Beyond Chain-of-Thought: Test-Time Compute Scaling for Deliberative Large Language Models

Anonymous

October 25, 2025

Abstract

Scaling the *training* compute of large language models (LLMs) has yielded remarkable generality, but many tasks still require adaptive, structured *test-time* computation to support extended deliberation, search, and self-evaluation. Existing practices—such as chain-of-thought (CoT), parallel self-consistency sampling, and simple tree-of-thought search—offer important first steps but leave open a systematic theory of how and when to allocate additional inference-time computation and what algorithms achieve the best return on cognitive effort. This paper proposes a unifying formalization of *budgeted deliberation*, defines the *Budget–Performance Frontier* for test-time compute, and introduces a family of theoretically grounded algorithms that go substantially beyond linear CoT and naive parallelization. Drawing on resource-rational analysis, decision theory, submodular optimization, bandit indices, risk-sensitive control, and insights from cognitive psychology and neuroscience, we develop: Index-Guided Deliberation across competing reasoning threads; Risk-Sensitive Monte Carlo Tree-of-Thought; Abduction–Deduction–Refutation loops driven by information gain; Counterfactual Self-Consistency as message passing; Market-based Decompose–Recompose allocation; Branch-and-Bound with LLM-admissible heuristics; Annealed Populations of Thoughts; and Deliberative External Memory with ephemeral indexing. We provide conditions under which these methods enjoy anytime behavior, approximate optimality, or risk control, and we propose evaluation criteria that quantify the efficiency of test-time compute. No empirical results or program code are required; the contributions are conceptual and theoretical.

1 Introduction

Recent LLMs demonstrate impressive problem-solving abilities, especially when prompted to "think aloud" [1] or when multiple samples are reconciled via self-consistency [2]. However, naive increases in the number of tokens or parallel samples often yield diminishing returns, lack principled stopping rules, and can entrench spurious reasoning. We ask: *How should inference-time computation be allocated to maximize expected solution quality per unit of cognitive effort?* And: *What algorithms move us beyond linear chain construction or blind parallelism toward structured, risk-aware deliberation?*

We propose a normative framework that treats test-time reasoning as a metareasoning problem [4, 5, 6], where actions include proposing steps, branching, verifying, retrieving, and deciding to stop. Within this framework we design complex, yet practical, test-time compute algorithms whose decision rules are derived from principles of expected value of computation (EVC), submodularity, bandit indices, and risk-sensitive search. To ground these algorithms psychologically and neurally, we connect to dual-process theories [9], working memory and cognitive control [10, 11], and hippocampal replay as sample-based planning [12].

Contributions.

1. A formal model of *budgeted deliberation* with compute costs at the granularity of cognitive actions.
2. The *Budget–Performance Frontier* (BPF): a target curve for evaluation and a lens to analyze diminishing returns.
3. Ten test-time algorithms with theory-backed properties: Index-Guided Deliberation (IGD), Risk-Sensitive Monte Carlo Tree-of-Thought (RS-MCTT), Abduction–Deduction–Refutation loops (ADR), Counterfactual Self-Consistency (CSC), Decompose–Recompose via Subproblem Markets (DRSM), Branch-and-Bound with LLM Heuristics (BB-LLM), Annealed Population of Thoughts (APT), Deliberative External Memory with Ephemeral Indexing (DEMI), Probabilistic Self-Verification (PSV), and Dual-Process Gating (DPG).
4. Approximation guarantees and risk bounds under natural assumptions (submodularity, concentration, admissible heuristics).
5. A set of evaluation metrics (frontier area, marginal value-of-compute, reliability at fixed risk) to compare methods without committing to specific datasets or implementations.

2 Budgeted deliberation: a formalization

Let x denote an instance drawn from unknown distribution $\mathcal{D}$. An LLM with parameters θ interacts with a *deliberation environment* by producing tokens and invoking *cognitive actions* $a \in \mathcal{A}$ such as:

PROPOSE, BRANCH, CRITIQUE, VERIFY, RETRIEVE, REVISE, DECIDE.

Each action incurs a compute cost $c(a) \geq 0$ (e.g., tokens, wall-clock, FLOPs) and transforms the deliberation state $s_t = (x, h_t)$ where h_t is the transcript of thoughts, branches, and external-memory writes up to time t. At any step, the agent may output a final answer y and stop.

Definition 1 (Deliberation policy)**.** A *deliberation policy* π maps states to distributions over actions (including DECIDE). Given a utility $U(y; x) \in [0, 1]$ measuring solution quality, a compute penalty $\lambda \geq 0$, and a costed trajectory $\tau = (a_1, \ldots, a_T, y)$ with cost $C(\tau) = \sum_{t=1}^{T} c(a_t)$, the *expected net utility* is

$$J(\pi; \lambda) \;=\; \mathbb{E}_{x\sim\mathcal{D},\,\tau\sim\pi}\big[U(y; x) - \lambda\, C(\tau)\big]. \tag{1}$$

Equivalently, with a hard budget B, we define the *Budget–Performance Frontier*

$$P(B) \;=\; \sup_{\pi:\,\mathbb{E}[C(\tau)]\leq B} \mathbb{E}\big[U(y; x)\big]. \tag{2}$$

The function P is nondecreasing in B; beyond trivial monotonicity, understanding its curvature is key for principled scheduling and stopping.

Assumption 1 (Diminishing returns)**.** *For a given x, the expected improvement from adding a set S of cognitive micro-actions exhibits diminishing returns: the set function $F(S) = \mathbb{E}[U(y; x) \mid S \text{ executed}]$ is monotone submodular.*

Proposition 1 (Concavity of the BPF under submodularity)**.** *Under Assumption 1 and a randomized micro-action cost model with bounded variance, the* smoothed *frontier P is concave in B to first order. Consequently, the marginal value-of-compute $\partial P/\partial B$ is nonincreasing.*

Proof sketch. View deliberation as selecting costly micro-actions from a ground set $\mathcal{S}$ with costs c_s. The relaxation to fractional selection leads to a continuous knapsack with a concave closure when F is submodular (via the multilinear extension). Standard arguments then imply diminishing marginal returns in the budget [7]. □

Expected value of computation (EVC). At state s_t, the *EVC* of an action a is

$$\mathrm{EVC}(a \mid s_t) \;=\; \mathbb{E}[\,\Delta U \mid s_t, a\,] \;-\; \lambda\, c(a), \qquad \Delta U := U(y^\star_{t+1}; x) - U(y^\star_t; x), \tag{3}$$

where $y^\star_t$ is the best-so-far answer. A greedy-EVC policy that applies the highest positive-EVC action until all EVCs are nonpositive is an *anytime* procedure.

Theorem 1 (Greedy near-optimality)**.** *Under Assumption 1, unit-cost micro-actions, and perfect local EVC estimates, the greedy-EVC policy achieves at least a* $(1 - 1/e)$ *fraction of the optimal net utility at any budget* B*.*

Proof idea. Map to monotone submodular maximization under a cardinality constraint and apply the Nemhauser bound [7]. The per-step EVC ordering is equivalent to the marginal gains ordering. □

3 Beyond CoT: ten algorithms for deliberative test-time compute

We now describe algorithms that elevate test-time computation from linear chains or naive parallelism to structured, adaptive, and risk-aware deliberation.

3.1 Index-Guided Deliberation (IGD)

Maintain m *reasoning threads* (e.g., distinct decompositions, hypotheses, or proof directions). Let thread i at depth ℓ have a posterior over incremental improvements $R_{i,\ell}$ (estimated from self-evaluations, critics, or heuristics). Define a discount $\gamma \in (0, 1]$ and per-step cost c_i. The *index* for thread i is

$$\tau_i \;=\; \sup_{n \geq 1} \frac{\mathbb{E}\big[\sum_{\ell=1}^{n} \gamma^{\ell-1} R_{i,\ell}\big]}{\mathbb{E}[\sum_{\ell=1}^{n} \gamma^{\ell-1} c_i]}. \tag{4}$$

At each step, extend the thread with maximal τ_i if $\tau_i > \lambda$; otherwise stop.

Proposition 2 (Approximate optimality of index policy)**.** *If (i) thread improvements are conditionally independent across threads given history, (ii) each thread's improvement process is stochastically nonincreasing, and (iii) costs are stationary, then the IGD policy is optimal for the relaxed, per-thread discounted objective; for the undiscounted budgeted objective, it is a near-optimal heuristic with strong empirical support in multi-armed metareasoning [4, 6].*

3.2 Risk-Sensitive Monte Carlo Tree-of-Thought (RS-MCTT)

Generalizing tree-of-thought search [3], consider a thought tree with node value random variable $X(s)$ induced by stochastic rollouts. Replace risk-neutral evaluation $\mathbb{E}[X]$ with an entropic risk measure

$$\rho_\eta(X) \;=\; \frac{1}{\eta} \log \mathbb{E}[e^{\eta X}], \quad \eta < 0 \text{ (risk-averse)}, \quad \eta > 0 \text{ (risk-seeking)}. \tag{5}$$

Selection uses an upper confidence functional with risk:

$$\textsc{Select}\ a = \arg\max_a \; \rho_\eta\big(\widehat{Q}(s,a)\big) \;+\; \kappa \sqrt{\tfrac{\log N(s)}{N(s,a)}}.$$

Compute *scales* via depth-dependent $\eta(d)$ and exploration constant $\kappa(d)$; anneal toward risk aversion near decision depth to suppress brittle reasoning. RS-MCTT provides tunable reliability at fixed compute.

3.3 Abduction–Deduction–Refutation (ADR) loops

Abduction: propose hypotheses H explaining the instance; *Deduction:* derive predictions or subgoals; *Refutation:* attempt to falsify H via internal checks or counterexamples. Allocate compute by *information gain*:

$$\Delta_{\mathrm{IG}}(a \mid s_t) \;=\; I\big(Y; Z_a \,\big|\, s_t\big) - \lambda\, c(a), \tag{6}$$

where Z_a is the observation exposed by action a (e.g., a test calculation or sub-proof). Choose the action with maximal positive Δ_{IG}. This embodies Popperian falsification and Bayesian surprise in a single scheduling principle.

3.4 Counterfactual Self-Consistency (CSC)

Self-consistency averages over parallel chains [2]. CSC augments chains with *counterfactual constraints* (e.g., algebraic relations, logical implications, invariants) and performs message passing among chains, downweighting chains that violate shared constraints. Let k be the number of chains and suppose correct chains have advantage $\delta = \mathbb{P}(\text{vote correct}) - \frac{1}{2}$. If counterfactual checks reject an ϵ fraction of incorrect chains while rejecting at most ϵ' of correct ones, then for majority vote $\hat{y}$:

$$\mathbb{P}\big(\hat{y} \neq y^\star\big) \;\leq\; \exp\Big(-2k\,(\delta - \tfrac{\epsilon-\epsilon'}{2})^2\Big), \tag{7}$$

by a Chernoff–Hoeffding argument [8]. Thus modest-quality constraints can exponentially tighten self-consistency.

3.5 Decompose–Recompose via Subproblem Markets (DRSM)

Let a problem decompose into subproblems $p \in \mathcal{P}$ linked by a coupling penalty $g(\{y_p\})$ capturing consistency or resource coupling. Introduce *compute prices* $\mu_p \geq 0$ and define the Lagrangian

$$\mathcal{L}(\{y_p\}, \mu) \;=\; \sum_p \big(U_p(y_p; x) - \mu_p C_p\big) \;-\; g(\{y_p\}), \tag{8}$$

where C_p is compute allocated to p. A myopic *auction* allocates the next token of compute to the subproblem with highest *surplus* $S_p = \partial \mathbb{E}[U_p]/\partial C_p - \mu_p$. Updating prices to enforce global consistency (dual ascent) yields an anytime, market-based scheduler that concentrates compute where marginal gains are largest, while softly penalizing incoherence.

3.6 Branch-and-Bound with LLM Heuristics (BB-LLM)

For tasks admitting verifiable objectives (e.g., satisfiability, arithmetic, program synthesis with testable specs), define an admissible heuristic $\hat{V}(s)$ that *upper-bounds* the attainable utility from state s. If the LLM self-evaluation $\tilde{V}(s)$ is stochastically *downbiased*, i.e., $\mathbb{E}[\tilde{V}(s)] \leq V^\star(s)$, then with high probability we can form an admissible bound via a concentration correction. Branch on promising subproblems, prune when $\hat{V}(s)$ falls below the incumbent.

Proposition 3 (Anytime guarantee)**.** *If the heuristic is admissible and branching is finite, BB-LLM returns an optimal solution under unbounded budget and an ε-optimal solution whenever the best unexpanded node is within ε of the incumbent bound.*

3.7 Annealed Population of Thoughts (APT)

Maintain a population $\{(\tau_i, s_i)\}_{i=1}^{M}$ of candidate thought trajectories with scores S_i (e.g., calibrated self-evaluation, verifier outputs). At each epoch: (i) *mutate* by extending or perturbing trajectories; (ii) *recombine* by merging prefixes/suffixes; (iii) *reweight* via a Boltzmann transform

$$w_i \;\propto\; \exp\big(\beta\, S_i\big), \tag{9}$$

with an *annealing* schedule $\beta : 0 \to \beta_{\max}$ that gradually concentrates on high-quality reasoning while preserving diversity early on. APT unifies self-consistency, beam search, and evolutionary strategies into a single test-time compute mechanism with explicit diversity control.

3.8 Deliberative External Memory with Ephemeral Indexing (DEMI)

Augment the LLM with an *ephemeral memory* M scoped to the instance. Actions include $\textsc{Write}(k, v)$, $\textsc{Retrieve}(q)$, and $\textsc{Rewrite}(k, v')$, each incurring costs for serialization and attention. Let $G(M)$ be the expected utility gain from memory state M. If G is submodular in the multiset of writes (diminishing returns of additional notes), a greedy memory policy that writes the note with largest marginal gain per cost is $(1 - 1/e)$-approximate. For retrieval, define locality-sensitive queries and allocate compute to index-building only when anticipated future retrieval saves more than its cost (amortized EVC).

3.9 Probabilistic Self-Verification (PSV)

When weak verifiers are available (e.g., unit checks, invariants, dimensional analysis, type consistency), integrate them probabilistically into the decision rule. Suppose a candidate y passes m conditionally independent checks with likelihood ratios ℓ_j. Then the posterior odds of correctness update multiplicatively:

$$\frac{\mathbb{P}(y \text{ correct} \mid \text{checks})}{\mathbb{P}(y \text{ incorrect} \mid \text{checks})} \;=\; \frac{\mathbb{P}(y \text{ correct})}{\mathbb{P}(y \text{ incorrect})} \cdot \prod_{j=1}^{m} \ell_j. \tag{10}$$

Schedule additional checks if the expected log-odds increment exceeds λ times the check cost. PSV gives calibrated, compute-aware stopping for verification-heavy domains.

3.10 Dual-Process Gating (DPG)

Inspired by dual-process theories [9], *System 1* produces fast, low-compute answers with a confidence proxy q (e.g., entropy, margin, or self-evaluation), while *System 2* engages structured algorithms (IGD, RS-MCTT, ADR). Define a gating threshold $\theta(B)$ (possibly budget-dependent) and escalate iff $q < \theta(B)$. If q is a surrogate for the Bayes error with Lipschitz calibration, one obtains risk-control bounds of the form

$$\mathbb{P}(\text{error} \wedge \text{no escalate}) \;\le\; \mathbb{E}[\mathbb{I}\{q \ge \theta\} \cdot \mathrm{err}(q)] \;\le\; \varepsilon(\theta),$$

yielding target reliability at minimal expected compute.

4 Stopping rules and compute allocation

4.1 Single-thread optimal stopping

Consider a single reasoning thread whose incremental improvement R_t is a supermartingale with $\mathbb{E}[R_{t+1} \mid \mathcal{F}_t] \le \mathbb{E}[R_t \mid \mathcal{F}_t]$ and bounded increments. The optimal stopping time under penalty λ satisfies

$$\text{Stop at time } \tau^\star \;=\; \inf\{t \ge 1 : \mathbb{E}[R_{t+1} \mid \mathcal{F}_t] \le \lambda\, c\}.$$

Proposition 4 (Risk-aware threshold)**.** *If R_t is sub-Gaussian with proxy σ_t^2, then a risk-averse EVC threshold is $\mathbb{E}[R_{t+1} \mid \mathcal{F}_t] - \alpha\sigma_t \le \lambda c$, with α chosen for the desired tail probability of over-spending.*

4.2 Multi-thread scheduling as Whittle-style indexability

For multiple threads with decoupled dynamics under a Lagrange relaxation, *indexability* obtains when increasing the compute price λ monotonically shrinks the active set of states; IGD is then optimal for the relaxed problem. While exact indexability is task-dependent, monotone posteriors and nonincreasing returns suffice in many reasoning domains [6].

5 Psychology and neuroscience links

The proposed algorithms mirror aspects of human deliberation. IGD parallels *rational metareasoning* and selective attention to promising lines of thought [6]. RS-MCTT resonates with *risk-sensitive control* and flexible exploration–exploitation. ADR formalizes *hypothesis testing* and the role of *falsification*. APT reflects population-based cognition and *global workspace* accumulation of evidence [13]. DEMI aligns with *working memory* and gating by prefrontal mechanisms [10, 11]. Replay-like search in RS-MCTT and APT connects to hippocampal *preplay* and *vicarious trial and error* [12].

6 Evaluation without experiments: what to measure

To compare test-time compute algorithms abstractly:

- **Frontier area (FA):** $\int_0^{B_{\max}} P(B)\, dB$ approximated by discrete budgets; larger area indicates better use of compute across scales.
- **Marginal value-of-compute (MVC):** $\Delta P/\Delta B$ near operational budgets; steeper MVC is preferable.
- **Reliability at fixed budget:** Error or risk under RS-MCTT/DPG-type controls.
- **Diversity efficiency:** Quality gain per *independent* reasoning dimension (APT/CSC).
- **Verifier yield:** Utility gain per unit of verification compute (PSV/BB-LLM).
- **Memory return:** Improvement per memory operation (DEMI).

Recommended evaluation protocol. For comparability with recent reports on test-time compute, we recommend: (i) reporting BPF curves over budgets $B \in \{0.25, 0.5, 1, 2, 4\}\times$ a task-specific baseline; (ii) including at least one arithmetic task (e.g., GSM8K), one math task (e.g., MATH), one code task (e.g., HumanEval with unit tests), and one symbolic/planning domain; (iii) baselines of greedy, best-of-n, self-consistency, and a ToT variant; and (iv) explicit accounting of tokenized compute (prompt + generation), wall-clock on a fixed accelerator, and any gating policy (DPG) used. These practice recommendations make BPF and marginal value-of-compute directly comparable to recent compute-optimal inference studies and TTC surveys.

7 Related work

Prompted reasoning traces such as CoT [1] and self-consistency [2] improve LLM problem-solving but lack formal compute allocation. Tree-of-thought search [3] and debate-style multi-agent prompting [14] add structure but often use fixed schedules. Our framework builds on metareasoning [4, 5, 6], submodular optimization [7], risk-sensitive control, and bandit indices, and connects them to LLM inference.

8 Limitations and societal considerations

The theory presumes access to calibrated self-evaluations and verifiers; miscalibration may misallocate compute or prematurely stop. Risk-sensitive search mitigates hallucination but may be conservative on creative tasks. Market-like and multi-agent procedures (DRSM, APT) can amplify biases if scoring functions are biased. Responsible deployment should pair compute-scaling with audit trails, verifiers, and explicit stopping policies.

9 Conclusion

We introduced a unifying formalism for test-time compute scaling and ten algorithmic blueprints that transcend linear chains and naive parallelism. The central message is that *where* and *when* to spend computation can be decided using principled quantities (EVC, indices, risk measures, information gain, submodular gains), yielding anytime procedures with theoretical backing and cognitive plausibility. Future work can instantiate these designs across domains and refine calibrations for self-evaluation and verification.

10 Scope and positioning relative to concurrent work

Recent work studies inference/test-time scaling empirically and surveys the area. Wu *et al.* analyze inference scaling laws and compute-optimal inference, showing Pareto trade-offs between extra tokens/strategies and model size [16]. Ji *et al.* survey test-time compute from System-1 adjustments to System-2 deliberate reasoning [17]. Guan *et al.* propose *Deliberative Alignment*, reasoning explicitly over safety specifications before answering [18]; Guo *et al.* introduce *Reward Reasoning Models* that leverage deliberate test-time compute for reward modeling [19]. Our contribution is complementary: we provide a formal budgeted decision-theoretic substrate (BPF/EVC/indices/risk) and theory-backed algorithmic blueprints that can host these approaches and make their compute–performance trade-offs explicit.

11 Threats to validity and reproducibility

Our guarantees rely on assumptions that may be violated in practice (e.g., independence across threads for IGD, indexability under Whittle-style relaxations, calibration of self-evaluation signals, and approximate submodularity of gains). We recommend: (i) reporting calibration diagnostics for self-scores and verifiers; (ii) ablation of risk schedules in RS-MCTT; (iii) stress tests where action costs are stochastic; and (iv) disclosing hardware, batching, and gating policies. Sharing small, reference implementations (pseudo-code suffices) will improve reproducibility and comparability.

Appendix: Additional sketches

Submodular memory gains (DEMI). Let notes be elements of a ground set $\mathcal{S}$ and $G : 2^{\mathcal{S}} \to \mathbb{R}_+$ be the expected utility from a set of notes under a fixed retrieval policy. If G is monotone submodular and note costs are unit or bounded, greedy selection achieves $(1 - 1/e)$ of the optimal G under equal-cost budgets; with heterogeneous costs, cost-benefit greedy achieves the same factor under the standard knapsack relaxation.

CSC bound derivation. Let $Z_i \in \{0, 1\}$ indicate whether chain i is correct. After filtering, each incorrect chain is kept with prob. $\leq 1 - \epsilon$, each correct chain with prob. $\geq 1 - \epsilon'$. The effective margin becomes $\delta' = \delta - (\epsilon - \epsilon')/2$. Hoeffding's inequality on the filtered Bernoulli sum yields Eq. (7).

References

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-Thought prompting elicits reasoning in large language models. *arXiv:2201.11903*, 2022.

[2] Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*, 2023.

[3] Shunyu Yao, Dian Yu, Jeffrey Zhao, et al. Tree of Thoughts: Deliberate problem solving with large language models. *arXiv:2305.10601*, 2023.

[4] Stuart Russell and Eric Wefald. *Do the Right Thing: Studies in Limited Rationality*. MIT Press, 1991.

[5] Eric J. Horvitz. Reasoning under varying and uncertain resource constraints. In *AAAI Workshop on Limited Rationality*, 1989.

[6] Falk Lieder and Thomas L. Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1, 2020.

[7] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[8] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[9] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[10] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[11] Samuel J. Gershman and Nathaniel D. Daw. Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, 68:101–128, 2017.

[12] Brad E. Pfeiffer and David J. Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79, 2013.

[13] Stanislas Dehaene and Jean-Pierre Changeux. Experimental and theoretical approaches to conscious processing. *Neuron*, 70(2):200–227, 2011.

[14] Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv:1805.00899*, 2018.

[15] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[16] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models. *arXiv:2408.00724*, 2024.

[17] Yixin Ji, Juntao Li, Yang Xiang, Hai Ye, Kaixin Wu, Kai Yao, Jia Xu, Linjian Mo, and Min Zhang. A Survey of Test-Time Compute: From Intuitive Inference to Deliberate Reasoning. *arXiv:2501.02497*, 2025.

[18] Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. Deliberative Alignment: Reasoning Enables Safer Language Models. *arXiv:2412.16339*, 2024.

[19] Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward Reasoning Model. *arXiv:2505.14674*, 2025.

[20] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo Planning. In *ECML*, 2006.

[21] Ronald A. Howard and James E. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.

[22] John C. Gittins. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society: Series B*, 41(2):148–164, 1979.

[23] Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298, 1988.

[24] Laurent Itti and Pierre F. Baldi. Bayesian Surprise Attracts Human Attention. In *NIPS*, 2006; see also *Vision Research*, 49(10):1295–1306, 2009.