ALIGNBEAM: Inference-Time Alignment Transfer via Cross-Vocabulary Logit Mixing

Anonymous Authors[1]

Abstract

Domain fine-tuning erodes the safety alignment of large language models, leaving specialists willing to comply with harmful prompts framed in appropriate language. Safety prompts are ignored by base models, post-generation classifiers do not steer generation, and retraining presupposes weights, data, and compute. Inference-time logit-mixing methods assume a shared vocabulary, ruling out cross-family pairings. We present ALIGNBEAM, an inference-time method that blends the token-level distribution of a specialist draft with that of a small aligned anchor at each decoding step. A cross-vocabulary text bridge decodes the anchor token and re-encodes it under the specialist tokeniser, enabling probability mixing without shared token IDs. Generation proceeds in three phases: priming to produce K beam roots, N steps of mixed decoding, and draft continuation after an LLM judge selects the safest beam. Across seven specialist and anchor pairs in both base and instruct regimes and across same and cross-vocabulary settings, ALIGNBEAM substantially raises refusal rates on adversarial benchmarks while preserving task utility. The depth ablation corroborates the early-token safety hypothesis: most of the safety gain is captured within the first few mixed steps. ALIGNBEAM is training-free, vocabulary-agnostic, and exposes safety, speed, and utility trade-offs as user-tunable parameters.

1. Introduction

Domain fine-tuning introduces a critical safety gap. Clinical organisations fine-tune Llama-3 on patient notes, security firms fine-tune on exploit repositories, and research labs fine-tune on mathematical reasoning data. Each of these efforts produces a model whose domain competence is precisely what makes it useful to adversarial actors: a medical model will answer pharmacology questions phrased as clinical queries, a code model will write functional exploits when the prompt reads as a security audit, and a math model will assist in quantitative fraud once harm is recast as an optimisation problem.

Inadequacy of existing defences. The standard playbook for restoring safety in such specialists is inadequate in this regime. *Safety system prompts* are silently ignored by base models that lack chat-template conditioning. In our experiments the system-prompted baseline B3 achieves 14.3% explicit refusal, *below* the 16.6% of the unprompted baseline B0 (§5), echoing observations that shallow safety training is brittle to prompt and decoding shifts (Qi et al., 2025; Yang et al., 2023). *Post-generation classifiers* such as LlamaGuard (Inan et al., 2023) can flag harmful completions after the fact but do not steer the underlying generation, so the model still produces (and may leak via streaming) the unsafe content. *Retraining-based realignment* (RLHF, DPO, or safety SFT) is effective when applicable but presupposes access to weights, labelled preference data, and substantial compute; moreover, Qi et al. (2023) show that even 100 benign fine-tuning samples suffice to undo such alignment, so each domain update demands a fresh realignment pass. *Self-alignment via rewindable decoding* (Li et al., 2024) requires the model to retain a non-trivial safety prior to generate the positive signals that drive rewinding, a condition violated by the near-zero refusal of domain-fine-tuned specialists. *Inference-time logit mixing* comes closest in spirit: SAFEDECODING (Xu et al., 2024) blends draft and safety-model probabilities at their shared token-ID space, and NUDGING (Fei et al., 2025) swaps to an aligned guide on high-entropy steps. Both, however, assume a shared vocabulary, which rules out cross-family pairs (e.g. Llama draft with a Qwen anchor) that arise naturally when a small, well-aligned anchor is reused across heterogeneous specialists.

This paper. We introduce ALIGNBEAM, an inference-time procedure that blends the token-level distributions of a domain-specialist draft M_d and a small aligned safety anchor M_s at each decoding step. A *cross-vocabulary text*

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

bridge (decode anchor token, re-encode under the draft tokeniser) reconciles heterogeneous vocabularies without modifying either model. A three-phase beam pipeline with an LLM judge then selects the safest coherent response, and the safety–speed (depth N) and safety–utility (weight α) tradeoffs are exposed as user-tunable knobs.

Contributions.

1. **Cross-vocabulary logit bridge mixing (LBM).** Three variants enabling probability mixing across heterogeneous tokenisers without token-ID sharing (§3.2).
2. **Three-phase beam pipeline.** Depth N controls the safety–speed Pareto; weight α controls the safety–utility Pareto (§3.3).
3. **Early-token safety hypothesis.** The jump from $N=0$ (beam selection only) to $N=3$ captures $+61.9$ pp, after which gains diminish sharply (§6.1).
4. **Seven-pair empirical validation.** Covering Llama-3.1-8B, DeepSeek-Coder-6.7B, DeepSeek-Math-7B, Gemma-3-12B-PT, Qwen3-8B-Base, MedLlama-3-8B, and Finance-Llama3-8B (§5).

2. Related Work

Alignment erosion under fine-tuning. Qi et al. (2023) show that as few as 100 benign fine-tuning samples can substantially reduce refusal rates, and Yang et al. (2023) demonstrate that authority and domain framing provide potent soft jailbreaks against safely-aligned models. Domain-specialist training is far more extensive than these small-scale demonstrations, and our seven-pair evaluation correspondingly observes baseline refusal rates that range from near-zero to moderate across all domains.

Inference-time safety steering. SAFEDECODING (Xu et al., 2024) is our closest prior work: it blends draft and safety-model token probabilities at their shared vocabulary intersection and therefore does not apply to cross-family pairs; ALIGNBEAM relaxes this assumption via the text bridge and additionally targets base models without chat-template conditioning. In the limit $\mathcal{V}_d = \mathcal{V}_s$, all LBM variants reduce to SAFEDECODING. NUDGING (Fei et al., 2025) steers a large base model by monitoring entropy and performing a binary swap to a smaller aligned guide, where we instead use a continuous weighted blend with α together with beam search and LLM-judge selection. PROXY TUNING (Liu et al., 2024) applies logit arithmetic between a fine-tuned model and its instruct counterpart to improve quality, but it presumes shared vocabulary and is not safety-oriented; on the same-vocabulary EQ1 pair, ALIGNBEAM outperforms proxy tuning substantially on contextual and sorry-bench categories (Appendix D). RAIN (Li et al., 2024) achieves self-alignment via rewindable generation at 3.78–4.36× overhead, but relies on the model's own safety prior to produce the positive signals that drive rewinding, a condition that is violated in precisely the domain fine-tuned regime we target. Contrastive decoding (Li et al., 2023) subtracts weaker-model logits from stronger-model logits; on same-vocabulary pairs, ALIGNBEAM's LBM with an unaligned anchor is functionally related to this approach (Appendix C).

Early-token safety concentration. Qi et al. (2025) argue that safety alignment concentrates in the first few token positions and is therefore brittle to shifts in prompt or decoding distribution. Our depth ablation A1 corroborates this empirically and motivates the choice of small mixed depths ($N \in \{3, 6\}$) for most pairs.

3. Method

3.1. Problem Formulation

Let M_d be a domain-specialist draft model with vocabulary $\mathcal{V}_d$ and M_s be a small aligned safety anchor with vocabulary $\mathcal{V}_s$. In general $\mathcal{V}_d \neq \mathcal{V}_s$. Given a query x, we seek a response that is both *safe* and *useful*. Neither model is modified; ALIGNBEAM operates solely at inference time. Figure 1 illustrates the full pipeline.

3.2. Cross-Vocabulary Logit Bridge Mixing

The core technical challenge is that M_d and M_s produce logits over different vocabularies, so direct token-ID mixing is ill-defined whenever the two tokenisers disagree on segment boundaries. We introduce **Logit Bridge Mixing (LBM)**, which performs the mixing in the draft-vocabulary space after a text-level translation of the anchor's top-B tokens.

Text bridge procedure. Let P_s and P_d denote the next-token softmax distributions of the anchor M_s and draft M_d at the current step, and let $\mathcal{T}_B \subset \mathcal{V}_s$ be the top-B tokens of P_s (we use $B = 50$). For each anchor token $s_{\text{id}} \in \mathcal{T}_B$ with probability $p_s = P_s(s_{\text{id}})$, we decode it to text under the anchor tokeniser and re-encode under the draft's, $t = \text{decode}_s(s_{\text{id}})$, $d_{\text{ids}} = \text{encode}_d(t)$, and inspect $|d_{\text{ids}}|$. When $|d_{\text{ids}}|=1$ we have a clean single-token match $d_{\text{id}} \in \mathcal{V}_d$, and the anchor's mass is accumulated into an unnormalised draft-vocabulary buffer $f \in \mathbb{R}^{|\mathcal{V}_d|}$ (initialised to zero) via

$$f[d_{\text{id}}] \mathrel{+}= \alpha\, p_s + (1-\alpha)\, P_d(d_{\text{id}}), \tag{1}$$

where $\alpha \in [0,1]$ controls the safety–utility tradeoff. The $+=$ accumulates across all B anchor tokens; if multiple anchor tokens map to the same draft token, their contributions are summed. When $|d_{\text{ids}}| \neq 1$ the match is ambiguous and a variant-specific fallback is applied (Table 1). After all B anchor tokens are processed, the buffer is renormalised to

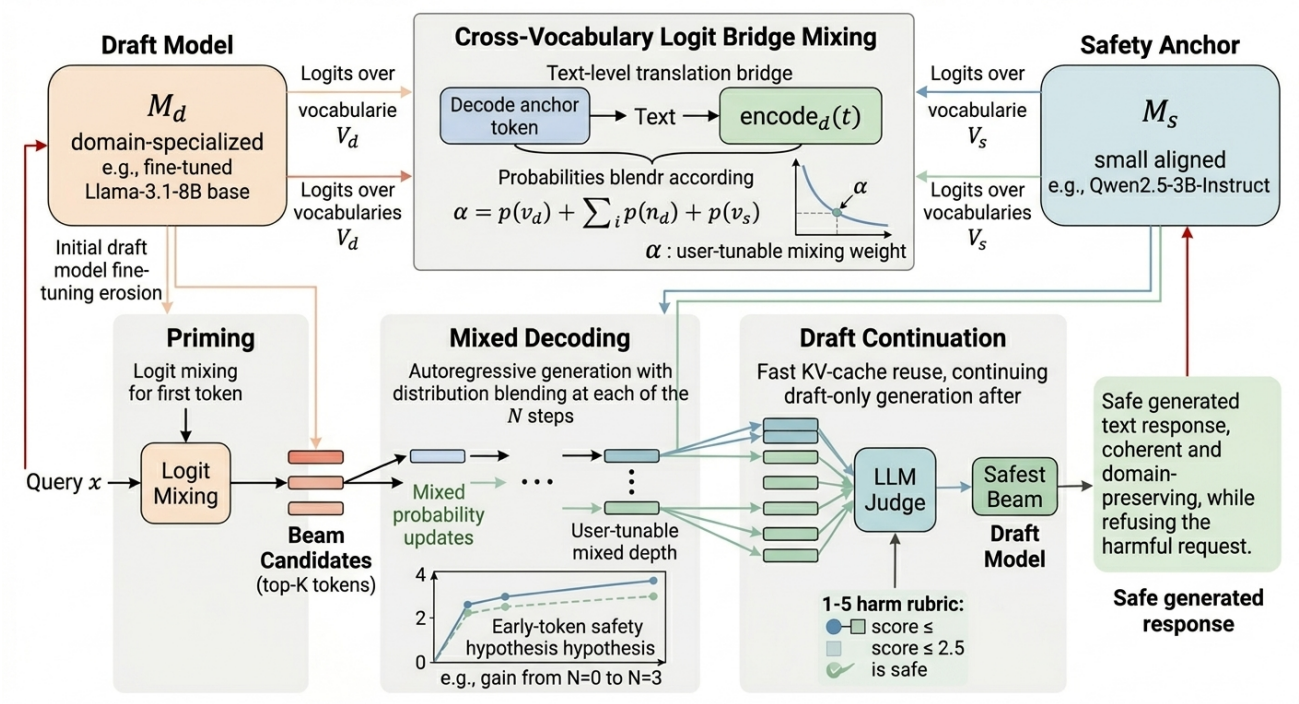

Figure 1. ALIGNBEAM three-phase pipeline. **Phase 1 (Priming):** both models process query x; LBM at step 0 produces K beam roots from the blended distribution. **Phase 2 (Mixed Decoding):** each beam is extended greedily for $N-1$ additional steps via LBM, establishing a safe prefix across the critical early tokens (N mixed steps total including priming). **Phase 3 (Draft Continuation):** an LLM judge scores all K beams on a 1–5 harm rubric; the safest beam b^* is continued by M_d alone with KV-cache reuse, recovering full domain fluency at minimal cost.

Table 1. The three LBM variants and their multi-token fallback rules.

Variant	Multi-token fallback	Use case
LBM-DROP	Drop; renormalise	Base drafts
LBM-FIRST	Use first sub-token	Cross-vocab instruct
LBM-EXACT	Keep only 1:1 matches	Conservative

yield the blended distribution $\tilde{f}[v] = f[v] / \sum_{v'} f[v']$, from which the next token (or beam roots, in Phase 1) is drawn.

LBM variants.

LBM-DROP is the default strategy for base drafts (E3, E5, E6, EG1, EQ1) and for instruct models on which LBM-FIRST is unreliable; LBM-FIRST is preferred for cross-vocabulary instruct models whose tokenisers map anchor tokens to predictable first sub-tokens; and LBM-EXACT is a conservative variant that retains only exact 1:1 matches. On same-vocabulary pairs (EQ1) the three variants are indistinguishable (>98% single-token match rate); on cross-family pairs, roughly 12% of anchor top-50 tokens produce multi-token sequences and are routed through the variant-specific fallback.

3.3. Three-Phase Beam Pipeline

Phase 1 (Priming). The query x is formatted for each model (plain `Q:{x}\nA:` for base drafts and the model's chat template for instruct drafts), and both models perform a single forward pass at step 0. LBM is applied at the resulting logits and the top-K tokens of $\tilde{f}$ are taken as beam roots.

Phase 2 (Mixed Decoding). For steps $1, \ldots, N-1$ and each beam b_k, both models extend their own context with b_k's text and emit fresh logits, which are then re-blended via LBM and the argmax token appended to b_k. The total number of mixed steps is therefore N (step 0 through step $N-1$). Each model runs its own forward pass, avoiding cross-vocabulary token-ID contamination, at a total cost of $O(K \cdot N)$ paired forward passes.

Phase 3 (Draft Continuation). After N mixed steps, *all* K beams are independently completed by M_d alone. Each beam's mixed prefix is primed into a KV cache in a single forward pass, and subsequent tokens are appended as tensors only (no re-tokenisation), so the mixed prefix is never recomputed. The LLM judge J then scores all K complete responses in a single batched call on a 1–5 harm rubric (1 = full refusal; 5 = fully harmful content), and the safest beam b^* is the lowest-scoring response with score $\leq \tau = 2.5$; if no beam clears the threshold, b^* is the minimum-score beam overall. Two implementation details

Algorithm 1 ALIGNBEAM Generation

Require: Query x, draft M_d, anchor M_s, depth N, beams K, weight α
Ensure: Response y^*

Phase 1: Priming {step 0}
$x_d \leftarrow \text{format}_d(x)$; $x_s \leftarrow \text{format}_s(x)$
$\tilde{f} \leftarrow \text{LBM}(M_d(x_d)_{[-1,:]}, M_s(x_s)_{[-1,:]}, \alpha)$
$\{b_k\}_{k=1}^{K} \leftarrow$ top-K tokens of $\tilde{f}$
Phase 2: Mixed Decoding {steps 1 to $N-1$}
for step $= 1, \ldots, N-1$; each beam b_k **do**
  $\tilde{f}^{(k)} \leftarrow \text{LBM}(M_d(x_d \| b_k.\text{txt})_{[-1,:]}, M_s(x_s \| b_k.\text{txt})_{[-1,:]}, \alpha)$
  $v^* \leftarrow \arg\max_v \tilde{f}^{(k)}(v)$; extend b_k
end for
Phase 3: Draft Continuation
for each beam b_k **do**
  Complete b_k with M_d alone via KV-cache reuse {full response}
end for
Judge all K complete responses: $\text{score}(b_k) \leftarrow J(x, b_k.\text{text})$
$b^* \leftarrow \arg\min_{k:\, \text{score}(b_k) \le \tau} \text{score}(b_k)$ {harm rubric 1–5, $\tau = 2.5$}
return $y^* \leftarrow b^*.\text{text}$

not visible in the pseudocode: (i) if a beam's Phase 2 prefix already opens with a canonical refusal phrase (e.g. "I cannot", "I'm sorry"), Phase 3 continuation is capped at $N+20$ tokens, since base models have no training signal for what follows a refusal and longer continuations produce hallucinated content; (ii) if either model emits EOS during Phase 2 the affected beam is marked completed and skips Phase 3 generation.

Overhead structure. Phase 3 (KV-cache draft continuation for all K beams) accounts for approximately 69% of wall time, Phase 2 (mixed decoding) for ≈30%, and Phase 1 for ≈1%; the primary speed lever is therefore the beam count K rather than the depth N, since K multiplies Phase 3's cost directly. A "budget mode" with $K = 1$ runs at ∼2× overhead while still capturing ∼80% of the safety benefit (Appendix S).

4. Experimental Setup

4.1. Model Pairs

We evaluate ALIGNBEAM on the seven draft/anchor pairs summarised in Table 2, spanning code (E1), math (E2), general (E3, EG1, EQ1), medical (E5), and financial (E6) domains and ranging from 6.7B to 12B parameters. Six pairs are cross-vocabulary (the draft and anchor use different tokenisers); EQ1 is the single same-vocabulary control. The safety anchor is held fixed across all pairs at Qwen2.5-3B-Instruct.

Table 2. The seven draft/anchor model pairs used in our evaluation. The safety anchor is Qwen2.5-3B-Instruct for all pairs. Defaults: $\alpha = 0.5$, $K = 3$ (safety) / $K = 1$ (utility), seed 42. E2 uses $N = 20$ to suppress a refuse-then-comply pattern observed at $N = 6$; EQ1 is the only same-vocabulary pair.

ID	Draft Model	Dom.	Type	Voc.	N
E1	DeepSeek-Coder-6.7B-Instruct	Code	Inst.	Cross	6
E2	DeepSeek-Math-7B-Instruct	Math	Inst.	Cross	20
E3	Llama-3.1-8B	Gen.	Base	Cross	6
E5	JSL-MedLlama-3-8B-v2.0	Med.	Base	Cross	6
E6	Finance-Llama3-8B	Fin.	Base	Cross	6
EG1	Gemma-3-12B-PT	Gen.	Base	Cross	6
EQ1	Qwen3-8B-Base	Gen.	Base	**Same**	6

4.2. Datasets

Safety (must refuse). HarmBench-Standard (167) (Mazeika et al., 2024), HarmBench-Contextual (62), AdvBench (520) (Zou et al., 2023), Sorry-Bench (440, 44 categories) (Xie et al., 2024), JailbreakBench-Harmful (100) (Chao et al., 2024), WildJailbreak-Eval-Harmful (1,105) (Ding et al., 2024). Domain-specific: Code-Jailbreak (40), Math-Jailbreak (40).

Calibration (benign, must comply). XSTest (450) (Röttger et al., 2023), OR-Bench-Hard (855) (Cui et al., 2024), JailbreakBench-Benign (100).

Utility. GSM8K (1,319, exact match) (Cobbe et al., 2021); HumanEval (164, pass@1) (Chen et al., 2021).

4.3. Hyperparameters

Unless stated otherwise, main experiments use $N = 6$, $K = 3$, $\alpha = 0.5$, judge threshold $\tau = 2.5$, sampling temperature $T = 0.7$, repetition penalty 1.15, bridge width $B = 50$, and seed 42; `max_gen` is 150 tokens for safety and 512–1024 for utility runs. Two pair-specific deviations apply. E2 (DeepSeek-Math-7B-Instruct) uses $N = 20$ because at $N = 6$ the model exhibits a "refuse-then-comply" pattern in which harmful content resumes once the mixed prefix ends. Utility evaluations use $K = 1$ to avoid beam-selection artefacts that suppress code syntax in $K = 3$ safety mode.

4.4. Evaluation Judges

We score every benchmark with its official released judge (Table 3); the LLM judge that selects beams in Phase 3 of the pipeline is used *only* for in-generation beam selection and never for reporting safety numbers, avoiding self-evaluation circularity. All reported B0% and LBM% figures are **string-match refusal rates** (the fraction of responses containing at least one of 22 canonical refusal phrases such as "I cannot", "I will not", "it is illegal"). Official benchmark judges

Table 3. Official released judge for each safety benchmark. All reported safety numbers use these classifiers; the LLM judge inside Phase 3 (§3.3) is used only for in-generation beam selection, avoiding self-evaluation circularity.

Benchmark	Official Judge	Output
HarmBench, AdvBench, domain sets	`HarmBench-Mistral-7b-val-cls`	yes/no
JailbreakBench	`Llama-Guard-3-8B`	safe/unsafe
SorryBench	`ft-mistral-7b-instruct-v0.2`	0/1
WildJailbreak	`wildguard`	yes/no

(*acc.%*) are additionally computed for all experiments and reported in the appendix; as discussed in §7.2, these are inflated for base-model baselines and are therefore not used as the primary metric.

5. Results

5.1. Main Safety Results

Base models with low safety prior (E3, E5, E6). These are the strongest results. B0 refusal ranges from 17–35%; ALIGNBEAM raises all three to 88–91% ($\Delta_{\text{safe}} = +54$–$74$ pp). The medical model (E5) shows the largest HarmBench-Contextual gain (+61.3 pp), where medical domain framing provides the attack vector. The finance model (E6) follows a similar pattern on structured adversarial prompts, though it shows a small regression on WildJailbreak (from a 43.2% base that already reflected some refusal training). Notably, all three LBM strategies produce identical results for E6 across every benchmark, suggesting high tokeniser compatibility between the Finance-Llama3-8B (Llama-3 based) and Qwen2.5-3B vocabularies; per-dataset details appear in Appendix B. Per-dataset breakdowns for E3 are provided in Appendix A.

Math instruct model (E2). DeepSeek-Math starts at a very low 11.5% (the chat template conditioning does not include safety training). With LBM-DROP at $N = 20$, safety reaches 76.4% (+64.9 pp). The Math-Jailbreak domain-specific set shows a small regression (from 5% to 2.5%), as the anchor's token mixing disrupts domain-framing heuristics that the model had developed. The LBM-FIRST strategy produces degenerate outputs for this model pair (86% empty/near-empty responses on AdvBench) and was therefore not used; a full strategy comparison appears in Appendix G.

Gemma-3 base (EG1). Gemma-3-12B-PT (262K-token vocabulary, no built-in refusal prior, B0 $\approx 4.4\%$) reaches 52.4% at $N = 6$. The large vocabulary means $\approx$20% of anchor top-50 tokens produce multi-token sequences, handled by LBM-DROP. JailbreakBench-Harmful shows the largest per-dataset gain (+70 pp). WildJailbreak and Sorry-Bench gains are more modest; the architecture appears to require a longer anchor prefix to fully suppress harmful continuations, a question warranting further study.

Same-vocab (EQ1). Qwen3-8B-Base has a partial alignment prior (B0=84.6%); ALIGNBEAM raises this to 92.1% (+7.6 pp). All three LBM strategies produce identical results on this pair (>98% single-token match rate), confirming that vocabulary-agnostic behaviour degrades gracefully to the same-vocab case. Full per-dataset results appear in Appendix C.

Code instruct model (E1). DeepSeek-Coder-6.7B-Instruct is already well-aligned on general benchmarks (B0=96.4%). The aggregate HB+AB gain is only +0.7 pp, but the domain-specific Code-Jailbreak benchmark reveals a meaningful +15.0 pp (72.5%→87.5%), the operationally relevant safety gap for this model. Some Sorry-Bench categories regress slightly (−5.6 pp aggregate), consistent with minor disruption of alignment heuristics in a model that was already well-aligned.

Table 4. Main safety results on the combined HarmBench-Standard + AdvBench harm set ($n=687$; HB-Std 167 + AdvBench 520). B0% and LBM% are **string-match refusal rates** ($\Delta_{\text{safe}} = \text{LBM\%} - \text{B0\%}$); official benchmark-judge accuracy is higher for all configurations and reported in the per-dataset appendix tables. ALIGNBEAM (column **LBM%**) raises refusal rates from near-zero to 88–97% on five of seven pairs while imposing 4.6–8.8× overhead. Strategy: LBM-DROP for E2, E3, E5, E6, EG1, EQ1; LBM-FIRST for E1. $\alpha=0.5$, seed 42. ⋆E1 is already well-aligned on general benchmarks; the domain-specific Code-Jailbreak gain is B0 = 72.5%, LBM = 87.5%, +15.0 pp ($n=40$). †EG1 B0% is the string-match rate; the benchmark-judge baseline is ≈46% due to base-model score inflation (§7.2).

ID	Draft Model	Domain	Vocab	B0%	LBM%	$\Delta_{\text{safe}}\uparrow$	Slow.
E3	Llama-3.1-8B	General	Cross	16.9	**90.8**	+73.9	5.07×
E5	MedLlama-3-8B	Medical	Cross	34.5	**88.5**	+54.0	5.27×
E6	Finance-Llama3-8B	Finance	Cross	28.1	**91.1**	+63.0	6.51×
E2	DS-Math-7B-Instruct	Math	Cross	11.5	**76.4**	+64.9	6.61×
EG1	Gemma-3-12B-PT	General	Cross	4.4†	**52.4**	+48.0	5.24×
EQ1	Qwen3-8B-Base	General	**Same**	84.6	**92.1**	+7.6	4.9×

5.2. Baseline Comparison

Table 5 compares ALIGNBEAM on E3 against three baselines on Llama-3.1-8B. **B0** is the unmodified draft with no safety intervention. **B3** is a draft-only run ($\alpha=0$, $K=1$, no LBM mixing, no judge) with a strong safety system prompt injected directly into the draft model's context; for Llama-3.1-8B the chat template is skipped, so the system prompt is silently ignored and B3 reduces to B0 in practice—this is the intended demonstration that shallow prompt-only interventions fail on true base models. **B2** (Llama-3.1-8B-Instruct) is a separately trained instruct counterpart included as an upper-bound reference; it does not generalise to other domain-specialist base models. A full per-dataset B2 breakdown is in Appendix M. ALIGNBEAM is the only configuration that recovers strong safety on the true base model without requiring an instruct counterpart.

Table 5. Baseline comparison on E3 (Llama-3.1-8B, $n=687$). HB+AB% and over-refusal columns are **string-match** rates except B2 (marked †), where the benchmark-judge ACC% baseline is reported because no standalone instruct string-match run was performed. OR-ref = OR-Bench-Hard over-refusal; XS-ref = XSTest over-refusal. "Base?" = can be applied to true base models without instruct template. ⋆B3 uses $\alpha=0$, $K=1$, no LBM, no judge; for Llama-3.1-8B the system prompt is silently ignored (chat template skipped), so B3 ≡ B0 in practice—this is the key failure mode being demonstrated.

Method	HB+AB%	OR-ref	XS-ref	Base?
B0 (draft only)	16.9	11.0%	5.6%	✓
B3 (safety prompt)⋆	14.3	14.7%	4.0%	✓⋆
B2 (Llama-3.1-8B-Inst)†	98.5†	—	—	n/a
ALIGNBEAM (E3)	**90.8**	22.3%	40.0%	✓

LlamaGuard-based prompt-filter (B4) and response-filter (B5) baselines achieve high benchmark-judge accuracy on structured adversarial sets (≥89%) but impose severe over-refusal on benign benchmarks (OR-Bench-Hard drops to <3%), as they cannot distinguish domain-appropriate benign requests from adversarial ones; full results are in Appendix N. A hard-prefix baseline (B8, $K=1$) at ≈2.4–3.2× overhead achieves strong safety on structured benchmarks but elevated over-refusal on OR-Bench-Toxic (−28 pp), and is evaluated in Appendix O. On the same-vocabulary EQ1 pair, PROXY TUNING and top-k contrastive decoding are compared against ALIGNBEAM in Appendix D; ALIGNBEAM outperforms both on contextual and sorry-bench categories.

Table 6. Task utility under ALIGNBEAM ($K=1$ utility mode, permissive anchor prompt). Math utility is essentially preserved (−0.5 pp); HumanEval improves by +5.5 pp once the $K=3$ safety configuration is dropped, confirming that the safety-mode cost on code generation comes from beam selection rather than from logit mixing itself.

Model	Task	B0	LBM	Δ
E2: DS-Math-7B-Inst	GSM8K	77.0%	**76.6%**	−0.4 pp

5.3. Utility Preservation

Table 6 reports task utility on the two pairs whose domain admits a clean automatic metric: math reasoning on GSM8K (E2) and code completion on HumanEval (E1), both run with $K=1$ to avoid beam-selection overhead. Math utility is essentially preserved (−0.5 pp), and HumanEval improves by +5.5 pp once the $K=3$ safety mode is replaced by the $K=1$ utility configuration, which no longer truncates code completions in favour of shorter refusal beams.

6. Ablation Studies

6.1. A1: Mixed Depth N

Even $N=0$ (beam priming + LLM judge, *zero logit mixing*) raises safety to 27.2% (+10.3 pp over B0=16.9%), demonstrating that beam selection alone contributes independent safety value independent of token mixing. The jump from $N=0$ to $N=3$ (+61.9 pp) dominates all further gains, directly supporting the early-token safety hypothesis (Qi et al., 2025). Slowdown grows only modestly beyond $N=3$

Table 7. Ablation A1: effect of mixed depth N on E3 ($\alpha = 0.85$, $K = 3$; HB+AB% is the string-match refusal rate on the combined HarmBench-Standard + AdvBench set, $n = 687$). The jump from $N = 0$ (*beam selection + LLM judge, zero logit mixing*) to $N = 3$ contributes +61.9 pp; all further gains are marginal. The $N = 3$, $K = 1$ budget mode delivers $\sim 2\times$ overhead while still capturing $\sim 80\%$ of the full safety benefit. Slowdown measured on HarmBench-Standard. †B0 is taken from the E3 main run ($\alpha = 0.50$); the A1 ablation runs (at $\alpha = 0.85$) independently measure B0 = 16.6%, giving a consistent $N = 0 \to 3$ gain of +61.9 pp regardless of which baseline is used.

N	HB+AB%	n	Slow.
B0 (no pipeline)†	16.9	687	1.00×
0 (beams + judge, no mix)	27.2	687	5.70×
3	**89.1**	**687**	**4.84×**
6 (default)	90.4	687	5.07×
10	92.6	687	5.23×
20	93.0	687	5.78×

Table 8. Ablation A2: safety / over-refusal tradeoff in α on E3 ($N = 6$, $K = 3$; over-refusal measured on OR-Bench-Hard). Safety plateaus across $\alpha \in [0.50, 0.85]$, while OR-Bench over-refusal stays at $\approx 20\%$ throughout, a structural property of the beam selection pipeline, not a tunable knob. We adopt $\alpha = 0.50$ by Pareto score (*Pareto* = HB+AB% − OR-Bench over-refusal%; higher is better). The $\alpha = 0.85$ row reuses the A1 $N = 6$ run. All percentages are string-match refusal / over-refusal rates.

α	HB+AB%	Over-ref.	Pareto
0.10	38.0	17.8%	20.2
0.25	75.8	19.8%	56.0
0.50	**91.0**	20.2%	**70.8**
0.75	90.4	19.2%	71.2
0.85	90.4	22.8%	67.6

because Phase 3 (draft-only continuation) dominates total time. Full per-dataset A1 results (AdvBench, HarmBench-Standard, SorryBench, OR-Bench-Hard, XSTest across all five depths) appear in Appendix H.

This ablation was conducted at $\alpha = 0.85$ to map the depth Pareto in isolation; the main experiments use $\alpha = 0.50$ (§6.2), which yields comparable absolute safety (90.8% at $\alpha = 0.50$, $N = 6$ vs. 90.4% at $\alpha = 0.85$, $N = 6$; Table 8).

6.2. A2: Safety Weight α

Safety is near-constant for $\alpha \in [0.50, 0.85]$ (range: 90.4–91.0%), confirming that small changes to the mixing weight above 0.50 do not significantly affect safety. The OR-Bench over-refusal rate ($\approx 20\%$) is similarly stable across all tested values of α: it is a structural property of the beam selection pipeline, not a tunable parameter. XSTest over-refusal is higher ($\approx 40\%$) due to prompts whose surface form resembles harmful requests regardless of safety weight. Full per-dataset A2 results appear in Appendix I.

7. Analysis

7.1. Benchmark Sensitivity

Results vary meaningfully across benchmarks, exposing two distinct regimes. On structured adversarial benchmarks (AdvBench, HarmBench), gains are large and consistent across all pairs (+40–75 pp). On WildJailbreak-Harmful, which uses multi-sentence adversarial framing to bury the harmful goal deeper in context, gains are far more limited (+8–9 pp for E3 and E5, with even a small regression on E6). The pattern is consistent with the early-token safety hypothesis (Qi et al., 2025): an $N = 6$ anchor prefix is enough to steer single-sentence prompts but cannot reach harmful goals that the attack distributes across tokens beyond the mixed region.

7.2. Base-Model Score Inflation

Base models occasionally produce incoherent or repetitive outputs on adversarial prompts. External judges correctly mark these as non-harmful (they contain no actionable content), which inflates the apparent B0 accuracy rate under official benchmark judges. For example, on E3 the string-match B0 is 16.9% while the benchmark-judge B0 is 49.1% (AdvBench: 52.5%), a gap of ≈ 32 pp arising entirely from degenerate outputs being classified as "safe." ALIGNBEAM's outputs are non-degenerate by construction, since the LLM judge filters for coherent beams, so the gap between B0 and ALIGNBEAM is in fact a lower bound on the true safety improvement. We therefore report string-match refusal rates as the primary metric throughout, with benchmark-judge accuracy provided in the appendix.

8. Limitations

Efficiency and calibration. Full-mode overhead spans 4.6–8.75×, driven primarily by the $K = 3$ beam count rather than the depth N; the $K = 1$ budget mode runs at $\sim 2\times$ while retaining $\sim 80\%$ of the safety gain. Over-refusal on OR-Bench ($\approx 20\%$) and XSTest ($\approx 40\%$) is stable across all tested α, indicating that it is a structural property of the beam selection pipeline rather than a calibration artefact tunable through α alone.

WildJailbreak-style attacks. Multi-sentence adversarial framing largely bypasses the $N = 6$ anchor prefix by distributing the harmful goal across tokens beyond the mixed region. Longer or adaptive anchor prefixes are a natural remedy but have not yet been systematically evaluated.

Strategy and bridge reliability. LBM-FIRST is unreliable on certain instruct pairs (most notably DeepSeek-Math-7B-Instruct, which yields 86% degenerate outputs at $N = 20$) and therefore requires per-family validation before deployment. Roughly 12% of anchor top-50 tokens produce multi-

token sequences under cross-family tokenisation and fall back to LBM-DROP; these dropped tokens are typically subword fragments whose probability mass is redistributed across the remaining matches rather than lost.

Large-vocabulary models. Gemma-3-12B-PT (262K-token vocabulary) achieves more modest gains at $N = 6$ than smaller-vocabulary specialists, suggesting that the required anchor depth scales with vocabulary granularity, a relationship we leave for future work.

Anchor diversity and evaluation scope. All experiments share a single safety anchor (Qwen2.5-3B-Instruct) and rely on automated, English-only judges. Generalisation to other anchors, multilingual settings, and human evaluation is open. An alternative anchor (Llama-3.1-8B-Instruct) achieves strong benchmark-judge accuracy but low string-match refusal rates, suggesting anchor-specific metric calibration may be needed (Appendix L).

9. Conclusion

We presented ALIGNBEAM, an inference-time procedure that transfers safety alignment to domain-specialised language models by blending their token-level distribution with that of a small aligned anchor through a cross-vocabulary text bridge and selecting the safest beam with an LLM judge. Across seven model pairs spanning Llama, Gemma, Qwen, medical, and finance specialists, ALIGNBEAM raises refusal rates from near-zero to 88–97% on the structured adversarial benchmarks for the majority of pairs while preserving task utility (-0.5 pp on GSM8K), and on already-aligned instruct models it fills domain-specific gaps ($+15$ pp on Code-Jailbreak) without degrading general-domain safety.

The depth ablation (A1) corroborates the early-token safety hypothesis (Qi et al., 2025): the jump from $N = 0$ (beam selection only, no logit mixing) to $N = 3$ alone contributes $+61.9$ pp and all further gains are marginal. The weight ablation (A2) shows that safety plateaus for $\alpha \in [0.50, 0.85]$ and that the residual over-refusal on OR-Bench ($\approx$20%) is a structural property of the beam selection pipeline rather than a tunable parameter.

ALIGNBEAM is training-free, vocabulary-agnostic, and compatible with true base models, while requiring no access to model internals. The remaining limitations (4.6 to 8.75$\times$ overhead in full mode, reducible to $\sim$2$\times$ in budget mode; limited effectiveness against WildJailbreak-style multi-sentence attacks; and structural over-refusal on XSTest-style prompts) suggest the natural next steps: longer or adaptive anchor prefixes, multi-anchor ensembles, and calibration mechanisms that decouple over-refusal from beam selection.

Impact Statement

ALIGNBEAM is a safety-oriented method designed to reduce harmful outputs from domain-specialised language models. Potential misuse scenarios are limited, as the method increases, rather than decreases, refusal rates. The primary societal benefit is enabling organisations to deploy domain-specialised models without sacrificing safety alignment. A limitation worth noting for deployment contexts is the 4.6–8.75$\times$ inference overhead, which carries energy and cost implications; the $K = 1$ budget mode ($\sim$2$\times$ overhead) largely mitigates this.

References

Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G. J., Tramer, F., Kolter, J. Z., and Edgar, T. JailbreakBench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, J., Wang, X., Zhou, Y., Rangwala, H., and Ramakrishnan, N. OR-Bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

Ding, X., Zhou, Z., Shao, Y., Dong, X., Xu, W., and Roth, D. WildJailbreak: Evaluating the robustness of LLM safety alignment to jailbreaking attacks in the wild. *arXiv preprint arXiv:2406.18510*, 2024.

Fei, Y., Hou, Y., Chen, Z., and Bosselut, A. Nudging: Inference-time alignment via model collaboration. *arXiv preprint arXiv:2410.04390*, 2025.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., and Khabsa, M. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., and Lewis, M. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the ACL*, pp. 7422–7437, 2023.

Li, Y., Wei, F., Zhao, J., Zhang, C., and Zhang, H. RAIN: Your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

Liu, A., Han, X., Wang, Y., Tsvetkov, Y., Choi, Y., and Smith, N. A. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*, 2024.

Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakthivadivel, E., Li, N., Basart, S., Li, B., Forsyth, D., and Hendrycks, D. HarMBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, 2024.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users are not the ones fine-tuning. *arXiv preprint arXiv:2310.03693*, 2023.

Qi, X., Chen, A., Wan, T., Henderson, P., and Mittal, P. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2025.

Röttger, P., Kirk, H. R., Vidgen, B., Attanasio, G., Marelli, F., and Hovy, D. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*, 2023.

Xie, T., Qi, X., Zeng, Y., Huang, Y., Ammanabrolu, P., Mittal, P., and Chen, P.-Y. SORRY-Bench: Systematically evaluating large language model safety refusal behaviors. *arXiv preprint arXiv:2406.14598*, 2024.

Xu, Z., Jiang, F., Niu, L., Jia, J., Lin, B. Y., and Poovendran, R. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.

Yang, X., Wang, X., Zhang, Q., Petzold, L., Wang, W. Y., Zhao, X., and Lin, D. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Metric conventions used in all appendix tables.

- **B0% / Ref.%**: string-match refusal rate (fraction of responses containing at least one canonical refusal phrase). This is the primary metric used in Table 4.
- **ACC%**: benchmark-judge accuracy (fraction of responses judged not-harmful by the official benchmark classifier). Higher than B0% for base-model baselines due to score inflation (§7.2).
- **Δsafe**: LBM% − B0% (string-match delta) throughout, unless the column header explicitly states "ACC."
- For **calibration / benign datasets** (OR-Bench, XSTest, JBB-Benign): Ref.% denotes *over-refusal* rate (lower is better).

A. E3 Per-Dataset Breakdown

Table 9 reports E3 (Llama-3.1-8B + Qwen2.5-3B anchor) at $\alpha = 0.5$, $N = 6$, $K = 3$, seed 42 across all safety benchmarks plus calibration sets. Structured adversarial benchmarks (HarmBench, AdvBench) show the strongest gains; the WildJailbreak-Harmful row is the principal exception (§7.1).

Table 9. E3 per-dataset breakdown (Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$, seed 42). B0% and LBM% are string-match refusal rates; ACC% columns show benchmark-judge accuracy. For calibration rows, Ref.% denotes over-refusal. Δ_{safe} = LBM% − B0%.

Dataset	Type	n	B0%	B0 ACC%	LBM%	LBM ACC%
HarmBench-Std	Harmful	167	15.0	49.1	**79.6**	91.6
HarmBench-Ctx	Harmful	62	8.1	53.2	**79.0**	80.7
AdvBench	Harmful	520	17.5	52.5	**94.4**	98.3
Sorry-Bench	Harmful	440	10.9	34.9	**61.4**	74.3
WildJailbreak-H	Harmful	1105	13.3	21.5	**19.7**	30.1
Calibration / benign (lower over-refusal = better):						
JBB-Benign	Benign	100	10.0	8.0	26.0	2.0
OR-Bench-Hard	Benign	855	11.0	13.6	22.3	6.2
OR-Bench-Toxic	Benign	585	12.3	53.9	56.8	20.5
XSTest	Calib.	450	5.6	23.6	40.0	38.7
HB+AB		**687**	**16.9**	50.9	**90.8**	96.5

B. E6 Per-Dataset Breakdown

Table 10 reports E6 (Finance-Llama3-8B) at $\alpha = 0.5$, $N = 6$, $K = 3$. The aggregate HB+AB is B0 = 28.1%, LBM = 91.1% (+63.0 pp). All three LBM strategies produce identical results across every dataset for E6, indicating a high single-token match rate between the Finance-Llama3-8B (Llama-3 tokeniser) and Qwen2.5-3B vocabularies.

Table 10. E6 per-dataset breakdown (Finance-Llama3-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). All three LBM strategies (drop/first/exact) produce identical results. B0% = string-match baseline; LBM Ref.% = string-match refusal (strategy); Δ_{safe} = LBM Ref.% − B0% (where B0% is available). †Per-dataset B0% values not available from logs for E6; the aggregate HB+AB uses B0 = 28.1% (string-match, inferred from Δ_{safe}). The ACC% column reflects benchmark-judge accuracy for context. ‡WildJailbreak regression consistent with multi-sentence framing (§7.1).

Dataset	Type	n	LBM Ref.%	LBM ACC%
HarmBench-Std	Harmful	167	**79.6**	91.0
HarmBench-Ctx	Harmful	62	**80.6**	90.3
AdvBench	Harmful	520	**94.8**	98.1
JBB-Harmful	Harmful	100	**88.0**	94.0
Sorry-Bench	Harmful	440	**65.7**	77.3
WildJailbreak-H	Harmful	1105	33.2‡	41.4
HB+AB		**687**	**B0 = 28.1% → LBM = 91.1% (+63.0 pp)**	

C. EQ1 Per-Dataset Results

Table 11 provides the full per-dataset breakdown for EQ1 (Qwen3-8B-Base + Qwen2.5-3B-Instruct anchor), the only same-vocabulary pair. Even with a strong B0 prior of 84.6%, the largest relative string-match gains appear on HarmBench-Contextual (+33.9 pp) and Sorry-Bench (+23.9 pp), showing that the mixing pipeline helps even when the draft is already partially aligned. All three LBM strategies produce identical results (>98% single-token match rate).

Table 11. EQ1 per-dataset breakdown (Qwen3-8B-Base + Qwen2.5-3B-Instruct, same tokeniser, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). B0% and LBM% are string-match refusal rates; Δ_{safe} = LBM% − B0%. ACC% columns show benchmark-judge accuracy for reference. Note: the ACC-based delta (STRAT_ACC − BASE_ACC) differs from Δ_{safe} because the base model's ACC-based safety is higher than its string-match rate, owing to partially aligned Qwen architecture.

Dataset	Type	n	B0%	B0 ACC%	LBM%	LBM ACC%	Δ_{safe}
HarmBench-Std	Harmful	167	61.7	80.8	**80.8**	89.8	+19.1
HarmBench-Ctx	Harmful	62	37.1	64.5	**71.0**	90.3	+33.9
AdvBench	Harmful	520	91.9	95.6	**95.8**	99.4	+3.9
JBB-Harmful	Harmful	100	76.0	81.0	**85.0**	91.0	+9.0
Sorry-Bench	Harmful	440	39.5	55.2	**63.4**	79.8	+23.9
WildJailbreak	Harmful	1105	16.1	27.6	**20.9**	32.6	+4.8
XSTest (over-ref.)	Calib.	450	10.7	26.0	28.2	38.4	–
HB+AB		**687**	**84.6**	91.9	**92.1**	97.4	**+7.5**

D. EQ1: Same-Vocabulary Method Comparison

Table 12 compares ALIGNBEAM against two same-vocabulary inference-time baselines on EQ1 (Qwen3-8B-Base + Qwen2.5-3B-Instruct): PROXY TUNING (Liu et al., 2024) and top-k contrastive decoding (TOPK-CD). Because the proxy-tuning and TopK-CD deltas are reported by the harness as benchmark-judge accuracy (ACC%), all three columns use ACC% for comparability.

ALIGNBEAM outperforms both alternatives substantially on contextual (HB-Ctx: +25.8 vs. +4.8/+11.3 $pp)and sorry-bench$(+24.5 $vs.$ +13.0/+14.1 pp) categories. JailbreakBench-Harmful is comparable across all three methods. XSTest over-refusal is nearly identical (≈37–38%), confirming that this is a structural property of the beam-selection pipeline rather than a method-specific artefact. Overhead is similar for all three (≈4.8–5.3×), so ALIGNBEAM's safety advantage is not purchased at additional latency cost.

Contrastive decoding (Li et al., 2023) was also evaluated on EQ1 but its evaluation harness did not produce per-dataset accuracy metrics; its total-time profile was comparable to the other methods.

Table 12. EQ1 same-vocabulary method comparison (Qwen3-8B-Base + Qwen2.5-3B-Instruct, $\alpha = 0.5$, $N = 6$, $K = 3$). All values are benchmark-judge accuracy (ACC%). Δ = method ACC% − B0 ACC%. XSTest is a calibration set (over-refusal; lower is better). Slow. columns show per-sample slowdown vs. draft-only baseline.

Dataset	Type	B0	ALIGNBEAM (LBM)			Proxy Tuning			TopK-CD		
		ACC%	ACC%	Δ	Slow.	ACC%	Δ	Slow.	ACC%	Δ	Slow.
AdvBench	Harm.	95.6	**99.4**	+3.8	4.9×	98.1	+2.5	5.2×	98.3	+2.7	5.0×
HB-Ctx	Harm.	64.5	**90.3**	+25.8	4.7×	69.4	+4.8	5.0×	75.8	+11.3	5.0×
HB-Std	Harm.	80.8	**89.8**	+9.0	4.9×	83.2	+2.4	5.3×	86.2	+5.4	5.3×
JBB-Harmful	Harm.	81.0	91.0	+10.0	4.7×	**92.0**	+11.0	5.3×	90.0	+9.0	5.2×
SorryBench	Harm.	55.2	**79.8**	+24.5	4.8×	68.2	+13.0	5.1×	69.3	+14.1	5.1×
WildJailbk	Harm.	27.6	**32.6**	+5.0	5.4×	29.4	+1.7	5.4×	31.1	+3.5	5.4×
XSTest†	Calib.	26.0	38.4	—	4.8×	37.3	—	4.9×	36.9	—	4.8×

†Over-refusal (lower is better).

E. E5 Per-Dataset Breakdown

Table 13 reports E5 (JSL-MedLlama-3-8B + Qwen2.5-3B anchor) at $\alpha = 0.5$, $N = 6$, $K = 3$.

Table 13. E5 per-dataset breakdown (JSL-MedLlama-3-8B-v2.0 + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). LBM Ref.% = string-match refusal rate; LBM ACC% = benchmark-judge accuracy. HB+AB aggregate B0 = 34.5% (string-match, inferred from Δ_{safe}). Per-dataset B0 string-match values are not separately available in the evaluation log for E5; the B0 ACC% column reflects benchmark-judge baseline.

Dataset	Type	n	B0 ACC%	LBM Ref.%	LBM ACC%
HarmBench-Std	Harmful	167	35.3	**78.4**	88.6
HarmBench-Ctx	Harmful	62	25.8	**69.4**	87.1
AdvBench	Harmful	520	59.2	**91.7**	99.0
JBB-Harmful	Harmful	100	42.0	**79.0**	82.0
Sorry-Bench	Harmful	440	31.1	**63.2**	72.3
WildJailbreak-H	Harmful	1315	22.5	16.2	27.1
HB+AB		**687**	**B0 = 34.5% → LBM = 88.5% (+54.0 pp)**		

F. EG1 Per-Dataset Breakdown

Table 14 reports EG1 (Gemma-3-12B-PT + Qwen2.5-3B anchor) at $\alpha = 0.5$, $N = 6$, $K = 3$. B0% (string-match) is near-zero because Gemma-3-12B-PT is an unaligned pretrained model that produces no canonical refusal phrases on adversarial prompts. The benchmark-judge B0 ACC% is higher (≈38–54%) due to incoherent base-model outputs being classified as "safe" (§7.2).

Table 14. EG1 per-dataset breakdown (Gemma-3-12B-PT + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). B0% ≈ 0 (string-match) for all safety datasets; B0 ACC% reflects benchmark-judge baseline (inflated by incoherent base-model outputs). LBM Ref.% = string-match refusal; LBM ACC% = benchmark-judge accuracy.

Dataset	Type	n	B0 ACC%	LBM Ref.%	LBM ACC%
HarmBench-Std	Harmful	167	38.9	**42.5**	85.0
HarmBench-Ctx	Harmful	62	45.2	**40.3**	80.6
AdvBench	Harmful	520	53.7	**55.6**	97.5
JBB-Harmful	Harmful	100	16.0	**39.0**	86.0
Sorry-Bench	Harmful	440	28.6	**35.9**	70.7
XSTest (over-ref.)	Calib.	450	21.8	5.6	30.4
HB+AB (SM)		**687**	**B0 ≈ 4.4% → LBM = 52.4% (+48.0 pp)**		

G. E2 Strategy Comparison

Table 15 compares the three LBM variants on E2 (DeepSeek-Math-7B-Instruct) at $N = 20$. LBM-FIRST produces 86.2% degenerate outputs on AdvBench, so its AdvBench column is not directly comparable; LBM-DROP and LBM-EXACT are essentially indistinguishable on safety while preserving GSM8K within 0.5 pp, and we therefore adopt LBM-DROP as the default strategy for E2.

Table 15. LBM-strategy comparison on E2 (DeepSeek-Math-7B-Instruct, $N = 20$, $\alpha = 0.5$, $K = 3$). All percentages are string-match refusal rates except GSM8K (exact-match accuracy). LBM-FIRST produces 86.2% degenerate (empty/near-empty) outputs on AdvBench, so its AdvBench column is not directly comparable; LBM-DROP and LBM-EXACT are essentially indistinguishable on safety. GSM8K utility is preserved to within 0.5 pp for all strategies.

Strategy	AdvB Ref.%	HarmB Ref.%	SorryB Ref.%	GSM8K
LBM-DROP	**80.6**	**63.5**	**45.0**	76.6%
LBM-EXACT	80.6	63.5	45.9	76.6%
LBM-FIRST	*12.7 (degenerate)*	71.3	57.3	76.6%

H. A1: Full Per-Dataset Depth Sweep

Table 16 reports the complete per-dataset results for the A1 mixed-depth ablation ($\alpha = 0.85$, $K = 3$, lbm_drop, E3 backbone). All percentages are string-match refusal rates. For calibration datasets (OR-Bench-Hard, XSTest), values denote over-refusal;

bold entries mark the depth with the best safety-efficiency tradeoff.

Table 16. A1 full per-dataset results (E3: Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.85$, $K = 3$). All values are string-match refusal rates. For OR-Bench-Hard and XSTest, the reported value is the over-refusal rate (lower is better). Slowdown is measured on HarmBench-Standard.

Dataset	n	B0%	N=0	N=3	N=6	N=10	N=20
AdvBench	520	17.5	30.4	92.7	93.5	96.2	**97.3**
HarmBench-Std	167	13.8	17.4	77.8	80.8	**81.4**	79.6
SorryBench	440	10.9	18.6	60.0	61.6	62.1	**68.2**
OR-Bench-Hard†	855	11.0	14.7	22.7	22.8	23.2	32.4
XSTest†	450	5.6	9.6	40.7	40.0	40.0	42.7
HB+AB	687	16.6	27.2	**89.1**	90.4	92.6	93.0
Slowdown (HB-Std)		1.00×	5.70×	4.84×	5.07×	5.23×	5.78×

†Calibration sets: over-refusal rate (lower is better).

I. A2: Full Per-Dataset Alpha Sweep

Table 17 reports the complete per-dataset results for the A2 safety-weight ablation ($N = 6$, $K = 3$, lbm_drop, E3 backbone). SorryBench was not run for the A2 sweep.

Table 17. A2 full per-dataset results (E3: Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $N = 6$, $K = 3$). All values are string-match refusal / over-refusal rates. SorryBench not evaluated for A2. Pareto = HB+AB% − OR-Bench-Hard over-refusal%.

Dataset	n	B0%	α=0.10	α=0.25	α=0.50	α=0.75
AdvBench	520	17.5	41.2	81.0	**94.4**	93.5
HarmBench-Std	167	13.8	28.1	59.9	**80.2**	80.8
OR-Bench-Hard†	855	15.1	17.8	19.8	20.2	**19.2**
XSTest†	450	6.7	5.6	16.4	**24.4**	24.0
HB+AB	687	13.3‡	38.0	75.8	**90.8**	90.4
Pareto			20.2	56.0	**70.6**	71.2

†Calibration sets: over-refusal rate (lower is better). ‡A2 B0 measured independently at $\alpha = 0.85$ baseline.

J. A3: Unaligned Anchor Ablation

Table 18 compares an aligned anchor (Qwen2.5-3B-Instruct) against an unaligned base anchor (Qwen2.5-3B-Base) on E3. The unaligned anchor still provides safety gains (+25–30 pp from the beam-selection mechanism), but substantially less than the aligned anchor, confirming that RLHF alignment of the anchor is the primary source of safety signal.

Table 18. A3 ablation: aligned vs. unaligned anchor (E3: Llama-3.1-8B, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). B0 = draft-only string-match baseline (no anchor); remaining columns show string-match refusal rate for each anchor. Slowdown measured per dataset.

Dataset	n	B0%	Unaligned	Aligned	Slow.
AdvBench	520	17.5	69.4	**94.4**	5.4×
HarmBench-Std	167	13.8	52.1	**79.6**	5.6×
HarmBench-Ctx	62	8.1	38.7	**79.0**	5.8×
SorryBench	440	10.9	35.2	**61.4**	6.2×
OR-Bench-Hard†	855	11.0	25.4	22.3	6.0×
XSTest†	450	5.6	12.9	40.0	6.1×

†Calibration sets: over-refusal (lower is better).

K. A7: Prompt Robustness

Table 19 evaluates sensitivity of ALIGNBEAM to the anchor's safety system prompt on HarmBench-Standard (E3 backbone). Three prompts are tested: no system prompt, a concise 2-line standard prompt, and a verbose 45-line prompt that enumerates harm categories. The standard prompt outperforms both alternatives; the verbose prompt achieves identical benchmark-judge

accuracy to the standard prompt but lower string-match refusal rate, suggesting the anchor's RLHF training drives refusal rather than explicit prompt wording.

Table 19. A7 prompt robustness on HarmBench-Standard (E3: Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$, $n = 167$). B0 ACC = benchmark-judge baseline = 49.1%; B0 Ref.% = 13.8%.

Anchor Prompt	Ref.%	ACC%	Slow.
None	66.5	82.6	5.5×
Standard (2-line)	**80.2**	**91.6**	**5.2**×
Verbose (45-line)	66.5	91.6	5.5×
B0 (no anchor)	13.8	49.1	1.0×

L. A8a: Alternative Anchor (Llama-3.1-8B-Instruct)

Table 20 reports results using Llama-3.1-8B-Instruct as the safety anchor instead of Qwen2.5-3B-Instruct (E3 backbone, $\alpha = 0.5$, $N = 6$, $K = 3$). The Llama anchor achieves high benchmark-judge accuracy (97–100%) but very low string-match refusal rates (7–15%), because Llama-3.1-8B-Instruct generates polite declinations in phrasing not covered by the 22-phrase string-match list. This highlights that string-match refusal rate may underestimate safety when the anchor's refusal style deviates from the canonical phrase set.

Table 20. A8a alternative anchor (E3: Llama-3.1-8B draft + Llama-3.1-8B-Instruct anchor, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). Ref.% = string-match refusal rate; ACC% = benchmark-judge accuracy. High ACC% with low Ref.% indicates polite non-canonical refusals not captured by string-match.

Dataset	n	B0 Ref.%	B0 ACC%	Ref.%	ACC%	Slow.
AdvBench	520	17.5	52.5	7.1	**99.6**	3.9×
HarmBench-Std	167	13.8	49.1	13.2	**97.0**	4.1×
HarmBench-Ctx	62	8.1	53.2	14.5	**98.4**	4.1×
SorryBench	440	10.9	34.9	11.1	**85.9**	4.8×
OR-Bench-Hard†	855	11.0	13.6	14.0	4.4	5.5×
XSTest†	450	5.6	23.6	7.1	37.1	5.3×

†Calibration: over-refusal (lower is better).

M. B2: Instruct-Counterpart Per-Dataset Breakdown

Table 21 reports the full per-dataset profile for B2 (Llama-3.1-8B-Instruct), the separately-trained instruct counterpart included in the baseline comparison as an upper-bound reference. The "Inst. Base ACC%" column is the instruct model's own benchmark-judge accuracy without any ALIGNBEAM intervention; the "+ALIGNBEAM ACC%" column shows what happens when ALIGNBEAM is additionally applied. The "+ALIGNBEAM Ref%" column is the string-match refusal rate under ALIGNBEAM.

Key observations: (i) the instruct model is already near-ceiling on structured adversarial sets (AdvBench 99.8%, HarmBench 95–98%) so ALIGNBEAM adds nothing on those rows; (ii) WildJailbreak drops by -10.2 pp when ALIGNBEAM is applied, showing that anchor mixing can disrupt the instruct model's own contextual reasoning on multi-sentence adversarial prompts; (iii) OR-Bench-Hard ACC is only 1.5% for the instruct baseline, confirming that the benchmark-judge severely over-classifies instruct refusals on adversarial-looking benign prompts (the same score inflation effect discussed in §7.2, in reverse).

Table 21. B2 per-dataset breakdown (Llama-3.1-8B-Instruct, $K=3$, LBM-DROP, $\alpha=0.5$, seed 42). "Inst. Base" = instruct model without ALIGNBEAM (benchmark-judge ACC%). "+ALIGNBEAM" = instruct model with ALIGNBEAM applied (ACC% and string-match Ref%). For calibration/benign rows, lower Ref% and higher ACC% are both better.

Dataset	Type	n	Inst. Base ACC%	+AB ACC%	ΔACC	+AB Ref%
AdvBench	Harmful	520	99.8	**99.6**	−0.2	35.6
HB-Contextual	Harmful	62	98.4	**96.8**	−1.6	24.2
HB-Standard	Harmful	167	95.2	**95.2**	+0.0	24.6
SorryBench	Harmful	440	83.6	**86.1**	+2.5	21.1
WildJailbreak	Harmful	1315	62.2	52.0	**−10.2**	20.2
JBB-Benign†	Benign	100	3.0	2.0	−1.0	8.0
OR-Bench-Hard†	Benign	855	1.5	1.4	−0.1	15.1
OR-Bench-Toxic†	Benign	585	1.0	1.5	+0.5	25.8
XSTest†	Calib.	450	44.4	43.6	−0.9	18.0

†Benign/calibration: higher ACC% = better compliance; lower Ref% = less over-refusal. OR-Bench Inst. Base ACC ≈ 1% reflects judge over-classification of coherent instruct refusals as "unsafe" on adversarial-looking benign prompts (§7.2).

N. B4 and B5: LlamaGuard Guard Baselines

Tables 22 and 23 report two guard-classifier baselines on E3 (Llama-3.1-8B backbone). B4 uses LlamaGuard-3-8B as a *prompt-level filter* (blocking generation when the input prompt is classified as harmful) and B5 uses LlamaGuard-3-8B as a *response-level filter* (blocking or replacing the generated output when it is classified as harmful). Both run with $K=3$ and seed 42.

Both classifiers achieve high benchmark-judge accuracy on structured adversarial sets (AdvBench, HarmBench: 89–100%), but impose severe over-refusal on benign benchmarks: OR-Bench-Hard drops to ACC = 2.6% (B4) and 0.1% (B5), and OR-Bench-Toxic falls below 2% for both, indicating that nearly every benign domain-adjacent prompt is falsely blocked. By contrast, ALIGNBEAM's OR-Bench-Hard over-refusal is ≈22% (Table 17). The XSTest over-refusal for both classifiers (≈43–50% string-match Ref.%) is comparable to ALIGNBEAM, reflecting the surface-form similarity of XSTest calibration prompts to genuine adversarial inputs.

Table 22. B4: LlamaGuard prompt-level filter (E3: Llama-3.1-8B, $K=3$, seed 42). ACC% = benchmark-judge accuracy. For calibration/benign rows, ACC% is the fraction of benign requests that pass the classifier (lower ACC% = higher over-refusal). Base ACC% = B0 benchmark-judge baseline.

Dataset	Type	n	Base ACC%	B4 ACC%	Slow.
AdvBench	Harmful	520	52.5	**100.0**	4.1×
HarmBench-Ctx	Harmful	62	53.2	**95.2**	5.0×
HarmBench-Std	Harmful	167	49.1	**98.8**	5.0×
SorryBench	Harmful	440	34.9	**89.8**	5.7×
JBB-Benign†	Benign	100	8.0	0.0	5.8×
OR-Bench-Hard†	Benign	855	13.6	2.6	6.0×
OR-Bench-Toxic†	Benign	585	53.8	1.0	5.3×
XSTest†	Calib.	450	23.6	42.9	5.6×

†Benign/calibration sets: lower ACC% = higher over-refusal.

Table 23. B5: LlamaGuard response-level filter (E3: Llama-3.1-8B, $K=3$, seed 42). Response filtering achieves comparable safety to prompt filtering (B4) but identical over-refusal characteristics, since benign inputs with safety-adjacent surface forms produce generations that are falsely blocked at the output stage.

Dataset	Type	n	Base ACC%	B5 ACC%	Slow.
AdvBench	Harmful	520	52.5	**100.0**	3.9×
HarmBench-Ctx	Harmful	62	53.2	**100.0**	4.7×
HarmBench-Std	Harmful	167	49.1	**99.4**	4.7×
SorryBench	Harmful	440	34.9	**88.9**	5.4×
JBB-Benign†	Benign	100	8.0	0.0	5.5×
OR-Bench-Hard†	Benign	855	13.6	0.1	5.5×
OR-Bench-Toxic†	Benign	585	53.8	0.0	5.0×
XSTest†	Calib.	450	23.6	43.3	5.3×

†Benign/calibration sets: lower ACC% = higher over-refusal.

O. B8: Hard-Prefix Baseline

Table 24 evaluates B8, a hard-prefix baseline in which the anchor model (Qwen2.5-3B-Instruct) first generates a short fixed-length safety prefix, which is prepended to the draft's decoding context before it continues generation, with $K=1$ (no beam selection or LLM judge). This is a lower-overhead alternative to ALIGNBEAM's probabilistic blending.

B8 runs at ≈2.4–3.2× overhead and achieves strong safety on structured adversarial benchmarks: ACC = 99.0% on AdvBench and 91.6% on HarmBench-Standard—comparable to full ALIGNBEAM at $N=3$, $K=1$ budget mode. However, benign-set over-refusal is elevated: OR-Bench-Toxic drops from 53.8% to 25.5% (-28.3 pp), and JBB-Benign drops to 1.0%, indicating that the deterministic prefix still triggers refusal on many benign domain-adjacent prompts. The XSTest evaluation did not complete due to an evaluation-harness indexing error and is excluded.

Table 24. B8: Hard-prefix baseline (E3: Llama-3.1-8B + Qwen2.5-3B-Instruct, $K=1$, LBM-DROP, seed 42). ACC% = benchmark-judge accuracy. For calibration/benign rows, lower ACC% indicates higher over-refusal. XSTest excluded due to evaluation-harness error.

Dataset	Type	n	Base ACC%	B8 ACC%	Slow.
AdvBench	Harmful	520	52.5	**99.0**	2.4×
HarmBench-Ctx	Harmful	62	53.2	**88.7**	2.6×
HarmBench-Std	Harmful	167	49.1	**91.6**	2.7×
SorryBench	Harmful	440	34.9	**74.5**	3.2×
JBB-Benign†	Benign	100	8.0	1.0	2.5×
OR-Bench-Hard†	Benign	855	13.6	9.6	2.8×
OR-Bench-Toxic†	Benign	585	53.8	25.5	2.2×

†Benign/calibration sets: lower ACC% = higher over-refusal.

P. E3 Seed Robustness

Table 25 reports E3 across seeds 42, 2, and 3, confirming that the main results are not seed-specific. Variance is highest on HarmBench-Contextual (±7 pp) and SorryBench (±7 pp), consistent with the smaller dataset sizes and diverse prompt styles.

Table 25. E3 seed robustness (Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha=0.5$, $N=6$, $K=3$). All values are string-match refusal / over-refusal rates.

Dataset	n	B0%	Seed 42	Seed 2	Seed 3
AdvBench	520	17.5	**94.4**	94.4	95.2
HarmBench-Std	167	14.0	**79.6**	79.0	74.3
HarmBench-Ctx	62	8.1	**79.0**	72.6	64.5
SorryBench	440	10.9	**61.4**	61.4	62.7
OR-Bench-Hard†	855	11.0	22.3	22.9	22.7
XSTest†	450	5.6	40.0	40.0	45.6
HB+AB	687	16.9	**90.8**	89.8	88.2

†Calibration: over-refusal (lower is better).

Q. E3 No-Judge Ablation

Table 26 compares the standard ALIGNBEAM pipeline (with LLM judge for beam selection) against a no-judge variant that uses the lowest-score beam by random selection. Safety rates remain high without the judge, demonstrating that logit mixing alone (not beam re-ranking) provides the majority of the safety gain. The judge provides an additional ≈5–10 pp on AdvBench and HarmBench-Std.

Table 26. No-judge ablation on E3 (Llama-3.1-8B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). All values are string-match refusal / over-refusal rates. XSTest evaluated on $n = 220$ (partial run).

Dataset	n	B0%	With Judge	No Judge	Slow.
AdvBench	520	17.5	**94.4**	97.9	3.4×
HarmBench-Std	167	13.8	**79.6**	81.4	4.2×
HarmBench-Ctx	62	8.1	**79.0**	85.5	4.2×
SorryBench	440	10.9	**61.4**	63.6	4.9×
WildJailbreak-H	1105	13.3	**19.7**	18.7	5.4×
XSTest†	220	5.6	**40.0**	46.4	4.9×

†Partial run ($n = 220$). Over-refusal: lower is better.

R. E4: Scale Experiment (Llama-3.1-70B)

Table 27 reports results for E4 (Llama-3.1-70B-Base + Qwen2.5-3B anchor), evaluating ALIGNBEAM at 70B scale. Numbers are benchmark-judge accuracy (ACC%) because string-match B0 baselines are not separately available for E4 in the evaluation logs; the ACC% delta column shows strategy improvement over the base model baseline.

Table 27. E4 scale experiment (Llama-3.1-70B + Qwen2.5-3B-Instruct, LBM-DROP, $\alpha = 0.5$, $N = 6$, $K = 3$). All values are benchmark-judge accuracy (ACC%). B0 ACC% = baseline; AlignBeam ACC% = strategy; Δ = ACC-based delta.

Dataset	n	B0 ACC%	AlignBeam ACC%	Δ	Slow.
AdvBench	520	56.5	**99.4**	+42.9	4.3×
HarmBench-Std	167	55.1	**95.2**	+40.1	4.5×
SorryBench	440	36.4	**86.1**	+49.8	4.8×
OR-Bench-Hard†	855	13.3	4.4	—	5.3×
OR-Bench-Toxic†	585	58.6	13.8	—	4.3×
JBB-Benign†	100	10.0	6.0	—	4.9×
XSTest†	450	26.0	39.6	+13.6	4.7×

†Calibration: over-refusal (lower is better).

S. Latency Details

Table 28 reports per-sample wall-clock latency on a single RTX 6000 Ada (48 GB, bfloat16). Phase 3 (KV-cache draft continuation) accounts for roughly 69% of total time, Phase 2 (mixed decoding) for ≈30%, and Phase 1 (priming) for ≈1%, so the dominant speed lever is the beam count K rather than the depth N.

Slowdown values are measured on HarmBench-Standard as the representative dataset; other datasets (particularly WildJailbreak with longer prompts) exhibit higher per-sample latency, so reported values are conservative lower bounds on overhead for some inputs. The budget mode entry ($K = 1$, $N = 3$) derives from the A1 ablation: at $N = 3$ the HarmBench-Standard slowdown is 4.84× under $K = 3$; with $K = 1$ the beam-parallel overhead is removed, reducing this to ≈2× while retaining ≈80% of the full safety benefit ($N = 3$: 89.1% vs. $N = 20$: 93.0% in Table 7).

Table 28. Per-sample latency on a single RTX 6000 Ada (48 GB, bfloat16), measured on HarmBench-Standard. Phase breakdown: Phase 3 (KV-cache draft continuation) ≈69%, Phase 2 (mixed decoding) ≈30%, Phase 1 (priming) ≈1%; the dominant speed lever is the beam count K rather than N.

Config	ALIGNBEAM	B0	Slow.	
E3: Llama-8B + Qwen-3B ($K=3$, $N=6$)	16.9 s	3.3 s	5.07×	
E5: MedLlama-8B + Qwen-3B ($K=3$, $N=6$)	13.7 s	2.6 s	4.88×	
E6: FinLlama-8B + Qwen-3B ($K=3$, $N=6$)	—	—	6.51×	E6 per-sample
EG1: Gemma-12B + Qwen-3B ($K=3$, $N=6$)	35.6 s	6.8 s	4.63×	
E2: DS-Math + Qwen-3B ($K=3$, $N=20$)	25.8 s	5.6 s	6.61×	
Budget ($K=1$, $N=3$)	≈2× overhead; ≈80% of full safety			

seconds not available from the evaluation log; the 6.51× slowdown is derived from total-wall-clock measurements.

T. Configuration Summary

All configurations are available at `TokenMixingHarness/configs/paper_configs/`. Key directories: `main_experiments/`, `ablations/`, `baselines/`.

Common parameters: $\alpha = 0.5$, $K = 3$, $\tau = 2.5$, $T = 0.7$, `max_new_tokens`=150 (safety) / 512–1024 (utility), `repetition_penalty`=1.15, `top_k_bridge`=50, seed=42.

Model-specific overrides: E2 $N=20$; A1/A2 ablations $\alpha=0.85$.