PACE: Pattern-Aware Multi-Action Control for LLM Agent Overthinking

Anonymous ACL submission

Abstract

LLM agents often exhibit overthinking that inflates token cost without proportional task gains. Existing controllers typically address this with a single mechanism: trajectory-control methods truncate via binary stop decisions, while process reward models score steps with multi-billion-parameter verifiers. Neither matches the structure of the problem, in which overthinking takes multiple distinguishable forms that call for different corrective actions. We characterize agent overthinking through four such patterns (context inflation, failure loops, reasoning redundancy, and silent failure) and propose **PACE** (**P**attern-**A**ware **C**ontrol of **E**xecution), a lightweight controller ($\sim$68K parameters) that dispatches one of four typed interventions (COMPRESS, REDIRECT, MODESWITCH, STOP) or CONTINUE from percentile-normalized trajectory signals. The percentile representation absorbs raw-magnitude differences across backbones, allowing a single checkpoint trained on Qwen3-4B-Thinking with HotpotQA to transfer zero-shot to four multi-hop search-QA benchmarks and three thinking-mode LLMs spanning two orders of magnitude in active-parameter scale. PACE preserves or improves EM on 10 of 12 deployment cells while consistently reducing tokens, outperforms four intervention baselines on the in-domain setting by 5.4 to 14.3 pp EM, and its multi-action design alone contributes +6.6 pp over a stop-only variant.

1 Introduction

Large language model (LLM) agents that reason over multiple steps, chaining search queries, invoking tools, and planning deliberatively, build on chain-of-thought and reasoning-acting paradigms (Wei et al., 2022; Yao et al., 2022). As these agents are deployed on harder tasks, they increasingly spend their inference budget on reasoning that does not improve task outcomes. We call this *overthinking*: the agent inflates its reasoning context beyond what a task requires, retries failing tool calls in loops, oscillates between previously considered conclusions, or terminates with a confidently wrong answer. Recent work characterizes overthinking as a pervasive efficiency bottleneck (Chen et al., 2025; Cuadron et al., 2025; Aggarwal et al., 2025), yet largely stops at characterization without a deployable corrective mechanism.

Two families of methods attempt such intervention, and both fall short of overthinking's multi-pattern structure. *Trajectory-control* methods such as T^3 (Zou et al., 2026) truncate RL training trajectories at belief-deviation onset, and inference-time *early-exit* methods (Lu et al., 2025) learn a binary halt-or-continue decision; both are lightweight but offer no recourse beyond termination. *Process reward models* (PRMs) (Lightman et al., 2024) score intermediate steps to guide search, but require model-specific training (typically billions of parameters) and reduce overthinking to a single scalar rather than acting on its distinct failure modes. Neither family demonstrates zero-shot transfer across both LLM backbones and task families.

We argue that overthinking in agentic settings is not a single phenomenon but four distinguishable patterns, each with an observable signature and a different corrective action: *context inflation* (over-long thoughts and ballooning context), *failure loops* (repeated tool errors), *reasoning redundancy* (oscillation between previously considered conclusions), and *silent failure* (confidently wrong terminations). The patterns map onto distinct interventions: a single stop decision cannot break a failure loop, and scalar compression cannot resolve oscillation. This taxonomy motivates **PACE** (Pattern-Aware Control of Execution), a lightweight controller ($\sim$68K parameters) that, at each reasoning step, dispatches one of four typed interventions (COMPRESS, REDIRECT, MODESWITCH, or STOP), or CONTINUE, from five percentile-normalized trajectory signals. The percentile repre-

sentation absorbs raw-magnitude differences across LLMs, allowing a single checkpoint to transfer zero-shot across two orders of magnitude in model scale.

Contributions. (1) Four-pattern taxonomy of agent overthinking. We identify four distinguishable patterns (context inflation, failure loops, reasoning redundancy, silent failure) from 12,262 trajectories spanning four benchmarks and three thinking-mode LLM backbones. Mann-Whitney U tests on trajectory signals confirm that the patterns are statistically separable on the underlying signals under Bonferroni correction (§3). **(2) PACE: a pattern-aware multi-action controller.** A lightweight dual-head design dispatches four typed interventions, or CONTINUE, from percentile-normalized trajectory signals. Percentile normalization absorbs cross-backbone magnitude differences, and the dual-head decomposition exposes the gate threshold τ as a post-training tuning knob without retraining (§4). **(3) Consistent zero-shot transfer across deployment shifts.** Trained on a single configuration, PACE preserves or improves EM on 10 of 12 deployment cells across four search-QA benchmarks and three thinking-mode LLMs spanning two orders of magnitude in active-parameter scale. Against four intervention baselines on the in-domain setting, PACE wins by 5.4 to 14.3 pp EM, and its multi-action dispatch alone contributes +6.6 pp over a stop-only variant (§5).

2 Related Work

Overthinking in reasoning LLMs and agents has been documented as redundant deliberation (Chen et al., 2025), inter-rollout disagreement (Sethi, 2026), divergent reasoning styles (Shahariar et al., 2025), confidence-related redundancy (Qiao et al., 2025), and trajectory-level failure taxonomies (Cuadron et al., 2025; Yang et al., 2026; Aggarwal et al., 2025). Post-hoc reflection (Shinn et al., 2023) addresses failures after termination. These works largely *characterize* overthinking; deploying real-time corrective mechanisms in multi-step agents remains open.

Process supervision and inference-time control. Process reward models (PRMs) supervise intermediate steps as step-level rewards (Lightman et al., 2024), and recent work uses PRM-derived signals as inference-time control: Bilal et al. (2026) dynamically allocate compute across reasoning tools, and Qiao et al. (2025) use a confidence detector for early termination in single-generation reasoning. These methods commit to a single intervention mechanism (scalar reward or termination). We adopt the same structural slot with two changes: percentile-normalized signals replace learned scalar rewards, enabling zero-shot transfer across backbones, and a typed action dispatch replaces continuous scores, matching the multi-pattern structure (§3).

Adaptive allocation, stopping, and compression. Test-time compute can be traded for accuracy (Snell et al., 2024), motivating difficulty-aware budgeting (Zhai et al., 2026) and budget-aware tool invocation (Liu et al., 2025; Paglieri et al., 2025). Early-stopping methods terminate via answer entropy (Laaouach, 2025) or output convergence (Mao et al., 2025; Nagle et al., 2026); Lu et al. (2025) extends binary halting to multi-step embodied agents, and Zou et al. (2026) truncates belief-deviated rollouts at training time. Compression methods prune within a generation (ConCISE (Qiao et al., 2025), Word Salad Chopper (Xie et al., 2025)) or at the prompt/cache level (Jiang et al., 2023; Pan et al., 2024; Xu et al., 2023; Xiao et al., 2024); a parallel line analyzes reasoning structure (Bogdan et al., 2025; Lee et al., 2025; Venhoff et al., 2025). These approaches share a *single-mechanism* assumption (budget, stop, or compress), which is insufficient for agent overthinking, where context-inflated trajectories need rewriting, failure loops need reformulation, and oscillating deliberation needs mode switching. This motivates our typed multi-action design.

3 Overthinking Patterns in LLM Agents

Prior work treats overthinking as a single failure mode requiring termination (Chen et al., 2025; Cuadron et al., 2025). In agentic settings, however, overthinking takes multiple distinguishable forms, each with its own observable signature. We identify four such patterns from 12,262 agent trajectories collected across multiple benchmarks and LLM backbones; pattern labels are derived by rule-based thresholding and validated by blind double annotation on a stratified 92-step sample (collection and annotation details in Appendix A).

Type A: Context Inflation. The agent accumulates a growing reasoning history in which new

thoughts rehash prior steps, yielding monotonically increasing thought length with no gain in success rate. Type A motivates COMPRESS.

Type B: Failure Loop. The agent issues structurally similar tool calls that return empty, error, or irrelevant results, retrying with minor surface variations rather than reformulating the task. Captured by elevated consecutive failure counts (≥ 3 vs. ≤ 1 in successes), Type B motivates REDIRECT.

Type C: Reasoning Redundancy. Intermediate decisions oscillate between previously considered sub-conclusions without converging. Unlike Type B, which oscillates on tool inputs, Type C oscillates on reasoning content. Particularly prevalent in thinking-mode models, it motivates MODESWITCH.

Type D: Silent Failure. Reasoning appears procedurally fluent but the final answer is incorrect. Unlike Types B and C, which have direct behavioral signatures, Type D manifests *statistically*: inflated mean and variance of thought length without the convergence cues of productive deliberation, motivating a conservative STOP.

Reliability. We validate the taxonomy at two levels. *Statistically*, Mann-Whitney U tests on the five trajectory signals (§4) confirm that the four patterns are statistically separable under Bonferroni at $p<0.01$. *Conceptually*, blind double annotation on a stratified 92-step sample (Appendix A) produced labels consistent with the rule-based first-pass after a calibration round clarifying the Type B vs Type C operational distinction. The controller is trained on rule-based labels rather than human consensus; deployment results (§5) provide the empirical utility test.

4 Methodology

We propose **PACE** (**P**attern-**A**ware **C**ontrol of **E**xecution), a learned dispatcher over fixed primitives. A small Transformer-based controller decides *when to intervene* and *which pattern is active*, while the interventions themselves are rule-based prompt rewrites. At each reasoning step, PACE processes a sliding window of recent steps and dispatches one of four typed interventions (COMPRESS, REDIRECT, MODESWITCH, or STOP), or CONTINUE (Figure 1). Three design principles motivate the architecture: *LLM-agnostic adaptation* via percentile normalization, *pattern-aware dispatch* via a typed intervention space matching the four patterns, and *post-training tunability* via a dual-head decomposition that decouples intervention timing from intervention type.

Trajectory signals. At each step t, we compute five prefix-level signals over a sliding window of size W (set to $W{=}5$ in all experiments), where $|\ell_i|$ is the assistant thought length at step i, osc_i indicates a self-correction marker ("wait," "actually," "however"), and fail_i indicates a tool-call failure.

Mean thought length, the average token count of thinking steps in the window, is the primary indicator of context inflation (Type A):

$$\mu_{\text{thought}}(t) = \tfrac{1}{W} \sum_{i=t-W+1}^{t} |\ell_i| \tag{1}$$

Thought variance measures fluctuation in step length. When elevated together with μ_{thought}, it forms the statistical signature of silent failure (Type D):

$$v_{\text{thought}}(t) = \tfrac{1}{W} \sum_{i=t-W+1}^{t} \big(|\ell_i| - \mu_{\text{thought}}(t)\big)^2 \tag{2}$$

Oscillation rate, the fraction of steps containing self-correction markers, targets reasoning-content oscillation (Type C):

$$\omega_{\text{osc}}(t) = \tfrac{1}{W} \sum_{i=t-W+1}^{t} \mathbb{1}[\text{osc}_i] \tag{3}$$

Consecutive failure count, the length of the current unbroken tool-call failure streak, targets failure loops (Type B):

$$c_{\text{fail}}(t) = \max\big\{k \geq 0 \,:\, \textstyle\prod_{j=0}^{k-1} \mathbb{1}[\text{fail}_{t-j}] = 1\big\} \tag{4}$$

Current thought length, the instantaneous token count of the present step, supplements μ_{thought} as a real-time context cost estimate (Type A):

$$\ell_{\text{cur}}(t) = |\ell_t| \tag{5}$$

The signal-to-pattern mapping is many-to-many: $\{\mu_{\text{thought}}, v_{\text{thought}}, \ell_{\text{cur}}\}$ target Type A; c_{fail} targets Type B; ω_{osc} targets Type C; and inflated $\{\mu_{\text{thought}}, v_{\text{thought}}\}$ without elevated $\{\omega_{\text{osc}}, c_{\text{fail}}\}$ characterize Type D.

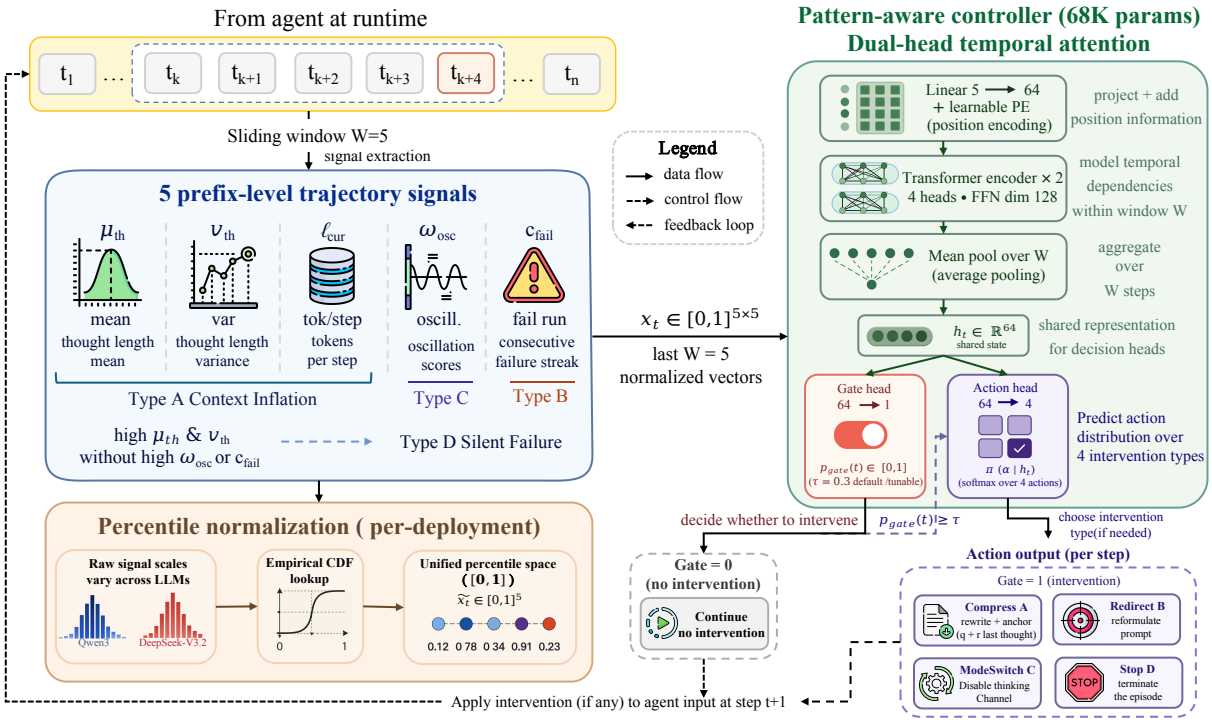

Figure 1: Pattern-Aware Controller. Five trajectory signals are percentile-normalized over a sliding window and dispatched to one of five typed actions via a dual-head decision module.

LLM-agnostic percentile normalization. We normalize each signal $s(t)$ against a per-deployment reference distribution $\mathcal{R}_{(L,b)}$ for LLM L on benchmark b via its empirical CDF (Eq. 6):

$$\tilde{s}(t) = \text{Percentile}\big(s(t);\, \mathcal{R}_{(L,b)}\big) = \frac{\big|\{r \in \mathcal{R}_{(L,b)} \mid r \leq s(t)\}\big|}{|\mathcal{R}_{(L,b)}|} \in [0,1]. \quad (6)$$

The reference is precomputed once per deployment from a modest sample of unseen trajectories, with no parameter updates. Since the empirical CDF is invariant under any strictly monotonic re-scaling of raw signal magnitudes, $\tilde{s}(t)$ shares the same distribution at the deployment ranks regardless of backbone scale. This provides the mathematical basis for the zero-shot transfer we observe across LLMs of widely varying scale (Table 1; reference distribution sizes in Appendix B.4).

Dual-head architecture. A linear projection lifts each step's normalized signals into a fixed-size embedding, summed with a learnable positional encoding over the sliding window. A lightweight Transformer encoder with mean pooling produces a state $\mathbf{h}_t$, and two linear heads operate in parallel. **Gate head**: outputs the probability of intervention via a sigmoid, controlling *when* to act:

$$p_{\text{gate}}(t) = \sigma(\mathbf{w}_g^\top \mathbf{h}_t + b_g) \quad (7)$$

Action head: produces a distribution over the four typed interventions, controlling *which* action to take (Δ^4 is the simplex over {COMPRESS, REDIRECT, MODESWITCH, STOP}):

$$\mathbf{p}_{\text{act}}(t) = \text{softmax}(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a) \in \Delta^4 \quad (8)$$

Dispatch rule: gates the action head with a threshold τ. Below the threshold, the trajectory continues unmodified:

$$a(t) = \begin{cases} \arg\max_k \, p_{\text{act}}^{(k)}(t) & \text{if } p_{\text{gate}}(t) > \tau, \\ \text{CONTINUE} & \text{otherwise.} \end{cases} \quad (9)$$

By decoupling *when* to intervene (gate) from *which* action (dispatch), the threshold τ becomes a post-training tuning knob that re-balances the EM-tokens trade-off without modifying any learned parameter; we use τ=0.3 as the default across all deployments and analyze its effect in §5.3. A single-head 5-way classifier cannot support this decomposition (Appendix B.2).

Action implementations. Each typed intervention is realized as a deterministic prompt rewrite at the conversation level. COMPRESS (Type A) replaces the conversation history with a compact pruned version retaining the user question, all past search queries, the retrieved tool results, and the

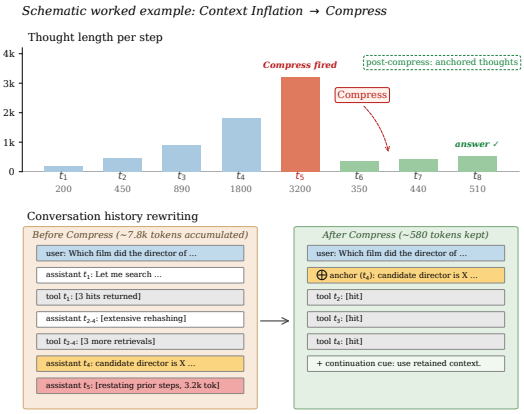

Figure 2: Schematic worked example of the Context Inflation to COMPRESS pattern. Token counts and content are stylized; quantitative results in Table 1.

most recent assistant thought as a *reasoning anchor* (illustrated in Figure 2); alternative templates are ablated in §5.4. REDIRECT (Type B) appends a reformulation prompt to the history. MODESWITCH (Type C) disables the model's thinking channel for thinking-capable LLMs. STOP (Type D) terminates the trajectory, applied conservatively because Type D's statistical signature is weaker than the direct behavioral signatures of Types B and C. Detailed implementations are in Appendix B.3.

Training objective. The controller is trained on trajectories from a single backbone-benchmark configuration. Each step receives a gate label based on trajectory outcome and high percentile-normalized signal levels, and an action label determined by a fixed priority over the patterns. We jointly minimize binary cross-entropy on the gate and cross-entropy on the action head; the action loss is computed only on intervention steps. Full labeling rules, split definitions, optimizer settings, and checkpoint selection are reported in Appendix B.

5 Experiments

We organize the experimental evaluation around three claims, one per subsection (§5.2–§5.4). All experiments use the same controller checkpoint trained only on HotpotQA-741 + Qwen3-4B-Thinking trajectories; no benchmark- or backbone-specific tuning is performed at deployment beyond a one-time reference-distribution precomputation (Appendix B.4).

5.1 Experimental Setup

Implementation details. The controller has 68,101 parameters in total. The input projection lifts each step's 5 normalized signals into a 64-dim embedding, summed with a learnable positional encoding over a window of size $W=5$. A two-layer Transformer encoder (4 heads, per-head dim 16, FFN dim 128) with mean pooling produces the state $\mathbf{h}_t$; two linear heads of size $64\to1$ (gate) and $64\to4$ (action) provide the dispatch decision. Unless otherwise noted, deployment uses gate threshold $\tau=0.3$ uniformly across all backbones and benchmarks. Full architecture and training hyperparameters are in Appendix B.

Benchmarks. We evaluate on four multi-hop search-QA benchmarks spanning 2-hop, 2–4 hop, and harder compositional reasoning. **HotpotQA** (Yang et al., 2018): full test split (N=7,405) and a stratified N=741 trajectory-level subsample (seed 42) used for paired-bootstrap evaluation; HotpotQA-741 is the in-domain setting (controller trained only on HotpotQA + Qwen3-4B-Thinking trajectories). **Bamboogle** (Press et al., 2023): N=125 (complete set; 2-hop hand-crafted hard examples). **2WikiMultiHopQA** (Ho et al., 2020): N=500 (seed-42 random subsample from test split; 2–4 hop comparison/compositional). **MuSiQue** (Trivedi et al., 2022): N=500 (seed-42 random subsample from dev split, as the test split has hidden leaderboard answers; 2–4 hop reasoning with disconnected sub-questions, the hardest of the four). All benchmarks beyond HotpotQA-741 are zero-shot transfer targets; the same random-sampling protocol (seed 42, N=500 where applicable) is applied across benchmarks for paired-bootstrap comparability.

LLM backbones. We evaluate on three thinking-mode LLM backbones spanning two model families and roughly 167× in scale. **Qwen3-4B-Thinking** (Yang et al., 2025) (open-source, 4B dense; the controller's training source). **Qwen3-30B-A3B-Thinking** (Yang et al., 2025) (open-source, 30B-A3B Mixture-of-Experts with 3B active parameters; 8× scale within the same family). **DeepSeek-V3.2-Thinking** (Liu et al., 2024) (closed-source API; 671B-A37B MoE; 167× scale, different family). We also report DeepSeek-V3.2 in non-thinking mode for completeness (Appendix D.1). The controller is trained *only* on Qwen3-4B-Thinking trajectories; deployment on the other two backbones is zero-shot via percentile normalization (§4).

Evaluation metrics. EM is the primary metric for all four QA benchmarks, following community convention (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). Comparisons use 95% paired bootstrap CIs (10,000 resamples). F1 and LLM-as-judge accuracy appear in Appendix D.1. Token-cost metrics are reported as the average total tokens consumed per question (input plus output across all LLM calls during the agent's trajectory).

Baselines. (i) *No Controller*, (ii) *Random* (p=0.1, per-trajectory seeded), (iii) *Token Budget* (cap=18,900, the baseline average), (iv) *Step-Count Cap* (K=6, the baseline median step count), and (v) *Stop-Only* (action space {CONTINUE, STOP} (Lu et al., 2025), sharing our gate checkpoint). Controller-based baselines (Stop-Only and ours) share the same checkpoint; full implementation details in Appendices B.4 and C.

5.2 Main Results

On HotpotQA with Qwen3-4B-Thinking (in-domain), our controller outperforms all four baselines with substantial margins (Table 1). Notably, every single-mechanism baseline (random intervention, hard token caps, hard step caps, and Stop-Only halting) *degrades EM* below the no-controller baseline, confirming that multi-pattern overthinking cannot be addressed by a single intervention type or by uniform truncation policies. The Stop-Only comparison (Δ=+6.61 pp vs. ours) directly attributes our gain to the *multi-action design*: both methods share the same controller, percentile pipeline, and gate threshold τ=0.3, and differ only in whether COMPRESS, REDIRECT, and MODESWITCH are enabled. The Step-Count Cap comparison (Δ=+14.30 pp) rules out the alternative explanation that simple step truncation could match our gains: forcing termination at K=6 (the baseline median) catastrophically loses −11.74 pp EM, indicating that early steps in the trajectory carry essential information that the multi-action controller selectively preserves. Held-out step-level test accuracy is 93.12% overall with 81.63% gate F1 (Appendix B.2); the in-pool/out-of-pool ΔEM gap on the 7,405-trajectory test split is within bootstrap noise, ruling out trajectory memorization (Appendix A.4).

5.3 Pareto-Dominant Zero-Shot Transfer

We evaluate the controller zero-shot along three orthogonal axes of deployment shift: **(i) cross-benchmark** transfer (HotpotQA to Bamboogle, 2WikiMultiHopQA, MuSiQue); **(ii) within-family scaling** (8×, Qwen3-4B to Qwen3-30B-A3B); and **(iii) cross-family transfer** (167×, Qwen3 to DeepSeek-V3.2). Table 1 reports the consolidated 3 LLMs × 4 benchmarks deployment matrix; the controlled Stop-Only comparison on the in-domain cell (§5.2) isolates the contribution of multi-action dispatch over single-mechanism halting.

Cross-benchmark transfer on Qwen3-4B. On the training-source backbone, the controller preserves or improves EM on all four benchmarks while saving tokens on each: HotpotQA-741 (+2.56 pp / −12.4%), 2WikiMultiHopQA (+0.80 pp / −12.2%), MuSiQue (+0.20 pp / −13.6%), and Bamboogle (EM-tied / −3.7%). The Bamboogle result is consistent with its already-short baseline trajectories (5.77 avg. steps vs. 6.03–6.40 on the other three), which leaves less room for compression. The unweighted-mean Average (ΔEM=+0.89 pp, CR 89%) is Pareto-optimal: a positive EM movement combined with substantial token reduction.

Scaling: 8× within-family and 167× cross-family transfer. Deploying the same checkpoint zero-shot at the unified τ=0.3, the controller achieves Pareto-optimal unweighted-mean performance on *all three* LLM backbones: ΔEM = +0.89 pp / CR 89% on Qwen3-4B (in-domain), +0.92 pp / 98% on Qwen3-30B-A3B (8× scale, 3B active MoE), and +0.19 pp / 91% on DeepSeek-V3.2-Thinking (167× scale, 37B active MoE, different family). At the individual cell level, EM is preserved or improved on 10 of 12 cells; the two cells with EM loss (Qwen3-30B MuSiQue −1.80 pp, DeepSeek Bamboogle −1.60 pp) trade EM for token saving, while Qwen3-30B Bamboogle shows the inverse trade-off (+4.80 pp EM at +8% tokens). Per-cell effect sizes on cross-backbone transfer are small (typically $<$ 2 pp), but the consistent direction of effect across 12 cells, combined with robust token-saving evidence (CR $\leq$ 89% on the two out-of-family LLM backbones' Average), supports the controller's transferability via percentile normalization (§4).

τ as a post-training tunability knob. The dual-head decomposition exposes τ as a post-training tuning knob, accommodating deployment-specific EM-tokens trade-offs without retraining. The sweep on HotpotQA-741 (Qwen3-4B; Figure 3)

Method	HotpotQA-741		Bamboogle		2WikiMHQA		MuSiQue	
	EM↑	ΔTok (%)↓	EM↑	ΔTok (%)↓	EM↑	ΔTok (%)↓	EM↑	ΔTok (%)↓
Qwen3-4B-Thinking: in-domain training source								
Baseline	39.81	—	56.00	—	60.20	—	16.00	—
Random (p=0.1)	36.97	−24.9	44.80	−22.1	52.20	−28.2	15.40	−30.3
Token Budget	36.57	−17.2	50.40	−13.2	54.00	−17.7	11.00	−28.3
Step-Count Cap (K=6)	28.07	−26.6	43.20	−24.8	41.40	−31.3	9.40	−30.4
Stop-Only (Lu et al., 2025)	35.76	−16.8	50.40	−13.9	52.40	−21.5	15.40	−19.8
PACE τ=0.3	**42.37**	**−12.4**	56.00	**−3.7**	**61.00**	**−12.2**	**16.20**	**−13.6**
Qwen3-30B-A3B-Thinking: 8× scale within family								
Baseline	41.16	—	56.00	—	63.00	—	20.80	—
Random (p=0.1)	38.46	−7.9	51.20	−11.2	55.00	−9.8	21.80	−16.6
Token Budget	40.21	−7.5	53.60	+5.7	59.40	−4.0	19.40	−17.8
Step-Count Cap (K=6)	40.08	−10.5	57.60	−9.1	59.00	−8.1	18.80	−12.4
Stop-Only (Lu et al., 2025)	41.02	−5.1	52.80	−3.3	59.40	−10.1	20.00	−13.0
PACE τ=0.3	**41.83**	**−0.3**	**60.80**	+8.0	63.00	**−10.4**	19.00	**−4.7**
DeepSeek-V3.2-Thinking: 167× scale, different family								
Baseline	46.55	—	67.20	—	62.80	—	24.40	—
Random (p=0.1)	44.80	−6.5	61.60	−3.5	55.60	+6.8	22.00	−1.7
Token Budget	48.44	−11.5	70.40	+7.5	62.60	−0.2	23.60	−17.2
Step-Count Cap (K=6)	43.58	−26.9	66.40	−19.4	58.60	−9.4	19.60	−30.4
Stop-Only (Lu et al., 2025)	40.89	−31.8	62.40	−21.3	57.60	−12.3	24.00	−26.7
PACE τ=0.3	**48.31**	**−13.0**	65.60	**−8.2**	**63.20**	**−3.1**	**24.60**	**−13.8**

Table 1: Zero-shot transfer of a unified τ=0.3 PACE checkpoint across 3 thinking-mode LLMs × 4 benchmarks, vs. four intervention baselines. Token Budget (cap = 18,900) is calibrated for Qwen3-4B; on smaller-trajectory backbones the cap may not bind. EM and ΔTok are both in percent. **Bold** (green): PACE > Baseline; underline (red): PACE < Baseline. Acc/F1/steps in App. D.1.

shows the trade-off explicitly: τ=0.3 is Pareto-dominant (+2.56 pp / −12.4%), τ=0.5 preserves EM at lower token cost (+0.40 pp / −11.1%), and τ=0.7 trades EM for further token reduction. Calibration uses a one-time ∼100-trajectory sweep on *unlabeled* trajectories, a zero-training-cost analogue of post-hoc temperature calibration.

5.4 Ablations

Per-action contribution. To attribute the controller's gain to individual typed actions, we mask each action one at a time, forcing its decisions to CONTINUE while keeping the others active (Table 2; Figure 4). On Qwen3-4B-Thinking at τ=0.3, every mask incurs EM degradation on average, with the ranking REDIRECT (−2.89 pp) > MODESWITCH (−1.75 pp) ≈ COMPRESS (−1.68 pp) > STOP (−0.12 pp). STOP's small average reflects its conservative trigger design (Type D's statistical signature is weaker than B/C's; §4); its contribution concentrates on HotpotQA (−1.75 pp). Figure 4(b) reveals a *two-dimensional* contribution structure: masking COMPRESS or MODESWITCH *raises* average token cost (+5.6%, +4.4%), while masking

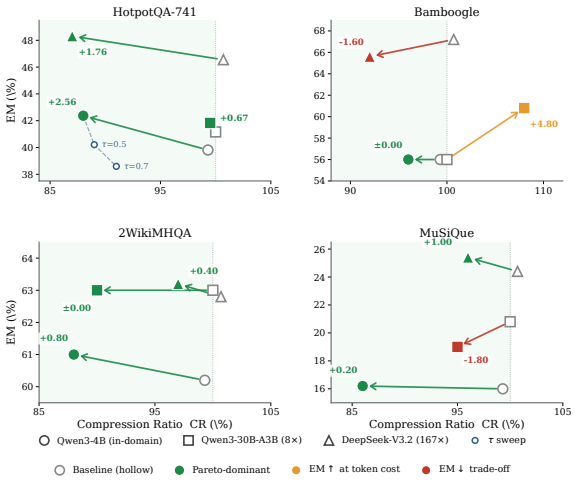

Figure 3: Pareto trade-offs across four QA benchmarks. Green markers denote Pareto-dominant settings that improve the EM-compression trade-off over the baseline (hollow markers), while several strong baselines remain Pareto-dominated.

REDIRECT *lowers* it (−6.5%). REDIRECT is thus the EM driver but a token consumer, whereas COMPRESS is the token saver; neither is discardable

without sacrificing one efficiency dimension. Consistently, the Stop-Only ablation (Table 1, −6.61 pp EM) shows that collapsing the entire action space incurs a larger penalty than any single mask.

Configuration	Hotpot		Bamb.		2Wiki	
	EM↑	ΔTok↓	EM↑	ΔTok↓	EM↑	ΔTok↓
Full PACE	**42.37**	—	**56.00**	—	**61.00**	—
mask one action (→ CONTINUE):						
COMPRESS	39.94	+10.1	56.00	−0.3	58.40	+6.9
REDIRECT	36.70	−11.5	51.20	−2.8	62.80	−5.2
MODESWITCH	39.13	+4.2	55.20	+7.2	59.80	+1.8
STOP	40.62	−0.8	56.00	+3.5	62.40	−1.1

Table 2: Per-action mask ablation on Qwen3-4B-Thinking (τ=0.3, V3 COMPRESS template). ΔTok (%): token change vs. Full PACE. Red: EM drop $\geq$ 4.5 pp vs. Full PACE.

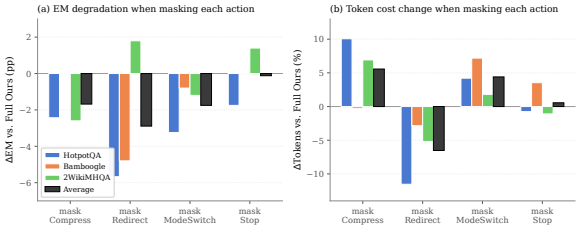

Figure 4: Per-action contribution on Qwen3-4B-Thinking (τ=0.3): (a) EM loss when each action is masked; (b) extra tokens (K) incurred when each action is masked. Taller bar means larger contribution. Dashed segments: cross-benchmark mean.

COMPRESS template variants. The COMPRESS action requires choosing what content to retain when pruning the conversation history. We ablate four prompt-rewrite templates that differ in the retained history subset: **V1** = results + last_thought (drops the user question); **V2** = queries + results (drops the last assistant thought); **V3** = queries + results + last_thought (deployed); **V4** = last 3 rounds + last_thought. We evaluate all four across three benchmarks at τ=0.3 on Qwen3-4B-Thinking (Table 3). All four variants are Pareto-dominant on in-domain HotpotQA-741 (EM gains +0.67–+2.56 pp at −10 to −13% tokens), but they differ sharply on the transfer benchmarks. V1 loses −5.60 pp EM on Bamboogle, justifying question preservation. V2 loses −1.60 pp on 2WikiMHQA; V4 loses −2.20 pp on 2WikiMHQA. Only the deployed **V3** achieves Pareto-dominance *across all three benchmarks* and attains the highest in-domain EM (42.37).

Variant	Bamb.		2Wiki		Hotpot	
	EM↑	ΔTok↓	EM↑	ΔTok↓	EM↑	ΔTok↓
Baseline	56.00	—	60.20	—	39.81	—
V1	50.40	−7.9	60.80	−12.7	41.43	−13.5
V2	56.00	−5.5	58.60	−13.7	40.48	−10.1
V3	**56.00**	**−3.7**	**61.00**	**−12.2**	**42.37**	**−12.4**
V4	56.00	−9.6	58.00	−16.5	41.02	−10.8

Table 3: COMPRESS template ablation (Qwen3-4B-Thinking, τ=0.3); see prose for V1–V4 definitions. ΔTok (%): token change vs. Baseline. Green row: V3 (deployed) is Pareto-dominant across all three benchmarks; red cell: EM loss $\geq$ 1 pp vs. Baseline.

Dual-head vs single-head architecture. The dual-head decomposition (∼68K parameters) achieves 93.12% overall step-level accuracy and 81.63% gate F1 on the held-out test split, versus 79.46% and 62.61% for a single-head 5-way classifier with strictly more parameters per head (+13.66 pp; Appendix B.2). The decomposition also enables the τ-tuning analysis above.

6 Conclusion

We characterized agent overthinking in multi-hop search-QA as four distinguishable patterns (context inflation, failure loops, reasoning redundancy, and silent failure) and proposed **PACE**, a lightweight pattern-aware controller that dispatches typed interventions from percentile-normalized trajectory signals. Trained on a single configuration, PACE transfers zero-shot across benchmarks and thinking-mode LLM backbones spanning two orders of magnitude in scale, attaining Pareto-optimal mean performance on every backbone while substantially outperforming single-mechanism intervention baselines.

These results suggest that the multi-pattern structure of agent overthinking admits lightweight, transferable control without large process reward models or mechanism-specific retraining, and that percentile normalization paired with typed-action dispatch is a promising recipe for deploying small learned controllers across heterogeneous reasoning agents.

Limitations

Task and backbone scope. Our evaluation is restricted to multi-hop search-QA on thinking-mode LLMs. The controller's sliding window requires trajectories of sufficient length, and preliminary

experiments on non-thinking backbones produced trajectories below this length. Generalization to other reasoning-intensive domains (tool-calling dialogues, embodied planning, code generation) and to non-thinking backbones is an open direction.

Effect size and evaluation breadth. Per-cell EM gains on out-of-domain backbones are modest (typically < 2 pp), with the controller's value manifesting as *consistent direction of effect* across 10 of 12 deployment cells combined with robust token savings, rather than large per-cell EM jumps. Full-scale evaluation on commercial-API backbones was constrained by inference budget; tightening per-cell estimates on larger evaluation sets and characterizing when multi-action dispatch yields large vs. marginal gains is left to future work.

Threshold calibration. We deploy a unified τ=0.3 across all backbones, but τ admits optional per-deployment calibration when practitioners want to trade EM for further token savings. Automatic τ-selection from intrinsic trajectory statistics is a natural extension.

Ethical Considerations

Use of AI assistants. We used AI assistants (specifically, large language models) in a limited and clearly bounded capacity during this work. Their use was restricted to: (i) language polishing of author-written drafts (grammar, phrasing, and readability), and (ii) auxiliary coding support such as boilerplate generation, debugging suggestions, and LaTeX formatting. All research ideas, experimental designs, claims, analyses, and conclusions are the authors' own. AI-generated text was reviewed and edited by the authors, and AI-generated code was manually verified before being incorporated into the experimental pipeline.

References

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. Optimalthinkingbench: Evaluating over and underthinking in llms. *arXiv preprint arXiv:2508.13141*.

Ahsan Bilal, Ahmed Mohsin, Muhammad Umer, Ali Subhan, Hassan Rizwan, Ayesha Mohsin, and Dean Hougen. 2026. What if we allocate test-time compute adaptively? *arXiv preprint arXiv:2602.01070*.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2025. Do not think that much for 2+ 3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, and 1 others. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 13358–13376.

Yassir Laaouach. 2025. Halt-cot: Model-agnostic early stopping for chain-of-thought reasoning via answer entropy. In *4th Muslims in ML Workshop co-located with ICML 2025*.

Jinu Lee, Sagnik Mukherjee, Dilek Hakkani-Tur, and Julia Hockenmaier. 2025. Reasoningflow: Semantic structure of complex reasoning traces. *arXiv preprint arXiv:2506.02532*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Tengxiao Liu, Zifeng Wang, Jin Miao, I Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, and 1 others. 2025. Budget-aware tool-use enables effective agent scaling. *arXiv preprint arXiv:2511.17006*.

Qingyu Lu, Liang Ding, Siyi Cao, Xuebo Liu, Kanjian Zhang, Jinxia Zhang, and Dacheng Tao. 2025. Runaway is ashamed, but helpful: On the early-exit behavior of large language model-based agents in embodied environments. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24014–24027.

Minjia Mao, Bowen Yin, Yu Zhu, and Xiao Fang. 2025. Early stopping chain-of-thoughts in large language models. *arXiv preprint arXiv:2509.14004*.

Alliot Nagle, Jakhongir Saydaliev, Dhia Garbaya, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. 2026. Terminator: Learning optimal exit points for early stopping in chain-of-thought reasoning. *arXiv preprint arXiv:2603.12529*.

Davide Paglieri, Bartłomiej Cupiał, Jonathan Cook, Ulyana Piterbarg, Jens Tuyls, Edward Grefenstette, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. 2025. Learning when to plan: Efficiently allocating test-time compute for llm agents. *arXiv preprint arXiv:2509.03581*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, and 1 others. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8021–8040.

Khushal Sethi. 2026. Don't overthink it: Inter-rollout action agreement as a free adaptive-compute signal for llm agents. *arXiv preprint arXiv:2604.08369*.

GM Shahariar, Ali Nazari, Erfan Shayegani, and Nael Abu-Ghazaleh. 2025. Modeling hierarchical thinking in large reasoning models. *arXiv preprint arXiv:2510.22437*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. 2025. Understanding reasoning in thinking language models via steering vectors. *arXiv preprint arXiv:2506.18167*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

Wenya Xie, Shaochen Zhong, Hoang Anh Duy Le, Zhaozhuo Xu, Jianwen Xie, and Zirui Liu. 2025. Word salad chopper: Reasoning models waste a ton of decoding budget on useless repetitions, self-knowingly. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33576–33586.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. 2026. Ares: Adaptive reasoning effort selection for efficient llm agents. *arXiv preprint arXiv:2603.07915*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhiyuan Zhai, Bingcong Li, Bingnan Xiao, Ming Li, and Xin Wang. 2026. Adaptive test-time compute allocation for reasoning llms via constrained policy optimization. *arXiv preprint arXiv:2604.14853*.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Haochen Yang, Mufei Li, James Cheng, Pan Li, and Yu Gong. 2026. Reducing belief deviation in reinforcement learning for active reasoning. In *The Fourteenth International Conference on Learning Representations*.

A Pattern Taxonomy and Trajectory Data

A.1 Trajectory Collection

We collected 12,262 complete agent trajectories spanning four benchmarks and three thinking-mode LLM backbones (Table 4). These trajectories underlie the four-pattern taxonomy (§3) and the controller's training data (§4).

Benchmark	Traj.	Avg. Steps
Qwen3-4B-Thinking		
HotpotQA	7,405	6.03
Bamboogle	125	5.77
2WikiMHQA	500	6.40
MuSiQue	500	6.32
Qwen3-30B-A3B-Thinking		
HotpotQA	741	3.43
Bamboogle	125	3.67
2WikiMHQA	500	3.69
MuSiQue	500	3.63
DeepSeek-V3.2-Thinking		
HotpotQA	741	4.44
Bamboogle	125	3.88
2WikiMHQA	500	4.12
MuSiQue	500	5.45
Total	**12,262**	—

Table 4: Trajectory pool: agent rollouts across 3 thinking-mode LLM backbones × 4 multi-hop search-QA benchmarks, used for taxonomy validation (§3) and controller training (Qwen3-4B-Thinking + HotpotQA only; §4).

A.2 Pattern-to-Action Rules

Pattern labels are derived by a deterministic rule-based first-pass over percentile-normalized signals (§4). A step receives gate$=1$ if and only if the trajectory is unsuccessful *and* at least one of $\{\tilde{\mu}_{\text{thought}}, \tilde{v}_{\text{thought}}, \tilde{\ell}_{\text{cur}}\}$ exceeds the 90th percentile. Conditional on gate$=1$, the action is determined by priority:

1. $\tilde{\omega}_{\text{osc}} > 0.75 \Rightarrow$ MODESWITCH (Type C).
2. $\tilde{\mu}_{\text{thought}} > 0.90 \wedge \tilde{v}_{\text{thought}} > 0.85 \Rightarrow$ STOP (Type D).
3. Raw $c_{\text{fail}} \geq 5 \Rightarrow$ REDIRECT (Type B).
4. Otherwise $\Rightarrow$ COMPRESS (Type A, the soft-fallback).

The resulting intervention rate is 20.81%, with COMPRESS as the majority class (68.12%), MODESWITCH 24.07%, STOP 7.38%, and REDIRECT 0.44%.

A.3 Annotation Protocol

We validate the four-pattern taxonomy (§3) through a blind double-annotation protocol on a stratified sample of 92 reasoning steps. Candidate steps are drawn from the training pool, stratified by the rule-based first-pass label using COMPRESS→type_a, REDIRECT→type_b, MODESWITCH→type_c, STOP→type_d. Target counts were 25 per type; we used 25/17/25/25 because the type_b candidate pool only contained 17 steps.

Each sample exposes the target step plus 6–8 surrounding steps, the question, gold answer, model final answer, and a correctness flag. All rule-based first-pass labels are removed before delivery (verified by automated leakage check). Two annotators independently labeled each sample as one of type_a, type_b, type_c, type_d, or none, choosing the most salient pattern when multiple applied. After the first round, the operational distinction between Type B (tool-input oscillation) and Type C (reasoning-content oscillation) was clarified, and both annotators independently revisited their labels. We report annotator-level statistics and per-pattern agreement in the supplementary materials.

Design choices that bound the protocol. Two design choices constrain how the protocol should be interpreted. First, candidate trajectories per pattern type are stratified by rule-based labels, so the protocol estimates inter-annotator consistency on *candidate* instances rather than on uniformly sampled trajectories. Second, Type D (Silent Failure) is by definition conditioned on final-answer correctness, which is exposed to annotators during labeling; agreement on Type D should therefore be interpreted as agreement on the residual pattern judgment given shared correctness information.

Reliability bounds. Two design choices bound the interpretation: (i) candidates are stratified by rule-based labels (not uniform sampling), so κ reflects consistency on flagged trajectories; (ii) Type D conditions on final-answer correctness, exposed to annotators during blind labeling.

A.4 Anti-Leakage Analysis

To rule out trajectory-level memorization, we partition the 7,405 HotpotQA test trajectories into 5,658 *in-pool* (originated from the controller training pool, seed-42 stratified into 4,538/566/554 for train/val/offline-test, trajectory-disjoint) and 1,747 *out-of-pool* (withheld from every training stage).

The in-pool/out-of-pool ΔEM gap is $\sim$0.2 pp, well within bootstrap noise. The same memorization test trivially holds on Qwen3-30B-A3B-Thinking and DeepSeek-V3.2 deployments, since their trajectories never enter training.

N=741 **subsample composition.** A seed-42 stratified sample of the 7,405 split: 554 in-pool, 187 out-of-pool. The subsample preserves the in-pool/out-of-pool ΔEM relationship within bootstrap noise. We use N=741 for cross-LLM comparisons because running the full split on commercial APIs is cost-prohibitive.

Output validity. Controller intervention does not introduce invalid (empty, non-parseable) outputs. Across all 12 deployment cells, the controller produces parseable outputs for $\geq 99\%$ of trajectories; the few invalid outputs are matched between baseline and controller runs (the controller flips zero baseline-valid outputs to invalid).

B Controller Implementation

B.1 Architecture and Hyperparameters

The controller (Table 5) is a 68,101-parameter dual-head module. A linear input projection lifts 5-dim signals to 64-dim embeddings, summed with a learnable positional encoding over a window of size W=5. A two-layer Transformer encoder (4 heads, per-head dim 16, FFN dim 128) with mean pooling produces a state $\mathbf{h}_t$. Two linear heads (gate 64→1, action 64→4) output the dispatch decision (Eq. 9).

Component	Params
Positional encoding (5×64)	320
Input projection ($5 \rightarrow 64$)	384
Transformer encoder (2L, 4H, ff128)	66,944
Post-attention LayerNorm	128
Gate head ($64 \rightarrow 1$)	65
Action head ($64 \rightarrow 4$)	260
Total	**68,101**

Table 5: Controller parameter breakdown.

Training data. We train on HotpotQA + Qwen3-4B-Thinking with trajectory-level seed-42 splits (4,538/566/554), yielding 14,791/1,907/1,874 step entries after sliding-window expansion.

Training procedure. Optimizer Adam (β_1=0.9, β_2=0.999, ε=10^{-8}); learning rate 10^{-3}; batch size 256 step entries; up to 100 epochs with patience-10 early stopping on validation accuracy; dropout 0.1. Loss: $\mathcal{L} = \text{BCE}(p_{\text{gate}}, y_{\text{gate}}) + \text{CE}(\mathbf{p}_{\text{act}}, y_{\text{act}}; \text{ignore}{=}-100)$, with the action loss computed only on intervention steps. We train two loss configurations (with/without inverse-frequency class weighting) $\times$ 3 seeds $\in \{42, 43, 44\}$, selecting by composite validation score $0.45 \cdot \text{gate-F1} + 0.35 \cdot \text{acc} + 0.20 \cdot$ STOP-precision. Single NVIDIA H20 GPU, $\sim$2 min per run (early-stopped at epoch $\sim$50–70). The selected checkpoint is validated via a brief 100-question smoke test on Bamboogle+HotpotQA before full deployment.

B.2 Dual-Head vs Single-Head Ablation

The dual-head decomposition (68,101 params) achieves 93.12% overall step-level accuracy and 81.63% gate F1 versus 79.46% / 62.61% for a single-head 5-way classifier (34,757 params), a +13.66 pp gain. Beyond accuracy, the dual-head design enables post-training τ threshold tuning without retraining the action distribution.

B.3 Action Implementations

COMPRESS. Replaces conversation history with: (i) the user question, (ii) all past search queries, (iii) retrieved tool results, (iv) the most recent assistant message (*reasoning anchor*), and (v) a short continuation cue. This is variant V3, deployed as the empirically Pareto-dominant choice across benchmarks (Table 8).

REDIRECT. Appends a reformulation prompt asking the agent to restate the goal in different terms before issuing the next tool call. The full conversation context is retained.

MODESWITCH. For thinking-mode LLMs (Qwen3-Thinking, DeepSeek-V3.2 with thinking enabled), disables the model's thinking channel for the remainder of the trajectory, forcing concise execution. Targets Type C (reasoning-content oscillation).

STOP. Terminates the trajectory and extracts a final answer: first searches the last thought for a \boxed{·} pattern; absent that, uses the last sentence. Applied conservatively because Type D's statistical signature is weaker than B/C's behavioral signatures.

B.4 Reference Distribution Computation

For each deployment configuration, the reference distribution $\mathcal{R}_{(L,b)}$ is computed from a sample of unseen trajectories (Table 6); the percentile mapping (Eq. 6) uses linear interpolation over the empirical CDF with $[0, 1]$ clipping at the bounds. Precomputation takes $\sim$2 minutes per deployment and is performed once.

Deployment	Traj.	Signals
HotpotQA, Qwen3-4B	7,405	54,061
HotpotQA, Qwen3-30B-A3B	741	2,543
HotpotQA, DeepSeek-V3.2	741	3,290
Bamboogle, Qwen3-4B	125	788
Bamboogle, Qwen3-30B-A3B	125	459
Bamboogle, DeepSeek-V3.2	125	485
2WikiMHQA, Qwen3-4B	500	3,200
2WikiMHQA, Qwen3-30B-A3B	500	1,745
2WikiMHQA, DeepSeek-V3.2	500	2,060
MuSiQue, Qwen3-4B	500	3,160
MuSiQue, Qwen3-30B-A3B	500	1,815
MuSiQue, DeepSeek-V3.2	500	2,725

Table 6: Reference distribution source statistics. All thinking-mode backbones; non-thinking variants use the same benchmark subsamples.

Sensitivity to reference size. We computed deployment ΔEM on HotpotQA-741 (Qwen3-4B) with reference sizes $\{50, 100, 200, 500, 7405\}$: ΔEM varies within ± 0.35 pp across sizes, with n=50 the noisiest. We recommend ≥ 100 trajectories.

B.5 Inference Cost

The controller adds two $\sim$68K-parameter forward passes per agent step. Memory footprint: $\sim$0.8 MB. Inference is negligible relative to the LLM's per-step cost.

C Baseline Implementation

C.1 Random

Samples uniformly among the four non-CONTINUE actions at each step with probability p=0.1. Sub-seeded per-trajectory from a master RNG (seed 42) to ensure approximately uniform action sampling across trajectories.

C.2 Token Budget

Terminates trajectories when cumulative tokens exceed the cap. Cap = 18,900 (1.20$\times$ Qwen3-4B HotpotQA baseline mean). The cap is calibrated for Qwen3-4B; on smaller-trajectory backbones the cap may not bind (Table 1).

C.3 Step-Count Cap

Force-terminates trajectories exceeding K=6 reasoning steps (the Qwen3-4B baseline median). Isolates fixed step-budget policy from learned intervention timing. Like Token Budget, K=6 is calibrated for Qwen3-4B and rarely triggers on larger backbones with sparser baseline overthinking.

C.4 Stop-Only

Uses the trained controller's gate and action heads but masks all non-STOP predictions to CONTINUE at deployment time, isolating multi-action design from gate timing. The observed STOP trigger rate (3.14%) exceeds the controller's internal STOP component rate (0.99%) because masking shifts the gate distribution.

D Extended Results

D.1 Per-Cell Metrics with F1 and Steps

Table 7 reports F1 and average steps in addition to the EM/tokens already shown in Table 1, plus the DeepSeek-V3.2 (non-thinking) runs. The controller's step inflation is bounded by design: at τ=0.3 on the in-domain cell, step count is 5.70 vs. baseline 6.03 (a 5.5% decrease), since COMPRESS-induced rewriting causes the agent to re-derive intermediate conclusions, lengthening some subsequent steps but shortening the overall trajectory.

D.2 COMPRESS Template Ablation: F1 Details

Table 8 reports F1 alongside EM and tokens for the four COMPRESS templates (Table 3 of the main text). V3 (deployed) ties or leads on F1 across all three benchmarks.

Method	HotpotQA-741			Bamboogle			2WikiMHQA			MuSiQue		
	Acc↑	F1↑	Step	Acc↑	F1↑	Step	Acc↑	F1↑	Step	Acc↑	F1↑	Step
Qwen3-4B-Thinking: in-domain training source												
Baseline	42.51	51.80	6.03	58.40	69.86	5.77	67.80	70.97	6.40	17.40	23.54	6.32
Random (p=0.1)	40.08	49.22	4.89	47.20	57.99	4.86	60.20	62.68	5.09	16.80	22.55	4.87
Token Budget	40.48	48.18	5.49	52.80	63.22	5.19	61.80	64.00	5.90	13.00	17.84	5.25
Step-Count Cap (K=6)	34.27	39.41	4.99	44.00	53.39	5.06	50.00	50.83	5.18	11.00	14.51	5.10
Stop-Only (Lu et al., 2025)	39.00	47.12	5.55	53.60	63.51	5.56	61.00	62.36	5.68	17.00	22.60	5.65
PACE τ=0.3	45.47	54.57	5.70	59.20	68.96	5.80	67.60	70.64	6.14	17.60	23.30	5.92
Qwen3-30B-A3B-Thinking: 8× scale within family												
Baseline	44.12	54.04	3.43	58.40	68.46	3.67	68.20	71.98	3.69	22.20	28.88	3.63
Random (p=0.1)	42.64	52.37	2.91	51.20	59.73	2.91	63.00	66.19	2.99	18.20	23.23	2.90
Token Budget	41.97	51.10	3.15	53.60	62.91	3.17	62.20	64.88	3.29	16.60	21.57	3.16
Step-Count Cap (K=6)	37.92	45.18	3.01	48.80	57.94	3.16	54.20	54.91	3.12	13.20	16.63	3.08
Stop-Only (Lu et al., 2025)	42.24	51.48	3.28	56.80	67.57	3.42	64.80	67.75	3.41	20.60	26.12	3.37
PACE τ=0.3	44.93	54.21	3.46	61.60	73.17	3.66	68.80	71.90	3.49	21.60	27.52	3.69
DeepSeek-V3.2-Thinking: 167× scale, different family												
Baseline	54.11	62.46	4.44	74.40	80.65	3.88	77.40	76.35	4.12	30.60	36.82	5.45
Random (p=0.1)	51.82	59.04	4.37	69.60	74.13	3.81	72.00	71.10	4.24	29.00	33.49	5.35
Token Budget	55.73	62.68	4.32	79.20	84.16	4.17	77.00	77.08	4.13	31.80	35.66	5.12
Step-Count Cap (K=6)	51.01	57.51	3.96	74.40	77.67	3.60	75.40	73.89	3.95	28.20	30.98	4.58
Stop-Only (Lu et al., 2025)	48.31	54.61	3.79	70.40	76.32	3.58	75.20	73.46	3.93	31.00	35.08	4.66
PACE τ=0.3	55.46	63.21	4.22	72.80	79.12	3.89	78.80	77.58	4.10	30.80	36.38	4.98

Table 7: Extended metrics for the deployment matrix: LLM-as-judge accuracy (Acc, %), F1 (%), and avg. steps per question. Same τ=0.3 checkpoint and baselines as Table 1; EM and ΔTok reported there.

COMPRESS variant	Bamboogle			2WikiMHQA			HotpotQA		
	Acc	F1	Step	Acc	F1	Step	Acc	F1	Step
Baseline	58.40	69.86	5.77	67.80	70.97	6.40	42.51	51.80	6.03
V1: results + last_thought	51.20	65.94	5.77	67.00	71.30	6.13	44.39	53.89	5.64
V2: queries + results	58.40	68.31	5.95	66.20	69.38	6.06	43.72	54.63	5.81
V3: q+r+last_thought	**59.20**	**68.96**	5.80	**67.60**	**70.64**	6.14	**45.47**	**54.57**	5.70
V4: last3 + r + last_thought	59.20	68.52	5.58	64.80	68.11	5.96	43.99	53.36	5.76

Table 8: F1-extended COMPRESS template ablation (Qwen3-4B-Thinking, τ=0.3). V1 drops the question; V2 drops the last thought; V3 (deployed) keeps both; V4 keeps only the last 3 rounds. Red: Acc loss $\geq$ 1 pp.