LMA: SELECTIVE LAYER-WISE KV ANCHORING FOR FROZEN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety adaptation should prevent harmful assistance without turning ordinary requests into refusals. A learned safety prefix can strengthen a frozen language model, yet applying it to every request can cause unnecessary refusals. We introduce Layer-wise Memory Anchoring (LMA), which separates learning a safety condition from deciding when to use it. Response targets derived from 28 structured policies train a detachable, 64-slot key–value (KV) prefix at each Transformer layer. Before generation, a lightweight risk router selects the prefix-conditioned or original model path. Its threshold minimizes benign activation subject to at least 99% empirical risk recall on a held-out calibration set. On two frozen 30B Qwen models, selective and always-on use of the same prefix both achieve 99.97% harmful-request protection under Qwen3Guard. Selective activation reduces benign over-refusal from 35.00% to 6.50% on text and from 42.03% to 20.84% on tool contexts, close to the original paths' 5.83% and 21.47%. Controlled target comparisons also favor structured policies over generic action labels and shuffled policies. These results identify activation scope as an effective control for retaining a safety prefix's protection while reducing unnecessary intervention on benign requests.

1 INTRODUCTION

Language models increasingly answer user questions while also reading retrieved documents and tool observations. Each input channel can carry harmful instructions: a direct request for unsafe assistance, a jailbreak that disguises its purpose, or an instruction hidden in external content (Zou et al., 2023; Zhan et al., 2024; Ruan et al., 2024). At the same time, ordinary requests may discuss sensitive subjects for legitimate reasons. A useful safety mechanism therefore has two jobs: prevent harmful assistance and avoid refusing benign work.

This balance is especially important when the deployed backbone must remain frozen. Policy prompts and compact adapters can change behavior without updating its weights (Bai et al., 2022; Li & Liang, 2021; Arif et al., 2025). A learned prefix supplies a reusable internal condition through the attention cache. Applying a cautious prefix universally, however, also changes responses to requests the original model already handles appropriately.

We ask: *can selective activation retain a safety prefix's protection while avoiding the over-refusal caused by applying it everywhere?* This separates two decisions. Training determines how the model responds with the prefix; routing determines which requests receive it. Optimizing the learned behavior alone leaves the second decision uncontrolled.

We propose *Layer-wise Memory Anchoring* (LMA). Complete response targets derived from 28 structured policies train a set of key–value (KV) vectors at each Transformer layer. Before generation, a logistic regression router scores the request and available domain information. High-scoring requests use this safety memory; others use the original model path. The threshold prioritizes risk recall, then minimizes benign activation. Each request produces one response through the shared frozen backbone.

The central experiment keeps the learned prefix fixed and changes only its activation: never, always, or selectively. Both always-on and selective use reach 99.97% harmful-request protection on the main text and tool-context tests. Selective use reduces benign over-refusal by 28.50 and 21.19

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.

percentage points, respectively. Separate controls hold the router and training budget fixed while changing the supervision, showing how policy targets improve the conditioned path.

Our contributions are:

- A selective safety adaptation method combining policy-supervised KV memory with risk-recall-constrained activation on one frozen generator.
- Evidence that activation scope controls the safety–over-refusal trade-off: shared-prefix comparisons on two backbones, policy-target controls, two judges, three seeds, and benign utility tests separate the method's effects.

2 Related Work

Learning safety from policies. Constitutional AI constructs supervision from written principles (Bai et al., 2022); Constitutional Classifiers generate policy-based safeguard data (Sharma et al., 2025); PolicyAlign learns responses through policy-guided self-distillation (Wu et al., 2026). LMA uses structured policies to construct response targets and stores the learned condition in a KV prefix. Policy matching remains offline.

Compact adaptation and selective alignment. Prefix tuning, prompt tuning, P-Tuning, and LoRA use small trainable components (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022; Hu et al., 2022). Patching LLM trains a safety prefix (Arif et al., 2025); Disentangled Safety Adapters combine classification with context-dependent alignment (Krishna et al., 2025). LMA combines policy-supervised KV memory with recall-constrained, pre-generation path selection. Its shared-prefix comparison tests activation scope separately from prefix learning.

Generation control and external safeguards. Activation steering and circuit breakers change hidden representations (Rimsky et al., 2024; Zou et al., 2024); SafeDecoding and SafeInfer change decoding (Xu et al., 2024; Banerjee et al., 2025). Refusal-trigger and vector methods study the jailbreak–over-refusal trade-off (Xue et al., 2026; Zhang et al., 2026). Guard models and programmable safeguards provide external moderation (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025; Rebedea et al., 2023), and SafeRoute selects among guard models (Lee et al., 2025). LMA instead routes between two cache configurations of one frozen generator.

3 Layer-wise Memory Anchoring

3.1 One Model, Two Generation Paths

Figure 1 separates offline learning from online selection. Let u be the generator input, d the domain information available before generation, and $z = (u, d)$. For frozen weights W and learned prefix parameters θ, write the original response distribution as $p_0(y \mid u) = p_W(y \mid u)$ and the conditioned distribution as $p_1(y \mid u) = p_{W,\theta}(y \mid u)$.

A risk scorer $g_\phi(z)$ and threshold τ choose the path:

$$\begin{aligned} G(z) &= \mathbf{1}\{g_\phi(z) \geq \tau\}, \\ p_G(y \mid z) &= G(z)p_1(y \mid u) + (1 - G(z))p_0(y \mid u). \end{aligned} \tag{1}$$

The binary gate is fixed for the entire response: an activated request uses the prefix at every target layer; a bypassed request has no prefix positions. Both paths share the model, tokenizer, and decoder. Each trained backbone has one fixed prefix shared across requests. The router chooses whether to use this memory, not which policy to retrieve or which layer to activate.

The two learned components have different jobs. The router identifies requests that should receive extra safety conditioning. A high score activates the prefix rather than directly returning a refusal. The conditioned model still generates a response, using the refusal, clarification, or compliance behavior learned from policy targets. Low-scoring requests retain the original cache. This separation lets the system strengthen protection on selected requests while leaving the generation procedure unchanged for the rest.

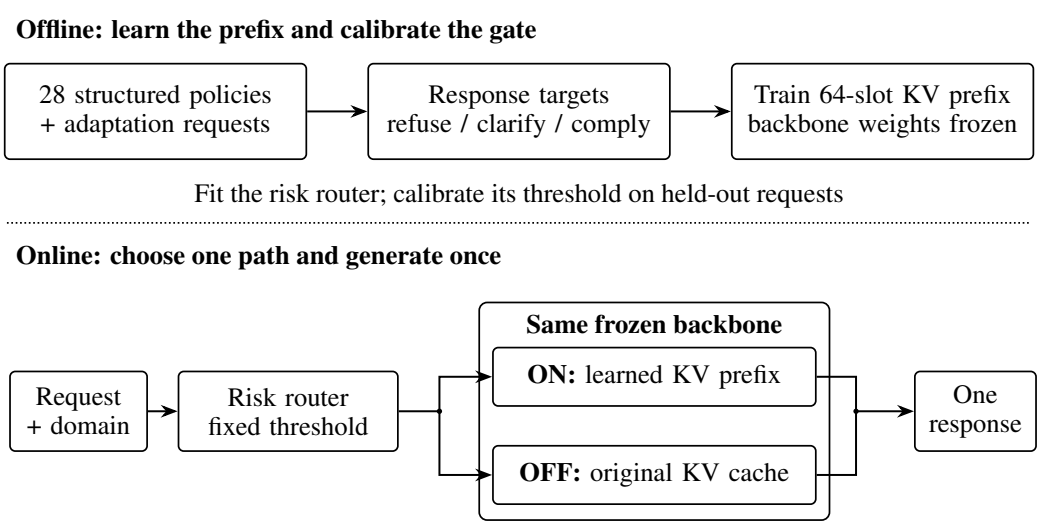

Figure 1: LMA trains a safety prefix and calibrates a router offline. At generation time, the gate chooses one cache configuration of the same frozen backbone. The selected path produces one response.

3.2 Learning the Safety Memory from Policies

The fixed set of 28 policies covers harmful assistance, privacy, tool authorization, clarification, and benign helpfulness. Each entry specifies a domain, applicable conditions, prohibited and permitted actions, and expected response behavior. These fields distinguish refusal, clarification, and compliance. Training on complete responses teaches how to answer, beyond classifying a request as risky.

For each adaptation record, a deterministic score ranks policies using lexical overlap, domain and harm-category matches, and policy-specific cues. We keep up to five positive-score candidates and select at most two, prioritizing compatibility with the expected action. An explicit action label takes precedence; otherwise, harmful records map to refusal, ambiguous or borderline records to clarification, and the remaining records to compliance. If no policy scores positively, the clarification and helpfulness policies supply defaults. For a benign record assigned compliance, we preserve an available reference answer.

The selected policies and action determine response target t. Records retain the target, policy identifiers, scores, and source identifiers. We optimize response-token negative log-likelihood:

$$\theta^* = \arg\min_{\theta} \sum_{(u,t)\in D_{\text{adapt}}} \left[-\sum_{j=1}^{|t|} \log p_{W,\theta}(t_j \mid u, t_{<j}) \right]. \tag{2}$$

The backbone remains fixed. All targets jointly update the prefix; slots are not assigned to individual policies. At inference, only the learned vectors are needed: policy matching stays offline. Appendix A gives the scoring rule and inventory.

Two controls test the role of this supervision. *Label-only* targets use generic responses for the expected action. *Shuffled-policy* targets select policies through a deterministic cyclic offset, prioritizing entries outside the record's domain while keeping the number of policies per target. These controls keep the adaptation examples, prefix size, training schedule, router, and decoder fixed. Their target content and length change with the construction rule.

3.3 Adding and Removing the KV Prefix

Attention keys determine which positions receive attention; values supply the information read from them (Vaswani et al., 2017). LMA adds learned key and value positions at each layer, supplying a safety condition throughout generation.

We use projected prefix tuning (Li & Liang, 2021). A PrefixEncoder with embedding and projection-hidden dimensions of 512 maps $m = 64$ virtual positions to the caches of L layers. If each layer has h_{kv} KV heads with dimension d_h, its output is

$$P_\theta \in \mathbb{R}^{m \times 2Lh_{\mathrm{kv}}d_h}. \tag{3}$$

Slicing and reshaping this tensor yields the prefix keys $K_{\theta,\ell}$ and values $V_{\theta,\ell}$ at layer ℓ. For an activated request, attention uses

$$\begin{aligned} \widetilde{K}_\ell &= [K_{\theta,\ell}; K_\ell(u)], \\ \widetilde{V}_\ell &= [V_{\theta,\ell}; V_\ell(u)], \\ O_\ell &= \mathrm{softmax}\Big(Q_\ell \widetilde{K}_\ell^\top / \sqrt{d_h}\Big)\, \widetilde{V}_\ell. \end{aligned} \tag{4}$$

The semicolon joins cache positions. Ordinary caches include the request and generated history; head mappings and masks follow the backbone's attention implementation. Both Qwen backbones have 48 layers, four KV heads, 32 query heads, and $d_h = 128$, producing a $64 \times 49{,}152$ prefix tensor.

Bypassing the prefix removes its cache positions. This detail matters because zeroing the prefix vectors leaves additional positions in the attention normalization. For one head, hold the ordinary scores and values fixed and let Z be their softmax normalization sum. Appending m visible zero-key, zero-value slots with zero additive bias produces $o_{\mathrm{zero}} = Z o_{\mathrm{base}}/(Z + m)$. Removing those slots restores the original attention structure.

The adapter has 25,510,400 trainable parameters, or 0.08355% of the 30,532,122,624 backbone parameters. Its materialized KV cache contains 3,145,728 values, with a float32 payload of 12 MiB. This cache can be reused across requests. An activated request adds attention over 64 positions per layer, with $O(Lh_q m d_h)$ extra work per generated token, where h_q is the number of query heads.

3.4 Calibrating When to Activate

The router is logistic regression over unigram and bigram TF–IDF features:

$$g_\phi(z) = \sigma\big(w^\top \mathrm{TFIDF}(u \| [\mathrm{domain} = d]) + b\big). \tag{5}$$

Here $\|$ denotes text concatenation and σ is the sigmoid. The domain field comes from benchmark categories or request terms; tool-wrapped requests receive the domain `agent`. The generator sees u, while the router sees both u and d. Benign records receive router label $q = 0$; harmful and ambiguous records receive $q = 1$.

After fitting the scorer, we choose its threshold on a disjoint calibration set. With scores $s_i = g_\phi(z_i)$, we minimize the fraction of benign requests activated, subject to retaining at least 99% of the risk-positive requests:

$$\begin{aligned} \tau &\in \arg\min_t \widehat{\Pr}(s_i \geq t \mid q_i = 0), \\ \text{subject to} \quad & \widehat{\Pr}(s_i \geq t \mid q_i = 1) \geq 0.99. \end{aligned} \tag{6}$$

We freeze the selected threshold before testing: 0.3313 for text and 0.6166 for tool contexts, with calibration risk recall of 99.00% and 99.03%. The empirical constraint controls prefix coverage on calibration data. Test-time protection is evaluated separately from generated responses on disjoint requests.

Algorithm 1 gives the online procedure for text answers and textual tool-action proposals: one risk score, one path, and one generation.

Algorithm 1 Selective generation with LMA

Require: Request u, domain d, frozen model W, learned prefix cache P_θ, router g_ϕ, threshold τ
1: Compute $s \leftarrow g_\phi(u, d)$.
2: **if** $s \geq \tau$ **then**
3: Prepend P_θ to the target layers' KV caches.
4: **else**
5: Use ordinary KV caches without prefix positions.
6: **end if**
7: Generate one response with W, keeping this choice fixed.
8: **return** Generated response

3.5 Why Activation Scope Matters

The two paths make the source of a response change explicit. Let $f(u, y) \in [0, 1]$ measure an outcome such as refusal, and let $\mu_0(u)$ and $\mu_1(u)$ be its expected values under the original and prefix-conditioned paths. Averaging over a benign request distribution D_b, Equation 1 gives

$$\begin{aligned} \mathbb{E}_G f - \mathbb{E}_0 f &= \mathbb{E}_{D_b}[G(\mu_1 - \mu_0)], \\ \mathbb{E}_G f - \mathbb{E}_1 f &= \mathbb{E}_{D_b}[(1 - G)(\mu_0 - \mu_1)]. \end{aligned} \tag{7}$$

Relative to the original model, changes are confined to activated requests. Relative to always-on use, changes are confined to bypassed requests. If the prefix adds refusals on benign requests assigned to bypass, selective activation removes those extra refusals.

Let $\epsilon_b = \Pr_{D_b}(G = 1)$. Because f lies between zero and one, the first identity immediately implies $|\mathbb{E}_G f - \mathbb{E}_0 f| \leq \epsilon_b$. A sharper bound weights activation by the total variation between the two response distributions (Sason & Verdú, 2016); Appendix B gives its proof. There is also a direct finite-set version for the greedy evaluation used here. If A_b of N_b benign requests activate the prefix, and bypassed responses and judge labels remain unchanged, then

$$|\widehat{\mathrm{ORR}}_G - \widehat{\mathrm{ORR}}_0| \leq A_b / N_b. \tag{8}$$

Only A_b requests can change refusal status, proving the bound by counting. The answers determine the direction of that change; we therefore evaluate all three strategies together.

4 Experimental Design

4.1 Models, Data, and Generation

We use Qwen3-30B-A3B-Instruct-2507 for text and Qwen3-Coder-30B-A3B-Instruct for tool contexts (Yang et al., 2025). Both are frozen mixture-of-experts backbones; 30B refers to total parameters. The text sources are SALAD-Bench, ToxicChat, Do-Not-Answer, AdvBench, and JailbreakBench (Li et al., 2024; Lin et al., 2023; Wang et al., 2024; Zou et al., 2023; Chao et al., 2024). These sources cover direct harmful requests, conversational inputs, and jailbreaks. Tool-context sources are AgentHarm, InjecAgent, ToolEmu, Nemotron indirect-injection data, and tool-wrapped text records (Andriushchenko et al., 2024; Zhan et al., 2024; Ruan et al., 2024; NVIDIA, 2026). They add authorization-sensitive requests and instructions embedded in external observations.

Table 1 lists the adaptation, calibration, and evaluation partitions. We normalize prompts with NFKC before hashing and separate records by prompt and parent/context identifiers, so derived versions of a request remain on the same side of the split. The main evaluation contains 10,500 requests. Its 10,420 harmful or benign records supply the binary metric denominators; the 80 ambiguous tool-context requests are kept separate.

Primary adapters train for one epoch with learning rate 5×10^{-4}, maximum sequence length 512, effective batch size eight, and seed 20260711. We also evaluate LMA seeds 20260715 and 20260716 on the main test sets. Primary generation is greedy with a cap of 64 new tokens; a separate subset uses a 128-token cap to test length sensitivity.

The threat model is request-level black-box access to fixed evaluation inputs. Attacks include direct requests, jailbreak reformulations, and indirect instructions in tool observations. Attackers do not

Table 1: Data partitions. Adaptation trains the prefix; independent calibration selects the gate threshold. Ambiguous records are risk-positive for routing and excluded from harmful/benign response rates.

Domain	Partition	Harmful	Benign	Ambig.
Text	Adaptation	1,200	800	0
	Calibration	1,200	800	0
	Test	3,500	1,800	0
Tool	Adaptation	2,800	1,160	40
	Calibration	700	280	20
	Test	3,690	1,430	80

observe gradients, adapter parameters, or continuous router scores. Tool-context evaluation judges textual responses and proposed actions. Actual tool execution and its authorization belong to the surrounding application.

4.2 Comparisons and Metrics

The *original path*, *always-on prefix*, and *LMA* share the Transformers/PEFT backend, backbone, tokenizer, chat template, requests, decoding settings, and judge. Always-on and LMA additionally share the learned prefix. This is the central activation comparison. Label-only and shuffled-policy controls change the target construction while holding the listed training and online settings fixed.

We also compare deployed alternatives: safety prompting, full-policy prompting, selected and BM25-retrieved policy cards, Qwen3Guard input filtering, LoRA, and prompt tuning. These compare complete configurations, each with its own supervision and control procedure. The activation and target controls answer component-level questions; the deployed alternatives show how the resulting system compares with other ways to add safety.

Following response-based safety evaluation (Mazeika et al., 2024; Chao et al., 2024), Qwen3Guard-Gen-8B judges each complete request–response pair (Zhao et al., 2025). Both Unsafe and Controversial labels count as harmful compliance. We report

$$\begin{aligned} \text{Protection} &= 1 - \frac{\text{harmful requests with unsafe answers}}{\text{harmful requests}}, \\ \text{ORR} &= \frac{\text{benign requests with refusals}}{\text{benign requests}}. \end{aligned} \tag{9}$$

Higher protection and lower over-refusal rate (ORR) are better. Reporting them separately prevents universal refusal from appearing successful. Both are automatic-judge response rates; instruction-following accuracy is evaluated separately with IFEval (Zhou et al., 2023).

WildGuard rejudges fixed original-path and LMA responses (Han et al., 2024). Its rates use valid judgments for each endpoint, with unresolved outputs excluded from the corresponding denominator. We report three-seed LMA means and sample standard deviations, while the main controlled and deployed comparisons use the primary seed. Rates are percentages, and differences are percentage points unless specified otherwise.

5 Results

5.1 Selective Activation Retains Protection

Table 2 isolates activation of the same prefix. Always-on use raises protection from 93.77% to 99.97% on text and from 97.40% to 99.97% on tool contexts, but also raises ORR to 35.00% and 42.03%. LMA retains 99.97% protection while reducing ORR to 6.50% and 20.84%: improvements of 28.50 and 21.19 points over always-on use. Relative to the original path, protection improves by 6.20 and 2.57 points, with ORR changes of only $+0.67$ and -0.63 points. Selective activation combines prefix-level protection with benign refusal close to the original model.

Table 2: Controlled comparisons under Qwen3Guard (%). The upper group changes activation of the same prefix. The lower group changes training targets with LMA's router and training budget fixed. Text has 3,500 harmful/1,800 benign requests; tool context has 3,690/1,430.

	Text		Tool context	
Condition	Prot.↑	ORR↓	Prot.↑	ORR↓
Original path	93.77	5.83	97.40	21.47
Always-on prefix	99.97	35.00	99.97	42.03
LMA	**99.97**	**6.50**	**99.97**	**20.84**
Label-only targets	98.80	9.78	99.59	24.55
Shuffled policies	98.20	11.50	99.40	27.06

Table 3: Router calibration (%). Each feature set selects its threshold under the same risk-recall constraint. "Active" is the benign activation rate. LMA uses words + domain.

	Text		Tool context	
Features	Recall↑	Active↓	Recall↑	Active↓
Words	99.00	41.13	99.03	27.50
Words + domain	99.00	39.38	99.03	27.50
Words/chars + domain	99.00	26.63	99.03	21.07
Domain only	99.75	97.25	100.00	100.00

The gate activates 12.00% of benign text requests and 19.02% of benign tool-context requests on test, versus 39.38% and 27.50% on calibration. Activation is distinct from refusal: an activated benign request can be answered, and the original path can refuse. The shared-prefix contrast measures the response-level benefit of selective activation at the deployed threshold.

5.2 Calibration Controls Prefix Coverage

The router prioritizes high risk recall, then limits benign activation. Table 3 shows why request text matters at this operating point: domain-only routing activates almost every benign request, while words plus domain activate 39.38% of benign text and 27.50% of benign tool requests on calibration data. The generation experiments use this word-plus-domain router with thresholds fixed before testing.

Word–character features further reduce calibration benign activation to 26.63% and 21.07%. These rows compare request selection, not generated responses. End-to-end protection and refusal are measured separately using the fixed router in Table 2.

5.3 Policy Targets Improve the Selected Path

The lower rows of Table 2 hold the router and training configuration fixed. Compared with label-only targets, structured policy targets improve protection by 1.17 points on text and 0.38 on tool contexts, while reducing ORR by 3.28 and 3.71 points. Shuffling policies worsens both metrics in both settings. Thus, the complete policy-derived targets improve the conditioned path beyond generic action supervision. Together, these controls separate what the prefix learns from where it is applied.

5.4 Comparison with Deployed Alternatives

LMA has the highest protection and lowest ORR among the deployed alternatives in Table 4. Compared with full-policy prompting, ORR is lower by 21.72 points on text and 15.94 on tool contexts, while protection is higher by 0.57 and 1.76 points. Selected policy cards and BM25 retrieval also yield lower protection and higher ORR in these runs.

Table 4: Deployed alternatives on the same test requests, judged by Qwen3Guard (%). Input filter uses Qwen3Guard. Each row represents its complete supervision and control configuration; Table 2 provides the matched component comparisons.

	Text		Tool context	
Method	Prot.↑	ORR↓	Prot.↑	ORR↓
Safety prompt	97.80	14.50	98.89	26.50
Full policy prompt	99.40	28.22	98.21	36.78
Selected policy card	99.20	30.50	98.70	24.83
BM25 policy card	99.29	30.11	98.29	30.49
Input filter	99.71	15.22	99.84	31.47
LoRA	98.51	12.00	99.70	28.04
Prompt tuning	97.49	15.00	98.51	32.03
LMA	**99.97**	**6.50**	**99.97**	**20.84**

Table 5: WildGuard rejudgment (%). Valid harmful/benign counts are 3,500/1,798 and 3,500/1,797 for text original/LMA, and 3,689/1,428 and 3,690/1,429 for tool original/LMA.

Domain	Path	Protection↑	ORR↓
Text	Original	96.57	6.79
	LMA	99.94	7.23
Tool	Original	92.19	30.39
	LMA	99.95	21.06

Input filtering, LoRA, and prompt tuning follow the same ordering. These comparisons assess complete deployed configurations. The shared-prefix control in Table 2 isolates the activation decision within LMA.

5.5 Agreement across Judges and Training Seeds

WildGuard rejudges the same primary responses (Table 5) and agrees with Qwen3Guard on the comparison's direction: LMA improves protection in both settings, has slightly higher text ORR than the original path, and lowers tool-context ORR. Protection remains above 99.9% under either judge.

Three training seeds retain this operating point (Table 6). Qwen3Guard mean protection is 99.96% on text and 99.95% on tool contexts; ORR is $6.54 \pm 0.06\%$ and $21.07 \pm 0.47\%$. WildGuard likewise places mean protection above 99.9%. These are variations across LMA training runs, with each judge assessing the same generated responses.

5.6 Benign Utility and Evaluation Conditions

Table 7 separates benign refusal from instruction-following accuracy. On 1,187 benign over-refusal prompts (Zhang et al., 2025), LMA reduces Qwen3Guard ORR from 43.13% to 18.53%; WildGuard confirms the direction, from 50.97% to 20.05%. On 541 IFEval prompts, strict accuracy changes from 80.41% to 78.37%, a 2.04-point cost. Thus, reduced refusal does not mean unchanged task accuracy.

On source families absent from adaptation, LMA has no harmful compliance among 998 text and 1,050 tool requests. Benign text ORR changes from 26.00% to 18.00% ($n = 100$), but tool-context ORR increases from 8.17% to 28.37% ($n = 208$). The benign-refusal cost therefore depends on the request distribution: the main-test balance does not extend uniformly to unseen sources.

Doubling the generation cap from 64 to 128 tokens preserves Qwen3Guard protection at 99.85% on text and 99.86% on tool contexts. ORR changes from 6.49% to 6.78% and 21.09% to 21.45%, respectively. These subsets contain 661 harmful/339 benign text and 709 harmful/275 benign tool requests.

Table 6: LMA on the main test sets across three training seeds: mean ± sample standard deviation (%).

Domain	Judge	Protection↑	ORR↓
Text	Qwen3Guard	99.96 ± 0.02	6.54 ± 0.06
	WildGuard	99.92 ± 0.03	7.15 ± 0.18
Tool	Qwen3Guard	99.95 ± 0.03	21.07 ± 0.47
	WildGuard	99.94 ± 0.04	21.80 ± 0.77

Table 7: Benign utility (%). The over-refusal test contains 1,187 requests and uses Qwen3Guard; IFEval contains 541 prompts and uses verifiable instruction checks.

	Benign test	IFEval accuracy↑	
Method	ORR↓	Strict	Loose
Base	43.13	80.41	83.55
Safety prompt	60.57	80.41	83.55
Full policy	82.48	72.09	77.08
LMA	18.53	78.37	81.15

5.7 A Separate Test on Another Backbone

On Mistral-Small-3.2-24B-Instruct-2506, we use a separately trained compatible prefix and a text stress set of 192 harmful and 808 benign requests. Under WildGuard, LMA raises protection from 61.98% to 98.96%, while ORR rises from 11.14% to 18.56%. Safety prompting gives 95.31% protection with 46.66% ORR; full-policy prompting gives 100.00% with 56.81% ORR. LMA thus approaches full-policy protection with substantially fewer benign refusals.

This test shows the complete design operating on another backbone and request composition; the matched Qwen experiment isolates selective activation. Their absolute rates refer to different request sets.

5.8 Deployment Cost

LMA reuses a 12 MiB float32 prefix and adds one routing decision before generation. In the text generation-and-postprocessing protocol, 500 requests at batch size 40 take 1.2395 seconds per example for the base endpoint and 1.4217 for LMA: a 14.70% increase. A separate 64-token-cap protocol gives base/LMA times of 5.8017/11.1992 at batch size one ($n = 20$), 3.3297/8.1970 at batch size eight ($n = 160$), and 1.4963/1.7674 at batch size 32 ($n = 640$). These are batch-normalized times, with a separate base reference for each protocol. The compact module's measured overhead is workload-dependent.

Deployment stores the backbone, its materialized prefix, and the calibrated router. Runtime selects a cache configuration and generates once, without retrieving policy text or generating an intermediate policy analysis. This provides a detachable safety condition with fixed-size reusable memory; batching and activation coverage determine its serving cost.

6 Conclusion

LMA separates learning a safety condition from choosing where to apply it. On two frozen Qwen backbones, selective activation retains the same prefix's observed 99.97% harmful-request protection while reducing benign over-refusal by 28.50 and 21.19 percentage points. Structured policy targets further improve the conditioned path. The central result is that activation scope is a useful control in safety adaptation: on the main test distributions, a calibrated gate retains the prefix's protection while keeping benign refusal close to the original model.

AI USE STATEMENT

We used generative AI tools to assist with language editing and grammar checking, and to check LaTeX typesetting. The authors take responsibility for the final text, research claims, and associated artifacts.

ETHICS STATEMENT

This work studies harmful-request benchmarks to improve safety control. Such evaluation inputs and outputs require appropriate access controls and respect for their source licenses. The reported results concern automatically judged responses under fixed request sets. Application-level tool authorization, sandboxing, and monitoring remain necessary when a generator is connected to executable tools.

REPRODUCIBILITY STATEMENT

Section 3 specifies the prefix parameterization, training objective, routing rule, and complete inference procedure. Section 4 reports data partitions, training and generation settings, and metric definitions. The controlled comparisons in Section 5 distinguish changes to activation from changes to supervision. Appendix A provides the deterministic policy-selection rule and inventory.

REFERENCES

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, et al. AgentHarm: A benchmark for measuring harmfulness of LLM agents, 2024.

Huzaifa Arif, Keerthiram Murugesan, Ching-Yun Ko, Pin-Yu Chen, Payel Das, and Alex Gittens. Patching LLM like software: A lightweight method for improving safety policy in large language models, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, et al. Constitutional AI: Harmlessness from AI feedback, 2022.

Somnath Banerjee, Sayan Layek, Soham Tripathy, Shanu Kumar, Animesh Mukherjee, and Rima Hazra. SafeInfer: Context adaptive decoding time safety alignment for large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27188–27196, 2025. doi: 10.1609/aaai.v39i26.34927.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, et al. JailbreakBench: An open robustness benchmark for jailbreaking large language models, 2024.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations, 2023.

Kundan Krishna, Joseph Y. Cheng, Charles Maalouf, and Leon A. Gatys. Disentangled safety adapters enable efficient guardrails and flexible inference-time alignment, 2025.

Seanie Lee, Dong Bok Lee, Dominik Wagner, Minki Kang, Haebin Seong, Tobias Bocklet, Juho Lee, and Sung Ju Hwang. SafeRoute: Adaptive model selection for efficient and accurate safety guardrails in large language models. In *Findings of the Association for Computational Linguistics:*

ACL 2025, pp. 2053–2069. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.105.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. SALAD-Bench: A hierarchical and comprehensive safety benchmark for large language models, 2024.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4694–4702. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.311.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-short.8.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal, 2024.

NVIDIA. Nemotron-RL agentic indirect prompt injection v1. `https://huggingface.co/datasets/nvidia/Nemotron-RL-Agentic-Indirect-Prompt-Injection-v1`, 2026. Hugging Face dataset.

Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo Guardrails: A toolkit for controllable and safe LLM applications with programmable rails, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=GEcwtMk1uA`.

Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016. doi: 10.1109/TIT.2016.2603151.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, et al. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: Evaluating safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 896–911. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-eacl.61.

Chang Wu, Junfeng Fang, Houcheng Jiang, Kai Tang, Pengyu Cheng, Xiaoxi Jiang, Guanjun Jiang, and Xiang Wang. PolicyAlign: Direct policy-based safety alignment for large language models, 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 (Volume 1: Long Papers)*, pp. 5587–5605. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.303.

Zhiyu Xue, Zimo Qi, Guangliang Liu, Bocheng Chen, and Ramtin Pedarsani. Deactivating refusal triggers: Understanding and mitigating overrefusal in safety alignment. In *Proceedings of TrustNLP 2026*, pp. 402–412. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.trustnlp-main.26.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report, 2025.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.624.

Haonan Zhang, Dongxia Wang, Yi Liu, Kexin Chen, and Wenhai Wang. LLM-VA: Resolving the jailbreak-overrefusal trade-off via vector alignment, 2026.

Zhehao Zhang, Weijie Xu, Fanyou Wu, and Chandan K. Reddy. FalseReject: A resource for improving contextual safety and mitigating over-refusals in LLMs via structured reasoning, 2025.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, et al. Qwen3Guard technical report, 2025.

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.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers, 2024.

A Policy Selection Details

This appendix records the policy inventory, deterministic target-selection rule, and a distributional bound supporting the main-text analysis.

A.1 Policy Inventory

The 28 entries in Table 8 group their fields into identity (identifier, domain, title), policy (principles, conditions, prohibited and permitted actions), response (obligations, exceptions, expected action, response policy), retrieval (keywords and scenario features), and provenance metadata. The fields `legal_source` and `jurisdiction` are project-defined provenance labels. Target records retain selected policy identifiers. The policy JSONL's recorded SHA-256 identity is the concatenation of these two lines:

```
81a8abc6d2598eadc86536022c90581c
1602fede81a98c171b3744245926bf92
```

Table 8: Policy inventory. Identifiers omit their shared `CONST-` prefix.

ID	Policy role
CY1	Unauthorized access and credential misuse
CY2	Malware, exploits, and destructive payloads
CY3	Defensive security assistance
PR1	Personal data protection
PR2	Data security and confidentiality
FR1	Fraud, phishing, and social engineering
FR2	Financial abuse and evasion
AG1	Explicit authorization for tool use
AG2	Resistance to indirect prompt injection
AG3	Human oversight in high-impact decisions
VI1	Violence and weapon-related harm
SH1	Self-harm and suicide safety
DR1	Illicit drug production and misuse
BIO1	Biological and chemical harm
HH1	Hate, harassment, and abuse
SE1	Minors, coercion, and sexual safety
MED1	Medical safety and professional boundaries
LEG1	Legal-information boundaries
FIN1	Financial-advice boundaries
INF1	Misinformation, deception, and impersonation
CIV1	Civic and public-process integrity
IP1	Copyright and proprietary content
CH1	Child safety
BOUND1	Clarification under ambiguity
JB1	Resistance to policy evasion
UTIL1	Helpfulness and restraint of over-refusal
TRAN1	Uncertainty and non-deceptive communication
COMP1	Legal compliance and auditability

A.2 Relevance Scoring and Action-Compatible Selection

Let tok lowercase a string and extract the set of terms matching `[a-z0-9_]+`. For adaptation record a, the query concatenates prompt, domain, risk label, expected action, and harm category. Searchable policy text concatenates identifier, domain, title, policy, boundary and response fields, keywords, and scenario features. Missing fields are empty strings or lists. For policy r, define

$$\omega(a, r) = \frac{|\,\mathrm{tok}(a) \cap \mathrm{tok}(r)|}{\max(1, |\,\mathrm{tok}(a)|)}. \tag{10}$$

If k_{kw} policy-keyword substrings occur in the prompt, its score is

$$s(a,r) = \omega(a,r) + 1.5I_{\mathrm{dom}} + 0.7I_{\mathrm{harmdom}} + \min(1.5, 0.25k_{\mathrm{kw}}) + 1.2I_{\mathrm{benign,UTIL}} + I_{\mathrm{ambig,BOUND/AG}} + 1.5I_{\mathrm{jailbreak,JB}} + 0.8I_{\mathrm{agent}}. \tag{11}$$

Here I_{dom} indicates an exact domain match or containment of the adaptation domain in the policy domain; I_{harmdom} indicates that the policy domain appears in the harm-category text. The remaining indicators match benign records to UTIL1, ambiguous records to BOUND1 or AG1, jailbreak-tagged records to JB1, and agent-modality records to agent-domain policies. These labels are used offline.

Keep positive-score candidates; if none remain, insert BOUND1 and UTIL1 with score one. Rank by descending score, breaking ties by descending identifier, and keep at most five. For a benign or borderline record with fewer than five candidates, append UTIL1 with score 0.75 if absent. Use an explicit expected action when available; otherwise map harmful records to refusal, ambiguous or borderline records to clarification, and the rest to compliance. Select action-compatible policies first, then fill from the remaining candidates up to two. Construct the response target using the selected entries and action, preserving an available reference answer for benign compliance. Save the request, target, selected identifiers, scores, source coordinates, and hashes.

For the shuffled control, take the first available value among `prompt_sha256`, `example_id`, and prompt text. The sum of its encoded bytes determines a cyclic offset into the policy list. Prefer out-of-domain policies and retain the original policy count. Label-only supervision supplies generic action responses. These controls retain the examples, encoder size, training schedule, router, and decoding settings; their target content and lengths follow the respective construction rules.

B Activation-Weighted Response Change

Use a common stopping convention and generation cap. For request u, let p_0^u and p_1^u denote the original and prefix-conditioned response distributions. The gate is fixed before generation. For a benign request distribution D_b, let $\epsilon_b = \mathbb{E}_{D_b} G$. Any statistic $f(u,y) \in [0,1]$ satisfies, for $\epsilon_b > 0$,

$$|\mathbb{E}_{D_b,p_G} f - \mathbb{E}_{D_b,p_0} f| \le \epsilon_b \mathbb{E}_{D_b|G=1}\, \mathrm{TV}(p_1^u, p_0^u). \tag{12}$$

Proof. For each request, Equation 1 gives $\mathbb{E}_{p_G} f - \mathbb{E}_{p_0} f = G(\mathbb{E}_{p_1} f - \mathbb{E}_{p_0} f)$. Average over requests and apply the triangle inequality. A difference of expectations of a $[0,1]$-valued statistic is bounded by total variation. Conditioning on $G = 1$ yields Equation 12. If $\epsilon_b = 0$, the expectations are equal.

For natural logarithms and finite conditional KL divergence, Pinsker's inequality (Sason & Verdú, 2016) and Jensen's inequality give

$$|\mathbb{E}_{D_b,p_G} f - \mathbb{E}_{D_b,p_0} f| \le \epsilon_b \min\left\{1, \sqrt{\tfrac{1}{2}\mathbb{E}_{D_b|G=1} D_{\mathrm{KL}}(p_0^u \| p_1^u)}\right\}. \tag{13}$$

This distributional bound complements the finite-set counting argument in Section 3. The latter directly covers the reported greedy evaluation when bypassed responses and judgments are unchanged.