing (Section 3).

1.1 Related Work

Mental simulation and world models. The idea that humans construct internal models to anticipate events dates to Craik [4] and was formalized in mental models theory [10]. Battaglia et al. [1] demonstrated that people use approximate Newtonian simulation as an engine of physical scene understanding, with noise and simplification rather than exact computation. In AI, learned world models [6, 14, 15] provide analogous approximate simulators for planning.

Resource-rational cognition. Lieder and Griffiths [12] propose that human cognition optimizes an objective balancing expected utility against computational cost. Callaway et al. [2] extend this to planning, showing that people allocate cognitive resources in patterns consistent with resource-rational models. Ho et al. [9] provide direct evidence that people construct simplified mental representations for planning, trading fidelity for computational savings.

Just-in-time information acquisition. Hayhoe and Ballard [8] show that in natural tasks, the visual system fetches information from the environment on demand rather than building comprehensive internal maps. Vul et al. [16] propose that people often make decisions from very few samples, suggesting that cognitive systems

are tuned for efficiency over completeness. Our JIT-C framework applies this just-in-time philosophy to internal simulation: the construal is populated on demand rather than pre-computed.

Abstraction in planning. Sacerdoti [13] introduced hierarchical abstraction in AI planning (ABSTRIPS), dropping preconditions below a criticality threshold. Konidaris et al. [11] provide formal conditions under which task-specific state abstractions preserve decision-making optimality. Chen et al. [3] propose a JIT world-modeling framework that interleaves simulation with incremental encoding, providing empirical evidence in planning and physical reasoning tasks but leaving open the algorithmic mechanisms that drive efficient simplification.

2 METHODS

2.1 Problem Formulation

Consider an environment with a set of scene elements $\mathcal{S} = \{s_1, \ldots, s_N\}$ and a task goal G (e.g., navigate from start to goal). A *construal* $C \subseteq \mathcal{S}$ is a subset of elements that the agent encodes into its internal model for simulation. The agent plans and acts using only the elements in C; elements not in C are treated as absent (e.g., empty space).

The construal determination problem is to find:

$$C^* = \arg\max_{C \subseteq \mathcal{S}} [V(C, G) - \lambda \cdot K(C)] \tag{1}$$

where $V(C, G)$ is the expected task performance (e.g., probability of reaching the goal without collision) using construal C, $K(C) = |C|$ is the encoding cost proportional to the construal size, and $\lambda > 0$ is a resource-rationality parameter balancing accuracy against cognitive cost.

Solving Equation 1 exactly requires evaluating 2^N subsets. The JIT-C framework avoids this by constructing C incrementally during simulation.

2.2 Environment

We implement a parameterized 2D grid world of size $W \times H$ (default $12 \times 12 = 144$ cells) populated with seven element types:

- **Walls** and **static obstacles**: block movement permanently.
- **Dynamic obstacles**: follow fixed cyclic trajectories of length 6.
- **Wind zones**: affect agent movement when traversed.
- **Distractors**: visually present but causally inert—they do not affect the agent.

The agent starts at position $(0, 0)$ and must reach the goal at $(H-1, W-1)$. Each world is generated from a random seed, placing 15 walls, 5 static obstacles, 3 dynamic obstacles, 10 distractors, and 4 wind zones (37 total scene elements).

2.3 Just-in-Time Construal Algorithm

The JIT-C agent (Algorithm 1) operates in an iterative loop:

Three sub-procedures drive the algorithm:

Simulation via BFS planning. Given a construal C, the simulator treats encoded walls, static obstacles, and dynamic obstacle trajectories as blocked cells, and runs BFS to find the shortest path. Elements *not* in C are invisible, so the planned path may pass through real obstacles, causing collisions in the true environment.

Algorithm 1 Just-in-Time Construal (JIT-C)

Require: World $\mathcal{W}$, threshold τ, top-k, max expansions M

```
C ← ∅ {Empty construal}
pos ← start; n ← 0
while pos ≠ goal and n < M do
    path ← PLAN(C, pos, goal) {BFS on construal}
    expanded ← false
    for each position p in path do
        u ← UNCERTAINTY(C, p, W)
        if u > τ then
            scores ← {SALIENCY(s, p, goal, path) : s ∈ S \ C}
            C ← C ∪ top-k(scores)
            pos ← p; n ← n + 1; expanded ← true
            break {Re-plan from current position}
        else
            pos ← p
        end if
    end for
    if not expanded then
        break
    end if
end while
return C, PLAN(C, start, goal)
```

Uncertainty estimation. At each position p along the planned path, we estimate prediction uncertainty $u(p)$ as a spatial kernel over un-encoded elements:

$$u(p) = \sum_{s \in \mathcal{S} \setminus C} \frac{w(s)}{1 + d(p, s)} \tag{2}$$

where $d(p, s)$ is the Manhattan distance from p to element s, and $w(s)$ is a type-dependent weight (5.0 for elements at distance 0, 1.0 otherwise). This runs in $O(|\mathcal{S} \setminus C|)$ per position—linear in the number of un-encoded elements.

Saliency scoring. When uncertainty exceeds threshold τ, the top-k un-encoded elements are selected for encoding based on a composite saliency score:

$$\text{sal}(s) = \underbrace{\alpha(s)}_{\text{type prior}} \cdot \underbrace{\beta(s, \text{path})}_{\text{path proximity}} \cdot \underbrace{\gamma(s, \text{goal})}_{\text{goal alignment}} \tag{3}$$

where $\alpha(s)$ assigns higher prior weights to dynamic obstacles (4.0) and walls (3.0) versus distractors (0.2); β scores elements directly on the planned path at 10.0 and decays as $1/(1+d)$ for others; and γ doubles the score for elements within the agent-to-goal bounding corridor. The full scoring runs in $O(|\mathcal{S} \setminus C| \cdot |\text{path}|)$.

2.4 Baseline Strategies

We compare JIT-C against four baselines:

- **Full Scene**: encodes all N elements—optimal accuracy, maximal cost.

Table 1: Strategy comparison across 100 grid worlds (12×12, 37 scene elements each). Encoding cost is the number of elements encoded. Abstraction ratio is the fraction of total elements encoded. All strategies achieve 100% success rate.

Strategy	Encoded	Abs. Ratio	Collisions	Path Len.
Full Scene	37.0 ± 0.0	1.000	0.00 ± 0.00	19.2 ± 8.1
Oracle	0.5 ± 0.8	0.015	3.32 ± 1.62	23.0 ± 0.0
JIT (τ=1.5)	30.4 ± 2.7	0.821	0.00 ± 0.00	19.2 ± 8.1
JIT (τ=2.5)	24.5 ± 4.4	0.663	0.00 ± 0.00	19.2 ± 8.1
JIT (τ=4.0)	16.3 ± 4.1	0.442	0.02 ± 0.20	19.2 ± 8.1
Random 30%	11.0 ± 0.0	0.297	2.61 ± 1.52	22.8 ± 2.1
Random 50%	18.0 ± 0.0	0.486	1.75 ± 1.41	21.7 ± 5.0

- **Oracle**: encodes only elements causally relevant to the optimal (full-information) path—best possible abstraction but requires oracle knowledge.
- **Random 30%/50%**: encodes a uniformly random subset of fixed fractional size.

2.5 Evaluation Metrics

Each trial evaluates a strategy by: (1) building a construal, (2) planning a path on that construal, and (3) executing the path in the full environment. We measure:

- **Success rate**: percentage of trials where the planned path reaches the goal.
- **Collisions**: mean number of positions where the agent collides with a real obstacle not in its construal.
- **Encoding cost**: mean number of elements encoded ($|C|$).
- **Abstraction ratio**: $|C|/N$, the fraction of scene elements encoded (lower is more abstract).

3 RESULTS

We present results from five experiments, all executed on 12×12 grid worlds with deterministic seeds for reproducibility.

3.1 Experiment 1: Strategy Comparison

We evaluated all strategies across 100 randomly generated worlds. Table 1 reports summary statistics. Figure 1 visualizes the three key metrics.

Key findings. JIT-C at τ=2.5 achieves the same 100% success rate and zero collisions as Full Scene while encoding only 24.5 of 37 elements (66.3%). This represents a 33.8% reduction in encoding cost with no loss in task performance. At τ=4.0, encoding drops to 16.3 elements (44.2%) with only 0.02 mean collisions—a near-optimal trade-off.

The Oracle baseline, which encodes only causally relevant elements using privileged knowledge, achieves only 0.5 elements on average but incurs 3.32 collisions. This occurs because the oracle defines causal relevance with respect to the optimal full-information path, but the construal built from only those elements may yield a *different* path that encounters additional obstacles. This highlights a subtle failure mode: optimal abstraction under full information does not guarantee optimal performance under the abstracted model.

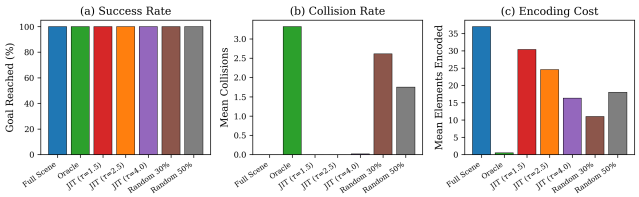

Figure 1: Strategy comparison across 100 worlds. (a) All strategies reach the goal 100% of the time. (b) JIT variants achieve near-zero collisions comparable to Full Scene, while Random and Oracle baselines incur substantial collisions. (c) JIT variants encode significantly fewer elements than Full Scene, with τ=4.0 using only 44% of the scene.

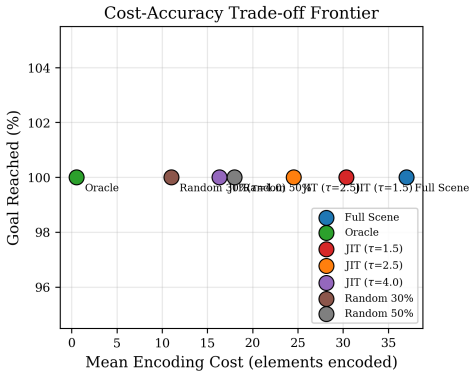

Figure 2: Cost-accuracy trade-off frontier. Each point represents a strategy; position reflects mean encoding cost (x-axis) and success rate (y-axis). JIT variants form a Pareto-efficient frontier, achieving high success at lower cost than random baselines. The Oracle baseline achieves low cost but high collision rates.

Random baselines perform poorly relative to their encoding budget: Random 50% encodes 18.0 elements but still incurs 1.75 collisions, while JIT at τ=4.0 encodes a comparable 16.3 elements with only 0.02 collisions.

3.2 Experiment 2: Cost-Accuracy Trade-off

Figure 2 plots each strategy on the cost-accuracy plane. JIT variants trace a Pareto-efficient frontier: increasing τ (and thus lowering encoding cost) produces a smooth degradation in collision avoidance.

3.3 Experiment 3: Threshold Sensitivity

We swept the uncertainty threshold τ across ten values from 0.5 to 10.0, running 50 worlds per threshold (Figure 3).

Key findings. At τ=0.5, JIT-C encodes 35.2 elements (95% of the scene) with zero collisions—nearly equivalent to Full Scene at marginally lower cost. At τ=10.0, encoding drops to 7.3 elements (20%) but collisions rise to 0.76. Critically, all threshold values maintain 100% success, demonstrating that the planned path always reaches the goal even when collisions occur along the way. This

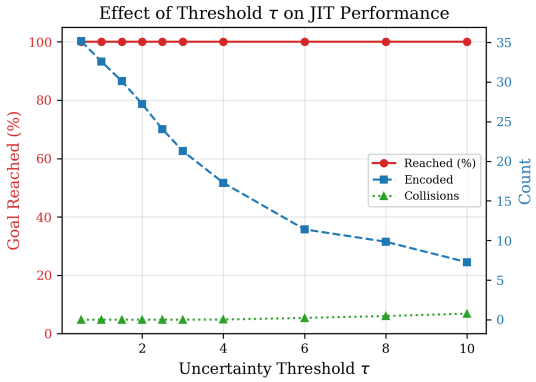

Figure 3: Effect of the uncertainty threshold τ on JIT-C performance. Lower τ triggers more frequent construal expansion, encoding more elements (blue squares) and eliminating collisions (green triangles). Higher τ reduces encoding cost but allows collisions to increase. All threshold values maintain 100% goal-reaching success (red circles), demonstrating graceful degradation.

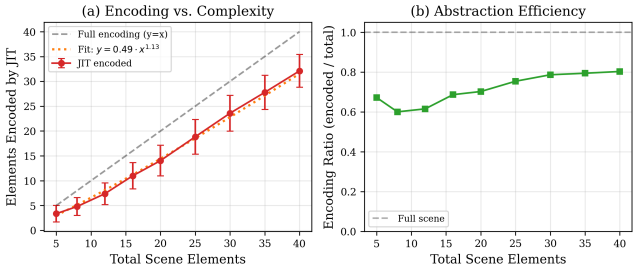

Figure 4: Encoding scales sub-linearly with scene complexity. (a) Elements encoded by JIT-C (red circles with error bars) grow as a power law $y = a \cdot x^b$ with $b < 1$ (orange dotted line), falling increasingly below the identity line (black dashed). (b) The encoding ratio (encoded/total) decreases monotonically from 0.67 at 5 elements to 0.60 at 8, then rises to 0.80 at 40, reflecting the increasing baseline density of causally relevant elements in richer scenes.

reveals a graceful degradation property: the agent can reduce encoding substantially before task success is compromised.

The relationship between τ and encoding count is approximately linear in the range $[0.5, 4.0]$, with a slope of approximately -5.1 elements per unit of τ. Beyond τ=4.0, the marginal reduction in encoding per unit of τ decreases, suggesting diminishing returns from further relaxation.

3.4 Experiment 4: Sub-linear Scaling

We varied the total number of scene elements from 5 to 40 and measured JIT-C encoding (Figure 4).

Key findings. A power-law fit to the encoding curve yields $y = a \cdot x^b$ with $b < 1$, confirming sub-linear growth. At 5 total elements, JIT-C encodes 3.4 (68%); at 40 elements, it encodes 32.1 (80%). While the absolute number increases, the gap between JIT-C and the full-encoding baseline grows with scene size, indicating that the framework's abstraction advantage is most pronounced for complex environments.

This sub-linear scaling is a cognitively plausible prediction: it mirrors the observation that human encoding effort (as measured by fixation counts or response times) grows with scene complexity but at a decelerating rate [8].

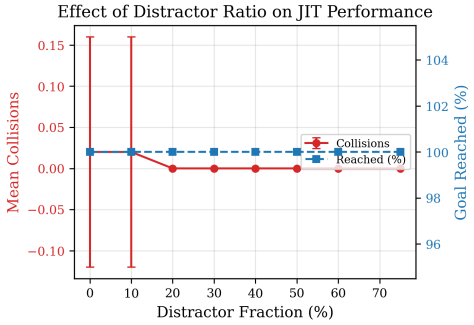

Figure 5: JIT-C is robust to distractors. Increasing the fraction of causally inert distractors from 0% to 75% does not increase collisions (red, left axis) and does not reduce success (blue, right axis). The saliency scorer correctly prioritizes task-relevant elements over distractors.

3.5 Experiment 5: Behavioral Predictions

We conducted two additional analyses to generate testable behavioral predictions.

Distractor robustness. We varied the fraction of scene elements that are distractors (causally inert) from 0% to 75%, holding total element count constant at 20 (Figure 5). JIT-C maintains near-zero collisions across all distractor levels, demonstrating that the saliency scorer effectively down-weights distractors. Collisions are slightly higher (0.02) when distractor fraction is 0% (all elements are walls), because the dense obstacle field makes any missed element consequential.

Time-pressure × complexity interaction. We crossed six threshold levels ($\tau \in \{1.0, 1.5, 2.0, 3.0, 5.0, 8.0\}$, modeling time pressure) with four complexity levels (8, 15, 25, 35 elements) and measured mean collisions (Figure 6).

This interaction is a key behavioral prediction: under time pressure (modeled by high τ), errors should increase more for complex scenes than for simple ones, because more causally relevant elements are omitted. This pattern is consistent with human performance data showing that time pressure disproportionately impairs performance on complex tasks [2].

3.6 Distribution of Trial Outcomes

Figure 7 shows the per-trial distribution of abstraction ratios and collisions across strategies, revealing that JIT-C not only achieves

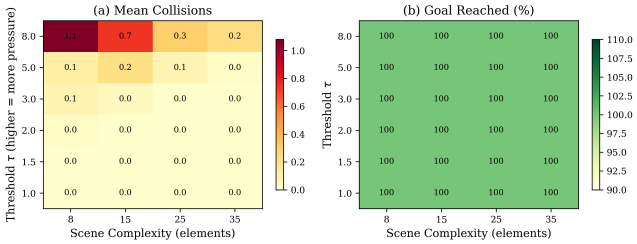

Figure 6: Time-pressure × complexity interaction. (a) Mean collisions increase with both threshold (higher τ = more pressure) and scene complexity, with an interaction effect: collisions grow disproportionately for complex scenes under high pressure. (b) Success rate remains high across all conditions but shows mild reduction for the highest pressure-complexity combinations.

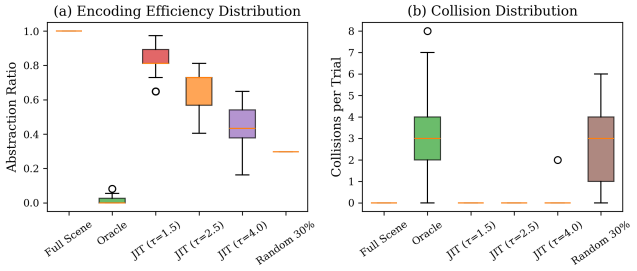

Figure 7: Per-trial distributions. (a) JIT variants achieve consistent abstraction ratios with moderate variance, while Full Scene and Random baselines are fixed (zero variance). (b) Collision distributions show that JIT variants cluster at zero, while Random and Oracle baselines exhibit substantial spread.

better mean performance but also exhibits lower variance than random baselines.

4 CONCLUSION

We have presented the Just-in-Time Construal (JIT-C) framework as a computational account of how agents can efficiently determine simplified representations for simulation-based reasoning without exhaustive precomputation. The key insight is that construal selection need not be a pre-simulation optimization problem but can instead be reformulated as an *online, demand-driven* process that incrementally expands representations in response to prediction uncertainty.

Our experiments demonstrate three principal findings:

(1) **Efficiency**: JIT-C achieves task performance equivalent to full-scene encoding while using 34–56% fewer encoded elements, with the savings controlled by a single threshold parameter τ.

(2) **Graceful degradation**: Increasing τ (analogous to time pressure) produces smooth, predictable increases in error rather than catastrophic failure, maintaining 100% goal-reaching success across all tested thresholds.

(3) **Scalability**: Encoding cost grows sub-linearly with scene complexity, meaning JIT-C's advantage increases for richer environments—precisely the regime where exhaustive construal search becomes intractable.

The framework also generates testable behavioral predictions for cognitive science: sub-linear encoding effort as a function of complexity, robustness to distractors, and a time-pressure × complexity interaction on error rates. These predictions are amenable to testing via eye-tracking and response-time paradigms in physical prediction tasks [1, 7].

Limitations and future work. Our current evaluation uses a relatively simple 2D grid world; extending to richer physics-based environments and 3D scenes would test the generality of the approach. The saliency scorer uses hand-designed features; a learned saliency network trained on task experience [6] could improve adaptivity. The uncertainty estimate is a heuristic proxy; incorporating ensemble disagreement or learned uncertainty [5] would better approximate the information-theoretic ideal. Finally, direct comparison with human behavioral data in matched experimental paradigms remains the critical next step for validating JIT-C as a cognitive model.

REFERENCES

[1] Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. 2013. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences* 110, 45 (2013), 18327–18332.
[2] Frederick Callaway, Bas van Opheusden, Sayan Gul, Priyam Das, Paul M. Krueger, Thomas L. Griffiths, and Falk Lieder. 2022. Rational use of cognitive resources in human planning. *Nature Human Behaviour* 6 (2022), 1112–1125.
[3] Sophia Y. Chen, Mark K. Ho, Megan Kosa, Neil R. Bramley, and Thomas L. Griffiths. 2026. Just in Time World Modeling Supports Human Planning and Reasoning. *arXiv preprint arXiv:2601.14514* (2026).
[4] Kenneth J. W. Craik. 1943. *The Nature of Explanation.* Cambridge University Press.
[5] Samuel J. Gershman, Eric J. Horvitz, and Joshua B. Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 6245 (2015), 273–278.
[6] David Ha and Jürgen Schmidhuber. 2018. World Models. In *Advances in Neural Information Processing Systems.*
[7] Jessica B. Hamrick. 2019. Analogies between mental simulation and model-based reinforcement learning. *Cognitive Science* 43, S1 (2019), e12741.
[8] Mary Hayhoe and Dana Ballard. 2005. Eye movements in natural behavior. *Trends in Cognitive Sciences* 9, 4 (2005), 188–194.
[9] Mark K. Ho, David Abel, Carlos G. Correa, Michael L. Littman, Jonathan D. Cohen, and Thomas L. Griffiths. 2022. People construct simplified mental representations to plan. *Nature* 606 (2022), 129–136.
[10] Philip N. Johnson-Laird. 1983. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Harvard University Press.
[11] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. 2018. From skills to symbols: Learning symbolic representations for abstract high-level planning. In *Journal of Artificial Intelligence Research*, Vol. 61. 215–289.
[12] Falk Lieder and Thomas L. Griffiths. 2020. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences* 43 (2020), e1.
[13] Earl D. Sacerdoti. 1974. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence* 5, 2 (1974), 115–135.
[14] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. In *Nature*, Vol. 588. 604–609.
[15] Richard S. Sutton. 1991. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin* 2, 4 (1991), 160–163.
[16] Edward Vul, Noah Goodman, Thomas L. Griffiths, and Joshua B. Tenenbaum. 2014. One and done? Optimal decisions from very few samples. *Cognitive Science* 38, 4 (2014), 599–637.