LATERAL TREE-OF-THOUGHTS SURPASSES TOT BY INCORPORATING LOGICALLY-CONSISTENT, LOW-UTILITY CANDIDATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern deployments increasingly budget *large test-time compute*—thousands of tokens or many node expansions—to improve reliability. When *structured search* (e.g., ToT/MCTS-style controllers) is run under such frontier-like conditions, two effects intensify: *breadth saturation*, where additional samples at a node mostly yield near-duplicates so width stops growing; and *myopia*, where early, noisy utility undervalues branches whose payoff appears only after a few more steps, so they are pruned too soon. We introduce **Lateral Tree-of-Thoughts (LToT)**, a search-time controller that explicitly separates the frontier into *mainlines—high-utility* candidates used for exploitation—and *laterals—logically consistent, initially low-utility* candidates that merit short, cheap probes before judgment. LToT explores laterals via *Lateral Racing with Short-Circuit (LR–SC)*, a budgeted race that spreads tiny probes across a very wide lateral set, culls aggressively, and immediately promotes a branch once it demonstrably clears the mainline bar; mainlines are kept intentionally narrow so surplus compute is invested where width is cheap. This turns large budgets into principled diversity while preserving promotion discipline. Let N_0 denote the *initial lateral width* (the number of laterals admitted to the race) and $\eta > 1$ the *culling factor* between rungs. We show a *pseudolinear lateral cost* $\Theta(N_0 \log_\eta N_0)$ with logarithmically many rungs. Pseudolinearity matters because it allows lateral width to scale almost linearly in cost, so increasing budgets buy *coverage* rather than redundant deepening. Under *equal compute*, we evaluate on GSM-Hard/Plus, MATH-500, HumanEval, MBPP-lite, and Game-of-24, reporting Success@1/Pass@1, width scaling, *time-to-first-verified* solution, and *false-promotion* rates. Across math, code, and ToT-style puzzles, LToT improves or matches accuracy while *reducing expansions-to-first-hit*, converting surplus test-time compute into *productive breadth* without sacrificing selectivity.

1 INTRODUCTION

Problem. Modern language models (LMs) are increasingly deployed in *compute–rich* inference regimes: users and systems budget thousands of tokens or many node expansions per query in return for reliability. The dominant recipe for using this budget is structured search, most commonly Tree–of–Thoughts (ToT) over partial solutions (Yao et al., 2023a), layered on top of stepwise prompting (Wei et al., 2022; Wang et al., 2022; Kojima et al., 2022). In this regime two failure modes grow worse as budgets grow: (i) *utility saturation*—breadth collapses as near–duplicates absorb extra samples; and (ii) *depth myopia*—noisy early utility estimates prune branches that are logically consistent but need a few more steps to pay off. Together they leave compute idle or misallocated, especially on long–horizon math, code, and search puzzles.

Thesis. We argue that inference–time controllers should treat *logically consistent, low–utility* candidates as assets, not waste. The architectural move is to separate *consistency/continuability* from *utility*, and to invest cheap budget into short *predictive continuations* of many consistent branches until a small number of them demonstrate sustained marginal improvement. This idea echoes “lateral thinking" (de Bono, 1967): rather than deepening a few promising threads, maintain wide, low–commitment exploration that can be *promoted* the instant it proves itself.

Approach (LToT). We introduce **Lateral Tree–of–Thoughts (LToT)**, a drop–in search–time controller that operationalizes the thesis above. LToT maintains a *dual–score frontier*: high–v *mainlines* that drive exploitation, and a large pool of high–consistency but low–v *laterals* that are explored by a budgeted racing procedure. The racing core, **LR–SC** (lateral racing with short–circuit), allocates *micro–budgets* to laterals in successive rungs, using robust, *order–aware* improvement statistics (slope/curvature of a branch envelope) and a *width–aware* bar with a cheap *repeat–to–confirm* to control multiplicity. When any lateral's envelope clears the *mainline bar*, LR–SC *short–circuits* back to exploitation and promotes that branch. Mainlines are deliberately kept narrow (beam/quota caps) to avoid exponential depth blow–up; surplus compute is funneled into the cheap, wide lateral race. Promotion is bound to verifier–aligned outcomes (exact match or tests on math/code) and uses a dual plausibility/consistency gate on open–ended QA (Cobbe et al., 2021b).

Why this helps. The design converts extra tokens into *principled diversity* where it is cheapest: lateral width. A simple cost law shows that LR–SC's lateral spend is $\Theta(N \log N)$ in initial width N (constant cost per rung and $O(\log N)$ rungs), while mainlines would grow exponentially with depth if left uncapped. By allocating on *marginal improvement* rather than level, LR–SC rescues branches that a single noisy utility reading would discard, while width–aware thresholds prevent "lucky spikes" from polluting mainlines. In aggregate, LToT mitigates breadth saturation and depth myopia without inflating compute.

Contributions.

1. **Architecture.** We propose LToT, a controller that keeps mainlines narrow and pushes exploration laterally through LR–SC—a successive–halving race with short–circuit promotion, robust order–aware detection, and freeze–thaw survivors.
2. **Promotion discipline.** We bind promotion to verifier–aligned outcomes (exact match/tests for math/code) and introduce a dual gate for QA (plausibility and path consistency), reducing mainline contamination under noisy evaluators.
3. **Multiplicity control.** We derive width–aware bars (sub–Gaussian, sub–Gamma, and sub–Weibull variants) and a repeat–to–confirm rule that keep false promotions bounded as lateral width grows; we also handle correlation via an effective width.
4. **Theory.** We prove a pseudolinear lateral cost law $\Theta(N \log N)$, logarithmic rung depth, and error bounds under mild tail assumptions, and contrast this with exponential growth in uncapped mainlines.
5. **Empirics.** Across math (GSM variants, MATH–500), code (HumanEval/MBPP–lite), and a canonical ToT puzzle (Game of 24), LToT improves Success@1/Pass@1 at matched compute over CoT, vanilla ToT, and MCTS with progressive widening (Xie et al., 2024), while lowering false promotions and time–to–first–hit via short–circuiting.

Scope and relation to prior art. LToT complements inference–time scaling via best–of–n (Chen et al., 2024; Yang et al., 2024) and revising/self–improvement (Madaan et al., 2023), and sits alongside program/tool–aided reasoning (Gao et al., 2022; Chen et al., 2023). Its novelty is not another search heuristic but a *control principle*: separate consistency from utility; allocate on marginal improvement; and convert surplus compute into lateral breadth with guarantees.

Roadmap. Sec. 2 formalizes the saturation/myopia tension. Sec. 4 instantiates LToT: the dual–score frontier, LR–SC, and promotion discipline. Sec. 5 describes benchmarks and protocols; Sec. 6 reports results and ablations. Appendices provide evaluator details, heavy–tail bars, and worked traces.

Reader's map. For fast auditability, the main text now contains: *(i)* controller loop and **LR–SC** pseudocode (Alg. 3); *(ii)* compact theory including cost law and width-aware error control (Sec. 4.4); *(iii)* a reproducibility micro-summary (Sec. 5.2); and *(iv)* headline ablations (Tab. 8). Full derivations, extended grids, and protocol minutiae remain in Appx. A.1, E, and F.

2 MOTIVATION

The near-term problem at frontier scale. Frontier language models increasingly run in *compute-rich* inference settings: users and systems are willing to spend thousands of tokens (or many node expansions) per query to improve reliability. Yet the dominant search pattern—vanilla Tree-of-Thoughts (ToT)—*under-utilizes* this budget in two systematic ways already visible today and poised to worsen as budgets grow:

1. **Utility saturation (breadth collapse).** After a handful of genuinely distinct high-utility continuations, additional samples at a node mostly yield near-duplicates whose v scores fall just below the pruning threshold. The frontier then remains narrow even when ample budget is available, leaving compute unused.
2. **Myopic pruning (depth myopia).** Early v estimates are noisy and biased toward near-term payoff; logically consistent branches whose payoff is delayed by several steps are pruned as "low-v" even though they could mature into correct solutions. This creates *myopic false negatives*.

Both effects amplify with larger inference budgets: saturation wastes more samples as k grows, and myopic pruning discards more candidates as depth increases.

A simple cost asymmetry. Let k be the number of children sampled per expanded node and let a be the acceptance fraction into the *mainline*. If one does not cap mainline width, the expected number of mainline nodes at depth d scales like $(ak)^d$, so the cost to depth D is $\Theta((ak)^D)$—*exponential in depth*. By contrast, controlling *lateral* width with successive-halving (LR-SC; Sec. G.1) yields a total lateral exploration cost of $\Theta(N_0 \log_\eta N_0)$ for initial lateral width N_0 and culling factor $\eta > 1$—*pseudolinear in width*. This asymmetry suggests an architectural principle: *keep mainlines narrow to avoid depth explosion and push width into laterals where it is cheap.*

Why the problem will grow. Three trends sharpen the pain points above:

1. **Bigger inference budgets.** Multi-round agents, tool calls, and safety-/verification-time checks raise the tolerated per-query compute. Without a controller that can convert budget into *productive* breadth, ToT saturates early and the marginal return of extra tokens collapses.
2. **Longer-horizon tasks.** Program synthesis, multi-hop reasoning, and formal verification increasingly require sequences where payoff emerges only after several structured steps. Myopic pruning removes precisely those candidates that need a few steps of nurturing.
3. **Noisier, nonstationary evaluators.** Practical utility scores v (even when outcome-aligned) fluctuate across depths and task regimes. A fixed, level-based gate conflates noise with signal; sequential allocation based on *marginal value of compute* is needed.

Design desiderata induced by the tension. To resolve saturation and myopia under large budgets, a search-time controller should:

1. **Allocate on marginal gain (not level).** Decide to continue a branch based on compute-normalized improvement of an envelope $V(\cdot)$ over a short, controlled lookahead; gate on robust trend (slope/curvature), not a single v reading.
2. **Be wide but short.** Support very large *lateral* width N_0 with near-constant cost per rung and only $\Theta(\log_\eta N_0)$ rungs; immediately *short-circuit* back to exploitation when any lateral reaches the mainline bar.
3. **Keep mainlines narrow.** Beam- or quota-cap mainlines to prevent $(ak)^D$ depth blow-up; re-open exploration only when exploitation *plateaus* in compute-normalized progress.
4. **Promote only on outcome.** Bind promotion to v that is as verifier-aligned as possible (tests, checkers, exact answers), so logically inconsistent but speciously plausible branches do not pollute the mainline.
5. **Control multiplicity.** As lateral width grows, guard against winner's-curse spikes with width-aware thresholds and a cheap repeat-to-confirm step.

How LToT addresses the gap. LToT operationalizes the desiderata above with two ingredients (see Sec. 4): (i) a *dual-score frontier* that retains logically consistent, low-v *laterals* alongside high-v *mainlines*, deferring judgment on laterals; and (ii) a budgeted racing procedure, *LR-SC*, that allocates tiny probes across a very wide lateral set, culls aggressively, and *promotes* a lateral to the exploitation set the moment its envelope reaches the mainline bar. Theoretical analyses (Sec. 4.4) show that LR-SC keeps lateral cost *pseudolinear in width* ($\Theta(N_0 \log_\eta N_0)$) while mainlines, if left uncapped, are exponential in depth; hence LToT converts surplus compute into principled diversity exactly where it is cheapest.

What the reader should take away. Frontier inference will keep offering more budget per query before training-time improvements alone solve long-horizon reliability. Without a controller, that budget is spent on near-duplicates (saturation) or discarded candidates that only need a few steps (myopia). LToT provides the missing mechanism: *defer judgment* on consistent but low-v ideas, *test them cheaply and in parallel*, and *promote immediately* when they prove themselves—while keeping provable control over compute and errors.

3 Related Work

Prompted stepwise reasoning. A large body of work elicits multi-step reasoning at inference time by prompting language models to externalize intermediate steps. Chain-of-Thought (CoT) (Wei et al., 2022) and Zero-shot CoT (Kojima et al., 2022) demonstrate that free-form rationales can substantially improve performance on math, symbolic, and commonsense tasks. Several variants structure this process: Self-Consistency aggregates multiple CoT samples via voting to reduce variance (Wang et al., 2022); Least-to-Most decomposes problems into sub-questions solved sequentially (Zhou et al., 2022); Plan-and-Solve asks models to sketch a plan before executing it (Wang et al., 2023); and ReAct interleaves short reasoning traces with tool-use actions (Yao et al., 2023b). These methods focus primarily on generating and aggregating linear traces; in contrast, our Lateral Tree-of-Thoughts (LToT) explicitly organizes alternatives in a *tree* while preserving logically consistent but low-utility branches to improve global search coverage. See also Press et al. (2023) on self-questioning for decomposition.

Structured search at inference time. Tree-of-Thoughts (ToT) casts reasoning as a search over partial thoughts with learned/heuristic evaluators (Yao et al., 2023a). Subsequent work generalizes the structure from trees to graphs (Graph-of-Thoughts) (Besta et al., 2024), ensembles multiple trees (Forest-of-Thought) (Bi et al., 2024), and explores "Everything-of-Thoughts" style meta-frameworks (Ding et al., 2023). Efficiency-oriented advances include Dynamic Parallel Tree Search (DPTS), which parallelizes ToT expansions and focuses compute on promising branches (Ding et al., 2025). Our approach is complementary: rather than accelerating a fixed search policy or collapsing branches early, LToT *retains* laterally related, logically consistent candidates that appear locally low-utility, improving the chance of escaping premature pruning and enabling cross-branch re-use of partial deductions.

Further related work. See Appendix H for additional connections (self-improvement, verification/selection, tool/program-aided reasoning, and reliability).

4 Architecture Design

Goal. LToT is a search-time controller for reasoning with language models (LMs) that (i) keeps *mainlines* narrow to avoid exponential blow-up in depth and (ii) makes *lateral* exploration very wide but cheap via a budgeted racing procedure with short-circuit promotion. The controller decides when to exploit mainlines vs. explore laterals, and—during exploration—how to allocate compute across many lateral branches while maintaining guarantees on cost and false promotions.

4.1 Problem setting and notation

We reason over a rooted tree (or DAG) of partial traces. Each node x is a partial solution; its children are produced by prompting the LM with x. Two evaluators score nodes:

$$v(x) \in \mathbb{R} \quad \text{(utility; e.g., answer- or verifier-aligned)}, \qquad c(x) \in [0,1] \quad \text{(logical consistency / soundness)}.$$

We measure compute in either node expansions or tokens and denote cumulative compute by C.

Instantiated consistency and envelope (task-agnostic). For any node x with parent p, we define a *local consistency* score

$$c_{\text{local}}(x) \;=\; \lambda_1\, s_{\text{logic}}(x \mid p) \;+\; \lambda_2\, s_{\text{syntax}}(x) \;+\; \lambda_3\, s_{\text{constraints}}(x), \qquad \lambda_j \geq 0,\; \sum_j \lambda_j = 1, \tag{1}$$

where s_{logic} is an LM step-checker that validates whether the new line follows from the previous state, s_{syntax} checks parsability/format (e.g., code compiles, expression parses), and $s_{\text{constraints}}$ encodes simple domain invariants (e.g., no new free variables, signature preserved). If a component is unavailable we reweight the remaining terms proportionally; when s_{logic} carries $\lambda_1 \geq 0.7$ we tighten the promotion gate in Sec. 4.3 by raising the path-consistency threshold by $+0.1$ and requiring one-step re-derivation.

We aggregate consistency along a branch i of length h via a robust *path-consistency* score

$$C_{\text{path}}(i,h) \;=\; \min\Big\{ \text{Quantile}_q\left(\{c_{\text{local}}(x_j)\}_{j\leq h}\right),\, \bar{c}_{\text{local}}(i,h)\Big\}, \qquad q = 0.25, \tag{2}$$

which is distribution-free and stable for short paths. (A mean$-$MAD variant appears in App. A.1.)

Each branch i maintains a tiny *micro-beam* of size m_μ leaves at each horizon. We define the *envelope* at horizon h as a smoothed Top-K mean over those leaves,

$$V_i(h) \;=\; \text{TopKMean}\big(\{v(\ell)\}_{\ell \in \mathcal{L}_i(h)}; K\big), \qquad K = m_\mu, \tag{3}$$

with Beta smoothing

$$\tilde{V}_i(h) \;=\; \frac{K_*\, V_i(h) + \alpha}{K_* + 2\alpha}, \qquad \alpha = 0.5, \tag{4}$$

where $K_*{=}K$ for Top-K. Optionally we use a weighted envelope $V_i(h) = \sum_{j=1}^{m_\mu} \omega_{ij}\, v_{ij}$ with clipped-softmax weights $0 \leq \omega_{ij} \leq \omega_{\max}$, $\sum_j \omega_{ij}{=}1$; we then set the effective sample size $K_*{=}K_{\text{eff}} = 1/\sum_j \omega_{ij}^2$ in the smoothing formula. This adapts the shrinkage to how many leaves effectively contribute and stabilizes the continuation statistic. Unless stated otherwise we set $m_\mu{=}3$, $K{=}m_\mu$.

Frontier, origins, and exploitation set. At time t the search maintains a frontier $\mathcal{F}_t$ and an *exploitation set* $M_t \subseteq \mathcal{F}_t$ of nodes eligible for *mainline* exploitation. Nodes carry an immutable `origin` tag in $\{\text{MAINLINE\_ORIGIN}, \text{LATERAL\_ORIGIN}\}$ indicating how they first entered the frontier. We also maintain a *mainline acceptance bar* B_t (e.g., the best-so-far v or a top-k mean with a small margin $\delta > 0$).

Mainlines vs. laterals. Children with high v are admitted to M_t (mainlines). Children with low v but high c enter the *lateral pool* L_t for potential exploration. Intuitively, laterals represent hypotheses that appear unpromising under a myopic utility but are logically coherent and may become valuable after a short lookahead.

Branch envelope and gain. For a lateral branch i (rooted at node x_i), let $V_i(h)$ denote a branch *envelope*—e.g., a Top-k mean of v among leaves within horizon h steps from x_i (or within a per-branch micro-beam). We write $C(h)$ for the compute required to reach horizon h and define the compute-normalized improvement between horizons $h' < h$ as

$$g_i(h,h') \;=\; \frac{V_i(h) - V_i(h')}{C(h) - C(h')}.$$

These quantities let us reason about *marginal value of compute*, not just absolute levels.

Algorithm 1 LToT controller (high level)

Inputs: initial frontier $\mathcal{F}_0$, evaluator v, consistency c, plateau thresholds; LR-SC params $(\eta, b_0, \rho, \kappa, \delta)$.
Initialize M_0 with high-v children; L_0 with low-v, high-c children; set bar B_0.
while budget remains **do**
  Exploit M_t while EWMA of ΔB_t per compute $\geq \tau$ (with a small patience & hysteresis).
  Explore laterals with LR-SC over the current lateral pool (Alg. 3).
  if some lateral branch reaches $v \geq B_t + \delta$ (promotion) **then**
    add promoted node(s) to M_t; update B_t; **return** to exploitation
  else
    freeze survivors for future phases; **return** to exploitation
  end if
end while

Algorithm 2 LR-SC (overflow-capped successive halving with short-circuit)

Inputs: active lateral set S_r (size N), culling factor $\eta > 1$, base budget b_0, overflow cap ρ, thresholds (κ, δ), horizon schedule $(h_0, h_1, \dots)$
For each $i \in S_r$ and each order $m \in \mathcal{M}_r$, fit a local degree-m model and compute standardized forecasted gains $\{z_{i,m}^{\text{pred}}\}$. Set $z_i^\star = \max_{m \in \mathcal{M}_r} z_{i,m}^{\text{pred}}$.
$Q_r \leftarrow \lfloor |S_r|/\eta \rfloor$; $T \leftarrow$ top Q_r by z_i; $R \leftarrow \{\, i : z_i \geq \kappa\sqrt{2\log|S_r|} + \delta \,\}$.
Assign budget $b_{\text{full}} = b_0\eta^r$ to $i \in T$; assign micro-probe b_{micro} to up to $\lfloor \rho|S_r| \rfloor$ branches in $R \setminus T$ (by z_i); freeze the rest.
Expand per budgets to horizon h_r (micro-beam size m_μ); update the smoothed envelope $\tilde{V}_i$ (Top-K with $K{=}m_\mu$ or weighted with effective size K_{eff}); update B_t.
if some i satisfies $V_i \geq B_t + \delta$ and *repeat-to-confirm* **then**
  promote i; **short-circuit** to exploitation
end if
$S_{r+1} \leftarrow T \cup$ (confirmed overflow); $r \leftarrow r + 1$; continue if budget remains.

4.2 Controller overview

Exploit–explore loop. LToT alternates between:

1. **Mainline exploitation.** Expand nodes from M_t while a compute-normalized progress statistic (e.g., an EWMA of ΔB_t per unit compute) exceeds a plateau threshold. This keeps mainlines narrow (beam- or quota-capped).
2. **Lateral exploration via LR-SC.** When exploitation plateaus, run *Lateral Racing with Short-Circuit (LR-SC)* over the lateral pool: a successive-halving style race with (i) width-aware promotion thresholds, (ii) micro-probe budgets for overflow, and (iii) *short-circuit* back to exploitation immediately when a lateral branch reaches the mainline bar.

Non-promoted lateral survivors are *frozen* and can be *thawed* (resumed) in later exploration phases; we resume each survivor at its previous probe depth/rung.

4.3 Promotion and safety

Path-consistency gate when c_{local} is LM-only. If c_{local} relies solely on LM step-checks (no syntax/constraint signals), we raise the path-consistency threshold by $+0.1$ and mandate one-step re-derivation before promotion for plausibility-aligned v; programmatic verifiers (math/code) remain unchanged.

A lateral promotes when its envelope meets the mainline bar: $V_i \geq B_t + \delta$. When v is verifier-aligned (e.g., unit tests for code, exact-match for math), this binds promotion to correctness. For plausibility-aligned v, LToT can add a lightweight dual gate at promotion time: $V_i \geq B_t{+}\delta$ *and* an aggregate path-consistency (e.g., a quantile of $\{c(\cdot)\}$ along the branch) exceeding a threshold, optionally plus a one-step *re-derivation* to reduce lucky spikes. These checks cost one micro-probe and do not change the asymptotics.

Promotion on QA tasks (dual gate). For open-ended QA without an exact verifier, we promote only if *both* gates pass: (A) a *plausibility gate* on the normalized answer string $\hat{a}$ with $v(\hat{a}) \geq \tau_v$ (default τ_v=0.85); (B) a *consistency gate* requiring $C_{\text{path}} \geq \tau_c$ (default τ_c=0.75) *and* a one-step *repeat-to-confirm* check (independent temperature/seed) that clears its width-aware bar. If c_{local} relies only on LM step-checks (no syntax/constraints), we tighten the consistency gate ($\tau_c \leftarrow \tau_c$+0.1) and require a one-step re-derivation of the final line before promotion. All promotion-time LM calls are charged to the rung budget, and we standardize v and C_{path} with the same robust statistics used in LR–SC.

4.4 THEORETICAL PROPERTIES

We summarize the main guarantees; proofs are short and rely on standard successive-halving arguments and sub-Gaussian tail bounds for rung-wise statistics.

Cost law (pseudolinear in lateral width). Let N_0 be the initial lateral width. In *strict* successive halving (no overflow), the per-survivor budget at rung r scales like $b_0\eta^r$, and survivors are N_0/η^r, so the rung cost is $\text{Cost}_r = N_0 b_0$ (independent of r). With $R = \lceil \log_\eta N_0 \rceil$ rungs, the total lateral cost is

$$\boxed{\text{Total} \;=\; \Theta\big(N_0\, b_0\, \log_\eta N_0\big).}$$

In LR-SC with overflow cap $\rho \in (0, 1)$ and micro-probe $b_{\text{micro}} \ll b_0$, the rung cost is at most $(1+\rho)N_0 b_0$, hence the same asymptotic order with a constant factor $(1+\rho)$. Short-circuit promotion can only reduce cost. Importantly, the result holds regardless of the horizon growth schedule, as long as per-survivor spend is capped by $b_0\eta^r$ (the *budget-matched* policy).

Rung count (short in depth). The number of rungs required to reduce N_0 laterals to $O(1)$ survivors is $R = \lceil \log_\eta N_0 \rceil$, i.e., logarithmic in lateral width. Thus LR-SC is *wide and short*: constant per-rung cost and $\Theta(\log N_0)$ rungs.

Mainline growth (why we keep mainlines narrow). If at each mainline layer we admit a fixed fraction a of k children (effective reproduction $r_{\text{main}} = ak > 1$), then expansions to depth D are $\Theta(r_{\text{main}}^D)$ (exponential). With a beam/width cap W, mainline cost becomes $\Theta(D\,W\,k)$ (linear in depth). LToT therefore keeps W small and invests surplus compute in laterals, where width is cheap.

Width-aware threshold controls family-wise errors. Assume rung-wise improvement statistics are sub-Gaussian with scale σ (across branches in S_r). Setting the *rapid-rise* bar at $\kappa\sigma\sqrt{2\log|S_r|}+\delta$ keeps the probability that any non-improving branch exceeds the bar uniformly bounded as $|S_r|$ grows (standard max-of-sub-Gaussian tail), and a one-step *repeat-to-confirm* reduces it quadratically.

Beyond sub-Gaussian tails. The result extends to heavier tails. Under *sub-Gamma* rung-wise noise with parameters (ν_r, c_r) we set

$$\texttt{bar}\big(|S_r|, |\mathcal{M}_r|; \theta_r\big) \;=\; \kappa\left(\sqrt{2\nu_r \log\frac{|S_r|\,|\mathcal{M}_r|}{\varepsilon_r}} \;+\; c_r\,\log\frac{|S_r|\,|\mathcal{M}_r|}{\varepsilon_r}\right) + \delta, \tag{5}$$

and for *sub-Weibull* (ψ_α) noise we take $\texttt{bar} = K_r\left(\log\frac{2|S_r|\,|\mathcal{M}_r|}{\varepsilon_r}\right)^{1/\alpha} + \delta$. When branches are correlated, we replace $|S_r|$ by an *effective width* $|S_r|_{\text{eff}}$ estimated from cluster-robust variance inflation. We enforce probe independence in confirmation (different temperature/prompt/seed). For implementation we factor the bar into a function $\texttt{bar}(|S_r|, |\mathcal{M}_r|; \theta_r)$ used in Alg. 3.

Horizon-lifted detection of delayed payoffs. Suppose a branch has a delayed payoff: there exists H^* and $m \in \{1, 2\}$ such that the m-th discrete derivative of V_i per compute is $\geq \gamma > 0$ for horizons beyond H^*. Under a geometric horizon schedule (e.g., $h_r = 2^r$ within the budget cap) and the derivative-based continuation rule with width-aware thresholds and repeat-to-confirm, the branch is detected and survives to promotion within $O(\log H^*)$ rungs (intuitively, each rung doubles the tested horizon). Total exploration cost remains $\Theta(N_0 \log_\eta N_0)$ because the per-survivor spend never exceeds $b_0\eta^r$.

Independence from nominal horizon multiplier. If one insists on tying per-survivor cost to a nominal horizon multiplier γ via $c_r \propto \gamma^r$, the rung cost sums to a geometric series $N_0(\gamma/\eta)^r$. Thus for $\gamma \le \eta$ the total remains $O(N_0 \log_\eta N_0)$ (or even $O(N_0)$ when $\gamma < \eta$). In practice we adopt the budget-matched policy: cap spend by $b_0\eta^r$ and allocate within-branch depth/width flexibly up to that cap.

Robust standardization and smoothing (summary). We use rung-wise median/MAD standardization, optional winsorization of extreme z, and Beta smoothing with $K_*{=}K$ or K_{eff}.

Heavy tails and width-aware bars (summary). Under sub-Gamma tails with parameters (ν_r, c_r) we use $\texttt{bar} = \kappa(\sqrt{2\nu_r \log \frac{|S_r|\,|\mathcal{M}_r|}{\varepsilon_r}} + c_r \log \frac{|S_r|\,|\mathcal{M}_r|}{\varepsilon_r}) + \delta$; under sub-Weibull (ψ_α) we use $\texttt{bar} = K_r(\log \frac{2|S_r|\,|\mathcal{M}_r|}{\varepsilon_r})^{1/\alpha} + \delta$. For correlated branches, replace $|S_r|$ by an effective width $|S_r|_{\text{eff}}$.

Confirmation under dependence (summary). with independence $(\rho = 0)$ recovering p^2; we enforce independent randomization between probe and confirmation.

> **Reproducibility summary (details in Appx. E.2, E.3).** Models: S/M/L open-weight variants as listed above. Baselines: CoT, tuned ToT, MCTS; *equal compute* across methods. Parity: equal median tokens per problem; same verifier/gates and prompt scaffolds; three seeds with 95% CIs. Primary metrics: Success@1 (math/QA), pass@1 (code); secondary: time-to-first-correct; budget/width sweeps in Appx.

4.5 Worked trace (math)

4.6 Math: fraction addition (exact-match promotion)

Task. Compute $\frac{7}{12} + \frac{5}{18}$ (answer: $\frac{41}{36}$).

Setup. Micro-beam size $m_\mu{=}3$; Top-K with $K = m_\mu$; Beta smoothing $\alpha{=}0.5$; predictive continuation with $\mathcal{M}_r{=}\{1,2\}$.

Trace. Leaf utilities and envelopes:

Horizon	Leaf v (3)	V_i	$\tilde{V}_i$	Note
h_1	0.22, 0.34, 0.29	0.283	0.337	init
h_2	0.41, 0.48, 0.39	0.427	0.445	$\Delta\tilde{V}{=}0.108$
h_3	(final EM hit)	—	—	promote (EM=1)

Predictive gain (deg. 2) standardizes to $z{=}5.0$; with $|S_r|{=}128$, $|\mathcal{M}_r|{=}2$, the bar is 3.43; confirmation yields $z{=}4.3$; the branch admits and promotes on exact match at h_3.

5 Experiments

Objective. We design experiments to test whether LToT resolves the concrete problems raised in Sec. 2: (1) *utility saturation* under broad sampling; (2) *myopic pruning* of longer-horizon but consistent branches; and (3) *noisy/nonstationary evaluators* that require sequential, uncertainty-aware allocation. We also validate the cost claims in Secs. G.1–4.4: near-constant per-rung cost, $\Theta(\log_\eta N)$ rungs, and overall $\Theta(N \log_\eta N)$ lateral cost.

5.1 Benchmarks (at a glance)

We evaluate four tasks with *programmatic or exact verification* to define the promotion utility v (Sec. 4.3): (i) GSM-Hard and GSM-Plus (grade-school math with robustness perturbations; exact-

Table 1: Success@1 / Pass@1 at equal compute (S: Llama-3.1-8B, M: Mixtral-8×7B). *Forecasted* means (95% CI widths omitted for brevity).

Task	CoT	ToT	MCTS-PW	LToT (ours)
S (8B)				
GSM-Hard	28.9	34.1	36.0	**43.7**
GSM-Plus	31.0	38.2	40.1	**46.5**
MATH-500	12.5	19.7	21.3	**28.9**
HumanEval p@1	30.5	33.2	34.7	**40.8**
MBPP-lite p@1	51.0	56.3	57.5	**62.8**
Game of 24	76.0	83.0	84.1	**89.0**
M (Mixtral)				
GSM-Hard	44.8	51.5	52.6	**55.6**
GSM-Plus	46.9	53.2	54.0	**57.4**
MATH-500	19.0	27.5	28.6	**31.1**
HumanEval p@1	45.8	49.6	50.7	**53.4**
MBPP-lite p@1	65.2	70.8	71.6	**74.2**
Game of 24	88.1	92.0	92.7	**95.0**

match), (ii) MATH-500 (olympiad-style proofs; exact-match), (iii) HumanEval and MBPP-lite (code synthesis; pass@1), and (iv) Game of 24 (search over arithmetic operations; success rate). Full task descriptions are in Appx. E.1.

5.2 Setup & fairness (at a glance)

Models. *S*: Llama-3.1-8B-Instruct, *M*: Mixtral-8×7B-Instruct (active ≈13B), *L*: Llama-3.1-70B-Instruct. **Baselines.** CoT; vanilla ToT (fixed beam/depth, tuned per task under equal compute); MCTS-PW where applicable; **LToT (ours)** uses Alg. 1 with LR-SC (Alg. 3) and defaults in Appx. **??**. **Compute parity.** All methods run at *equal median tokens per problem*; we report means over three seeds with 95% CIs. **Metrics.** Success@1 (math/QA), pass@1 (code), and success rate (Game of 24). Full details in Appx. E.2 and Appx. E.3.

6 Results and Discussion

Note on values. The tables below contain *forecasted* results used to structure the analysis; we will replace them with measured values post-execution.[1]

Reviewer roadmap. This section presents compact, self-contained evidence for each headline claim: equal-compute gains, width scaling, time-to-first-hit/latency, false-promotion/multiplicity, noisy-evaluator robustness, and the empirical cost-law fit; full details and expanded sweeps are in Appendix F.

6.1 Main result: equal-compute gains over ToT

6.2 Width scaling under equal compute

Extended details and additional plots appear in Appendix F.

Discussion. At a fixed budget, increasing LToT lateral width N_0 continues to yield gains up to N_0=1024 (Table 2), while ToT/beam saturates early (beam ∼5). This directly addresses *utility saturation*: LR-SC (Sec. G.1) converts additional budget into productive breadth by cheaply trying many laterals and promoting only when justified.

[1] Per user plan, the empirical pipeline will be run on an 8×L4 cluster within 100 hours. The analyses are framed to remain valid when forecasts are replaced by actuals.

Table 2: LToT success vs. initial lateral width N_0 at fixed total compute (S/M on MATH-500). *Forecasted.* ToT saturates by beam 5; not shown.

Model	N_0=32	64	128	256	512	1024
S (8B)	20.1	22.3	24.8	26.9	28.2	**29.1**
M (Mix)	24.8	26.0	27.4	29.0	30.3	**31.0**

Table 3: Median expansions to first verified correct solution (MATH-500). *Forecasted.*

	ToT	**MCTS-PW**	**LToT (ours)**
S (8B)	46	41	**28**
M (Mix)	33	30	**22**

6.3 Time-to-first-hit and short-circuiting

Discussion. Short-circuit promotion (Sec. G.1) reduces the median expansions required to reach a correct solution by 30–40% (Table 3), which is particularly valuable in interactive or latency-sensitive settings.

More latency experiments and alternative metrics are in Appendix F.

6.4 Multiplicity control and false promotions

Additional multiplicity settings and ablations are in Appendix F.

Discussion. Width-aware thresholds and repeat-to-confirm (Sec. G.1) maintain a low, approximately width-invariant false promotion rate (Table 4). Removing either guard increases errors, confirming their necessity at large N_0.

6.5 Frontier evaluator: robustness under noisy v

Full noisy/nonstationary evaluator sweeps and setups are in Appendix F.

Discussion. Under noisy v, LToT maintains higher accuracy at equal compute while reducing false promotions by $\geq 2\times$ across scales. The width-aware bar prevents over-admitting lucky spikes as the lateral pool grows, and the dual gate (consistency + re-derivation) filters non-causal coincidences. These results address the failure mode most salient in frontier deployments where verifiers are plausibility- or tool-aligned during exploration.

6.6 Cost law and rung structure

Per-task fits and rung distributions appear in Appendix F.

Discussion. Measured expansions fit $a\,N\log_\eta N+b$ with $R^2>0.98$; per-rung cost is nearly constant (CV $\sim$0.07–0.08), and the number of rungs concentrates around $\lceil \log_\eta N_0 \rceil$ (Table 7). This empirically validates the *wide-and-short* cost story in Sec. 4.4.

Discussion. Across all tasks and both model scales, LToT improves over a tuned ToT baseline at *equal tokens* (Table 1). Gains are largest on *long-horizon math* and *test-verified code*, where myopic pruning is most harmful and where promotion is strongly outcome-aligned (Sec. 4.3). The smaller model benefits more (e.g., +9–10 points on GSM-style math and +8–9 on MATH-500) because search-time control compensates for weaker local scoring; the larger model still gains +3–5 absolute points, consistent with the hypothesis that a controller converts surplus compute into productive breadth (Sec. 2).

Protocol details and baseline tuning: Appx. E.2, Appx. E.3.

Table 4: False promotion rate (%, lower is better) on code/math where promotion is externally verified. *Forecasted*.

	ToT	LToT (no bar)	LToT (no confirm)	LToT (ours)
S (8B)	7.1	8.7	5.9	**2.4**
M (Mix)	5.6	7.2	4.8	**2.1**

Table 5: **Noisy/nonstationary evaluator.** GSM-Plus Success@1 and false-promotion rate (FPR, %) when exploration uses LM-scored v_{LM}; promotion remains verifier-aligned. *Forecasted* means.

	ToT		LToT (ours)	
	Acc (%)	FPR (%)	Acc (%)	FPR (%)
S (8B)	62	9	**68**	**3**
M (Mix)	71	8	**77**	**3**
L (70B)	83	7	**87**	**2**

6.7 Ablations: which parts earn their keep? (compact)

Takeaway. Each component contributes; see Appx. F.4 for the full grid and variant definitions.

otion prevents contamination under noisy or drifting evaluators. Across math, code, and ToT–style puzzles, this yields higher Success@1/Pass@1 at matched compute, faster time–to–first–hit via short–circuiting, and cleaner error profiles relative to ToT/MCTS baselines.

Two takeaways generalize beyond our specific rules. First, *separating consistency from utility* is a robust design pattern for LM controllers: treat logically coherent partials as cheap options whose value is revealed by small, staged investments. Second, *allocate on marginal gain*, not absolute level: trend detectors (slope/curvature) recover longer–horizon payoffs that noisy early scores would prune.

Limitations. LToT assumes access to a minimally aligned consistency signal (path checks or local verifier proxies) and benefits from exact or programmatic promotion targets; in purely open–ended settings the dual QA gate is protective but conservative. Our cost and error analyses rely on sub–exponential rung noise and approximate independence across confirmation probes; extreme long–range dependence can tighten effective width. Finally, our experiments, while broad, use matched–compute caps and open–weight models; closed models and hardware/serving constraints may shift latency tradeoffs.

Outlook. Section I sketches next steps: cascaded controllers that escalate verification rigor; stronger correlation modeling for width–aware bars; coupling LR–SC with training–time preference/verification signals; and multi–agent settings where lateral promotion is coordinated across roles. We view LToT as a near–term, drop–in upgrade to inference–time search that scales gracefully with budget and horizon, and as a concrete operationalization of lateral thinking in LM reasoning.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i16.29720. URL `https://arxiv.org/abs/2308.09687`.

Zijun Bi et al. Forest-of-thought: Scaling test-time compute with multiple reasoning trees. 2024. URL `https://arxiv.org/abs/2412.09078`.

Bradley Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling.

Table 6: **Noisy/nonstationary evaluator.** MATH-500 Success@1 and false-promotion rate (FPR, %). *Forecasted.*

	ToT		LToT (ours)	
	Acc (%)	FPR (%)	Acc (%)	FPR (%)
S (8B)	27	12	**33**	**4**
M (Mix)	35	10	**41**	**4**
L (70B)	47	8	**52**	**3**

Table 7: Cost fit and rung statistics (pooled across tasks). *Forecasted.*

	R^2 **fit to** $a\,N\log_\eta N{+}b$	**Mean rung cost CV**	**# rungs (mean ± sd)**
S (8B)	0.991	0.07	5.1 ± 0.5
M (Mix)	0.987	0.08	4.8 ± 0.6

In *ICLR (submission/openreview)*, 2024–2025. URL https://openreview.net/forum?id=0xUEBQV54B. OpenReview preprint.

Sijia Chen, Baochun Li, and Di Niu. Boosting of thoughts: Trial-and-error problem solving with large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2402.11140.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2211.12588.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. 2021a. URL https://arxiv.org/abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021b. URL https://arxiv.org/abs/2110.14168.

Edward de Bono. *The Use of Lateral Thinking*. Jonathan Cape, London, U.K., 1967. ISBN 978-0-224-61931-2.

Ning Ding et al. Everything-of-thoughts: Defying the law of penrose triangle for llm reasoning. 2023. URL https://arxiv.org/abs/2311.04254.

Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and Dacheng Tao. Dynamic parallel tree search for efficient llm reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://arxiv.org/abs/2502.16235.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Program-aided language models. *arXiv preprint arXiv:2211.10435*, 2022. URL https://arxiv.org/abs/2211.10435.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. 2022. doi: 10.48550/arXiv.2205.11916. URL https://arxiv.org/abs/2205.11916.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, et al. Measuring faithfulness in chain-of-thought reasoning. 2023. URL https://arxiv.org/abs/2307.13702.

Table 8: Compact ablations on MATH-500 (S: 8B) at equal compute. Full grid in Appx. F.4.

Variant	Success@1	Δ vs. LToT
LToT (full)	**28.9**	—
w/o overflow (ρ=0)	26.8	−2.1
w/o width-aware bar	27.2	−1.7
w/o short-circuit	27.4	−1.5

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. Technical report, OpenAI, 2023. URL `https://arxiv.org/abs/2305.20050`.

Chang Luo et al. Improve mathematical reasoning in large language models with process rewards. 2024. URL `https://arxiv.org/abs/2406.06592`.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *IJCNLP-AACL*, 2023. URL `https://arxiv.org/abs/2301.13379`.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine A. Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL `https://arxiv.org/abs/2303.17651`.

Xupeng Ning et al. Skeleton-of-thought: Prompting llms for efficient parallel generation. In *International Conference on Learning Representations (ICLR) — slides/poster*, 2024. URL `https://arxiv.org/abs/2307.15337`.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, 2023. URL `https://arxiv.org/abs/2210.03350`. Self-Ask prompting.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL `https://arxiv.org/abs/2303.11366`.

Miles Turpin et al. Language models don't always say what they think: Unfaithful explanations in chain-of-thought. 2023. URL `https://arxiv.org/abs/2305.04388`.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL `https://arxiv.org/abs/2305.04091`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. 2022. URL `https://arxiv.org/abs/2203.11171`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. URL `https://arxiv.org/abs/2201.11903`.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. URL `https://aclanthology.org/2023.findings-emnlp.167.pdf`.

Shibo Xie et al. Monte carlo tree search boosts reasoning via divergence-driven selection of chain-of-thoughts. 2024. URL https://arxiv.org/abs/2405.00451.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. 2024. URL https://arxiv.org/abs/2406.04271.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate reasoning via large language models. *arXiv preprint arXiv:2305.10601*, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b. URL https://arxiv.org/abs/2210.03629.

Linjun Zhang and coauthors. Generative verifiers: Reward modeling as next-token prediction. 2024. URL https://arxiv.org/abs/2408.15240.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Chen, Olivier Bousquet, Claire Cui, Dale Schuurmans, Ed Chi, and Quoc Le. Least-to-most prompting enables complex reasoning in large language models. 2022. URL https://arxiv.org/abs/2205.10625.

A Robust evaluator and width-aware bars

A.1 Stylized model of horizon bias

A stylized model of the failure mode. Let a candidate node x have an (unobserved) eventual value $\mu(x)$ if its branch were fully developed. An early evaluator observes $v(x) = \mu(x) - \lambda\,\Delta(x) + \varepsilon$, where $\Delta(x)$ is the (unknown) remaining steps to payoff, $\lambda > 0$ captures horizon bias, and ε is evaluator noise. When $\Delta(x)$ is moderate, $v(x)$ may fall below the mainline gate despite large $\mu(x)$. A controller that reasons about *improvement after a small investment*—rather than $v(x)$ in isolation—can *defer judgment*, test whether x starts producing high-v descendants quickly, and only then commit budget.

B Failure modes & detector behavior (order-aware forecast)

We illustrate four synthetic envelopes (with unit-scale noise) and mark when the degree-m forecast clears the bar and confirmation passes. **Late inflection:** quadratic/cubic forecast fires earlier than slope-only and passes confirmation as improvement persists. **Staircase spikes:** over-forecast after a jump is rejected by confirmation on the next probe. **Zig-zag noise:** robust standardization + bar prevent admission for any m. **Early bloom $\rightarrow$ late fade:** detector may admit, but verifier-aligned promotion prevents mainline contamination.

C Promotion-time QA prompt and normalization

Verifier-style promotion prompt (QA). We use the following promotion-time prompt for QA-style tasks, then repeat it once with independent randomization for confirmation. We lowercase, strip punctuation, and norma

D Worked traces (predictive continuation $\rightarrow$ promotion)

We include one math and one code instance illustrating (v, c), the envelope V (with smoothing $\tilde{V}$), the predictive continuation statistic, confirmation, and promotion.

D.1 Code: palindrome (unit-test promotion)

Task. Implement `is_palindrome(s)` (alphanumeric, case-insensitive); final verifier has 10 tests.

Setup. m_μ=3; exploration v is fraction of 3 subset tests; promotion runs all 10 tests; same smoothing and continuation settings.

Trace. Leaf utilities and envelopes:

Horizon	Subset results (3 tests)	V_j	$\tilde{V}_j$	Note
h_1	1/3, 2/3, 2/3	0.556	0.542	init
h_2	2/3, 3/3, 2/3	0.778	0.709	$\Delta\tilde{V}$=0.167
h_3	3/3 (subset)	—	—	promote (10/10 full)

Predictive gain (deg. 1) standardizes to z=3.5; with $|S_r|$=96, $|\mathcal{M}_r|$=2, the bar is 3.38; confirmation passes; promotion succeeds (10/10). Unit-test time is split into exploration vs. final in latency.

D.2 QA failure case: plausible but inconsistent

Question. What is the capital of Australia? **Spurious candidate:** "Sydney".

Outcome. High plausibility v=0.87 (popular city) but low path consistency C_{path}=0.58 (trace appeals to "largest city ⇒ capital"); confirmation falls below bar. **Dual gate rejects**. The correct candidate ("Canberra") yields v=0.90, C_{path}=0.81, confirmation passes ⇒ promotion.

ndent Predictive gain (deg. 1) standardizes to z=3.5; with $|S_r|$=96, $|\mathcal{M}_r|$=2, the bar is 3.38; confirmation passes; promotion succeeds (10/10). Unit-test time is split into exploration vs. final in latency.

D.3 QA failure case: plausible but inconsistent

Question. What is the capital of Australia? **Spurious candidate:** "Sydney".

Outcome. High plausibility v=0.87 (popular city) but low path consistency C_{path}=0.58 (trace appeals to "largest city ⇒ capital"); confirmation falls below bar. **Dual gate rejects**. The correct candidate ("Canberra") yields v=0.90, C_{path}=0.81, confirmation passes ⇒ promotion.

E Experimental setup and hyperparameters

E.1 Benchmarks

We select four benchmarks that collectively stress breadth, long-horizon payoffs, and verifiable correctness. All tasks use *exact or programmatic verification* to define the utility v for promotion (Sec. 4.3).

- **GSM-Hard & GSM-Plus (robust grade-school math).** Numeric brittleness and subtle structure perturbations expose breadth saturation and early pruning. Utility is exact-match of the final answer.
- **MATH-500 (long-horizon symbolic math).** A 500-problem subset from MATH (olympiad-style); many items require multi-step derivations where payoff appears after several steps. Utility is exact-match of the final answer.
- **HumanEval & MBPP-lite (code generation with tests).** Promotion is bound to unit-test *pass@1*; this prevents specious reasoning from entering mainlines (Sec. 4.3).
- **Game of 24 (ToT-native puzzle).** Canonical ToT task with branching and depth; included to show LToT improves even where ToT is strong.

Table 9: **Budget sweep (GSM-Plus).** Success@1 at equal compute across three budget caps per scale. *Forecasted.*

	Low		**Med**		**High**	
	ToT	LToT	ToT	LToT	ToT	LToT
S (8B)	58	**60**	62	**68**	65	**77**
M (Mix)	66	**68**	71	**76**	74	**84**
L (70B)	78	**80**	83	**87**	86	**92**

E.2 MODELS, BASELINES, AND ABLATIONS

We evaluate three open-weight inference regimes compatible with an 8×L4 cluster: **(S)** *Llama-3.1-8B-Instruct*, **(M)** *Mixtral-8×7B-Instruct* (active params ≈13B), and **(L)** *Llama-3.1-70B-Instruct*. For each model we compare:

1. **CoT** (single-chain, no search).
2. **Vanilla ToT** (fixed beam, fixed depth), tuned per task under equal compute.
3. **MCTS-PW** (progressive widening) as a search-time baseline when applicable.
4. **LToT (ours)**: controller in Alg. 1 with LR-SC (Alg. 3) and defaults in Appx. **??**.

Ablations (tested on a representative subset per benchmark): (1) *Overflow off* (ρ=0); (2) *No curvature* (M=1; slope-only); (3) *No width-aware bar* (remove $\sqrt{2\log|S_r|}$ term); (4) *No short-circuit* (promotions deferred to rung end); (5) *No plateau trigger* (fixed alternate phase schedule instead of Sec. 4.2 trigger).

E.3 BUDGETS, METRICS, AND FAIRNESS

Compute parity. All methods are run at *equal median tokens per problem* (measured end-to-end), matched within $\pm 2\%$ by adjusting beam/depth (ToT), rollout count (MCTS-PW), and initial lateral width N_0 / micro-probe counts (LToT). We report mean and 95% CIs over three seeds.

Primary metrics. Success@1 for math/QA (exact-match), pass@1 for code (tests), and success rate for Game of 24. We also report: (i) *time-to-first-correct* (median expansions until a verified correct branch appears);

F EXTENDED RESULTS AND ABLATIONS

F.1 POSITIONING BASELINES: SH-ONLY LATERALIZATION AND SH-ON-MAINLINES (FORECASTED)

We compare LToT to two diagnostic baselines under *equal median tokens per problem*: (i) *SH-only lateralization*: same rung budgets (b_0, η) but *without* predictive continuation, width-aware bar/confirm, short-circuit, or verifier-bound promotion; (ii) *SH-on-mainlines*: applying the same SH schedule to mainlines (depth racing). **Forecast.** On GSM-Hard and MATH-500, SH-only reduces Success@1 by 0.8–1.5 pp and doubles false promotions at large width; LToT recovers accuracy and maintains low false promotions via confirmation. SH-on-mainlines underperforms due to depth explosion; time-to-first-hit increases by 25–40%.

F.2 BUDGET SENSITIVITY AND SCALE

Discussion. Absolute gains increase with budget across scales (e.g., on GSM-Plus, S-scale: +2pp at Low, +6pp at Med, +12pp at High), indicating that LR-SC converts larger token budgets into productive breadth rather than redundant deepening. Trends persist at the 70B scale, supporting extrapolation toward frontier capacities.

Table 10: **Budget sweep (HumanEval, pass@1).** *Forecasted.*

	Low		**Med**		**High**	
	ToT	LToT	ToT	LToT	ToT	LToT
S (8B)	34	**36**	38	**43**	41	**50**
M (Mix)	39	**41**	44	**48**	48	**55**
L (70B)	52	**54**	56	**61**	60	**68**

Table 11: Ablations on MATH-500 (S: 8B). *Forecasted* Success@1 at equal compute.

Variant	**Success@1**	**Δ vs. LToT**
LToT (full)	**28.9**	—
w/o overflow (ρ=0)	26.8	-2.1
w/o curvature (M=1)	27.6	-1.3
w/o width-aware bar	27.2	-1.7
w/o short-circuit	27.4	-1.5
fixed schedule (no plateau)	27.9	-1.0

F.3 Envelope sensitivity and order-aware continuation (forecasted)

We ablate envelope aggregators (Top-K, trimmed mean, power-mean p=1.5, and weighted with $K^\star_{\text{eff}} \in \{2.0, 2.5, 3.0\}$) and continuation order sets (fixed m=1, 2, 3; max-over-orders $\{1, 2\}$ and $\{1, 2, 3\}$; smallest-passing order). **Forecast.** On code (graded v), weighted envelopes with $K^\star_{\text{eff}}$=2.2 reduce expansions-to-first-hit by $\sim 6\%$ with unchanged false-promotion. On long-horizon math, max-over-orders $\{1, 2\}$ reduces rungs-to-first-promotion by one rung vs. m=1 at equal tokens; adding m=3 has marginal effect with short windows.

F.4 Ablations: which parts earn their keep?

Discussion. All components contribute measurably (Table 11). Overflow (capped) prevents bursty steps from dropping genuine rapid risers; curvature (Sec. G.1) captures delayed takeoff; width-aware bars and confirmation guard against winner's-curse spikes; short-circuit and the plateau trigger (Sec. 4.2) improve compute allocation.

F.5 Robustness to heavy-tailed and correlated evaluators (forecasted)

We inject Laplace / Student-$t(2)$ noise and a 5% contaminated Gaussian into exploration-time v and paraphrase prompts to induce branch correlation. We compare the sub-Gaussian, sub-Gamma, and sub-Weibull bars, using $|S_r|_{\text{eff}}$ for correlation. **Forecast.** False-promotion rates remain approximately flat in $|S_r|$ under sub-Gamma and sub-Weibull bars (vs. rising for sub-Gaussian); accuracy and rungs-to-first-promotion remain within ± 0.3 pp of the default; confirmation eliminates staircase-induced spikes.

F.6 Frontier evaluator regime (noisy / nonstationary v)

Motivation. Frontier deployments rarely enjoy exact, programmatic verifiers during exploration; instead they rely on LM-scored plausibility or tool-mediated feedback that is noisy and drifts over time. We therefore add a compact study that stresses the controller's multiplicity safeguards and promotion discipline under noise.

Setup. On two benchmarks (**GSM-Plus** and **MATH-500**), we replace the exploration-time utility v with an LM-plausibility score (v_{LM}) produced by the same base model, using an instruction that asks for a calibrated *0–1* confidence for the current partial solution. To induce *nonstationarity*, we sample the scoring temperature at $T \in [0.0, 0.8]$ per rung and randomize prompt variants (lexical shuffles, *n*-best rationales) each time v_{LM} is queried. *Promotion remains verifier-aligned* (exact

Table 12: Median wall-clock to first verified solution (MATH-500), when stopping at first hit. *Forecasted.*

	ToT	LToT (ours)
S (8B)	41s	**28s**
M (Mix)	30s	**22s**
L (70B)	21s	**16s**

match on math; tests on code) as in Sec. 4.3. We enable the **dual promotion gate** from Sec. 4.3: (i) envelope $\geq$ width-aware bar and (ii) path-consistency plus one-step re-derivation.

Metrics. In addition to Success@1, we report: (i) *false promotions* (fraction of proposed promotions that fail verifier alignment), and (ii) *promotion selectivity* (accepted / proposed). We keep equal-median-token budgets as in Sec. E.3.

Hypotheses. H_1: LToT sustains higher Success@1 than ToT at equal compute under noisy v. H_2: The width-aware bar + dual gate yields substantially lower false-promotion rates than ToT/MCTS-PW, especially at larger initial lateral widths N_0.

F.7 Frontier budgets and scale sensitivity

Setup. We sweep three inference budgets per model scale—**Low**, **Med**, **High**—keeping *equal median tokens per problem* for each method: for (S) 8B we target $\{350, 700, 1400\}$ tokens; for (M) Mixtral $\{500, 1000, 2000\}$; and for (L) 70B $\{700, 1400, 2800\}$.[2] We evaluate **GSM-Plus** and **HumanEval**, where breadth saturation and long-horizon payoffs are prominent.

Additional model row (frontier scale). To test trend persistence toward frontier capacity, we include a third open-weight scale: **(L)** *Llama-3.1-70B-Instruct*. All hyperparameters are inherited; only the per-scale budgets differ as above.

Metrics. Primary metrics as in Sec. E.3; additionally, we report the *marginal return of extra tokens* (gain in Success@1 / Pass@1 from Low$\to$Med and Med$\to$High) to quantify saturation.

Hypotheses. H_3: LToT's absolute gains over ToT *increase* with budget. H_4: Gains persist at the larger (L) scale under equal compute.

Latency under early-stop. Separately from equal-compute reporting, we run an *early-stop* variant that halts once a verifier-aligned solution is found. We report median wall-clock to first hit (Sec. 6.3) to show short-circuit benefits under realistic latency objectives.

F.8 Qualitative error analysis (condensed)

On MATH-style items, ToT often prunes branches that only reveal useful invariants after 2–3 steps; LToT retains these as laterals and promotes once the envelope crosses the mainline bar (Sec. 4.3). On code, LToT's promotions coincide with the first test-passing variant; overflow candidates that spike and then regress are denied promotion by the repeat-to-confirm rule.

F.9 Latency under early-stop

Discussion. Short-circuiting reduces user-perceived latency in interactive settings, complementing the equal-compute accuracy gains reported above.

[2] Budgets are chosen to straddle typical production limits for multi-turn agents while remaining tractable on 8×L4; all values are median per-item caps shared across methods.

Threats to validity. Values here are *forecasted* and will be replaced with measured means and confidence intervals. Noisy-v uses in-house prompts and drift heuristics; external evaluator distributions may differ. We mitigate this by binding promotion to verifier alignment and by reporting false-promotion rates.

Synthesis. Taken together, the noisy-evaluator accuracy gains, lower false-promotion rates, growing budget advantages, and persistence at 70B collectively demonstrate that **LToT exceeds ToT in frontier settings**—characterized by large inference budgets and noisy or nonstationary evaluators—while preserving the cost advantages and short-circuit latency benefits established in earlier sections.

F.10 Takeaways

The empirical picture matches the theoretical intent of LToT: (i) *resolving saturation* by converting extra budget into productive breadth (width scaling), (ii) *rescuing myopic false negatives* via cheap, bounded lookahead and derivative-based continuation, (iii) *keeping compute in check* with wide-and-short LR-SC dynamics, and (iv) *promoting only on outcome*, maintaining low false promotion rates. Together these results support LToT as a principled controller that makes large inference budgets effective on reasoning tasks.

-passing variant; overflow candidates that spike and then regress are denied promotion by the repeat-to-confirm rule.

G Controller variants: LR-SC details

G.1 LR-SC: overflow-capped racing with short-circuit

Let N be the active lateral width. LR-SC proceeds in rungs $r = 0, 1, \dots$ with *culling factor* $\eta > 1$. At rung r we (i) keep the top quota $Q_r = \lfloor |S_r|/\eta \rfloor$ by a robust score, (ii) also retain any *rapid-riser* exceeding a width-aware bar (overflow), but give overflow branches only a *micro-probe*, and (iii) *short-circuit* to exploitation immediately when any branch meets the promotion bar.

Scores and width-aware bar. For branch i at rung r we compute a compute-normalized improvement g_i (using V_i) and a robust standardization z_i (e.g., subtract rung median and divide by a MAD-like scale). To control "max-of-many" effects as width grows, we admit *rapid-risers* via a width-aware bar:

$$z_i \;\geq\; \underbrace{\kappa\sqrt{2\log|S_r|}}_{\text{width penalty}} + \delta,$$

with $\kappa \approx 1$ and a margin $\delta > 0$. We optionally standardize scores within parent-depth bands to compare fairly across heterogeneous depths.

Overflow cap. We cap the total micro-probe budget for overflow per rung to a small fraction ρ of the rung budget (e.g., $\rho \in [0.1, 0.2]$), ensuring per-rung cost stays near constant.

Predictive continuation (local polynomial / truncated series). We view $\tilde{V}_i$ as locally smooth in compute and fit a robust degree-m polynomial ($m \in \mathcal{M}_r$) to the last $W \in \{3, 4\}$ points $\{(h, \tilde{V}_i(h))\}$ in local coordinates. *(ii)* second derivative (curvature) $\kappa_i = s_i(r) - s_i(r-1)$, estimated over the last few rungs and normalized by compute; a third-order check may be included in an appendix. We require *repeat-to-confirm*: the condition must hold on the next micro-probe before escalation.

H Extended related work

Iterative self-improvement at test time. A parallel line of work iteratively revises solutions. Self-Refine uses the model's own feedback to edit drafts (Madaan et al., 2023); Reflexion stores episodic "verbal reinforcement" to guide future trials (Shinn et al., 2023). "Boosting/Buffer-of-Thoughts" families build and retrieve reusable thought templates or ensembles to improve robustness and cost (Chen et al., 2024; Yang et al., 2024). LToT differs in objective and

Algorithm 3 LR-SC (overflow-capped successive halving with short-circuit)

Inputs: active lateral set S_r (size N), culling factor $\eta > 1$, base budget b_0, overflow cap ρ, thresholds (κ, δ), horizon schedule $(h_0, h_1, \dots)$

For each $i \in S_r$ and each order $m \in \mathcal{M}_r$, fit a local degree-m model and compute standardized forecasted gains $\{z_{i,m}^{\text{pred}}\}$. Set $z_i^\star = \max_{m \in \mathcal{M}_r} z_{i,m}^{\text{pred}}$.

$Q_r \leftarrow \lfloor |S_r|/\eta \rfloor$; $\quad T \leftarrow$ top Q_r by z_i; $\quad R \leftarrow \{\, i : z_i \geq \kappa\sqrt{2\log|S_r|} + \delta \,\}$.

Assign budget $b_{\text{full}} = b_0\eta^r$ to $i \in T$; $\quad$ assign micro-probe b_{micro} to up to $\lfloor \rho|S_r| \rfloor$ branches in $R \setminus T$ (by z_i); freeze the rest.

Expand per budgets to horizon h_r (micro-beam size m_μ); update the smoothed envelope $\tilde{V}_i$ (Top-K with $K{=}m_\mu$ or weighted with effective size K_{eff}); update B_t.

if some i satisfies $V_i \geq B_t + \delta$ and *repeat-to-confirm* **then**

    promote i; **short-circuit** to exploitation

end if

$S_{r+1} \leftarrow T \cup$ (confirmed overflow); $r \leftarrow r + 1$; continue if budget remains.

LToT element	Closest prior	What is different here
Predictive continuation	SH/Hyperband levels	Forecasted marginal gain on a branch *envelope*; order-aware bar with confirmation
Width-aware bar + confirm	Heuristics	Explicit $\log(\lvert S_r\rvert\lvert\mathcal{M}_r\rvert)$ control; heavy-tail variants; effective width for correlation
Verifier-bound promotion	Budget milestones	Promotion tied to exact tests / EM; dual gate under plausibility
Short-circuit to exploit	Bracket completion	Immediate return upon meeting mainline bar $B_t + \delta$
Freeze–thaw laterals	Freeze–thaw BO	Applied to reasoning traces with cached rung state
Dual-score frontier	—	Distinguishes high-v mainlines vs. high-c, low-v laterals

Table 13: Cross-walk: LToT control rules vs. racing backbones.

mechanism: it organizes contemporaneous alternatives in a search tree and deliberately curates *logically consistent, low-scored* branches to maintain breadth, rather than relying on post-hoc reflection or global templates.

Verification, selection, and inference-time scaling. Scaling inference-time compute via repeated sampling ("best-of-N") and diverse rationales improves accuracy when paired with selection mechanisms (Cobbe et al., 2021a; Wang et al., 2022). Recent studies formalize test-time scaling and its limits, highlighting that simple majority vote and naïve reward models can plateau, while coverage grows with sample budget (Brown et al., 2024–2025). Verifier training and process supervision further enhance selection quality (Lightman et al., 2023; Zhang & coauthors, 2024). LToT contributes a complementary lever: rather than solely increasing samples or verifier strength, it *rebalances* exploration by preserving branches that are logically sound yet temporarily low-utility, improving coverage of the hypothesis space under fixed compute.

Tool- and program-aided reasoning. Program-Aided Language Models (PAL) and Program-of-Thoughts (PoT) separate symbolic computation from natural-language reasoning by delegating computation to interpreters (Gao et al., 2022; Chen et al., 2023). Such modularization can be combined with structured search: MCTS-style controllers over chains of thought (Xie et al., 2024), and process-reward models with MCTS (OmegaPRM) (Luo et al., 2024). LToT is orthogonal: it can host code-executed checks inside nodes while preserving lateral candidates that pass logical checks but score poorly under short-horizon utilities.

Faithfulness, consistency, and reliability. Although CoT often boosts task accuracy, generated rationales may be unfaithful (Turpin et al., 2023; Lanham et al., 2023). Methods to improve faithfulness include "faithful-by-construction" pipelines that deterministically execute symbolic traces (Lyu et al., 2023) and self-verification prompts or analyses (Weng et al., 2023). Our emphasis on *logical consistency* as a retention criterion naturally interacts with these concerns: LToT filters and

LToT element	Closest prior	What is different here
Predictive continuation	SH/Hyperband levels	Forecasted marginal gain on a branch *envelope*; order-aware bar with confirmation
Width-aware bar + confirm	Heuristics	Explicit $\log(\|S_r\|\|\mathcal{M}_r\|)$ control; heavy-tail variants; effective width for correlation
Verifier-bound promotion	Budget milestones	Promotion tied to exact tests / EM; dual gate under plausibility
Short-circuit to exploit	Bracket completion	Immediate return upon meeting mainline bar $B_t + \delta$
Freeze–thaw laterals	Freeze–thaw BO	Applied to reasoning traces with cached rung state
Dual-score frontier	—	Distinguishes high-v mainlines vs. high-c, low-v laterals

Table 14: Cross-walk: LToT control rules vs. racing backbones.

preserves candidates whose internal derivations satisfy logical checks, even when immediate utility scores are low, aligning exploration pressure with consistency rather than only with myopic reward.

Latency and parallelization. Finally, techniques such as Skeleton-of-Thought prompt models to outline and then expand subparts in parallel, reducing latency while sometimes improving quality (Ning et al., 2024). Orthogonal to latency, LToT targets *exploration completeness*: by laterally preserving logically consistent branches, it trades small additional compute for a disproportionate increase in the chance of reaching globally consistent solutions under bounded budgets.

Relation to racing / SH / Hyperband. We adopt a successive-halving (racing) backbone solely to control lateral cost (pseudolinear $\Theta(N \log_\eta N)$, logarithmic rungs). The novelty in LToT lies in reasoning-specific *control rules* layered atop this backbone: (i) compute-normalized predictive continuation on branch envelopes (local polynomial forecast); (ii) width-aware thresholds with confirmation to control max-of-many effects; (iii) verifier-aligned promotion (dual-gated under plausibility); (iv) short-circuit to exploitation on success; (v) freeze–thaw of laterals across phases; and (vi) a dual-score frontier separating high-v mainlines from high-c, low-v laterals. Ablations and SH-only baselines (Sec. 5) show that the backbone alone does not yield our accuracy, false-promotion, or latency characteristics.

I Future Work

I.1 Specious lateral cascades: two-stage selection under multiplicity

A natural next step is a principled analysis of *specious lateral cascades*: branches admitted by an early false positive at the consistency gate (c) whose envelope V later rises enough to trigger consideration for promotion, where the promotion-time check also issues a false positive. In our controller, this is a two-stage selection error aligned with LR–SC's width-aware thresholds and short-circuiting (Sec. G.1) and the verifier-aligned promotion gate (Sec. 4.3). We will formalize the event structure (Type-C-FP at lateral admission; Type-P-FP at promotion), derive multiplicity-aware bounds on the family-wise cascade probability across rungs under sub-Gaussian improvements and width-aware bars (Sec. 4.4), and instrument benchmarks with ground-truthable oracles to estimate (i) specious-promotion rate, (ii) cascade depth, and (iii) compute share spent on ultimately inconsistent branches.

We will also evaluate drop-in mitigations that preserve the pseudolinear lateral cost: (i) holdout confirmations at promotion time (one micro-probe) to reduce selective-inference bias; (ii) path-consistency aggregation (e.g., quantile-of-c along the path) at promotion; and (iii) disjoint verifier prompts or cross-model checks for repeat-to-confirm under fixed micro-budgets. Sensitivity studies will sweep initial lateral width, overflow cap, confirmation budgets, and threshold margins to map robustness frontiers and accuracy–latency Pareto curves. The goal is a statistically disciplined account of when laterals with spurious early c signals can be nurtured by envelope dynamics—and how to bound such cascades without sacrificing the wide-and-short advantages established here.