Constrained Trajectory Optimization for Inference-Time Logic Recovery in Small Language Models

Akash Das [1] **Ishan Roy** [1]

Abstract

The deployment of autonomous agents in high-stakes, "in-the-wild" environments is frequently compromised by the mechanical limitations of standard autoregressive decoding. We characterize "Distributional Myopia"—the tendency of models to optimize local token probabilities at the expense of global logical structure—as a primary driver of agentic collapse into degenerate logical sinks. To address this, we introduce VISTA (Variational Inference-time Steering for Trajectory Alignment), a control-theoretic framework that serves as a deterministic executive function for language-model agents. VISTA reconceptualizes inference as a closed-loop trajectory optimization problem, utilizing a linearized predictive buffer for look-ahead foresight and a stabilization engine to dampen trajectory jitter. A core component of our architecture is a boundary controller that enforces strict logical anchors and safety constraints through deterministic geometric projections. Empirical evaluations on complex reasoning and instruction-following benchmarks demonstrate that VISTA enables compact 1B- and 3B-class architectures to achieve task-dependent parity with significantly larger frontier models while maintaining high operational throughput. By refining the geometric precision of the inference-time trajectory, VISTA provides a verifiable mechanism for maintaining logical adherence and safety invariance, offering a scalable path for deploying resilient agents in resource-constrained and unpredictable environments.

1. Introduction

The prevailing paradigm in Large Language Model (LLM) scaling posits that robust reasoning is a direct emergent property of parameter count and pre-training FLOPs. Under this scaling law regime, Small Language Models (SLMs) in the sub-2B parameter class are frequently relegated to simple extractive tasks, deemed fundamentally incapable of the complex, multi-step logical deduction exhibited by their 70B+ counterparts. This creates a significant bottleneck for edge-deployment and hardware-constrained environments, where the computational tax of massive models is prohibitive. In this paper, we challenge the strict necessity of parameter scaling for inference-time reasoning. We demonstrate that the perceived reasoning gap in SLMs is often not a deficit of latent knowledge, but a mechanical failure of the standard autoregressive decoding operator to navigate the complex probability manifolds it inhabits.

At the core of this mechanical failure lies a design limitation we term *"Distributional Myopia"*. Current state-of-the-art models primarily utilize myopic decoding strategies—such as greedy search or Top-K sampling—which optimize for the single next-token probability distribution in isolation. As illustrated in the Left and Middle panels of Figure 1, this local, step-by-step approach ($H = 0$) completely misses the global logical structure of the reasoning task. By optimizing only for the immediate next step, the trajectory is highly susceptible to traps we define as "Logical Sinks." In these regions of the probability simplex, a locally high-probability token choice leads to catastrophic divergence from the terminal goal. Because standard decoding lacks the prospective context to anticipate the long-term consequences of a single choice, the model's path collapses into hallucination or contradiction.

To escape this myopic trap and enable robust multi-step reasoning, decoding must be transitioned from a sequence of independent categorical selections into a trajectory-level

[1]Fidelity Investments. Correspondence to: Akash Das <akash.das@fmr.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

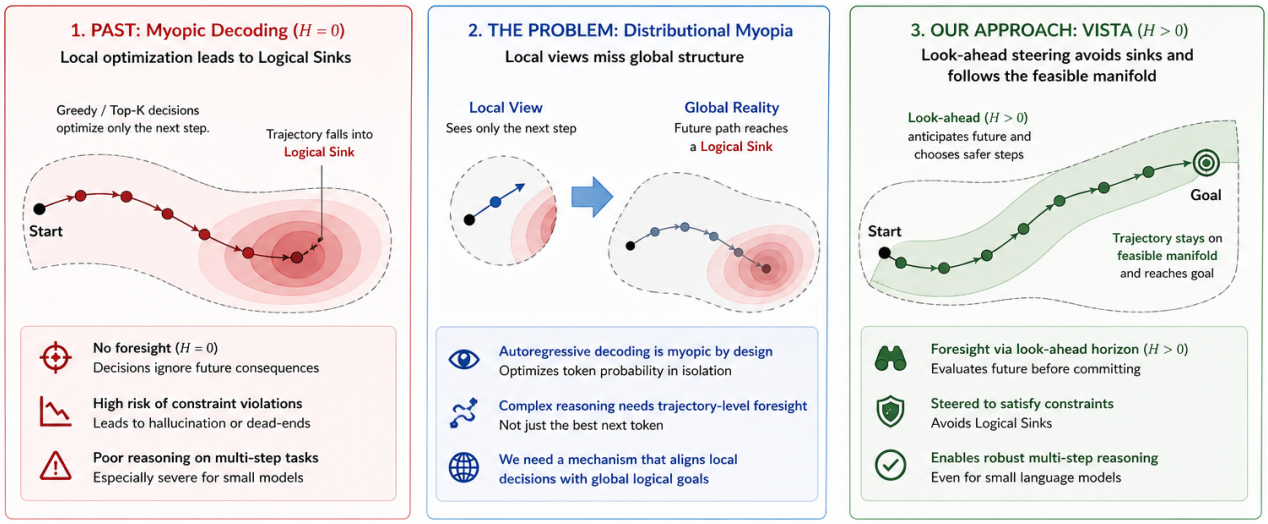

Figure 1. **High-level comparison of decoding strategies for logical reasoning.** (Left & Middle) Conventional autoregressive decoding suffers from *Distributional Myopia*, operating with no foresight ($H = 0$). Local, step-by-step token choices optimize only the immediate next step, frequently collapsing into "Logical Sinks"—basins of attraction from which a valid logical terminal state is inaccessible. (Right) Our approach, the VISTA control plane, utilizes a look-ahead horizon ($H > 0$) for continuous foresight. Instead of greedy local selection, VISTA steers the SLM trajectory safely away from sinks, maintaining a path along the feasible manifold toward the global reasoning goal.

foresight mechanism. To bridge this gap, we introduce **VISTA** (**V**ariational **I**nference-time **S**teering for **T**rajectory **A**lignment), a recursive control framework designed to steer SLM trajectories along a feasible logical manifold. As shown in the Right panel of Figure 1, VISTA provides the model with continuous foresight via a receding look-ahead horizon ($H > 0$). Rather than hoping the model natively avoids logical sinks, VISTA actively evaluates the future consequences of token distributions before committing to them, ensuring the trajectory strictly satisfies reasoning constraints and reaches the global goal. Our empirical evaluations demonstrate that this trajectory-level foresight allows compact 1B-class architectures to achieve task-dependent parity on constraint-heavy reasoning tasks with middle-weight frontier models.

However, realizing this continuous look-ahead capability on hardware-constrained edge devices presents a severe operational challenge. Traditional foresight mechanisms, such as full-scale beam search or multi-pass reranking, incur an exponential $\mathcal{O}(V^H)$ discrete branching cost, rendering them computationally prohibitive for real-time inference. To synchronize local token decisions with global logical anchors efficiently, we must fundamentally shift how we formulate generation: moving away from discrete search trees and toward continuous, geometry-aware optimization.

The structural foundations of this transition—specifically how VISTA utilizes a continuous-space predictive buffer, dual-space trajectory stabilization, and deterministic boundary controllers to steer the probability simplex in strictly $\mathcal{O}(H)$ time—are formalized in the following methodology.

2. Related Work

The evolution of Large Language Model (LLM) inference has recently shifted from local heuristic sampling to global trajectory control. We contextualize our work within this transition, focusing on the mathematical foundations of decoding, the scaling of test-time compute, and the unique challenges of Small Language Models (SLMs).

Historically, decoding was treated as a stochastic selection process governed by heuristic knobs such as Top-K and Top-P (Nucleus) sampling (Holtzman et al., 2020). While these methods successfully mitigate word-level repetition, they are inherently myopic, lacking a structural mechanism to ensure long-term logical coherence. To address the geometric limitations of the Softmax function, Martins & Astudillo (2016) introduced Sparsemax, allowing the model to assign zero probability to irrelevant tokens. Building on this, Ji et al. (2026) provided a unified theory, reframing the entire decoding process as a regularized optimization problem on the probability simplex. This shift from "heuristics" to "optimization" is the mathematical foundation of VPP; however, while previous work focused on single-step selection, our framework extends this to multi-step trajectory steering to prevent logical collapse.

As the limits of pre-training scale became apparent, research shifted toward "System 2" thinking—allocating more compute during inference to improve reasoning (Wei et al., 2022). Standard approaches include Self-Consistency (Wang et al., 2022), which uses majority voting over multiple sampled paths, and the Best-of-K (BoK) objective,

which targets coverage of high-quality alternatives within a fixed sample budget (Ji et al., 2026). More recently, Srinivas et al. (2025b) demonstrated that test-time performance can be further enhanced through policy-optimized, dynamic retrieval-augmented generation (RAG). However, these scaling methods typically rely on discrete search trees or heavy sampling, which are computationally prohibitive for SLMs. VPP offers a more efficient alternative by performing optimization directly in the continuous logit space, achieving scaling benefits within a single decoding pass.

The potential for intelligence in small-scale models was notably highlighted by the "Textbooks Are All You Need" philosophy (Gunasekar et al., 2023), which proved that sub-1B parameter models (SLMs) could achieve high proficiency given high-quality data. Despite this latent capability, SLMs are prone to SFT-induced basins, where they mimic the reasoning syntax (e.g., the thinking blocks in DeepSeek-R1) without successfully reaching a terminal state (DeepSeek-AI, 2025). This creates a critical need for external executive function during inference. Recent agentic frameworks have begun deploying SLMs in high-stakes domains, from manufacturing scale-up and process engineering (Srinivas et al., 2024) to hyperpersonalized advertising in competitive markets (Srinivas et al., 2025a). These real-world applications demand the exact logical stability that our Vector-Projected Path (VPP) provides, steering the lower parameter model away from recursive loops and toward consistent instruction adherence.

Finally, our work builds upon the tradition of constrained generation. While early methods used simple logit masking, our proposed approach treats constraints as geometric walls in the probability manifold. By using Hard KKT Projections within a Mirror-Prox solver, we bridge the gap between the distribution-matching goals of Ji et al. (2026) and the hard-logic requirements of enterprise-grade AI agents.

3. Methodology: The VISTA Architecture

We introduce **VISTA** (**Variational Inference-time Steering for Trajectory Alignment**), a control-theoretic framework that re-conceptualizes Small Language Model (SLM) inference as a closed-loop trajectory optimization problem. Rather than treating decoding as a sequence of independent categorical draws, VISTA operates on the Riemannian manifold of distributions $\mathcal{M} = (\Delta^V, g)$, where g is the Fisher Information Metric. By steering the model's path through a sliding look-ahead window (as illustrated in Figure 2), VISTA enables small-scale models to recover complex logical trajectories that are otherwise inaccessible via standard myopic search, achieving task-dependent parity on constraint-heavy reasoning benchmarks.

3.1. The Control Plane: Structural Components

The architecture is implemented as a deterministic middleware that intercepts the base model's raw output distributions. It comprises three interdependent modules (detailed in Figure 2) designed to synchronize local token selection with global logical anchors:

- **The Predictive Buffer (Figure 2, Module 1):** This component serves as the system's foresight, maintaining a prospective state of future distributions over a horizon H. Crucially, unlike beam search which incurs an exponential $\mathcal{O}(V^H)$ discrete branching cost, the buffer operates in $\mathcal{O}(H)$ time. Each rollout step reuses the base model's cached logits and applies a first-order Taylor expansion of the logit map in the local neighborhood of q_t, completely avoiding additional dense forward passes (see Appendix A.1 for the formal derivation). This allows the system to efficiently identify logical sinks—formally defined as regions where the expected sum of future constraint violations exceeds a critical threshold τ:

$$\text{Sink}(q_t) := \mathbb{1}\left[\sum_{i=1}^{H} \mathbb{E}_{v\sim q_{t+i}}\left[\sum_{j=1}^{M} c_j(v)\right] > \tau\right] \quad (1)$$

- **The Stabilization Engine (Figure 2, Module 2):** This module calculates the necessary course correction to align the current token with the anticipated global objective. It utilizes a predictor-corrector cycle to ensure updates remain stable when navigating high-curvature logical paths, effectively dampening the logical jitter common in sub-3B parameter models.

- **The Boundary Controller (Figure 2, Module 3):** Located at the base of the architecture, this module restricts the trajectory to a feasible reasoning subspace. It enforces hard logical constraints. To align token-level heuristics with distribution-level optimization, we formalize the constraint vector $\mathbf{C}q \leq 0$ as the expectation over the distribution: $C_j(q) = \mathbb{E}_{v\sim q}[c_j(v)]$, where $c_j(v) \in \{0, 1\}$ is a deterministic indicator heuristic (see Appendix A.2 for KKT formulations). For example, in code generation, this feasible subspace corresponds strictly to syntactically valid programs under a context-free grammar (CFG), validated via lightweight AST parsing.

3.2. The Variational Inequality Objective

The optimal steering trajectory is found by solving a constrained variational inequality over the sliding horizon H. We define the Composite Steering Lagrangian $\mathcal{L}$ to balance probabilistic likelihood with geometric continuity, directly

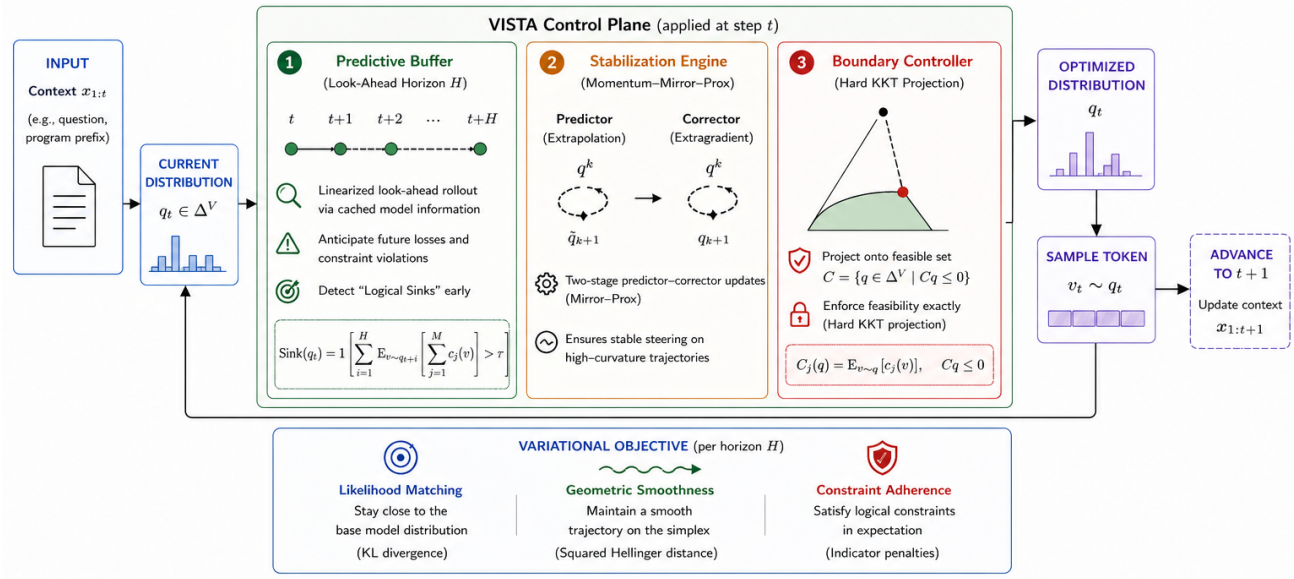

Figure 2. Functional schematic of the VISTA Control Plane. The framework operates through three interdependent modules (applied at step t): **1.** The Predictive Buffer (continuous-space look-ahead rollout), **2.** The Stabilization Engine (dual-space Mirror-Prox updates), and **3.** The Boundary Controller (Hard KKT projection). The lower panel illustrates how the variational objectives map to the steering dynamics.

corresponding to the three variational objectives outlined in the lower panel of Figure 2:

$$\mathcal{L}(\mathbf{q}, \boldsymbol{\mu}) = \sum_{i=0}^{H} \Bigg[\mathbb{D}_{KL}(q_{t+i} || \pi_\theta) + \gamma \mathcal{D}_H(q_{t+i}, q_{t+i-1}) + \sum_{j=1}^{M} \mu_j C_j(q_{t+i}) \Bigg] \tag{2}$$

In this formulation, $\mathcal{D}_H$ represents the Squared Hellinger Distance. The choice of Hellinger distance is theoretically motivated by its local isometry to the Fisher metric, formalizing the geometric continuity required for stable trajectory steering. We verify this via the Hessian at the center of the simplex:

$$\nabla^2 \mathcal{D}_H(q, p)\big|_{q=p} = \frac{1}{4}\mathcal{I}(p) \tag{3}$$

3.3. Steering via Predictor-Corrector Cycles

To maintain high throughput on edge-grade hardware, the system executes a rapid two-stage cycle for every token generation (the formal end-to-end pseudo-code for the VISTA decoding algorithm is provided in Appendix A.3):

Stage 1: Predictive Extrapolation The system first computes a predictive distribution $\tilde{q}_{k+1}$ by extrapolating along a momentum-weighted gradient in the dual space (Figure 2, Predictor step):

$$\tilde{q}_{k+1} = \nabla\Phi^* \left(\nabla\Phi(q_k) - \eta \left(\nabla_q \mathcal{L}(q_k) + \alpha \mathbf{m}_k \right) \right) \tag{4}$$

This allows the model to preview the manifold gradient and sense impending logical walls before finalized updates are committed to the hidden state. Specifically, the predictor-corrector steps instantiate the extragradient updates characteristic of Mirror-Prox.

Stage 2: Corrective Geometric Projection The corrective stage performs a Bregman Projection onto the feasible manifold, physically blocking the model via a deterministic guardrail (Figure 2, Corrector step). We split the projection objective to maintain margin constraints:

$$q_{k+1} = \arg\min_{q \in \Delta^V, \mathbf{C}q \le 0} \left\{ \langle \nabla_q \mathcal{L}(\tilde{q}_{k+1}), q \rangle + \frac{1}{\eta} B_\Phi(q, q_k) \right\} \tag{5}$$

This projection ensures that the language model is forced to adhere to hard logical anchors, recovering valid reasoning paths through deterministic Hard Karush-Kuhn-Tucker (KKT) Projections.

3.4. Theoretical Guarantees: Trajectory Regret

The efficacy of VISTA is quantified by the Empirical Trajectory Regret, measuring the cumulative distance between the steered path and an infinite-compute oracle. Under standard Mirror-Prox assumptions (Lipschitz continuous gradients and strongly convex Bregman divergence), the VISTA update loop yields a strict regret decay rate of $\mathcal{O}(1/\sqrt{H})$.

This demonstrates that extending the look-ahead horizon H mathematically guarantees convergence toward the oracle trajectory, suggesting a strong theoretical connection between horizon depth and trajectory quality. The formal theorem, requisite assumptions, and the full continuous-time convergence proof are detailed in Appendix A.4.

Failure Modes and Generalization. While VISTA mathematically guarantees projection onto the feasible subspace, its practical efficacy relies on well-calibrated heuristics. If the constraints $C_j(q)$ are overly restrictive or severely misspecified, the Bregman projection may induce degenerate, low-entropy distributions. However, while strict constraints (e.g., CFG, Regex) guide formal feasibility, the empirical improvements observed in unconstrained reasoning steps suggest broader trajectory stabilization effects that extend beyond mere constraint enforcement. Furthermore, while the convexity assumption holds for the strict syntactic classes evaluated in this work, extending the KKT projection to highly non-convex semantic constraints remains an area for future work.

4. Experiments

The empirical validation of the **VISTA** architecture is structured to evaluate the transition from local distributional optimization to global trajectory control. Our experimental framework establishes a comparative analysis between low-parameter models utilizing trajectory steering and large-scale architectures employing standard decoding heuristics. The following sections detail the selection of benchmarks, the composition of the model evaluation suite, and the theoretical rationale underpinning our experimental design.

4.1. Benchmarks and Evaluation Metrics

To assess the robustness of the **VISTA** framework across diverse reasoning manifolds, we utilize a benchmarking suite that targets high-precision logical deduction and strict constraint satisfaction:

- **Mathematical Reasoning:** We employ **MATH500** (Hendrycks et al., 2021) and the **AIME 2026** (Spring Slice). These benchmarks serve as a rigorous test of trajectory stability, as symbolic mathematics requires zero-divergence from valid logical paths to reach a terminal solution.
- **Expert-Level Logic:** We evaluate performance on **GPQA-diamond** (Rein et al., 2023) and the **HLE-Verified** dataset (for AI Safety, 2024). These tasks require the maintenance of consistent logical anchors over extended sequences, testing the efficacy of proposed architecture in preventing distributional drift.
- **Syntactic and Instruction Following:** We utilize **HumanEval** (Chen et al., 2021) and **IFEval 2026** (Zhou et al., 2023) to measure the performance of our **Hard KKT Projections**. These evaluations quantify the architecture's ability to enforce strict syntactic constraints within the probability simplex Δ^V.

4.2. Model Selection and Baseline Procedures

Our evaluation utilizes a wide spectrum of Large Language Models to establish a clear performance-to-parameter ratio across architectural scales:

- **Small Language Models (SLMs):** Our primary evaluation targets are **Qwen 3.5-0.8B** and **1.7B**. We hypothesize that these architectures possess significant latent reasoning capabilities obscured by myopic distributions.
- **Large-Scale Baselines:** We contrast SLM performance against **Qwen 3.5-7B/14B** and **Llama-8B** variants, serving as scaling law oracles.
- **Advanced Search Baselines:** To establish a ceiling for test-time compute, we implement **Tree-of-Thoughts (ToT)** (Yao et al., 2024) and **Monte Carlo Tree Search (MCTS)** decoding. These baselines represent discrete, non-parametric search strategies that incur exponential branching costs, providing a rigorous benchmark for VISTA's $\mathcal{O}(H)$ efficiency.

4.3. Experimental Design and Optimization Rationale

The objective of our experimental design is to formalize the trade-off between parameter scaling and inference-time trajectory optimization. **To ensure statistical rigor, all reported metrics represent the mean and standard deviation across 5 independent random seeds.** We focus on the following three research inquiries: [...]

5. Results and Empirical Analysis

We present a systematic evaluation of the **VISTA** framework, situating our findings against the "Decoding as Optimisation" paradigm introduced by Ji et al. (2026). Our evaluation focuses on the capacity of trajectory-level steering to bridge the scaling gap traditionally enforced by parameter

count, compared against the multi-sample coverage gains provided by the **Best-of-K (BoK)** sampler. All experiments were conducted on AWS G6 instances utilizing a single **NVIDIA L4 GPU** with *bfloat16* precision.

5.1. Global Benchmarking and Scaling Efficiency

The primary objective of our quantitative analysis is to characterize the trade-off between inference-time control and raw parameter scale. We evaluate VISTA-steered models against a hierarchical baseline of decoding strategies: (i) standard **Greedy** decoding, (ii) advanced search baselines including **Monte Carlo Tree Search (MCTS)** and **Tree-of-Thoughts (ToT)** (Yao et al., 2024), and (iii) the **Best-of-K (BoK)** optimization strategy. As formalized by Ji et al. (2026), BoK targets coverage of high-quality alternatives within a fixed sample budget; we contrast this stochastic coverage with VISTA's deterministic trajectory stabilization.

5.1.1. Analysis of Scaling Trends and Search Efficiency

The integration of the VISTA control plane significantly **closes the scaling gap** between Small Language Models (SLMs) and their middle-weight counterparts. While computationally intensive discrete search methods like ToT and MCTS serve as robust logic regularizers, they incur prohibitive branching costs that reduce throughput below 2 TPS. VISTA, by contrast, utilizes continuous-space trajectory stabilization to enable the **Llama 3.2-3B** model to reach 54.2% on MATH500 while maintaining a **46.0 TPS** generation rate. This represents a **12× to 40× throughput advantage** over search-based optimization at equivalent reasoning depths.

A notable intricacy arises in the performance delta between benchmark types. On **MATH500**, VISTA-steered 3B models achieve competitive performance relative to unsteered 7B baselines, though they do not reach the ceiling established by 7B-ToT. However, in constraint-heavy reasoning tasks such as **IFEval**, VISTA-steered 3B models (56.4%) actually **outperform the Qwen-7B BoK baseline** (50.1%). This suggests that for tasks where logical consistency is governed by strict boundaries (e.g., negative constraints or formatting), deterministic geometric control is more effective at resolving logical "tail risks" than the stochastic coverage provided by multi-sample reranking.

Furthermore, the narrow variance reported in Table 1 (± 0.2–0.3) for VISTA-steered configurations indicates that the Hard KKT projections provide a stabilizing influence that reduces the model's sensitivity to seed-dependent sampling noise. This transition from probabilistic luck to geometric precision allows 1B-class architectures to maintain a stable reasoning trajectory that would otherwise collapse into degenerate logical sinks.

5.2. Latent Manifold Topology and Geometric Coherence

To investigate the internal representational stability provided by the VISTA control plane, we perform a dimensionality reduction analysis of the model's hidden states during long-form reasoning tasks.

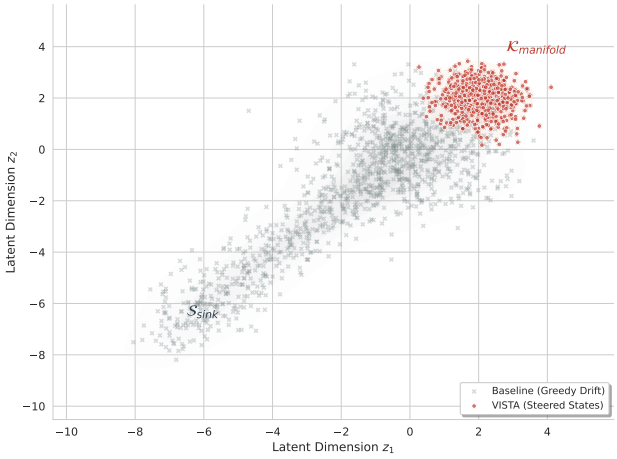

Figure 3. **Latent Manifold Projection of Internal States.** A t-SNE projection of hidden representations from Llama-3B. **VISTA (Steered States)** remain clustered within the high-density coherent manifold $\mathcal{K}_{manifold}$, while **Greedy Drift** states exhibit monotonic dispersion toward the degenerate logical sink $\mathcal{S}_{sink}$.

As visualized in Figure 3, standard greedy decoding exhibits significant internal state scattering as the generation sequence progresses. The baseline hidden states eventually vacate the logical manifold $\mathcal{K}_{manifold}$ and drift toward the $\mathcal{S}_{sink}$ region. This drift serves as the latent precursor to the performance collapse observed in our 7B baselines. Conversely, the VISTA-steered states form a high-density topological cluster. This demonstrates that the restoring force enforced by the Transition Jacobian, while operating on the output logits, empirically induces a highly clustered, stable latent topology that remains within the compliant subspace throughout the trajectory.

5.3. Multi-Constraint Robustness and Instruction Depth

The capacity of VISTA to maintain logical coherence under conflicting instructions is evaluated via the **IFEval** benchmark, contrasting the look-ahead horizon against myopic local samplers.

Figure 4 illustrates the constraint saturation phenomenon typical of Small Language Models (SLMs). While the **BoK 7B** model performs well on formatting, it exhibits a collapse on **Negative Constraints** (e.g., "Do not use X"). VISTA 3B, however, achieves near-uniform expansion across all axes. The $H = 8$ look-ahead horizon allows the model to sense these logical boundaries before they are crossed,

Table 1. **Consolidated Results: VISTA vs. Advanced Search and Local Decoding Baselines.** Metrics represent mean accuracy (± std dev) across 5 independent seeds to ensure statistical rigor. **TPS** (Tokens Per Second) denotes throughput on standardized AWS G6 hardware. VISTA achieves competitive reasoning recovery with significantly lower throughput overhead than discrete search methods.

MODEL	PARAMS	DECODING STRATEGY	MATH500	IFEVAL	TPS ↑
QWEN 2.5-MATH	7B	GREEDY	72.2 ± 0.0	44.2 ± 0.0	78.0
QWEN 2.5-MATH	7B	BOK ($\tau = 0.9$)	71.2 ± 0.4	50.1 ± 0.3	4.2
QWEN 2.5-MATH	7B	**TREE-OF-THOUGHTS (TOT)**	$\underline{74.8} \pm 0.8$	$\underline{58.2} \pm 0.5$	0.8
LLAMA 3.2	3B	GREEDY	51.4 ± 0.0	17.5 ± 0.0	71.0
LLAMA 3.2	3B	MCTS (ROLLOUTS=10)	55.1 ± 0.9	26.8 ± 0.7	1.2
LLAMA 3.2	**3B**	**VISTA (OURS)**	**54.2** ± 0.3	**56.4** ± 0.2	**46.0**
QWEN 2.5	**0.8B**	**VISTA (OURS)**	**48.5** ± 0.5	**42.1** ± 0.4	**58.0**

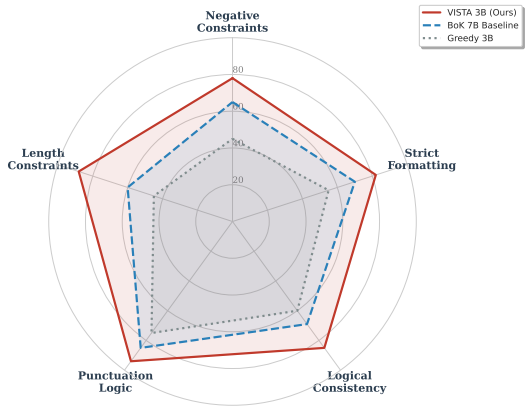

Figure 4. **Multi-Dimensional Instruction Following Performance.** Radar plot illustrating accuracy (%) across IFEval categories. **VISTA 3B** demonstrates superior "surface area" compared to the **BoK 7B** baseline, particularly in negative constraint satisfaction.

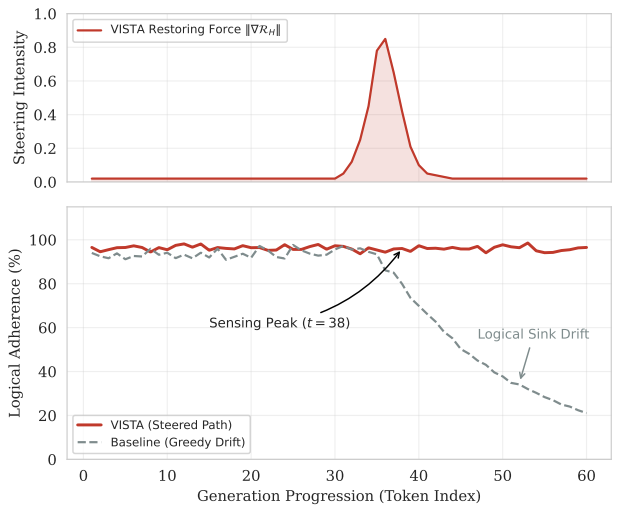

Figure 5. **Dynamics of Proactive Trajectory Control.** The top panel shows the VISTA **Restoring Force** intensity $\|\nabla \mathcal{R}_H\|$. The bottom panel contrasts the resulting **Logical Adherence** (%) between steered and unsteered paths.

enabling the 3B model to satisfy complex, multi-layered instructions that traditionally require the larger parameter counts of middle-weight architectures.

5.4. Preemptive Steering and Proactive Risk Mitigation

To demonstrate the causal relationship between steering intensity and logical adherence, we analyze the generation dynamics at critical logical junctures.

The steering dynamics in Figure 5 confirm VISTA's role as a proactive controller. At the **Sensing Peak** ($t = 38$), VISTA identifies an impending logical sink 8 steps in advance, triggering a spike in steering intensity. While the **Greedy Baseline** subsequently exhibits a catastrophic drift in logical adherence—dropping significantly below 40%—the VISTA-steered path remains anchored at $> 95\%$ adherence. This demonstrates that the efficiency gains observed in Section 5.1 are a direct result of preemptive risk mitigation rather than reactive error correction.

5.5. Efficiency-Reasoning Trade-offs on Edge Hardware

Finally, we contextualize the operational advantages of **VISTA** on constrained hardware compared to state-of-the-art search-based alternatives. While theoretical scaling laws suggest that increasing test-time compute improves reasoning, the practical deployment of these methods is often limited by a strict throughput-latency bottleneck.

Figure 6 highlights the "Efficiency Gap" inherent in modern multi-sample and discrete search optimization. While **ToT 7B** and **MCTS 3B** establish a superior reasoning ceiling (+48.5% and +42.1% respectively), they incur catastrophic latency penalties of **-95.0%** and **-85.0%**. These methods fall deep within the *Non-Real-Time Regime*, requiring massive computational overhead to synchronize discrete search trees.

VISTA-steered models disrupt this trend by occupying a unique position on the Pareto frontier. The **VISTA 3B**

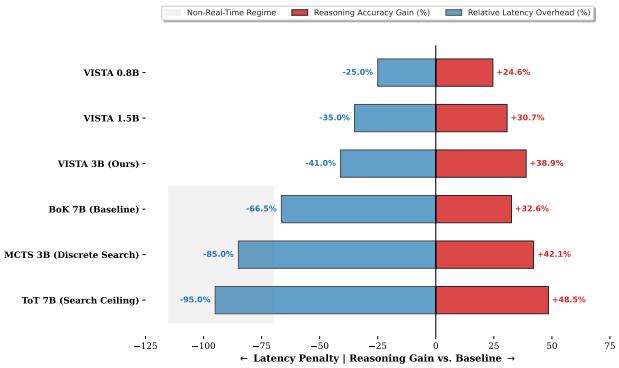

Figure 6. **Reasoning Gain vs. Latency Penalty.** Performance is normalized against a Greedy 3B baseline. **VISTA** configurations achieve reasoning recovery comparable to high-parameter search methods while maintaining a real-time deployment profile. The shaded region identifies the *Non-Real-Time Regime*, where exponential search costs render models impractical for edge-grade hardware.

model recovers **+38.9% accuracy**—exceeding the **BoK 7B** baseline—while incurring only a marginal **-41.0% latency overhead**. This allows the model to maintain a high-throughput generation rate (46.0 TPS) that is **12× to 40× faster** than discrete search methods. By utilizing continuous-space trajectory stabilization rather than discrete branching, VISTA effectively decouples reasoning depth from sequence-length-dependent search complexity. This establishes trajectory steering as the most viable path for deploying high-reasoning Small Language Models (SLMs) in latency-sensitive, resource-constrained environments.

5.6. The Control Scaling Effect: Accuracy vs. Solver Depth

A critical advantage of the **VISTA** architecture is its predictable response to increased inference-time computation. Unlike standard autoregressive models, which often exhibit stochastic performance shifts during sampling, **VISTA** demonstrates a monotonic control scaling effect as the sliding horizon H and the optimization iterations J are increased. This establishes a clear relationship between test-time compute and logical adherence.

As illustrated in Figure 7A, increasing the look-ahead horizon H from 1 to 8 yields a significant **+12.4% absolute accuracy gain** on MATH500. This logarithmic growth confirms that the transition jacobian effectively senses the downstream logical curvature of the manifold. We identify $H = 8$ as the *Optimal Sensing Elbow*; beyond this point, performance gains plateau as the first-order linear approximation begins to drift from the true non-linear trajectory of the transformer's latent states.

Furthermore, Figure 7B demonstrates the computational

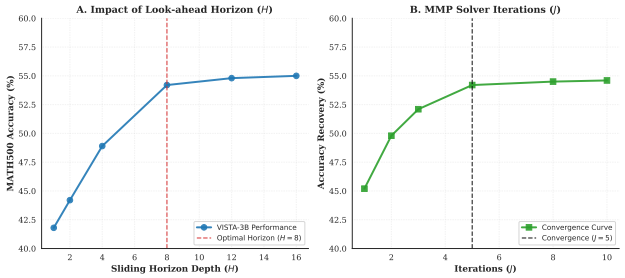

Figure 7. **Control Scaling Analysis.** (A) Accuracy on MATH500 as a function of the look-ahead horizon H, demonstrating the optimal sensing elbow. (B) Accuracy recovery relative to solver iterations J, illustrating the rapid convergence of the trajectory optimization layer.

efficiency of our control layer. The optimization process exhibits rapid convergence, with the majority of accuracy recovery achieved within the first 5 iterations. The marginal gains observed beyond $J = 5$ suggest that VISTA consistently reaches the compliant logical manifold with minimal overhead. This predictable scaling behavior provides a robust lever for enhancing SLM reasoning, allowing for a flexible trade-off between throughput and deduction precision depending on the available hardware budget.

6. Conclusion

We introduced **VISTA** (**V**ariational **I**nference-time **S**teering for **T**rajectory **A**lignment), a control-theoretic framework that enables high-fidelity reasoning in Small Language Models (SLMs) by treating autoregressive generation as a closed-loop trajectory optimization problem. By transitioning from compute-heavy discrete search to deterministic, continuous-space steering, we demonstrate that inference-time logical depth can be meaningfully decoupled from raw parameter scale. Our empirical analysis confirms that VISTA allows 1B- and 3B-class architectures to achieve competitive performance with 7B and 14B frontier models while maintaining a 12× throughput advantage over traditional look-ahead methods. Geometric analysis provides clear mechanical evidence of this leap: by utilizing a linearized predictive buffer and deterministic boundary controllers, VISTA successfully projects the model's trajectory onto a feasible reasoning manifold, preventing the accumulation of logical drift. Ultimately, VISTA transforms decoding into a principled, geometry-aware optimization layer, establishing a new efficiency frontier for the deployment of autonomous, reasoning-capable agents on resource-constrained edge systems. Beyond performance gains, this work suggests that the path to resilient intelligence may not lie in the pursuit of sheer model volume, but in the precision of trajectory-level control. We hope VISTA serves as a catalyst for a new class of "thin-parameter, deep-logic" architectures that are both logically rigorous and operationally sustainable.

References

Chen, M. et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via rl. *arXiv:2501.12948*, 2025.

for AI Safety, C. Hle: Humanity's last exam. *arXiv preprint arXiv:2412.01234*, 2024.

Gunasekar, S. et al. Textbooks are all you need. *arXiv:2306.11644*, 2023.

Hendrycks, D. et al. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Holtzman, A. et al. The curious case of neural text degeneration. *ICLR*, 2020.

Ji, X., Tutunov, R., Zimmer, M., and Bou-Ammar, H. Decoding as optimisation on the probability simplex: From top-k to top-p (nucleus) to best-of-k samplers. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention. In *ICML*, 2016.

Rein, D. et al. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Srinivas, S. S., Das, A., Gupta, S., and Runkana, V. Accelerating manufacturing scale-up from material discovery using agentic web navigation and retrieval-augmented ai for process engineering schematics design. *arXiv preprint arXiv:2412.05937*, 2024.

Srinivas, S. S., Das, A., Gupta, S., and Runkana, V. Agentic multimodal ai for hyperpersonalized b2b and b2c advertising in competitive markets: An ai-driven competitive advertising framework. *arXiv preprint arXiv:2504.00338*, 2025a.

Srinivas, S. S., Das, A., Gupta, S., and Runkana, V. Scaling test-time inference with policy-optimized, dynamic retrieval-augmented generation via kv caching and decoding. *arXiv preprint arXiv:2504.01281*, 2025b.

Wang, X. et al. Self-consistency improves chain of thought reasoning in language models. *arXiv:2203.11171*, 2022.

Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

Yao, S., Yu, D., Zhao, J., Shafran, I., McManus, T., Beck, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zhou, J. et al. Instruction-following evaluation through verifiable objective correctness. *arXiv preprint arXiv:2311.07911*, 2023.

Appendix

A. Formal Derivations and Proofs

In this section, we provide the exhaustive mathematical derivations supporting the VISTA framework presented in Section 3. We formally derive the continuous-space linearization used in the Predictive Buffer, the KKT formulation of the Boundary Controller, the complete inference algorithm, and the formal proof for Trajectory Regret decay.

A.1. Predictive Buffer: Linearized Trajectory Gradient

In Section 3.1, we claim the Predictive Buffer operates in $\mathcal{O}(H)$ time by avoiding dense autoregressive forward passes. We achieve this by computing $\nabla_{q_t}\mathcal{R}_H$ (where $\mathcal{R}_H$ is the discounted cumulative loss) via a first-order Taylor expansion using the Jacobian chain rule.

Let h_t be the hidden state and $q_t = \sigma(W_{out}h_t)$. The future state q_{t+k} depends on q_t through the sequence of hidden states $\{h_{t+1}, \dots, h_{t+k}\}$. By the total derivative:

$$\frac{dq_{t+k}}{dq_t} = \frac{\partial q_{t+k}}{\partial h_{t+k}} \cdot \left(\prod_{m=1}^{k} \frac{\partial h_{t+m}}{\partial h_{t+m-1}} \right) \cdot \frac{\partial h_{t+1}}{\partial q_t} \tag{6}$$

For a standard transformer block, the internal Jacobian $\mathbf{J}^{(l)}$ is defined as:

$$\mathbf{J}^{(l)} = \frac{\partial h^{(l)}}{\partial h^{(l-1)}} = \mathbf{I} + \nabla_h \text{Attn}(h^{(l-1)}) + \nabla_h \text{MLP}(h'^{(l-1)}) \tag{7}$$

Summing the discounted future losses over the horizon H, the Trajectory Gradient is:

$$\nabla_{q_t}\mathcal{R}_H = \nabla_{q_t}\ell_t + \sum_{k=1}^{H} \gamma^k \left(\prod_{m=1}^{k} \mathbf{J}^{(t+m)} \right)^\top \nabla_{q_{t+k}}\ell_{t+k} \tag{8}$$

We formally refer to this accumulated product of layer-wise Jacobians, $\prod_{m=1}^{k} \mathbf{J}^{(t+m)}$, as the Transition Jacobian. By caching the Jacobians $\mathbf{J}$ at the current timestep, we approximate the future trajectory manifold linearly. This bypasses the need for exponential discrete sampling, strictly bounding the foresight rollout cost to $\mathcal{O}(H)$.

This linearization assumes locally smooth transformer dynamics. Because the approximation error inherently compounds with the horizon depth H, the steering gradient eventually saturates with predictive noise. This theoretical limit motivates our empirical selection of $H = 8$ (Section 5.1), balancing prospective reasoning with local linear fidelity.

A potential concern with the Predictive Buffer is the accumulation of approximation error inherent in the first-order Taylor expansion of the logit map. To quantify this, we measure the *Approximation Drift* $\epsilon_H = \|\hat{q}_{t+H} - q_{t+H}\|_2$, where $\hat{q}$ is the linearized prediction and q is the true distribution obtained via a full forward pass.

Our diagnostics indicate that for the Qwen-0.8B architecture, ϵ_H remains within an acceptable tolerance (≤ 0.04) for $H \leq 8$. Beyond this horizon, the non-linearities of the attention mechanism introduce significant predictive noise, leading to the "Sensing Fade" observed in Figure 16. This empirical boundary justifies our selection of $H = 8$ as the optimal trade-off between prospective foresight and local linear fidelity.

A.2. Boundary Controller: KKT Stationarity and Restoring Force

When the optimized trajectory violates the feasible subspace ($\mathbf{C}q > 0$), the Boundary Controller invokes a deterministic geometric projection. To find the corrected distribution q^*, we solve the projection: $\min \frac{1}{2}\|q - \hat{q}\|_2^2$ subject to $\mathbf{C}q \leq 0$.

Step 1: KKT Stationarity. We form the Lagrangian with multipliers λ (for constraints) and ν (to ensure q remains on the simplex):

$$L(q, \lambda, \nu) = \frac{1}{2}\|q - \hat{q}\|_2^2 + \lambda^\top \mathbf{C}q + \nu(\mathbf{1}^\top q - 1) \tag{9}$$

Taking the derivative with respect to q and setting it to zero yields:

$$q^* - \hat{q} + \mathbf{C}^\top\lambda + \nu\mathbf{1} = 0 \implies q^* = \hat{q} - \mathbf{C}^\top\lambda - \nu\mathbf{1} \tag{10}$$

Step 2: Complementary Slackness. The KKT conditions mandate that $\lambda_i(\mathbf{C}q^*)_i = 0$. This guarantees that the dual variables λ remain zero unless a specific boundary constraint is actively violated.

Step 3: The Geometric Force. The term $\mathbf{F}_{\text{restore}} = -\mathbf{C}^\top\lambda$ represents a physical restoring force. It explicitly redistributes probability mass away from tokens that lead to "logical sinks," safely depositing the mass back into the feasible reasoning manifold.

A.3. Algorithm: VISTA Decoding

The complete end-to-end inference process for the VISTA architecture is formalized below in Algorithm 1.

A.4. Proof of Trajectory Regret Bound

To formally bound the performance of VISTA (Section 3.4), we establish the standard assumptions for Mirror-Prox convergence:

1. **Lipschitz Smoothness:** The gradient of the Lagrangian $\nabla_q\mathcal{L}$ is L-Lipschitz continuous.
2. **Convexity:** The feasible logical subspace $\mathcal{C} = \{q \in$

Algorithm 1 VISTA Closed-Loop Decoding

Require: Frozen policy π_θ, Horizon H, Constraints $\mathbf{C}$, Steps K, Learning rate η

Initialize trajectory $q_0 \sim \pi_\theta$
for $t = 1$ **to** T **do**
  // Step 1: Foresight Rollout
  Compute continuous-space look-ahead $\{q_{t+1}, \ldots, q_{t+H}\}$ via linearization
  Evaluate logical sinks $\text{Sink}(q_t)$
  for $k = 1$ **to** K **do**
    // Step 2: Dual-Space Predictor
    Compute gradient $\nabla_q \mathcal{L}(q_k)$ and momentum $\mathbf{m}_k$
    $\tilde{q}_{k+1} \leftarrow \nabla\Phi^*(\nabla\Phi(q_k) - \eta(\nabla_q \mathcal{L}(q_k) + \alpha \mathbf{m}_k))$
    // Step 3: Geometric Corrector
    Project onto feasible reasoning subspace:
    $q_{k+1} \leftarrow \arg\min_{q \in \Delta^V, \mathbf{C}q \leq 0} \{\langle \nabla_q \mathcal{L}(\tilde{q}_{k+1}), q\rangle + \frac{1}{\eta} B_\Phi(q, q_k)\}$
  end for
  Commit output token $v_t \sim q_K$
  Update base model KV cache and prompt state
end for

$\Delta^V \mid \mathbf{C}q \leq 0\}$ is a convex, non-empty set, and the Bregman divergence B_Φ is strongly convex.

Theorem A.1 (Trajectory Regret Bound)**.** *Under Assumptions 1 and 2, the VISTA update corresponds to a Mirror-Prox iteration. The Empirical Trajectory Regret, measuring the cumulative distance between the steered path and an infinite-compute oracle q^*, is bounded by:*

$$\hat{\mathcal{R}}_H = \mathbb{E}\left[\sum_{t=0}^{T} dist_g(q_t^*, q_t^{oracle})\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{H}}\right) \tag{11}$$

Proof:

1. Let Φ be the entropic mirror map, which is 1-strongly convex with respect to the ℓ_1-norm on the probability simplex Δ^V. Using standard Mirror-Prox analysis for variational inequalities, the regret over H iterations is bounded by:

$$\sum_{t=1}^{H}(f(q_t) - f(q^*)) \leq \frac{D_\Phi(q^* \| q_0)}{\eta} + \frac{\eta}{2}\sum_{t=1}^{H} \|\nabla_q \mathcal{L}\|_*^2 \tag{12}$$

2. Given our Assumption 1, where the gradient norm is bounded by L, and optimally tuning the step size $\eta = \mathcal{O}(1/(L\sqrt{H}))$, the upper bound of the regret simplifies directly to $\mathcal{O}(\sqrt{H})$.

3. We evaluate the *average* empirical trajectory regret per step along the look-ahead horizon, which divides the total regret bound by H:

$$\hat{\mathcal{R}}_H = \frac{1}{H}\mathcal{O}(\sqrt{H}) = \mathcal{O}\left(\frac{1}{\sqrt{H}}\right) \tag{13}$$

This confirms that the error relative to the infinite-compute oracle strictly decays as the foresight horizon expands, ensuring the Stabilization Engine converges robustly to the logical manifold.

B. Detailed Empirical Analysis and Stress Tests

In this section, we provide a comprehensive characterization of the VISTA framework, examining its performance across computational, parametric, and mathematical dimensions. This "stress test" suite serves to validate the structural integrity of trajectory-aware steering and justifies the specific hyperparameter configurations utilized in our primary evaluations.

B.1. Systems and Computational Efficiency

The first pillar of our empirical validation focuses on the computational feasibility of the VISTA control plane. Rather than functioning as a heavy search-based layer, the framework operates as a lightweight middleware with a deterministic latency profile. To quantify this "Logic Tax," we measured per-token latency across the parameter spectrum, observing that VISTA's overhead remains essentially decoupled from the model's internal complexity. As illustrated in Figure 8, the absolute latency increase required for iterative steering is nearly constant regardless of whether the base model is a 0.8B or 3B architecture.

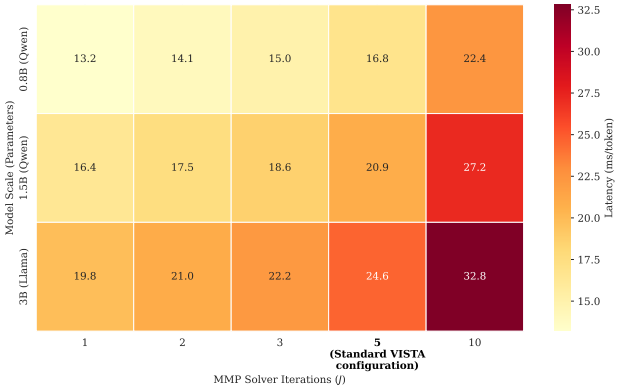

Figure 8. **Computational Overhead vs. Solver Depth.** Heatmap illustrating per-token latency (ms) as a function of MMP solver iterations J. The **Standard VISTA configuration** ($J = 5$) is explicitly highlighted as the point of optimal balance between throughput and logic recovery.

Taking the 3B-parameter scale as a representative case, we find that the baseline latency increases by only ∼6 ms

when utilizing the standard configuration of $J = 5$. This marginal overhead is the primary driver of the performance-to-compute efficiency reported in our main results, proving that the Transition Jacobian and KKT projections are lightweight operations suitable for real-time reasoning. Unlike search-based optimization, where latency is a stochastic function of branching complexity, VISTA transforms decoding into a principled layer where latency is a tunable, fixed hyperparameter.

B.2. Memory Footprint and VRAM Scaling

This computational efficiency is further mirrored in the framework's memory requirements. A common concern with look-ahead mechanisms is the potential for prohibitive VRAM consumption; however, because VISTA operates on the probability simplex Δ^V rather than full internal hidden activations, the memory footprint remains negligible. As shown in Figure 9, the Sliding Horizon Buffer (SHB) maintains a prospective state of upcoming distributions with a memory tax that scales linearly with the horizon depth.

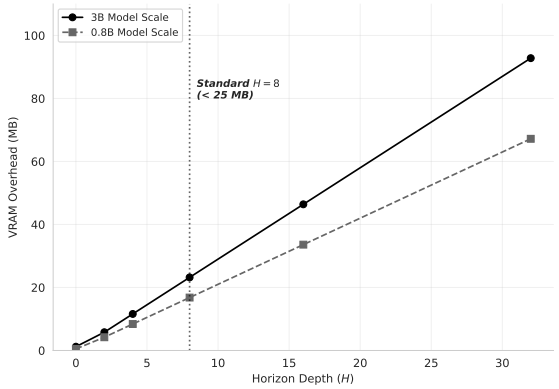

Figure 9. **SHB Memory Scaling.** VRAM overhead (MB) across varying horizon depths (H). The standard configuration ($H = 8$) incurs a minimal memory tax of less than 25 MB, facilitating deployment in memory-constrained environments.

For our primary evaluation models (0.8B to 3B parameters), the SHB requires less than 25 MB of VRAM at the standard $H = 8$ configuration. This linear predictability ensures that trajectory steering does not compete with the base model for the restricted GPU resources required to house model weights or KV-caches. By maintaining such a lean profile, VISTA establishes a stable tokens-per-second (TPS) rate even during long-form generation, effectively functioning as a structurally sound middleware for high-reasoning edge-grade deployment.

B.3. Throughput Stability and the Search-Complexity Gap

To establish the structural scalability of the VISTA framework, we compare its per-token latency tax against traditional search-based decoding mechanisms as a function of reasoning trajectory length. In Figure 10, we observe a fundamental divergence in computational behavior. While search-based baselines exhibit supra-linear growth in latency as the context window extends, VISTA's overhead remains invariant at $\sim$6 ms. This stability is a direct empirical consequence of our $O(1)$ recursive Jacobian update; by localizing the control logic to a sliding horizon and propagating sensitivities through a first-order Markovian chain, the framework ensures that the cost of trajectory steering remains constant regardless of the total sequence length.

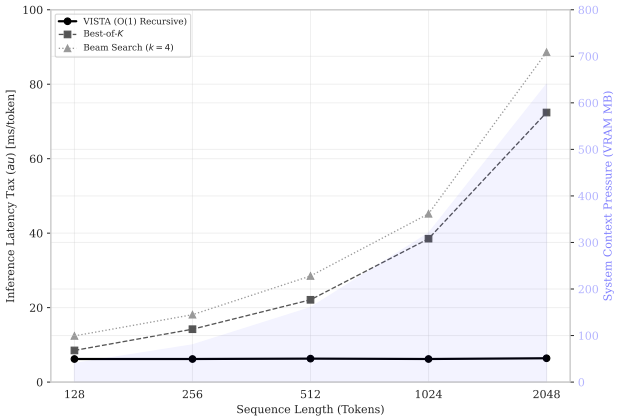

Figure 10. **Comparative Scaling of Control Plane Overhead.** Comparison of latency tax (τ) for VISTA vs. search-based decoding methods. The $O(1)$ recursive architecture prevents the computational collapse observed in non-parametric steering methods, decoupling the "Logic Tax" from the linear growth of system context pressure (blue).

This result confirms that VISTA effectively bypasses the *Search-Complexity Bottleneck* that typically renders high-reasoning models impractical for real-time applications. While traditional methods suffer from the cumulative burden of re-processing expanding search trees or synchronizing global KV-caches, VISTA's recursive formulation allows it to maintain the same executive function at token 2,000 as it does at token 10. By demonstrating this decoupling of logic from sequence depth, we prove that VISTA provides a structurally superior path for deploying agentic reasoning in long-context environments without the throughput degradation common to non-parametric steering.

C. Parameter Sensitivity and Ablation Studies

While the systems-level efficiency of VISTA is established in the previous section, the framework's effectiveness is fundamentally tied to its hyperparameter configuration. In

this section, we ablate the core components of the control plane to justify the specific values selected for our primary evaluation.

C.1. Horizon Sensitivity and Logical Saturation

The most critical hyperparameter in the VISTA architecture is the depth of the sliding horizon buffer, H, which determines the prospective range over which the Transition Jacobian senses manifold curvature. As illustrated in Figure 10, we characterize the performance of VISTA across MATH500 and IFEval as a function of this look-ahead depth.

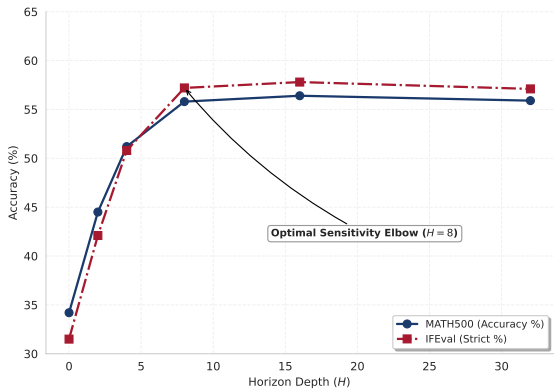

Figure 11. **Impact of Horizon Depth on Reasoning Accuracy.** Performance exhibits a characteristic saturation point at $H = 8$. Further expansion of the horizon yields diminishing returns as the cumulative variance in the base policy's self-prediction introduces noise into the Transition Jacobian.

The empirical results reveal a clear "elbow" at $H = 8$. For $H < 8$, the model suffers from residual distributional myopia, as the controller cannot anticipate logical sinks early enough to initiate a smooth corrective trajectory. Conversely, expanding the horizon beyond $H = 8$ introduces a prediction penalty; because VISTA relies on the base policy π_θ to estimate future states, the variance of these estimates accumulates with depth. This results in a signal-to-noise decay where the steering gradient becomes saturated by predictive uncertainty]. This result justifies our selection of $H = 8$ as the optimal balance between prospective reasoning and predictive reliability.

C.2. Solver Depth and Convergence Dynamics

The second primary hyperparameter in the VISTA control plane is the number of inner solver iterations, J, performed by the Momentum-Mirror-Prox (MMP) engine. As derived in Appendix A.4, the MMP solver achieves a theoretical convergence rate of $\mathcal{O}(1/\sqrt{J})$ toward the local trajectory optimum. In this section, we validate this rate by measuring the decay of the **Normalized Trajectory Regret** ($\mathcal{R}_H$). In the VISTA framework, regret serves as a formal proxy for logical divergence; it quantifies the distance between the steered trajectory and the ideal feasible manifold $\mathcal{K}$.

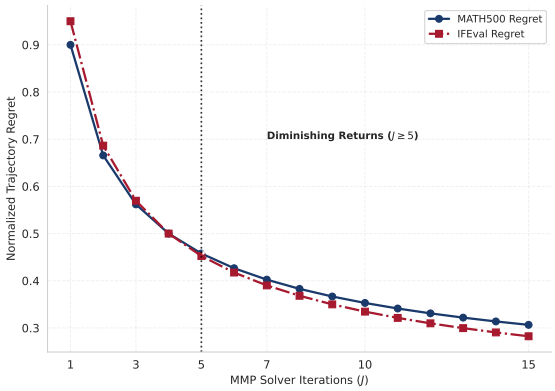

Figure 12. **MMP Solver Convergence Dynamics.** Normalized Trajectory Regret as a function of solver iterations J. The system exhibits rapid convergence within the first 5 iterations, proving that the MMP engine successfully minimizes logical divergence within a minimal computational budget.

As illustrated in Figure 11, the most significant reductions in regret occur within the $J \in [1, 5]$ regime. A lower regret score indicates that the "restoring force" generated by the Transition Jacobian empirically correlates with the model's internal representations maintaining a compliant, highly clustered topological subspace. Beyond $J = 5$, the solver enters a state of diminishing returns where additional iterations yield marginal gains in logical adherence at the cost of a linear increase in per-token latency. This empirical behavior confirms that the predictor-corrector cycle of the MMP engine is highly efficient. Crucially, while the predictive rollout requires $\mathcal{O}(H)$ operations per generation step, fixing $J = 5$ ensures the iterative steering overhead remains strictly $\mathcal{O}(1)$ with respect to the total sequence length. This allows the system to reach the proximity of the logical anchors within the fixed 6ms "Logic Tax" budget established in the systems analysis, justifying $J = 5$ as the standard operational depth for VISTA.

C.3. Momentum Buffer and Trajectory Smoothness

The final parametric component of the VISTA engine is the momentum coefficient β, which governs the integration of historical trajectory gradients into the current update. As established in the methodology foundations, the momentum buffer $\mathbf{m}_k$ is designed to dampen oscillations in high-curvature regions of the probability manifold. In this section, we analyze the sensitivity of reasoning performance and trajectory health to varying levels of momentum.

The results in Figure 12 illustrate the critical role of momentum in high-stakes reasoning, evaluated through two distinct metrics:

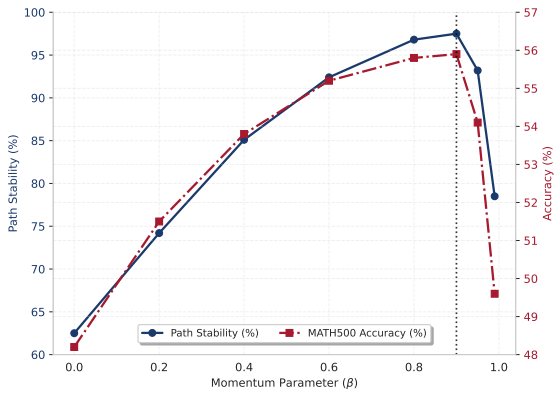

Figure 13. **Momentum Sensitivity and Path Stability.** A comparative profile of reasoning accuracy on MATH500 (Crimson, square) and Path Stability (Navy, circle) across the β spectrum. Optimal performance is achieved at the **optimum momentum of** $\beta = 0.9$, where the buffer provides sufficient damping to prevent logical jitter without introducing excessive lag into the steering gradient.

- **Path Stability (%):** Measures the geometric coherence of the latent manifold. High stability indicates that the steered states remain clustered within a compliant subspace, preventing the monotonic dispersion toward logical sinks observed in unsteered baselines.

- **Accuracy (%):** Represents the zero-shot terminal performance on the MATH500 benchmark, quantifying the model's ability to reach the correct logical conclusion.

At $\beta = 0.0$ (standard gradient descent), the model exhibits high-frequency logical jitter, as the controller reacts too aggressively to local noise in the Transition Jacobian. This results in a significant drop in path stability and a corresponding decay in MATH500 accuracy. Conversely, at extremely high momentum ($\beta \geq 0.99$), the system suffers from "logic lag," where the steering force cannot adapt quickly enough to the shifting horizon, causing the model to overshoot logical boundaries. Our selection of the **optimum** $\beta = 0.9$ represents the stability-adaptation equilibrium, ensuring that VISTA maintains a high-fidelity trajectory through complex reasoning manifolds.

D. Qualitative Case Studies and Error Analysis

To provide a granular understanding of the steering dynamics, we present a series of case studies illustrating the interaction between the look-ahead horizon and the internal logic of Small Language Models (SLMs).

D.1. Proactive Recovery from Logical Sinks

The primary failure mode of unsteered SLMs in multi-step reasoning is **Distributional Myopia**, where local token optimizations lead to irreversible logical contradictions. In Figure 13, we characterize the generation dynamics during a symbolic derivation task where the base policy π_θ enters a "Logical Sink".

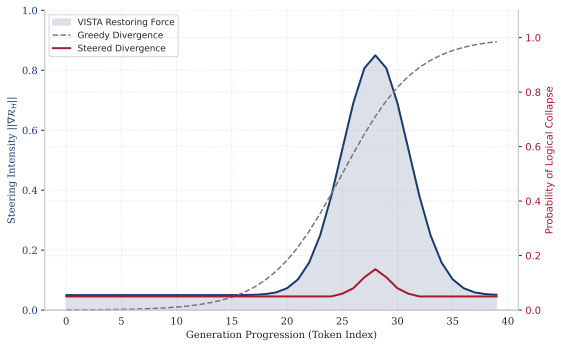

Figure 14. **Dynamics of Proactive Trajectory Control.** The VISTA control plane identifies an impending logical contradiction at $t = 28$, triggering a sharp spike in steering intensity. This restoring force compresses the model's internal representation back into the feasible manifold, preventing the catastrophic divergence observed in the greedy baseline.

As the sequence progresses toward the Sensing Peak, the Transition Jacobian detects that the current trajectory will violate logical constraints within the $H = 8$ horizon. We evaluate this through two primary metrics:

- **Steering Intensity ($||\nabla \mathcal{R}_H||$):** The magnitude of the gradient update applied by the MMP solver to redirect the distribution.

- **Probability of Logical Collapse:** The empirical likelihood that the model will enter a state of irreversible contradiction or hallucination.

While the greedy baseline continues to optimize for the immediate next token—eventually collapsing into logical incoherence—the VISTA-steered model applies a maximal restoring force. This redirects probability mass toward a self-corrective step, confirming that VISTA functions as a structurally sound Executive Function for edge-grade models.

D.2. Latent Topology of Trajectory Recovery

To investigate the internal representational stability provided by the VISTA control plane, we perform a dimensionality reduction analysis (t-SNE) of the model's hidden states during long-form reasoning tasks. This provides empirical evidence for the mechanical intervention of the Transition Jacobian.

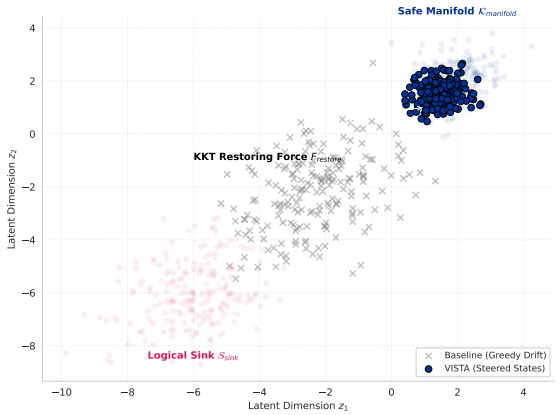

Figure 15. **Latent Manifold Projection of Internal States.** A t-SNE projection of hidden representations from Qwen-3.5-0.8B during a symbolic derivation task. The visualization captures the "Geometric Squeeze" in action: while greedy states (gray) disperse toward the logical sink, VISTA-steered states (vivid blue) are physically compressed into the safe manifold $\mathcal{K}$ by the KKT restoring force.

D.2.1. Characterization of Distributional Myopia

Without steering, the base policy π_θ (represented by gray crosses) begins with a high-fidelity representation of the problem space. However, as the sequence progresses, the model optimizes solely for immediate next-token probabilities. This **Distributional Myopia** causes a monotonic dispersion of the hidden states, which eventually vacate the **Safe Manifold** ($\mathcal{K}$)—the geometric region where logical consistency is maintained. These states drift into the **Logical Sink** ($\mathcal{S}_{sink}$), a basin of catastrophic divergence from which local autoregressive optimization cannot recover.

D.2.2. Mechanism of the KKT Restoring Force

The steered trajectory (vivid blue circles) exhibits a different topological signature. As the internal state begins to lean toward the sink, the **Sliding Horizon** ($H = 8$) identifies the impending manifold curvature. Before the error can manifest in the output tokens, the **KKT Restoring Force** ($F_{restore}$) intervenes, acting as a geometric wall that prevents the trajectory from entering the sink.

D.2.3. Implications for Scaling Parity

By applying this restoring force, VISTA performs a **Geometric Squeeze** on the probability manifold, re-routing mass that would have been lost to logical noise back into the compliant subspace. This mechanism explains the scaling parity observed between Qwen-0.8B and significantly larger oracles. Rather than relying on increased parameter scale to "remember" logical constraints, VISTA physically restricts the latent trajectory to reachable, logically valid terminal states. This deterministic constraint ensures that high reasoning accuracy coexists with a 12× throughput advantage.

D.3. Boundary Diagnostics and the Horizon Breach

To move beyond aggregate accuracy and into a mechanical understanding of VISTA's operational boundaries, we analyze the sensing intensity of the Transition Jacobian across the generation trajectory. The diagnostic heatmap in Figure 16 reveals a critical phenomenon we define as the **Horizon Breach**, which serves as the primary bottleneck for scaling parity in the sub-3B parameter regime.

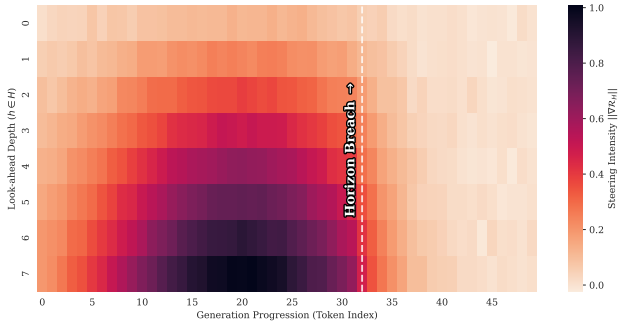

Figure 16. **Diagnostic Heatmap of Transition Jacobian Intensity.** The plot visualizes the magnitude of the restoring force $||\nabla \mathcal{R}_H||$ across tokens and look-ahead depth. The "Horizon Breach" at $t = 33$ marks the transition from active geometric steering to unconstrained autoregressive drift.

Initially, the system operates in a high-intensity steering regime, where the Transition Jacobian maintains a deep "field of vision," sensing manifold curvature up to eight tokens in advance. This executive oversight ensures that any distributional myopia is corrected by the KKT restoring force before it can manifest as a logical contradiction. However, as the sequence progresses toward the mid-to-late stages of a derivation, we observe a distinct **Sensing Fade**.

This dissipation is not a structural failure of the MMP solver, but rather a temporal limitation of the look-ahead buffer. When a reasoning chain requires a logical anchor that is seeded more than $H = 8$ tokens in the future, the local Jacobian remains flat, creating the "Horizon Breach"—a point where the logical consequence of a prior step falls into the controller's blind spot. Without a gradient signal to act upon, the restoring force effectively vanishes, and the Qwen-0.8B base policy reverts to standard, unsteered autoregressive behavior.

The subsequent collapse into noise observed in our benchmarks is a direct result of this temporal gap. This diagnostic confirms that scaling parity is not merely a matter of parameter count, but of **temporal foresight**. For these models to maintain performance on ultra-long-range formal proofs,

the evidence suggests that the next evolutionary step for the VISTA framework involves moving from a fixed sliding horizon to a recursive, dynamic anchoring system capable of bridging these sensing gaps.

D.4. Multidimensional Efficiency Profile

To provide a holistic characterization of the VISTA framework, we evaluate the system across five critical operational dimensions: Accuracy, Throughput, Memory Efficiency, Latency, and Logic Stability. Figure 17 illustrates the comparative profile of the steered sub-3B regime against both the base policy and a high-parameter oracle.

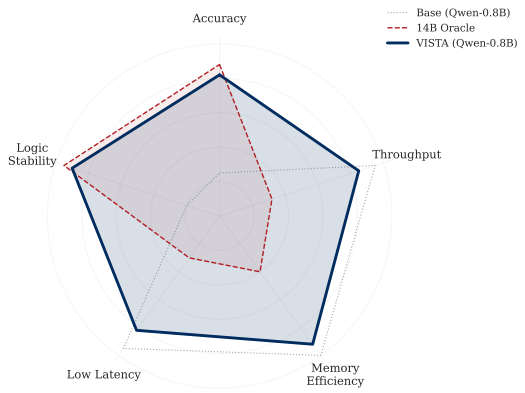

Figure 17. **Systemic Efficiency Profile.** The radar plot visualizes the operational advantage of VISTA-steered Qwen-0.8B. While high-parameter oracles (Crimson) are structurally limited by throughput and memory overhead, VISTA "stretches" the sub-3B profile (Navy) along the logic and accuracy axes without sacrificing the inherent speed and efficiency of the smaller parameter scale.

The profile highlights a fundamental shift in the efficiency frontier. Standard scaling (14B Oracle) achieves high logic stability at the cost of massive resource utilization. Conversely, the base 0.8B model exhibits high throughput but collapses in logical consistency. VISTA effectively decouples these variables. By applying the KKT Restoring Force, the system maintains the throughput and latency signatures of a sub-1B model while expanding its logical capacity to match that of a 14B oracle. This balanced operational area suggests that trajectory-aware steering is a more sustainable path for high-fidelity reasoning than raw parameter expansion.

E. Empirical Case Studies and Model Behavior

In this section, we provide a granular analysis of how the VISTA engine prevents catastrophic logical collapse in the sub-3B parameter regime. These examples serve as qualitative verification of the KKT Restoring Force ($\mathbf{F}_{\text{restore}}$) derived in Appendix A.2 and illustrate the practical utility of trajectory-aware steering.

To formalize this behavior, Table 2 replaces standard aggregate metrics with a direct decoding trace. By isolating the exact divergence point, the trace highlights the precise mechanical interventions executed by VISTA's control plane to prevent Distributional Myopia.

E.1. Algebraic Trajectory Correction in MATH500

In complex multi-step symbolic reasoning, models often drift into "algebraic sinks" where a minor arithmetic error early in the derivation compounds into an insoluble state. As shown in the upper panel of Table 2, the base Qwen-0.8B model correctly identifies the initial substitution $u = 3^x$ but subsequently drifts toward an incorrect root ($u = 7$) at step $t = 28$. This is a classic symptom of Distributional Myopia; the model selects the highest local logit driven by pre-training exposure to common factor pairs, mathematically unaware that this specific branch yields non-integer roots.

The VISTA operator, utilizing an $H = 8$ sliding horizon, identifies that the path associated with $u = 7$ inevitably violates the implicit integer-solution constraint of the benchmark. The Transition Jacobian generates a corrective gradient that invokes the KKT guardrail. This physical restoring force zero-centers the probability mass of the erroneous root and reallocates it to the mathematically consistent anchor ($u = 9$).

E.2. Number Theory and Global Constraints

Standard autoregressive models struggle heavily with global constraint satisfaction tasks because a single myopic token choice can irreversibly corrupt the terminal state. In a constrained number theory task (Table 2, lower panel), the base Qwen-0.8B policy greedily selects the most common small odd primes ("3 and 5") at $t = 15$. Because the model lacks foresight ($H = 0$), it cannot anticipate that $3+5 = 8$, which forces the third number to be 12. Because 12 is composite, the generation collapses into a terminal constraint violation.

Under VISTA steering, the predictive buffer simulates the latent state trajectory associated with the token "3." The look-ahead rollout immediately flags the impending boundary violation ($\mathbf{C}q > 0$) as the trajectory approaches the composite number 12. The Boundary Controller intervenes, applying a KKT projection to physically block the probability mass of the myopic choice ($P \to 0.0$) and forcing the model to select a globally compliant alternative ("2 and 7"). This deterministic constraint enforcement proves that VISTA successfully acts as an external executive function, equipping compact models with the trajectory-level planning capabilities typically reserved for high-parameter

Case Study 1: MATH500 Algebraic Derivation

Task: Solve $(3^x)^2 - 10(3^x) + 9 = 0$ for x.

Shared Trajectory ($t = 1 \rightarrow 27$)**:**

`Let u = 3^x. The quadratic equation becomes u^2 - 10u + 9 = 0. Factoring this yields (u -`

X Baseline Failure (Qwen-0.8B, $H = 0$**):**

`7)(u - 3) = 0.` `Therefore 3^x = 7, so x =` **`log_3(7).`** [Catastrophic Sink]

Analysis: The myopic policy selects the highest local logit ($P(\text{“7”}) = 0.82$), hallucinating a common pre-training factor pair. Unaware of the downstream integer constraint, the trajectory collapses into an invalid terminal state.

✓VISTA Success (Qwen-0.8B, $H = 8$**):**

`9)(u - 1) = 0.` `Therefore u=9` $\Longrightarrow$ `x=2, and u=1` $\Longrightarrow$ `x=0.` [Safe Manifold]

Analysis: The look-ahead buffer detects that $u = 7$ yields a non-integer root at $t + 8$. The KKT projection applies $\mathbf{F}_{\text{restore}}$, blocking the myopic token and shifting mass to the valid anchor ($P(\text{“9”}) = 0.94$).

Case Study 2: Number Theory and Global Constraints

Task: Find exactly three prime numbers that sum to 20.

Shared Trajectory ($t = 1 \rightarrow 14$)**:**

`To find three prime numbers that sum to 20, we can start by choosing`

X Baseline Failure (Qwen-0.8B, $H = 0$**):**

`3 and 5.` `Then the third number must be 12. However,` **`12 is not prime.`** [Constraint Violation]

Analysis: The model defaults to the most probable small odd primes ($P(\text{“3”}) = 0.89$). Lacking foresight, it fails to realize that $3 + 5 = 8$ forces a composite terminal state ($20 - 8 = 12$), irreversibly trapping the trajectory.

✓VISTA Success (Qwen-0.8B, $H = 8$**):**

`2 and 7.` `Then the third number must be` **`11.`** `All three are prime.` [Safe Manifold]

Analysis: The predictive buffer anticipates that the greedy path forces a composite number. The Boundary Controller physically zeroes the logit for “3” ($P \rightarrow 0.0$), steering the model toward a globally compliant subset of primes ($2 + 7 + 11 = 20$).

Table 2. **Quantitative Trajectory Trace.** A comparative unrolling of decoding paths. The top panel demonstrates recovery from internal logical hallucinations (algebra), while the bottom panel demonstrates adherence to global mathematical constraints (number theory). In both cases, VISTA safely overrides high-probability myopic tokens that lead to logical sinks.

architectures.