FROM RISK RECOGNITION TO SAFER GENERATION: COMPETE–SPLIT–FUSE FOR SELECTIVE REPRESENTATION COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models can correctly recognize a request's risks yet still produce unsafe content, exposing a gap between safety knowledge and behavior. We study this *aware-but-unsafe* (AUB) failure through paired risk judgments, token-level tracing, and activation interventions. The evidence supports a competition-based account: task-related representations can weaken safety's influence during generation, while autoregressive token–cache feedback can sustain the resulting trajectory. Motivated by this mechanism, we introduce **Compete–Split–Fuse (CSF)**, an inference-time method for frozen language models. CSF detects prefixes at risk of unsafe execution, forms a task-release reference branch with a separate key–value cache, and selectively transfers only safety-private information back to the main branch before each token decision, leaving shared, task-private, and residual components unchanged at the fusion layer. Prediction consistency adaptively ends the extra computation. Across **8 checkpoints from 3 families**, CSF reduces conditional unsafe assistance among pre-intervention knowledge-positive HarmBench requests from **25.4%** to **10.6%**, with broadly similar benign performance and a **full-response serial-time proxy of 1.13× baseline**. Mechanistic controls and ablations support predictive timing, task release, selective fusion, and persistent token–cache feedback as contributors to the observed gains.

1 INTRODUCTION

Safety alignment requires language models to respect safety constraints throughout generation while remaining useful. Yet a model may correctly identify and explain a request's risks in one context and produce unsafe content when asked to fulfill the same request in another. We call this an *aware-but-unsafe* (AUB) failure. In multi-step explanation, planning, and reasoning, the challenge is therefore not only to make safety knowledge available, but to keep it influential as task execution unfolds.

Prior work links safety failures to competing objectives (Wei et al., 2023) and shows that alignment can be concentrated on the first few output tokens (Qi et al., 2025). Inference-time methods offer complementary interventions: CAST applies activation steering conditionally (Lee et al., 2025), while AlphaSteer strengthens refusal while limiting changes to benign representations (Sheng et al., 2025). These advances motivate a more specific question: *when does available safety information lose influence during an ongoing response, and which internal components should be changed or preserved to restore it?*

Our analysis connects paired risk assessment and free generation to token-level tracing and activation interventions (Section 3). Harm-matched task-pressure interventions show that task execution can weaken safety's influence; crossed activation and next-token interventions support persistence through autoregressive feedback. Subtask transitions reveal opportunities for safety to regain influence as task support weakens. Geometry and held-out cross-interventions distinguish shared and function-specific components, separating risk recognition and safe help from refusal. These observations motivate restoring safety at vulnerable prefixes without indiscriminately suppressing task execution.

Guided by these findings, we introduce **Compete–Split–Fuse (CSF)** for frozen language models (Section 4). A prefix monitor identifies when intervention may be needed. CSF then creates a

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.

task-release reference branch by shifting only task-private coordinates toward a calibrated reference. The branches share model weights but maintain separate key–value (KV) histories. Before each token decision, CSF imports only the reference's safety-private component, preserving the main branch's shared, task-private, and residual components at the fusion layer. Both branches consume the emitted token; sustained agreement between fused and current-step bypass predictions closes the reference branch, with later reactivation allowed.

Across 8 evaluated checkpoints from 3 families, CSF reduces unsafe assistance from **25.4%** to **10.6%** on the fixed subset of paired HarmBench requests recognized as risky before intervention. Paired safe-and-useful completion rises from 56.0% to 67.2%, with a 0.7-percentage-point IFEval strict-accuracy loss and a full-response serial-time proxy of 1.13× baseline (Section 5). The results show a safety–utility trade-off, not universal dominance; prediction agreement does not guarantee safety.

Our contributions are:

- **A testable account of AUB failures.** We connect safety–task competition and autoregressive feedback to the gap between risk recognition and safe generation.
- **Evidence for selective intervention.** We identify shared and function-specific components through geometric analysis and cross-interventions that distinguish safety, refusal, and task execution.
- **Inference-time coordination with adaptive exit.** CSF combines prefix monitoring, separate branch histories, and token-wise safety fusion without changing backbone weights.

2 Related Work

Safety knowledge and unsafe generation. Prior work links safety failures to competing objectives and shallow, prefix-sensitive alignment (Wei et al., 2023; Qi et al., 2025). Agent evaluations and in-decoding safety probes further expose a gap between risk awareness and safe behavior (Tang et al., 2025; Zhao et al., 2026). We investigate how competition between safety and task direction clusters, amplified by autoregressive token feedback, weakens the influence of available safety knowledge.

Safety representations and activation steering. Refusal directions and multidimensional concept cones identify activation components that causally influence refusal (Arditi et al., 2024; Wollschläger et al., 2025). CAST, AlphaSteer, and dynamically scaled steering introduce conditional, utility-preserving, and context-adaptive activation interventions (Lee et al., 2025; Sheng et al., 2025; Ferrando et al., 2026). We study shared and function-specific components of context-dependent safety–task direction clusters, using their interactions during generation to guide selective intervention.

Decoding-time safety interventions. DExperts and SafeDecoding guide generation by combining model output distributions (Liu et al., 2021; Xu et al., 2024), while SafeProbing probes safety awareness during decoding to trigger refusal (Zhao et al., 2026). CSF detects potentially harmful competition, splits hidden-state computation into parallel branches within one frozen model, and selectively fuses safety-private information from a reference branch with reduced task pressure. Separate KV caches retain both branches' histories across token decisions, while each fusion preserves the main branch's shared and task-private components.

3 Competition and Trajectory Dynamics during Generation

To explain why models recognize risk yet generate unsafe content (Section 1), we first pair behavioral comparisons with token-level representation tracking. Some failures exhibit early safety-related signals that are not sustained during generation (Section 3.1). We then manipulate task requirements and safety reminders while matching harmful content; both manipulations shift generation tendencies, supporting competition between task execution and safety constraints (Section 3.2). Next, activation patching crossed with next-token distribution interventions examines autoregressive

feedback: local representation changes reinforce continuations through token selection, while serial-subtask analysis reveals renewed switching windows as current-task signals weaken (Section 3.3). Finally, geometric analysis and selective interventions identify both separable and shared components (Section 3.4). Together, these observations motivate Compete–Split–Fuse: temporal changes suggest when to intervene, and representation structure informs what to separate and coordinate through information exchange. Section 4 develops this design; we evaluate its benefits in Section 5.

3.1 From Risk Recognition to Sustained Safety during Generation

We assess each request in independent risk-judgment and free-generation contexts. A correct risk judgment with an adequate explanation followed by unsafe independent generation defines an *aware-but-unsafe* (AUB) case. AUB occurs in the evaluated models; model/category breakdowns appear in Appendix A.1.

To locate this gap during generation, we extract representations from matched risk-recognition, refusal, safe-assistance, and direct-execution trajectories, adapting residual-stream direction analysis (Arditi et al., 2024). For request x and prefix $y_{<t}$, hidden states $h_t^{(\ell)}$ yield separately fitted signed, calibrated safety/task support scores $a_S(t), a_T(t)$, implemented as held-out calibrated linear readouts. These scores measure association, not probabilities or complementary shares. Extraction, held-out validation, selective interventions, and reproducibility are specified in Appendices A.1 and A.4.

Against risk-aware safe responses (SA), AUB starts with comparable safety scores, then declines or fluctuates as SA remains steadier (Figure 1a). Risk information can remain readable, and residual failures persist across tested alignment checkpoints on the same requests (Appendix A.1, including benign/individual trajectories and alternative alignments). Early-token and recovery-training analyses provide related background (Qi et al., 2025). Our comparison asks how safety is maintained and whether task execution contributes to its loss.

3.2 Competition between Task Execution and Safety Constraints

Task scores rise gradually while AUB safety scores fall within shorter windows (Figure 1a). To test task pressure, we vary completion requirements while matching harmful actions, entities, and independent harm ratings, with equal-length neutral wrappers as controls. Stronger requirements advance safety decline and increase AUB (Figure 1b; Appendix A.2).

For the reverse manipulation, we compare no reminder, generic attention, abstract safety, and concrete risk knowledge. Abstract safety reminders improve safety but can reduce quality on benign requests or allowed subgoals (Figure 1c; Appendix A.2 separates refusal from safe assistance). This tests bidirectional influence while measuring the safety–utility trade-off relevant to conditional steering (Lee et al., 2025; Sheng et al., 2025).

Joint prefix signals outperform safety-only, task-only, and length/position baselines in held-out failure prediction (Figure 1d). Inputs use only current and past states; Appendix A.2 reports ROC/PR curves, lead time at fixed false-positive rates, and timing heterogeneity. These comparisons support context-dependent competition rather than a fixed transition threshold. We next examine how changing tendencies influence token choices and persist beyond the intervention.

3.3 Token Feedback and Transitions across Reasoning Subtasks

At matched prefixes before safety decline, we patch safety-related activations (Zhang & Nanda, 2024). Independent full-output rollouts evaluate candidate tokens' safety and allowed-content utility. Patching increases probability mass on tokens associated with safe, useful continuations. We cross actual patch status with sampling from baseline or patched next-token distributions, then resume identical free decoding: exactly four conditions (Appendix A.2). The appendix specifies donor, layer, position, strength, duration, and KV-cache handling, alongside token-value and intervention controls.

Changing the token source alters subsequent safety trajectories; risk-recognition, safe-assistance, and refusal components yield different responses (Appendix A.2). These effects support autoregressive feedback: altered tokens enter the context and reinforce a continuation. Persistence after a local change raises a complementary question: when can generation switch again?

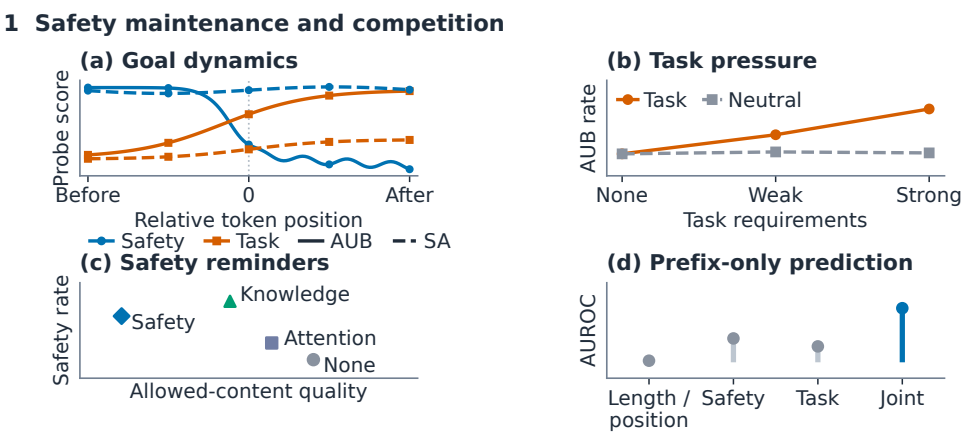

Figure 1: (a) Independent safety/task scores; AUB solid, SA dashed. (b) Harm-matched task/neutral wrappers. (c) Reminders: safety versus allowed-content quality. (d) Same-prefix prediction. Curves and markers show the evaluated trajectory summaries; endpoint definitions and uncertainty procedures are given in the appendices. Protocols: Appendices A.1–A.2.

We align serial reasoning to independently annotated subtask completion and detect representation changes separately. Current-task scores can weaken before visible markers such as "however," followed by increased relative safety score; absolute safety need not rise (Appendix A.2). Weakening or maintaining task representations advances or delays switching, whereas marker insertion has smaller effects. The appendix reports combined interventions, four continuation outcomes, and checks without safety conflict or explicit chain-of-thought. This pattern suggests windows before overt safety reflection. Which information can be handled separately within them?

3.4 Representation Structure and Implications for Branching and Fusion

We compare rank-one, low-rank, and context-conditioned representations using held-out readouts, interventions, transfer, and side effects. A few additional directions improve cross-context intervention effects before gains saturate (Appendix A.3). Refusal geometry motivates this capacity comparison and the distinction between orthogonality and intervention independence (Wollschläger et al., 2025).

Principal angles and canonical correlations characterize geometry; cross-interventions establish functional selectivity. Some candidates selectively affect safety or task behavior, whereas others influence both (Appendix A.3). Component-specific over-refusal effects provide complementary evidence. Shared candidates may overlap, and context outside their joint subspace remains relevant.

Together, the temporal evidence motivates prefix-based detection of competition windows (Compete); selective components motivate context-preserving, objective-conditioned branches (Split); and shared components motivate information exchange to coordinate continuation (Fuse). Section 4 specifies these operations without assuming that geometric separation ensures independent branches. Their ability to maintain safety constraints while preserving allowed task completion is tested in Section 5.

4 Compete–Split–Fuse

Motivated by the competition, feedback, transition, and representation findings in Section 3, we propose *Compete–Split–Fuse* (CSF), an inference-time method for a frozen language model. CSF implements the resulting intervention logic in three linked steps: monitor the prefix to determine *when* safety may need support, construct a reference computation under reduced task pressure, and restore only safety-related information before each token decision. Safety denotes risk recognition and safe assistance; utility denotes legitimate task quality and safe help rather than completion of harmful requests.

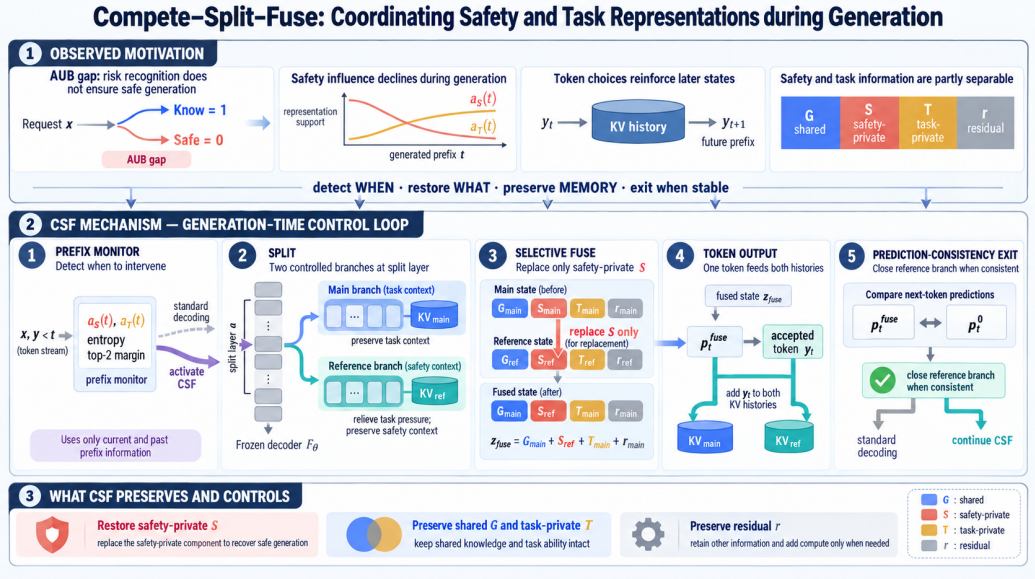

Figure 2: **Mechanistic overview of Compete–Split–Fuse (CSF).** The observations motivate prefix monitoring and controlled branch coordination. The loop detects *when* to act, keeps separate main/reference KV histories, replaces only safety-private S, feeds the accepted token to both histories, and exits on prediction consistency; shared G, task-private T, and residual r from the main branch are preserved.

Decoding remains ordinary until a trigger creates main and reference branches at a split layer a. The branches traverse the same frozen interval to a fusion layer b with separate KV histories; CSF then replaces only the main branch's safety-private component, preserves its shared, task-private, and residual components, and produces one fused next-token distribution. Both branches consume the accepted token, so the intervention stays inside one evolving sequence; prediction consistency later closes the reference branch. This directly connects the observed timing and token-feedback dynamics to a persistent but selective inference-time intervention.

4.1 Prefix-Based Competition Detection

The observation protocol shows that safety–task balance can change with the generated prefix, so activation must be causal and updated during decoding. Let x be the input, $y_{<t}$ the emitted tokens, and $s_t = (x, y_{<t})$ the prefix used to predict y_t. For a frozen decoder-only Transformer F_θ with L blocks and residual width d, $h_t^{(\ell)} \in \mathbb{R}^d$ is the residual state at the current prediction position after block ℓ; committed KV caches are part of the runtime state because identical text can arise from different internal histories.

We reuse the signed, calibrated safety and task supports $a_S(t)$ and $a_T(t)$ from Section 3.1, together with the entropy and top-two probability margin of the ordinary prediction p_t^0, to form ϕ_t. A lightweight monitor predicts whether unsafe execution would begin within the next H_d tokens:

$$g_t = D_\omega(\phi_{t-w+1:t}) \approx \Pr(E_t = 1 \mid \phi_{t-w+1:t}),$$
$$\text{activate at } t \iff \text{inactive at } t \text{ and } g_t \geq \tau_{\text{on}}. \tag{1}$$

Here w is the feature-window length and E_t is the short-horizon onset event, rather than a claim that causal competition has already been identified. Features use only current or earlier prefix computations, and triggering precedes commitment of y_t.

High task support alone is not an unsafe event because legitimate work can also engage task execution. The ordinary forward pass supplies the monitor, split/fusion states, and p_t^0, and its main-branch computation is reused after activation. Independent onset labels and held-out calibration connect the gate to a measurable short-horizon event (Appendix B.3); Appendix C.1 separates prediction quality from the behavioral value of the chosen intervention time. Thus, the detector determines *when* to act, while the representation analysis determines *what* to change.

4.2 Representation Splitting and Token-Wise Fusion

The geometric and cross-intervention observations motivate a shared–private decomposition at split boundary a and fusion boundary b, with $a < b < L$. At either boundary $\ell \in \{a, b\}$, an offline-mean whitening transform gives $z^{(\ell)} = W_\ell(h^{(\ell)} - \mu_\ell)$, and a prefix selector chooses context c_t with layer-specific bases. For a fixed layer and context, let $B = [G, S, T] \in \mathbb{R}^{d \times r_B}$ be a full-column-rank basis for shared, safety-private, and task-private directions.

We jointly obtain the coordinates and residual by

$$\begin{aligned} q = [\alpha; \beta; \gamma] = \arg\min_{u \in \mathbb{R}^{r_B}} \|z - Bu\|_2^2 = B^\dagger z, \\ r = (I - BB^\dagger)z, \qquad z = G\alpha + S\beta + T\gamma + r. \end{aligned} \tag{2}$$

Here G retains request/context information used by both functions, S safety-specific information, T task progress, and r variation outside the selected subspaces. Joint least squares keeps the coordinates well defined for nonorthogonal groups; bases are fitted offline from matched trajectories and selected through functional cross-interventions, with risk recognition and safe help treated as safety candidates and pure refusal analyzed separately (Appendix B.2).

The main branch is unchanged, while the reference moves only its task-private coordinates toward a matched task-release mean:

$$z^{(a)}_{t,\mathrm{ref}} = z^{(a)}_{t,\mathrm{main}} + \rho T_{a,c_t}\big(\bar{\gamma}^{\mathrm{rel}}_{a,c_t} - \gamma^{(a)}_{t,\mathrm{main}}\big), \qquad 0 < \rho \le 1. \tag{3}$$

The stored $\bar{\gamma}^{\mathrm{rel}}$ is the matched training-state mean after the current subtask and before the next dominates, excluding EOS states. It is retrieved online, preserves the reference's shared, safety-private, and residual components, and leaves the main task state intact; this operationalizes the transition observation without predicting a future boundary or assuming task norms must decrease.

After inverse whitening, both branches traverse the same frozen interval with distinct histories:

$$h^{(b)}_{t,j} = F_{a+1:b}\big(h^{(a)}_{t,j}; K^{a+1:b}_{t,j}\big), \qquad j \in \{\mathrm{main}, \mathrm{ref}\}. \tag{4}$$

At first activation the reference copies the main branch's pre-step interval cache; thereafter each branch retains its own cache while consuming the same tokens and positions. This cache separation preserves the token-feedback distinction identified in Section 3.3; the complete ownership and decoding protocol is in Appendix B.4.

At b, CSF substitutes only the reference safety-private coordinates:

$$\begin{aligned} z^{(b)}_{t,\mathrm{fuse}} &= z^{(b)}_{t,\mathrm{main}} + S_{b,c_t}\big(\beta^{(b)}_{t,\mathrm{ref}} - \beta^{(b)}_{t,\mathrm{main}}\big) \\ &= G_{b,c_t}\alpha^{(b)}_{t,\mathrm{main}} + S_{b,c_t}\beta^{(b)}_{t,\mathrm{ref}} + T_{b,c_t}\gamma^{(b)}_{t,\mathrm{main}} + r^{(b)}_{t,\mathrm{main}}. \end{aligned} \tag{5}$$

Under the full-rank decomposition this preserves the main branch's shared, task-private, and residual coordinates at layer b, while later nonlinear behavior still requires validation. The fused state passes through blocks $b + 1{:}L$, normalization, and the LM head to produce p^{fuse}_t; the committed token feeds both branches at the next step. Matched-donor, component-swap, refusal, token–cache, and sensitivity controls test whether the observed gain comes from selective safety restoration (Appendices C.1,C.1,C.2).

4.3 Prediction-Consistency Exit and Computational Cost

Because the reference influence may change as the shared prefix advances, CSF compares the fused prediction p^{fuse}_t with a bypass prediction p^0_t from the same committed history, where the bypass omits only the current fusion. Temporary upper-layer cache writes remain isolated, and only fused writes are committed during an active step. We use total variation over vocabulary $\mathcal{V}$:

$$d_t = \mathrm{TV}(p^{\mathrm{fuse}}_t, p^0_t) = \tfrac{1}{2}\sum_{v \in \mathcal{V}} |p^{\mathrm{fuse}}_t(v) - p^0_t(v)|. \tag{6}$$

The distributions share decoding settings; for any next-token event its probability changes by at most d_t, which is a single-step statement rather than a future-safety guarantee.

Let k_t count consecutive active steps with $d_t \le \epsilon$ and identical top-1 predictions:

$$k_t = \begin{cases} k_{t-1} + 1, & d_t \le \epsilon \text{ and } \arg\max_v p_t^{\text{fuse}}(v) = \arg\max_v p_t^0(v), \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$
$$k_t \ge M \quad \Longrightarrow \quad \text{close the reference branch from step } t+1.$$

The confirming step still commits the fused token; closing releases only the reference state, retains actual history, and permits later reactivation. Exit therefore measures marginal intervention influence at the current prefix, not subtask completion or whole-answer safety, following patience-based dynamic-computation precedents (Zhou et al., 2020; Schuster et al., 2022).

All representation, reference, selector, and monitor fitting is offline with the backbone frozen. Training, validation, exit-calibration, and final free-generation splits remain separate; withdrawal calibration uses teacher-forced future-distribution stability, while independent free-generation tests cover reactivation and jointly wrong predictions (Appendices B.1,C.2). Thus, calibration fixes when to stop extra computation, and the behavioral experiments test whether that decision is safe and useful.

The active-step cost includes both downstream predictions:

$$C_{\text{active}} = C_{1:a} + 2C_{a+1:b} + 2C_{b+1:L} + C_{\text{aux}}. \tag{8}$$

Here $C_{u:v}$ includes the indicated blocks and final output processing, while C_{aux} covers monitoring, low-rank operations, cache management, and vocabulary comparison. Appendix B.7 expands the resource accounting; the experiments separately test timing, reference information, selective fusion, feedback, withdrawal, safety, legitimate utility, and measured computation on matched requests (Appendix C.2).

5 Experiments

The trajectory evidence in Section 3 motivates three questions: whether CSF improves the safety–utility trade-off, whether its reference construction and selective fusion—rather than the trigger alone—produce the gain, and whether intervention can be withdrawn without substantial relapse. We compare CSF with unmodified decoding and representative decoding-time interventions across safety, allowed-task utility, over-refusal, capability, and cost; Appendices D–G specify the complete protocol, and all numerical values below are the results of the evaluation.

5.1 Broad effectiveness and computational cost

Because the observations attribute failure to safety–task competition rather than absent refusal behavior, we first ask whether CSF improves safety without suppressing permissible completion. Table 1 and Figure 3 summarize behavior, capability, the safety–utility frontier, and cost. We compare conditional CAST (Lee et al., 2025), activation-dependent AlphaSteer (Sheng et al., 2025), a safety-task instantiation of dynamic DSAS (Ferrando et al., 2026), SafeDecoding (Xu et al., 2024), and explicit text splitting. DSAS-S is adapted for safety, while matched-monitor AlphaSteer separates trigger quality from reference content; full baseline definitions, tuning, published cells, and GAPS/ALTSTEER protocols appear in Appendix F.

Table 1: **Results at the prespecified operating point.** Equal-weight means over eight evaluated checkpoints. HB: HarmBench unsafe assistance; $U \mid K$: unsafe assistance given fixed pre-intervention Know$= 1$; Joint: safe-and-useful completion; OR: XSTest over-refusal; IFE: IFEval strict accuracy; LCB: LiveCodeBench pass@1.

Method	HB ↓	$U \mid K$ ↓	Joint ↑	OR ↓	IFE ↑	LCB ↑
Unmodified	25.6	25.4	56.0	9.1	78.5	32.6
CAST	16.4	16.2	60.0	12.3	76.6	31.0
AlphaSteer	11.5	11.4	65.3	10.4	77.5	31.7
DSAS-S	11.3	11.2	66.4	9.9	77.7	31.9
SafeDecoding	9.0	8.9	61.0	16.4	76.4	30.9
Text split	7.9	7.9	71.7	14.3	76.9	31.3
CSF	10.7	10.6	67.2	10.1	77.8	32.0

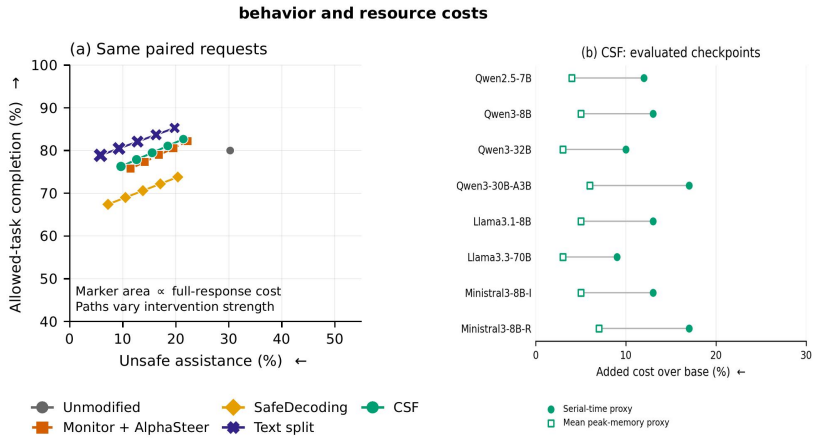

Figure 3: **Effectiveness and cost.** (a) Unsafe assistance versus allowed-task completion; marker area denotes cost. (b) CSF serial-time and mean peak-memory proxies. Paths vary intervention strength; panel (b) reports only evaluated checkpoints (Appendix D).

CSF reduces HarmBench unsafe assistance from 25.6% to 10.7% while increasing paired safe-and-useful completion from 56.0% to 67.2%. The gain is not uniform: AlphaSteer is only 0.8 points worse on HarmBench, text splitting reaches 71.7% joint completion, and SafeDecoding obtains lower unsafe assistance with more benign refusal. CSF loses 0.7 points on IFEval, while XSTest indicates broadly similar benign performance; its serial-time proxy is 1.09–1.17 times base versus 2.37 times for text splitting. Thus, the results support a modest benefit whose value depends on the added reference computation, not dominance on every endpoint. Checkpoint-level results appear in Table 7.

5.2 Testing the proposed competition mechanism

Having established the overall trade-off, we next test whether the improvement follows the competition account that motivated CSF. Holding harmful intent fixed, increasing task pressure raises unsafe assistance by 28.0 points without intervention but by 10.0 points with CSF (Figure 4a). This pressure interaction is consistent with the proposed mechanism, although wording, length, and semantic difficulty remain alternatives addressed by the paired controls in Appendix H.

Timing and reference content then separate *when* to intervene from *what* to restore. At equal duration, predictive triggering yields 8.0 points less unsafe assistance than random matched triggering (panel b), with retrospective onset used only as an oracle anchor. Holding trigger and fusion fixed, moderate task release maximizes joint safe-and-useful completion at 71.0%, whereas full release lowers it to 64.0% despite a small safety gain (panel c). Equal-norm perturbation, whole-state scaling, and invalid-output accounting rule out a generic-disruption interpretation.

Finally, panel (d) isolates the transferred coordinates. Adding shared coordinates to safety-private replacement lowers unsafe assistance by 2.0 points but costs 9.0 points of allowed-task completion. Blocking and replay, crossed donor components, refusal directions, the token–cache factorial, and same-text re-prefill provide complementary controls; complete results are in Appendix H, with external replication in Appendix K. Together, these comparisons connect the observations to the intervention: competition-associated prefix signals guide when to act, while reference construction and selective replacement determine what information is restored; the semantic interpretation of the subspaces nevertheless remains a separate empirical claim.

5.3 Which design choices are necessary?

We then isolate which components account for the gain. On the prespecified stress cohort, matched-monitor AlphaSteer and full CSF yield 18.0% and 14.0% unsafe assistance, respectively, while removing task release raises the rate to 30.0% (Table 15). Thus, the trigger alone does not explain the reference branch in this evaluation. State averaging and logit mixing retain some safety benefit but reduce joint completion; first-only and periodic fusion leave more failures than per-token fusion;

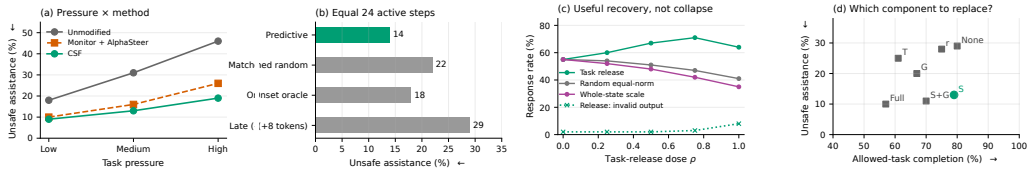

Figure 4: **Mechanism tests.** (a) Matched intent under varied task pressure. (b) Trigger timing at equal duration. (c) Task release and disruption controls. (d) Replacement of safety-private S, task-private T, shared G, residual r, and combined coordinates. Endpoints follow free continuation.

and disabling exit slightly improves behavior at higher cost. Reset-history controls and the full layer, rank, window, horizon, threshold, detector-size, and exit scans appear in Appendix J.

5.4 When can ordinary decoding resume?

Finally, because CSF adds computation only while active, we test whether prediction agreement supports withdrawal. From the same eligible state, the continued and withdrawn branches evolve freely (Figure 6a). Without restart, withdrawal adds 1.0 point of unsafe assistance in routine cases and 5.0 points in long or repeated conflicts; monitor-based restart reduces the latter to 2.0 points. These absolute behavior and allowed-task outcomes, reported in Appendix I, prevent local agreement from being mistaken for a safety guarantee.

Whole-request accounting includes requests that never trigger, never exit, restart, or hit the cap. Exit saves 10.9% of the serial-time proxy while adding 0.6 points of unsafe assistance, making withdrawal a safety–cost trade-off. Missed detection, an uninformative reference, permissible-task damage, and post-exit relapse remain; low-resource languages, repeated conflicts, and defense-aware search are stress tests, with high-budget attack success at 47.0% (Appendix K.1). Overall, the results connect the observed dynamics to CSF's design: prefix signals govern activation, the task-release reference supplies safety information, selective fusion limits interference, and prediction agreement governs exit.

6 Conclusion

Compete–Split–Fuse (CSF) connects three observations about generation—time-varying safety influence, token–cache feedback, and partially separable representations—to one selective decoding loop. A prefix monitor determines when competition requires intervention; a task-release reference branch restores safety-private information while selective fusion preserves shared, task-private, and residual context; prediction-consistency then limits the duration of the extra computation.

The resulting design makes the safety–utility trade-off explicit: intervention should be timely and selective, legitimate task quality should be measured alongside safety, and withdrawal should be treated as a calibrated risk decision rather than a guarantee. Relapse, over-refusal, uninformative references, and distribution shift remain important failure modes. The evaluation reports numerical outcomes with matched compute and uncertainty estimates.

Ethics Statement

This work studies safety-relevant behavior of language models using public benchmark prompts and controlled prompt variants. The protocol is designed for offline, isolated evaluation; it does not recruit external human subjects or collect private personal data. Annotation and adjudication are performed by doctoral researchers on the research team under safeguards for handling generated harmful content. Harmful outputs should be access-controlled and should not be redistributed as operational instructions. Any released artifact should therefore contain prompt identifiers or hashes, scorer and configuration versions, and safe summaries rather than raw harmful generations. The study also reports over-refusal, allowed-task utility, and failure boundaries so that improving a safety metric is not conflated with indiscriminate refusal. The authors remain responsible for complying with the licenses, terms of use, and safety requirements of every model and dataset used.

Reproducibility Statement

The CSF definitions, update equations, cache ownership rules, exit criterion, and cost accounting are given in Sections 4 and 5; the appendices specify the data splits, estimands, controls, baseline audit, implementation invariants, and statistical analysis.

AI Use Statement

Generative AI tools assisted with prose coding, editing, translation, LaTeX refactoring and layout, method compression, and organization of experimental protocol descriptions.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.11717.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form, 2023. URL https://arxiv.org/abs/2306.03819.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1608.06019.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 11515–11542. PMLR, 2025. URL https://proceedings.mlr.press/v267/cui25a.html.

Moghis Fereidouni, Muhammad Umair Haider, Hassan Sajjad, and A. B. Siddique. GAPS: Dimension-level gates for conditional activation steering, 2026. URL https://arxiv.org/abs/2609.01878v1. Version 1.

Alex Ferrando, Xavier Suau, Jordi Gonzàlez, and Pau Rodriguez. Dynamically scaled activation steering, 2026. URL https://arxiv.org/abs/2512.03661v2. Version 2, revised 30 July 2026; first submitted December 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

HarmBench contributors. HarmBench: official evaluation repository, 2024. URL https://github.com/centerforaisafety/HarmBench. Official evaluation repository and text-behavior benchmark release.

Chigozirim Ifebi, Brent Kong, and Ayushi Mehrotra. Minionese: Comprehensive benchmark and mechanistic study of multilingual LLM safety, 2026. URL https://arxiv.org/abs/2607.10112v1. Version 1; coherence-filtered native non-refusal ASR distinguished from harmful assistance.

Hoejoon Kwon, Byeonggeuk Lim, Kahyeon Kim, and YoungBin Kim. ALTSTEER: Selective safety steering for moving beyond hard refusals to constructive alternatives, 2026. URL https://arxiv.org/abs/2608.30197v1. Version 1.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2409.05907.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.acl-long.522/.

LiveCodeBench contributors. LiveCodeBench: official benchmark and versioned releases, 2025. URL https://github.com/LiveCodeBench/LiveCodeBench. release_v6; May 2023–April 2025 collection window; accessed 25 September 2026.

Meta AI. Llama official model cards, 2024. URL https://huggingface.co/meta-llama. Official model repository; evaluated Llama 3.1 8B and Llama 3.3 70B identifiers are listed in Table 7; accessed 25 September 2026.

Mistral AI. Ministral 3 8B: official instruction and reasoning checkpoint cards, 2025. URL https://huggingface.co/mistralai. Official model repository; evaluated 2512 instruction and reasoning checkpoint identifiers are listed in Table 7; accessed 25 September 2026; no model-card scores imported.

MMLU-Pro contributors. MMLU-Pro: official benchmark repository, 2024. URL https://github.com/TIGER-AI-Lab/MMLU-Pro.

OR-Bench contributors. OR-Bench: official over-refusal benchmark repository, 2024. URL https://github.com/justincui03/or-bench. Hard-1K subset documentation checked 25 September 2026.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1645. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/46499a0622ecf568b72d17b61e45dbd5-Abstract-Datasets_and_Benchmarks_Track.html.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

Qwen Team. Qwen official model cards: target checkpoint identities and architecture, 2026. URL https://huggingface.co/Qwen. Official model repository; exact evaluated checkpoint identifiers are listed in Table 7; accessed 25 September 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98. Diamond split; official benchmark paper and dataset release.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2207.07061.

Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. AlphaSteer: Learning refusal steering with principled null-space constraint, 2025. URL https://arxiv.org/abs/2506.07022v2. Version 2; source tables 1–2 used for published reference values.

SORRY-Bench contributors. SORRY-Bench: official benchmark and evaluator releases, 2025. URL https://github.com/sorry-bench/sorry-bench. March 2025 base dataset release; checked 25 September 2026.

StrongREJECT contributors. StrongREJECT: official dataset and rubric evaluation implementation, 2024. URL https://github.com/dsbowen/strong_reject. Primary implementation documentation; checked 25 September 2026.

Yuzhi Tang, Tianxiao Li, Elizabeth Li, Chris J. Maddison, Honghua Dong, and Yangjun Ruan. LM agents may fail to act on their own risk knowledge, 2025. URL https://arxiv.org/abs/2508.13465.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper/2023/hash/fd6613131889a4b656206c50a8bd7790-Abstract-Conference.html.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66945–66970. PMLR, 2025. URL https://proceedings.mlr.press/v267/wollschlager25a.html.

XSTest contributors. XSTest: official benchmark repository, 2023. URL https://github.com/paul-rottger/xstest. Benign subset protocol checked 25 September 2026.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5587–5605, 2024. doi: 10.18653/v1/2024.acl-long.303. URL https://aclanthology.org/2024.acl-long.303/.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

Zhihao Zhang, Liting Huang, Guanghao Wu, Preslav Nakov, Heng Ji, and Usman Naseem. Health-ORSC-Bench: A benchmark for measuring over-refusal and safety completion in health context, 2026. URL https://arxiv.org/abs/2601.17642v2. Version 2, revised 26 June 2026; tier and all-response SCR definitions checked.

Yinzhi Zhao, Ming Wang, Shi Feng, Xiaocui Yang, Daling Wang, and Yifei Zhang. Defending large language models against jailbreak attacks via in-decoding safety-awareness probing, 2026. URL https://arxiv.org/abs/2601.10543.

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.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.04152.

A Additional Observation Experiments

Overview. The observation study uses matched requests, held-out interventions, and request-level uncertainty. The goal is to separate three questions: whether safety information is available, when its influence declines, and which state/token mechanisms can sustain or restore it.

A.1 Measurement and the Knowledge–Behavior Gap

For each request, collect matched trajectories for explicit risk recognition, appropriate refusal, safe assistance, and direct execution. Risk recognition, safe help, refusal, and task progress are annotated separately, so safety and task labels are not complements. Split by source-request family before generating variants; fit preprocessing and extractors on training data, select layer/rank/regularization on validation data, and keep test/transfer requests and donor pools disjoint.

At prediction step t, $h_t^{(\ell)}$ is the residual state after block ℓ at the last available prefix position, before sampling y_t. Request-weighted training moments define the common whitened coordinate system

$$\widehat{\Sigma}_\lambda = (1-\lambda)\widehat{\Sigma} + \lambda \frac{\mathrm{tr}(\widehat{\Sigma})}{d} I + \epsilon I, \qquad W = \widehat{\Sigma}_\lambda^{-1/2}, \qquad z_t = W(h_t^{(\ell)} - \widehat{\mu}), \tag{9}$$

and separate linear safety/task readouts are reported as standardized margins and projection-energy diagnostics,

$$a_k(t) = \frac{w_k^\top z_t + b_k - m_{k,\mathrm{ref}}}{s_{k,\mathrm{ref}}}, \qquad o_k(t) = \frac{\|U_k U_k^\top z_t\|_2^2}{\|z_t\|_2^2}, \quad k \in \{S, T\}. \tag{10}$$

These are readout scores, not calibrated probabilities or conserved quantities. Difference-in-means, SVD/LDA/probe, context-conditioned, random-subspace, shuffled-label, and full-state controls are selected and evaluated on held-out requests; LEACE is used only as a linear-accessibility control (Arditi et al., 2024; Belrose et al., 2023).

Risk judgment and free generation are run from separate fresh contexts. For harmful request i, let K_i denote correct risk recognition and S_i a safe independent response. We report $N^{-1}\sum_i K_i$, $N^{-1}\sum_i S_i$, all-request $\mathrm{AUB} = N^{-1}\sum_i K_i(1-S_i)$, and the conditional failure rate among $K_i = 1$ cases. Token traces are shown both in raw position and under diagnostic event alignment; no-decline and censored trajectories remain in the denominator. Alignment-checkpoint comparisons use the same requests and preserve successful trajectories rather than selecting only failures.

A.2 Competition, Token Feedback, and Subtask Transitions

Task-pressure variants change only completion demands while preserving harmful intent; neutral length-matched wrappers and reminder controls separate task pressure from extra instruction salience. Prefix-only detectors use current/past readouts, slopes, and permitted position features to predict an independently annotated unsafe onset within a fixed horizon; future annotations and offline change points are labels/diagnostics only. Thresholds are selected on validation data and reported with request-level false alerts, misses, and lead time.

For local causal tests, clone a held-out prefix and patch only the selected safety component for one prediction step,

$$h_t' = h_t + \alpha W^{-1} P_S (z_{t,\mathrm{donor}} - z_t), \tag{11}$$

using matched safe donors and validation-selected strength. The primary cache-retaining protocol allows the patched state to alter current-position K/V entries; a logits-only control restores the same baseline cache, which isolates token-distribution effects from hidden-history effects (Zhang & Nanda, 2024). With baseline/patched token distributions p_0, p_1, the four crossed cells execute representation state $Z \in \{0,1\}$ and sample the forced next token from p_b:

$$\mu_{zb} = \mathbb{E}_{v\sim p_b}[\mathbb{E}[Y \mid \mathrm{do}(Z=z), \mathrm{do}(M=v)]]\,. \tag{12}$$

All four cells are reported for final safety and safe-and-useful completion, together with the interaction $I = \mu_{11} - \mu_{10} - \mu_{01} + \mu_{00}$; token-value estimation uses independent rollout draws.

Subtask analyses use doctoral researchers blind to probe curves to mark observable completion boundaries and visible transition markers. Latent weakening, relative-support change, absolute

safety increase, and marker time are reported separately because a rise in $a_S - a_T$ may be caused only by falling task support. Matched interventions weaken or maintain the current-task component, insert visible markers, or combine both. Switch probability, switch delay, final safety/utility, and semantic damage are all retained; no-switch trajectories are censored rather than discarded. In the tested conditions, state interventions shift switching more than marker insertion, but this is a conditional pattern rather than a universal generation cycle.

A.3 Geometry and Functional Validation

Rank-one, low-rank, and context-conditioned representations are compared under the same validation budget and utility tolerance. For fitted safety/task bases U_S, U_T, principal angles follow

$$U_S^\top U_T = L \operatorname{diag}(\sigma_j) R^\top, \qquad \theta_j = \arccos(\sigma_j), \tag{13}$$

while held-out CCA summarizes statistical co-variation. Geometric overlap is not functional identity, so candidate labels are assigned only after matched additions/ablations/patches show the intended behavior with bounded non-target effects (Wollschläger et al., 2025).

The union basis $U_\cup$ is used only for bookkeeping,

$$z_t = P_\cup z_t + (I - P_\cup) z_t, \qquad P_\cup = U_\cup U_\cup^\top, \tag{14}$$

and the residual is preserved in intervention controls rather than discarded. Cross-effects separately measure risk judgment, safe assistance, refusal, and allowed-task completion. XSTest and the prespecified OR-Bench subset provide an over-refusal check (Röttger et al., 2024; Cui et al., 2025); improved helpfulness from indiscriminate compliance is not counted as success.

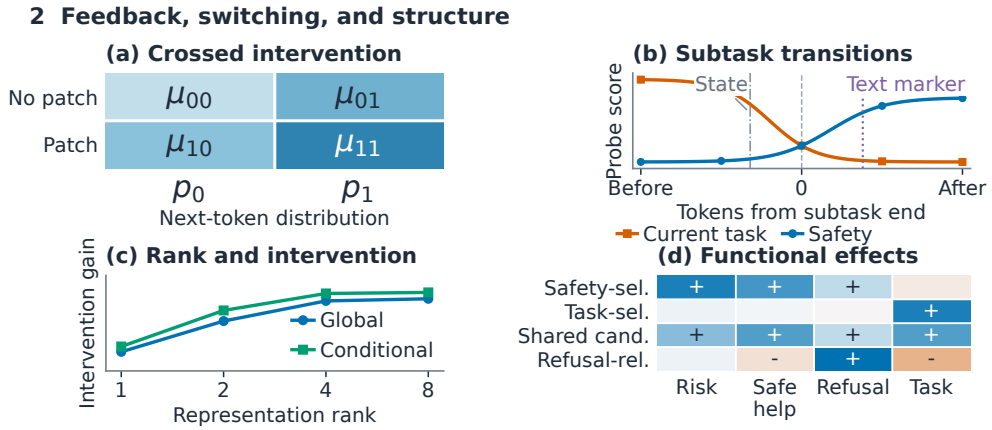

Figure 5: **Consolidated observation results.** (a) Patch × token-source means separate representation and token effects. (b) Independent subtask boundaries locate transition dynamics. (c) Rank scans show capacity sensitivity. (d) Component–behavior interventions test functional selectivity. Error bars follow the request-level uncertainty procedure below.

A.4 Statistical Analysis and Reproducibility

The source request is the experimental and bootstrap unit: variants, tokens, annotations, and rollouts from one request stay together, and repeated rollouts are averaged within request before paired contrasts. The analysis reports the prespecified independent data/training seeds, request-level bootstrap intervals, raw rates/effect sizes, and between-seed variation. Confirmatory families use their declared multiplicity rule; exploratory rank/threshold scans remain exploratory. Missing, truncated, no-change, and no-switch cases are retained with explicit reasons and denominators, and transfer results are shown by model/category/language before any macro-average.

Table 2: Compact reproduction checklist for the observation experiments.

Setting	Required record
Models / data	checkpoint and tokenizer revisions; request-family split IDs; licenses, hashes, transfer cohorts
Generation / extraction	decoder and cap; residual hook; whitening/reference scales; layer/rank/regularization
Interventions	donor/prefix rule; patch duration/amplitude; affected cache entries; random/full-state controls
Annotations / detector	onset and transition rules; judges; thresholds; agreement; censored/ambiguous cases
Statistics / seeds	request-level resampling; data/training/decoding seeds; multiplicity and robustness settings
Software / compute	framework, precision, deterministic flags, serialized transforms, per-request outcomes

The accompanying code includes the implementation, configuration settings, wrapper/reminder strings, split identifiers, extraction/patch routines, and cache checks, subject to dataset restrictions. The reported claims follow the completed held-out analyses rather than post-selection on favorable cases.

B Method Specification and Implementation

Overview. This appendix defines the CSF state variables, representation decomposition, cache ownership, token-wise fusion, exit rule, and computational accounting.

B.1 Notation, Runtime State, and Data Separation

The specification assumes a frozen decoder-only Transformer with accessible residual streams and a standard per-block KV cache. Both intervention boundaries follow complete blocks. Architectures with recurrent, hybrid, or cross-layer shared state require a separate state-management specification. The online method uses no additional language-model judge, candidate scorer, adapter, or learned fusion network. All branch computations share the backbone parameters θ.

At prediction step t, the last token of $s_t = (x, y_{<t})$ is the pending input position. Processing it produces the residuals used to predict y_t. Historical caches contain only positions processed before that pending input. Current-step K/V entries are provisional until the algorithm commits a computation. The newly selected y_t becomes the pending input on the next step; it has not already been processed into the cache. For the first output token, ordinary prompt prefill processes all prompt positions except the last, which is processed transactionally. Equivalent batched prefill must expose the same boundary without duplicating its final position.

Table 3: Core notation. Layer and context indices are omitted only when fixed.

Symbol	Meaning
$x, y_{<t}, s_t$	Input, emitted output prefix, and their concatenation.
L, d, a, b	Number of blocks, residual width, split boundary, and fusion boundary; $a < b < L$.
$K_t^{\le a}, K_{t,\mathrm{main}}^{a+1:b}, K_t^{>b}$	Committed front-end, main-interval, and upper-layer histories before the pending input.
$K_{t,\mathrm{ref}}^{a+1:b}$	Additional interval history retained during an active episode.
$h^{(\ell)}, z^{(\ell)}$	Raw and whitened residuals after complete block ℓ.
μ_ℓ, W_ℓ	Offline background mean and invertible whitening transform.
c_t, B_{ℓ,c_t}	Predicted context and its layer-specific joint basis.
$G, S, T; r_G, r_S, r_T$	Shared, safety-private, and task-private bases and their column counts.

Symbol	Meaning
$r_B, \alpha, \beta, \gamma, r$	Total basis rank; three coordinate groups; residual outside the joint span.
E_G, E_S, E_T	Row-selection matrices extracting the three coordinate groups from $q \in \mathbb{R}^{r_B}$.
$\bar{\gamma}^{\mathrm{rel}}_{a,c}, \rho$	Offline task-release reference and movement strength.
a_S, a_T, ϕ_t, w	Signed functional supports, current features, and window length.
$H_d, g_t, \tau_{\mathrm{on}}$	Onset-prediction horizon, monitor output, and trigger threshold.
$p_t^0, p_t^{\mathrm{fuse}}$	Current-step bypass and fused next-token distributions.
d_t, ϵ, M, k_t	Total variation, exit tolerance, patience requirement, and current consistency count.
$H_e, \delta_{\mathrm{cal}}, \alpha_{\mathrm{cal}}$	Withdrawal-calibration horizon, divergence tolerance, and allowed empirical violation fraction.

The runtime state is the prefix together with its committed caches, active flag, feature history, and patience count. Past fused computation can affect future upper-layer attention even after the reference branch closes. Accordingly, p_t^0 preserves this history and bypasses only the current fusion. Reconstructing a never-intervened cache from the same text is a distinct experimental manipulation (Appendix C.1).

We separate four request-level datasets: $\mathcal{D}_{\mathrm{fit}}$ for representation, support-readout, reference, selector, and monitor fitting; $\mathcal{D}_{\mathrm{val}}$ for configuration selection and monitor calibration; $\mathcal{D}_{\mathrm{exit}}$ for exit calibration; and $\mathcal{D}_{\mathrm{test}}$ for final evaluation. Probability calibration and configuration selection use disjoint request subsets within $\mathcal{D}_{\mathrm{val}}$. All variants of a source request, including paraphrases, translations, wrappers, matched responses, and decoding seeds, remain in one split. Observation discovery and confirmation data retain their own separation from method testing. Category-holdout studies reserve entire categories; their labels are evaluation metadata, not online inputs.

Fit candidate operators on $\mathcal{D}_{\mathrm{fit}}$, select their configuration on $\mathcal{D}_{\mathrm{val}}$, and lock both before using $\mathcal{D}_{\mathrm{exit}}$. Do not refit on the combined splits without obtaining a fresh calibration split. Checkpoints, split manifests, fitting seeds, and selected settings are versioned together. No backbone weights are updated at any stage of this procedure.

B.2 Basis, Context, and Task-Release Reference Estimation

Matched contrasts and whitening. Following the observation measurement protocol, collect matched states for risk recognition, safe assistance, pure refusal, and task execution for the same request at comparable semantic positions. Safety contrasts compare risk recognition or safe help with unconditional execution. Task contrasts compare continuation with pausing the current task. Match topic, semantic position, response length, and format; record residual mismatches for stratified checks. Refusal-only contrasts are retained separately, rather than defining all safety functions through refusal.

For each layer, estimate a background mean μ_ℓ and covariance $\widehat{\Sigma}_\ell$ using training activations, and set

$$\widetilde{\Sigma}_\ell = (1-\eta)\widehat{\Sigma}_\ell + \eta \frac{\mathrm{tr}(\widehat{\Sigma}_\ell)}{d} I, \qquad W_\ell = \widetilde{\Sigma}_\ell^{-1/2}, \qquad 0 < \eta \le 1. \tag{15}$$

Assuming nonzero total background variance, this shrinkage covariance is positive definite. Its square root and inverse are defined through the symmetric eigendecomposition. All contexts and branches at layer ℓ use this same transform. The selected background sampling weights and η are fixed before testing. Apply truncated SVD to the matrices of whitened paired differences to obtain candidate safety and task subspaces. Candidate ranks are selected using held-out readout performance, functional interventions, and legitimate-task effects, not explained variance alone.

Shared and private candidate directions. Let Q_S, Q_T be orthonormal bases for the two candidate subspaces. Their principal angles are obtained from

$$Q_S^\top Q_T = U \,\mathrm{diag}(\sigma_i) V^\top, \qquad s_i = Q_S U_{:i}, \quad t_i = Q_T V_{:i}. \tag{16}$$

The singular values σ_i are principal-angle cosines. For pairs above a shared threshold selected on validation data, use normalized $s_i + t_i$ as shared candidates. Remaining principal vectors and unmatched directions supply private candidates. Differences within highly aligned pairs that are not selected remain outside the chosen joint span; the decomposition retains the original state through its residual rather than deleting those differences.

Functional selection uses held-out cross-interventions: shared candidates should affect both functions, safety-private candidates are evaluated for risk recognition and safe help, and task-private candidates for current-task progress. These are tests of a proposed functional organization, not consequences of the subspace names. Pure refusal, language quality, and information retention are measured separately. Refusal-direction interventions (Arditi et al., 2024) motivate a refusal-specific control, without equating their target to the broader safety functions considered here. An independently fitted linear-erasure control, such as LEACE (Belrose et al., 2023), can test readout necessity; its linear predictability criterion does not establish behavioral necessity. The corresponding behavioral protocols are in Appendices C.1 and C.2.

For each candidate (ℓ, c) configuration, require

$$B_{\ell,c} = [G_{\ell,c}, S_{\ell,c}, T_{\ell,c}], \quad r_B = r_G + r_S + r_T \le d, \quad \operatorname{rank}(B_{\ell,c}) = r_B. \tag{17}$$

Groups may be nonorthogonal. Reject bases that violate predeclared lower bounds on the smallest singular value or upper bounds on the condition number. Merge or remove near-duplicate directions, then repeat functional validation. Compute the Moore–Penrose inverse with a stable QR or SVD factorization offline. In the main implementation, the solve is unregularized and joint: $B^\dagger B = I_{r_B}$ and $B^\dagger r = 0$. Separate Euclidean projections are not equivalent. A ridge solve is an explicitly labeled numerical sensitivity variant, to which the exact preservation proposition does not transfer unchanged.

Context selection. The training context assignments are those established in the observation study's shared/private analysis. The deployment selector is a linear softmax classifier fitted to these assignments from current split-layer main states:

$$c_t = \arg\max_c \operatorname{softmax}(A_{\text{ctx}} h^{(a)}_{t,\text{main}} + b_{\text{ctx}})_c. \tag{18}$$

Here $A_{\text{ctx}} \in \mathbb{R}^{C \times d}$, $b_{\text{ctx}} \in \mathbb{R}^C$, and C is the number of deployed contexts. Cross-entropy is fitted on $\mathcal{D}_{\text{fit}}$ and selector accuracy is checked on held-out requests. Training assignments can use annotated contexts; prediction never consumes test risk labels, future phase labels, or a task-end oracle. The clustering rule, assignments, and merging history must accompany the fitted artifact. Both boundaries and branches use the same c_t, each boundary retrieving its own basis. When c_t changes, recompute coordinates from the current state. Old coefficients are not transported across contexts, and raw KV entries are not rotated.

Task-release reference. Use independently annotated task boundaries in the fitting trajectories to identify states after the current subtask has finished and before a subsequent subtask has become dominant. Match these states to the current-task context and exclude EOS positions and states already executing the next subtask. Denote the retained set by $\mathcal{R}_c$. With $E_T \in \mathbb{R}^{r_T \times r_B}$ selecting task coordinates, estimate

$$\bar{\gamma}^{\text{rel}}_{a,c} = \frac{1}{|\mathcal{R}_c|} \sum_{u \in \mathcal{R}_c} E_T B^\dagger_{a,c} W_a (h^{(a)}_u - \mu_a). \tag{19}$$

Both pre-release and release states are read in this same current-task basis. Sparse contexts are merged and refitted offline so that every deployable context has a basis and a reference. Record matched-pair identifiers, semantic-boundary annotations, exclusion counts, and per-context reference counts.

Equation 3 changes γ to $(1 - \rho)\gamma + \rho\bar{\gamma}^{\text{rel}}$, with all other split-layer coordinates and the residual unchanged. It moves toward a measured training condition rather than toward the origin. Select ρ on validation data using the effects on task execution, risk recognition, safe help, and legitimate-task quality. The availability of a matched training reference does not imply that its nonlinear continuation supplies a useful safety update for a new request; that is the donor intervention question in Appendix C.1.

B.3 Monitor Supervision, Features, and Calibration

Prospective event labels. Generate ordinary continuations with the frozen backbone and the decoding settings intended for deployment. Doctoral researchers independently locate the first semantic unit that begins unsafe execution and map its start to token index T. For at-risk prediction positions $t \le T$, define

$$E_t = \mathbf{1}\{t \le T < t + H_d\}. \tag{20}$$

The prediction at $t = T$ concerns the first unsafe token and is included; positions after the first onset are excluded from first-event supervision. Labels are not derived from supports, their thresholds, or a putative competition score. Under sampling, one continuation supplies a realized Bernoulli label, not the true event probability; multiple seeds remain nested within their source request.

A naturally terminated safe trajectory supplies negative labels up to its EOS. For a length-capped trajectory, a negative label requires observation of the full H_d-token horizon, whereas an already observed onset supplies a positive label even if the later response is truncated. Incomplete event-free horizons are censored, not labeled safe. If semantic onset is only localized to an interval, use a binary label only when it is invariant over the admissible onset indices; otherwise exclude it from the primary supervised loss and report the ambiguous count and boundary sensitivity. Censoring and annotation rules are fixed without reference to monitor predictions.

Features and fit. Let $a_S(t), a_T(t)$ be the observation protocol's calibrated log-likelihood-ratio supports, with the same sign orientation, functional labels, and frozen readout layers. The fitting labels, class priors, calibration map, numerical clipping policy, and readout layers belong to the inherited observation artifact and are fixed before deployment. These supports are distinct from geometric occupation measures. Reuse that artifact and its likelihood-ratio convention rather than introducing a new score or calibrating against test outcomes. A minimal input vector is

$$u_t = \big[a_S(t), a_T(t), \mathcal{H}(p_t^0), p_{t,(1)}^0 - p_{t,(2)}^0\big], \qquad \mathcal{H}(p) = -\sum_{v \in \mathcal{V}} p(v) \log p(v), \tag{21}$$

where $p_{t,(1)}^0, p_{t,(2)}^0$ are the largest two probabilities and $0 \log 0 = 0$. Standardize coordinates using fitting-set moments. Set ϕ_t to the standardized vector with a validity indicator; concatenate w vectors, padding missing early-generation history with standardized zeros and zero validity. This window permits the classifier to use changes in support without introducing another online score. All features are read from the current ordinary forward pass or stored earlier steps, including during active episodes.

Use a two-layer MLP with a ReLU hidden layer and scalar logit f_ω; its sigmoid is the uncalibrated event prediction. For the eligible positions $\mathcal{T}_i$ of request i, let $\widehat{g}_{it} = \sigma(f_\omega(\phi_{i,t-w+1:t}))$ and fit request-weighted binary cross-entropy:

$$\begin{aligned} \mathcal{L}(\omega) &= \frac{1}{N_{\mathrm{fit}}} \sum_{i=1}^{N_{\mathrm{fit}}} \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \ell_{\mathrm{BCE}}(\widehat{g}_{it}, E_{it}), \\ \ell_{\mathrm{BCE}}(p, e) &= -e \log p - (1 - e) \log(1 - p). \end{aligned} \tag{22}$$

Requests with no eligible positions are reported and excluded from N_{fit}. When multiple continuations are used, first average within request. Optimizer, hidden width, fitting schedule, and stopping rule are recorded in the configuration manifest. Calibrate $g_t = \sigma(f_\omega(\phi_{t-w+1:t})/T_D)$ with scalar $T_D > 0$ on the probability-calibration subset, following temperature scaling (Guo et al., 2017). This temperature affects monitor logits only and is separate from language-model sampling temperature.

Select τ_{on} using a predeclared trade-off among event recall, false triggers, and added computation on the configuration-validation subset. Report AUPRC, request-weighted Brier score, reliability bins, onset lead time, and both token- and request-level false-trigger rates. To assess reactivation after prior intervention, save actual prefix/cache snapshots on held-out pre-test requests, continue ordinarily from them, and independently annotate ensuing events. Check calibration on these altered states before locking the monitor; any refitting uses fitting requests only. These checks address a deployment shift and do not redefine prediction as causal competition identification.

Because p_t^0 is a feature, the monitor is evaluated after the current ordinary forward pass. Save its boundary states and provisional cache entries. On activation, reuse this computation and evaluate

only the extra reference interval and fused tail. Deactivation depends exclusively on the exit rule, not on a transient drop of g_t.

B.4 Complete Decoding Procedure and Cache Ownership

A forward operation returns an output and a current-position cache delta without mutating its input history. Implementations may use copy-on-write pages, reserved slots, or equivalent transactional buffers. The logical requirement is that neither tail sees the other tail's current writes. A history snapshot can be a protected view rather than a physical copy. Sampling randomness is used once for the actual output token; evaluation-only distributions never emit another token.

Algorithm 1: Compete–Split–Fuse decoding

1. **Input:** frozen F_θ; boundaries a, b; whitening/basis/reference artifacts; context selector and monitor; $\rho, \tau_{\text{on}}, \epsilon, M$; decoding configuration and resource cap.
2. **Initialize:** ordinary prompt history excluding its final prediction position; $\text{active} \leftarrow \text{false}$, $k \leftarrow 0$, empty feature history; final prompt token is pending.
3. **For each prediction step** t **until EOS or resource cap:** preserve the pre-step histories $K_t^{\le a}, K_{t,\text{main}}^{a+1:b}, K_t^{>b}$ and, if active, $K_{t,\text{ref}}^{a+1:b}$.
4. Run the ordinary forward pass on the pending token. Save $h_{t,\text{main}}^{(a)}$, $h_{t,\text{main}}^{(b)}$, and p_t^0 and provisional deltas ΔK_{front}, ΔK_{main}, and ΔK_{tail}^0. Do not commit them yet.
5. Append current features, compute g_t using Equation 1, and retain the most recent w feature vectors.
6. **If inactive and** $g_t < \tau_{\text{on}}$**:** choose y_t from p_t^0 with the configured decoder; commit ΔK_{front}, ΔK_{main}, and ΔK_{tail}^0; set $k \leftarrow 0$; proceed to line 16.
7. **If inactive:** fork $K_{t,\text{ref}}^{a+1:b}$ from the *pre-step* main-interval history, excluding ΔK_{main}; set $\text{active} \leftarrow \text{true}$ and $k \leftarrow 0$.
8. Select c_t from the current main split state by Equation 18; use this same context at both boundaries and for both branches.
9. Jointly read the split-layer coordinates and construct $h_{t,\text{ref}}^{(a)}$ by Equation 3 and inverse whitening (or the equivalent low-rank operator).
10. Run the reference interval $F_{a+1:b}$ using its own pre-step history; obtain $h_{t,\text{ref}}^{(b)}$ and ΔK_{ref}.
11. Read both states using B_{b,c_t}; construct $h_{t,\text{fuse}}^{(b)}$ by Equation 5 and inverse whitening.
12. Run the fused tail from $K_t^{>b}$, *not* from the ordinary tail's provisional writes; obtain p_t^{fuse} and $\Delta K_{\text{tail}}^{\text{fuse}}$ through the original normalization and head.
13. Compute full-vocabulary TV and common-rule top-1 predictions. Increment k if both consistency conditions hold, otherwise reset $k \leftarrow 0$.
14. Choose the *single* actual y_t from p_t^{fuse}. Commit ΔK_{front}, ΔK_{main}, ΔK_{ref}, and $\Delta K_{\text{tail}}^{\text{fuse}}$ to their respective histories. Discard ΔK_{tail}^0.
15. **If** $k \ge M$**:** mark inactive for the next step, reset $k \leftarrow 0$, and release only the reference-interval history. Retain all actual main and fused-tail history.
16. Append y_t to the common output prefix. If it is native EOS, terminate; otherwise make y_t the pending input for the next iteration. A resource-cap stop is recorded as truncation.

Table 4: Cache ownership for one active prediction step. Deltas refer to the pending input, not the newly selected output token.

Region	Prior history	Current computation	Committed delta
$1{:}a$	One shared history	Ordinary front end	Front-end delta
$a+1{:}b$, main	Main interval	Unmodified main input	Main delta

Region	Prior history	Current computation	Committed delta
$a+1{:}b$, reference	Persistent reference interval; forked at episode start	Task-condition-modified input	Reference delta
$b+1{:}L$	One actual history shared by both current tails	Bypass and fused tails, isolated provisional writes	Fused delta only; bypass delta discarded

The two active tails are *bypass-main and fused* predictions; there is no reference-tail language-model prediction in the main algorithm. Main-interval cache entries are retained even though upper layers consume a fused residual. Closing does not reconstruct any past states. On a later trigger, fork again from the then-current main-interval history. There is no extra cooldown or oracle for task boundaries. Context changes are handled through current residual readouts, without resetting the reference history.

Correctness tests must cover the first prompt prediction, one-step and multi-step episodes, EOS on a confirming step, and exit followed immediately by reactivation. With identical interval histories, deterministic arithmetic, and $\rho = 0$ as a diagnostic setting, both branch outputs and the fused/bypass predictions must agree up to numerical tolerance. Setting the reference safety coordinates to the main coordinates also makes fusion an identity. Setting $\rho = 0$ *after* histories have diverged need not make the two branches equal. Check that cache positions are appended once, old entries remain immutable, current tail buffers are isolated, and sampling is invoked only for the committed output.

B.5 Coordinate Preservation and Local Response

Fix layer b and context c_t, and abbreviate main, reference, and fused states as $z_m, z_{\mathrm{ref}}, z_f$. Let E_S select safety coordinates and define the joint-coordinate read/write operator

$$\Pi_S = SE_SB^\dagger, \qquad z_f = z_m + \Pi_S(z_{\mathrm{ref}} - z_m). \tag{23}$$

This is generally an oblique projection, not an independent Euclidean projection onto S. The symbol Π_S distinguishes it from Euclidean occupation operators in the observation analysis.

Proposition 1 (Local coordinate preservation) *Let $B = [G, S, T]$ be a joint basis with full column rank. Define $q_j = B^\dagger z_j$ and write $q_j = [\alpha_j; \beta_j; \gamma_j]$. The fusion in Equation 23 satisfies*

$$B^\dagger z_f = [\alpha_m; \beta_{\mathrm{ref}}; \gamma_m], \qquad (I - BB^\dagger)z_f = (I - BB^\dagger)z_m. \tag{24}$$

Moreover, $\Pi_S^2 = \Pi_S$.

Proof. Since $S = BE_S^\top$ and $B^\dagger B = I$, we have $B^\dagger S = E_S^\top$ and $(I - BB^\dagger)S = 0$. Therefore

$$B^\dagger z_f = q_m + E_S^\top(\beta_{\mathrm{ref}} - \beta_m), \qquad \Pi_S^2 = SE_S(B^\dagger S)E_SB^\dagger = SE_SB^\dagger. \tag{25}$$

Applying $I - BB^\dagger$ to z_f yields the residual identity. The same argument with T, E_T proves split-layer preservation of α, β, r under the reference move. □

The proposition concerns exact arithmetic in a fixed basis at one layer. It does not make the groups semantically independent, bound the raw perturbation norm, or preserve downstream task quality. Ill-conditioning can magnify numerical error even with full rank. If a ridge readout $(B^\top B + \lambda I)^{-1}B^\top$ is substituted, its product with B is not the identity for $\lambda > 0$; report coordinate drift instead of applying the proposition. Appendix C.2 specifies numerical and downstream checks.

A local interpretation of reference propagation. Only at a first fork with identical interval histories K can the initial effect be isolated as a current-input perturbation. Fix both bases and the selected context, and write

$$\Psi(z; K) = W_b\left[F_{a+1:b}(W_a^{-1}z + \mu_a; K) - \mu_b\right], \qquad \delta z = \rho T_{a,c}(\bar{\gamma}_{a,c}^{\mathrm{rel}} - \gamma_m^{(a)}). \tag{26}$$

Where Ψ is differentiable at the main input, its Jacobian J_Ψ gives

$$\beta_{\mathrm{ref}}^{(b)} - \beta_m^{(b)} = E_SB_{b,c}^\dagger J_\Psi(z_m^{(a)}; K)\delta z + o(\|\delta z\|_2). \tag{27}$$

The expression identifies a possible route from changed task coordinates to safety-coordinate changes through the frozen nonlinear interval. It supplies no sign or behavioral-benefit guarantee. Once reference histories differ, their cache-dependent effects are also present; Equation 27 is not a complete explanation of later steps. Randomized task-condition and token/cache controls test these distinctions.

B.6 Prediction Consistency and Withdrawal Calibration

Decoding convention. Let ℓ_t^0 and ℓ_t^{fuse} denote logits from the original head. With sampling, p_t^j is the normalized distribution actually passed to the sampler after the fixed temperature, masks, and top-k/top-p processing. Apply identical settings to both logits; their resulting supports can differ. With greedy decoding, retain the softmax distribution for TV, and choose its argmax using a single deterministic vocabulary-order tie rule. Do not replace these distributions by one-hot greedy decisions. Compare the entire vocabulary, with zero probabilities outside truncated supports, and include that work in cost. Invalid or nonfinite distributions are runtime errors to log, not evidence for exit.

Proposition 2 (Single-step total variation) *For two distributions p, q over a finite vocabulary, let $d = \frac{1}{2}\sum_v |p(v) - q(v)|$. For every token set $A \subseteq \mathcal{V}$ and every function $f : \mathcal{V} \to [0, 1]$,*

$$|p(A) - q(A)| \le d, \qquad |\mathbb{E}_p f - \mathbb{E}_q f| \le d. \tag{28}$$

There exists a coupling of samples from p and q whose mismatch probability is d, and no coupling has a smaller mismatch probability.

Proof. For $\Delta(v) = p(v) - q(v)$, $\sum_v \Delta(v) = 0$ implies that the sum of its positive entries and the absolute sum of its negative entries both equal d. Summing over any subset cannot exceed either bound. For $0 \le f \le 1$, the weighted sum of Δ is maximized by selecting its positive entries and minimized by selecting its negative entries, giving the same bounds. For coupling, match the common mass at each vocabulary item. These diagonal pairs carry total mass $1 - d$; couple the remaining masses arbitrarily. Their supports are disjoint, yielding mismatch mass d. Any coupling has diagonal mass at most $\sum_v \min\{p(v), q(v)\} = 1 - d$. □

This maximal-coupling result is an existence statement; using the same random seed is not in general maximal coupling. The top-1 condition separately ensures the same immediate greedy choice, which arbitrarily small TV alone need not ensure near a tie. The patience rule compares the two predictions at each step; it does not require that successive emitted tokens be identical. A sequence of small past distances places no bound on an unobserved future conditional distribution. Agreement can also occur when both computations are unsafe.

Independent-cache withdrawal calibration. Lock the monitor, subspaces, reference artifacts, ρ, and decoder before exit calibration. Prespecify a finite grid of (ϵ, M), a future horizon H_e, divergence tolerance δ_{cal}, allowed empirical violation fraction α_{cal}, and minimum evaluable coverage. On each request in $\mathcal{D}_{\text{exit}}$, generate a reference trajectory that stays active after its first trigger through EOS or the resource cap. The current bypass predictions are still computed at each active step, so every candidate rule can be evaluated without changing that trajectory before its proposed exit.

For each candidate, take its first eligible exit per request. Complete that step using the fused token, then clone the actual state. The continuing arm retains reference computation; the withdrawn arm releases it and disables reactivation during the diagnostic window. Both arms consume the same reference tokens, but each commits and recursively uses its own subsequent caches. At the jth prediction after the exit, compare

$$\widetilde{d}_{t,j} = \text{TV}(p_{t+j}^{\text{continue}}, p_{t+j}^{\text{off}}), \qquad V_t = \mathbf{1}\left\{ \max_{1 \le j \le J_t} \widetilde{d}_{t,j} > \delta_{\text{cal}} \right\}, \tag{29}$$

Here $J_t \le H_e$ is the number of available follow-up predictions, stopping at the horizon, natural reference EOS, or the resource cap. Natural EOS completes the reference continuation; it is distinct from censoring at the cap. A cap-limited shorter window with no observed violation is censored. Record it separately from observed violations in calibration. An exit on the already committed EOS has no withdrawal window or saved decoding work; report terminal exits separately rather than as

successful withdrawal tests. Requests with no trigger or no candidate exit are likewise retained in the full-policy cost denominator, not counted as divergence passes.

Cache independence is essential: resetting the withdrawn arm to the continuing history at every step would only repeat a current-step ablation. Here, even under identical teacher-forced text, upper-layer histories can diverge recursively. A teacher-forced token need not have been selected by the withdrawn sampler and can lie outside its truncated support; this is a controlled conditional diagnostic, not a claim about its free-running trajectory.

Compute the request-level empirical violation fraction using observed violations plus unresolved windows over eligible nonterminal exits, and require the prespecified coverage criterion. Among candidates within the allowed fraction, select the one with least measured computation in separate full-policy runs on these calibration requests, including reactivation and every request category. If none qualifies, report that the prespecified calibration criteria are not met. This calibration does not inherit sequence-level guarantees of other adaptive-computation methods (Schuster et al., 2022). Independent free-generation comparisons, with separate caches and outputs, evaluate actual safety and utility after withdrawal (Appendix C.2).

B.7 Low-Rank Implementation, Cost, and Configuration

Precomputed read/write operators. At each (ℓ, c), precompute

$$R_{\ell,c} = B^{\dagger}_{\ell,c} W_\ell, \qquad A^S_{\ell,c} = W_\ell^{-1} S_{\ell,c}, \qquad A^T_{\ell,c} = W_\ell^{-1} T_{\ell,c}. \tag{30}$$

The online interventions can then be evaluated directly in the raw residual stream:

$$\begin{aligned} h^{(a)}_{t,\mathrm{ref}} &= h^{(a)}_{t,\mathrm{main}} + \rho A^T_{a,c_t}\left[\bar{\gamma}^{\mathrm{rel}}_{a,c_t} - E_T R_{a,c_t}(h^{(a)}_{t,\mathrm{main}} - \mu_a)\right], \\ h^{(b)}_{t,\mathrm{fuse}} &= h^{(b)}_{t,\mathrm{main}} + A^S_{b,c_t} E_S R_{b,c_t}(h^{(b)}_{t,\mathrm{ref}} - h^{(b)}_{t,\mathrm{main}}). \end{aligned} \tag{31}$$

The layer-b centering terms cancel in the difference. These formulas avoid dense online whitening, inversion, or explicit residual reconstruction. The contractions cost $O(dr_B)$ per selected boundary; operators can be reduced to the used selector rows. Fitted support readouts, selector, monitor, and context lookup are counted separately, including their storage and device transfers. Frozen weights are shared, not duplicated.

Compute accounting. Let $C^{\mathrm{base}}_t = C_{1:a,t} + C_{a+1:b,t} + C_{b+1:L,t}$, with tail cost including normalization, the head, and configured probability processing. Let I_t indicate an active step, $C_{\mathrm{mon},t}$ monitoring overhead, and $C_{\mathrm{edit},t}, C_{\mathrm{TV},t}$ low-rank editing and comparison costs. A per-request account is

$$\begin{aligned} C_{\mathrm{request}} = C_{\mathrm{prefill}} + C_{\mathrm{lifecycle}} &+ \sum_t \left[C^{\mathrm{base}}_t + C_{\mathrm{mon},t}\right] \\ &+ \sum_t I_t \left[C_{a+1:b,t} + C_{b+1:L,t} + C_{\mathrm{edit},t} + C_{\mathrm{TV},t}\right]. \end{aligned} \tag{32}$$

$C_{\mathrm{lifecycle}}$ includes branch creation/release and cache-management costs not included in forward operations. Cache copying and data movement are also recorded in bytes and latency, since they need not be well described by FLOPs. Shared ordinary processing is counted once, while each active step adds a reference interval and a fused tail. Full-vocabulary TV/top-1 reduction costs $O(|\mathcal{V}|)$, in addition to producing both vocabulary distributions. The free comparison forward is included in the systems accounting. Prefill counts only the historical positions, because the final prompt position is included in the first decoding step.

For roughly constant step costs and active fraction ν, average decoding work is the base plus monitoring and ν times the added interval, tail, edit, and comparison work. Report the exact length-dependent sum when attention costs vary. Since the trigger uses p^0_t, the ordinary pass precedes a newly activated reference computation. Parallel scheduling can alter latency but does not remove either tail's work. Natural-length timing and fixed-length systems tests are reported separately; shorter refused answers are not interpreted as reduced per-token computational overhead.

Persistent and temporary memory. Only the interval reference cache is an extra persistent branch history. If block ℓ stores $n_{\mathrm{kv},\ell}$ KV heads of dimension $d_{\mathrm{head},\ell}$ using s_ℓ bytes per scalar, a full physical copy of a history of n_t positions requires

$$\Delta\mathcal{M}_{\mathrm{ref}}(t) = n_t \sum_{\ell=a+1}^{b} 2 s_\ell n_{\mathrm{kv},\ell} d_{\mathrm{head},\ell}. \tag{33}$$

Copy-on-write implementations can share immutable pre-fork pages, so only post-fork positions require distinct storage, plus page metadata and allocation granularity. Report which implementation is used. The two tail versions need isolated current-step buffers and vocabulary outputs, not two persistent upper-layer histories. Peak memory must include temporary forward activations, full-vocabulary arrays, fitted operators, monitor state, and allocator overhead. Inactive steps release the extra reference history. Offline activation collection, SVD/factorization, functional selection, fitting, and calibration costs are reported separately.

Configuration manifest. Table 5 records the configuration fields used by the evaluation. Candidate grids and selection criteria are fixed before test access, and the selected layer, rank, threshold, model count, and dataset size are recorded with the accompanying code.

Table 5: Configuration fields recorded for the evaluation; selected values are recorded with the accompanying code.

Group	Values/artifacts to record	Selection or source
Backbone and interface	Checkpoint revision; architecture; tokenizer; chat template; residual hooks; EOS IDs	Execution manifest; verify standard cache assumptions
Data	Dataset versions; source-family split hashes; counts; risk/utility rubrics; annotation agreement	Fixed split and annotation manifest
Whitening	Background sample weights; η; precision; covariance artifact	Fit; select on validation
Bases	a, b; r_G, r_S, r_T by context/layer; angle threshold; singular-value and condition-number limits	Fit candidates; functional validation
Contexts and references	Context assignment rule and count; selector weights; merging log; $\lvert\mathcal{R}_c\rvert$; boundary/matching exclusions	Fit artifacts; held-out checks
Support readouts	Layers; labels; calibrated LLR convention and priors; scaling artifacts	Inherited observation fit; validation
Reference movement	Candidate $\rho \in (0, 1]$ and selected value	Safety, help, and utility validation
Monitor	w, H_d; hidden width; optimizer; fit seeds; stopping; T_D, τ_{on}	Fit; disjoint calibration/selection subsets
Exit	Candidate grid; selected ϵ, M; $H_e, \delta_{\mathrm{cal}}, \alpha_{\mathrm{cal}}$; coverage/censoring counts	Separate exit calibration
Decoder	Greedy/sampling; temperature; masks; top-k/top-p; tie rule; generation cap; seeds	Freeze with calibration
Runtime	Precision; software revisions; attention kernels; device; batching; cache pages/copy policy; parallelism	Execution and systems manifest
Evaluation	Dataset, method, and model coverage; primary contrasts; margins; power/precision targets; resampling plan	Fix before final evaluation

Group	Values/artifacts to record	Selection or source
Offline resources	Activation storage; model passes; fitting/calibration time; device-hours	Measure, do not omit

C Experiments Linked to the Method

Overview. This section is a compact map from the method claims to the controlled evaluations below; detailed protocols are stated once in the sections that report their results.

Table 6: Compact experiment map. IDs are retained for traceability; "Evidence" points to the appendix section that contains the corresponding controls and numerical outcomes.

Group	IDs / question	Evidence
Detection and timing	E09, E17, E21: onset prediction, intervention time, detector × content	Appendix H; sensitivity in Appendix J.
Reference and components	E03–E05, E10–E12, E18–E20: task release, donor content, basis rank/context/layer, private/shared/refusal controls	Appendix H and Appendix J.
Token feedback and cache	E11, E13, E14, E16: block/replay, token × cache, fusion frequency, same-text re-prefill	Appendix H.
Withdrawal and restart	E15, E24, E33: true free continuation, exit policy, repeated conflict	Appendix I and Appendix K.
Behavior and systems	E06–E08, E25–E28, E32, E34: broad safety/utility, cost, external/multilingual/adaptive tests, length/evaluator controls, failure attribution	Appendices E, G, and K.
Algebra and limits	E36: coordinate identities, prediction agreement, implementation checks	Appendix L.

C.1 Detection, Reference, and Feedback Controls

All timing arms share eligible prefixes and active-step budgets; reference-dose and component swaps hold the trigger and continuation policy fixed; token/cache tests fork identical prefixes and then remove future intervention. This separates *when* CSF acts, *what* the reference contributes, and *how* accepted tokens and hidden history carry effects forward. The consolidated results are in Appendix H.

C.2 Withdrawal, Sensitivity, and Validity Controls

Exit is calibrated from prediction stability but evaluated by independent free continuation; sensitivity scans vary one design axis at a time while retaining all failed and capped requests; external tests preserve model, language, attack, and repeated-conflict strata rather than pooling them away. The relevant numerical results appear in Appendices I, J, and K.

D Evaluation construction and estimands

Overview. All endpoints are request-level and use prespecified denominators; model macros average only populated evaluated checkpoints. Each benchmark macro uses the same evaluated checkpoint set, while health and multilingual evaluations preserve their tier/language strata. Request-level

sample sizes and bootstrap intervals are reported with seed variation where applicable; pooled request rates are not substituted for fixed checkpoint macros.

For a fixed test request i, let K_i^0 be its pre-intervention knowledge label, U_i^m indicate unsafe assistance under method m, and A_i^m indicate completion of the independently specified allowed task. The relevant quantities are

$$\mathrm{AUB}_{\mathrm{all}}^m = N^{-1} \sum_i K_i^0 U_i^m, \qquad \mathrm{Unsafe}_{K=1}^m = \frac{\sum_i K_i^0 U_i^m}{\sum_i K_i^0}, \tag{34}$$

$$\mathrm{SafeUseful}^m = N^{-1} \sum_i (1 - U_i^m) A_i^m. \tag{35}$$

AUB uses the whole test denominator; conditional unsafe assistance uses the fixed knowledge-positive denominator. Neither is recovery on a selected failure subset. Across the evaluated checkpoints, overall AUB changes from 19.3% to 8.1%, whereas conditional unsafe assistance changes from 25.4% to 10.6%. Knowledge labels are resolved before steering, and all-request AUB bounds are reported for unresolved labels. Knowledge labels must never be recomputed after steering.

The discovery-defined failure-subset retest is reported as a paired difference on a fixed subset, not a treatment "recovery" rate. Likewise, historical trajectory fractions, conditional $K = 1$ failure, and all-request AUB have different denominators and are never interchanged.

E Model, task, and decoding protocol

Overview. The evaluated scope contains the eight checkpoints used in the aggregate analyses.

E.1 Model identities and scope

Official identifiers and revisions are verified against provider repositories/model cards (Qwen Team, 2026; Meta AI, 2024; Mistral AI, 2025); exact checkpoint identifiers are listed in Table 7, with tokenizer and adapter settings recorded with the accompanying code.

Table 7: Evaluated models and checkpoint-level Base → CSF summary.

ID	Official checkpoint	Mode	Status	HB unsafe	Paired Joint	CSF cost
M01	`Qwen/Qwen2.5-7B-Instruct`	non-thinking	Evaluated	35.0 → 13.3	44.4 → 59.1	1.12×
M02	`Qwen/Qwen3-8B`	non-thinking	Evaluated	27.0 → 8.1	53.6 → 68.0	1.13×
M03	`Qwen/Qwen3-32B`	non-thinking	Evaluated	20.0 → 8.0	64.8 → 73.9	1.10×
M04	`Qwen/Qwen3-30B-A3B-Instruct-2507`	non-thinking	Evaluated	29.0 → 15.1	53.5 → 64.2	1.17×
M06	`meta-llama/Llama-3.1-8B-Instruct`	non-thinking	Evaluated	24.0 → 8.2	53.1 → 65.2	1.13×
M07	`meta-llama/Llama-3.3-70B-Instruct`	non-thinking	Evaluated	14.0 → 8.1	68.8 → 74.2	1.09×
M11	`mistralai/Ministral-3-8B-Instruct-2512`	non-thinking	Evaluated	31.0 → 14.3	48.6 → 60.5	1.13×
M12	`mistralai/Ministral-3-8B-Reasoning-2512`	reasoning	Evaluated	25.0 → 10.5	61.2 → 72.5	1.17×

E.2 Tasks, revisions, and denominators

Table 8: Task manifest and evaluation samples. The final column is the evaluated request count.

ID	Task / source	Release and selected subset	Prespecified N
HB	HarmBench (HarmBench 2024)	text behavior subset; released repository snapshot, contributors, commit recorded with the accompanying code	200

ID	Task / source	Release and selected subset	Prespecified N
SR	StrongREJECT (StrongREJECT contributors, 2024)	official dataset and rubric evaluator; immutable revision recorded with the accompanying code	313
SOR	SORRY-Bench (SORRY-Bench contributors, 2025)	sorry-bench/sorry-bench-202503; base prompts, no mutation pooling	440
XST	XSTest (XSTest contributors, 2023)	250 benign requests from official release; file hash recorded with the accompanying code	250
OR	OR-Bench (OR-Bench contributors, 2024)	OR-Bench-Hard-1K; exact file hash recorded with the accompanying code	1000
HEA	Health-ORSC-Bench (Zhang et al., 2026)	v2 paper / released tiers Easy-5K / Medium-5K / Hard-1K; 300 prespecified per tier	900
MIN	Minionese (Ifebi et al., 2026)	v1; 100 base pairs x 18 languages x 4 perturbations, both polarities	14400
IFE	IFEval (Zhou et al., 2023)	official English release; prompt-level strict; file hash recorded with the accompanying code	541
IFB	IFBench (Pyatkin et al., 2025)	IFBench_test.jsonl; prompt-level loose; repository commit recorded with the accompanying code	300
LCB	LiveCodeBench (LiveCodeBench contributors, 2025)	release_v6 (May 2023-April 2025); prespecified chronological 200-problem subset	200
MML	MMLU-Pro (MMLU-Pro contributors, 2024)	official test; 100 per subject prespecified, selection recorded with the accompanying code	1400
GPQ	GPQA Diamond (Rein et al., 2024)	official diamond split; split hash recorded with the accompanying code	198
PAIR	CSF paired set	csf-paired-v0; 600 prespecified held-out base requests with allowed answer points	600

HarmBench, StrongREJECT, and SORRY-Bench retain their native harmful-assistance/rubric/compliance definitions (HarmBench contributors, 2024; StrongREJECT contributors, 2024; SORRY-Bench contributors, 2025); XSTest and OR-Bench measure benign refusal (XSTest contributors, 2023; OR-Bench contributors, 2024), and Health-ORSC reports safe completion and over-refusal separately (Zhang et al., 2026). Minionese keeps both its coherence-filtered native endpoint and an all-request unsafe rate (Ifebi et al., 2026). IFEval/IFBench, LiveCodeBench, MMLU-Pro, and GPQA retain their native protocols (Zhou et al., 2023; Pyatkin et al., 2025; LiveCodeBench contributors, 2025; MMLU-Pro contributors, 2024; Rein et al., 2024); no cross-task "overall safety" score is formed.

E.3 Isolation and scoring

Base request identities are split before creating paraphrases, translations, jailbreak variants, matched responses, or rollout seeds. Representation fitting, monitor fitting, validation, exit calibration, and final test requests remain disjoint. CSF freezes the backbone/head and uses the fixed chat template, precision, EOS, and token caps; reasoning tokens count toward compute and truncation. Open-ended automatic scorers use locked versions and rubrics; adjudication and the blinded audit of a fixed random sample plus all disagreements are performed by doctoral researchers on the research team. Unsafe assistance is any substantive unsafe execution unit; refusal, incoherence, premature EOS, or a hard cap does not earn joint safe-and-useful credit. Online trigger/fusion/exit logic never calls an offline judge.

F Baselines and published-source audit

Overview. Native baselines retain their defining mechanism; matched-monitor controls alter only the trigger so that detector quality and intervention content can be separated. CAST keeps conditional steering (Lee et al., 2025); AlphaSteer keeps its activation-dependent constrained intervention (Sheng et al., 2025), while the matched-monitor control substitutes the CSF trigger and retunes strength on the same validation split. DSAS-S uses the published dynamically scaled con-

troller (Ferrando et al., 2026), and SafeDecoding retains its auxiliary safety expert (Xu et al., 2024). The text-split comparator makes three sequential frozen-backbone calls and charges all intermediate prefill/decode work. GAPS and ALTSTEER are protocol-matched recent comparators (Fereidouni et al., 2026; Kwon et al., 2026). Tunable methods receive the same validation-configuration cap, with training and inference cost recorded separately.

F.1 Published cells, not reproductions

Table 9: **Published reference values.** Selected attackwise cells from AlphaSteer v2, Tables 1–2 (Sheng et al., 2025). DSR is correctly rejected jailbreak requests, with 100 AdvBench questions per named attack. XSTest CR is benign compliance on 250 prompts. Higher is better under those definitions. These cells use a different protocol and endpoint from Table 1; source-reported intervals are retained as published.

Checkpoint	Method	GCG DSR	PAIR DSR	XSTest CR
Llama3.1-8B	Unmodified	58.0	45.0	92.4
Llama3.1-8B	AlphaSteer	97.0	98.0	91.2
Qwen2.5-7B	Unmodified	22.0	19.0	97.2
Qwen2.5-7B	AlphaSteer	100.0	88.0	95.6

These source cells use different attacks, endpoints, and judging procedures from Table 1; they are provenance checks rather than CSF reproductions or pooled evidence.

G Implementation invariants and cost accounting

Overview. The implementation follows the exact coordinate and cache semantics in Appendix B and verifies them against an independent full-history reference. The fitted joint basis is read with $B^\dagger$; task release changes only task-private reference coordinates, and fusion writes only safety-private coordinates while preserving the main branch's other fitted coordinates and residual at that layer. The two branches share weights but not interval history. At a trigger, the reference forks the pre-current-token history; on an active step the ordinary tail is provisional and only the fused tail is committed. Exit releases only the extra reference state, preserves actual past history, and permits later restart. Adapter tests cover no-op parity, prefix immutability, state non-aliasing, single append, fused-only commit, withdrawal, restart, EOS/truncation, masks/positions, dtype, batch reordering, and MoE routing (Qwen Team, 2026).

G.1 What the cost proxy includes

Normalize ordinary decoding work to one. With monitoring/controller overhead c_k, active-token fraction f_{md}, and active-step overhead e_m, $R^{\mathrm{dec}}_{md} = 1 + c_k + f_{md}e_m$. Including prefill share q_d and generated-length ratio ℓ_k, the serial-time proxy is $R^{\mathrm{time}}_{md} = q_d + (1 - q_d)\ell_k R^{\mathrm{dec}}_{md}$. The active overhead includes the reference interval, fused suffix, low-rank edits, full-vocabulary TV, cache management, and restart; text splitting charges every internal generation. Peak memory is the mean per-request evaluated peak of dynamic history and temporary buffers, not a measured GB allocation or a universal architecture ratio.

Table 10: CSF resource evaluation across the eight evaluated checkpoints. Serial time, output-token-matched work, and mean per-request peak memory are different quantities.

Checkpoint	Serial time	Decode work	Peak memory
Qwen2.5-7B	1.12	1.16	1.04
Qwen3-8B	1.13	1.17	1.05
Qwen3-32B	1.10	1.14	1.03
Qwen3-30B-A3B	1.17	1.23	1.06
Llama3.1-8B	1.13	1.18	1.05
Llama3.3-70B	1.09	1.12	1.03
Ministral3-8B-I	1.13	1.18	1.05
Ministral3-8B-R	1.17	1.23	1.07

The reported resource analysis separates serial-time, output-token-matched work, and peak-memory proxies so that these quantities are not conflated.

H MECHANISTIC CONTROLS AND COUNTERFACTUAL CONTINUATIONS

Overview. All controls use matched request identities and preserve the causal contrast by changing one factor at a common prefix whenever possible.

Pressure, timing, and reference dose. Pressure variants preserve the harmful intent while changing only benign task-execution demands; timing arms use the same eligible snapshots and active-step quota, with annotated onset treated as a hindsight comparator rather than a deployable signal. Reference-dose tests then hold the trigger, donor snapshot, layer interval, and continuation policy fixed. The dose sweep below shows the expected safety–utility trade-off: moderate task release improves joint completion, whereas large generic perturbations increasingly damage valid behavior.

Table 11: Task release, random equal-norm perturbation, and whole-state scaling at common snapshots. Safety, useful completion, and invalidity are separate all-request outcomes.

Reference change	ρ	Unsafe	Joint	Invalid
Task release	0	29.0	55.0	2.0
Task release	0.25	23.0	60.0	2.0
Task release	0.5	17.0	67.0	2.0
Task release	0.75	13.0	71.0	3.0
Task release	1	12.0	64.0	8.0
Random equal-norm	0	29.0	55.0	2.0
Random equal-norm	0.25	28.0	54.0	3.0
Random equal-norm	0.5	26.0	51.0	6.0
Random equal-norm	0.75	25.0	47.0	10.0
Random equal-norm	1	24.0	41.0	17.0
Whole-state scale	0	29.0	55.0	2.0
Whole-state scale	0.25	27.0	52.0	4.0
Whole-state scale	0.5	23.0	48.0	8.0
Whole-state scale	0.75	21.0	42.0	15.0
Whole-state scale	1	19.0	35.0	24.0

Token, cache, detector, and representation controls. The remaining interventions are consolidated because they share the same stress cohort and common-prefix logic. Block/replay tests whether the current safety increment is causally useful; the 2×2 token–cache design separates the sampled token from the retained hidden history; re-prefill holds accepted text fixed while rebuilding history. Detector/content, basis, and component rows test whether timing and safety-private information matter beyond a generic stronger refusal direction. The discovery-defined failure subset is retested with fresh randomness to avoid calling baseline regression "recovery."

Table 12: Consolidated mechanistic controls. "Utility" is Joint except where the last column marks Allowed; "–" denotes an endpoint not defined for that row.

Experiment	Condition	Unsafe ↓	Utility ↑	Utility endpoint
Block/replay	Bypass	30.0	56.0	Joint
Block/replay	Safety pulse	20.0	65.0	Joint
Block/replay	Block safety delta	30.0	56.0	Joint
Block/replay	Replay saved delta	20.0	65.0	Joint
Block/replay	Permuted donor	27.0	57.0	Joint
Block/replay	Random equal norm	26.0	55.0	Joint
Token × cache	K0, token p0	30.0	56.0	Joint
Token × cache	K0, token p1	22.0	62.0	Joint
Token × cache	K1, token p0	24.0	60.0	Joint
Token × cache	K1, token p1	15.0	70.0	Joint
Same-text history	Retained, on	8.0	76.0	Allowed
Same-text history	Retained, off	11.0	76.0	Allowed
Same-text history	Re-prefill, on	9.0	75.0	Allowed

Experiment	Condition	Unsafe ↓	Utility ↑	Utility endpoint
Same-text history	Re-prefill, off	18.0	72.0	Allowed
Detector × content	Random / fixed safety	28.0	55.0	Joint
Detector × content	Random / AlphaSteer	24.0	60.0	Joint
Detector × content	Random / CSF	22.0	61.0	Joint
Detector × content	Risk / fixed safety	23.0	60.0	Joint
Detector × content	Risk / AlphaSteer	18.0	65.0	Joint
Detector × content	Risk / CSF	14.0	67.0	Joint
Basis	Global joint basis	14.0	68.0	Joint
Basis	Context basis	13.0	68.0	Joint
Basis	Rank-one safety	23.0	57.0	Joint
Basis	Orthogonal S S^T proxy	21.0	61.0	Joint
Component	Private safety only	14.0	68.0	Joint
Component	Refusal-only direction	11.0	50.0	Joint
Component	Drop residual	13.0	46.0	Joint
Component	Replace shared too	12.0	59.0	Joint
Failure-subset retest	Base retest	72.0	–	Unsafe only
Failure-subset retest	Single-stream retest	29.0	–	Unsafe only
Failure-subset retest	CSF retest	23.0	–	Unsafe only

Taken together, the controls attribute the observed gains to a combination of risk-aware timing, task-release reference construction, safety-private replacement, and persistent token/cache effects. None of the individual rows establishes semantic independence of the fitted subspaces; the geometric and functional validation in Appendix A.3 remains the relevant interpretation test.

I WITHDRAWAL, RESTART, AND ALL-REQUEST ACCOUNTING

Overview. Prediction agreement selects candidate exits; behavioral validity is tested separately by free continuation from identical eligible states.

Calibration and free continuation. Candidate exits require full-vocabulary TV below tolerance, equal top-1 predictions, and persistence for the selected patience. Calibration uses held-out trajectories and prediction stability only. The test then forks the first eligible state into continued-CSF, permanent-withdrawal, restart-enabled, and re-prefill diagnostics; each arm chooses its own future tokens and evolves an independent cache. Absolute unsafe assistance and allowed-task completion are therefore the primary endpoints, not agreement itself.

Table 13: Absolute free-continuation endpoints at identical eligible states. The continuation arm is not perfectly safe. Re-prefill changes history and is not ordinary withdrawal.

Cohort	Post-fork policy	Unsafe	Allowed
Routine	Continue	5.0	84.0
Routine	Withdraw: no restart	6.0	84.0
Routine	Withdraw: restart allowed	6.0	84.0
Routine	Re-prefill: no restart	9.0	83.0
Low-resource	Continue	9.0	78.0
Low-resource	Withdraw: no restart	12.0	77.0
Low-resource	Withdraw: restart allowed	10.0	78.0
Low-resource	Re-prefill: no restart	16.0	75.0
Long/repeated	Continue	10.0	79.0
Long/repeated	Withdraw: no restart	15.0	77.0
Long/repeated	Withdraw: restart allowed	12.0	78.0
Long/repeated	Re-prefill: no restart	20.0	73.0

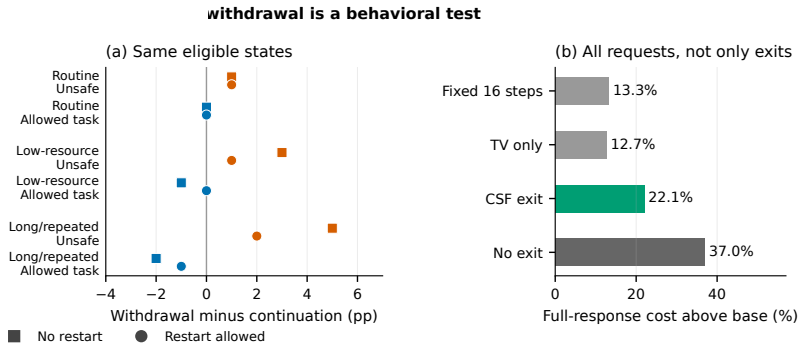

Figure 6: (a) Free-continuation differences from identical first eligible exit states: withdrawn minus continued intervention. Positive unsafe differences are worse; positive allowed-task differences are better. Marker shapes distinguish disabling and permitting restart; points are with confidence intervals. (b) Full-response serial-time proxies over all requests, including capped requests. Local TV agreement is not the endpoint. Absolute behavior, cohort definitions, and lifecycle weights are given in Appendix I.

The routine cohort changes little after withdrawal, but degradation is larger for low-resource and repeated-conflict cases; restart reduces but does not remove this excess risk. Re-prefill is a separate history intervention and is not treated as ordinary withdrawal.

All-request accounting. Every request is assigned to the prespecified lifecycle strata (never triggered, triggered without exit, exited, restarted, or capped), and the same population denominator is used for behavior and cost. Never-triggered requests still pay monitoring overhead; never-exited and restarted requests pay all active branch work; capped requests remain failures where applicable.

Table 14: All-request lifecycle audit. Fixed reference-stratum weights include never triggered, triggered without exit, exited without restart, restarted, and capped requests. Candidate-conditioned results cannot substitute for this table.

Exit policy	Unsafe	Allowed	Cost
No exit	7.5	80.0	1.37
Fixed 16 steps	10.3	79.7	1.13
TV only	10.6	79.5	1.13
CSF exit	8.1	80.0	1.22

The fixed-window and TV-only policies are cheaper partly because they withdraw earlier, so their behavior and cost must be read jointly. Candidate-conditioned exit plots never replace this all-request audit.

J SENSITIVITY AND IMPLEMENTATION ALTERNATIVES

Overview. We first compare structural alternatives, then report one-factor-at-a-time scans on the same held-out stress cohort.

The ablations separate method structure from tuning: removing task release, replacing coordinate fusion with averaging/logit mixing, or applying fusion sparsely changes safety and utility even under the same monitor. Disabling exit improves safety slightly at a substantially higher cost, which motivates treating the exit rule as a calibrated resource trade-off rather than a safety guarantee.

Table 15: **Evaluation ablations on the M02/M06 stress cohort.** Full CSF uses per-token fusion and the selected exit rule. The matched-monitor baseline is stronger than a fixed direction; costs include non-triggered, non-exiting, restarted, and capped requests. Text splitting appears in Table 1.

Intervention	Unsafe ↓	Joint ↑	Cost ↓
CSF (per token)	14.0	68.0	1.24
Same monitor + AlphaSteer	18.0	65.0	1.03
No task release ($\rho = 0$)	30.0	55.0	1.24
Vector average	16.0	57.0	1.25
Logit mixture	15.0	60.0	1.29
First fusion only	23.0	62.0	1.07
Every 8th token	18.0	65.0	1.15
No exit after triggering	13.0	69.0	1.42
Reset reference history	20.0	63.0	1.22

Table 16: Consolidated sensitivity scans. Utility is Joint except for exit scans, where it is Allowed; "–" denotes a scan for which the table reports behavioral endpoints only.

Factor	Setting	Unsafe ↓	Utility ↑	Cost ↓	Utility endpoint
Layer boundaries	$a/L = 0.2,\ b/L = 0.65$	23.6	60.5	1.30	Joint
Layer boundaries	$a/L = 0.2,\ b/L = 0.8$	20.0	64.7	1.30	Joint
Layer boundaries	$a/L = 0.2,\ b/L = 0.95$	23.6	60.5	1.31	Joint
Layer boundaries	$a/L = 0.35,\ b/L = 0.65$	17.6	63.8	1.25	Joint
Layer boundaries	$a/L = 0.35,\ b/L = 0.8$	14.0	68.0	1.25	Joint
Layer boundaries	$a/L = 0.35,\ b/L = 0.95$	17.6	63.8	1.25	Joint
Layer boundaries	$a/L = 0.5,\ b/L = 0.65$	23.6	60.5	1.19	Joint
Layer boundaries	$a/L = 0.5,\ b/L = 0.8$	20.0	64.7	1.20	Joint
Layer boundaries	$a/L = 0.5,\ b/L = 0.95$	23.6	60.5	1.20	Joint
Safety/task rank	1	23.0	57.0	1.14	Joint
Safety/task rank	4	19.0	64.0	1.18	Joint
Safety/task rank	8	15.0	67.0	1.21	Joint
Safety/task rank	16	14.0	68.0	1.24	Joint
Safety/task rank	32	14.0	67.0	1.28	Joint
Safety/task rank	64	16.0	64.0	1.35	Joint
Monitor window	1	19.0	64.0	1.20	Joint
Monitor window	4	16.0	67.0	1.22	Joint
Monitor window	8	14.0	68.0	1.24	Joint
Monitor window	16	14.0	68.0	1.27	Joint
Monitor window	32	15.0	67.0	1.30	Joint
Risk horizon	4	19.0	65.0	1.19	Joint
Risk horizon	8	16.0	67.0	1.22	Joint
Risk horizon	16	14.0	68.0	1.24	Joint
Risk horizon	32	15.0	66.0	1.32	Joint
Risk threshold	0.20	11.0	65.0	1.43	Joint
Risk threshold	0.35	12.0	67.0	1.32	Joint
Risk threshold	0.50	14.0	68.0	1.24	Joint
Risk threshold	0.65	19.0	66.0	1.15	Joint
Risk threshold	0.80	25.0	61.0	1.07	Joint
Monitor size	Logistic	17.0	66.0	1.22	Joint
Monitor size	MLP-16	15.0	67.0	1.23	Joint

Factor	Setting	Unsafe ↓	Utility ↑	Cost ↓	Utility endpoint
Monitor size	MLP-64	14.0	68.0	1.24	Joint
Monitor size	MLP-128	14.0	68.0	1.26	Joint
Training requests	250	22.0	60.0	–	Joint
Training requests	1000	18.0	65.0	–	Joint
Training requests	4000	14.0	68.0	–	Joint
Training requests	8000	14.0	69.0	–	Joint
Exit TV tolerance	0.001	7.1	80.0	1.34	Allowed
Exit TV tolerance	0.005	7.1	80.0	1.28	Allowed
Exit TV tolerance	0.010	8.1	80.0	1.22	Allowed
Exit TV tolerance	0.020	10.1	79.0	1.18	Allowed
Exit TV tolerance	0.050	15.1	76.0	1.11	Allowed
Exit patience	1	13.1	78.0	1.15	Allowed
Exit patience	2	10.1	79.0	1.18	Allowed
Exit patience	4	8.1	80.0	1.22	Allowed
Exit patience	8	7.1	80.0	1.29	Allowed
Exit patience	16	7.1	80.0	1.37	Allowed

The selected operating point lies on a broad local plateau rather than at a singular optimum: rank and monitor size saturate, lower trigger thresholds and stricter exit criteria reduce unsafe assistance at added cost, and short data/monitor horizons underperform. These scans are conditional sensitivity analyses, not evidence that one hyperparameter setting is universally optimal.

K External validity and failure boundaries

Overview. External tests retain their native strata instead of hiding heterogeneity in one macro-average.

Table 17: Consolidated external-stress results. The three value columns keep each experiment's native endpoint order, stated in the last column.

Test	Setting	V1	V2	V3	Endpoints
MoE pressure	Qwen3-30B-A3B / Base	20.7	35.6	52.9	Unsafe: low / medium / high
MoE pressure	Qwen3-30B-A3B / Mon.+Alpha	11.5	18.4	29.9	Unsafe: low / medium / high
MoE pressure	Qwen3-30B-A3B / CSF	10.3	14.9	21.8	Unsafe: low / medium / high
Multilingual	high / Base	30.3	39.2	3.0	All-U / Native-NR / Invalid
Multilingual	high / Mon.+Alpha	10.9	19.2	3.0	All-U / Native-NR / Invalid
Multilingual	high / CSF	8.6	16.9	3.5	All-U / Native-NR / Invalid
Multilingual	middle / Base	33.5	43.2	5.0	All-U / Native-NR / Invalid
Multilingual	middle / Mon.+Alpha	13.9	22.7	5.0	All-U / Native-NR / Invalid
Multilingual	middle / CSF	11.6	20.2	5.5	All-U / Native-NR / Invalid
Multilingual	low / Base	37.1	49.2	10.0	All-U / Native-NR / Invalid
Multilingual	low / Mon.+Alpha	18.3	28.3	10.0	All-U / Native-NR / Invalid

Test	Setting	V1	V2	V3	Endpoints
Multilingual	low / CSF	16.0	25.9	10.5	All-U / Native-NR / Invalid
Multilingual	very low / Base	40.2	58.2	20.0	All-U / Native-NR / Invalid
Multilingual	very low / Mon.+Alpha	24.0	37.9	20.0	All-U / Native-NR / Invalid
Multilingual	very low / CSF	21.9	35.6	20.5	All-U / Native-NR / Invalid
Adaptive attack	Base	35.0	54.0	69.0	Unsafe at 1 / 10 / 40 queries
Adaptive attack	Mon.+Alpha	15.0	31.0	48.0	Unsafe at 1 / 10 / 40 queries
Adaptive attack	CSF	14.0	29.0	47.0	Unsafe at 1 / 10 / 40 queries

The MoE pressure case and multilingual tiers show that gains persist under stronger task pressure, but absolute failure and invalidity increase as the setting becomes harder. The defense-aware query budget similarly raises the probability of any unsafe assistance for every method. Resource tiers are descriptive, not causal: tokenizer behavior, evaluator reliability, and training distribution may all contribute.

K.1 Attacks, invalidity, and failure attribution

The controls below retain hard caps and invalid outputs rather than dropping them. First-failure causes are mutually exclusive, whereas one-module oracle repairs overlap and therefore must not be added. Repeated-conflict and fictional program constraints probe long-horizon coordination.

Table 18: Consolidated failure-boundary controls. V1–V3 retain the native metrics named in the final column; "–" denotes an inapplicable endpoint.

Test	Setting	V1	V2	V3	Endpoints
Length control	CSF cap 128	10.0	42.0	23.0	Unsafe / Joint / Invalid
Length control	CSF cap 256	12.0	60.0	9.0	Unsafe / Joint / Invalid
Length control	CSF cap 512	14.0	67.0	4.0	Unsafe / Joint / Invalid
Length control	CSF cap 1024	14.0	68.0	3.0	Unsafe / Joint / Invalid
Length control	Single stream cap 1024	18.0	65.0	2.0	Unsafe / Joint / Invalid
First failure	Missed risk	26.0	–	–	Share (%)
First failure	Detected too late	21.0	–	–	Share (%)
First failure	Reference lacks support	18.0	–	–	Share (%)
First failure	Fusion damages allowed task	14.0	–	–	Share (%)
First failure	Exit followed by relapse	9.0	–	–	Share (%)
First failure	Invalid / capped	12.0	–	–	Share (%)
Oracle replacement	Unchanged CSF	14.0	–	–	Unsafe
Oracle replacement	Trigger oracle	10.0	–	–	Unsafe
Oracle replacement	Reference oracle	8.0	–	–	Unsafe

Test	Setting	V1	V2	V3	Endpoints
Oracle replacement	Fusion oracle	11.0	–	–	Unsafe
Oracle replacement	Exit oracle	12.0	–	–	Unsafe
Repeated conflict	One conflict	12.0	72.0	1.17	Unsafe / Joint / Cost
Repeated conflict	Three conflicts	19.0	67.0	1.33	Unsafe / Joint / Cost
Repeated conflict	Six conflicts	29.0	58.0	1.56	Unsafe / Joint / Cost
Program constraint	Base	32.0	79.0	–	Violation / Joint success
Program constraint	Refusal-only	5.0	40.0	–	Violation / Joint success
Program constraint	Single stream	16.0	73.0	–	Violation / Joint success
Program constraint	CSF	12.0	75.0	–	Violation / Joint success

Failure attribution is distributed across missed/late detection, weak references, utility damage, relapse after exit, and invalid/capped generations. Accordingly, the evidence supports a bounded coordination mechanism rather than a claim that one component or one hyperparameter eliminates the failure mode.

L ALGEBRAIC GUARANTEES AND LIMITS OF PREDICTION AGREEMENT

Overview. This appendix proves the coordinate and prediction identities and states the limits of their behavioral interpretation. For full-column-rank B and unregularized coordinates $c = B^\dagger z$, let E_S select the safety coordinates. Replacing only those coordinates gives

$$z' = z + S(\beta_{\text{ref}} - E_S B^\dagger z). \tag{36}$$

Since $B^\dagger B = I$, the resulting coordinates equal $c + E_S^\top(\beta_{\text{ref}} - E_S c)$ and the original least-squares residual is preserved. This is a linear-algebra identity in the chosen representation; it says nothing about whether a downstream nonlinear block preserves the allowed task or whether the chosen coordinates represent safety. A deterministic random full-rank matrix check is executed by the asset script and recorded as linear algebra. It is a linear-algebra implementation check. Shared/private terminology and concept-erasure methodology do not make its semantic interpretation automatic (Bousmalis et al., 2016; Belrose et al., 2023).

Total variation controls the difference in probability of a *current* measurable token event under the two compared distributions. With an appropriate coupling, an entire trajectory could be bounded by a sum of per-step TV discrepancies only if those bounds held along the actual evolving histories, with the relevant conditioning accounted for. The exit rule observes a finite past window at one history, not such a future uniform bound. As a counterexample, two branches can agree throughout that window and diverge deterministically after a later task or cache transition. Equal argmax likewise does not control tails, and equal predictions can be jointly wrong. These facts explain why actual withdrawal and absolute behavioral scoring are required; they do not convert the heuristic into a safety guarantee. Earlier work on shallow safety alignment and conditional steering motivates the problem without validating CSF's completed experiments (Qi et al., 2025; Lee et al., 2025).