MASK-LOCAL ADAPTERS FOR PARALLEL DECODING WITH FROZEN AUTOREGRESSIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adding multi-token prediction to a pretrained language model requires choosing where to allocate trainable capacity and how to accept predictions made without their actual preceding tokens. We study MASK-local residual adapters that update future placeholder positions while freezing the backbone and vocabulary projection. A no-MASK bypass retains the original sequential path, while real-token cache recommit supports approximate parallel decoding. After three training epochs, 3.22M and 4.21M adapter parameters achieve 57.36% and 61.39% future-position agreement on Qwen2.5-7B and Llama3.1-8B, respectively. On matched Qwen evaluation windows, agreement exceeds a PaSS-style embedding baseline by 26.60 percentage points. In a four-benchmark development screen, Llama's clean sequential path reproduces all 214 reference sequences. Its separate confidence-based parallel mode accepts 1.34–2.09 tokens per step, with substantial accuracy losses on HumanEval and BBH. On a 64-question Qwen GSM8K runtime screen, throughput rises from 45.39 to 54.58 tokens/s while accuracy changes from 58/64 to 55/64. Comparisons of confidence, direct-error supervision, and tree-derived risk characterize approximate acceptance within this parameter-efficient extension.

1 INTRODUCTION

Autoregressive language models generate a continuation by repeatedly predicting the next token conditioned on the tokens already emitted. Each decision requires a new model evaluation even when several upcoming tokens are predictable from the same context. Parallel generation instead proposes multiple future tokens in one evaluation and commits a selected prefix of that proposal. The opportunity is to amortize a large model's computation across several emitted tokens.

We ask whether this capability can be added to a deployed model while preserving its original parameters and sequential computation. This setting makes parallel generation an optional extension: the same backbone can serve an unchanged sequential path or a learned multi-token path. Unlike continued training of the full model, capability extension isolates the new behavior in explicit trainable modules. We study whether small residual branches at future placeholder positions can learn useful proposals while leaving the real-token path intact.

We organize the problem around proposal quality, acceptance, and execution. A good parallel proposal approximates the sequential model at several positions without receiving their actual preceding tokens. Acceptance determines how much of that proposal to emit. Execution must then condition subsequent proposals on the real accepted tokens, rather than on the placeholders used to predict them. Figure 1 connects these components.

Our proposal module adds residual adapters after decoder blocks, restricted to MASK positions. We train these modules using continuations generated by the frozen model itself. To study acceptance, we define block disagreement and construct risk targets using decision-gap moments and a finite autoregressive tree. Small controllers predict risk scores from the same forward pass that produces the candidates. Finally, the decoder merges real-token recommit with the next proposal, exposing the benefit of multiple accepted tokens without retaining MASK-derived history.

Our main contribution is a MASK-local allocation of trainable capacity for frozen-backbone future-token prediction, evaluated on two model families and against a matched-window embedding base-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

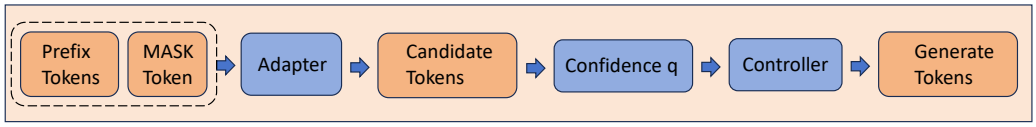

Figure 1: System overview. The adapter stage denotes the frozen backbone with MASK-local residual branches. The controller reads hidden states and proposal statistics, including confidence q, and selects a prefix for real-token recommit.

line. We also study confidence and learned risk scores within a shared acceptance interface, separating proposal quality from the decision to emit multiple tokens. Finally, real-token recommit enables an execution study that reports task accuracy and measured speed alongside acceptance length. The sequential bypass preserves an original-model option, while the parallel policies make approximate acceptance decisions.

2 Related Work

Speculative and Adaptive Decoding. Speculative decoding uses inexpensive candidate generation followed by target-model verification, accepting multiple tokens when the proposal and target agree (Leviathan et al., 2023; Chen et al., 2023). The verifier can evaluate candidate positions in parallel while preserving the target sampling distribution through the appropriate acceptance rule. Adaptive generation also chooses how much work to perform at a decoding step. Our setting uses the original model itself to generate candidates and makes an approximate acceptance decision from proposal confidence and learned risk features. Thus, generation and risk assessment share a forward pass; this is a different operation from exact target verification.

PaSS. PaSS learns look-ahead input tokens to obtain parallel drafts from a single model, avoiding a separate drafter (Monea et al., 2023). It then verifies the draft under the target model. We share the frozen-backbone, single-model motivation, but place additional capacity throughout the decoder at MASK positions rather than only in the input embeddings. This increases the trainable budget while keeping it small relative to the backbone. Our matched-window comparison measures the resulting local future-token agreement; it isolates proposal learning rather than comparing complete PaSS sampling and our approximate greedy decoder.

Frozen-backbone prediction heads. Medusa-1 trains additional future-token prediction heads on a frozen backbone and verifies candidate continuations with tree attention (Cai et al., 2024). Frozen-backbone multi-token prediction is therefore an established setting. Our design instead places residual capacity at MASK positions throughout the decoder and reuses the original vocabulary projection. Our acceptance study uses same-forward risk assessment without target verification. Establishing an advantage over prediction heads requires a matched-budget comparison beyond the embedding baseline reported here.

MARS. MARS teaches autoregressive models multi-token generation through continued training with masked and clean autoregressive objectives, confidence-based acceptance, and a block-cache implementation (Jin et al., 2026). We retain the original backbone weights and allocate new capacity to a bypassable branch. Our control study additionally compares confidence with risk scores derived from the same forward pass. We use real-token recommit and merged proposal scheduling in this setting, building on the general cache-scheduling principle rather than attributing a new verification guarantee to merging.

Table 1: Mechanism taxonomy: exact verification versus same-forward risk assessment.

	Separate exact verification	Same-forward risk assessment
Separate drafter	Standard speculative decoding	—
No separate drafter	PaSS	MARS; our framework

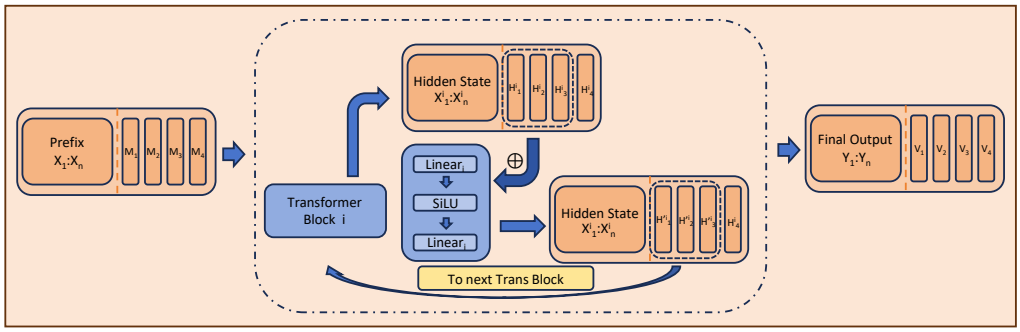

Figure 2: MASK-local adapter. The residual branch updates only M_1–M_3 after each decoder block. Real-token positions retain the block output. The original final normalization and LM head produce all candidate distributions.

3 Learning Parallel Capabilities with a Frozen Backbone

Let $p_\theta(\cdot \mid h)$ be a frozen causal language model at a real-token prefix h. A proposal module with trainable parameters ϕ produces distributions $Q_{\phi,j}(\cdot \mid h)$ and greedy candidates $a_j = \arg\max_v Q_{\phi,j}(v \mid h)$ for $j = 1, \ldots, W$. Our implementation uses $W = 4$. A policy selects $K \in \{1, \ldots, W\}$ and emits $a_{1:K}$, clipped at the first stop token. The backbone θ and the original vocabulary projection remain fixed throughout adapter training. Sequential decoding omits all MASK inputs and does not invoke the learned residual updates.

3.1 MASK-Local Adapter Architecture

We append four virtual MASK positions to the prefix and run the original causal architecture. At decoder block ℓ, let z_i^ℓ be the block output at position i. The added branch is

$$\tilde{z}_i^\ell = z_i^\ell + m_i B_\ell \, \mathrm{SiLU}(A_\ell z_i^\ell), \quad A_\ell \in \mathbb{R}^{r \times d}, \quad B_\ell \in \mathbb{R}^{d \times r}, \tag{1}$$

where $r = 16$ and m_i selects the first three MASK positions. This nonlinear residual bottleneck allocates capacity to the proposal positions without changing the base attention or feed-forward weights. It differs from low-rank weight updates such as LoRA (Hu et al., 2021): the branch is position-selective and is absent on real-token inputs.

Each active MASK embedding is a fixed parent MASK vector plus a trainable role vector e_j. The implementation appends a fourth MASK to preserve its fixed proposal shape, although causality makes that position unnecessary for the four predictions. The final real-token state predicts a_1, and MASK states M_1, M_2, M_3 predict a_2, a_3, a_4 (Figure 2). With L layers, the adapter and role-vector count is $2Ldr + 3d$, giving 3,222,016 parameters for Qwen2.5-7B and 4,206,592 for Llama3.1-8B. Controller parameters are additional.

3.2 Training and Sequential Preservation

For each training prefix h, we greedily unroll the frozen model to obtain targets $y^*_{1:4}$. Only future-position targets supervise the new branch:

$$\mathcal{L}_{\text{adapter}}(\phi) = -\sum_{j=2}^{4} v_j \log Q_{\phi,j}(y_j^* \mid h), \tag{2}$$

where v_j is a binary validity indicator that excludes targets after the first end-of-sequence token. The first candidate already uses the causal real-token state. Training windows are drawn from reference-text contexts, but the prediction targets come from the model being adapted. This separates context construction from teacher labeling and allows each model family to use its native tokenizer and stopping convention.

When no MASK is present, every m_i is zero and the wrapper calls the original model on token IDs. Equation 1 therefore provides a structural bypass rather than a distillation objective for preserving sequential behavior. We test this property using both same-shape logits and complete greedy sequences. Merged decoding uses a different execution shape, so its K1 arm is a separate runtime control: it accepts one token while still computing the MASK proposal. This distinction lets us measure the clean sequential path independently of numerical trajectory changes in a batched proposal.

4 Block Disagreement and Adaptive Acceptance

4.1 Block Disagreement and Confidence

Fix a proposal $a_{1:W}$ at prefix h. For an autoregressive history $X_{<j}$, define the conditional decision $g_j(X_{<j}) = \arg\max_v p_\theta(v \mid h, X_{<j})$ with deterministic tie-breaking. Under a specified history law ν, extension-block disagreement is

$$\Delta_K^\nu(h, a) = \mathbb{P}_{X\sim\nu}[\exists j \in \{2, \dots, K\} : g_j(X_{<j}) \neq a_j], \qquad \Delta_1^\nu = 0. \tag{3}$$

The distribution over histories matters: our U-bank uses a locally normalized Top-8 tree, defined below. Direct supervision instead evaluates a single greedy continuation and records $e_K = \mathbf{1}[a_{1:K} \neq y^*_{1:K}]$. Direct-label training excludes contexts with a first-token mismatch. The extension event also omits the first position, so these targets, as used for controller training, do not assess first-token errors from the merged numerical path. Task correctness evaluates the completed answer separately.

The confidence policy summarizes the least confident accepted future position:

$$q_1 = 0, \qquad q_K = 1 - \min_{2\le j\le K} \max_v Q_{\phi,j}(v \mid h) \quad (K \ge 2), \qquad K_q = \max\{K : q_K \le 0.05\}. \tag{4}$$

It is inexpensive and naturally adapts block length to the proposal. However, confidence describes a proposal distribution, whereas disagreement depends on how conditional predictions change with earlier tokens. Consider a sequential rule that copies the immediately preceding token. If the proposal is certain about (A, B), with $A \neq B$, its confidence score can be zero while the sequential continuation after A must produce A. The example establishes that proposal confidence alone does not bound conditional disagreement. Tree-derived targets describe conditional behavior across alternative histories. Their utility for predicting the deployed greedy error is an empirical question.

4.2 Moment and Tree Risk Targets

For $j \in \{2, \dots, W\}$ and $K \in \{2, \dots, W\}$, define the decision gap

$$G_j(x_{<j}) = \max_{v\neq a_j} p_\theta(v \mid h, x_{<j}) - p_\theta(a_j \mid h, x_{<j}), \qquad G_K^{\max} = \max_{2\le j\le K} G_j. \tag{5}$$

A mismatching greedy decision implies $G_K^{\max} \ge 0$; ties are included conservatively. For mean μ_K and variance σ_K^2 under a fixed history law, Cantelli gives

$$\Delta_K^\nu \le \mathbb{P}_\nu(G_K^{\max} \ge 0) \le C_K, \qquad C_K = \begin{cases} \sigma_K^2/(\sigma_K^2 + \mu_K^2), & \mu_K < 0, \\ 1, & \mu_K \ge 0. \end{cases} \tag{6}$$

For the implemented tree targets, each expanded node branches on its eight most likely tokens and renormalizes those transitions, defining a history law $\widetilde{P}$. Let $\widetilde{G}_j$ denote the gap computed with these induced Top-8 node probabilities, and let $\widetilde{G}_K^{\max} = \max_{2\le j\le K} \widetilde{G}_j$. Top-8 truncation preserves the top-ranked decision but changes gap magnitudes. Enumeration gives $T_K = \mathbb{P}_{\widetilde{P}}(\widetilde{G}_K^{\max} \ge 0)$. Applying Equation 6 to the mean and variance of this same truncated gap under $\widetilde{P}$ gives $\widetilde{C}_K$. We retain the name *exactU* for the tail-adjusted target below. Here exact refers to enumeration of T_K, not an exact deployment-error probability:

$$U_K^{\text{exact}} = R_K T_K + (1 - R_K), \qquad U_K^{\text{Cantelli}} = R_K \widetilde{C}_K + (1 - R_K), \tag{7}$$

where R_K is the corresponding raw retained-path mass. Thus, under the shared tree measure, $\Delta_K^{\widetilde{P}} \le T_K \le U_K^{\text{exact}} \le U_K^{\text{Cantelli}}$. For either target, define $\bar{U}_K^\bullet = \max_{2\le k\le K} U_k^\bullet$. These monotone targets

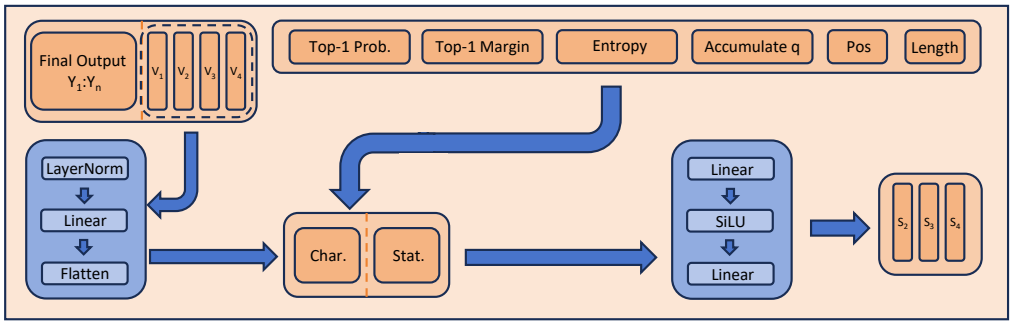

Figure 3: Lightweight controller. The projected inputs are pre-LM-head hidden states for the four candidate positions, rather than token IDs or vocabulary logits. The projection is shared across positions. Three separately trained instances implement direct, U-rank, and joint supervision; only the selected policy's head runs online.

retain the tree-law upper bounds. The raw retained mass provides a conservative adjustment under this law, whose predictive value is not implied by the bound. They do not certify full-vocabulary risk or the error on a particular greedy continuation. Appendix A specifies the measure and proofs. The enumerated exactU target supervises the U-rank and joint arms. The Cantelli construction is an analytical comparator, with no separately trained Cantelli controller evaluated here.

4.3 Lightweight Controllers and Acceptance

The online controller predicts a risk score without constructing a tree. Figure 3 shows its inputs. Four pre-LM-head hidden states are normalized, projected to 32 dimensions each, and flattened. We concatenate these 128 features with 18 scalar statistics, comprising four top-1 probabilities, four margins, four normalized entropies, four q values, generated offset, and prefix length. A $146 \to 96 \to 3$ SiLU MLP produces scores s_2, s_3, s_4.

We compare three supervision targets using the same architecture and initialization. **U-rank** learns the relative order of exactU targets on matched-confidence, cross-question pairs. **Direct** follows the qualitative control motivation: if deployment needs a risk ordering rather than an accurately calibrated U value, directly supervise greedy prefix disagreement using binary cross-entropy. **Joint** adds the two objectives. Writing $U_K = \bar{U}_K^{\text{exact}}$, for a non-tied pair (a, b) and candidate length K the ranking term is

$$\mathcal{L}_{\text{rank}} = \text{softplus}[-\text{sign}(U_K(b) - U_K(a))(s_K(b) - s_K(a))]\,. \tag{8}$$

All arms use confidence-matched pairs selected using greedy success and failure labels, so U-rank is not a label-free training arm. U supplies the pair ordering, including orderings discordant with greedy error. Appendix B details sampling and selection. The head scores need not be monotone across K. Their percentile cuts select actions without certifying risk.

Confidence first selects K_q using Equation 4. Low and high score cuts are calibrated at weighted 20th and 80th percentiles for each candidate length. Define $\mathcal{D}$ as the event that $K_q \geq 2$, $q_{K_q} \in [0.015, 0.05]$, and the current-length score is marked high. Define $\mathcal{U}$ as the event that $K_q < 4$, $q_{K_q+1} \in (0.05, 0.10]$, and the next-length score is marked low. The policy makes one adjustment,

$$K' = \begin{cases} K_q - 1, & \mathcal{D}, \\ K_q + 1, & \neg\mathcal{D} \text{ and } \mathcal{U}, \\ K_q, & \text{otherwise.} \end{cases} \tag{9}$$

Both events use the original confidence-selected K_q. Downward review has priority if both events hold, and the revised length is not reviewed recursively. The decoder emits the selected prefix, clipped at the first stop token and the remaining generation budget. This separates the eligibility bands from the head's high/low decision. The all-down variant removes only the confidence-band restriction from $\mathcal{D}$, retaining $K_q \geq 2$ and the high-score test. Upward eligibility is unchanged. No test-set policy search is used.

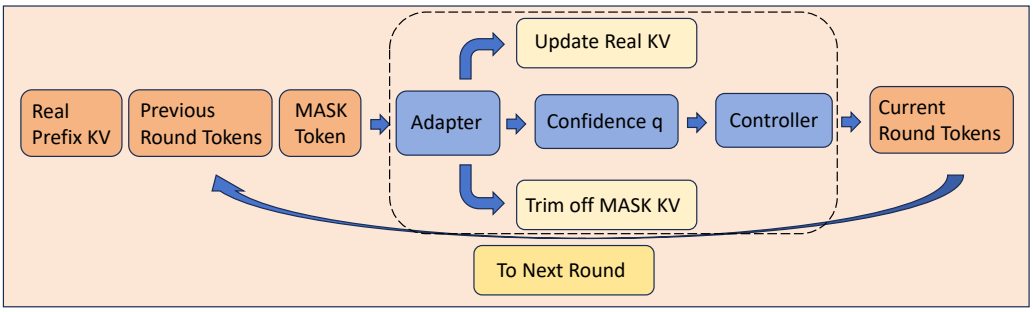

Figure 4: Merged cache schedule. Pending accepted IDs are processed as real inputs before the new MASK positions. The cache keeps only real-token states. Candidate IDs from the current proposal remain pending until the next round.

5 Executing Parallel Decoding with Real-Token History

5.1 Real-Token Recommit and Merged Proposals

A MASK's key/value state represents a placeholder computation, not the real token predicted from it. After accepting token IDs, we therefore recommit those IDs through the backbone before using them as history. This makes the cache represent the emitted sequence. The mechanism also supports a clean separation between proposal capacity and cache semantics: changing how many tokens are accepted does not authorize retaining their MASK-derived KV states.

Figure 4 depicts the steady-state schedule. At round t, the cache contains a committed real prefix h, and the previous round has emitted a pending suffix $a_{1:k}$. We run one causal forward on $[a_{1:k}, M_1, M_2, M_3, M_4]$ with that cache. The real positions extend committed history. The final real-token state predicts the next proposal's first token, and M_1–M_3 predict its remaining three tokens. Cropping the cache to $|h| + k$ removes every MASK KV entry. The newly accepted candidate IDs become the next pending suffix. Initial clean prefill and the first proposal form the request setup; subsequent rounds combine commit and proposal.

5.2 Execution Cost and Parallelism

For a long request, let ℓ_t be the number of tokens actually emitted after clipping in round t, and c_t its merged computation cost including selection. Excluding setup, throughput is $\sum_t \ell_t / \sum_t c_t = \bar{\ell}/\bar{c}$, where both averages are over rounds. Relative to serial cost c_{AR} per token, the approximate throughput ratio is $\bar{\ell} c_{\mathrm{AR}}/\bar{c}$. Multiple accepted tokens can amortize a more expensive proposal round. Context length, cache operations, and controller activation affect this cost. We therefore report dedicated timing separately from Mean K, retaining the original full vocabulary projection and excluding diagnostic collection from timed generation.

6 Experiments

6.1 Experimental Setup

We study Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct (Qwen et al., 2025; Grattafiori et al., 2024), with Qwen2.5-0.5B as a scale diagnostic. The adapter recipe is intentionally compact: 16,000 training questions and three epochs, with the backbone frozen throughout. Four benchmark subsets cover GSM8K (64), HumanEval (64), IFEval (32), and BBH (54, two per task) (Cobbe et al., 2021; Chen et al., 2021; Zhou et al., 2023; Suzgun et al., 2022). We compare HF, clean K1, merged K1, q-only, direct, U-rank, and joint under frozen settings. Appendix B gives hardware, splits, and scoring; all reported results are development screens with fixed protocols.

The controls answer different questions. HF versus clean K1 tests the bypass, while clean versus merged K1 tests the proposal execution path with single-token acceptance. Merged K1 versus q-only

describes the additional change with multi-token acceptance. For example, Llama IFEval scores 23, 25, and 26 correct under clean, merged, and q-only, respectively. Thus its three-question gain over clean is not entirely a difference from enabling multi-token acceptance. These are path comparisons, not an additive causal decomposition.

6.2 Parallel Capability Acquisition

Table 2 shows the central capability gain from a small frozen-backbone extension. Llama's aggregate future-position agreement rises from 0.76% to 61.39%, while Qwen reaches 57.36%, after changing only the MASK-local branch. The same architecture is trained separately for the two backbone families without modifying their base parameters. On Qwen's matched evaluation windows, the adapter reaches 57.36% versus 30.76% for the PaSS-style embedding baseline, a 26.60-point difference. These measurements isolate proposal learning before closed-loop acceptance. Aggregate agreement pools individual future positions. It does not measure the probability that all tokens in an accepted prefix are correct, because errors across positions can be dependent. Closed-loop task outcomes therefore provide a separate evaluation of the learned proposals.

Table 2: Validation local agreement (%). Each model uses 4,000 windows from 1,000 questions. p_j denotes position-j token agreement, not task accuracy. Gray parentheses show the percentage-point change from the Initial adapter within the same backbone. Aggregate pools valid positions 2–4.

Backbone	Proposal	p_2	p_3	p_4	Aggregate
Qwen7B	Initial adapter	6.04% (0.00)	3.25% (0.00)	2.38% (0.00)	3.90% (0.00)
	PaSS-style embedding	44.64% (+38.60)	27.55% (+24.30)	19.88% (+17.50)	30.76% (+26.86)
	Trained adapter	72.55% (+66.51)	56.36% (+53.11)	42.94% (+40.56)	57.36% (+53.46)
Llama8B	Initial adapter	0.94% (0.00)	0.31% (0.00)	1.01% (0.00)	0.76% (0.00)
	Trained adapter	74.92% (+73.98)	61.21% (+60.90)	47.86% (+46.85)	61.39% (+60.63)

6.3 Trainable Capacity

The adapter budgets are approximately 0.042% of Qwen2.5-7B and 0.05% of Llama3.1-8B. These counts measure trainable capacity, not total training computation. Teacher labeling, backbone computation, and offline controller-bank construction also incur costs that are not quantified here. The PaSS-style comparison uses matched evaluation windows, but does not isolate parameter count from capacity placement. Appendix H gives a descriptive comparison with published MARS results under distinct protocols.

6.4 Sequential Fidelity and Runtime Paths

Llama clean K1 reproduces all 214 HF sequences in the four-task screen. Same-shape audits also preserve real-token logits when toggling the MASK branch. Merged K1 reproduces 132/214 sequences, while its task scores remain close to HF (Table 3). Earlier Qwen7B runs reproduce 64/64 sequences through clean K1 and 37/64 through merged K1 on the GSM8K runtime screen. The structural bypass therefore provides an unchanged sequential option, while merged execution is evaluated as its own numerical path.

6.5 Closed-Loop Quality across Tasks

Table 3 reports every fixed strategy. Confidence-only decoding accepts 1.34–2.09 tokens per step on Llama. Its IFEval score changes from 23/32 to 26/32 and GSM8K from 49/64 to 47/64, while HumanEval changes from 54/64 to 36/64 and BBH from 41/54 to 27/54. Qwen's HumanEval U-rank arm gains eight net correct tasks relative to q-only, whereas Llama's gains one. These development-screen results describe task-dependent operating points, without establishing a general improvement from risk supervision.

Table 3: Seven strategies on fixed screens. IF/BBH/GSM/HE denote IFEval (32), BBH (54), GSM8K (64), and HumanEval (64). Each cell shows accuracy (%), change from clean K1 in percentage points, and Mean K. HumanEval uses the uniformly corrected builtins scorer.

	Qwen2.5-7B				*Llama3.1-8B*			
Policy	IF	BBH	GSM	HE	IF	BBH	GSM	HE
HF K1	75.0% (0.0) $K = 1.00$	46.3% (0.0) $K = 1.00$	90.6% (0.0) $K = 1.00$	92.2% (0.0) $K = 1.00$	71.9% (0.0) $K = 1.00$	75.9% (0.0) $K = 1.00$	76.6% (0.0) $K = 1.00$	84.4% (0.0) $K = 1.00$
Clean K1	75.0% (0.0) $K = 1.00$	46.3% (0.0) $K = 1.00$	90.6% (0.0) $K = 1.00$	92.2% (0.0) $K = 1.00$	71.9% (0.0) $K = 1.00$	75.9% (0.0) $K = 1.00$	76.6% (0.0) $K = 1.00$	84.4% (0.0) $K = 1.00$
Merged K1	71.9% (-3.1) $K = 1.00$	46.3% (0.0) $K = 1.00$	89.1% (-1.6) $K = 1.00$	92.2% (0.0) $K = 1.00$	78.1% (+6.3) $K = 1.00$	72.2% (-3.7) $K = 1.00$	78.1% (+1.6) $K = 1.00$	84.4% (0.0) $K = 1.00$
q-only	75.0% (0.0) $K = 1.19$	40.7% (-5.6) $K = 1.56$	85.9% (-4.7) $K = 1.72$	62.5% (-29.7) $K = 1.97$	81.3% (+9.4) $K = 1.34$	50.0% (-25.9) $K = 1.93$	73.4% (-3.1) $K = 1.69$	56.3% (-28.1) $K = 2.09$
Direct	75.0% (0.0) $K = 1.16$	46.3% (0.0) $K = 1.52$	79.7% (-10.9) $K = 1.70$	70.3% (-21.9) $K = 1.93$	78.1% (+6.3) $K = 1.33$	48.1% (-27.8) $K = 1.95$	73.4% (-3.1) $K = 1.69$	59.4% (-25.0) $K = 2.03$
U-rank	75.0% (0.0) $K = 1.16$	42.6% (-3.7) $K = 1.55$	82.8% (-7.8) $K = 1.71$	75.0% (-17.2) $K = 1.91$	71.9% (0.0) $K = 1.37$	51.9% (-24.1) $K = 1.97$	70.3% (-6.3) $K = 1.70$	57.8% (-26.6) $K = 2.07$
Joint	75.0% (0.0) $K = 1.16$	42.6% (-3.7) $K = 1.52$	78.1% (-12.5) $K = 1.69$	70.3% (-21.9) $K = 1.93$	75.0% (+3.1) $K = 1.35$	50.0% (-25.9) $K = 1.96$	73.4% (-3.1) $K = 1.69$	57.8% (-26.6) $K = 2.06$

6.6 Understanding Controller Decisions

The supervision objectives lead to different decisions, with task-dependent outcomes. Qwen HumanEval changes from 40/64 under q-only to 48/64 under U-rank and 45/64 under direct. On Llama, direct reaches 38/64 compared with 36/64 for q-only and 37/64 for U-rank. These comparisons do not isolate the information contributed by U from pair selection or changes in acceptance length. Same-prefix evaluation and label-permutation controls would be needed for that attribution.

Saved BBH traces show that U-rank can shorten a confidence-selected K4 block to K3 or extend a K2 block to K3. These records demonstrate policy actions, rather than independently verified corrections (Appendix F).

Token-aligned Llama cases expose both successful intervention and coverage. In HumanEval-51, q-only accepts a K2 block spelling `not vowels`, omitting `in`. At the same saved prefix and block start, direct and joint reduce K to one, emit `not`, and subsequently generate `in vowels`; both pass the test. U-rank keeps K2 and produces invalid code. In HumanEval-63, the block `):\n ->` has $q_2 = 0.01039$, outside downward review, and all heads leave it unchanged. In HumanEval-43, a repeated `seen` lies inside a reviewed K4 block but all heads keep K4. These examples separate the information learned by the head from the range of blocks it is allowed to review. Full same-prefix counterfactual replay is a targeted follow-up, rather than an inference from final answer changes alone.

The earlier all-down comparison holds weights and cuts fixed while expanding only downward eligibility. On Qwen0.5B, direct changes from 24/64 to 26/64; on Qwen7B it remains 43/64. The experiment establishes that review scope changes realized actions without guaranteeing the same quality effect across scales. Appendix F records these comparisons and the paired Llama outcomes.

Paired outcomes and review coverage. The Llama HumanEval pairs distinguish recovered answers from new errors. Relative to q-only, direct recovers six tasks but loses four, yielding the observed net gain of two. U-rank recovers two and loses one, yielding a net gain of one. Against HF, q-only loses 18 correct answers and gains none. On BBH, it loses 15 HF-correct answers and gains one. Thus net accuracy alone hides substantial differences in which examples change, and a controller intervention is not uniformly beneficial.

Static parsing provides a complementary failure diagnostic. All 15 q-only HumanEval outputs classified as invalid code fail parsing of the extracted first code block. Their first divergence from merged

Table 4: Llama q-only quality and measured speed on the same development tasks. Throughput and latency ratios use HF K1. Ratios below one mean lower throughput or shorter latency, respectively.

Task	HF accuracy	q-only accuracy	Throughput ratio	Latency ratio
GSM8K	49/64	47/64	$0.926\times$	$0.972\times$
HumanEval	54/64	36/64	$1.140\times$	$0.921\times$
IFEval	23/32	26/32	$0.738\times$	$1.286\times$
BBH	41/54	27/54	$1.060\times$	$0.955\times$

K1 falls at proposal positions two, three, and four in 10, three, and two cases, respectively. Five of these first-divergence blocks are ineligible for downward review. These observations distinguish first-divergence blocks outside the review scope from those eligible for score-based review. They do not identify the syntax-error location or establish that shortening the first divergent block would repair the answer. Appendix F gives the trace-alignment details.

6.7 End-to-End Efficiency

The Qwen7B GSM8K runtime screen increases throughput from 45.39 to 54.58 tokens/s at Mean K 1.7169; mean request latency decreases from 6.242 to 5.128 seconds. Accuracy changes from 58/64 to 55/64 on the same screen. These measurements demonstrate that frozen-backbone proposals can amortize their execution cost, while explicitly retaining the corresponding quality point.

For Llama, we time all 214 requests under seven policies with three repetitions on an A100-SXM4-40GB. Timing includes prefill, decoding, acceptance control, and real-token commit. All 4,494 measured outputs match their previously scored sequences for the same policy. Table 4 pairs q-only speed with its task accuracy, using HF K1 as the speed reference. On HumanEval and BBH, throughput increases accompany substantial accuracy changes. The complete seven-policy timing table is in Appendix D.

Each policy is warmed before measurement, requests and policies are order-randomized, and CUDA is synchronized around the complete generation loop. The interval includes proposal construction, acceptance, recommit, and cache maintenance, but excludes scoring and trace replay. Mean K is not a speed guarantee. Because policies can generate different output lengths, mean request latency also reflects the amount of generated text. These measurements report fixed operating points rather than an equal-quality speed frontier.

When acceptance amortizes proposal cost. The merged K1 control exposes the cost of running the proposal path while accepting one token. Across the four Llama tasks, its throughput is 0.549–0.553 times HF K1 throughput. The q-only throughput is approximately 1.69, 2.07, 1.34, and 1.92 times that of merged K1 on GSM8K, HumanEval, IFEval, and BBH. These ratios are close to the corresponding Mean K values of 1.69, 2.09, 1.34, and 1.93. The comparison helps explain why multi-token acceptance improves throughput relative to merged K1 on every task, yet exceeds HF throughput only on HumanEval and BBH. Because context lengths and output trajectories differ across policies, this is a descriptive amortization pattern, not a fixed per-round cost estimate or a hardware-level attribution.

7 Conclusion

MASK-local adapters provide a small trainable extension for future-token prediction on frozen autoregressive models. The two-backbone study shows substantial local agreement gains, including a matched-window improvement over an embedding baseline on Qwen. A separate no-MASK path preserves sequential decoding, while real-token recommit supports approximate parallel execution. The acceptance and timing studies characterize how this learned capability translates into task-dependent quality and speed. They motivate further matched-budget proposal comparisons and independent evaluation of risk-guided acceptance.

AI USE STATEMENT

During the preparation of this work, the authors used AI assistance ChatGPT solely to generate boilerplate and utility code for the experimental pipelines, and to refine the manuscript's text, phrasing, and rhetorical transitions. The AI was not involved in formulating the core research questions, designing the underlying methodologies, conducting the physical experiments, or interpreting the experimental findings. All generated code was independently audited, debugged, and verified through internal unit tests and execution logs by the authors. The human authors reviewed, edited, and approved all scientific claims, textual revisions, and code implementations, and remain entirely responsible for the integrity and final content of this paper.

REPRODUCIBILITY STATEMENT

Appendix B records the available adapter, controller, and scoring settings, and Appendix A gives definitions and proofs. The LaTeX source and figure PDFs accompany this draft. The current draft does not include the source data manifest, complete per-example outputs, or executable training and evaluation pipelines. Appendix G identifies the missing artifacts needed for independent reproduction.

ETHICS STATEMENT

This study uses existing model checkpoints and benchmark data. Approximate parallel acceptance changes outputs and should be evaluated for the intended application before deployment. Generated code is untrusted: the resource-limited host evaluator used here is not a security sandbox. Model and dataset licenses remain applicable to reuse and redistribution.

REFERENCES

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads, 2024. URL https://arxiv.org/abs/2401.10774.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, et al. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Ziqi Jin, Lei Wang, Ziwei Luo, and Aixin Sun. Mars: Enabling autoregressive models multi-token generation, 2026. URL https://arxiv.org/abs/2604.07023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Giovanni Monea, Armand Joulin, and Edouard Grave. Pass: Parallel speculative sampling, 2023. URL https://arxiv.org/abs/2311.13581.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, et al. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

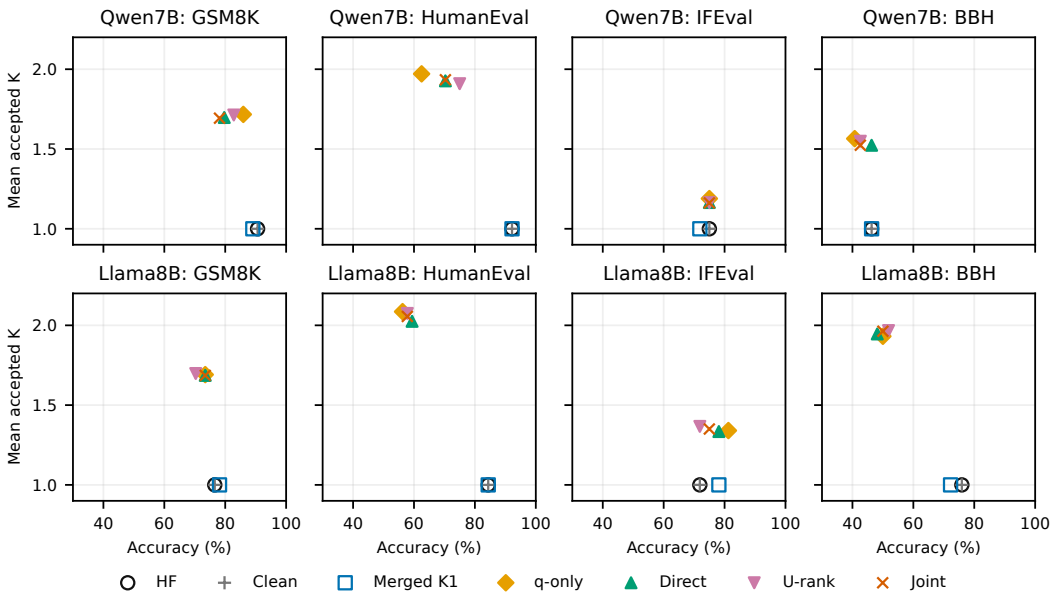

Figure 5: Quality–parallelism points for all seven strategies. Horizontal axes show subset accuracy and vertical axes show Mean K. Points are neither interpolated nor jittered; coincident policies overlay. Each point summarizes the fixed tasks, not an independent training replicate. These plots accompany the exact tables rather than imply an uncertainty interval.

A Mathematical Definitions and Bounds

A.1 Decision gaps and disagreement

Fix the proposal tokens and a history law ν. If a conditional greedy token differs from a_j, some competing token has probability at least as large as $p_\theta(a_j \mid h, x_{<j})$. Consequently, a disagreement at any extension position implies $G_K^{\max} \geq 0$. This implication handles deterministic ties by including all ties in the upper event. It need not be an equality because a tie can still be resolved in favor of a_j. Taking expectations proves the first inequality in Equation 6. The first token is an anchor and is not included in the extension-block event. Direct-label training excludes contexts with a first-token mismatch and masks positions beyond EOS.

A.2 Cantelli construction

Let $Z = G_K^{\max} - \mu_K$, so $\mathbb{E}Z = 0$ and $\mathbb{E}Z^2 = \sigma_K^2$. When $\mu_K < 0$, set $t = -\mu_K > 0$. For any $b \geq 0$, on $Z \geq t$ we have $(Z+b)^2 \geq (t+b)^2$. Markov's inequality gives $\mathbb{P}(Z \geq t) \leq (\sigma_K^2 + b^2)/(t+b)^2$. Choosing $b = \sigma_K^2/t$ yields $\sigma_K^2/(\sigma_K^2 + t^2)$. If $\mu_K \geq 0$, we use the trivial upper bound one. If variance is zero and mean is negative, the tail is zero. This proves the stated Cantelli bound without a distributional assumption beyond finite variance.

A.3 Tree law and exactU

At every expanded prefix, let $S_8(x)$ be the eight highest-ranked tokens under the same deterministic tie-breaking rule and $c(x) = \sum_{v \in S_8(x)} p_\theta(v \mid x)$. The tree transition is $\widetilde{p}(v \mid x) = p_\theta(v \mid x)/c(x)$ for $v \in S_8(x)$ and zero otherwise. Products of these transitions define $\widetilde{P}$. Define $\widetilde{G}_j$ by replacing p_θ with $\widetilde{p}$ in the definition of G_j. A proposal token omitted from S_8 has induced probability zero and a retained top token has positive probability. The original and induced distributions have the same top-ranked decision. Thus disagreement implies $\widetilde{G}_K^{\max} \geq 0$, and enumeration gives T_K for this event. In general $\widetilde{G}_j \neq G_j$, so $\widetilde{C}_K$ must be computed from the moments of $\widetilde{G}_K^{\max}$ under $\widetilde{P}$, not from the original gaps.

For fixed $R \in [0,1]$, the map $f_R(t) = Rt + 1 - R$ is increasing and satisfies $f_R(t) \geq t$ for $t \in [0,1]$. Therefore $\Delta_K^{\widetilde{P}} \leq T_K \leq f_{R_K}(T_K) \leq f_{R_K}(\widetilde{C}_K)$. Running maxima preserve these inequalities. In the current bank, R_K is computed from raw Top-8 path probabilities through the corresponding levels, whereas T_K uses locally renormalized transitions. Local normalization is not generally the same as conditioning the full history law on remaining inside the tree. Thus Equation 7 specifies the implemented tail-adjusted training target and bounds the stated tree disagreement; it is not, by itself, a proof of a full-vocabulary probability certificate.

If a full-distribution certificate is desired, the retained contribution must instead be evaluated using the original raw history masses. For a retained-history set $\mathcal{T}$, a valid form is $\sum_{x \in \mathcal{T}} P(x)\mathbf{1}[G_K^{\max}(x) \geq 0] + P(X \notin \mathcal{T})$, using gaps under the original conditionals. This distinction is a measure specification, not a learned-controller property: neither tree enumeration nor a moment bound transfers its upper-bound status automatically to an MLP score.

B Training and Evaluation Details

B.1 Adapter construction and data

The Qwen7B branch has 28 decoder blocks and width 3,584; Llama8B has 32 blocks and width 4,096. Each block has bias-free down/up matrices of rank 16 with zero-initialized up weights. Three learned role vectors distinguish active MASK positions. The parent MASK embedding is fixed. Backbone parameters, output embeddings, and the LM head remain frozen.

Both main runs use the same 16,000 source questions and three epochs. Text contexts are tokenized separately for each backbone, and teacher labels are generated by that backbone. Qwen's realized schedule has 59,024 windows and 5,535 optimizer updates; Llama's has 57,837 windows and 5,424 updates. Calibration and validation each use 1,000 questions and 4,000 windows. Windows use distinct offsets $[0, \min(8, \lfloor n/3 \rfloor), \lfloor n/2 \rfloor, n]$ after the reference context is limited to the prefix budget. Short Llama references are retained when those offsets remain distinct. The Llama maximum prefix length is 508, leaving four MASK positions in the training input. Llama EOS handling includes IDs 128001, 128008, and 128009.

The Llama training configuration uses learning rate 10^{-4}, a 3% warmup fraction, effective batch size 32, microbatch size 2, gradient clipping at one, and no weight decay. It uses BF16 SDPA without gradient checkpointing. Hardware is an NVIDIA A100-SXM4-40GB with PyTorch 2.5.1+cu121, Transformers 4.57.0, NumPy 2.2.6, and Python 3.10.13. Complete Qwen and 0.5B configurations are not included in the current draft and cannot be inferred from this Llama configuration.

B.2 Native controller bank and fitting

The Llama bank has 96 training, 32 selection, and 96 calibration questions, yielding 572, 189, and 572 contexts. Each rollout samples up to two uniform and four boundary contexts with deduplication and inverse inclusion weights. Full trees are built for training and selection contexts (761 total); calibration uses greedy labels and score distributions rather than full trees. Training and selection contain 340 and 34 matched pairs under the frozen pair rule.

All heads share LayerNorm, a $d \to 32$ projection applied to four states, and the $146 \to 96 \to 3$ MLP. Statistics are standardized using a weighted training mean and standard deviation. Pair construction matches K, base selection, 64-token offset bucket, and q within 0.005, uses different questions, and selects up to four partners per anchor. Pairs contrast matching and mismatching greedy prefixes; U differences below 0.02 are ranking ties. Hence the U-rank arm is not trained on a label-free pair population.

Each head uses seed 1701, AdamW at 3×10^{-4} with weight decay 0.01, 20 epochs, and 40 updates per epoch. Each update draws 64 pairs and 64 natural examples. Direct supervision combines 80% hard-endpoint BCE and 20% weighted natural BCE. Joint adds ranking with coefficient one. All arms select the epoch with highest selection-set safety-pair concordance, breaking exact ties toward the earliest epoch. Calibration cuts use only the original calibration population, weighted 20/80 quantiles, and valid positions within the q band. Minimum support is 16 rows, eight questions, and question effective sample size six. No held-out task accuracy is used to tune these cuts.

B.3 Benchmark scoring and populations

We use 64 GSM8K items, the first 64 HumanEval tasks, 32 IFEval prompts, and two items from each of 27 BBH tasks. These are fixed development subsets; prior use of the GSM8K subset is not treated as a sealed generalization test. Generation budgets are 1,024 tokens for GSM8K, HumanEval, and BBH, and 1,280 for IFEval. Tokenization is native to each model's chat template. Generation is greedy; stop-token handling follows the model contract.

GSM8K uses frozen numeric extraction including a last-number fallback. HumanEval extracts the first Python code block (or uses body completion when appropriate) and executes the task's `check(candidate)` with the corrected standard computational builtins. The uniform correction leaves original outputs unchanged. IFEval uses strict prompt-level instruction adherence. BBH uses frozen strict exact match, with a separately reported format-secondary metric. The HumanEval worker runs in a resource-limited process, not a security sandbox. For Llama, all GSM8K arms have zero matches under the separate explicit-answer extractor; primary numeric accuracy should therefore not be relabeled as explicit-answer accuracy.

C Additional Quantitative Results

Table 5 complements the main accuracy comparison with the realized acceptance lengths of the same runs.

Table 5: Mean accepted K for all strategies in the four-task quality screens.

	Qwen7B				Llama8B			
Policy	GSM	HE	IF	BBH	GSM	HE	IF	BBH
HF / clean / merged K1	1	1	1	1	1	1	1	1
q-only	1.7169	1.9710	1.1892	1.5649	1.6911	2.0859	1.3406	1.9313
Direct	1.6975	1.9272	1.1646	1.5234	1.6863	2.0252	1.3343	1.9461
U-rank	1.7130	1.9093	1.1636	1.5486	1.6979	2.0746	1.3665	1.9670
Joint	1.6932	1.9340	1.1622	1.5237	1.6863	2.0557	1.3502	1.9629

D Runtime Timing

Table 6 reports the dedicated Llama timing screen. Each cell reports throughput (tokens/s) / mean request latency (s). Throughput is the median of three repeat-level aggregate rates; latency averages all measured requests. The largest within-policy, within-dataset throughput range across repetitions is 0.135% of its median.

Table 6: Dedicated Llama timing.

Policy	GSM8K	HumanEval	IFEval	BBH
HF K1	40.72 / 4.721	40.55 / 2.853	40.72 / 10.280	40.32 / 6.214
Clean K1	40.82 / 4.709	40.66 / 2.845	40.82 / 10.255	40.45 / 6.193
Merged K1	22.36 / 8.145	22.31 / 5.250	22.42 / 18.573	22.29 / 10.648
q-only	37.69 / 4.591	46.21 / 2.628	30.04 / 13.223	42.72 / 5.937
Direct	37.36 / 4.750	44.56 / 2.789	29.75 / 13.821	42.78 / 6.137
U-rank	37.62 / 4.727	45.63 / 2.601	30.47 / 13.583	43.25 / 6.176
Joint	37.36 / 4.801	45.23 / 2.677	30.11 / 13.595	43.14 / 6.091

E Cache and Numerical Diagnostics

The same-shape audit compares real-token KV and first-candidate logits under identical input shapes with future interventions. Llama passes this audit for pending lengths one through four. The cross-shape clean-history comparisons record maximum logit differences of 0.40625, 0.3125, 0.25, and

0.3125 in the saved audit; the first comparison changes argmax. These observations motivate separate HF, clean K1, and merged K1 controls. They do not establish that all numerical differences arise from a single kernel operation. The production decoder crops every MASK KV entry and checks committed history before each round.

F Controller Cases and Scope

The saved U-rank traces provide concrete examples of both directions. In a saved BBH down-review record, the controller marks a proposed $K = 4$ block as current-length high risk and selects $K' = 3$; its confidence vector is $(0, 0.00079, 0.02742, 0.02742)$. This records a block that U-rank scores as high risk and shortens, without independently establishing that the original block was incorrect. Conversely, in a saved BBH up-review record, a $K = 2$ block has next-position confidence $q_3 = 0.06573$ and is labeled next-length low risk; U-rank selects $K' = 3$ rather than stopping at two tokens. These traces demonstrate that the head implements both risk review and opportunity exploration; they are mechanism examples, not standalone correctness certificates.

In Llama HumanEval, q-only changes 18 previously correct HF answers to incorrect answers and changes none in the opposite direction. Relative to q-only, direct recovers six tasks and loses four, while U-rank recovers two and loses one. On BBH, q-only loses 15 HF-correct items and gains one. The BBH format-secondary correct counts are 43, 43, 40, 29, 29, 30, and 31 in the seven-policy order, so the observed difference is not explained solely by strict answer formatting.

All 15 q-only HumanEval outputs labeled invalid code fail static parsing of the first code block itself. Their first divergence from merged K1 occurs at positions two, three, and four of a proposal in 10, three, and two cases. Five of these first-divergence blocks are not eligible for downward review. First trajectory divergence is not necessarily the syntax-error location: HumanEval-51 first diverges earlier, but the missing `in` occurs in the later block described in the main text. Native-tokenizer decoding reproduces all 448 saved HumanEval texts exactly, supporting token-to-block alignment.

The all-down study on a separate 64-item GSM8K cohort keeps heads and quantile cuts fixed. Qwen0.5B direct/U-rank/joint change from 24/21/24 correct under boundary review to 26/22/25 under all-down. Qwen7B changes from 43/46/43 to 43/46/44. Q-only scores are 24 and 45, respectively. These results are not merged with the four-task screen because their cohorts and runtime settings differ.

G Scope and Reproducibility Gaps

All task results are fixed development screens, and prior use of the GSM8K subset prevents treating it as a sealed generalization test. We report point estimates without uncertainty intervals or multiple training seeds. The first 64 HumanEval tasks and two items per BBH task do not establish full-benchmark performance. Existing task outcomes do not isolate the independent contribution of tree supervision or establish an equal-quality speed advantage over verified decoding.

Independent reproduction requires the 16k-question source dataset and version, split and deduplication manifests, benchmark item IDs and exact prompts, the parent MASK vector and role initialization, complete Qwen/0.5B and PaSS-style training configurations, controller-bank provenance and rollout policies, the tree-root conditioning convention and retained-mass depth/EOS rules, Mean K aggregation and stop-clipping conventions, and the fallback when calibration support is insufficient. Executable scoring code, per-example outputs, and resource measurements for labeling, adapter fitting, and tree-bank construction are also not included in this draft. These omissions are not filled by transferring the reported Llama settings to another run.

H PUBLISHED MARS RESULTS IN CONTEXT

MARS uses continued full-model training, whereas our main runs freeze the backbone and train the stated adapter budget. Trainable parameter counts do not compare total training cost. Table 7 reproduces selected published MARS $B = 4$, $\tau = 0.95$ results and places individual q-only policies beside them. Each change uses its own method's sequential reference. Different checkpoint histories, cohorts, prompts, and scorers remain confounders after this subtraction. These rows are descriptive context, not a controlled comparison or evidence of a scaling law.

Table 7: Cross-protocol context. Cells show accuracy (%), change from each method's sequential reference, and Mean K. MARS numbers are from its Table 4 (Jin et al., 2026). Our 0.5B screen is separate from the main 7B/8B evaluation. Parameter counts for our rows include adapters and role vectors, with no controller for q-only.

Model / policy	Trainable	IFEval	BBH	GSM8K	HumanEval
MARS Qwen 0.5B	full model	45.5% (-5.8) $K = 1.20$	26.6% (0.0) $K = 1.90$	31.1% (-1.7) $K = 1.51$	39.6% (-0.6) $K = 1.26$
Ours Qwen 0.5B / 4-controller mean	n/r	37.5% (+3.1) $K = 1.22$	7.14% (-0.5) $K = 1.18$	40.6% (-12.5) $K = 1.28$	44.3% (-20.1) $K = 1.35$
MARS Qwen 7B	full model	63.0% (-5.2) $K = 1.15$	54.3% (-0.3) $K = 2.60$	71.0% (-2.2) $K = 1.79$	80.5% (-1.2) $K = 1.60$
Ours Qwen 7B / q	3.22M	75.0% (0.0) $K = 1.19$	40.7% (-5.6) $K = 1.56$	85.9% (-4.7) $K = 1.72$	62.5% (-29.7) $K = 1.97$
Ours Llama 8B / q	4.21M	81.3% (+9.4) $K = 1.34$	50.0% (-25.9) $K = 1.93$	73.4% (-3.1) $K = 1.69$	56.3% (-28.1) $K = 2.09$

The main Qwen7B q-only GSM8K result is 55/64 (85.9%) against clean K1 at 58/64 (90.6%). The separate all-down cohort in Appendix F has q-only at 45/64 (70.3%). Its result is not combined with the main screen's baseline or interpreted as a matched scale comparison.