A Measure-Theoretic Analysis of Reasoning in Transformers: Structural Generalization and Approximation Limits

Yuyang Zhang
McGill University
yuyang.zhang@mail.mcgill.ca

Yifu Zhang*
McGill University
yifu.zhang3@mail.mcgill.ca

Xuehai Zhou*
McGill University
xuehai.zhou@mail.mcgill.ca

Xiaoyin Chen†
Mila – Quebec AI Institute
xiaoyin.chen@mila.quebec

Abstract

While empirical scaling laws for reasoning in large language models (LLMs) are well established, it remains unclear when a model trained on one class of search traces can generalize to structurally different out-of-distribution (OOD) problems. We study this question by projecting each discrete search trajectory into a twelve-dimensional space of structural coordinates—such as tree depth and backtracking rate—and quantifying task shifts via the Wasserstein-1 (W_1) distance. Using Kantorovich duality, we bound OOD generalization risk in terms of this transport distance, the architecture's structural Lipschitz constant, and capacity lower bounds from network depth and width. This analysis yields two main architectural conclusions. First, under absolute positional encoding (APE), translating a reasoning subtrajectory to later token positions almost surely alters its internal attention scores. When these perturbations misroute greedy exact-match search, the structural Lipschitz constant is $\Omega(1)$, and target risk has an order-one lower bound whenever misrouted structures carry enough target mass to exceed the approximation error by a fixed margin. By contrast, rotary position embedding (RoPE) preserves local attention scores under translation for fixed content vectors; under our baseline smoothness assumption, its transport penalty scales as $\mathcal{O}(W_1/T_{\max})$, where $T_{\max}$ is the maximum sequence length. Second, by reducing sequential backtracking to a Dyck-k language under an implicit-depth assumption, we prove that fixed-depth Transformers face an irreducible risk floor when sufficiently many target searches exceed a multiple of their layer count. Adding layers removes this bottleneck, whereas widening the network only reduces a separate Barron-space term in the risk lower bound. Experiments on 54 Transformers trained from scratch on Countdown support these predictions. Further evaluations in the appendix extend our framework to broader settings: we quantify structural shifts across public agent, embodied, and robotic trajectories; evaluate 33 open-weight LLMs under global position-index shifts; and test when splitting nested tasks across model calls improves final-answer accuracy.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Chain-of-thought prompting [Wei et al., 2022, Yao et al., 2023] and post-training methods such as reinforcement learning [Lightman et al., 2024, OpenAI, 2024a, Guo et al., 2025, Lambert et al., 2025] enable large language models (LLMs) to solve complex problems by generating explicit, multi-step reasoning traces. In formal domains such as combinatorial planning and mathematical problem-solving, these traces represent serialized traversals of an implicit search tree. Despite strong empirical scaling laws for reasoning, a central theoretical question remains open: *under what architectural conditions can an autoregressive model generalize its search capabilities across structural domain shifts?* When a model trained on shallow search traces is evaluated on deeper, backtracking-heavy out-of-distribution (OOD) trees, standard token-level log-likelihoods and zero-one task losses fail to capture the structural divergence between training and test distributions.

Prior theoretical and empirical studies have largely examined the relevant factors in isolation. Circuit-complexity bounds place distribution-agnostic limits on sequential reasoning by restricting fixed-depth, bounded-precision Transformers to TC^0 [Hahn, 2020, Merrill and Sabharwal, 2023, Feng et al., 2023]. Work on length extrapolation compares positional mechanisms such as absolute positional encoding (APE) and rotary position embedding (RoPE) [Kazemnejad et al., 2023, Ruoss et al., 2023], yet rarely accounts for circuit depth limits. Conversely, data-centric studies analyze distribution shifts in reasoning traces while treating the network architecture as a black box [Zhao et al., 2026, Prystawski et al., 2023]. Determining when a reasoning model will generalize under structural shift requires analyzing how circuit capacity, positional encodings, and data distributions interact.

Our Framework. We introduce a measure-theoretic framework that connects four dimensions of LLM reasoning: physical layer depth (L), representation width (m), positional encoding (PE), and continuous distributional shift. Projecting each variable-length search trajectory into a continuous metric space $\mathcal{X} \subset \mathbb{R}^{12}$ defined by twelve structural statistics—such as tree depth and backtracking rate—represents each reasoning dataset as a pushforward probability measure on $\mathcal{X}$, whose task shift is quantified by the Wasserstein-1 (W_1) transport distance.

To link this continuous transport distance to discrete zero-one risk, we introduce an ϵ-relaxed structural sufficiency condition (Assumption 1), under which a single structural risk function $\mathcal{L}_f : \mathcal{X} \to [0, 1]$ approximates the model's mean failure rate on each task within ϵ. Kantorovich duality then bounds the risk change between tasks by $K_{f,\mathrm{PE}} W_1 + 2\epsilon$, where $K_{f,\mathrm{PE}}$ is the Lipschitz constant of $\mathcal{L}_f$. Combining this transport bound with capacity lower bounds from network width and circuit depth yields Theorem 2: target risk is bounded below by the maximum of the transport-adjusted source risk, a Barron-space width term, and a TC^0 depth term, so any single active bottleneck keeps target risk bounded away from zero. This unified bound yields two main architectural results.

Translation Equivariance and the Lipschitz Gap. In tree search, an identical subtree traversal can appear at different absolute token positions depending on how many branches were explored beforehand. Whereas APE indexes tokens by absolute position, RoPE parameterizes its attention kernel by relative offsets [Su et al., 2024, Zhao et al., 2024] (see Appendix F for related schemes). We prove that APE breaks translation symmetry: shifting a sub-trajectory to later positions almost surely alters its internal attention scores (Lemma 2). When such perturbations misroute greedy exact-match search (Assumption 2), APE incurs a structural Lipschitz constant of $K_{f,\mathrm{APE}} = \Omega(1)$, leading to $\Omega(1)$ OOD risk whenever misrouted structures carry enough target mass to exceed ϵ by a fixed margin (Theorem 1, Corollary 1). In contrast, RoPE preserves local scores under translation for fixed content vectors, maintaining $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$ under our baseline smoothness assumption.

Circuit Depth vs. Width in Sequential Backtracking. Building on the TC^0 expressivity limits of bounded-precision Transformers [Merrill and Sabharwal, 2023], we establish a depth-dependent risk floor for recursive search. Mapping sequential backtracking onto the Dyck-k parenthesis language links maximum tree depth χ_1 directly to parenthesis nesting depth (Lemma 3). Motivated by limits on fixed-depth self-attention on hierarchical languages [Hahn, 2020], we assume that the layer count L of a TC^0 Transformer caps the maximum tree depth resolvable in a single forward pass at $\alpha_{\mathrm{circ}} L$ (Assumption 5). Scaling physical depth L is thus necessary to avoid an $\Omega(1)$ risk floor when sufficient target mass lies on deep trees; increasing hidden width m cannot remove this barrier, as it only reduces a separate Barron-space lower-bound term of order $m^{-1/2}$ (Lemma 4; cf. Barron [1993], Makovoz [1996]).

Empirical Verification. We test these theoretical predictions on the Countdown combinatorial planning task using a controlled grid of 54 Transformers trained from scratch across three parameter scales (Small, Medium, Large), three aspect ratios (Deep, Wide, Balanced), and two positional encodings (APE vs. RoPE). Models are trained on traces generated via breadth-first search (BFS, emphasizing lateral expansion), depth-first search (DFS, emphasizing recursive backtracking), or a randomized mixture (MIXED) that chooses BFS or DFS with equal probability and varies the search heuristic. Our experiments support four predictions:

- TC^0 **Depth Ceiling:** On the backtracking-heavy DFS evaluation set, all 54 models fall below the TC^0 ceiling motivated by Lemma 5 and fitted to these same accuracies (Appendix G.3). At fixed depth, doubling the hidden width changes peak accuracy by less than 1 percentage point.
- **Depth–Width Asymmetry:** Scaling parameters via width alone leads to plateaus or degradation. Under APE, Deep architectures outperform Wide ones across all 18 Medium- and Large-scale comparisons by up to 11.25 points—a gap that RoPE narrows to under 1 point at the Large scale.
- **Stability under Positional Shifts:** Under the position shifts of the MIXED distribution, scaling Wide APE models to the Large scale triggers a sharp accuracy drop (42.15%), whereas matched Large Wide RoPE models remain stable (54.25%).
- **Consistent Advantage of RoPE:** Overall, RoPE outperforms APE in 25 of 27 matched model pairs ($p < 10^{-5}$, one-sided sign test; winning 71 of 81 evaluation comparisons), including all 18 Medium- and Large-scale pairs, with its advantage growing with model scale.

Finally, we extend our evaluation to agent trajectories and pretrained LLMs in Appendix M. When Qwen3-8B attempts the same SWE-bench Verified issues under mini-SWE-agent and OpenHands, we observe a Wasserstein-1 shift on every structural coordinate and a rise in resolve rate from 1.7% to 6.4% (Appendix M.2.2). Testing 33 open-weight LLMs under the position-index shifts routinely introduced by serving infrastructure (left padding, prefix caching, and sliding windows) shows that shifting by $+256$ positions degrades APE accuracy by up to 52 percentage points, while leaving pooled RoPE and ALiBi accuracy virtually unchanged in fp32 when token distances are preserved (at most one net answer change per 3,000; Appendix M.3.2). Lastly, decomposing depth-8 nested tool calls into single-level sub-calls boosts the accuracy of eight open-weight LLMs from 1.5–15.5% to 43.5–99% (Appendix M.4.2).

2 Preliminaries and Measure-Theoretic Formulation

We model a reasoning task as a distribution over problem instances paired with reference search trajectories. To compare tasks, we define the trajectory-level zero-one risk of an autoregressive model and project variable-length trajectories into a shared space of structural statistics; full constructions and proofs appear in Appendices A and B.

2.1 The Measurable Space of Search Trajectories

Let $\mathcal{Z}$ be the finite set of grammar-valid search trajectories over vocabulary $\mathcal{V}$ with length at most $T_{\max}$. Each trajectory $z = x_{\text{init}}^{(z)} \oplus y_{\text{search}}$ concatenates an initial problem prompt $x_{\text{init}}^{(z)}$ with a reference search trace y_{search}. We equip $\mathcal{Z}$ with its discrete σ-algebra and consider source and target measures μ_S, μ_T. Given only $x_{\text{init}}^{(z)}$, a Transformer $f \in \mathcal{F}_{L,m,\text{PE}}$ with L layers, hidden width m, and positional encoding PE generates a candidate trace $f(x_{\text{init}}^{(z)})$. A deterministic verifier S_z checks whether this trace solves the problem ($S_z > 0$), inducing the zero-one loss

$$\ell(f, z) = 1 - \mathbb{I}_{(0,\infty)}\left(S_z\left(f(x_{\text{init}}^{(z)})\right)\right). \tag{1}$$

For any measure μ, the expected task risk is $\mathcal{R}_\mu(f) = \int_{\mathcal{Z}} \ell(f, z)\, d\mu(z)$.

2.2 Structural Projection and Pushforward Measures

We equip the compact structural space $\mathcal{X} \subset \mathbb{R}^{12}$ spanning the attainable coordinate vectors with the Euclidean metric $d_{\mathcal{X}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2$; its coordinate ranges yield $\operatorname{diam}(\mathcal{X}) = \mathcal{O}(T_{\max})$ (Appendix A.1).

Definition 1 (Structural Projection Operator Φ)**.** *We define a deterministic, Borel-measurable projection operator $\Phi : \mathcal{Z} \to \mathcal{X}$ that maps each trajectory $z \mapsto \boldsymbol{\chi} = [\chi_1, \ldots, \chi_{12}]^\top$ into three groups of summary statistics (formally defined in Appendix A.1): Tree geometry and volume: the maximum, mean, and standard deviation of the search tree depth (χ_1, χ_2, χ_3), along with total search volume χ_4 (the number of node references in the trace); Transition dynamics: the fractions of consecutive node transitions that descend deeper (χ_5, deepening rate), backtrack upward (χ_6, backtracking rate), or move laterally at the same depth (χ_7, lateral transition rate); Pruning and task semantics: the total prune count and pruning rate (χ_8, χ_9), a binary indicator of whether the search reaches the target (χ_{10}), and the counts of multiplicative (χ_{11}) and additive (χ_{12}) operators.*

Pushing the source and target trajectory distributions μ_S and μ_T forward through Φ yields the structural pushforward measures $\nu_S := \Phi_{\#}\mu_S$ and $\nu_T := \Phi_{\#}\mu_T$ in $\mathcal{P}(\mathcal{X})$. We quantify the structural domain shift between tasks via the Wasserstein-1 (Kantorovich–Rubinstein) distance [Villani, 2009]:

$$W_1(\nu_S, \nu_T) = \inf_{\pi \in \Pi(\nu_S, \nu_T)} \int_{\mathcal{X} \times \mathcal{X}} d_{\mathcal{X}}(\mathbf{u}, \mathbf{v}) \, \mathrm{d}\pi(\mathbf{u}, \mathbf{v}), \tag{2}$$

where $\Pi(\nu_S, \nu_T)$ is the set of joint couplings on $\mathcal{X} \times \mathcal{X}$ with marginals ν_S and ν_T. Because $\mathcal{X}$ is bounded for any finite $T_{\max}$, this transport cost is finite and satisfies $W_1(\nu_S, \nu_T) = \mathcal{O}(T_{\max})$.

Why a structural projection? Token-level likelihoods do not distinguish search strategies: a shallow breadth-first trace and a deep backtracking trace can share similar token perplexity while executing different search algorithms. Projecting trajectories onto $\mathcal{X}$ abstracts away surface token identities, representing each reasoning task by its distribution of structural behaviors so that strategies can be compared directly via $W_1(\nu_S, \nu_T)$.

2.3 Error Decomposition via Kantorovich Duality

We relate the continuous domain shift $W_1(\nu_S, \nu_T)$ to the discrete expected risk $\mathcal{R}_\mu(f)$ through a sufficiency condition under which, up to ϵ, this risk depends on μ only through $\Phi_{\#}\mu$.

Assumption 1 (ϵ-Relaxed Structural Sufficiency)**.** *Let $\epsilon > 0$ be an irreducible approximation error. We assume there exists a measurable risk function $\mathcal{L}_f : \mathcal{X} \to [0, 1]$ such that for any trajectory measure $\mu \in \{\mu_S, \mu_T\}$, the absolute difference between the true expected risk on $\mathcal{Z}$ and the structural risk integrated over $\Phi_{\#}\mu$ on $\mathcal{X}$ is bounded by ϵ:*

$$\left| \mathcal{R}_\mu(f) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi}) \, d(\Phi_{\#}\mu)(\boldsymbol{\chi}) \right| \leq \epsilon. \tag{3}$$

Assumption 1 posits that a single structural risk function $\mathcal{L}_f$ approximates the model's mean failure rate on each task within ϵ, contributing at most 2ϵ when comparing two risks. In formal planning, reference search depth and backtracking frequency capture much of a problem's difficulty, while instance-specific details omitted by Φ enter the residual ϵ (Appendix A.2).

Appendix J tests this condition across two Countdown models differing in scale, training strategy, and positional encoding. A gradient-boosting classifier trained on the twelve coordinates $\boldsymbol{\chi}$ alone predicts per-problem success with an out-of-fold ROC-AUC of about 0.93, outperforming either trace length or the problem specification in every fold (mean McFadden pseudo-R^2 of 0.48–0.51, versus 0.44–0.46 for length and 0.40–0.41 for problem specification). When a classifier fitted on one search strategy is evaluated on another, its mean predicted failure probability directly estimates target risk.

Lemma 1 (ϵ-Relaxed Error Decomposition)**.** *Assume the induced risk function $\mathcal{L}_f(\boldsymbol{\chi})$ is Lipschitz continuous on $(\mathcal{X}, d_{\mathcal{X}})$ with minimal Lipschitz constant $K_f = \sup_{\boldsymbol{u} \neq \boldsymbol{v}} |\mathcal{L}_f(\boldsymbol{u}) - \mathcal{L}_f(\boldsymbol{v})| / d_{\mathcal{X}}(\boldsymbol{u}, \boldsymbol{v})$. Then the generalization error gap satisfies*

$$|\mathcal{R}_{\mu_T}(f) - \mathcal{R}_{\mu_S}(f)| \leq K_f \cdot W_1(\nu_S, \nu_T) + 2\epsilon. \tag{4}$$

Proof Sketch. The triangle inequality and Assumption 1 bound $|\Delta\mathcal{R}|$ by $|\int \mathcal{L}_f \, d\nu_T - \int \mathcal{L}_f \, d\nu_S| + 2\epsilon$. If $K_f = 0$, the structural integrals agree. For $K_f > 0$, the normalized function $g = \mathcal{L}_f / K_f$ satisfies $\|g\|_{\mathrm{Lip}} \leq 1$, so Kantorovich–Rubinstein duality [Villani, 2009] bounds $|\int g \, d\nu_T - \int g \, d\nu_S|$ by $W_1(\nu_S, \nu_T)$. Multiplying by K_f yields the claim (full proof in Appendix B.2). □

3 Theoretical Limits of Architectural Choices

We next examine how positional encodings govern the Lipschitz constant K_f under sequence translations, and how width m and depth L constrain single-pass capacity on recursive backtracking.

3.1 Sequence Equivariance and the Lipschitz Gap

Search trajectories are modular: the same local sub-trajectory (e.g., a subtree traversal) can occur at arbitrary token indices depending on how much of the tree was explored before it. To quantify sensitivity to these offsets, we analyze the attention kernel under sequence translations.

Definition 2 (Minimal Sequence Translation Operator $\mathcal{T}_k$)**.** *Let $z \in \mathcal{Z}$. For admissible insertions that preserve the grammar and length bound, $\mathcal{T}_k$ ($k \in \mathbb{N}^+$) inserts one structurally inert node of k tokens immediately before a local sub-trajectory ω of z.*

Under Φ, this increments the search volume ($\chi_4' = \chi_4 + 1$) while shifting every token in ω by $+k$ in absolute position. Hence the distance in $\mathcal{X}$ satisfies

$$d_{\mathcal{X}}(\Phi(z), \Phi(\mathcal{T}_k(z))) = \left(1^2 + \sum_{j \neq 4} (\Delta\chi_j)^2\right)^{1/2} =: \delta \leq C_\delta, \tag{5}$$

where C_δ is independent of k and $T_{\max}$.

Holding the input content vectors fixed, we evaluate the pre-softmax attention score $A_{a,b}$ between two tokens in ω originally at indices a and b, now shifted to $a + k$ and $b + k$.

Lemma 2 (APE Absolute Coordinate Sensitivity)**.** *Assume the network parameters (projection matrices $\mathbf{W}_Q, \mathbf{W}_K$ and absolute embeddings $\mathbf{p}_t$) are generically parameterized, with $\mathbf{W}_Q^\top \mathbf{W}_K \neq \mathbf{0}$. Under $\mathcal{T}_k$ ($k > 0$), the local attention kernel of an APE model changes almost surely: $\Delta A^{\mathrm{APE}} = |\widetilde{A}^{\mathrm{APE}}_{a,b} - A^{\mathrm{APE}}_{a,b}| > 0$.*

Proof Sketch. Expanding the bilinear forms of the APE attention kernel cancels the position-independent terms, leaving a nonzero polynomial in the projection matrices and the distinct position embeddings. Because its zero set has Lebesgue measure zero, the scores differ almost surely. By contrast, RoPE uses orthogonal rotation matrices satisfying $(\mathbf{R}_\Theta^{a+k})^\top \mathbf{R}_\Theta^{b+k} = \mathbf{R}_\Theta^{b-a}$, giving $\Delta A^{\mathrm{RoPE}} = 0$ (proof in Appendix C.2). □

Consistent with Lemma 2, shifting all position indices by 16, 64, or 256 across eight pretrained models (Experiment 4, Appendix M.3.1) alters peak first-layer scores in the four APE models by 0.42 to 1.19 times the maximum absolute unshifted score, whereas the largest relative change across all layers in the four RoPE models remains below 4.5×10^{-12} in float64.

Assumption 2 (Autoregressive Sensitivity via Markov Compounding)**.** *We assume the search task has baseline structural smoothness $\mathcal{O}(1/T_{\max})$, inherited by models with translation-invariant local kernels (Appendix C). Because generating a search trajectory is a Markov decision process over a directed acyclic graph, we assume that, for every sufficiently large $T_{\max}$, there is an admissible translation for APE whose local attention-kernel perturbation ($\Delta A > 0$) at a critical routing step diverts greedy decoding into an incorrect subtree from which the target state cannot be reached under exact-match evaluation. The structural risk therefore changes by at least the uniform guessing error rate: $\Delta\mathcal{L} = |\mathcal{L}_f(\boldsymbol{\chi}') - \mathcal{L}_f(\boldsymbol{\chi})| \geq 1 - 1/|\mathcal{V}| =: \gamma > 0$.*

Theorem 1 (The Lipschitz Gap)**.** *Under Assumption 2, for a sufficiently large sequence capacity $T_{\max}$, the Lipschitz constant of the induced risk function for APE is asymptotically larger than that of RoPE: $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$ and $K_{f,\mathrm{APE}} = \Omega(1)$.* (Proof in Appendix C.5.)

From positional encoding to generalization risk. Because $\mathcal{L}_f$ depends on f, the positional encoding directly governs the smoothness of $\mathcal{L}_f$ over $\mathcal{X}$. Applying $\mathcal{T}_k$ shifts every token in a sub-trajectory ω by k positions while perturbing $\boldsymbol{\chi}$ by at most $\delta \leq C_\delta$. Under APE, this index shift alters internal attention scores almost surely (Lemma 2), which in exact-match search can divert greedy decoding into an incorrect subtree (Assumption 2). A bounded displacement δ thus induces an $\Omega(1)$ change in risk, forcing $K_{f,\mathrm{APE}} = \Omega(1)$. Substituted into Lemma 1, this yields a constant-order lower bound on the coefficient of the transport penalty $K_f W_1(\nu_S, \nu_T)$, whereas RoPE's translation invariance maintains $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$.

3.2 Capacity Limits: The TC^0 Algorithmic Bottleneck

Whereas breadth-first search expands sibling branches level by level, recursive backtracking follows a last-in-first-out (LIFO) order: after a dead end, the model must return to the most recent unfinished branch in the search trace. The Dyck-k language $\mathcal{D}_k$ of balanced strings over k bracket types obeys the same LIFO structure.

Lemma 3 (Surjective Monoid Homomorphism to Dyck-k)**.** *Let k denote the maximum branching factor of the implicit search tree. Defining the traversal alphabet $\Sigma_{\mathrm{search}} = \{\delta_{\mathrm{down}}^{(1)}, \ldots, \delta_{\mathrm{down}}^{(k)}\} \cup \{\delta_{\mathrm{up}}^{(1)}, \ldots, \delta_{\mathrm{up}}^{(k)}\} \cup \{\omega_{\mathrm{eval}}\}$, there exists a surjective monoid homomorphism $\psi : \Sigma_{\mathrm{search}}^* \to \Sigma_{\mathrm{Dyck}}^*$ mapping child expansions to left parentheses $(_c$ and backtracking steps to right parentheses $)_c$. Every valid search trajectory projects to a prefix of a word in $\mathcal{D}_k$, where the maximum tree depth χ_1 determines the maximum parenthesis nesting depth.*

Circuit bounds for bounded-precision Transformers [Merrill and Sabharwal, 2023] and expressivity limits on hierarchical self-attention [Hahn, 2020] motivate our implicit-depth assumption (Appendix D.2.2). Let $\mathcal{L}^*(\boldsymbol{\chi}) = \inf_h \mathbb{E}_{z\sim\mu_T}[\ell(h, z) \mid \Phi(z) = \boldsymbol{\chi}]$ denote the Bayes-optimal structural risk (Appendix D.1).

Lemma 4 (Width Capacity Bound via Barron Spaces)**.** *Under Assumptions 1 and 4 (in particular $\mathcal{L}^* \in \mathcal{H}_{\mathrm{Barron}}$), any autoregressive Transformer hypothesis bounded by an internal representation width m satisfies the lower bound: $\mathcal{R}_{\mu_T}(f) \geq \frac{C_{\mathrm{width}}}{\sqrt{m}} - \epsilon$.* (Proof in Appendix D.1.1.)

Each routing decision during autoregressive generation is computed in one forward pass of L layers; under Assumption 5, the nesting depth it can resolve is at most $\alpha_{\mathrm{circ}} L$.

Lemma 5 (Implicit Depth Bottleneck Bound)**.** *Under Assumptions 1 and 5, defining the Borel measurable indicator function $P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}}) = \mathbb{I}_{(\alpha_{\mathrm{circ}} L, \infty)}(\chi_1)$, the expected target risk is bounded below by: $\mathcal{R}_{\mu_T}(f) \geq C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}\, d\nu_T - \epsilon$, where $C_{\mathrm{depth}} = 1 - \frac{1}{|\mathcal{V}|} \in (0, 1)$.*

4 The Main Bound

Theorem 2 (The Generalization Bound)**.** *Under Assumptions 1, 4, and 5 and a Lipschitz $\mathcal{L}_f$, for any autoregressive Transformer $f \in \mathcal{F}_{L,m,\mathrm{PE}}$, the target risk $\mathcal{R}_{\mu_T}(f)$ is bounded below by the maximum of three lower bounds, and from above by its source risk and domain shift penalty:*

$$\mathcal{R}_{\mu_T}(f) \geq \max\Big\{ \frac{C_{\mathrm{width}}}{\sqrt{m}} - \epsilon,\ C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}\, d\nu_T - \epsilon,\ \mathcal{R}_{\mu_S}(f) - K_{f,\mathrm{PE}}\, W_1(\nu_S, \nu_T) - 2\epsilon \Big\} \quad \text{(LB)}$$

$$\mathcal{R}_{\mu_T}(f) \leq \mathcal{R}_{\mu_S}(f) + K_{f,\mathrm{PE}}\, W_1(\nu_S, \nu_T) + 2\epsilon \quad \text{(UB)}$$

Proof Sketch. Kantorovich duality applied to the ϵ-relaxed structural risk (Lemma 1) gives (UB) and the measure-shift term of (LB); the Barron-space width bound (Lemma 4) and the implicit-depth bound (Lemma 5) give the other two terms (Appendix E). □

Why the bound takes a maximum. Because all three terms in (LB) constrain the same target risk $\mathcal{R}_{\mu_T}(f)$, their maximum reflects whichever bottleneck is active. Suppose, for instance, that a non-vanishing share of target searches reaches depth 20, exceeding the single-pass limit $6\alpha_{\mathrm{circ}}$ of a six-layer model. Widening the network reduces the width term, and using a translation-equivariant encoding reduces the transport penalty, yet neither change alters the depth term; for sufficiently small ϵ, those deep searches still impose an $\Omega(1)$ risk floor (Appendix I).

Corollary 1 (The Divergent OOD Guarantees of APE vs. RoPE)**.** *Let $\mathcal{R}_{\mu_S}(f_{\mathrm{RoPE}}) = \epsilon_{\mathrm{train}} \approx 0$ and $W_1(\nu_S, \nu_T) = \Omega(T_{\max})$. RoPE preserves translation equivariance: with $K_{f,\mathrm{RoPE}} \leq C_{\mathrm{smooth}}/T_{\max}$ (Theorem 1), its expected risk degrades at most proportionally to the normalized shift, $\mathcal{R}_{\mu_T}(f_{\mathrm{RoPE}}) \leq \epsilon_{\mathrm{train}} + C_{\mathrm{smooth}} \cdot (W_1/T_{\max}) + 2\epsilon$. Under APE, position-induced routing failures yield an $\Omega(1)$ expected risk: $\mathcal{R}_{\mu_T}(f_{\mathrm{APE}}) \geq \gamma \cdot p_{\mathrm{shift}} - \epsilon$, where p_{shift} is the target mass of structures on which routing fails ($\mathcal{L}_{f_{\mathrm{APE}}} \geq \gamma$; Assumption 2) and $\gamma p_{\mathrm{shift}} - \epsilon$ is bounded away from zero.*

Corollary 2 (The "Deep is Better" Law for Search Reasoning)**.** *Suppose the target measure ν_T assigns enough mass to trajectories satisfying $\chi_1 > \alpha_{\mathrm{circ}} L$ that the depth term in (LB) is bounded away from zero. Increasing the layer count to $L \geq \alpha_{\mathrm{circ}}^{-1} \sup_{\boldsymbol{\chi} \in \mathrm{supp}(\nu_T)} \chi_1$ eliminates this depth floor, whereas increasing width reduces only the width term, since $\lim_{m\to\infty}(C_{\mathrm{width}}/\sqrt{m} - \epsilon) = -\epsilon \leq 0$ and the depth term is independent of m.*

5 Empirical Verification

Experimental Design and Architecture Grid. To test the architectural predictions of Theorem 2, we vary layer depth L, hidden width m, and positional encoding while holding the core parameter budget ($12Ld^2$) nearly constant within each scale group. Motivated by depth-separation theorems [Eldan and Shamir, 2016, Levine et al., 2020] and TC^0 complexity bounds [Merrill and Sabharwal, 2023], we construct nine Transformer backbones spanning three parameter scales (Small, Medium, Large) and three aspect ratios (Deep, Wide, Balanced). To compare positional encodings at fixed model dimensions, we train each backbone with both APE and RoPE [Su et al., 2024], implemented via GPT-Neo and GPT-NeoX, respectively, which also differ in attention scaling and embedding tying (Appendix K.2). Each model is trained for three epochs with seed 42; crossing these 18 architectures with three training distributions yields the 54 models in Table 1 (Appendix H).

Task and Search Distributions. We use the Stream of Search (SoS) format [Gandhi et al., 2024] for Countdown to define the search grammar Γ and generate trajectories for the source and target measures (μ_S, μ_T). In Countdown, a model combines input numbers via basic arithmetic operations to reach a target value (e.g., reaching 10 from $[2, 3, 5]$). SoS serializes the explored search tree—including dead ends and backtracking steps—into a single autoregressive token sequence (Appendix K.4). At test time, the model receives only the input numbers and target, and the original SoS evaluator scores the path reported in the model's trace, which may differ from the reference search (Appendix K).

By varying the search algorithm in the SoS simulator, we generate $500{,}000$ ground-truth training trajectories for each of three structural measures: μ_{BFS} (Breadth-First Search), which expands states via a priority queue; μ_{DFS} (Depth-First Search), which uses recursive stack expansion with nearly four times the backtracking rate (χ_6) of μ_{BFS}; and μ_{MIXED} (Structural Mixture), which chooses BFS or DFS with equal probability and randomizes the search heuristic, exposing models to varied token positions. All models are evaluated on the same test problems: the BFS and DFS sets each take the first $2{,}000$ problems from the corresponding training set, and the MIXED set takes $1{,}000$ from each (Appendix K.5).

Estimating Structural Optimal Transport. To measure how strongly these search strategies shift the reference distribution on $\mathcal{X}$, we draw $20{,}000$ trajectories per strategy, standardize the twelve coordinates across all three pools, and draw $2{,}000$ trajectories per side without replacement. Solving the transport problem exactly with POT [Flamary et al., 2021] over ten random draws yields a baseline distance of 0.19 ± 0.06 between disjoint BFS samples. Switching from BFS to MIXED yields 1.38 ± 0.05, and switching to DFS yields 2.97 ± 0.05—both exceeding the baseline by more than sevenfold, with MIXED lying at roughly 46% of the BFS-to-DFS distance (Appendix G).

5.1 Positional Encodings as Structural Regularizers

We first examine the positional sensitivity predicted by Theorem 1. A model trained on MIXED encounters similar search steps at widely varying absolute token positions across BFS and DFS traces. Figure 1C compares Wide models across all three parameter scales on MIXED: under APE, scaling to the Large scale ($L = 12, d = 1{,}536$) causes accuracy to drop to 42.15%, whereas the matched Large Wide RoPE model reaches 54.25%—a gain of 12.1 percentage points. The peak

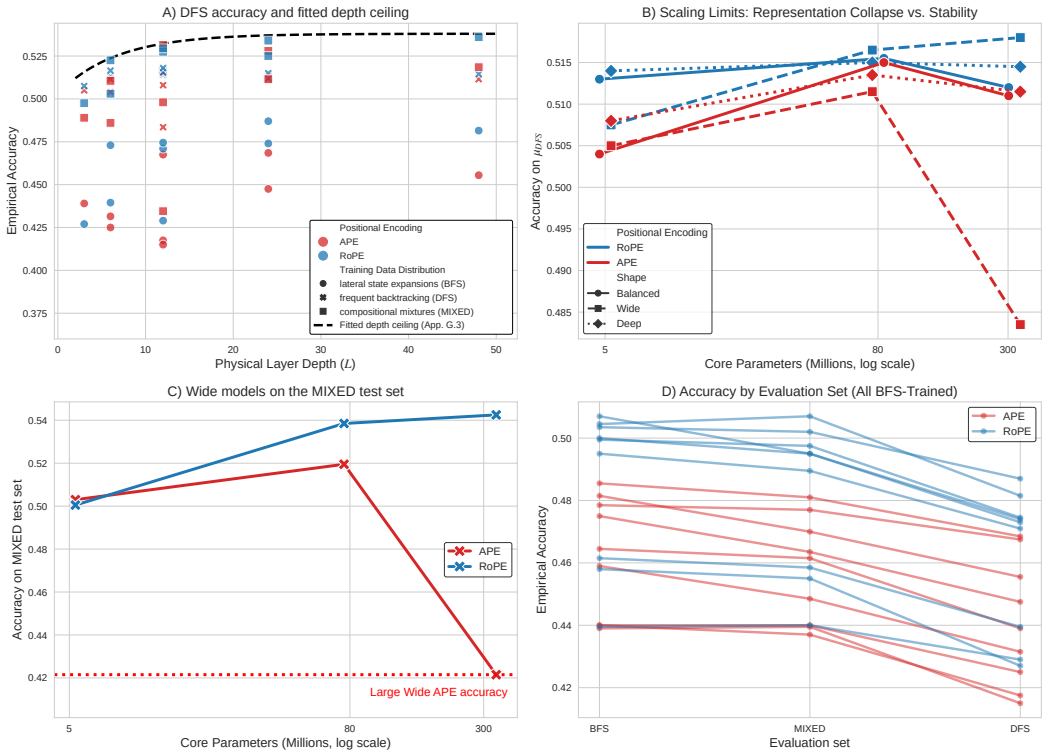


Figure 1: Countdown accuracy across model shapes and search strategies. **(A)** DFS accuracy for all 54 models, plotted with the TC^0 depth ceiling fitted to those accuracies (Appendix G.3). **(B)** Under DFS training and evaluation, Large Deep APE (51.15%) outperforms Large Wide APE (48.35%), whereas under RoPE the two shapes differ by less than 1 point. **(C)** Under MIXED training and evaluation, Wide APE accuracy drops at Large scale (42.15%) while Wide RoPE remains stable (54.25%). **(D)** BFS-trained models on the BFS, MIXED, and DFS evaluation sets, ordered by the Wasserstein distance $\widehat{W}_1$ between each strategy's training reference searches and those of BFS. RoPE wins 23 of 27 comparisons and achieves higher mean accuracy on each set; all three sets sample problems from the same population (Appendix K.5).

accuracies at 12, 24, and 48 layers lie within 1 point of each other on every Large-scale evaluation set (Appendix I).

Across the entire grid, RoPE outperforms APE in 71 of 81 paired evaluation comparisons and in 25 of 27 matched model pairs when averaged across test sets ($p < 10^{-5}$, one-sided sign test). The mean accuracy gain with RoPE grows from $+0.6$ points (Small) to $+1.2$ points (Medium) and $+3.3$ points (Large), with RoPE winning all 27 comparisons at the Large scale. Figure 1D illustrates this consistent advantage for BFS-trained models across all three evaluation sets, supporting the role of RoPE as a structural regularizer on $\mathcal{X}$.

Direct Intervention on Position Indices. To test the routing mechanism of Lemma 2 and Assumption 2 directly, we hold the trained model and input tokens fixed while shifting every position index by $+16$ or $+64$ (Experiment 6, Appendix M.3.3). Under teacher forcing on reference trajectories, we compare next-token predictions across the 94,278 routing tokens that specify the next node or arithmetic operation: Small Deep MIXED RoPE repeats 100% of its original choices at both offsets, whereas Large Deep BFS APE changes 25.7% of its choices at $+16$ and 28.3% at $+64$. We next locate an active leaf node with an unexplored operation in each model's own generated search, resume generation from that point at the original or shifted position indices, and check whether the completed search reaches the target. Every resumed RoPE search retains its original outcome (59.4% solved), whereas APE success falls from 52.7% to 12.2% at $+16$ and 13.4% at $+64$, confirming that local index shifts can misroute APE search and alter the final answer.

5.2 Depth versus Width Scaling and TC^0 Bottlenecks

Next, we examine the theoretical asymmetry between depth and width scaling. While resolving recursive backtracking bottlenecks requires increasing physical layer depth L (Lemma 5), widening the hidden dimension m reduces only the $m^{-1/2}$ Barron-space term in the risk lower bound (Lemma 4) and cannot remove the depth floor.

Our results under APE reflect this asymmetry: Deep architectures beat Wide ones across all 18 Medium- and Large-scale comparisons. At a matched core parameter budget, the Large Deep MIXED APE model (0.5330) outperforms its Wide counterpart (0.4215) by 11.15 percentage points. Two 12-layer models already reach this accuracy: Large Wide RoPE (0.5425) and the half-width Medium Balanced APE model (0.5360). Why, then, does the 48-layer Large Deep APE model outperform the 12-layer Large Wide APE model by this margin? As analyzed in Appendix H, the extra layers in Deep APE may provide routing redundancy that compensates for the position-index sensitivity seen in Wide APE networks.

Replacing APE with RoPE removes this positional sensitivity and allows Wide architectures to match Deep ones (Figure 1B). At the Large scale, RoPE reduces the gap between Deep and Wide models to under 1 percentage point across all nine comparisons (versus 2.6–11.25 points under APE), consistent with translation equivariance allowing a Transformer to approach the capacity set by its depth and width without positional degradation (Appendix I).

5.3 Testing the Width and Depth Predictions of the Unified Bound

Finally, we test the width and depth predictions of Theorem 2 on the backtracking-heavy DFS evaluation set. Figure 1A plots the DFS accuracies of all 54 models alongside the empirical TC^0 depth ceiling fitted to these scores, which bounds all 54 points from above (Appendix G.3). To test Corollary 2, we hold layer depth fixed and compare the best-performing model among the six at each shape on μ_{DFS}: doubling the hidden width changes peak accuracy by less than 1 percentage point. Specifically, at $L = 12$, doubling width from $d = 768$ to $d = 1{,}536$ (quadrupling core parameters from 84.9M to 339.7M) changes peak accuracy from 0.5315 to 0.5295; at $L = 24$, doubling width from $d = 512$ to $d = 1{,}024$ changes peak accuracy from 0.5280 to 0.5340 (see Appendix G.3 for $L = 6$).

The horizontal asymptote of the empirical ceiling in Figure 1A also reflects task constraints outside the depth term, primarily the evaluation generation budget. All models are evaluated via greedy decoding with a budget of 512 new tokens across the $2{,}000$ test problems per distribution (Appendix K). Within this 512-token limit, the ground-truth reference searches themselves reach the target on 46.45%–49.15% of the problems—a rate that most of our trained models surpass (Appendix G.3).

6 Related Work

Circuit complexity and Transformer expressivity. Prior theoretical work establishes that fixed-depth, bounded-precision Transformers lie within TC^0 [Merrill and Sabharwal, 2023], and that self-attention has expressivity limits on hierarchical languages such as Dyck when depth and head count remain fixed as sequence length grows [Hahn, 2020]. Classical depth-separation theorems further exhibit functions for which finite depth cannot be efficiently compensated by polynomial increases in width [Eldan and Shamir, 2016, Levine et al., 2020]. Whereas these studies provide worst-case, distribution-agnostic expressivity limits, our framework combines TC^0 bottlenecks with optimal transport and Barron-space approximation theory [Barron, 1993, Makovoz, 1996] to bound expected generalization risk under concrete data distributions.

Positional encodings as structural regularizers. Positional encodings strongly influence length extrapolation [Zhao et al., 2024], spanning learned absolute embeddings (APE), relative biases such as ALiBi [Press et al., 2022], and architectures without explicit positional encodings (NoPE) [Kazemnejad et al., 2023]. Recent methods—including functional interpolation (FIRE) [Li et al., 2024], bilevel encodings (BiPE) [He et al., 2024], and data-adaptive encodings (DAPE) [Zheng et al., 2024]—further improve long-context generalization. Here we analyze how internal attention scores respond when a local search segment is translated to later sequence positions. Under our baseline

smoothness assumption, relative encodings that preserve scores within the segment ($\Delta A = 0$ under $\mathcal{T}_k$) inherit an $\mathcal{O}(1/T_{\max})$ Lipschitz bound, whereas under APE the target mass of translated searches misrouted by score perturbations induces an $\Omega(1)$ risk floor (Theorem 1, Corollary 1).

Optimal transport and domain shifts in reasoning. Recent empirical studies model LLM reasoning as search-tree traversal [Gandhi et al., 2024] and attribute OOD failures to distributional shifts between training and test traces [Prystawski et al., 2023, Zhao et al., 2026], while treating the neural architecture as a black box. Conversely, classical optimal transport (OT) domain adaptation bounds target risk via Wasserstein distances over static Euclidean feature spaces [Courty et al., 2017, Redko et al., 2017]. We connect these two lines of work by projecting variable-length trajectories into a continuous structural metric space, coupling Kantorovich duality directly with Transformer circuit complexity and attention kernels in Theorem 2.

7 Conclusion and Limitations

We have presented a measure-theoretic framework that quantifies out-of-distribution generalization for search-based LLM reasoning. Projecting discrete reasoning trajectories into a structural metric space and applying Kantorovich duality yields two main architectural conclusions. First, translating a search segment alters APE attention scores almost surely and can divert greedy decoding into an incorrect branch; when enough target searches encounter such routing failures, both the structural Lipschitz constant and target risk incur $\Omega(1)$ lower bounds, whereas RoPE preserves local translation equivariance and controls the transport penalty. Second, recursive backtracking imposes a separate TC^0 risk floor when tree nesting exceeds what a single forward pass of L layers can resolve; increasing layer depth removes this floor, whereas increasing width alone reduces only the Barron-space term. Controlled experiments on Countdown and extended evaluations in the appendix support both predictions.

Limitations and future work. Our theoretical guarantees rely on the ϵ-relaxed structural sufficiency condition (Assumption 1), which is tailored to formal combinatorial search. On Countdown, where our three search measures share the same problem distribution, a constant predictor yields empirical residuals rounding to 0.004 and 0.014 for the two models evaluated in Appendix J; extending this sufficiency analysis to open-ended natural language reasoning remains an open problem. In addition, the choice of structural coordinates entails a trade-off between risk prediction and transport estimation: on Countdown, expanding Φ from 12 to 34 coordinates changes predictive ROC-AUC by at most 0.003 while roughly doubling the within-measure baseline for $\widehat{W}_1$ (Appendix M.1.1; see Appendix M.1 for general design principles).

Furthermore, because our three Countdown measures share the same underlying problem distribution, benchmarks that jointly shift problem difficulty and search strategy will enable more direct tests of the $K_f W_1$ transport term (Appendix K.5). Finally, our TC^0 bottleneck models routing within a single forward pass of autoregressive generation; formalizing how external inference-time search algorithms (e.g., Tree-of-Thoughts or Monte Carlo Tree Search) reshape the effective target measure ν_T is a promising direction for future work.

Broader impact. For long-horizon search ($T_{\max} \to \infty$), translation-equivariant encodings such as RoPE control the transport penalty under our smoothness assumption, whereas APE incurs an order-one risk floor when position shifts misroute enough target searches (Corollary 1). Similarly, because Lemma 5 bounds the nesting depth resolvable in a single forward pass by layer count, splitting deeply nested tasks across separate model calls reduces the depth each call must resolve (Appendix M.4). Potential negative impacts include over-reliance on these bounds in place of domain-specific safety testing and indirect dual-use risks associated with more capable reasoning models.

Acknowledgments and Disclosure of Funding

We thank Qihuang Zhang (Department of Epidemiology, Biostatistics and Occupational Health, McGill University) for supporting this work.

This research was enabled in part by compute resources, software, and technical assistance provided by the Mila cluster.

Funding. Computing resources were provided by Mila – Quebec AI Institute. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant (RGPIN-2023-03854). No other third-party funding was received.

Competing interests. The authors declare no competing interests.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, et al. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Ibragim Badertdinov, Alexander Golubev, Maksim Nekrashevich, Anton Shevtsov, Simon Karasik, Andrei Andriushchenko, Maria Trofimova, Daria Litvintseva, and Boris Yangel. SWE-rebench: An automated pipeline for task collection and decontaminated evaluation of software engineering agents. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-0788. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/21bec6ace947b1b58967b945c8ac0f10-Abstract-Datasets_and_Benchmarks_Track.html.

Andrew Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. doi: 10.1109/18.256500.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Agustín Piqueres Lajarín, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Ben Burtenshaw, Clémentine Fourrier, Haojun Zhao, Hugo Larcher, Mathieu Morlon, Cyril Zakka, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big — data-centric training of a fully open small language model. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=3JiCl2A14H. arXiv:2502.02737.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model, 2022. URL https://arxiv.org/abs/2211.05100.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large scale autoregressive language modeling with Mesh-Tensorflow. Zenodo, March 2021. URL https://doi.org/10.5281/zenodo.5297715. Software, version 1.0.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rJeXCo0cYX.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pages 73383–73394. Curran Associates, Inc., 2023. doi: 10.52202/075280-3209. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e8916198466e8ef218a2185a491b49fa-Paper-Datasets_and_Benchmarks.pdf.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017. doi: 10.1109/TPAMI.2016.2615921. URL https://arxiv.org/abs/1507.00504.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013. URL https://arxiv.org/abs/1306.0895.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Proceedings of the 29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940, 2016. URL https://arxiv.org/abs/1512.03965.

Embodied-CoT. Embodied-CoT/embodied_features_bridge. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/Embodied-CoT/embodied_features_bridge. Embodied features and reasonings of the ECoT project for the BridgeData V2 demonstrations. MIT license. Accessed 2026.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.15408.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of Search (SoS): Learning to search in language. In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.03683.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, December 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00306. URL http://dx.doi.org/10.1162/tacl_a_00306.

Zhenyu He, Guhao Feng, Shengjie Luo, Kai Yang, Liwei Wang, Jingjing Xu, Zhi Zhang, Hongxia Yang, and Di He. Two stones hit one bird: Bilevel positional encoding for better length extrapolation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17858–17876, 2024. URL https://arxiv.org/abs/2401.16421.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, pages 54107–54157, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.19466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023. URL https://arxiv.org/abs/2309.06180.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH. arXiv:2411.15124.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. Limits to depth efficiencies of self-attention. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.12467.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. Functional interpolation for relative positions improves long context transformers. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04418.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Y. Makovoz. Random approximants and neural networks. *Journal of Approximation Theory*, 85(1): 98–109, 1996. ISSN 0021-9045. doi: 10.1006/jath.1996.0031. URL https://www.sciencedirect.com/science/article/pii/S0021904596900313.

Daniel McFadden. Quantitative methods for analysing travel behaviour of individuals: Some recent developments. In David A. Hensher and Peter R. Stopher, editors, *Behavioural Travel Modelling*, pages 279–318. Croom Helm, London, 1979.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023. doi: 10.1162/tacl_a_00562. URL https://arxiv.org/abs/2207.00729.

Mistral AI. Mistral Small 3. Mistral AI news, January 2025. URL https://mistral.ai/news/mistral-small-3. Model: Mistral-Small-24B-Instruct-2501. Accessed 2026.

B. S. Mityagin. The zero set of a real analytic function. *Mathematical Notes*, 107(3–4):529–530, 2020. doi: 10.1134/S0001434620030189. URL https://doi.org/10.1134/S0001434620030189.

Nebius. nebius/SWE-agent-trajectories. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/nebius/SWE-agent-trajectories. CC-BY-4.0 license. Accessed 2026.

OpenAI. Learning to reason with LLMs. OpenAI blog, 2024a. URL https://openai.com/index/learning-to-reason-with-llms/. Accessed 2025.

OpenAI. Introducing SWE-bench Verified. OpenAI blog, August 2024b. URL https://openai.com/index/introducing-swe-bench-verified/. Accessed 2026.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2108.12409.

Ben Prystawski, Michael Y. Li, and Noah D. Goodman. Why think step by step? Reasoning emerges from the locality of experience. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.03843.

Python Software Foundation. Python 3.10.19. https://www.python.org/downloads/release/python-31019/, October 2025. Accessed 2026.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2017)*, pages 737–753, 2017. doi: 10.1007/978-3-319-71246-8_45. URL https://arxiv.org/abs/1610.04420.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1889–1903, 2023. URL https://arxiv.org/abs/2305.16843.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.409. URL https://aclanthology.org/2024.acl-long.409/.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. ETO-SFT-Trajectory: Expert trajectories for ETO. Hugging Face dataset, 2024b. URL https://huggingface.co/datasets/agent-eto/eto-sft-trajectory. Apache-2.0 license. Accessed 2026.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00663. URL http://dx.doi.org/10.1162/tacl_a_00663.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://arxiv.org/abs/2104.09864.

Team OLMo, Pete Walsh, Luca Soldaini, et al. 2 OLMo 2 furious, 2024. URL https://arxiv.org/abs/2501.00656.

Maria Trofimova, Anton Shevtsov, Ibragim Badertdinov, Konstantin Pyaev, Simon Karasik, and Alexander Golubev. OpenHands trajectories with Qwen3-Coder-480B-A35B-Instruct. Nebius blog, 2025. URL https://nebius.com/blog/posts/openhands-trajectories-with-qwen3-coder-480b. Dataset: https://huggingface.co/datasets/nebius/SWE-rebench-openhands-trajectories (CC-BY-4.0 license). Accessed 2026.

Alexander B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004. doi: 10.1214/aos/1079120131. URL https://doi.org/10.1214/aos/1079120131.

Cédric Villani. The Wasserstein distances. In *Optimal Transport: Old and New*, pages 93–111. Springer, Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9. doi: 10.1007/978-3-540-71050-9_6. URL https://doi.org/10.1007/978-3-540-71050-9_6.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.775. URL https://aclanthology.org/2022.emnlp-main.775/.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, pages 65882–65919, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a4b6ad6b48850c0c331d1259fc66a69c-Abstract-Conference.html.

Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.17453.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, pages 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pages 20744–20757. Curran Associates, Inc., 2022. doi: 10.52202/068431-1508. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.10601.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3157–3181. PMLR, 2025. URL https://proceedings.mlr.press/v270/zawalski25a.html.

Susan Zhang, Stephen Roller, Naman Goyal, et al. OPT: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Chengshuai Zhao, Zhen Tan, Pingchuan Ma, Dawei Li, Bohan Jiang, Yancheng Wang, Yingzhen Yang, and Huan Liu. Is chain-of-thought reasoning of LLMs a mirage? A data distribution lens. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 15231–15261. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.749. URL https://aclanthology.org/2026.findings-acl.749.

Liang Zhao, Xiachong Feng, Xiaocheng Feng, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, Bing Qin, and Ting Liu. Length extrapolation of transformers: A survey from the perspective of positional encoding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9959–9977, 2024. doi: 10.18653/v1/2024.findings-emnlp.582. URL https://arxiv.org/abs/2312.17044.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. DAPE: Data-adaptive positional encoding for length extrapolation. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14722.

A Formal Setup and Notation

This section collects the formal objects used throughout the appendix and establishes our notation for the measure-theoretic formulation of search-based reasoning.

A.1 Trajectory Space and Structural Feature Space

Let $\mathcal{V}$ be a finite vocabulary and let $T_{\max} \in \mathbb{N}^+$ denote the maximum physical sequence length. The ambient space of bounded-length sequences is defined as

$$\mathcal{Z}_{\text{all}} := \bigcup_{t=1}^{T_{\max}} \mathcal{V}^t. \tag{6}$$

Let Γ denote the autoregressive grammar governing valid search trajectories. The valid trajectory space is then

$$\mathcal{Z} := \{z \in \mathcal{Z}_{\text{all}} : z \text{ satisfies } \Gamma\}. \tag{7}$$

Since $|\mathcal{V}| < \infty$ and $T_{\max} < \infty$, the set $\mathcal{Z}$ is finite, and we equip it with the discrete σ-algebra

$$\Sigma_{\mathcal{Z}} := 2^{\mathcal{Z}}. \tag{8}$$

On this measurable space, we consider source and target probability measures

$$\mu_S, \mu_T \in \mathcal{P}(\mathcal{Z}), \tag{9}$$

where $\mathcal{P}(\mathcal{Z})$ denotes the space of probability measures on $(\mathcal{Z}, \Sigma_{\mathcal{Z}})$.

Under string concatenation $\oplus$, each trajectory $z \in \mathcal{Z}$ decomposes uniquely as

$$z = x_{\text{init}}^{(z)} \oplus y_{\text{search}}, \tag{10}$$

where $x_{\text{init}}^{(z)} \in \mathcal{V}^k$ represents the initial problem state and $y_{\text{search}} \in \mathcal{V}^{t-k}$ is the generated search trace.

We next define the continuous structural feature space. Let $\mathcal{X} \subset \mathbb{R}^{12}$ denote the structural state space, equipped with the Euclidean metric

$$d_{\mathcal{X}}(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} - \mathbf{v}\|_2. \tag{11}$$

This metric induces the Borel σ-algebra $\mathcal{B}(\mathcal{X})$. Because all trajectories in $\mathcal{Z}$ have physical length at most $T_{\max}$, every structural coordinate is bounded; taking $\mathcal{X}$ to be the compact box spanned by the coordinate ranges of $\Phi(\mathcal{Z})$, each coordinate range grows at most linearly with $T_{\max}$, so $\operatorname{diam}(\mathcal{X}) = \mathcal{O}(T_{\max})$.

Specifically, we construct the coordinate-level structural projection

$$\Phi : \mathcal{Z} \to \mathcal{X}, \qquad z \mapsto \boldsymbol{\chi}(z) = [\chi_1(z), \ldots, \chi_{12}(z)]^\top, \tag{12}$$

which maps each variable-length discrete trajectory to a twelve-dimensional structural vector.

Let N denote the total number of node references in z—namely, the node identifiers appearing in `"Generated Node"` and `"Moving to Node"` lines in order of appearance (including revisits)—and let N_{gen} denote the number of explicitly generated nodes. The definitions below assume $N \geq 1$, whereas any trajectory without node references is assigned $\chi_1 = \cdots = \chi_9 = 0$. For the i-th node reference, let id_i denote its sequence identifier, and define its structural tree depth by

$$D_i := |\mathrm{id}_i|. \tag{13}$$

In addition, let $C(z, \omega)$ denote the number of exact, non-overlapping occurrences of a substring ω in z.

The vector $\boldsymbol{\chi}(z)$ summarizes the search structure of a trajectory through three groups of statistics.

Tree-structural and volumetric statistics.

$$\chi_1(z) = \max_{1 \le i \le N} D_i \qquad \text{(maximum depth)}, \tag{14}$$

$$\chi_2(z) = \frac{1}{N} \sum_{i=1}^{N} D_i \qquad \text{(mean depth)}, \tag{15}$$

$$\chi_3(z) = \left(\frac{1}{N} \sum_{i=1}^{N} (D_i - \chi_2)^2 \right)^{1/2} \qquad \text{(depth standard deviation)}, \tag{16}$$

$$\chi_4(z) = N \qquad \text{(search volume)}. \tag{17}$$

Depth-transition statistics. Letting $M := \max(1, N-1)$, we define

$$\chi_5(z) = \frac{1}{M} \sum_{i=1}^{N-1} \mathbb{I}\{D_{i+1} > D_i\} \qquad \text{(deepening rate)}, \tag{18}$$

$$\chi_6(z) = \frac{1}{M} \sum_{i=1}^{N-1} \mathbb{I}\{D_{i+1} < D_i\} \qquad \text{(backtracking rate)}, \tag{19}$$

$$\chi_7(z) = \frac{1}{M} \sum_{i=1}^{N-1} \mathbb{I}\{D_{i+1} = D_i\} \qquad \text{(lateral transition rate)}. \tag{20}$$

Pruning and outcome statistics.

$$\chi_8(z) = C(z, \texttt{"No Solution"}) \qquad \text{(prune count)}, \tag{21}$$

$$\chi_9(z) = \chi_8 / N_{\text{gen}} \qquad \text{(pruning rate; 0 if } N_{\text{gen}} = 0), \tag{22}$$

$$\chi_{10}(z) = \mathbb{I}\{C(z, \texttt{"Goal Reached"}) \ge 1\} \qquad \text{(solution indicator)}, \tag{23}$$

$$\chi_{11}(z) = C(z, \texttt{"*"}) + C(z, \texttt{"/"}) \qquad \text{(multiplicative bias)}, \tag{24}$$

$$\chi_{12}(z) = C(z, \texttt{"+"}) + C(z, \texttt{"-"}) \qquad \text{(additive bias)}. \tag{25}$$

Since $\mathcal{Z}$ is finite and carries the discrete σ-algebra, every mapping from $\mathcal{Z}$ into $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ is measurable. In particular, Φ is Borel measurable and its image $\Phi(\mathcal{Z})$ is finite.

A.2 Theoretical Justification of Structural Sufficiency (ϵ-Relaxation)

Assumption 1 posits that the projection $\Phi(z)$ serves as a sufficient structural proxy for the expected zero-one risk up to a residual error ϵ. This relaxation is motivated by the algorithmic structure of formal combinatorial planning tasks such as Countdown, and is tested empirically in Appendix J on two trained models, where $\boldsymbol{\chi}$ alone predicts per-problem solving success with an out-of-fold ROC-AUC of about 0.93.

In formal search tasks, whether a model reaches a valid solution depends primarily on the structure of the search: exploring to sufficient tree depth (maximum depth χ_1), recovering from dead ends via backtracking (backtracking rate χ_6), and balancing multiplicative and additive operations (operator counts χ_{11} and χ_{12}). The feature space $\mathcal{X}$ records these structural properties, while the residual ϵ absorbs the effect of instance-specific details omitted by Φ (such as exact input numbers) on the mean task risk. Consequently, ϵ remains small whenever a single function of the reference search coordinates approximates the mean failure rate across the compared distributions (Appendix J). Under Assumption 1 and a Lipschitz structural risk function on $\mathcal{X}$, the trajectory-level risk gap is bounded by $K_f W_1(\nu_S, \nu_T) + 2\epsilon$ (Lemma 1), connecting discrete sequence generation to optimal transport on $\mathcal{X}$.

A.3 Pushforward Measures and Structural Shift

The structural source and target measures are defined as the pushforwards of μ_S and μ_T under Φ:

$$\nu_S := \Phi_{\#}\mu_S, \qquad \nu_T := \Phi_{\#}\mu_T. \tag{26}$$

Since Φ is Borel measurable, ν_S and ν_T are valid Borel probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Because $\Phi(\mathcal{Z})$ is finite and $\mathcal{X}$ is bounded, both measures have finite first moments; equivalently, $\nu_S, \nu_T \in \mathcal{P}_1(\mathcal{X})$. By definition, for any Borel set $A \in \mathcal{B}(\mathcal{X})$,

$$\nu_S(A) = \mu_S(\Phi^{-1}(A)), \qquad \nu_T(A) = \mu_T(\Phi^{-1}(A)). \tag{27}$$

Because $\mathcal{Z}$ is finite, these pushforward measures admit the explicit representations

$$\nu_S(A) = \sum_{z \in \mathcal{Z}} \mathbb{I}_A(\Phi(z))\, \mu_S(\{z\}), \tag{28}$$

$$\nu_T(A) = \sum_{z \in \mathcal{Z}} \mathbb{I}_A(\Phi(z))\, \mu_T(\{z\}). \tag{29}$$

To quantify the structural discrepancy between source and target reasoning behaviors, we use the Wasserstein-1 distance between their pushforward measures:

$$W_1(\nu_S, \nu_T) := \inf_{\pi \in \Pi(\nu_S, \nu_T)} \int_{\mathcal{X} \times \mathcal{X}} d_{\mathcal{X}}(\mathbf{u}, \mathbf{v})\, d\pi(\mathbf{u}, \mathbf{v}), \tag{30}$$

where $\Pi(\nu_S, \nu_T)$ denotes the set of all couplings of ν_S and ν_T—that is, probability measures on $\mathcal{X} \times \mathcal{X}$ with marginals ν_S and ν_T [Villani, 2009]. Since $\nu_S, \nu_T \in \mathcal{P}_1(\mathcal{X})$ and $\mathcal{X}$ is bounded, the transport cost is finite for every $\pi \in \Pi(\nu_S, \nu_T)$, ensuring that $W_1(\nu_S, \nu_T)$ is well-defined and finite.

A.4 Risk and Structural Risk

Let $\mathcal{F}_{L,m,\mathrm{PE}}$ denote the class of autoregressive Transformer models with layer depth L, hidden dimension m, and positional encoding scheme PE. Under the trajectory decomposition in Eq. (10), each $z \in \mathcal{Z}$ induces an initial problem prompt $x_{\mathrm{init}}^{(z)}$. For a model $f \in \mathcal{F}_{L,m,\mathrm{PE}}$, the trajectory generated from this prompt is

$$f(x_{\mathrm{init}}^{(z)}) \in \mathcal{Z}_{\mathrm{all}}. \tag{31}$$

Let $S_z : \mathcal{Z}_{\mathrm{all}} \to \mathbb{R}$ be a deterministic scoring function, parameterized by the target state of the reference trajectory z, that evaluates the formal validity of a generated output. We define the zero-one loss as

$$\ell(f, z) := 1 - \mathbb{I}_{(0,\infty)} \left(S_z \left(f(x_{\mathrm{init}}^{(z)}) \right) \right). \tag{32}$$

For any probability measure $\mu \in \mathcal{P}(\mathcal{Z})$, the expected risk of f under μ is then

$$\mathcal{R}_\mu(f) := \mathbb{E}_{z \sim \mu}[\ell(f, z)]. \tag{33}$$

In the main text, we relate this trajectory-level risk $\mathcal{R}_\mu(f)$ to a structural risk functional defined on $\mathcal{X}$ through the pushforward measure $\Phi_{\#}\mu$.

B Proofs of Structural Generalization Results

B.1 Structural Sufficiency Assumption

We first recall Assumption 1, which underpins the proofs below. It posits that for an irreducible approximation error $\epsilon > 0$ and each model f, there exists a Borel-measurable structural risk function $\mathcal{L}_f : \mathcal{X} \to [0, 1]$ satisfying, for every $\mu \in \{\mu_S, \mu_T\}$,

$$\left| \mathcal{R}_\mu(f) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d(\Phi_{\#}\mu)(\boldsymbol{\chi}) \right| \leq \epsilon. \tag{34}$$

B.2 Proof of Lemma 1

We first recall the statement of Lemma 1. Assume that $\mathcal{L}_f$ is Lipschitz continuous on $(\mathcal{X}, d_{\mathcal{X}})$ with minimal Lipschitz constant

$$K_f := \sup_{\boldsymbol{u} \neq \boldsymbol{v} \in \mathcal{X}} \frac{|\mathcal{L}_f(\boldsymbol{u}) - \mathcal{L}_f(\boldsymbol{v})|}{d_{\mathcal{X}}(\boldsymbol{u}, \boldsymbol{v})}. \tag{35}$$

Then the generalization error gap satisfies

$$|\mathcal{R}_{\mu_T}(f) - \mathcal{R}_{\mu_S}(f)| \leq K_f\, W_1(\nu_S, \nu_T) + 2\epsilon. \tag{36}$$

Proof. Let

$$\Delta\mathcal{R} := \left|\mathcal{R}_{\mu_T}(f) - \mathcal{R}_{\mu_S}(f)\right|. \tag{37}$$

Adding and subtracting the structural expectations under ν_T and ν_S gives

$$\begin{aligned}\Delta\mathcal{R} = \Bigg|&\mathcal{R}_{\mu_T}(f) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) + \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) \\ &- \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi}) + \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi}) - \mathcal{R}_{\mu_S}(f)\Bigg|.\end{aligned} \tag{38}$$

Applying the triangle inequality splits this into three terms:

$$\begin{aligned}\Delta\mathcal{R} \leq &\left|\mathcal{R}_{\mu_T}(f) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi})\right| \\ &+ \left|\int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi})\right| \\ &+ \left|\int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi}) - \mathcal{R}_{\mu_S}(f)\right|.\end{aligned} \tag{39}$$

By Assumption 1, the first and third terms are each bounded by ϵ, which yields

$$\Delta\mathcal{R} \leq \left|\int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi})\right| + 2\epsilon. \tag{40}$$

If $K_f = 0$, then $\mathcal{L}_f$ is constant on $\mathcal{X}$, so the integral difference vanishes and the claim holds immediately.

Assume now that $K_f > 0$, and define the normalized risk function

$$g(\boldsymbol{\chi}) := \frac{\mathcal{L}_f(\boldsymbol{\chi})}{K_f}. \tag{41}$$

By construction, g is 1-Lipschitz. Since $\nu_S, \nu_T \in \mathcal{P}_1(\mathcal{X})$, Kantorovich–Rubinstein duality for W_1 applies on $(\mathcal{X}, d_{\mathcal{X}})$ [Villani, 2009, Weed and Bach, 2019], giving

$$\left|\int_{\mathcal{X}} g(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \int_{\mathcal{X}} g(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi})\right| \leq W_1(\nu_S, \nu_T). \tag{42}$$

Substituting $g = \mathcal{L}_f / K_f$ back into Eq. (42) yields

$$\left|\int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_S(\boldsymbol{\chi})\right| \leq K_f\, W_1(\nu_S, \nu_T). \tag{43}$$

Combining Eq. (43) with Eq. (40) gives

$$\Delta\mathcal{R} \leq K_f\, W_1(\nu_S, \nu_T) + 2\epsilon, \tag{44}$$

which completes the proof. □

C Proofs of Positional Encoding Results

C.1 Attention Kernels and the Translation Operator

We compare the pre-softmax attention score between two tokens before and after their position indices are both incremented by k. To isolate the effect of positional indexing, we hold their content vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ fixed. Let $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d\times d}$ denote the learned query and key projections, with the attention head's fixed scaling factor absorbed into $\mathbf{W}_Q$. For absolute positional encoding (APE), let $\mathbf{p}_i, \mathbf{p}_j \in \mathbb{R}^d$ denote the absolute position embeddings. The pre-softmax attention score is

$$A_{i,j}^{\mathrm{APE}} = (\mathbf{x}_i + \mathbf{p}_i)^\top \mathbf{W}_Q^\top \mathbf{W}_K (\mathbf{x}_j + \mathbf{p}_j). \tag{45}$$

For rotary position embedding (RoPE) [Su et al., 2024], let $\mathbf{R}_\Theta^t \in \mathbb{R}^{d\times d}$ denote the orthogonal block-diagonal rotation matrix associated with position t. Using $(\mathbf{R}_\Theta^i)^\top = (\mathbf{R}_\Theta^i)^{-1} = \mathbf{R}_\Theta^{-i}$, the pre-softmax attention score is

$$\begin{aligned} A_{i,j}^{\mathrm{RoPE}} &= (\mathbf{R}_\Theta^i \mathbf{W}_Q \mathbf{x}_i)^\top (\mathbf{R}_\Theta^j \mathbf{W}_K \mathbf{x}_j) && (46)\\ &= \mathbf{x}_i^\top \mathbf{W}_Q^\top (\mathbf{R}_\Theta^i)^\top \mathbf{R}_\Theta^j \mathbf{W}_K \mathbf{x}_j && (47)\\ &= \mathbf{x}_i^\top \mathbf{W}_Q^\top \mathbf{R}_\Theta^{j-i} \mathbf{W}_K \mathbf{x}_j. && (48) \end{aligned}$$

Bounds on the structural perturbation δ. Write $z = x_{\mathrm{init}}^{(z)} \oplus y_{\mathrm{search}}$, and consider admissible insertions for which the resulting trajectory satisfies the search grammar Γ and has total length at most $T_{\max}$. On this domain, $\mathcal{T}_k$ inserts a single structurally inert node of k tokens immediately before the sub-trajectory ω, adding one node reference ($\chi_4' = \chi_4 + 1$) while shifting every token of ω forward by $+k$ positions. The Euclidean distance between $\boldsymbol{\chi} = \Phi(z)$ and $\boldsymbol{\chi}' = \Phi(\mathcal{T}_k z)$ is therefore

$$d_{\mathcal{X}}(\boldsymbol{\chi}, \boldsymbol{\chi}') = \left(1^2 + \sum_{j\neq 4} (\Delta\chi_j)^2\right)^{1/2} =: \delta, \tag{49}$$

so $\delta \geq 1$ even when all other coordinates remain unchanged. Definition 2 further requires a uniform upper bound $\delta \leq C_\delta$ independent of k and $T_{\max}$. Along trajectories of uniformly bounded tree depth, inserting a node reference at an already visited depth alters the mean depth χ_2 and depth variance by $\mathcal{O}(1/N)$—and hence the depth standard deviation χ_3 by at most $\mathcal{O}(1/\sqrt{N})$—while shifting each transition rate χ_5, χ_6, χ_7 by $\mathcal{O}(1/N)$. With the remaining counts and pruning rate unchanged, this yields $\delta = 1 + \mathcal{O}(1/N) \leq C_\delta$, allowing us to convert a constant-order jump in structural risk into an $\Omega(1)$ lower bound on the Lipschitz constant K_f.

C.2 Proof of Lemma 2

For completeness, we restate Lemma 2. Assume the network parameters (projection matrices $\mathbf{W}_Q, \mathbf{W}_K$ and absolute position embeddings $\mathbf{p}_t$) are generically parameterized (avoiding exact algebraic degeneracies), and that

$$\mathbf{W}_Q^\top \mathbf{W}_K \neq \mathbf{0}. \tag{50}$$

Then, under the translation operator $\mathcal{T}_k$ with $k > 0$, the local attention kernel of an APE model differs almost surely:

$$\Delta A^{\mathrm{APE}} = |\widetilde{A}_{a,b}^{\mathrm{APE}} - A_{a,b}^{\mathrm{APE}}| > 0 \qquad \text{a.s.} \tag{51}$$

Proof. Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be the content vectors of two tokens in ω, whose position indices shift under $\mathcal{T}_k$ from a, b to $a + k, b + k$. Substituting the shifted position embeddings into the APE attention score gives

$$\begin{aligned} \Delta A^{\mathrm{APE}} &:= \left|\widetilde{A}_{a,b}^{\mathrm{APE}} - A_{a,b}^{\mathrm{APE}}\right| && (52)\\ &= \left|(\mathbf{x}_a + \mathbf{p}_{a+k})^\top \mathbf{W}_Q^\top \mathbf{W}_K (\mathbf{x}_b + \mathbf{p}_{b+k}) - (\mathbf{x}_a + \mathbf{p}_a)^\top \mathbf{W}_Q^\top \mathbf{W}_K (\mathbf{x}_b + \mathbf{p}_b)\right|. && (53) \end{aligned}$$

Expanding both bilinear forms and canceling the shared content term $\mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{x}_b$ yields

$$\Delta A^{\mathrm{APE}} = \Big| \mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K (\mathbf{p}_{b+k} - \mathbf{p}_b) + (\mathbf{p}_{a+k} - \mathbf{p}_a)^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{x}_b + \mathbf{p}_{a+k}^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{p}_{b+k} - \mathbf{p}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{p}_b \Big|. \tag{54}$$

Writing $\mathbf{u} := \mathbf{p}_{a+k}$ and $\mathbf{v} := \mathbf{p}_{b+k}$, the signed score difference is

$$S(\mathbf{u}, \mathbf{v}) := \mathbf{u}^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{v} + \mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{v} + \mathbf{u}^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{x}_b - \Big(\mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{p}_b + \mathbf{p}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{x}_b + \mathbf{p}_a^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{p}_b \Big), \tag{55}$$

so the two scores coincide if and only if $S(\mathbf{u}, \mathbf{v}) = 0$. The quadratic term in the shifted embeddings is

$$\mathbf{u}^\top \mathbf{W}_Q^\top \mathbf{W}_K \mathbf{v} = \sum_{i=1}^{d} \sum_{j=1}^{d} u_i (\mathbf{W}_Q^\top \mathbf{W}_K)_{ij} v_j. \tag{56}$$

Because an unshifted index and a shifted index may coincide (e.g., when $b = a + k$), we view $S(\theta)$ as a polynomial in the joint parameter vector θ comprising $\mathbf{W}_Q, \mathbf{W}_K$ and the distinct position embeddings indexed by $\{a, b, a+k, b+k\}$. To verify that $S(\theta)$ is not identically zero, evaluate it at $\mathbf{W}_Q = \mathbf{W}_K = \mathbf{I}$ and $\mathbf{p}_j = tj\mathbf{e}_1$ for each distinct index j: the coefficient of t^2 is $(a+k)(b+k) - ab = k(a+b+k) > 0$, which remains strictly positive whether $a = b$ or a shifted index coincides with an original one. Because the zero locus $\{\theta : S(\theta) = 0\}$ of a non-identically-zero polynomial has Lebesgue measure zero in the joint parameter space [Mityagin, 2020] (containing only special algebraic configurations such as exact periodic embeddings), generic parameterization ensures

$$\Delta A^{\mathrm{APE}} > 0 \qquad \text{almost surely.} \tag{57}$$

□

C.3 RoPE Equivariance under Sequence Translation

For RoPE, translating both tokens by $+k$ preserves their relative offset $b - a$, yielding $\Delta A^{\mathrm{RoPE}} = 0$ for fixed content vectors in Theorem 1.

Proof. Let $\mathbf{x}_a, \mathbf{x}_b$ be translated from indices a, b to $a + k, b + k$. The translated RoPE kernel is

$$\widetilde{A}_{a,b}^{\mathrm{RoPE}} = \mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{R}_\Theta^{(b+k)-(a+k)} \mathbf{W}_K \mathbf{x}_b \tag{58}$$

$$= \mathbf{x}_a^\top \mathbf{W}_Q^\top \mathbf{R}_\Theta^{b-a} \mathbf{W}_K \mathbf{x}_b \tag{59}$$

$$= A_{a,b}^{\mathrm{RoPE}}, \tag{60}$$

and hence

$$\Delta A^{\mathrm{RoPE}} = \left| \widetilde{A}_{a,b}^{\mathrm{RoPE}} - A_{a,b}^{\mathrm{RoPE}} \right| = 0. \tag{61}$$

□

C.4 Details of Assumption 2

We now connect these local attention-score calculations to the structural risk function $\mathcal{L}_f$ in Theorem 1. Assumption 2 consists of two parts governing how $\mathcal{L}_f$ behaves under translation-invariant versus position-sensitive kernels.

(i) Global baseline smoothness. We assume that the underlying search task has intrinsic structural smoothness $K_{\mathrm{base}} = \mathcal{O}(1/T_{\max})$, and that any architecture whose local attention kernel is invariant to sequence translation ($\Delta A^{\mathrm{RoPE}} = 0$) inherits this baseline smoothness at the level of structural risk, giving $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$.

(ii) Autoregressive sensitivity and fragile logit margins. For an APE model $f_{\mathrm{APE}} \in \mathcal{F}_{L,m,\mathrm{APE}}$, autoregressive search generation acts as a deterministic Markov decision process. For every sufficiently large $T_{\max}$, we assume that an admissible translated pair $(z, \mathcal{T}_k(z))$ contains a critical routing step. At this step—such as selecting the next branch when backtracking in a deep subtree—let

$\Delta_{\text{margin}} > 0$ denote the pre-softmax logit gap between the Bayes-optimal routing token $a^\star$ and the highest-scoring competing token. On increasingly deep out-of-distribution trajectories, where this margin becomes small, the attention-score perturbation $\Delta A^{\text{APE}} > 0$ from Lemma 2 is assumed to invert the ranking between $a^\star$ and an incorrect routing token. Under greedy decoding and exact-match evaluation, selecting an invalid branch prevents the trajectory from reaching the target state, producing a jump in structural risk of at least

$$|\mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi}') - \mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi})| \geq \gamma, \qquad \gamma := 1 - \frac{1}{|\mathcal{V}|} > 0. \tag{62}$$

Routing separation under deterministic decoding. Formally, let $a^\star(z)$ denote the Bayes-optimal routing action at a critical transition of trajectory z, and let

$$a_f(z) := \arg\max_{a \in \mathcal{V}} p_f(a \mid h_z)$$

denote the greedy action chosen by f given its generated prefix h_z on prompt $x_{\text{init}}^{(z)}$. For a problem instance whose reference trajectory z' projects near $\boldsymbol{\chi}' = \Phi(\mathcal{T}_k z)$, Assumption 2 stipulates that at the critical transition,

$$a_f(z') \neq a^\star(z'),$$

diverting the model into an incorrect branch. In our Countdown evaluation, such a misrouted trace fails to produce a valid solution path within the 512-token generation budget and receives zero-one loss 1 from the verifier (Appendix K). The resulting separation in structural risk is bounded below by the uniform guessing error rate over the vocabulary:

$$|\mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi}') - \mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi})| \geq \gamma, \qquad \gamma := 1 - \frac{1}{|\mathcal{V}|} > 0. \tag{63}$$

While the exact-match risk difference between a solved and a failed trajectory can be as large as 1, Theorem 1 requires only that γ remain bounded away from zero as $T_{\max}$ grows.

Interpretation of Assumption 2 on $\mathcal{L}_f$. Applying $\mathcal{T}_k$ modifies the reference search y_{search} while leaving the initial problem $x_{\text{init}}^{(z)}$ unchanged; because the model receives only $x_{\text{init}}^{(z)}$, its pointwise loss on that specific instance satisfies $\ell(f, \mathcal{T}_k z) = \ell(f, z)$ (Appendix A.4). Rather than a change on the identical prompt, the jump in the structural risk function $\mathcal{L}_f$ between $\boldsymbol{\chi} = \Phi(z)$ and $\boldsymbol{\chi}' = \Phi(\mathcal{T}_k z)$ reflects how f performs across problem instances whose reference searches map near $\boldsymbol{\chi}'$: on those instances, the model encounters the corresponding local subtree at shifted token indices within its own generated trace, where the APE score perturbation misroutes the critical transition and induces the risk gap γ.

C.5 Proof of Theorem 1

For completeness, we restate Theorem 1, which holds under Assumption 2. For sufficiently large $T_{\max}$, let $f_{\text{APE}} \in \mathcal{F}_{L,m,\text{APE}}$ and $f_{\text{RoPE}} \in \mathcal{F}_{L,m,\text{RoPE}}$ be two matched autoregressive Transformer models that differ only in their positional encoding scheme. Then

$$K_{f,\text{RoPE}} = \mathcal{O}(1/T_{\max}), \qquad K_{f,\text{APE}} = \Omega(1), \tag{64}$$

and hence $K_{f,\text{RoPE}} \ll K_{f,\text{APE}}$.

Proof. By definition,

$$K_{f,\text{APE}} = \sup_{\boldsymbol{u} \neq \boldsymbol{v} \in \mathcal{X}} \frac{|\mathcal{L}_{f_{\text{APE}}}(\boldsymbol{u}) - \mathcal{L}_{f_{\text{APE}}}(\boldsymbol{v})|}{d_{\mathcal{X}}(\boldsymbol{u}, \boldsymbol{v})}. \tag{65}$$

Let $z \in \mathcal{Z}$ be a trajectory whose translated sub-trajectory under $\mathcal{T}_k$ contains a critical routing step in the sense of Assumption 2(ii), and set

$$\boldsymbol{\chi} = \Phi(z), \qquad \boldsymbol{\chi}' = \Phi(\mathcal{T}_k(z)). \tag{66}$$

Inserting the k-token inert node via $\mathcal{T}_k$ shifts the subsequent sub-trajectory to later token indices (Definition 2), which by Lemma 2 alters the local APE attention scores almost surely. At the critical routing step, Assumption 2(ii) yields

$$|\mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi}') - \mathcal{L}_{f_{\text{APE}}}(\boldsymbol{\chi})| \geq \gamma. \tag{67}$$

Since $d_{\mathcal{X}}(\boldsymbol{\chi}, \boldsymbol{\chi}') = \delta \le C_\delta$, evaluating the difference quotient at $(\boldsymbol{\chi}, \boldsymbol{\chi}')$ gives

$$K_{f,\mathrm{APE}} \ge \frac{|\mathcal{L}_{f_{\mathrm{APE}}}(\boldsymbol{\chi}') - \mathcal{L}_{f_{\mathrm{APE}}}(\boldsymbol{\chi})|}{d_{\mathcal{X}}(\boldsymbol{\chi}, \boldsymbol{\chi}')} \ge \frac{\gamma}{\delta} \ge \frac{\gamma}{C_\delta} = \Omega(1). \tag{68}$$

Combining this with $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$ from Assumption 2(i) yields $K_{f,\mathrm{RoPE}} \ll K_{f,\mathrm{APE}}$ for sufficiently large $T_{\max}$. □

D Proofs of Capacity Bounds

D.1 Structural Capacity Analysis

Definition 3 (Bayes-Optimal Structural Risk)**.** *At structural state $\boldsymbol{\chi}$, taking the infimum over predictors h of the initial problem defines the Bayes-optimal structural risk*

$$\mathcal{L}^*(\boldsymbol{\chi}) := \inf_h \mathbb{E}_{z\sim\mu_T}[\ell(h, z) \mid \Phi(z) = \boldsymbol{\chi}], \qquad \boldsymbol{\chi} \in \mathrm{supp}(\nu_T). \tag{69}$$

Under Assumption 4(i), $\mathcal{L}^*$ admits a Barron extension to $\mathcal{X}$.

Definition 4 (Barron Space)**.** *Let $\mathcal{X} \subset \mathbb{R}^{12}$ be compact. The Barron space $\mathcal{H}_{\mathrm{Barron}}(\mathcal{X})$ consists of continuous functions $g : \mathcal{X} \to \mathbb{R}$ that admit an extension to $\mathbb{R}^{12}$ whose Fourier transform $\hat{g}$ satisfies*

$$\int_{\mathbb{R}^{12}} \|\boldsymbol{\omega}\|_1 \, |\hat{g}(\boldsymbol{\omega})| \, d\boldsymbol{\omega} < \infty. \tag{70}$$

The infimum over such extensions defines the Barron seminorm $\|g\|_{\mathcal{B}}$ [Barron, 1993].

Assumption 3 (Tsybakov Margin Condition)**.** *There exist constants $C > 0$ and $\alpha \ge 0$ such that for the conditional label probability $\eta(\boldsymbol{\chi}) := \mathbb{P}_{z\sim\mu_T}(\ell(f, z) = 1 \mid \Phi(z) = \boldsymbol{\chi})$,*

$$\nu_T\big(|\eta(\boldsymbol{\chi}) - 1/2| \le t\big) \le C t^\alpha, \quad \forall t > 0.$$

While margin conditions such as Assumption 3 provide a standard route for relating $L_1(\nu_T)$ approximation error to excess classification risk [Tsybakov, 2004], comparing $\mathcal{L}_f$ against the nonnegative pointwise Bayes risk $\mathcal{L}^*$ (Definition 3) yields the lower bound of Lemma 4 directly: in Appendix D.1.1, combining this comparison with Assumption 1 produces a factor of $1/2$ and a residual of $\epsilon/2$, after which Assumption 4 supplies the $m^{-1/2}$ width scaling.

Assumption 4 (Structural Capacity Mapping and Approximation Limit)**.** *We assume three conditions on the structural regression problem over $\mathcal{X}$: **(i) Barron regularity of the target risk.** The Bayes-optimal structural risk belongs to the Barron space:*

$$\mathcal{L}^* \in \mathcal{H}_{\mathrm{Barron}}(\mathcal{X}), \qquad 0 < \|\mathcal{L}^*\|_{\mathcal{B}} < \infty. \tag{71}$$

***(ii) Finite-width structural hypothesis class.** For each autoregressive Transformer $f \in \mathcal{F}_{L,m,\mathrm{PE}}$, the induced structural risk function $\mathcal{L}_f$ belongs to a width-constrained hypothesis class*

$$\mathcal{L}_f \in \mathcal{H}_m \subset \mathcal{H}_{\mathrm{Barron}}(\mathcal{X}), \tag{72}$$

*whose approximation capacity is governed by the internal representation width m. **(iii) Width-limited approximation lower bound.** There exists a constant $C_{\mathcal{B}} > 0$, independent of m, such that*

$$\inf_{g\in\mathcal{H}_m} \int_{\mathcal{X}} |g(\boldsymbol{\chi}) - \mathcal{L}^*(\boldsymbol{\chi})| \, d\nu_T(\boldsymbol{\chi}) \ge \frac{C_{\mathcal{B}}}{\sqrt{m}}, \tag{73}$$

matching the $\mathcal{O}(m^{-1/2})$ approximation rate characteristic of Barron spaces [Barron, 1993, Makovoz, 1996].

D.1.1 Proof of Lemma 4

Proof. First, Assumption 1 lower-bounds the target risk by the expected structural risk up to ϵ:

$$\mathcal{R}_{\mu_T}(f) \ge \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi}) \, d\nu_T(\boldsymbol{\chi}) - \epsilon. \tag{74}$$

By definition of the pointwise Bayes risk $\mathcal{L}^*$, we also have $\int_{\mathcal{X}} \mathcal{L}^* \, d\nu_T \leq \mathcal{R}_{\mu_T}(f)$. Adding these two inequalities yields $\int_{\mathcal{X}} (\mathcal{L}_f + \mathcal{L}^*) \, d\nu_T \leq 2\mathcal{R}_{\mu_T}(f) + \epsilon$. Since both $\mathcal{L}_f$ and $\mathcal{L}^*$ are nonnegative, $|\mathcal{L}_f - \mathcal{L}^*| \leq \mathcal{L}_f + \mathcal{L}^*$ pointwise, and hence $\mathcal{R}_{\mu_T}(f) \geq \frac{1}{2} \int_{\mathcal{X}} |\mathcal{L}_f - \mathcal{L}^*| \, d\nu_T - \frac{\epsilon}{2}$.

Second, because $\mathcal{L}_f \in \mathcal{H}_m$, the approximation lower bound in Assumption 4(iii) gives

$$\int_{\mathcal{X}} |\mathcal{L}_f(\boldsymbol{\chi}) - \mathcal{L}^*(\boldsymbol{\chi})| \, d\nu_T(\boldsymbol{\chi}) \geq \frac{C_{\mathcal{B}}}{\sqrt{m}}. \tag{75}$$

Substituting this inequality and setting $C_{\text{width}} := C_{\mathcal{B}}/2$ yields $\mathcal{R}_{\mu_T}(f) \geq C_{\text{width}}/\sqrt{m} - \epsilon/2 \geq C_{\text{width}}/\sqrt{m} - \epsilon$. □

D.2 Depth Bottleneck via Dyck-Type Reduction

We first formalize how depth-first search transitions map onto balanced parentheses whose open brackets record unfinished branches, and then invoke our implicit-depth assumption to lower-bound the risk of a single L-layer forward pass.

D.2.1 DFS-to-Dyck Encoding

We give the full construction for Lemma 3. Let k denote the maximum branching factor of the search tree, and define the search transition alphabet

$$\Sigma_{\text{search}} := \{\delta_{\text{down}}^{(1)}, \ldots, \delta_{\text{down}}^{(k)}\} \cup \{\delta_{\text{up}}^{(1)}, \ldots, \delta_{\text{up}}^{(k)}\} \cup \{\omega_{\text{eval}}\}, \tag{76}$$

where $\delta_{\text{down}}^{(c)}$ denotes expansion to the c-th child, $\delta_{\text{up}}^{(c)}$ denotes backtracking from the c-th child, and ω_{eval} denotes terminal evaluation.

Let $\Sigma_{\text{Dyck}} = \{(_1,)_1, \ldots, (_k,)_k\}$, and let $\mathcal{D}_k \subset \Sigma_{\text{Dyck}}^*$ denote the Dyck-k language [Strobl et al., 2024]. Define the mapping

$$\psi : \Sigma_{\text{search}}^* \to \Sigma_{\text{Dyck}}^* \tag{77}$$

on generators by

$$\psi(\delta_{\text{down}}^{(c)}) = (_c, \qquad \psi(\delta_{\text{up}}^{(c)}) =)_c, \qquad \psi(\omega_{\text{eval}}) = \varepsilon, \tag{78}$$

and extend it homomorphically via $\psi(x \oplus y) = \psi(x) \oplus \psi(y)$ for all $x, y \in \Sigma_{\text{search}}^*$.

Viewing the visited nodes of a valid trajectory z as a walk on the search tree and expanding each step into parent–child transitions, depth-first search transitions correspond directly to pushes and pops on the search stack. Under ψ, the unmatched opening brackets at any step record the active path from the root to the current node, so $\psi(z)$ is always a prefix of a word in $\mathcal{D}_k$ (and forms a balanced word precisely when the walk returns to the root). Because the root node has identifier depth $D_i = 1$ in Appendix A.1, a trajectory with node references and maximum tree depth χ_1 has maximum bracket nesting depth $\chi_1 - 1$.

D.2.2 Complexity-Theoretic Interpretation and Finite Precision

At a backtracking transition $v_{t+1} = \delta_{\text{up}}^{(c)}$, selecting the correct branch index c requires identifying the most recent unmatched opening transition $(_c$ in the search history. While a single p-bit activation takes at most 2^p distinct values, a Transformer distributes state across multiple tokens and hidden channels; nonetheless, for the hard- and soft-attention models analyzed by Hahn [2020], holding the layer count and head count fixed restricts the ability to recognize hierarchical languages such as Dyck as sequence length grows.

Motivated by this limitation and the TC^0 circuit bounds for bounded-precision Transformers [Merrill and Sabharwal, 2023], Assumption 5 posits a linear relationship between the maximum tree depth χ_1 of a trajectory and the sequential circuit depth required to resolve its backtracking transitions within a single forward pass. Its reciprocal $\alpha_{\text{circ}} := c^{-1}$ (which may depend on the task and numerical precision p) specifies the maximum tree depth resolvable per physical Transformer layer.

Assumption 5 (Implicit State-Transition Complexity via Dyck Structure)**.** *There exists a precision-dependent constant $c > 0$ such that resolving a backtracking transition of structural depth χ_1 requires implicit computational depth at least*

$$D_{\text{implicit}} \geq c \, \chi_1. \tag{79}$$

Setting $\alpha_{\mathrm{circ}} := c^{-1}$, trajectories with $\chi_1 > \alpha_{\mathrm{circ}} L$ exceed the sequential depth available in a single L-layer forward pass, and we assume that every $f \in \mathcal{F}_{L,m,\mathrm{PE}}$ incurs structural risk at least that of uniform guessing over $\mathcal{V}$: $\mathcal{L}_f(\boldsymbol{\chi}) \geq 1 - 1/|\mathcal{V}|$ whenever $\chi_1 > \alpha_{\mathrm{circ}} L$.

D.2.3 Failure Beyond the Implicit Depth Threshold

Define the depth-exceeding region

$$\mathcal{X}_{\mathrm{severed}} := \{\boldsymbol{\chi} \in \mathcal{X} : \chi_1 > \alpha_{\mathrm{circ}} L\}, \tag{80}$$

on which Assumption 5 lower-bounds the structural risk by the uniform guessing error rate

$$C_{\mathrm{depth}} := 1 - \frac{1}{|\mathcal{V}|}. \tag{81}$$

We therefore define the depth-truncation indicator

$$P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}}) := \mathbb{I}_{(\alpha_{\mathrm{circ}} L, \infty)}(\chi_1), \tag{82}$$

which equals 1 precisely when $\boldsymbol{\chi} \in \mathcal{X}_{\mathrm{severed}}$.

D.2.4 Proof of Lemma 5

For completeness, we restate Lemma 5, which holds under Assumptions 1 and 5. Let $|\mathcal{V}| \geq 2$, and define $C_{\mathrm{depth}} := 1 - \frac{1}{|\mathcal{V}|} \in (0, 1)$. Then for any autoregressive Transformer hypothesis $f \in \mathcal{F}_{L,m,\mathrm{PE}}$,

$$\mathcal{R}_{\mu_T}(f) \geq C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_T(\boldsymbol{\chi}) - \epsilon. \tag{83}$$

Proof. For $\boldsymbol{\chi} \in \mathcal{X}_{\mathrm{severed}}$, Assumption 5 gives $\mathcal{L}_f(\boldsymbol{\chi}) \geq C_{\mathrm{depth}}$. Splitting the integral of $\mathcal{L}_f$ over $\mathcal{X}_{\mathrm{severed}}$ and its complement $\mathcal{X} \setminus \mathcal{X}_{\mathrm{severed}}$ and using $\mathcal{L}_f \geq 0$ yields

$$\begin{aligned}
\int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) &= \int_{\mathcal{X}_{\mathrm{severed}}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) + \int_{\mathcal{X} \setminus \mathcal{X}_{\mathrm{severed}}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) && (84)\\
&\geq \int_{\mathcal{X}_{\mathrm{severed}}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) && (85)\\
&\geq C_{\mathrm{depth}}\, \nu_T(\mathcal{X}_{\mathrm{severed}}) && (86)\\
&= C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_T(\boldsymbol{\chi}). && (87)
\end{aligned}$$

Applying Assumption 1,

$$\mathcal{R}_{\mu_T}(f) \geq \int_{\mathcal{X}} \mathcal{L}_f(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \epsilon, \tag{88}$$

which establishes the bound. □

D.2.5 Architecture-Invariance of the Physical Depth Limit

Lemma 6 (Architecture-Invariance of the Physical Depth Limit)**.** *The constant α_{circ} is determined by the sequential circuit depth required by the backtracking task and is therefore independent of the positional encoding scheme: $\alpha_{\mathrm{APE}} = \alpha_{\mathrm{RoPE}} = \alpha_{\mathrm{circ}}$.*

Proof. A single autoregressive forward pass executes L sequential Transformer layers regardless of whether positions are encoded via APE or RoPE. While RoPE preserves relative position offsets within the attention kernel (Appendix C.3), it does not increase the number of sequential composition layers in a forward pass, so the same implicit depth threshold $\alpha_{\mathrm{circ}} L$ applies to both encodings. □

E Proof of the Main Theorem and Corollaries

E.1 Proof of Theorem 2

For completeness, we restate Theorem 2, which holds under Assumptions 1, 4, and 5 and the Lipschitz continuity of $\mathcal{L}_f$ in Lemma 1. For any autoregressive Transformer hypothesis $f \in \mathcal{F}_{L,m,\mathrm{PE}}$, the target risk satisfies

$$\mathcal{R}_{\mu_T}(f) \geq \max \left\{ \begin{array}{c} \dfrac{C_{\mathrm{width}}}{\sqrt{m}} - \epsilon,\ C_{\mathrm{depth}} \displaystyle\int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_T(\boldsymbol{\chi}) - \epsilon, \\ \mathcal{R}_{\mu_S}(f) - K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T) - 2\epsilon \end{array} \right\}, \tag{89}$$

and $\mathcal{R}_{\mu_T}(f) \leq \mathcal{R}_{\mu_S}(f) + K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T) + 2\epsilon$.

Proof. Lemmas 4 and 5 give

$$\mathcal{R}_{\mu_T}(f) \geq \frac{C_{\mathrm{width}}}{\sqrt{m}} - \epsilon \quad \text{and} \quad \mathcal{R}_{\mu_T}(f) \geq C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_T(\boldsymbol{\chi}) - \epsilon.$$

Furthermore, Lemma 1 gives $|\mathcal{R}_{\mu_T}(f) - \mathcal{R}_{\mu_S}(f)| \leq K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T) + 2\epsilon$, which yields both the upper bound and the third lower bound $\mathcal{R}_{\mu_T}(f) \geq \mathcal{R}_{\mu_S}(f) - K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T) - 2\epsilon$. Because all three lower bounds hold simultaneously for every $f \in \mathcal{F}_{L,m,\mathrm{PE}}$, $\mathcal{R}_{\mu_T}(f)$ is bounded below by their maximum. □

E.2 Proof of Corollary 1

For completeness, we restate Corollary 1 and note a $T_{\max}$-independent upper bound on the RoPE transport term. Assume the source risk satisfies $\mathcal{R}_{\mu_S}(f_{\mathrm{RoPE}}) = \epsilon_{\mathrm{train}} \approx 0$, and suppose the target shift satisfies $W_1(\nu_S, \nu_T) = \Omega(T_{\max})$. Then the RoPE target risk degrades at most proportionally to the normalized shift: $\mathcal{R}_{\mu_T}(f_{\mathrm{RoPE}}) \leq \epsilon_{\mathrm{train}} + C_{\mathrm{smooth}}\, W_1(\nu_S, \nu_T)/T_{\max} + 2\epsilon \leq \epsilon_{\mathrm{train}} + C_{\mathrm{RoPE}} + 2\epsilon$, where $K_{f,\mathrm{RoPE}} \leq C_{\mathrm{smooth}}/T_{\max}$ (Theorem 1) and $C_{\mathrm{RoPE}} := C_{\mathrm{smooth}} \sup_{T_{\max}} \mathrm{diam}(\mathcal{X})/T_{\max} < \infty$ is independent of $T_{\max}$.

In contrast, suppose there exists a measurable subset $\mathcal{X}_{\mathrm{shift}} \subset \mathcal{X}$ with $\nu_T(\mathcal{X}_{\mathrm{shift}}) = p_{\mathrm{shift}} > 0$ on which $\mathcal{L}_{f_{\mathrm{APE}}}(\boldsymbol{\chi}) \geq \gamma$ for all $\boldsymbol{\chi} \in \mathcal{X}_{\mathrm{shift}}$, where $\gamma = \Omega(1)$. Then $\mathcal{R}_{\mu_T}(f_{\mathrm{APE}}) \geq \gamma p_{\mathrm{shift}} - \epsilon$, which remains bounded away from zero whenever $\epsilon < \gamma p_{\mathrm{shift}}$ by a fixed margin.

Proof. For RoPE, Lemma 1 gives

$$\mathcal{R}_{\mu_T}(f_{\mathrm{RoPE}}) \leq \mathcal{R}_{\mu_S}(f_{\mathrm{RoPE}}) + K_{f,\mathrm{RoPE}} W_1(\nu_S, \nu_T) + 2\epsilon. \tag{90}$$

By Theorem 1, $K_{f,\mathrm{RoPE}} \leq C_{\mathrm{smooth}}/T_{\max}$, so $K_{f,\mathrm{RoPE}} W_1(\nu_S, \nu_T) \leq C_{\mathrm{smooth}}\, W_1(\nu_S, \nu_T)/T_{\max} \leq C_{\mathrm{RoPE}}$ since $W_1(\nu_S, \nu_T) \leq \mathrm{diam}(\mathcal{X}) = \mathcal{O}(T_{\max})$. Substituting $\mathcal{R}_{\mu_S}(f_{\mathrm{RoPE}}) = \epsilon_{\mathrm{train}}$ yields both bounds.

For APE, Assumption 1 and nonnegativity of $\mathcal{L}_{f_{\mathrm{APE}}}$ yield

$$\begin{aligned} \mathcal{R}_{\mu_T}(f_{\mathrm{APE}}) &\geq \int_{\mathcal{X}} \mathcal{L}_{f_{\mathrm{APE}}}(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \epsilon && (91) \\ &\geq \int_{\mathcal{X}_{\mathrm{shift}}} \mathcal{L}_{f_{\mathrm{APE}}}(\boldsymbol{\chi})\, d\nu_T(\boldsymbol{\chi}) - \epsilon && (92) \\ &\geq \gamma\, \nu_T(\mathcal{X}_{\mathrm{shift}}) - \epsilon = \gamma p_{\mathrm{shift}} - \epsilon. && (93) \end{aligned}$$

□

E.3 Proof of Corollary 2

For completeness, we restate Corollary 2. Suppose the target measure assigns positive mass to trajectories exceeding the single-pass depth threshold: $p_{\mathrm{deep}} := \nu_T(\{\boldsymbol{\chi} \in \mathcal{X} : \chi_1 > \alpha_{\mathrm{circ}} L\}) > 0$. Then $\mathcal{R}_{\mu_T}(f) \geq C_{\mathrm{depth}}\, p_{\mathrm{deep}} - \epsilon$, imposing a non-vanishing risk floor whenever the right-hand side is bounded away from zero. As $m \to \infty$, the width-based lower bound $\frac{C_{\mathrm{width}}}{\sqrt{m}} - \epsilon$ tends to $-\epsilon \leq 0$,

so its positive part vanishes while leaving the depth term unchanged. Conversely, increasing the layer count to $L \geq \frac{1}{\alpha_{\mathrm{circ}}} \sup_{\boldsymbol{\chi} \in \mathrm{supp}(\nu_T)} \chi_1$ ensures $\nu_T(\{\boldsymbol{\chi} \in \mathcal{X} : \chi_1 > \alpha_{\mathrm{circ}} L\}) = 0$, eliminating the depth floor on $\mathrm{supp}(\nu_T)$.

Proof. By Lemma 5 and the definition $P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}}) = \mathbb{I}_{(\alpha_{\mathrm{circ}} L, \infty)}(\chi_1)$,

$$\mathcal{R}_{\mu_T}(f) \geq C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_T(\boldsymbol{\chi}) - \epsilon = C_{\mathrm{depth}}\, p_{\mathrm{deep}} - \epsilon. \tag{94}$$

Meanwhile, $\lim_{m \to \infty}(C_{\mathrm{width}}/\sqrt{m} - \epsilon) = -\epsilon \leq 0$. Finally, when $L \geq \alpha_{\mathrm{circ}}^{-1} \sup_{\boldsymbol{\chi} \in \mathrm{supp}(\nu_T)} \chi_1$, no point in $\mathrm{supp}(\nu_T)$ satisfies $\chi_1 > \alpha_{\mathrm{circ}} L$, so $p_{\mathrm{deep}} = 0$. □

F Algebraic Equivalence Classes of Positional Encodings

Theorem 1 compares learned APE and RoPE by testing whether shifting two tokens by the same offset $+k$ alters their pre-softmax attention score when their content vectors are held fixed. Applying the same translation test partitions common positional encodings into two classes, alongside architectures without explicit positional encodings (NoPE) [Kazemnejad et al., 2023].

1. Position-dependent attention kernels. Under learned absolute embeddings or additive sinusoidal embeddings, the attention score $A_{i,j}$ depends explicitly on the absolute indices (i, j). Lemma 2 shows that for generic learned APE parameters, a common translation $+k$ alters the score almost surely; when this perturbation misroutes search at a critical transition (Assumption 2), structural risk changes by at least γ over a displacement $\delta \leq C_\delta$, forcing $K_f = \Omega(1)$.

2. Translation-invariant attention kernels. RoPE parameterizes attention via relative rotations between queries and keys, whereas ALiBi [Press et al., 2022] and T5 relative positional bias add a function of the relative offset $j - i$ directly to the attention score. Because a common shift $+k$ preserves $j - i$, all three mechanisms satisfy $\Delta A = 0$ for fixed content vectors and inherit the baseline Lipschitz scaling $K_f = \mathcal{O}(1/T_{\max})$ under Assumption 2(i).

Finally, NoPE omits explicit positional embeddings altogether, so its local attention kernel trivially satisfies $\Delta A^{\mathrm{NoPE}} = 0$ for fixed content vectors. However, because causal masking breaks permutation symmetry across sequence positions, early Transformer layers can reconstruct absolute position indices in their hidden activations [Kazemnejad et al., 2023]; how NoPE responds to sequence translations in practice therefore depends on whether downstream layers rely on those reconstructed absolute coordinates.

G Estimation of Structural Wasserstein Distances

Theorem 2 expresses the domain-shift penalty in terms of the population Wasserstein-1 distance $W_1(\nu_S, \nu_T)$ between source and target reference-search distributions. To compare these distributions empirically, we sample reference trajectories from each distribution, standardize the twelve structural coordinates jointly across the pooled sample, and solve the exact unregularized optimal transport problem between the resulting empirical measures, denoting the sample estimate by $\widehat{W}_1$.

G.1 Computational Pipeline

We apply a uniform coordinate scaling across all comparisons. Let $\hat{\nu}_S$ and $\hat{\nu}_T$ denote the empirical measures supported on the standardized structural vectors drawn from the source and target distributions.

Joint standardization. Because search volume χ_4 counts node visits whereas transition rates χ_5, χ_6, χ_7 are proportions in $[0, 1]$, unscaled Euclidean distances would be dominated by differences in node counts. We therefore compute the mean and standard deviation of each coordinate across the pooled BFS, DFS, and MIXED trajectory pools and standardize all three distributions using these shared statistics.

Cost matrix construction. For standardized structural vectors $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^{12}$, we define the pairwise Euclidean cost matrix

$$M_{ij} = \|\boldsymbol{u}_i - \boldsymbol{v}_j\|_2. \tag{95}$$

For the Sinkhorn cross-check, we normalize the cost matrix to $[0, 1]$ via

$$\widetilde{M}_{ij} = \frac{M_{ij}}{\max_{i,j} M_{ij} + 10^{-9}}. \tag{96}$$

Exact transport. Assigning uniform weight $1/N$ to each sampled trajectory, we compute $\widehat{W}_1(\hat{\nu}_S, \hat{\nu}_T) = \min_{\mathbf{P}\in\Pi(\mathbf{p},\mathbf{q})}\langle \mathbf{P}, \mathbf{M}\rangle_F$ with $\mathbf{p} = \mathbf{q} = N^{-1}\mathbf{1}$ using the exact network-simplex solver in Python Optimal Transport [Flamary et al., 2021]. As a consistency check on the relative ordering of shifts, we also evaluate the debiased Sinkhorn divergence on $\widetilde{\mathbf{M}}$ with entropic regularization $\lambda = 0.1$ [Cuturi, 2013], subtracting within-sample self-transport terms to remove entropic bias. Both estimators place the within-distribution baselines well below the BFS–MIXED shift, which in turn lies below the BFS–DFS shift.

Subsampling. From each 500,000-trajectory training distribution, we draw a random pool of 20,000 trajectories to compute the joint standardization statistics. For each pairwise distance estimate, we draw $N = 2{,}000$ trajectories per side without replacement (using two disjoint samples of 2,000 when comparing a distribution against itself). We repeat this procedure across ten fixed random seeds and report the mean and standard deviation across draws.

G.2 Estimates and Interpretation

To benchmark between-distribution shifts against finite-sample variation, we compare disjoint subsamples (1) and (2) from the same distribution alongside subsamples from distinct distributions at matched sample size $N = 2{,}000$. Across the ten random draws, we obtain

$$\widehat{W}_1(\hat{\nu}^{(1)}_{\mathrm{BFS}}, \hat{\nu}^{(2)}_{\mathrm{BFS}}) = 0.19 \pm 0.06, \qquad \widehat{W}_1(\hat{\nu}_{\mathrm{BFS}}, \hat{\nu}_{\mathrm{MIXED}}) = 1.38 \pm 0.05,$$
$$\widehat{W}_1(\hat{\nu}_{\mathrm{BFS}}, \hat{\nu}_{\mathrm{DFS}}) = 2.97 \pm 0.05,$$

together with $\widehat{W}_1(\hat{\nu}^{(1)}_{\mathrm{DFS}}, \hat{\nu}^{(2)}_{\mathrm{DFS}}) = 0.28 \pm 0.05$ and $\widehat{W}_1(\hat{\nu}_{\mathrm{DFS}}, \hat{\nu}_{\mathrm{MIXED}}) = 1.76 \pm 0.06$. The BFS–MIXED distance exceeds the within-BFS baseline by more than sevenfold, and the BFS–DFS distance exceeds it by nearly sixteenfold, confirming that the twelve structural coordinates cleanly separate the three search strategies relative to resampling noise.

G.3 Empirical Calibration of the Theoretical Ceiling

Figure 1A compares model accuracies on the DFS evaluation set against an empirical depth ceiling motivated by Lemma 5. In Theorem 2, the circuit-depth lower bound states that

$$\mathcal{R}_{\mu_{\mathrm{DFS}}}(f) \geq C_{\mathrm{depth}} \int_{\mathcal{X}} P_{\mathrm{implicit}}(\boldsymbol{\chi}; L, \alpha_{\mathrm{circ}})\, d\nu_{\mathrm{DFS}}(\boldsymbol{\chi}) - \epsilon, \tag{97}$$

where the integral equals the fraction of DFS reference trajectories whose maximum tree depth χ_1 exceeds $\alpha_{\mathrm{circ}} L$. Because population accuracy is $1 - \mathcal{R}_{\mu_{\mathrm{DFS}}}(f)$, this risk floor translates into an upper bound on accuracy that rises in discrete steps as L increases (since χ_1 takes finitely many integer values, with $\chi_1 \leq 4$ on Countdown; Appendices A.1 and K.5).

To summarize the empirical ceiling across the layer counts evaluated in our 54-model grid on DFS (the most challenging of the three test sets), we fit a smooth exponential envelope:

$$\mathrm{Accuracy}_{\max}(L) = \mathcal{A}_{\max} - \beta \cdot \exp(-\alpha \cdot L), \tag{98}$$

with $\beta, \alpha > 0$ and horizontal asymptote $\mathcal{A}_{\max}$. While the depth term alone permits population accuracy up to $1+\epsilon$ once L is large enough for the truncated mass to vanish, the empirical asymptote $\mathcal{A}_{\max}$ also reflects task constraints outside the depth bound, primarily the evaluation generation limit of 512 new tokens (Appendix K). Within that 512-token budget, the ground-truth reference searches themselves reach the target on 46.45% of BFS test problems, 48.60% of DFS problems, and 49.15% of MIXED problems; most of our trained models exceed these reference rates within the same token budget, reaching a peak DFS accuracy of 53.60%.

Table 1: The nine Transformer backbones used across the 54 training runs. Within each parameter scale, we vary layer depth L and hidden width $d = m$ while holding core non-embedding parameter counts comparable.

Scale / Shape	Layers (L)	Width ($d = m$)	Heads (h)	Core Params[†]	Total Params[‡]
Small Deep	12	192	4	5.3 M	24.8 M
Small Balanced	6	256	4	4.7 M	30.7 M
Small Wide	3	384	4	5.3 M	44.3 M
Medium Deep	24	512	8	75.5 M	127.6 M
Medium Balanced	12	768	12	84.9 M	163.0 M
Medium Wide	6	1024	16	75.5 M	179.5 M
Large Deep	48	768	12	339.7 M	418.1 M
Large Balanced[*]	24	1024	16	302.0 M	406.2 M
Large Wide	12	1536	24	339.7 M	495.9 M

Each shape is trained under both positional encodings (APE, RoPE) and all three search distributions (BFS, DFS, MIXED).

[†]*Core Params* = $12Ld^2$*, excluding vocabulary and position embeddings.*

[‡]*Total Params* = $12Ld^2 + 10Ld + 2d + 2Vd + 1024d$ *($V = 50{,}257$), corresponding to untied GPT-Neo embeddings (an upper bound: APE ties input/output embeddings, and RoPE omits the learned position table).*

[*]*Width* $d = 1024$ *keeps Large Balanced close to the other Large shapes in core parameter count while maintaining integer head dimensions.*

Setting $\mathcal{A}_{\max} \approx 0.538$, $\beta \approx 0.035$, and $\alpha \approx 0.15$ places the envelope in Figure 1A above all 54 measured DFS accuracies. To isolate the effect of network width at fixed physical depth, we compare the highest DFS accuracy attained among the six models at each (L, d) configuration (spanning two positional encodings and three training distributions): at $L = 12$, increasing width from $d = 192$ to $d = 768$ raises peak accuracy by 1.55 percentage points (51.60% to 53.15%), whereas doubling d further to $1{,}536$ (quadrupling core parameters from 84.9M to 339.7M) slightly reduces peak accuracy by 0.20 points to 52.95%; at $L = 24$, doubling width from $d = 512$ to $d = 1{,}024$ raises peak accuracy by 0.60 points (52.80% to 53.40%); and at $L = 6$, quadrupling width from $d = 256$ to $d = 1{,}024$ raises peak accuracy by 0.95 points (51.30% to 52.25%).

The two width doublings starting from $d \geq 512$ ($L = 12$ and $L = 24$) also permit matched comparisons within each positional encoding and training distribution: across all six APE pairs, the wider model achieves lower DFS accuracy (by 0.25 to 9.70 percentage points), whereas the six matched RoPE pairs change by -0.30 to $+1.30$ points, leaving the best accuracy at each depth within 1 percentage point.

H Supplementary Empirical Analyses

H.1 Mitigation by Translation Equivariance

While Section 5 compares aspect ratios and positional encodings across training distributions, Figure 2 isolates the effect of aspect ratio when both training and evaluation are fixed to μ_{DFS} at the Large and Medium scales.

At the Large scale (Figure 2A), shifting parameters from width into depth ($L = 12 \to 24$) accounts for nearly all of the accuracy gain under APE, and this monotone ordering from Wide to Balanced to Deep holds across all nine training–evaluation combinations at the Large scale. In contrast, Large Wide RoPE already reaches 51.80% with $L = 12$ layers, slightly ahead of Large Deep RoPE (51.45% at $L = 48$). At the Medium scale (Figure 2B), 6-layer and 12-layer APE models already reach 51.15% and 51.50%, showing that 12 layers are physically sufficient to achieve the $\sim 51\%$ accuracy level on Countdown DFS.

Why, then, does the 12-layer Large Wide APE model ($d = 1{,}536$) drop to 48.35% on DFS (and 42.15% on MIXED)? By Lemma 6, APE and RoPE share the same single-pass circuit depth threshold $\alpha_{\mathrm{circ}} L$, but Corollary 1 shows that they differ in routing stability under sequence translations: perturbations to APE attention scores can misroute search decisions (Lemma 2, Assumption 2). The recovery of Large Balanced and Large Deep APE suggests that additional layers provide routing redundancy that partially compensates for position-index sensitivity in wide APE networks, whereas RoPE's translation-invariant kernel ($\Delta A^{\mathrm{RoPE}} = 0$) stabilizes routing directly at $L = 12$.

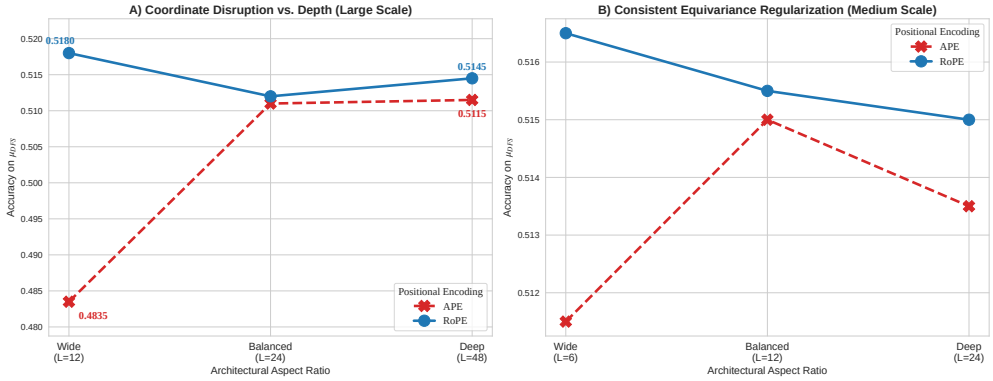


Figure 2: DFS test accuracy across Wide, Balanced, and Deep aspect ratios for models trained on μ_{DFS} under APE (red dashed) and RoPE (blue solid). **(A)** At the Large scale, increasing depth from $L = 12$ (Wide) to $L = 24$ (Balanced) and $L = 48$ (Deep) raises APE accuracy from 48.35% to 51.10% and 51.15%, whereas RoPE already achieves 51.80% at $L = 12$ and 51.45% at $L = 48$. **(B)** At the Medium scale ($L \in \{6, 12, 24\}$), APE achieves 51.15%, 51.50%, and 51.35%, while RoPE achieves 51.65%, 51.55%, and 51.50%—within 0.5 percentage points across all three shapes.

H.2 Baseline Capacity and Structural Variance

Figure 3 examines how sensitive each positional encoding is to aspect-ratio changes when training and evaluation use matching search strategies. For each parameter scale, training distribution, and positional encoding, we plot the mean accuracy across the Deep, Balanced, and Wide backbones alongside ± 1 standard deviation across the three shapes, which we term *structural variance*.

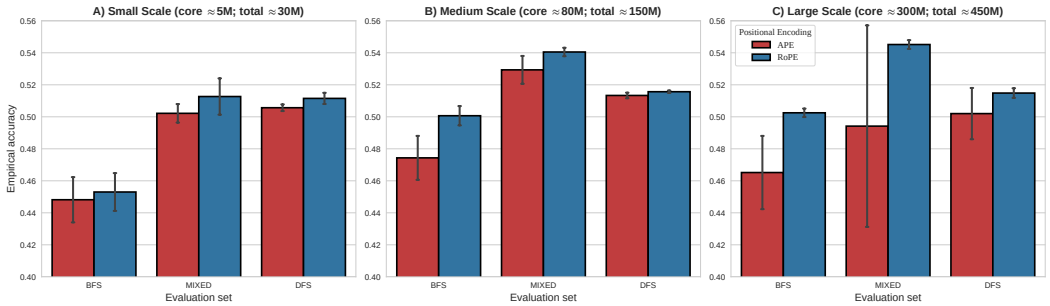


Figure 3: Accuracy on the BFS, MIXED, and DFS test sets across Small **(A)**, Medium **(B)**, and Large **(C)** parameter scales. Each bar reports the mean accuracy across the Deep, Balanced, and Wide backbones trained with the matching search strategy under APE (red) or RoPE (blue), with error bars indicating ± 1 standard deviation across the three aspect ratios. Structural variance across aspect ratios is largest for Large APE on MIXED.

At the Large scale (Figure 3C), APE accuracy increases monotonically with layer depth across all three training distributions, with the largest Deep–Wide gap (11.15 percentage points; Figure 1C) occurring on MIXED and driving the wide APE error bar. Across every distribution at the Medium and Large scales, RoPE exhibits smaller structural variance across aspect ratios than APE, and achieves a higher minimum accuracy across the three shapes in eight of the nine scale–distribution combinations (the sole exception being Small BFS, where the minima differ by 0.05 percentage points).

I Further Intuition on the Bounds

What bounds the accuracy of RoPE models. While Corollary 1 shows how RoPE avoids the translation-induced routing failures of APE—controlling the transport penalty $K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T)$ in Theorem 2—both encodings remain subject to the circuit-depth floor $C_{\mathrm{depth}} \int P_{\mathrm{implicit}}\, d\nu_T - \epsilon$

and the Barron-space width bound $C_{\mathrm{width}}/\sqrt{m} - \epsilon$. On MIXED training and evaluation, replacing APE with RoPE raises Large Wide accuracy from 42.15% to 54.25%, matching the other Large configurations; across all Large evaluation sets, peak accuracies at $L \in \{12, 24, 48\}$ lie within 1 percentage point of each other, and doubling the largest tested widths at fixed depth alters peak DFS accuracy by less than 1 point (Appendix G.3).

Controlling ϵ and the dimension of $\mathcal{X}$. Enriching the projection Φ with finer features—such as token n-grams or operand values—can reduce the smallest attainable sufficiency residual ϵ in Assumption 1, but increasing the ambient dimension of $\mathcal{X}$ also increases Euclidean distances and the sample complexity of empirical Wasserstein estimation [Weed and Bach, 2019]. Our twelve coordinates balance these two effects on Countdown: they separate the three search strategies well above sampling noise (Appendix G), while expanding to 34 coordinates yields negligible improvement in risk prediction (Appendix M.1.1).

How K_f enters the experiments. Under Assumption 2, the attention-score sensitivity of Lemma 2 implies $K_{f,\mathrm{APE}} = \Omega(1)$ versus $K_{f,\mathrm{RoPE}} = \mathcal{O}(1/T_{\max})$ (Theorem 1). Section 5 tests this separation across model scales, aspect ratios, and search distributions, while Experiment 6 (Appendix M.3.3) directly perturbs token position indices along identical search prefixes.

J Empirical Check of Structural Sufficiency (Assumption 1)

To evaluate Assumption 1 empirically, we test how accurately the twelve structural coordinates $\boldsymbol{\chi} = \Phi(z)$ of a problem's reference search predict whether a trained model solves that problem, and whether a predictor fitted on one search distribution transfers to estimate mean risk on another.

Setup and Dataset. As in Section 5, each model f receives only the initial problem prompt $x_{\mathrm{init}}^{(z)}$ and generates a trace via greedy decoding (Appendix K), incurring zero-one loss $\ell(f, z) = 1$ on failure and 0 on success (Appendix A.4); because decoding is deterministic, repeated prompts receive identical labels. We generate $50{,}000$ fresh reference trajectories (seed 2026) split nearly evenly across μ_{BFS} (16,667), μ_{DFS} (16,667), and μ_{MIXED} (16,666). When matched by target and unordered operands, 37.7% of these problems appear in at least one training set (39.3% for the $3{,}000$-trajectory subset used for f_2), nearly all of which are three-operand instances; we therefore also report results restricted to problems absent from all training sets (Table 2).

Evaluated Models and Feature Sets. We evaluate two models from Section 5 that differ in scale, training strategy, and positional encoding: Small Deep MIXED RoPE (f_1, evaluated on all $50{,}000$ trajectories) and Large Deep BFS APE (f_2, evaluated on the first $1{,}000$ trajectories per measure). For each model, we train predictors on five feature sets: (1) all twelve structural coordinates $\boldsymbol{\chi} = \Phi(z)$; (2) $\boldsymbol{\chi}$ excluding the reference solution indicator χ_{10}; (3) reference trace token length (LEN); (4) the problem specification (PROB: target value, operand count, and sorted operands, which specifies the prompt up to operand order); and (5) operand count alone.

Fitting and Scoring. We fit scikit-learn's `HistGradientBoostingClassifier` with default hyperparameters [Pedregosa et al., 2011] and evaluate out-of-fold predictions across stratified five-fold cross-validation using ROC-AUC and McFadden's pseudo-$R^2 = 1 - \log L_{\hat{\mathcal{L}}} / \log L_0$, where L_0 is the likelihood of a constant training-fold base-rate predictor [McFadden, 1979] (code and per-problem labels are provided in Appendix L). Table 2 codes solving success ($P(\ell = 0)$) as the positive class; we also fit failure ($P(\ell = 1)$). Because each training fold for f_1 exceeds $10{,}000$ examples, `HistGradientBoostingClassifier` enables internal validation-based early stopping whose split depends on label coding: flipping the label coding alters Table 2 metrics by at most 0.003 in most comparisons, and shifts cross-strategy mean risk predictions for f_1 (when trained on BFS or DFS alone) by up to 0.04.

Trajectory-Level Predictability. In Table 2, the twelve structural coordinates $\boldsymbol{\chi}$ achieve the highest out-of-fold ROC-AUC on the full sample (0.927 for f_1 and 0.929 for f_2) and mean McFadden pseudo-R^2 values of 0.51 and 0.48, well above the 0.2–0.4 threshold considered a strong fit [McFadden, 1979]. Dropping χ_{10} reduces ROC-AUC by at most 0.002. For f_1, $\boldsymbol{\chi}$ outperforms both LEN and PROB on both metrics in every fold; for f_2, $\boldsymbol{\chi}$ wins every fold on ROC-AUC and beats LEN in three of five folds on McFadden's index (0.48 vs. 0.46 for LEN and 0.41 for PROB). While operand count alone accounts for substantial baseline variation (models solve roughly 80% of 3-operand

Table 2: Out-of-fold prediction of per-problem loss ℓ by $\hat{\mathcal{L}}$: ROC-AUC and McFadden pseudo-R^2 (mean across five folds, with $\pm$ population s.d. across folds shown for $\boldsymbol{\chi}$). "Not in training": subset of problems absent from all training sets ($n = 31{,}173$ for f_1 and $1{,}822$ for f_2, predominantly 4-operand instances). "Within μ": five-fold cross-validation restricted to a single search measure ($16{,}667/16{,}667/16{,}666$ for f_1; $1{,}000$ each for f_2).

	f_1 ($n = 50{,}000$)		f_2 ($n = 3{,}000$)	
Input	ROC-AUC	McFadden	ROC-AUC	McFadden
$\boldsymbol{\chi}$	0.927 ± 0.002	0.51 ± 0.01	0.929 ± 0.007	0.48 ± 0.03
$\boldsymbol{\chi}$ without χ_{10}	0.926	0.50	0.927	0.47
LEN	0.903	0.44	0.917	0.46
PROB	0.890	0.40	0.906	0.41
Operand count only	0.797	0.27	0.853	0.40
$\boldsymbol{\chi}$, not in training	0.905	0.44	0.910	0.36
$\boldsymbol{\chi}$, within μ_{BFS}	0.905	0.44	0.940	0.49
$\boldsymbol{\chi}$, within μ_{DFS}	0.952	0.60	0.912	0.39
$\boldsymbol{\chi}$, within μ_{MIXED}	0.918	0.48	0.919	0.41

problems versus 10%–20% of 4-operand problems), adding the structural coordinates of the reference search raises ROC-AUC from 0.797 to 0.927 (McFadden $0.27 \to 0.51$) for f_1 and from 0.853 to 0.929 ($0.40 \to 0.48$) for f_2, ranking problem difficulty more accurately than the prompt specification PROB itself. On problems absent from all training sets, $\boldsymbol{\chi}$ retains an ROC-AUC of 0.905 (f_1) and 0.910 (f_2), with McFadden values of 0.44 and 0.36; within individual search distributions, ROC-AUC ranges from 0.905 to 0.952 for f_1 and from 0.912 to 0.940 for f_2.

Transfer Across Search Measures. Because $\mu_{\mathrm{BFS}}, \mu_{\mathrm{DFS}}, \mu_{\mathrm{MIXED}}$ sample problems from the same underlying distribution and the model sees only $x_{\mathrm{init}}^{(z)}$ (Appendix K.5), a fixed model has identical population risk across all three measures. On our fresh samples, the empirical failure rates on BFS, DFS, and MIXED are $0.502, 0.494, 0.500$ for f_1 and $0.511, 0.511, 0.490$ for f_2. Consequently, a constant predictor set to the pooled failure rate incurs a maximum empirical residual across the three measures of 0.004 for f_1 and 0.014 for f_2 (or 0.007 and 0.021 when calibrated on one strategy and evaluated on another). When we instead fit the non-constant classifier $\hat{\mathcal{L}}$ on the structural coordinates of one search strategy and apply it to another, target ROC-AUC ranges from 0.87 to 0.95, and the maximum absolute gap between the mean predicted risk and observed target risk is 0.043 for f_1 when fitting failure (0.061 when fitting success) and 0.038 for f_2.

Sub-Population Risk Tracking. To induce variation in ground-truth risk across sub-populations, we partition trajectories by search strategy, operand count, and target magnitude band (yielding 54 groups for f_1 and 18 for f_2). Across groups, the mean absolute gap between cross-fitted predicted risk and observed group risk is 0.031 for f_1 and 0.038 for f_2 using $\boldsymbol{\chi}$ (versus 0.297 and 0.353 for a constant predictor, 0.030 and 0.048 for LEN, and 0.010 and 0.016 for PROB). The maximum group gap under $\boldsymbol{\chi}$ is 0.114 (f_1) and 0.081 (f_2), exceeding the 95th-percentile maxima expected from binomial sampling noise alone (0.037 and 0.070); when trained on a single strategy and transferred to the groups of the other two, the maximum group error rises to 0.196 for f_1 (0.237 under success coding) and 0.116 for f_2. Across groups, predicted mean risk under $\boldsymbol{\chi}$ achieves $R^2 = 0.983$ (f_1) and 0.985 (f_2) against observed group risk (compared with ≈ 0 for a constant predictor, 0.986 and 0.977 for LEN, 0.998 and 0.996 for PROB, and 0.969 and 0.964 for operand count).

Summary. Overall, $\boldsymbol{\chi}$ strongly predicts per-problem solving success—outperforming LEN, PROB, and operand count in ROC-AUC on both models and generalizing to problems absent from training—while across the three full search distributions (which share one problem population), the empirical sufficiency residuals for a constant predictor round to 0.004 (f_1) and 0.014 (f_2).

K Experimental Details and Reproducibility

This section documents the compute infrastructure, architectural implementations, training hyperparameters, task formatting, and evaluation settings for the 54 Transformer models in Section 5.

K.1 Compute Environment

The original Countdown runs were executed on a Slurm-managed Linux cluster equipped with 4- and 8-GPU NVIDIA A100 (40 GB) nodes. Each training job was allocated one A100 GPU, 16 CPU cores, and 128 GB of RAM under a seven-day wall-clock limit, requiring 2.7–4.6 hours per run at the Small scale, 8.2–42.8 hours at Medium, and 18.7–162.1 hours at Large ($\approx$ 1,371 GPU-hours in total). Evaluation jobs used one A100 GPU, 24 CPU cores, and 128 GB of RAM ($\approx$ 490 GPU-hours in total), and generating the per-problem outcome labels in Appendix J required an additional 17 GPU-hours on single-GPU jobs (8 CPU cores, 48 GB RAM). Completing two missing configurations required a further 95 GPU-hours for retraining and evaluation. Trajectory generation, $\widehat{W}_1$ estimation (Appendix G), and gradient-boosting fits (Appendix J) ran on CPUs.

For the extended experiments in Appendix M, Experiments 1 and 2 ran on CPUs; Experiment 3 used $\approx$ 23.5 GPU-hours on 80 GB A100 and 48 GB L40S GPUs (with CPU harness grading); Experiment 4 used $\approx$ 0.45 GPU-hours on 80 GB H100 GPUs; Experiment 5 used $\approx$ 106 GPU-hours on H100 and L40S GPUs; Experiment 6 used $\approx$ 4.5 GPU-hours on 80 GB A100 and H100 GPUs; and Experiments 7 and 8 used $\approx$ 132 and $\approx$ 13 GPU-hours, respectively, on A100 and L40S GPUs. The released Countdown environment uses Python 3.10, PyTorch 2.5.1 (CUDA 12.4), Hugging Face `transformers` 4.57.1, `datasets` 4.4.1, and `accelerate` 1.12.0, with PyTorch SDPA attention for GPT-NeoX, eager attention for GPT-Neo, and deterministic cuDNN operations (`torch.backends.cudnn.deterministic = True`) during evaluation.

Retraining check. Independently retraining all 18 Small configurations of Section 5 from scratch using the released code and freshly regenerated training trajectories reproduces test accuracies across all three evaluation sets within 0.02 (mean absolute difference 0.006 across the 54 model–test-set pairs, using $\approx$ 72 GPU-hours on NVIDIA L40S and A100 GPUs).

K.2 Architectural Implementations

Each positional encoding is implemented using its standard Hugging Face causal LM backbone:

- **APE:** `GPTNeoForCausalLM` with learned absolute positional embeddings over a 1,024-token context window and full causal attention (`attention_types` set to global attention in every layer).
- **RoPE:** `GPTNeoXForCausalLM` with `rotary_pct = 1.0` (applying rotary embeddings to the full head dimension) and sequential residual blocks (`use_parallel_residual = False`) to match the layer ordering of GPT-Neo. Beyond positional encoding, the two implementations also differ in attention-logit scaling and input/output embedding weight tying.

Both model families use GELU activations (exact GELU in GPT-NeoX and the tanh approximation `gelu_new` in GPT-Neo) and vocabulary size $|\mathcal{V}| = 50{,}257$. Medium and Large models enable gradient checkpointing (`gradient_checkpointing = True`) during training to reduce activation memory. All models are trained and evaluated in `bfloat16`.

K.3 Training Hyperparameters

All 54 configurations were trained from scratch under identical optimization settings:

- **Data & Context:** 500,000 trajectories per training measure, padded or truncated to $T_{\max} = 1{,}024$ tokens.
- **Loss Function:** Standard autoregressive cross-entropy loss. Padding tokens are masked with label -100; because the `<|endoftext|>` token serves both as the padding token and as the start/end delimiter of each trajectory, these boundary delimiters are masked from the loss as well.
- **Batch Size:** Effective batch size of 32 trajectories per step across all scales (per-device batch size 32 for Small; per-device batch size 4 with 8 gradient accumulation steps for Medium and Large).

- **Learning Rate & Optimizer:** AdamW with peak learning rate $\eta = 8 \times 10^{-5}$, 1,000-step linear warmup followed by linear decay to zero, and weight decay 0.01.
- **Epochs & Seeds:** Exactly 3 training epochs per run, with global random seed fixed to 42 (`random.seed(42)`, `torch.manual_seed(42)`) for data shuffling and weight initialization.

K.4 The Countdown Task and the Stream of Search Format

In Countdown, a solver must combine input numbers via basic arithmetic operations $(+, -, \times, /)$ to reach a target integer, using each input number at most once in the reference search. For instance, given target 10 and inputs $[2, 3, 5]$, a depth-first search first tries $2 \times 5 = 10$ (leaving $[3, 10]$), explores the dead ends $3 + 10 = 13$ and $10 - 3 = 7$, and backtracks to the root; it next tries $2 \times 3 = 6$, hits two more dead ends (11 and 1), backtracks again, and finally expands $2 + 3 = 5$ followed by $5 + 5 = 10$ to reach the goal.

Stream of Search (SoS) [Gandhi et al., 2024] serializes both failed branches and successful paths into a single text trace. Below is the trajectory produced by the DFS generator on this instance (using the sum heuristic with pruning threshold equal to the target); at test time, the model receives only the first line and generates the remainder:

```
Current State: 10:[2, 3, 5], Operations: []
Exploring Operation: 2*5=10, Resulting Numbers: [3, 10]
Generated Node #0,0: 10:[3, 10] Operation: 2*5=10
Moving to Node #0,0
Current State: 10:[3, 10], Operations: ['2*5=10']
Exploring Operation: 3+10=13, Resulting Numbers: [13]
13,10 unequal: No Solution
Moving to Node #0,0
Current State: 10:[3, 10], Operations: ['2*5=10']
Exploring Operation: 10-3=7, Resulting Numbers: [7]
7,10 unequal: No Solution
Moving to Node #0
Current State: 10:[2, 3, 5], Operations: []
Exploring Operation: 2*3=6, Resulting Numbers: [5, 6]
Generated Node #0,1: 10:[5, 6] Operation: 2*3=6
...                          (two dead ends below node #0,1, then backtrack to #0)
Exploring Operation: 2+3=5, Resulting Numbers: [5, 5]
Generated Node #0,2: 10:[5, 5] Operation: 2+3=5
Moving to Node #0,2
Current State: 10:[5, 5], Operations: ['2+3=5']
Exploring Operation: 5+5=10, Resulting Numbers: [10]
10,10 equal: Goal Reached
```

K.5 Evaluation Procedure

Every model is evaluated on the same $2{,}000$ problems per search distribution: the BFS and DFS evaluation sets take the first $2{,}000$ trajectories from their respective training sets (generated with the training seed, and thus overlapping with training problem instances), while the MIXED evaluation set combines $1{,}000$ trajectories from each. Prompts consist solely of the initial state line `<|endoftext|>Current State: [Target]:[Nums], Operations: []` followed by a newline.

We decode greedily (`do_sample=False`, `num_beams=1`) for up to 512 new tokens, running RoPE models with left padding at batch size 32 and APE models with batch size 1 and no padding. The original SoS verifier [Gandhi et al., 2024] parses the operation list reported in the final state of the generated trace and checks arithmetic validity and target attainment (permitting operand reuse in the verifier); a positive score yields zero-one loss 0, and 1 otherwise.

Comparisons across measures. Because the BFS, DFS, and MIXED generators sample operand counts, operands, and targets from the same distribution, they differ only in how the reference solver traverses the search tree: DFS backtracks nearly four times as frequently as BFS (mean χ_6 of 0.18 vs. 0.05 on the evaluation sets, with maximum tree depth $\chi_1 \leq 3$ for DFS and $\chi_1 \leq 4$ for BFS), placing their pushforward distributions far apart in $\mathcal{X}$ (Appendix G).

Because a trained model receives only the initial problem prompt and is scored on its own generated search, its true population risk is identical across $\mu_{\mathrm{BFS}}, \mu_{\mathrm{DFS}}, \mu_{\mathrm{MIXED}}$. On the 3,000-problem subset of Appendix J, for example, Large Deep BFS APE exhibits failure rates of 0.511, 0.511, and 0.490 across the three samples, whereas on the fixed 2,000-problem evaluation sets its accuracy drops from 48.15% on BFS to 45.55% on DFS due to finite-sample differences in problem difficulty across the three fixed test sets. Averaging across the 18 models trained under each strategy, mean test accuracies on (BFS, MIXED, DFS) are $(47.40\%, 46.99\%, 45.13\%)$ for BFS-trained models, $(51.32\%, 52.14\%, 51.05\%)$ for DFS-trained models, and $(51.75\%, 52.07\%, 51.10\%)$ for MIXED-trained models. While the DFS test sample is the hardest for all three groups, the BFS-to-DFS accuracy drop is largest for BFS-trained models (2.27 percentage points, versus 0.27 for DFS-trained and 0.65 for MIXED-trained models). All comparisons between architectures, scales, and positional encodings within any evaluation set use identical problem instances.

L Code Availability

Source code for data generation, model training, evaluation, and structural optimal-transport analysis is available at `https://github.com/Yyuzrah/AMTA4R`.

M Empirical Checks of the Terms of the Main Bound

In the main text, the structural projection Φ of Theorem 2 maps each Countdown reference search to the twelve coordinates $\boldsymbol{\chi} \in \mathbb{R}^{12}$ defined in Appendix A.1. To extend this analysis beyond Countdown, we first describe how to choose a structural projection Φ_D for a sequential decision domain D (Appendix M.1). We then test the terms of Theorem 2 in eight experiments, summarized in Table 3 (Appendix M.5).

- **Sufficiency residual (ϵ) vs. transport estimation (W_1).** Experiment 1 (Appendix M.1.1) evaluates nested coordinate expansions on Countdown (12, 16, 24 and 34 coordinates) to test whether adding finer features improves risk prediction enough to offset the higher sample complexity of estimating W_1.
- **Structural domain shift ($W_1(\nu_S, \nu_T)$).** Experiments 2 and 3 (Appendix M.2) apply domain-specific projections Φ_D to public logs of language agents, coding agents and robotic arms, testing whether $\widehat{W}_1$ separates structurally distinct tasks and captures shifts induced by changing an agent's execution harness while holding the model fixed.
- **Positional sensitivity and the Lipschitz factor ($K_{f,\mathrm{PE}}$).** Experiments 4–6 (Appendix M.3) test the predictions of Lemma 2 and the routing mechanism in Assumption 2 by shifting position indices. They measure attention scores in eight pretrained models, answers in 33 models under serving conditions (left padding, prefix caching and sliding windows), and search decisions in two trained Countdown models.
- **The TC^0 depth bottleneck (C_{depth}).** Experiments 7 and 8 (Appendix M.4) investigate whether decomposing deeply nested plans and tool-call chains into separate model calls restores success by reducing the nesting each call must resolve (Lemma 5).

All source code, per-pair, per-model and per-run results, and a catalog of candidate structural coordinates across Countdown and six additional domains are available at `https://github.com/Yyuzrah/AMTA4R`.

M.1 How the Coordinates of Φ Are Chosen

Given a sequential decision domain D, a structural projection Φ_D maps each variable-length trajectory z into a fixed-dimensional vector of d_Φ coordinates. Recall from Theorem 2 that the target risk upper bound adds the transport penalty $K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T)$ and 2ϵ (Assumption 1) to the source risk, whereas the lower bound takes the maximum over the source-transfer floor $\mathcal{R}_{\mu_S}(f) - K_{f,\mathrm{PE}} W_1(\nu_S, \nu_T) - 2\epsilon$, the Barron-space width bound $C_{\mathrm{width}}/\sqrt{m} - \epsilon$, and the circuit-depth floor $C_{\mathrm{depth}} \int P_{\mathrm{implicit}}\, d\nu_T - \epsilon$.

These terms guide how Φ_D is constructed from trajectory events (such as agent actions and observations or nodes visited in a search tree) via four design principles:

- **Controlling the sufficiency residual (ϵ) via structural abstraction.** Because ν_S and ν_T characterize task distributions, each coordinate is computed from a reference trajectory generated by a reference solver on the problem instance (and may include reference outcome indicators such as χ_{10}). To keep ϵ small while abstracting away surface semantics, coordinates should capture the types and sequential ordering of trajectory events, leaving instance-specific token identities (such as exact operand values or object names) to the residual ϵ (Appendix A.2).
- **Ensuring well-behaved transport costs (W_1) and homogeneous scaling.** Bounded coordinates suffice for $W_1(\nu_S, \nu_T)$ to remain finite, and choosing coordinate ranges that grow at most linearly with the maximum sequence length retains a diameter of order at most $T_{\max}$. Furthermore, because the Lipschitz constant $K_{f,\mathrm{PE}}$ is defined with respect to the Euclidean metric on $\mathcal{X}$, a unit displacement along any coordinate must carry a consistent meaning across the compared distributions. Categorical events therefore enter as frequencies or counts per category (analogous to the multiplicative and additive operator counts χ_{11} and χ_{12} in Countdown), and any feature standardization or weighting is applied uniformly across the compared distributions.
- **Preserving the Lipschitz gap ($K_{f,\mathrm{APE}} \gg K_{f,\mathrm{RoPE}}$) under sequence translation.** The proof of Theorem 1 lower-bounds $K_{f,\mathrm{APE}}$ by the risk jump γ divided by the Euclidean displacement δ induced by $\mathcal{T}_k$ (Definition 2), which inserts a single structurally inert event of k tokens (such as `x=x`). For the lower bound $\gamma/\delta = \Omega(1)$ to hold across admissible insertion lengths k and context limits $T_{\max}$, δ must remain strictly positive and bounded above independently of both, meaning that every coordinate must change by at most a bounded amount under $\mathcal{T}_k$. In Countdown, search volume χ_4 counts node references and increments by exactly one regardless of k, keeping δ positive, whereas raw token length would grow by k and drive γ/δ toward zero for long insertions.
- **Capturing the TC^0 depth bottleneck (C_{depth}) and controlling sample complexity.** At least one coordinate should record the maximum structural depth a single forward pass must resolve relative to the threshold in Assumption 5; in Countdown, where the root has identifier depth one, $\chi_1 - 1$ gives the Dyck-k parenthesis nesting depth of Lemma 3. Finally, keeping d_Φ compact limits the sample complexity of empirical W_1 estimation, which can grow rapidly with ambient dimension [Weed and Bach, 2019].

Because Lemmas 1 and 4 and Assumption 4 are formulated directly on $\mathcal{X}$, Theorem 2 retains its form whenever $\mathcal{X} \subset \mathbb{R}^{d_\Phi}$ is defined by a domain projection Φ_D with these properties and its assumptions hold in D. On Countdown, our twelve coordinates $\boldsymbol{\chi}$ consist of event counts, transition rates, and tree depths; coordinates other than χ_4 change by at most a bounded amount under admissible insertions of $\mathcal{T}_k$ (Appendix C.1), and Appendix J evaluates how accurately they predict task success.

M.1.1 Experiment 1: More Coordinates on Countdown

Motivation and trade-off. Expanding a projection Φ with additional coordinates introduces a trade-off in Theorem 2. On one hand, the smallest attainable sufficiency residual ϵ can only decrease or remain unchanged, since any risk function over the smaller feature set remains available. On the other hand, if the original coordinates retain their scales, Euclidean distances—and hence $W_1(\nu_S, \nu_T)$—can only increase or remain unchanged, and higher dimensionality also increases the sample complexity of estimating W_1 [Weed and Bach, 2019]. Consequently, expanding Φ tightens the upper bound only when the reduction in 2ϵ outweighs the growth in $K_{f,\mathrm{PE}} W_1$.

Nested projections and setup. Experiment 1 evaluates this trade-off on Countdown across four nested projections of $d \in \{12, 16, 24, 34\}$ coordinates, all sharing $\Phi_{12}(z) = \boldsymbol{\chi}(z)$ as their first twelve coordinates. The 22 additional coordinates are introduced in three cumulative stages: Φ_{16} adds four tree-shape features (e.g., node revisit ratio); Φ_{24} adds eight traversal and working-memory features (e.g., longest consecutive streak of failed operations); and Φ_{34} adds ten further memory, nesting, action, and surface-length features (including line count, with four of the 22 additional coordinates measured in tokens, such as trace length; note that the bounded-change requirement under $\mathcal{T}_k$ applies specifically to establishing the asymptotic gap of Theorem 1).

For each projection Φ_d, we perform two evaluations. First, following the out-of-fold evaluation of Appendix J, we train a gradient-boosting classifier on $\Phi_d(z)$ to predict whether a model solves

the instance of z, evaluating f_1 (Small Deep MIXED RoPE, $L = 12$) on $50{,}000$ problems and f_2 (Large Deep BFS APE, $L = 48$) on $3{,}000$ problems across ten repetitions of stratified five-fold cross-validation with identical fold splits across all four projections. Second, we compute empirical Wasserstein distances $\widehat{W}_1$ and within-measure disjoint-sample baselines across the BFS, DFS, and MIXED training distributions following Appendix G.

Results. Expanding from Φ_{12} to Φ_{34} alters mean out-of-fold ROC-AUC by less than 0.003 (remaining at 0.927–0.929 for both models) and mean McFadden pseudo-R^2 by less than 0.016. By contrast, the estimated transport distances nearly double: within-measure sampling baselines grow by a factor of 1.6–$2.3\times$ (e.g., from 0.19 to 0.44 on BFS), and between-measure distances grow by 1.8–$2.0\times$ (e.g., from 2.97 to 5.65 between BFS and DFS), while the ratio of mean baseline to mean between-measure distance remains near 0.13. Because the twelve baseline coordinates capture nearly all of the predictive signal without inflating transport distances, we adopt Φ_{12} on Countdown. In general, candidate coordinate expansions in a new domain can be screened by comparing the gain in risk prediction against the induced growth in $\widehat{W}_1$ and its sampling baseline.

M.2 Agent, Embodied and Robot Trajectories

We next extend our structural projections Φ_D and optimal transport diagnostics beyond synthetic search to interaction logs from language agents, software engineering agents, and robotic manipulation. Experiment 2 tests whether $\widehat{W}_1$ separates trajectory groups across tasks requiring distinct action sequences, while Experiment 3 isolates how a single coding model's trajectories and resolve rate change when it operates under two different agent harnesses on the same issues.

M.2.1 Experiment 2: Structural Shift between Groups of Trajectories

Benchmarks and Trajectory Sources. To evaluate whether $\widehat{W}_1$ assigns larger distances to tasks requiring distinct procedural steps and smaller distances to tasks sharing subgoal sequences, we analyze reference trajectories from four public benchmarks:

- **Language Agent Environments (ETO).** Expert trajectories released with Exploration-based Trajectory Optimization (ETO) [Song et al., 2024a,b] across three environments: ALFWorld household tasks ($3{,}119$ trajectories) [Shridhar et al., 2021], ScienceWorld laboratory experiments ($1{,}483$ trajectories) [Wang et al., 2022], and WebShop online shopping ($1{,}824$ trajectories, comprising 905 human and 919 GPT-4 demonstrations) [Yao et al., 2022].
- **Software Engineering Agents (SWE-rebench).** Resolved issue-fixing trajectories on SWE-rebench [Badertdinov et al., 2025] released by Nebius, comparing two model–harness pairings: $12{,}467$ resolved runs across 835 GitHub issues by the 70B Llama model `swe-agent-llama-70b` in the SWE-agent harness [Yang et al., 2024, Nebius, 2024], versus $32{,}161$ resolved runs across $3{,}792$ issues by Qwen3-Coder-480B-A35B-Instruct [Qwen Team, 2025] in the OpenHands harness [Wang et al., 2025, Trofimova et al., 2025].
- **Robotic Manipulation (BridgeData V2).** $33{,}252$ teleoperated robot-arm demonstrations with non-empty natural language instructions from BridgeData V2 [Walke et al., 2023], annotated with step-level movement primitives from Embodied Chain-of-Thought (ECoT) [Zawalski et al., 2025, Embodied-CoT, 2024].
- **Grounded Grid-World Navigation (BabyAI).** $4{,}000$ solved episodes per instruction category in BabyAI [Chevalier-Boisvert et al., 2019], generated with BALROG [Paglieri et al., 2025] and Minigrid [Chevalier-Boisvert et al., 2023] and solved using the reference BabyAI bot.

Projection and Evaluation. For each domain, we parse interaction logs into event sequences and track subgoal nesting (such as picking up and placing objects in ALFWorld or pushing and popping subgoals on the BabyAI bot's stack). Each domain projection Φ_D comprises 12 to 16 structural coordinates—including maximum and mean nesting depth, action counts, and domain-specific revisit and failure rates—with all projections except the coding-agent projection also including a small subset of surface coordinates (S) such as total action count.

Across these benchmarks, we construct 52 pairwise comparisons spanning six families: 15 pairs among the 6 ALFWorld task types, 3 pairs among the 3 ETO environments, 1 pair comparing human versus GPT-4 demonstrations in WebShop, 2 pairs comparing SWE-agent against OpenHands (once on the 554 GitHub issues solved by both systems and once across all resolved issues), 21 pairs among the 7 manipulation skills in BridgeData V2, and 10 pairs among the 5 instruction categories in BabyAI.

For groups exceeding 20,000 runs, we first subsample 20,000 runs without replacement. Following Appendix G, we standardize coordinates via pooled Z-scores within each family and compute $\widehat{W}_1$ over 10 random seeds using $N = 2{,}000$ units per sample or half the size of the smaller group, whichever is smaller ($N = 139$–$2{,}000$); for coding-agent comparisons we sample GitHub issues and draw one resolved run per sampled issue, whereas for all other comparisons we sample trajectories directly. To quantify shift magnitude relative to sampling noise, we define the separation ratio R as the mean between-group $\widehat{W}_1$ divided by the larger of the two within-group disjoint-sample baselines, and classify a pair as clearly separated when $R \geq 3$ and the coefficient of variation (CV) of the between-group $\widehat{W}_1$ across seeds is at most 0.10.

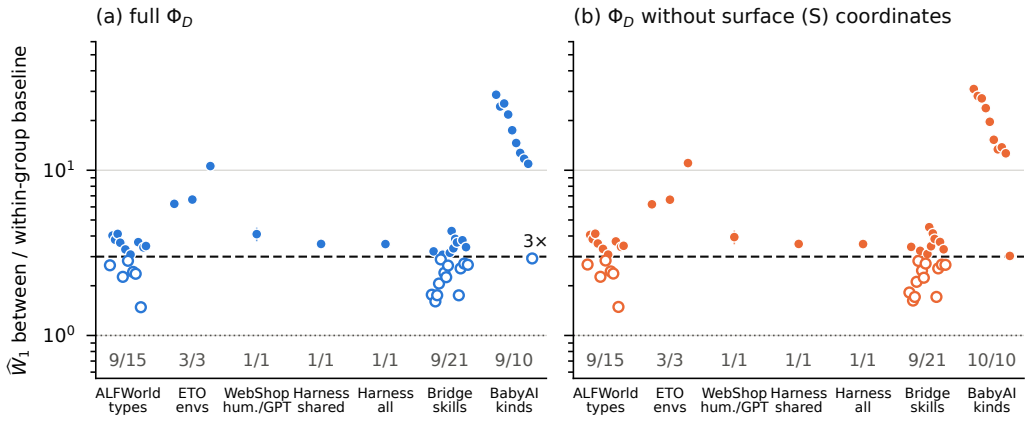


Figure 4: Structural shift between groups of reference trajectories (Experiment 2). The vertical axis shows the separation ratio R (between-group $\widehat{W}_1$ divided by the larger of the two within-group baselines) on a logarithmic scale, with one marker per group pair jittered horizontally within each column. **(a)** Full domain projection Φ_D. **(b)** Projection with surface coordinates (S) removed. Columns from left to right show the six ALFWorld task types, the three ETO environments, human versus GPT-4 demonstrations in WebShop (hum./GPT), SWE-agent versus OpenHands (where both model and harness differ) on the shared subset of mutually resolved issues (Harness shared) and on all resolved issues (Harness all), the seven BridgeData V2 skills, and the five BabyAI instruction categories. Filled markers indicate clearly separated pairs ($R \geq 3$, CV ≤ 0.10); hollow markers indicate unseparated pairs. The dashed line ($3\times$) marks $R = 3$ and the dotted line marks $R = 1$; error bars (mostly smaller than the markers) show ± 1 s.d. of the between-group $\widehat{W}_1$ across seeds divided by the larger baseline. Fractions along the bottom report the number of separated pairs per column.

Results. Figure 4 plots the separation ratio R across all 52 pairs. Between-group distances exhibit low variance across seeds (CV ≤ 0.095), with 33 of the 52 pairs meeting the separation threshold ($R \geq 3$), rising to 34 pairs when surface coordinates (S) are excluded (Figure 4b):

- **Distinct environments, demonstrators, and instructions.** $\widehat{W}_1$ clearly separates all 3 ETO cross-environment pairs ($R = 6.26$–10.60), 9 of 10 BabyAI instruction pairs ($R = 10.94$–28.65), human versus GPT-4 shopping strategies in WebShop ($R = 4.11$), both SWE-agent versus OpenHands coding comparisons ($R = 3.58$), and 9 of 15 ALFWorld task pairs.
- **Tasks with shared subgoal structure.** Task pairs that share similar action sequences yield ratios below $R = 3$: for example, heating versus cooling an object in ALFWorld produces the smallest ratio in the study ($R = 1.48$), and put-next-to versus pick-up-then-go-to in

BabyAI yields $R = 2.92$. Thus, in both ALFWorld and BabyAI, instructions that require structurally similar action sequences exhibit small separation ratios even when their task labels differ. In BridgeData V2, human teleoperation demonstrations vary substantially even within a single skill, yielding within-skill baselines of 1.23–1.99 (compared with maxima of 1.39 in ALFWorld and 0.61 in ETO); relative to these larger baselines, 9 of 21 skill pairs exceed $R = 3$.

M.2.2 Experiment 3: One Model in Two Agent Harnesses

Motivation and Setup. Because the coding-agent comparison in Experiment 2 varied both the model and its harness, Experiment 3 holds the underlying model fixed and varies only the agent harness—the scaffolding software that formats prompts, exposes tool interfaces, and executes commands—to test how tool structure alters both trajectory geometry and issue resolution.

We evaluate Qwen3-8B [Qwen Team, 2025] in bfloat16 with thinking mode disabled, served via vLLM [Kwon et al., 2023] under identical decoding parameters in both harnesses (temperature 0.7, top-p 0.8, top-k 20, up to 4,096 tokens per response, and at most 100 model calls per episode). We compare mini-SWE-agent (v2.4.6) [Yang et al., 2024], which provides only a single bash shell tool, against OpenHands [Wang et al., 2025], which exposes dedicated structured tools for file viewing and editing alongside bash execution.

Both harnesses are evaluated on the same repository-stratified sample of 300 human-verified Python issues from the 500-instance SWE-bench Verified benchmark [Jimenez et al., 2024, OpenAI, 2024b], with each harness attempting each issue once in a fresh container built from its official image. Using the 16 coding-agent coordinates from Experiment 2 (counting single-file bash reads and writes as file interactions), we compute $\widehat{W}_1$ and R over 10 random seeds with subsamples of $N = 150$ episodes drawn without replacement from each harness, and estimate within-harness baselines by splitting the 300 episodes of each harness into two random disjoint halves.

Results. Changing the harness substantially reshapes how Qwen3-8B works through an issue: the between-harness distance is $\widehat{W}_1 = 3.81$ (CV $= 0.024$) against within-harness baselines of 1.48 (mini-SWE-agent) and 1.74 (OpenHands), yielding an overall ratio of $R = 2.19$ relative to the larger baseline. Considered individually, each of the 16 coordinates exhibits a between-harness distance of $1.95\times$ to $11.5\times$ its within-harness baseline, with the largest shift occurring in code reading: lacking a dedicated file-viewing tool in mini-SWE-agent, Qwen3-8B averages 50.4 read commands (and 73.8 total tool calls) per episode, compared with just 17.3 read calls (43.5 total calls) in OpenHands.

This structural shift is accompanied by a marked difference in task success: across the 299 instances whose reference patches pass the test suite, Qwen3-8B's resolve rate nearly quadruples from 1.7% (5/299) under mini-SWE-agent to 6.4% (19/299) under OpenHands, a statistically significant gain of 4.7 percentage points (95% CI $[2.0, 7.7]$, McNemar test $p = 0.0013$). Practically, this indicates that operational monitors designed to flag anomalous interaction patterns—such as excessive file-read loops—should calibrate their thresholds to the specific harness in deployment.

M.3 Lemma 2 under Shifted Position Indices

When a block of tokens is shifted k positions later in a sequence, every token's absolute position index changes while all pairwise relative distances within the block remain invariant. Lemma 2 predicts a sharp contrast across positional encodings under such shifts: with content vectors held fixed, APE adds a learned embedding for each absolute position index, so translating a sub-trajectory almost surely perturbs attention scores within it (Appendix C.1). By contrast, RoPE rotates queries and keys so that their inner product depends only on relative distance [Su et al., 2024], and ALiBi subtracts a head-specific slope times token distance from each attention score [Press et al., 2022]; under a uniform position shift, RoPE attention scores and ALiBi post-softmax attention weights therefore remain invariant in exact arithmetic (Appendix C.3).

Experiments 4–6 trace the consequences of this difference from internal attention scores to generated answers and, finally, to end-to-end Countdown search accuracy. The first two experiments evaluate 33 public pretrained base models spanning three positional encodings: twelve APE models across GPT-2 (124M–1.5B) [Radford et al., 2019], GPT-Neo (125M–2.7B) [Black et al., 2021], and OPT (125M–6.7B) [Zhang et al., 2022]; five ALiBi models from the BLOOM family (560M–7.1B)

[BigScience Workshop, 2022]; and sixteen RoPE models across Pythia (six models, 160M–6.9B) [Biderman et al., 2023], GPT-J 6B [Wang and Komatsuzaki, 2021], OLMo-2 (1B and 7B) [Team OLMo et al., 2024], SmolLM2 (135M–1.7B) [Ben Allal et al., 2025], and Qwen2.5 (0.5B–7B) [Qwen Team, 2024].

M.3.1 Experiment 4: Attention Scores of Pretrained Models

Inputs and comparison. To test whether shifting position indices alone alters the attention scores of pretrained models, we evaluate four APE models (GPT-2 124M and 355M; GPT-Neo 125M and 1.3B) and four RoPE models (Pythia 160M, 410M, and 1.4B; SmolLM2 360M) on 512 sequences of 512 tokens (128 sequences each from Countdown DFS traces, Countdown BFS traces, Dyck-4 strings nested up to depth 12, and Python 3.10 standard-library source files [Python Software Foundation, 2025]). Each sequence is evaluated five times—twice at its original position indices $0, \ldots, 511$ and once at each shift $k \in \{16, 64, 256\}$—all within each model's context window, yielding 96 model, input-family, and shift combinations.

We record the maximum absolute change in pre-softmax attention scores relative to the maximum absolute unshifted score in the same layer (reporting both the first-layer ratio and the maximum across all layers), alongside the fraction of positions where the greedy next-token prediction changes and the shift in sequence log-likelihood (in nats per token). Every configuration is evaluated in fp32 (with TF32 disabled) and repeated in float64 to isolate floating-point rounding effects.

Attention scores and predictions. Across the 48 APE combinations, the peak first-layer attention-score perturbation reaches 0.42–$1.19\times$ the largest absolute unshifted score, flipping 14.0–87.4% of greedy next-token predictions and reducing sequence log-likelihood by 0.20–6.77 nats per token. Under RoPE, by contrast, the relative score change in fp32 remains below 6.3×10^{-6} in the first layer and 3.2×10^{-3} across all layers—comparable to the numerical difference between fp32 and float64 at the unshifted positions—and drops below 4.5×10^{-12} in float64; in fp32, fewer than 0.039% of RoPE next-token predictions change, all occurring at near-ties between the top two logits. Across all eight pretrained models, internal score sensitivity closely matches Lemma 2, with position shifts frequently altering next-token decisions under APE.

M.3.2 Experiment 5: Answers under Shifted Position Indices

Tasks and scoring. To examine whether these next-token perturbations degrade downstream task accuracy under realistic serving conditions, we evaluate all 33 pretrained models on 500 items across each of six tasks: five few-shot tasks (copying 6–24 words, reversing 2–4 words, closing open brackets in a Dyck-4 string, adding two one- or two-digit integers, and completing a single step of a Countdown search) plus a key–value retrieval task whose prompt scales with the model's context window up to $3{,}000$ tokens. Using greedy decoding with KV caching, we pair each shifted generation with the model's unshifted generation on the same item and score exact-match accuracy.

Position changes. We test shifts $k \in \{16, 64, 256, 1024\}$ wherever the shifted prompt and generation budget fit within the model's pretrained context limit. Four fp32 conditions preserve all pairwise distances among visible tokens: **(i)** adding k to every position index; **(ii)** using masked left padding and increasing all visible-token position indices by k; **(iii)** prefix caching up to the final question while shifting both the cached prefix and the question by $+k$; and **(iv)** shifting position indices within a streaming attention window that retains four attention sinks and the 128 most recent cache slots [Xiao et al., 2024] (compared against the unshifted window on the five few-shot tasks).

In addition, condition **(v)** alters relative distances directly by keeping the first P cached prefix tokens at their original indices while starting the question at $P + k$, increasing every prefix-to-question distance by k without changing the text; and condition **(vi)** repeats the uniform shift of (i) in bfloat16. We evaluate paired accuracy differences across items and flag changes whose paired bootstrap 95% confidence interval excludes zero.

Generated answers. Across the four distance-preserving fp32 conditions (i)–(iv), the 21 RoPE and ALiBi models alter only 40 out of $948{,}000$ generated answers, with net pooled accuracy changing by at most $1/3{,}000$ for every model and condition. At $k = 256$, however, the twelve APE models lose 13–52 percentage points of accuracy in conditions (i)–(iii) and 4–50 points under the sliding window (Figure 5a), with uniform position shifts altering 70–100% of their generated answers. When only the question is shifted in condition (v), altering prefix-to-question distances, accuracy drops

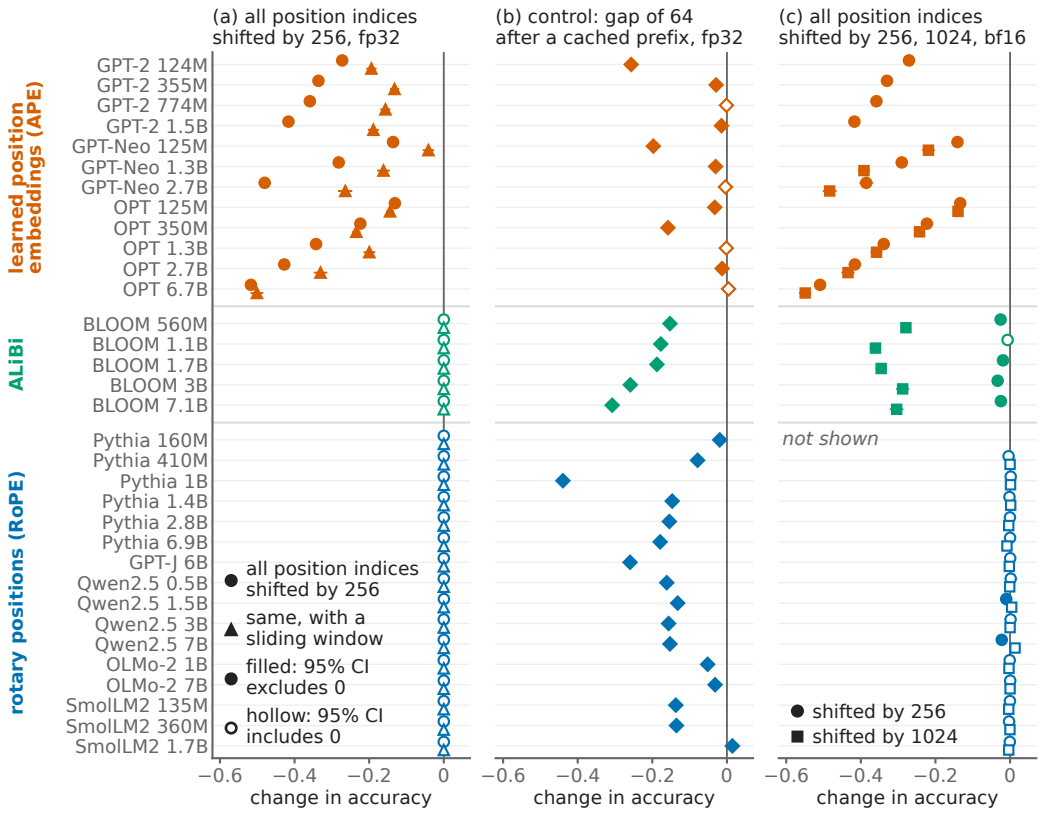


Figure 5: Change in exact-match accuracy after shifting position indices on identical inputs (Experiment 5). Each row corresponds to one of the 33 pretrained models, grouped and colored by positional encoding; the horizontal axis plots shifted minus unshifted accuracy, with error bars (mostly smaller than the markers) showing paired bootstrap 95% confidence intervals and filled markers indicating intervals that exclude zero. **(a)** In fp32, circles apply a uniform $+256$ position shift and triangles apply the same shift inside a sliding window; left padding and shifted prefix caching produce identical changes to circles for APE and alter pooled accuracy by at most $1/3{,}000$ for ALiBi and RoPE. **(b)** In fp32, diamonds keep the cached prefix at its original indices and shift only the question by $+64$. **(c)** In bf16, circles and squares apply uniform shifts of $+256$ and $+1024$, respectively, where context limits permit (Pythia 160M is omitted from panel (c) because its unshifted bf16 accuracy is 0.012).

across all three encodings—by up to 44 percentage points at $k = 64$ (Figure 5b)—with statistically significant changes in 63 of 64 RoPE model–offset pairs and 19 of 20 ALiBi pairs.

Under bf16 in condition (vi) (Figure 5c), APE models suffer similar accuracy losses, whereas RoPE models (excluding Pythia 160M) remain within 2.3 percentage points of their unshifted bf16 accuracy, with significant shifts limited to Qwen2.5 1.5B (at $k = 256$) and Qwen2.5 7B (at $k = 64, 256$). The ALiBi BLOOM models lose 0.7–3.3 points at $k = 256$ and 28–36 points at $k = 1024$: because BLOOM's implementation computes ALiBi biases from the key's absolute position index (which differs from relative distance by a query-dependent constant that cancels in exact arithmetic), bf16 mantissa rounding errors grow with absolute position index at large offsets.

M.3.3 Experiment 6: Two Countdown Models of Section 5

From a search decision to a solved problem. To connect position-index perturbations directly to the Countdown models of Section 5, we evaluate the two models from Appendix J—f_1 (Small Deep MIXED RoPE, $L = 12$) and f_2 (Large Deep BFS APE, $L = 48$)—in fp32 under uniform position shifts of $+16$ and $+64$, testing both teacher-forced routing decisions on reference traces and end-to-end search completion on the models' own rollouts.

Both evaluations anchor at the start of a leaf-level `Exploring Operation` line (e.g., `Exploring Operation: 3+10=13, Resulting Numbers: [13]`) where two operands remain. Specifically, we select the first qualifying line following either a dead end at that node or a backtrack to it while the search remains active and at least one unexplored non-target operation remains available there.

Decisions on reference searches. Under teacher forcing, we feed the reference trajectory prefix and evaluate greedy token predictions on the node identifier following `Moving to Node` and the arithmetic expression following `Exploring Operation` (Appendix K.4). Across the three evaluation sets, 2,863 reference trajectories contain a qualifying line starting between token indices 16 and 880 with at least one routing token before the end of the 896-token window. At both $+16$ and $+64$, the RoPE model (f_1) alters zero of its 94,278 routing predictions, whereas the APE model (f_2) flips 25.7% of its routing predictions at $+16$ and 28.3% at $+64$.

Success on the model's own searches. Across the first $1{,}000$ problems of each evaluation set, we locate the same qualifying line in each model's own unshifted rollout and resume free generation from the start of that line at the original or shifted position indices (requiring at least 64 tokens of remaining generation budget within the 512-token limit), yielding $2{,}128$ eligible searches for f_1 (RoPE) and $1{,}143$ for f_2 (APE). Scored as in Section 5, every eligible RoPE search preserves its exact outcome at both offsets (59.4% solved), whereas APE solve rate drops from 52.7% at the original positions to 12.2% at $+16$ and 13.4% at $+64$—demonstrating that position shifts directly disrupt APE's local routing decisions and end-to-end search success, as formalized in Assumption 2.

M.4 Splitting Deeply Nested Tasks into Separate Model Calls

When a sequential plan opens multiple nested subgoals before closing earlier ones, Lemma 3 maps those dependencies onto nested Dyck-k parentheses, and Lemma 5 establishes a lower bound on single-pass risk whenever the required nesting depth exceeds the circuit threshold $\alpha_{\mathrm{circ}}L$. An agent can circumvent this single-call bottleneck by decomposing deeply nested tasks across separate model invocations so that each call resolves only a bounded number of nesting levels. Experiments 7 and 8 measure the maximum nesting depth solvable in a single call and test whether multi-call decomposition restores accuracy beyond that threshold.

In both experiments, we first measure single-call accuracy at each nesting depth on a separate calibration set, fit a monotonically nonincreasing calibration curve, and extract a single-call capacity threshold: in Experiment 7, d_{ext} denotes the largest depth at which a single call solves at least 50% of plans, whereas in Experiment 8, $\hat{d}$ denotes the largest depth at which a single call correctly evaluates at least 80% of function chains. We then use these calibrated thresholds to bound the nesting depth assigned to each call on the test set (Figure 6).

M.4.1 Experiment 7: Nested Plans

Task and scoring. We train small Transformers from scratch on a synthetic plan-closing task in which `( a` opens a subgoal of type `a` and `) a` closes it. Each plan of depth d contains a spine of d nested subgoals together with $48 - d$ additional subgoals attached uniformly at random to the top level or to subgoals at depth below d; every plan thus contains exactly 48 subgoals drawn from eight types and spans 194 tokens, while its maximum nesting depth d varies. At each closing bracket, the model predicts the matching subgoal type given the prefix, and a plan is scored as solved only if all 48 closing-type predictions are correct.

Splitting a plan. Given a four-level plan `( a ( b ( c ( d ) d ) c ) b ) a` and a per-call depth budget of two levels, our structural splitter presents `( a ( b ) b ) a` to the first call while deferring the interior of `b` to subsequent calls, which receive `( b ( c ) c ) b` and `( c ( d ) d ) c`. The splitter relies purely on parenthesis structure (without access to closing-type targets), and the overall plan is counted as solved only if the model predicts every closing type correctly across all calls.

Models and depth calibration. To test how single-call depth capacity depends on network depth versus training depth, we train GPT-NeoX (RoPE) and GPT-Neo (APE) backbones across five architectures (Appendix K): four of width 256 with $L \in \{2, 4, 8, 12\}$ layers, and a fifth wide architecture with $L = 2$ layers and width 624 (matching the 12-layer model's non-embedding parameter count within 1.3%). For each maximum training depth $d_{\mathrm{train}} \in \{2, 4, 6, 8\}$, training instances are sampled

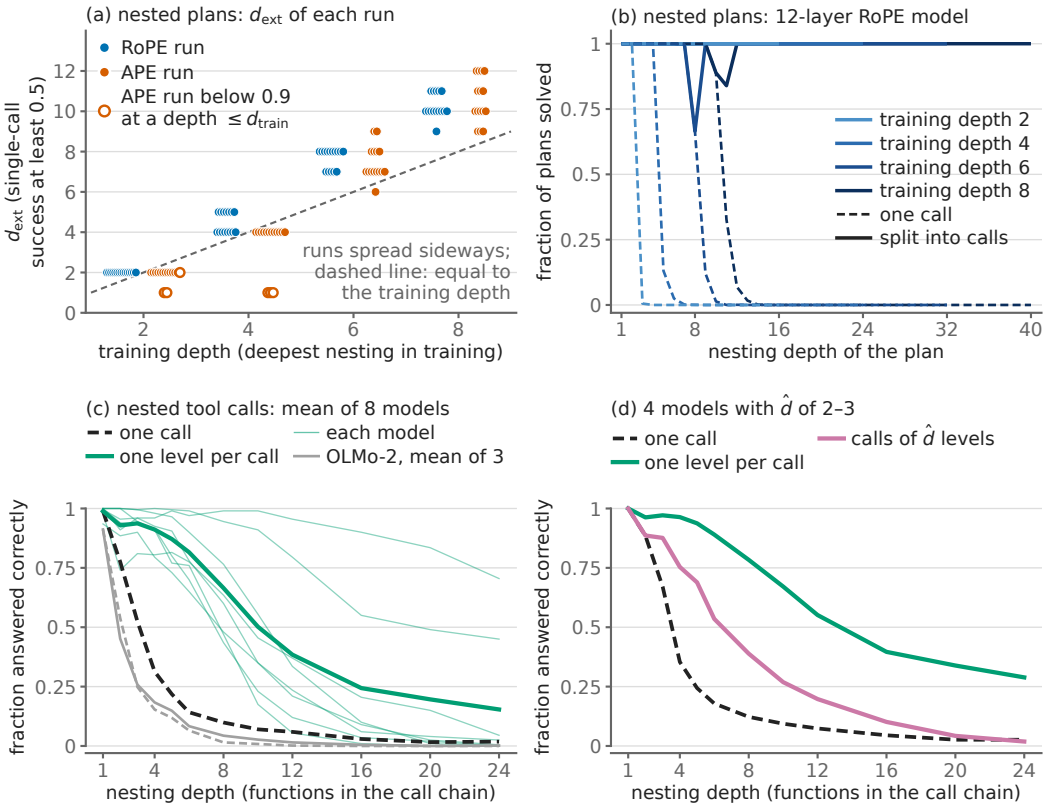


Figure 6: Effect of multi-call decomposition on nested plans **(a, b)** and nested function chains **(c, d)**. **(a)** Calibrated single-call depth limit d_{ext} (the deepest nesting solved at least 50% of the time in one call) versus maximum training depth d_{train} across all 120 training runs, with the dashed line marking $d_{\text{ext}} = d_{\text{train}}$. RoPE runs are plotted to the left of each d_{train} value and APE runs to the right (horizontally jittered to separate ties); hollow markers denote the six runs of the two-layer, width-624 APE backbone at $d_{\text{train}} \in \{2, 4\}$ that failed to reach 90% accuracy across all trained depths. **(b)** Fraction of test plans solved by the 12-layer RoPE backbone (averaged over three seeds, one shade per d_{train}) using a single call across all depths (dashed) versus using a single call up to d_{ext} and splitting deeper plans into calls of depth at most $\min(d_{\text{ext}}, d_{\text{train}})$ (solid); dips between d_{train} and d_{ext} occur where the model still uses a single call. **(c, d)** Exact-match accuracy over 200 test chains per depth. In **(c)**, the dashed black and thick solid green curves show the mean over the eight Qwen, Phi, and Mistral models under single-call evaluation and one-level-per-call splitting, respectively (with thin green curves showing individual models and gray curves showing the three OLMo-2 models). Panel **(d)** averages the four models with $\hat{d} \in \{2, 3\}$, comparing single-call evaluation (dashed), one level per call (green), and $\hat{d}$ levels per call (pink).

uniformly over depths $1, \ldots, d_{\text{train}}$, yielding $2 \times 5 \times 4 = 40$ configurations and 120 total runs across three random seeds.

Each run trains for $30{,}000$ steps with batch size 256 using AdamW (peak learning rate 10^{-3} with cosine decay) and cross-entropy loss on closing-type tokens. In five configurations where at least one seed fell below 90% accuracy at some depth $\leq d_{\text{train}}$, we retrained all three seeds for $60{,}000$ steps, after which three of the five configurations exceeded 90% across all trained depths and seeds.

For each run, we estimate d_{ext} on 200 calibration plans per depth and evaluate on 500 test plans per depth up to $4d_{\text{train}} + 8$. Plans of depth $\leq d_{\text{ext}}$ are evaluated in a single call, whereas deeper plans are recursively decomposed so that each call has nesting depth at most $D = \min(d_{\text{ext}}, d_{\text{train}})$ (since recursive splitting requires at least two levels per call, runs with $D < 2$ use a single call throughout).

Results. In 38 of the 40 configurations (114 runs), all three seeds achieved at least 90% single-call accuracy across every trained depth. Across every one of those 114 runs, mean single-call accuracy at depths beyond d_{ext} was at most 4.02%, whereas multi-call splitting restored mean accuracy to at least 98.7% (with at least 91% accuracy at every individual depth; see Figure 6b for the 12-layer RoPE backbone).

Across these 114 runs, d_{ext} tracked the maximum training depth d_{train} closely, showing no systematic dependence on layer count, width, or positional encoding (Figure 6a): for $d_{\text{train}} \in \{2, 4, 6, 8\}$, calibrated thresholds were $d_{\text{ext}} = 2$, 4–5, 6–9, and 9–12, respectively—always within four levels of d_{train}. Training on deeper plans also reduced the number of calls required under decomposition: beyond d_{ext}, resolving a plan required an average of 20.8 calls at $d_{\text{train}} = 2$ versus 4.5–4.6 calls at $d_{\text{train}} = 8$.

The only two configurations failing the 90% in-distribution threshold were the two-layer, width-624 APE models trained at $d_{\text{train}} \in \{2, 4\}$: five of their six runs reached only $d_{\text{ext}} = 1$ (too shallow to split), while the sixth reached $d_{\text{ext}} = 2$, where splitting raised mean accuracy on deeper plans from 0 to 0.998. Overall, across all models that mastered their training distribution, single-call evaluation generalized only near the maximum depth seen during training, while multi-call decomposition enabled near-perfect generalization to substantially deeper plans by keeping each call within that calibrated horizon.

M.4.2 Experiment 8: Nested Tool Calls with Open-Weight Language Models

Task and scoring. Finally, we test whether off-the-shelf instruction-tuned language models exhibit the same single-call depth bottleneck—and the same recovery under multi-call decomposition—when evaluating chains of nested Python function calls. Each prompt presents a 26-line registry of one-line functions and asks for the integer returned by a nested call such as `bell(4)`:

```
def lamp(x): return x + 2
def door(x): return lamp(x) + 3
def bell(x): return door(x) + 1
```

Under single-call evaluation, the model must compose all three functions in one model call to output 10. Under one-level-per-call decomposition, the first call evaluates the innermost function `lamp(4)`, the next prompt replaces `def lamp(x)` with the returned value to query `door(4)`, and the third call queries `bell(4)` (returning 6, 9, and 10 across the three calls); we score exact match on the final integer.

Each 26-line registry contains a chain of d functions (named after common English nouns, each adding 1, 2, or 3), two distractor functions, and $24 - d$ comment lines to keep the registry at 26 lines across depths, preceded by four worked examples. We generate up to eight tokens per call, extract the first integer produced, and evaluate on 100 calibration items and 200 test items at each depth $d \in \{1, 2, 3, 4, 5, 6, 8, 10, 12, 16, 20, 24\}$.

Models and calls. We evaluate eleven open-weight instruction-tuned models in bf16 using greedy decoding: three sizes each from Qwen2.5 (7B, 14B, 32B) [Qwen Team, 2024], OLMo-2 (7B, 13B, 32B) [Team OLMo et al., 2024], and Qwen3 (8B, 14B, 32B, with thinking disabled) [Qwen Team, 2025], alongside Phi-4 14B [Abdin et al., 2024] and Mistral-Small-24B-Instruct-2501 [Mistral AI, 2025].

We compare four evaluation modes: **(1)** a single call with the standard prompt; **(2)** a single call augmented with the explicit call chain (e.g., `bell(4) -> door(4) -> lamp(4)`); **(3)** d separate calls resolving one function level per call; and **(4)** $\lceil d/\hat{d} \rceil$ calls resolving up to $\hat{d}$ levels per call, where $\hat{d}$ is the largest depth at which the model achieves at least 80% single-call accuracy on the fitted calibration curve.

Results. Under single-call evaluation, the eleven models solve 175–200 of the 200 test chains at $d = 1$, but the best single-call score drops to $103/200$ at $d = 4$ and $31/200$ at $d = 8$; explicitly providing the call chain alters mean single-call accuracy across depths by less than 9.4 percentage points for every model. At the 80% calibration threshold, Qwen2.5 32B reaches $\hat{d} = 3$, Qwen3 14B, Phi-4, and Mistral-Small-24B reach $\hat{d} = 2$, and the remaining seven models reach $\hat{d} = 1$.

Decomposing chains into one level per call restores high accuracy across the eight Qwen, Phi, and Mistral models (Figure 6c): at $d = 8$, their performance rises from 3–31 solved chains (out of 200)

in a single call to 87–198 across eight calls, and at $d = 24$, Mistral-Small-24B still solves $141/200$ chains. By contrast, the three OLMo-2 models benefit far less at $d = 8$ (rising from 0–5 to 7–12 solved chains), as their single-call accuracy already degrades sharply at $d = 2$ (87–144 solved in one call, and 74–119 across two calls).

For the four models with $\hat{d} \in \{2, 3\}$, grouping $\hat{d}$ levels per call reduces the required invocations at $d = 8$ to just three or four calls (Figure 6d), solving 49–102 chains compared with 87–198 under one-level-per-call splitting; because each call feeds its predicted integer into the next prompt so that intermediate errors compound, all four models achieve higher end-to-end accuracy at $d = 8$ when resolving a single level per call.

M.5 Summary of the Eight Experiments

Together, Experiments 1–8 evaluate coordinate selection on Countdown, measure structural domain and harness shifts across agent and robot benchmarks, test positional sensitivity from attention kernels to downstream rollouts, and decompose deeply nested tasks across multiple model calls. Table 3 summarizes the design and key findings of all eight experiments in relation to Theorem 2.

Table 3: Summary of the eight experiments in Appendix M, detailing the experimental setting and primary empirical finding for each study.

Experiment	**Setting**	**Finding**
1. Coordinate expansion on Countdown (M.1.1)	Four nested projections (12, 16, 24, 34 coordinates), models f_1 and f_2, three training distributions	Expanding from 12 to 34 coordinates alters ROC-AUC by < 0.003 while inflating between-measure $\widehat{W}_1$ by 1.8–$2.0\times$; the 12-coordinate projection Φ_{12} provides the best trade-off.
2. Structural shift across agent and robot logs (M.2.1)	52 pairwise group comparisons across ETO, Nebius SWE-rebench, BridgeData V2, and BabyAI	33 of 52 pairs meet the clear separation threshold ($R \geq 3$, $\mathrm{CV} \leq 0.10$), rising to 34 without surface coordinates; tasks sharing subgoal sequences yield the smallest ratios.
3. Fixed model across two coding harnesses (M.2.2)	Qwen3-8B evaluated in mini-SWE-agent and OpenHands on 300 SWE-bench Verified issues	Between-harness $\widehat{W}_1$ reaches $2.19\times$ the larger within-harness baseline (1.95–$11.5\times$ per coordinate), and issue resolve rate rises from 1.7% to 6.4%.
4. Pretrained attention-score shifts (M.3.1)	Four APE and four RoPE pretrained models under uniform position offsets of 16, 64, and 256	Peak first-layer score perturbation under APE reaches 0.42–$1.19\times$ the max absolute unshifted score; across all RoPE layers, relative change stays below 4.5×10^{-12} in float64.
5. Downstream answers under shifted indices (M.3.2)	33 public models tested under uniform index shifts, left padding, prefix caching, and sliding windows	With pairwise distances preserved, 21 ALiBi/RoPE models alter only 40 of $948{,}000$ answers in fp32, whereas APE loses 13–52 accuracy points at $+256$ in conditions (i)–(iii); inserting a position gap degrades all three encodings.
6. Countdown rollouts under shifted indices (M.3.3)	Trained Countdown models f_1 (RoPE) and f_2 (APE) evaluated at offsets $+16$ and $+64$	All RoPE rollout outcomes remain identical (59.4% solved), whereas APE accuracy drops from 52.7% to 12.2% at $+16$ and 13.4% at $+64$ (≈ 40 and 39 percentage points).
7. Multi-call splitting on nested plans (M.4.1)	120 training runs across five shapes, two encodings, three seeds, and $d_{\mathrm{train}} \in \{2, 4, 6, 8\}$	Across the 114 runs that mastered their training depths, multi-call splitting raises mean accuracy beyond d_{ext} from $\leq 4.02\%$ to $\geq 98.7\%$; single-call d_{ext} tracks d_{train}.
8. Multi-call splitting on nested tool calls (M.4.2)	11 open-weight instruction-tuned models (7B–32B) evaluated on 200 function chains per depth	Eight models improve from 3–31 solved depth-8 chains (out of 200) in one call to 87–198 across eight calls; the four models with $\hat{d} \in \{2, 3\}$ solve 49–102 chains when grouping $\hat{d}$ levels per call.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state our theoretical bounds (Lipschitz constants, TC^0 depth limits, and Barron-space width scaling) and empirical scope (54 trained models on Countdown plus eight supplemental experiments in Appendix M), matching the theorems and experimental results throughout the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations and Future Work paragraph of Section 7 discusses the scope of the ϵ-relaxed structural sufficiency assumption, its extension to open-ended natural language reasoning, and next steps for the framework.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical assumptions are stated explicitly: Assumptions 1 and 2 in the main text, and Assumptions 3–5 in Appendix D. Formal, complete proofs are provided in Appendices B, C, D, and E.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We detail the 54 Transformer configurations in Table 1, the data generation process in Section 5, and the full training and evaluation protocols (hyperparameters, prompts, decoding settings, and evaluation sets) in Appendix K, with all source code publicly available (Appendix L). Appendix M details the models, datasets, experimental conditions, and results of the eight supplemental experiments, with full experimental scripts and per-pair, per-model, and per-run outputs provided in the code repository (Appendix L).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and instructions for regenerating the synthetic data and reproducing all experimental results are publicly available on GitHub (Appendix L).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training distributions ($\mu_{\mathrm{BFS}}, \mu_{\mathrm{DFS}}, \mu_{\mathrm{MIXED}}$), model aspect ratios, layer depths, and hidden widths are detailed in Section 5 and Table 1; the optimizer, learning rate schedule, batch sizes, epoch counts, random seeds, and evaluation sets are specified in Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report ± 1 standard deviation error bars to quantify structural variance across architectural aspect ratios under matched core-parameter budgets in Appendix H.2 (Figure 3); because each Countdown configuration is trained once (seed 42), these error bars measure variation across aspect ratios. We also report a one-sided sign test across the 27 matched, independently trained APE/RoPE model pairs (RoPE ahead in 25 of 27 pairs, $p < 10^{-5}$; Section 5), empirical Wasserstein distances $\widehat{W}_1$ as mean $\pm$ s.d. over 10 subsamples (Appendix G), and five-fold cross-validation s.d. (Appendix J). In Appendix M, we report $\widehat{W}_1$ mean and s.d. across 10 seeds with separation criteria ($R \geq 3$, $\mathrm{CV} \leq 0.10$) in Experiment 2, McNemar test p-values and 95% CIs in Experiment 3, paired bootstrap 95% CIs in Experiment 5, and three-seed training runs in Experiment 7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix K reports the hardware (NVIDIA A100 40 GB, one GPU per run), CPU and memory allocations, per-scale training wall-clock times, and total compute budgets: approximately 907 GPU-hours for training and 324 GPU-hours for evaluating the 54 Countdown configurations, plus 17 GPU-hours to label the instances in Appendix J and 57 GPU-hours for the retraining check (while the $\widehat{W}_1$ estimates of Appendix G and classifier analyses of Appendix J ran on CPUs only). For Appendix M, Experiments 1 and 2 ran on CPUs only, while Experiments 3–8 used approximately 23.5, 0.45, 106, 4.5, 119, and 13 GPU-hours, respectively, across NVIDIA A100, H100, and L40S GPUs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This is a theoretical and empirical machine learning study that fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Broader Impact paragraph of Section 7 discusses both positive impacts (guiding parameter allocation, positional-encoding choices, and multi-call decomposition for reliable and efficient reasoning systems) and potential negative impacts (such as over-relying on theoretical bounds evaluated under stated assumptions in place of domain-specific safety testing, or indirect dual-use risks associated with more capable reasoning models).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper studies theoretical capacity bounds and evaluates them on a formal combinatorial game (Countdown); the released research code poses no high risk of misuse or dual use.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We credit all assets: the Stream of Search code used for Countdown data generation and scoring [Gandhi et al., 2024] (Apache-2.0 license), Hugging Face Transformers, Accelerate and Datasets (Apache-2.0), PyTorch (BSD-3-Clause), scikit-learn [Pedregosa et al., 2011] (BSD-3-Clause), the POT library [Flamary et al., 2021] (MIT), and the pretrained GPT-Neo tokenizer (EleutherAI/gpt-neo-1.3B, MIT); for Appendix M, the pretrained GPT-2 models [Radford et al., 2019] (modified MIT license) and GPT-Neo models [Black et al., 2021] (MIT), the Pythia and SmolLM2 models [Biderman et al., 2023, Ben Allal et al., 2025] (Apache-2.0), the Python standard library (PSF License 2.0), the ETO trajectories [Song et al., 2024b] (Apache-2.0) of ALFWorld (MIT), ScienceWorld (Apache-2.0) and WebShop (MIT), the SWE-agent and OpenHands trajectory datasets of Nebius [Nebius, 2024, Trofimova et al., 2025] (CC-BY-4.0; the SWE-agent dataset card also refers its users to the Llama 3.1 license), the ECoT features [Embodied-CoT, 2024] (MIT) of BridgeData V2 [Walke et al., 2023] (CC-BY-4.0), Minigrid [Chevalier-Boisvert et al., 2023] (Apache-2.0) with the BabyAI bot [Chevalier-Boisvert et al., 2019] (BSD-3-Clause), the OPT models [Zhang et al., 2022] (OPT-175B license agreement, non-commercial research use), the BLOOM models [BigScience Workshop, 2022] (BigScience BLOOM RAIL License v1.0), GPT-J [Wang and Komatsuzaki, 2021] (Apache-2.0), the Qwen2.5 models [Qwen Team, 2024] (Apache-2.0; Qwen2.5 3B under the Qwen Research License), the OLMo-2 and Qwen3 models [Team OLMo et al., 2024, Qwen Team, 2025] (Apache-2.0), Phi-4 [Abdin et al., 2024] (MIT), Mistral-Small-24B-Instruct-2501 [Mistral AI, 2025] (Apache-2.0), vLLM [Kwon et al., 2023] (Apache-2.0), mini-SWE-agent and OpenHands [Yang et al., 2024, Wang et al., 2025] (MIT), and SWE-bench Verified [Jimenez et al., 2024, OpenAI, 2024b] (SWE-bench code and harness MIT); see Section 5 and Appendices J, K and M.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code, including the scripts that generate the synthetic data, is available on GitHub with documentation (Appendix L).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research consists of mathematical proofs and computational experiments, and does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable, as no human participants were involved in this study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are the subject of our theoretical and empirical analysis; no LLM is an important, original, or non-standard component of our core methods.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.