POSITIONAL INFERENCE FOR STRONGER ATTACKS ON LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimization-based adversarial attacks on large language models (LLMs) perform coordinate descent over a fixed set of token positions. While seminal works focus exclusively on suffix-based attacks, recent studies suggest that inserting tokens into alternate positions (prefix and infix) can achieve higher attack success rates (ASR). Although methods that move beyond suffixes often improve ASR, their position selection usually relies on architecture-dependent heuristics. We cast position selection as Bayesian inference: intermediate optimizer evaluations that existing attacks discard serve as observations about which positions are worth attacking, and existing heuristics can be formulated as prior beliefs about impactful positions. Because the resulting posteriors are Gaussian, beliefs inferred from individual attacks can be merged in closed form, so computation spent attacking one model becomes a positional prior for attacking another. Across 12 LLMs ranging from 0.6B to 32B parameters, and despite the exploration cost of jointly inferring positions and optimizing tokens, our attack improves mean ASR by 11.1 percentage points over suffix-based GCG and 3.6 points over attention-guided SlotGCG. The inferred posteriors reveal that vulnerability concentrates at both ends of the prompt, reconciling conflicting reports that favor suffixes or prefixes, yet correlates only weakly with the attention signal current heuristics rely on. Most interestingly, we find that positional knowledge transfers: in leave-one-model-out experiments, using posteriors merged from the other 11 models as a prior improves a gradient-free attack on the held-out model by 9–11 points over suffix placement and 6–7 points over transferred attention heuristics, matching suffix-only ASR with roughly a quarter of the query budget. Our results suggest that where to attack is learnable, transferable, and enables an important form of adaptive attacks for language models.

1 INTRODUCTION

Adversarial attacks are our primary tool for understanding the robustness (or lack thereof) of machine learning models, and for large language models (LLMs) they have become central to safety evaluation before deployment (Perez et al., 2022; Mazeika et al., 2024). A long line of work has shown that defenses which appear robust to a fixed set of attacks are routinely circumvented by attackers who tailor their strategy to the defense at hand (Carlini & Wagner, 2017; Athalye et al., 2018; Tramèr et al., 2020). The pitfall is that any static assumption made by a model of an attacker need not be made by a real adversary, and evaluations that rely on static assumptions can vastly overestimate the robustness of the models we deploy. The Greedy Coordinate Gradient (GCG) attack (Zou et al., 2023), which underpins much of the optimization-based jailbreak literature, makes one such static assumption in placing all adversarial tokens in the suffix of the prompt.

Both empirically and theoretically, there are compelling reasons to favor the suffix: tokens at the end of the prompt sit immediately before the model's response, a suffix can be appended to any prompt (which makes universal attacks straightforward), and successful suffixes have been found to attract a large share of the model's attention (Wang et al., 2025; Ben-Tov et al., 2026b). However, recent work has shown that adversarial prefixes can also produce successful jailbreaks while receiving comparatively little attention in most layers (Eddoubi et al., 2026a). Further, SLOTGCG (Jeong et al., 2026) finds that inserting tokens at infix positions, selected by an attention-based vulnerability score, yields substantially higher attack success rates (ASR) than suffix placement. These works clearly demonstrate that suffix-only GCG is insufficient for a comprehensive robustness analysis. Of course, while the attention-based allocation used by SLOTGCG demonstrates the flaw in only using suffix-based GCG, it proposes to replace it with yet

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.

another static choice made before optimization begins. Adding a second static heuristic into our model of attackers does not address the need for adaptive attacks for robustness evaluation.

In this work, we study a framework for adaptive positional attacks through Bayesian inference, in which the optimal allocation of adversarial tokens is an uncertain quantity learned during the attack. The idea behind our approach is simple: at every step, GCG evaluates hundreds of candidate substitutions and discards all but the best. We posit that if these candidates were spread across different allocations (that is, beyond just the suffix or a fixed heuristic), then discarded evaluations could be viewed as noisy observations of how useful each position is to attack. We use these observations to update a posterior over a latent positional profile, and where we believe existing heuristics are strong, we encode them as priors (a sufficiently concentrated prior recovers the heuristic itself). By choosing a simple linear-Gaussian model, posterior updates are closed-form and the best allocation under any profile can be computed exactly despite the combinatorial search space. We select allocations with Thompson sampling (Thompson, 1933; Russo et al., 2018) and call the resulting attack SPACE (*Sequential Posterior-guided Allocation from Conjugate Estimation*). A considerable benefit of this probabilistic treatment is that posteriors from different prompts and models can be merged in closed form. The computation spent attacking one model can therefore inform attacks on others, including models whose gradients we cannot access.

We evaluate SPACE on twelve instruction-tuned LLMs from six model families, ranging from 0.6B to 32B parameters. In the white-box setting, SPACE improves mean ASR by 11.1 percentage points over GCG and 3.6 points over SLOTGCG, despite the cost of exploring positions during the attack. The largest relative gains are on the models most robust to existing attacks: on the two Gemma-4 models, ASR rises from 6–10% to 18–20%. SPACE also jailbreaks 62 of the 600 model–behavior pairs that neither baseline can. We also consider a black-box setting where only output probabilities are available. There, merging the posteriors of eleven models into a prior for the twelfth improves the gradient-free BEAST attack (Sadasivan et al., 2024) by 9–11 percentage points over suffix placement and 6–7 points over transferred attention heuristics. It matches the ASR of suffix placement with roughly a quarter of the queries.

Beyond improvements in ASR, we highlight that our framework allows us to validate (or invalidate) the heuristics and explanations that motivate existing attacks. With a uniform prior, SPACE's posteriors are learned from attack outcomes alone, without access to attention, and so give an independent estimate of where each model is vulnerable. These posteriors are consistently U-shaped, concentrating on the prefix and suffix with some mass on early-position infix positions. This suggests that the suffix and prefix perspectives in the literature are each partially correct but incomplete. The posteriors also correlate only weakly with the attention-based scores used by SLOTGCG (mean Spearman $\rho = 0.34$), which casts doubt on attention as a proxy for positional vulnerability. Finally, we observe that even a fixed allocation learned from eleven other models posteriors outperforms heuristic allocation even if we do not perform inference during optimization. The state-of-the-art ASR, mechanistic attack insights, and transferability and reusability of our proposed attack indicate an important step towards adaptive, comprehensive, and compute efficient optimization-based attacks on language models. In summary, our contributions are:

- We formulate adversarial position selection as Bayesian inference, with the optimizer's intermediate losses as observations and existing heuristics as priors (Sections 3 and 4).
- We develop SPACE, an adaptive positional attack that works with both gradient-based and gradient-free optimizers, and show how its posteriors can be merged in closed form to transfer across models (Section 4).
- We show that SPACE outperforms existing positional attacks on twelve LLMs, and that transferred posteriors improve black-box attacks on held-out models (Section 5).
- We use SPACE's posteriors to test positional heuristics, finding a U-shaped vulnerability profile that only loosely tracks attention (Section 5).

2 Related Works

Jailbreak attacks. Since GCG (Zou et al., 2023), many jailbreak attacks have been proposed, based on genetic algorithm or beam search (Liu et al., 2024; Sadasivan et al., 2024), iterative refinement by an attacker LLM (Chao et al., 2025; Mehrotra et al., 2024), improvements to GCG's optimization (Jia et al., 2025), and simple attacks adapted to each target model (Andriushchenko et al., 2025); see Yi et al. (2024) for a thorough treatment. These works are mostly concerned with what adversarial content to use, and token-level attacks typically inherit GCG's placement of adversarial tokens in the suffix. Our framework is complementary and can be wrapped around any token-level optimizer, as we show for both GCG and BEAST.

Positional heuristics and their explanations. Mechanistic studies link the success of adversarial suffixes to attention, finding that effective suffixes draw a large share of attention and hijack the representations of the template tokens immediately preceding generation (Wang et al., 2025; Ben-Tov et al., 2026a). SLOTGCG (Jeong et al., 2026) turns this account into a placement rule, allocating adversarial tokens in proportion to a softmax over the attention each slot receives from the chat template's control tokens. However, adversarial prefixes can succeed while receiving comparatively little attention in most layers (Eddoubi et al., 2026b), whether attention explains model behavior is itself debated (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019), and other accounts of jailbreak success point to failure modes of safety training such as competing objectives (Wei et al., 2023) or to a refusal direction that adversarial suffixes suppress (Arditi et al., 2024). Because our posteriors are learned from attack outcomes alone, they can be used to test such explanations wherever they make predictions about position.

Choosing where to perturb. Where to perturb an input is a long-standing question in adversarial machine learning, and has been addressed with saliency maps (Papernot et al., 2016), feature-guided saliency distributions (Wicker et al., 2018), word-importance rankings (Jin et al., 2020), and random search over perturbation locations (Croce et al., 2022). Closest to our work, Lee et al. (2022) use Bayesian optimization to dynamically infer which positions of a text to substitute when attacking classifiers (see also Ru et al., 2020). We instead infer where to insert adversarial tokens in jailbreaks, using a conjugate model whose posteriors can be merged in closed form across prompts and models, and select allocations by Thompson sampling for linear bandits (Agrawal & Goyal, 2013). To the best of our knowledge, ours is the first work to treat the placement of adversarial tokens in jailbreak attacks as a problem of Bayesian inference.

3 PRELIMINARIES

Let $\mathcal{V}$ be a token vocabulary. A prompt $x=(z_1,...,z_L)\in\mathcal{V}^L$ is presented to an instruction-tuned model f wrapped by its chat template, $\tilde{x}:=H_{\text{pre}}\|x\|H_{\text{post}}$, with the control tokens $H_{\text{pre}},H_{\text{post}}$ held fixed. The language model is denoted as a function that sends an arbitrary length string in the vocabulary to the next token, $f:\mathcal{V}^\star\to\mathcal{V}$.

Threat Model. We assume we are given a target response $y^\star$ and assume that the adversary may make up to m insertions into the prompt (without modifying the chat template). The adversary is allowed to query the model and its gradients (white-box access), and we define the attack loss as the mean cross-entropy $\mathcal{L}(\tilde{x}'):=-\frac{1}{N^\star}\sum_i\log\mathbb{P}_f(z_i^\star\mid\tilde{x}'\|y^\star_{<i})$. Each of these attacks is local to the prompt x, so we denote the tuple of input, target response, and target model as a *task*: $\tau=(x,y^\star,f)$.

Slots and allocations. The adversary may decide *where* in the prompt to insert their m-many tokens. These choices are confined to the slot set $\mathcal{S}(x):=\{0,...,L\}$, of size $d(x):=L+1$ which indexes the gaps between prompt tokens. We let s denote the gap following z_s. The primary object we study is the *allocation*, $c\in\mathbb{Z}_{\geq 0}^{d(x)}$ which assigns c_s-many tokens to slot s; at budget m the feasible set is $\mathcal{C}_m(x):=\{c:\sum_s c_s=m\}$, with $|\mathcal{C}_m(x)|=\binom{m+d(x)-1}{d(x)-1}$. Let the tokens assigned to each slot be expressed as $\boldsymbol{z}^A:=\{\boldsymbol{z}_i\}_{i=1}^{d(x)}$ with $\boldsymbol{z}_s^A$ denoting the sequence of tokens to be inserted into the slot following z_s, we write $\tilde{x}(c,\boldsymbol{z}^A)$ for the wrapped result. Suffix GCG is the single point $c=me_L$, prefix attacks $c=me_0$, and SlotGCG one heuristically chosen c held fixed; we instead treat c as an inferential target.

Attack optimization. The ideal loss under an allocation, $\ell_\tau^\star(c):=\min_{\boldsymbol{z}^A\in\mathcal{V}^m}\mathcal{L}_\tau(\tilde{x}(c,\boldsymbol{z}^A))$, is unobservable as it is the solution of an extremely high dimensional, non-convex optimization problem. We therefore model the optimiser as a part of the adversary. Let $\mathsf{A}(c,\boldsymbol{z}_0^A,I)$ denote a stochastic token optimiser run for I steps from an initialisation $\boldsymbol{z}_0^A$. For a set of T-many test tasks, $\{\tau_i\}_{i=1}^T$, the optimizer takes the following parameters: a batch size B_t, number of optimization steps I_t. The total adversarial compute budget assumes atomic cost for each step and item in a batch, thus the T total optimizer cost is $\sum_t B_t I_t$.

Canonical positions. To compare positions across prompts and tokenizers, we follow the relative-position construction of SlotGCG (Jeong et al., 2026, App. K). For a collection of source tasks, let $L_{\max}$ be the largest tokenized prompt length and fix the shared slot set $\mathcal{S}^{\text{can}}:=\{0,...,L_{\max}\}$. On a prompt x of length L_x, canonical position j is realised at

$$\pi_x(j):=\text{round}\left(\frac{jL_x}{L_{\max}}\right),\qquad c_s^{\text{loc}}:=\sum_{j:\pi_x(j)=s}c_j. \tag{1}$$

Thus the endpoints map to the prefix and suffix, and counts are added when canonical positions coincide locally; their token blocks are concatenated in canonical order. The grid is held fixed during inference and reuse, with π_x computed using each task's tokenizer. Local heuristic scores such as VSS can be linearly interpolated onto this grid to initialise a canonical profile.

4 POSITIONAL INFERENCE

The number of potential allocations, $|\mathcal{C}_m(x)|$, is $\binom{m+d(x)-1}{d(x)-1}$. For a small, 100 token prompt where the adversary can insert $m=10$ adversarial tokens, this amounts to more than 10^{13} possible allocation vectors. While this rules out exhaustive search, understanding *where* to modify a prompt can lead to substantial gains in attack success rate. Throughout this section we will use $\mathcal{S}$ to denote local slots $\mathcal{S}(x)$ or canonical positions $\mathcal{S}^{\text{can}}$ interchangeably.

Existing heuristics such as SlotGCG or suffix/prefix GCG have a deterministic slot selection criteria. That is, they deterministically determine the $c \in \mathcal{C}_m(x)$ they will explore. By contrast, our framework treats the optimal allocation as an uncertain quantity to be learned. The sketch of our optimization process is simple: we first sample candidate allocations (discussed in §4.4), we then follow the same coordinate descent optimization of other methods to find approximately optimal token assignments for the sampled assignment, and finally, we observe the loss for each sampled allocation as a noisy observation of its true reward (discussed in §4.1) and we update the candidate sampling algorithm according to Bayes rule (discussed in §4.3). Of course, where we have substantial belief that existing heuristics are strong, we can encode them as prior distributions from which to sample initial allocations (discussed in §4.2). Below, we state the equations that underpin our attacks procedure and define its critical quantities and where they are discussed:

$$\textbf{Inference:}\quad \underbrace{p(\boldsymbol{w}^\tau \mid \mathcal{D}_n)}_{\textbf{posterior}\,(\S4.3)} \propto \underbrace{\prod_{i=1}^{n} p(r_i \mid c_i, \boldsymbol{w}^\tau)}_{\textbf{likelihood}\,(\S4.1)} \underbrace{p(\boldsymbol{w}^\tau)}_{\textbf{prior}\,(\S4.2)}$$

$$\textbf{Decision:}\quad \underbrace{a_{t,j} = \Psi_{m_t}(\widetilde{\boldsymbol{w}}_{t,j}; C^{(t-1)}),\qquad \widetilde{\boldsymbol{w}}_{t,j} \sim p(\boldsymbol{w}^\tau \mid \mathcal{D}_{n_{t-1}})}_{\textbf{allocation policy}\,(\S4.4)}. \tag{2}$$

Above, $\boldsymbol{w}^\tau \in \mathbb{R}^{L_x}$ is the task (τ) specific latent vector with the sth element, $\boldsymbol{w}_s^\tau$ modelling our belief about the utility of inserting tokens in the sth position; n is the number of observations we have made; c_i is the ith-observed allocation; r_i is the noisy reward associated with the ith allocation; $\mathcal{D}_{n_{t-1}}$ is the collection of all observations made up to the tth iteration of our algorithm with Ψ being the allocation policy (§4.4).

4.1 LIKELIHOOD

Given a sampled candidate allocation c, we run A for I steps from initialisation $\boldsymbol{z}_0^A$ which returns a token assignment that in turn yields a loss $\widehat{\ell}$. As discussed, we assume that this observed loss encodes information about the utility of the candidate allocation c. If the loss is far below the starting loss, then c has high utility, formally utility is defined as: $r := \mathcal{L}_{\tau,0} - \widehat{\ell}$, the difference between the loss on the unperturbed input, $\mathcal{L}_{\tau,0}$, and the optimised loss, $\widehat{\ell}$. Larger utility therefore indicates an allocation under which the optimiser achieved a greater reduction in attack loss.

In our implementation, we collect these observations after every optimiser step. Where we have sampled the same candidate vector multiple times, we collect their optimized token assignments into a set $\mathcal{G}_t(c)$. Writing $\widehat{\ell}_{t,j}$ for candidate j's best loss observed by that step, the group contributes $r_t(c) := \mathcal{L}_{\tau,0} - \min_{j \in \mathcal{G}_t(c)} \widehat{\ell}_{t,j}$. Thus every distinct allocation supplies one observation at each step. We highlight that the utility being measured depends on the number of times each distinct allocation was sampled, the optimisation procedure, the initialisation, and others. Because of the number of complicating factors and the improbability of solving the optimization exactly, we model the loss observations as uncertain.

A Linear Likelihood Model. Probabilistically modelling our likelihood, $p(r_i \mid c_i, \boldsymbol{w}^\tau)$, requires us to introduce assumptions about the relationship between an allocation vector c and the real-valued reward

observation. We begin by using the simplest possible model, a linear model between the transformed allocation $\boldsymbol{\phi}(c_i)$ and the observed reward r_i:

$$p(r_i \mid c_i, \boldsymbol{w}^\tau) = \mathcal{N}\big(r_i; \boldsymbol{\phi}(c_i)^\top \boldsymbol{w}^\tau, \sigma_\tau^2\big), \qquad \boldsymbol{\phi}(c) := \big(g(c_s)\big)_{s\in\mathcal{S}}, \qquad g(q) := \log(1+q). \tag{3}$$

The likelihood takes a task-dependent noise scale σ_τ^2 (discussed further in §4.2), coefficients $\boldsymbol{w}^\tau$, the allocation c_i, and a count transformation g shared across positions. We take $g(q) = \log(1+q)$ as this encodes diminishing utility where $w_{\tau,s} > 0$ and diminishing penalty where $w_{\tau,s} < 0$, and permits the exact allocation rule in §4.4. The function g acts on token counts after the positional profile has been initialised or combined with previous beliefs.

This likelihood makes the simplifying assumption that contributions from different positions are *additive*; modelling interactions between positions would require a richer feature map or likelihood. We also approximate the observation errors as independent Gaussian noise. Of course, a natural extension of this overly-simplistic model would be a likelihood that uses information about the optimization process (number of steps, number of times c_i was observed, type of optimizer) which we leave to future works.

4.2 Prior

One primary advantage of a principled probabilistic treatment of positional inference is the ability to express empirical and heuristic knowledge as a prior over *where* we should place our tokens. As we have selected a Gaussian likelihood model, we choose a Gaussian prior to maintain conjugacy (discussed in §4.3). Intuitively, the prior allows us to express where we believe the most effective slots are for a model before any observations are made. We initialise the prior profile from a nonzero, task-dependent vector $\theta_\tau \in \mathbb{R}^d$. Recalling that m is number of tokens to be inserted and allowing $\beta, \gamma > 0$ to be parameters controlling the shape of uncertainty, we then set our prior parameters to be:

$$\boldsymbol{w}^\tau \sim \mathcal{N}\left(\underbrace{\alpha_\tau \theta_\tau}_{\boldsymbol{\mu}_0}, \underbrace{\left(\frac{\beta\alpha_\tau}{d}\right)^2 I_d}_{\Sigma_0}\right), \qquad \alpha_\tau := \frac{\mathcal{L}_{\tau,0}}{g(m)\|\theta_\tau\|_1}, \qquad \sigma_\tau := \gamma \mathcal{L}_{\tau,0}. \tag{4}$$

The value of α_τ is a normalisation quantity. This is critically important as the profile is used as a predictive quantity in our likelihood (§4.1) and an allocative value in our decision framework (§4.4), we use the unperturbed loss to express its coefficients on the scale of the utilities we observe.

Encoding Heuristics as Priors. Setting $\theta_{\tau,s} = 1/d$ for every position s gives no initial preference for where to insert tokens (i.e., the uniform prior). Giving the suffix a larger prior coefficient favours suffix allocations. For more complex heuristics, using the Vulnerable Slot Score (VSS) to seed θ_τ allows us to use its attention-based observations to seed our probabilistic treatment (Jeong et al., 2026). These initial beliefs are then updated by the observed attack losses. Previously inferred profiles can also supply the prior, in which case we retain both their mean and covariance as described in §4.5.

4.3 Posterior

A considerable benefit of choosing a simple probabilistic model is that we have a closed-form posterior. After each optimiser step, we collect one group-best utility per distinct allocation. Collecting allocations into rows of a design matrix $\Phi_{i,:} = \boldsymbol{\phi}(c_i)^\top$ and their corresponding observations into a vector $\boldsymbol{r}$. Where we take $\mathcal{D}^+$ to denote all collected data, Gaussian conjugacy gives:

$$\begin{aligned} &\boldsymbol{w}^\tau \mid \mathcal{D}^+ \sim \mathcal{N}(\boldsymbol{\mu}^+, \Sigma^+), \\ &\Sigma^+ = \left(\Sigma^{-1} + \sigma_\tau^{-2}\Phi^\top\Phi\right)^{-1}, \qquad \boldsymbol{\mu}^+ = \boldsymbol{\mu} + \sigma_\tau^{-2}\Sigma^+\Phi^\top\big(\boldsymbol{r} - \Phi\boldsymbol{\mu}\big). \end{aligned} \tag{5}$$

The residual $\boldsymbol{r} - \Phi\boldsymbol{\mu}$ measures how the observed utilities differ from those predicted by our current profile. The update uses this difference to revise the coefficient estimates and reduces uncertainty in the directions

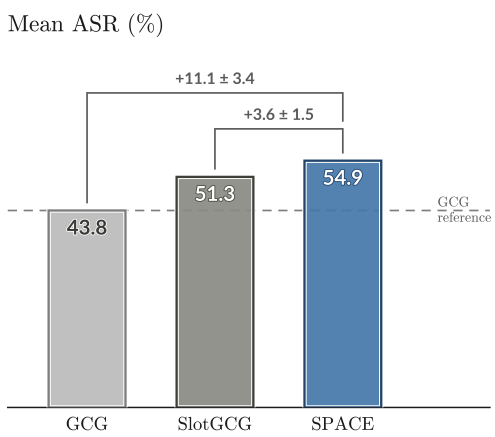

Figure 1: Mean ASR over the models of Table 1. SPACE achieves SOTA results on 8/12 models.

Model	GCG	SlotGCG	SPACE
Gemma-4-12B-it†	6.0	8.0	**20.0**
Gemma-4-26B-A4B-it†	8.0	10.0	**18.0**
Phi-4-Mini-Instruct	80.0	70.0	**80.0**
Llama-3.2-1B-Instruct‡	51.0	52.0	**60.0**
Nemotron-3.5-30B-A3B†	32.0	54.0	**60.0**
Qwen3-8B‡	64.0	61.0	**66.0**
Phi-4-14B	44.0	70.0	**72.0**
Qwen3-0.6B‡	48.0	51.0	**52.0**
Llama-2-7B-Chat‡	22.0	**46.0**	45.0
Llama-3.1-8B-Instruct‡	45.0	**68.0**	66.0
Mistral-7B-Instruct-v0.3‡	84.0	**86.0**	84.0
Qwen3-32B	**42.0**	40.0	36.0

Table 1: Final-state attack success rate (%) per model. ‡Mean over random seeds; the remaining 5 report seed 0 only due to compute constraints. †Runs for 250 rather than 500 steps.

measured by the evaluated allocations. Importantly, *all* distinct allocation groups at *every* optimiser step inform the posterior, so evaluations/observations that would usually be discarded or ignored in GCG variants are converted into evidence that supports our belief about where to attack a model's input.

4.4 Allocation policy

Once a posterior belief has been computed, we need a coherent framework for selecting allocations from our posterior. We provide two strategies: the first allocates according to the posterior mean, while the second samples profiles from the uncertain posterior.

Allocation under a given profile. We denote Ψ_b as a function that maps a positional profile to an allocation increment. We can think of this as an iterative process that, given committed counts C, tells us where to assign the next b tokens:

$$\Psi_b(\boldsymbol{w};C) \in \underset{a\in\mathcal{C}_b}{\operatorname{argmax}} \boldsymbol{\phi}(C+a)^\top \boldsymbol{w}, \qquad \Delta_s^C(j) := \boldsymbol{w}_s\big[g(C_s+j)-g(C_s+j-1)\big]. \tag{6}$$

Here $\Delta_s^C(j)$ is the marginal contribution of the j-th additional token at position s. Using $\boldsymbol{w}=\boldsymbol{\mu}$ maximises the posterior expected utility of the allocation. Although the feasible set is combinatorial, the concavity of g makes this allocation problem exactly solvable.

Proposition 1 (Exact incremental allocation)**.** *If* $\max_s \boldsymbol{w}_s \geq 0$*, initialise* $a=\mathbf{0}$ *and assign each successive token to a slot maximising* $\Delta_s^C(a_s+1)$ *over* s *with* $\boldsymbol{w}_s \geq 0$*. This yields an optimal increment for equation 6. If every coefficient is negative, then an optimal increment is be*$_{s^\star}$*, where* $s^\star \in \operatorname{argmax}_s \boldsymbol{w}_s\big[g(C_s+b)-g(C_s)\big]$.

Where $\boldsymbol{w}_s \geq 0$, the marginals are non-increasing, so greedy selection of the largest available contribution is globally optimal for the stated allocation objective. If every coefficient is negative, the objective is convex and the constraint is best satisfied by concentrating damage in one slot rather than spreading it. Proof in App. D.1.

Thompson sampling and commitment. A Bayes-optimal finite-horizon policy would integrate over future observations, posterior updates and observation-dependent actions at every round; this is intractable over $\mathcal{C}_m$. Thus, we use Thompson sampling. To get the jth sample in iteration t, we sample from our posterior utility vector, $\widetilde{\boldsymbol{w}}_{t,j} \sim p(\boldsymbol{w}^\tau \,|\, \mathcal{D}_{n_{t-1}})$, and then perform allocation using $\widetilde{\boldsymbol{w}}_{t,j}$. We highlight that not all m-many tokens need to be sampled each iteration, one is free to schedule the number of tokens done per iteration flexibly. The reason that this does not effect our algorithm is precisely because of the additive assumption of our likelihood and the greedy property of Proposition 1.

4.5 Merging positional priors

One major benefit of Gaussian conjugate inference is that posterior beliefs can be merged in closed form. Given posteriors on tasks $\{\tau_k\}_{k=1}^N$, denoted $p_{\tau_k}(\mathbf{w})$ are expressed in common canonical coordinates and

on a common scale, their product defines a merged Gaussian belief: $p_{\text{merge}}(\mathbf{w}) \propto \prod_{k=1}^{K} p_{\tau_k}(\mathbf{w})$. When each source's original prior p_{0,τ_k} is known, we can also avoid double-counting prior information. Dividing each posterior by its prior isolates the evidence learned from that source, which we combine with a single population prior p_{base}: $p_{\text{merge}}(\mathbf{w}) \propto p_{\text{base}}(\mathbf{w}) \prod_{k=1}^{K} \frac{p_{\tau_k}(\mathbf{w})}{p_{0,\tau_k}(\mathbf{w})}$.

Finally, nonnegative weights ω_k control how strongly each source contributes: $p_{\text{merge}}(\mathbf{w}) \propto p_{\text{base}}(\mathbf{w}) \prod_{k=1}^{K} \left[\frac{p_k(\mathbf{w})}{p_{0,k}(\mathbf{w})}\right]^{\omega_k}$. We assign larger weights to sources achieving greater fractional loss reduction. The merged belief remains Gaussian, allowing its mean and covariance to be rescaled and reused as the prior for a new task.

Transferable and Universal Attacks. Posterior merging serves as yet another critically important feature of our framework as it allows us to not just use observations to inform attacks about a given task τ (input, output, and model), but allows to merge the attack information across input (universal) and model (transferable). The ability to re-use the expensive computations of attacking language models to inform smarter, more adaptive attacks allows us to improve both attack success rates and efficiency by more than 10%.

5 Experiments

We evaluate our attacks on twelve instruction-tuned LLMs ranging two orders of magnitude in size, three also evaluated by SLOTGCG (Jeong et al., 2026) and nine newer additions. We validate our attacks experimentally on 50 harmful behaviour–target pairs from AdvBench as per the SLOTGCG paper (Jeong et al., 2026). For consistency, we align our attack settings to our baselines GCG and SLOTGCG: $m = 20$ adversarial tokens, a top-k pool of 256 and candidate batch of 512 (Zou et al., 2023; Jeong et al., 2026). All attacks receive exactly the same number of iterations. Similarly, we compute VSS using softmax temperature 8 (Jeong et al., 2026). For SPACE, we set $\gamma = 0.1$ and $\beta = 1$. ASR is the fraction of optimised prompts eliciting a valid jailbreak, judged for semantic compliance by GPT-4.1. Full setup details are found in Appendix A.

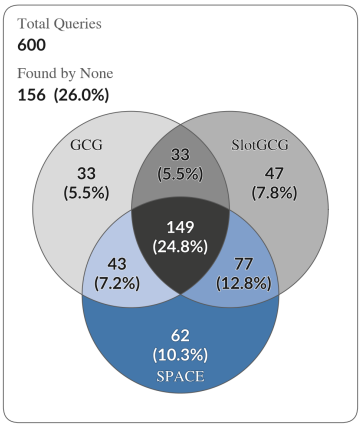

Figure 2: We study a total of 600 unique tasks. While methods individually attacked 43.8%, 51.3%, and 54.9% of these individually, together they were successful on 76% of tasks. Of the covered 76%, SPACE covers the most unique tasks (10.3%).

SPACE allocation beats heuristics. We begin by comparing SPACE against baseline attacks GCG and SLOTGCG, which represent typical suffix-based and attention-derived position selection heuristics respectively. We instantiate SPACE with an unbiased uniform prior, relying solely on adaptive positional inference. Fig. 1 shows substantial gains in mean ASR for SPACE, leading GCG (+11.1%) and SLOTGCG (+3.6%). Notably, Table 1 shows that SPACE tends to outperform on newer model families like Gemma-4 and Nemotron-3.5, while remaining close to the stronger baseline for older models like Llama-2 and Mistral-v0.3. This is suggestive of SPACE's adaptive learning strategy maintaining attack performance, in spite of growing robustness by newer models.

Extending from our above analysis, we plot Fig. 2 which reflects the overlap and disjoint subsets of attack outcomes on each task. Notably, SPACE uniquely covers 62 tasks previously unachievable by GCG and SLOTGCG. SPACE consequently contributes to a substantial expansion in coverage from 63.7% to 74.0%, reflecting greater discovery of model vulnerabilities.

SPACE with informed priors matches heuristics. Priors allow us to mimic – and potentially improve – heuristics. In this subsection, we compare SPACE, initialised with a suffix and VSS prior, against GCG and SLOTGCG respectively. Table 2 and Fig. 3 report marginal ASR gains for SPACE over respective baseline: +0.3% for the suffix arm and +0.5% for the VSS arm. We propose a few hypotheses for the poor ASR gains. For instance, the prior may have been initialised too strong such that it dominates early token allocation decisions, constraining exploration of other positions that may yield better loss reduction. Also, SPACE being a multi-round attack, may see tokens inserted later undergoing fewer optimisation steps, to the detriment of attack performance from poorly optimised token values. Fortunately, the plots suggest that SPACE performs no worse than their reference heuristic attack.

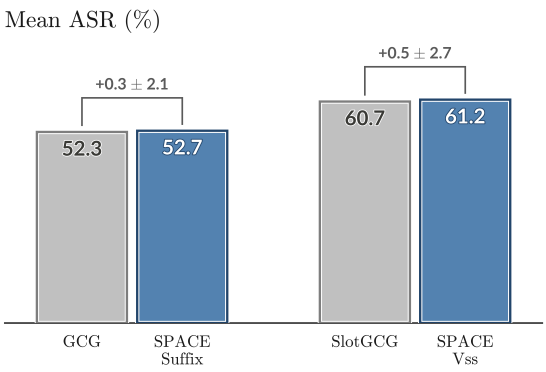

Figure 3: Mean ASR over the six models of Table 2, with the positional heuristic and matched prior for SPACE. Brackets give the *paired* mean margin ± 1 s.e.m. across models.

	Suffix heuristic		VSS heuristic	
Model	GCG	SPACE	SlotGCG	SPACE
Llama-2-7B	22.0	**30.0**	46.0	**47.0**
Llama-3.1-8B	**45.0**	44.0	**68.0**	58.0
Llama-3.2-1B	51.0	**56.0**	52.0	**59.0**
Mistral-7B	**84.0**	82.0	**86.0**	81.0
Qwen3-0.6B	**48.0**	46.0	51.0	**56.0**
Qwen3-8B	**64.0**	58.0	61.0	**66.0**
Mean	52.3	**52.7**	60.7	**61.2**
SPACE wins	—	2/6	—	4/6

Table 2: ASR (%) at 500 optimiser steps and 512 candidates per step, over the six models reported at two seeds. The version of SPACE uses a suffix- or VSS-weighted prior. **Bold** marks the higher of the two.

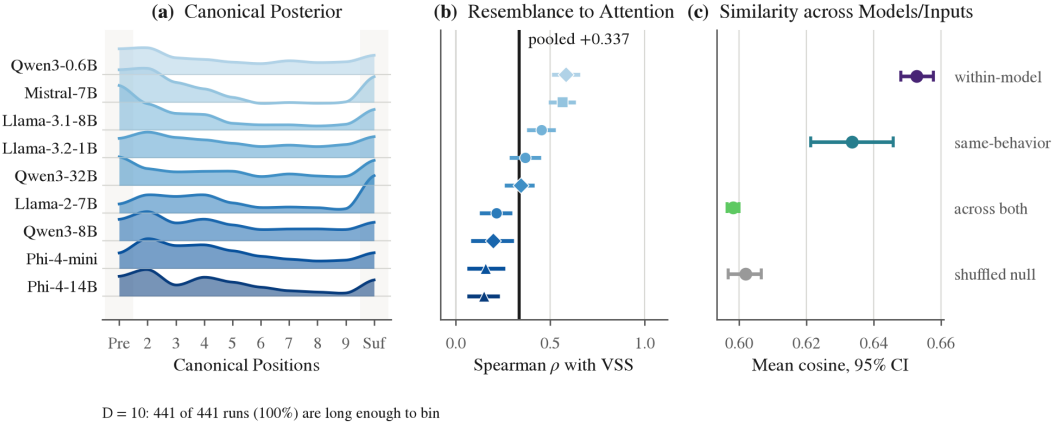

Figure 4: **Left:** The learned posteriors of all tasks on 9 models using 10 canonical coordinates. **Center:** The correlation of inferred posteriors with the VSS heuristic. **Right:** The cosine similarity of posteriors across models, tasks, and both compared with the cosine similarity with random noise (shuffled null).

From the results reported above, we note that SPACE performs more successfully than state-of-the-art heuristics. In this subsection, we inspect our learned posterior to understand what might explain SPACE's observed advantage.

A U-shaped posterior emerges. We plot Fig. 4a by pooling and projecting the optimised SPACE posterior of each behaviour down to 10 canonical positions. The resulting mean posterior effectively represents the universally preferred positions across behaviours for each model. We observe a distinct bimodal posterior across all models, particularly at the prefix and suffix positions. We also observe preferences for infix positions, albeit those closer to the prefix or suffix. This finding corroborates multiple seemingly conflicting perspectives in attack literature on optimal adversarial positioning (Ben-Tov et al., 2026a; Eddoubi et al., 2026b; Jeong et al., 2026), with our results implying that a combination attack comprising prefix, suffix and certain infix positions more effective than standalone positional heuristics.

SPACE posteriors diverge from heuristic. Comparing SPACE posterior profile against VSS, we plot Fig. 4b by similarly pooling and projecting VSS attention profiles across behaviours for each model. We then compute the mean Spearman correlation between the inferred posterior and each model's binned VSS attention profile. We report weak resemblance with attention, $\rho = 0.337$, ranging 0.150 on Phi-4-14B to 0.583 on Qwen3-0.6, almost a fourfold spread. Hence, we deduce that positional vulnerability is loosely related to attention and model-specific. This challenges the reliance of standalone heuristics being a representative oracle and supports our adaptive approach for stronger attacks.

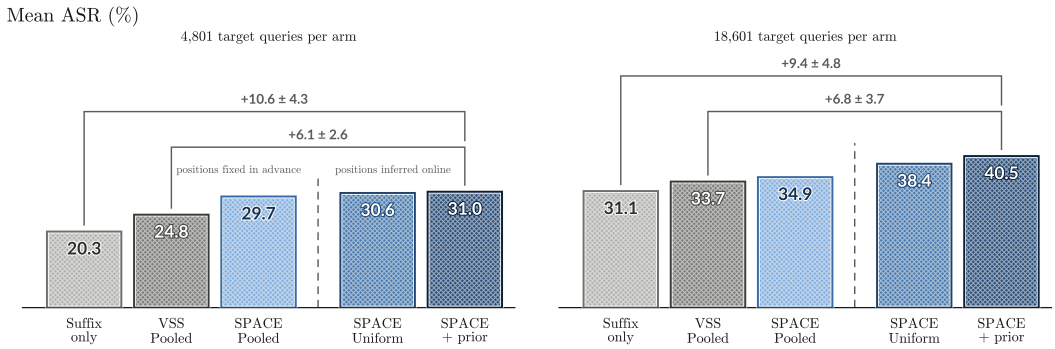

Figure 5: **Black-box posterior transfer:** We hold out one of our 12 models and merge the posteriors (for SPACE) and VSS (from SlotGCG) of the remaining 11 to inform slot selection. We plot the ASR of running BEAST using the specified allocations and observe a substantial ASR increase for posterior-informed attacks. *Left:* Attacks using 4801 target model queries. *Right:* Attacks using 18601 target model queries.

Posterior similarity across models and behaviours. We further analyze our inferred posteriors by computing pairwise cosine similarity between canonical posteriors between behaviours for the same model; and between models for the same behaviour. We then take the mean to obtain a scalar metric. Fig. 4c shows that posterior pairs from the same model are more similar (0.653) than pairs from different models (0.634 for the same behaviour, else 0.598). Across both model and behaviour, pairs are indistinguishable from a shuffled model null (0.602). This implies that our learned posterior tracks the model, not the input behaviour and suggestive of the model-specificity of positional vulnerability.

The remarkable benefits of inference in transferability. We now consider the *black-box* setting in which we cannot observe model internals (gradients or architecture) but can observe log probabilities using a fixed number of queries. We consider a black-box attack on each model in our 12 model pool using a leave-one-out approach. We study three different SPACE variants: *SPACE Pooled* which simply uses the MAP allocation vector of the pooled posteriors and does not perform positional inference during attack; *SPACE Uniform* which uses a uniform prior and performs positional inference during the attack; *SPACE + Prior* which uses the aforementioned pooled prior but also performs positional inference during the attack. For SPACE Pooled and Prior, we attack the left-out model using the merged posteriors of the 11 observed models. Similarly, for VSS we merge the VSS vector of the 11 observed models according to the procedure defined in (Jeong et al., 2026). In Figure 5 we run the BEAST algorithm for 4801 and 18601 steps respectively. We observe that *SPACE + Prior* achieves nearly an 10% ASR increase over both suffix baselines and more than 6% ASR increase over the VSS heuristic. Perhaps more impressive is the fact that at only 25% of the queries, *SPACE + Prior* is able to achieve the same ASR as the suffix-only heurisitic.

Limitations Table 4 reports a mean wall-clock time increase by 33% to run SPACE over our baseline attacks. While all attacks are strictly matched in terms of optimization steps postulate the primary overhead contributor to be SPACE maintaining multiple candidate allocations, compared to one committed allocation for GCG or SLOTGCG. As each candidate requires forward and backward passes to construct its own per-token top-k sampling pool, the overhead cost scales approximately linearly with the number of candidates. We do stress, however, that for our BEAST experiments wall clock is identical.

6 CONCLUSION

In this work, we cast the selection of adversarial token positions as Bayesian inference, with the optimizer's intermediate losses as observations and existing positional heuristics as priors. The resulting attack, SPACE, adapts where it places adversarial tokens as the attack proceeds and achieves higher mean attack success than suffix-based and attention-based placement across twelve LLMs. Its posteriors can also be merged across models to improve black-box attacks on held-out models. Beyond attack success, the inferred posteriors allow us to test the assumptions behind existing heuristics. We find that positional vulnerability is consistently concentrated at both ends of the prompt and only loosely related to attention, which highlights adaptivity of our attack.

7 AI USE STATEMENT

In this work, we used generative AI tools for helping to develop theoretical models or conceptual frameworks, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods and interpret results. We have not used generative AI tools for other tasks with required disclosure, that which are applicable to this work. Additionally, we used generative AI tools for helping to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, brainstorming, edit a research paper to improve readability and suggest a structure for a research paper. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey, LLM-generated code was verified and tested for correctness by 2 authors and LLM-generated artifacts and figures were verified for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Marah I Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio CT Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report. Technical Report MSR-TR-2024-57, Microsoft, December 2024. URL https://www.microsoft.com/en-us/research/publication/phi-4-technical-report/.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 127–135. PMLR, 2013.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hXA8wqRdyV.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4322.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283. PMLR, 2018.

Matan Ben-Tov, Mor Geva, and Mahmood Sharif. Universal jailbreak suffixes are strong attention hijackers. *Transactions of the Association for Computational Linguistics*, 14:1028–1050, 06 2026a. ISSN 2307-387X. doi: 10.1162/TACL.a.695. URL https://doi.org/10.1162/TACL.a.695.

Matan Ben-Tov, Mor Geva, and Mahmood Sharif. Universal jailbreak suffixes are strong attention hijackers. *Transactions of the Association for Computational Linguistics*, 14:1028–1050, 2026b. doi: 10.1162/TACL.a.695.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017. doi: 10.1145/3128572.3140444.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, and Matthias Hein. Sparse-RS: A versatile framework for query-efficient sparse black-box adversarial attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6437–6445, 2022. doi: 10.1609/aaai.v36i6.20595.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany M. Biron, Binh Tang, Bobbie Chern, Char lotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, E I Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cu-curell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshi tiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bash-lykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, R. S. Mughil Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Sheng-Ye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin R. Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smoth-ers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco (Paco) Guzmán, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry As-pegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Francis Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li,

Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthias Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Re-strepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Ro han Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, S. Yu. Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Kumar Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria O Ajayi, Victoria Montanez, Vijai Mohan, Vinay Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

Hicham Eddoubi, Umar Faruk Abdullahi, and Fadi Hassan. Beyond suffixes: Token position in GCG adversarial attacks on large language models. In *I Can't Believe It's Not Better Workshop at the International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2602.03265.

Hicham Eddoubi, Umar Faruk Abdullahi, and Fadi Hassan. Beyond suffixes: Token position in gcg adversarial attacks on large language models, 2026b. URL https://arxiv.org/abs/2602.03265.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

Seungwon Jeong, Jiwoo Jeong, Hyeonjin Kim, Yunseok Lee, and Woojin Lee. SlotGCG: Exploiting the positional vulnerability in LLMs for jailbreak attacks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Fn2rSOnpNf.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e9yfCY7Q3U.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, 2020. doi: 10.1609/aaai.v34i05.6311.

Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via Bayesian optimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12478–12497. PMLR, 2022.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Jwpw4qKkb.

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. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *Advances in Neural Information Processing Systems*, volume 37, pp. 61065–61105, 2024. doi: 10.52202/079017-1952.

Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025. URL https://arxiv.org/abs/2503.01743.

NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwalla, Andrew Tao, Anjulie Agrusa, Ankur Verma, Ann Guan, Anubhav Mandarwal, Arham Mehta, Ashwath Aithal, Ashwin Poojary, Asif Ahamed, Asit Mishra, Asma Kuriparambil Thekkumpate, Ayush Dattagupta, Banghua Zhu, Bardiya Sadeghi, Barnaby Simkin, Ben Lanir, Benedikt Schifferer, Besmira Nushi, Bilal Kartal, Bita Darvish Rouhani, Boris Ginsburg, Brandon Norick, Brandon Soubasis, Branislav Kisacanin, Brian Yu, Bryan Catanzaro, Carlo del Mundo, Chantal Hwang, Charles Wang, Cheng-Ping Hsieh, Chenghao Zhang, Chenhan Yu, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christopher Parisien, Collin Neale, Cyril Meurillon, Damon Mosk-Aoyama, Dan Su, Dane Corneil, Daniel Afrimi, Daniel Lo, Daniel Rohrer, Daniel Serebrenik, Daria Gitman, Daria Levy, Darko Stosic, David Mosallanezhad, Deepak Narayanan, Dhruv Nathawani, Dima Rekesh, Dina Yared, Divyanshu Kakwani, Dong Ahn, Duncan Riach, Dusan Stosic, Edgar Minasyan, Edward Lin, Eileen Long, Eileen Peters Long, Elad Segal, Elena Lantz, Ellie Evans, Elliott Ning, Eric Chung, Eric Harper, Eric Tramel, Erick Galinkin, Erik Pounds, Evan Briones, Evelina Bakhturina, Evgeny Tsykunov, Faisal Ladhak, Fay Wang, Fei Jia, Felipe Soares, Feng Chen, Ferenc Galko, Frank Sun, Frankie Siino, Gal Hubara Agam, Ganesh Ajjanagadde, Gantavya Bhatt, Gargi Prasad, George Armstrong, Gerald Shen, Gorkem Batmaz, Grigor Nalbandyan, Haifeng Qian, Harsh Sharma, Hayley Ross, Helen Ngo, Herbert Hum, Herman Sahota, Hexin Wang, Himanshu Soni, Hiren Upadhyay, Huizi Mao, Huy C Nguyen, Huy Q Nguyen, Iain Cunningham, Ido Galil, Ido Shahaf, Igor Gitman, Ilya Loshchilov, Itamar Schen, Itay Levy, Ivan Moshkov, Izik Golan, Izzy Putterman, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jatin Mitra, Jeffrey Glick, Jenny Chen, Jesse Oliver, Jian Zhang, Jiaqi Zeng, Jie Lou, Jimmy Zhang, Jinhang Choi, Jining Huang, Joey Conway, Joey Guman, John Kamalu, Johnny Greco, Jonathan Cohen, Joseph Jennings, Joyjit Daw, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kai Xu, Kan Zhu, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kevin Shih, Kezhi Kong, Khushi Bhardwaj, Kirthi Shankar, Krishna C. Puvvada, Krzysztof Pawelec, Kumar Anik,

Lawrence McAfee, Laya Sleiman, Leon Derczynski, Li Ding, Lizzie Wei, Lucas Liebenwein, Luis Vega, Maanu Grover, Maarten Van Segbroeck, Maer Rodrigues de Melo, Mahdi Nazemi, Makesh Narsimhan Sreedhar, Manoj Kilaru, Maor Ashkenazi, Marc Romeijn, Marcin Chochowski, Mark Cai, Markus Kliegl, Maryam Moosaei, Matt Kulka, Matvei Novikov, Mehrzad Samadi, Melissa Corpuz, Mengru Wang, Meredith Price, Michael Andersch, Michael Boone, Michael Evans, Miguel Martinez, Mikail Khona, Mike Chrzanowski, Minseok Lee, Mohammad Dabbah, Mohammad Shoeybi, Mostofa Patwary, Nabin Mulepati, Najeeb Nabwani, Natalie Hereth, Nave Assaf, Negar Habibi, Neta Zmora, Netanel Haber, Nicola Sessions, Nidhi Bhatia, Nikhil Jukar, Nikki Pope, Nikolai Ludwig, Nima Tajbakhsh, Nir Ailon, Nirmal Juluru, Nishant Sharma, Oleksii Hrinchuk, Oleksii Kuchaiev, Olivier Delalleau, Oluwatobi Olabiyi, Omer Ullman Argov, Omri Puny, Oren Tropp, Ouye Xie, Parth Chadha, Pasha Shamis, Paul Gibbons, Pavlo Molchanov, Pawel Morkisz, Peter Dykas, Peter Jin, Pinky Xu, Piotr Januszewski, Pranav Prashant Thombre, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Qing Miao, Qiyu Wan, Rabeeh Karimi Mahabadi, Rachit Garg, Ran El-Yaniv, Ran Zilberstein, Rasoul Shafipour, Rich Harang, Rick Izzo, Rima Shahbazyan, Rishabh Garg, Ritika Borkar, Ritu Gala, Riyad Islam, Robert Hesse, Roger Waleffe, Rohit Watve, Roi Koren, Ruoxi Zhang, Russell Hewett, Russell J. Hewett, Ryan Prenger, Ryan Timbrook, Sadegh Mahdavi, Sahil Modi, Samuel Kriman, Sangkug Lim, Sanjay Kariyappa, Sanjeev Satheesh, Saori Kaji, Satish Pasumarthi, Saurav Muralidharan, Sean Narentharen, Sean Narenthiran, Seonmyeong Bak, Sergey Kashirsky, Seth Poulos, Shahar Mor, Shanmugam Ramasamy, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shiqing Fan, Shreya Gopal, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Simeng Sun, Smita Ithape, Somshubra Majumdar, Soumye Singhal, Stas Sergienko, Stefania Alborghetti, Stephen Ge, Sugam Dipak Devare, Sumeet Kumar Barua, Suseella Panguluri, Suyog Gupta, Sweta Priyadarshi, Syeda Nahida Akter, Tan Bui, Teodor-Dumitru Ene, Terry Kong, Thanh Do, Tijmen Blankevoort, Tim Moon, Tom Balough, Tomer Asida, Tomer Bar Natan, Tomer Ronen, Tugrul Konuk, Twinkle Vashishth, Udi Karpas, Ushnish De, Vahid Noorozi, Vahid Noroozi, Venkat Srinivasan, Venmugil Elango, Victor Cui, Vijay Korthikanti, Vinay Rao, Vitaly Kurin, Vitaly Lavrukhin, Vladimir Anisimov, Wanli Jiang, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenfei Zhou, Will Jennings, William Zhang, Wojciech Prazuch, Xiaowei Ren, Yashaswi Karnati, Yejin Choi, Yev Meyer, Yi-Fu Wu, Yian Zhang, Yigong Qin, Ying Lin, Yonatan Geifman, Yonggan Fu, Yoshi Subara, Yoshi Suhara, Yubo Gao, Zach Moshe, Zhen Dong, Zhongbo Zhu, Zihan Liu, Zijia Chen, and Zijie Yan. Nvidia nemotron 3: Efficient and open intelligence, 2025. URL https://arxiv.org/abs/2512.20856.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.225.

Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. BayesOpt adversarial attack. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkem-lrtvH.

Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018. doi: 10.1561/2200000070.

Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Chegini, and Soheil Feizi. Fast adversarial attacks on language models in one GPU minute. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42976–42998. PMLR, 2024.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku

Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933. doi: 10.1093/biomet/25.3-4.285.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu,

Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 7 2023. URL https://www.semanticscholar.org/paper/104b0bb1da562d53cbda87aec79ef6a2827d191a.

Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1633–1645, 2020.

Zijun Wang, Haoqin Tu, Jieru Mei, Bingchen Zhao, Yisen Wang, and Cihang Xie. AttnGCG: Enhancing jailbreaking attacks on LLMs with attention manipulation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=prVLANCshF.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, pp. 80079–80110, 2023.

Matthew Wicker, Xiaowei Huang, and Marta Kwiatkowska. Feature-guided black-box safety testing of deep neural networks. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 10805 of *Lecture Notes in Computer Science*, pp. 408–426. Springer, 2018.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *ArXiv*, abs/2407.04295, 2024. URL https://api.semanticscholar.org/CorpusID:271038633.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Further Experimental Setup Details

A.1 Details on Target Models

We evaluate SPACE, GCG and SlotGCG on twelve instruction-tuned models from six different model families. They comprise:

- Meta's Llama-2 (Touvron et al., 2023) and Llama-3 families (Dubey et al., 2024): Llama-3.2-1B-Instruct, Llama-2-7B-Chat and Llama-3.1-8B-Instruct
- Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)
- Microsoft's Phi-4 family: Phi-4-Mini-Instruct (Microsoft et al., 2025) and Phi-4 (Abdin et al., 2024)
- Alibaba's Qwen3 family: Qwen3-0.6B, Qwen3-8B and Qwen3-32B (Yang et al., 2025)
- NVIDIA's Nemotron-3.5-Lightning-30B-A3B (NVIDIA et al., 2025)
- Google's Gemma-4 family: Gemma-4-12B-it and Gemma-4-26B-A4B-it (Team et al., 2026)

We configure model generation to be deterministic by setting temperature to zero. For each target model, we construct their model-specific chat template using `tokenizer.apply_chat_template`. We disable thinking mode for the Qwen3 and Nemotron-3.5-Lightning models as explicit thinking controls were included in their chat template. For the remaining models, we used their default chat template settings. Table 3 summarises the chat templates used by each target model.

Table 3: Overview of chat templates for evaluated target models.

Model	Prompt format
Llama-2-7B-Chat	`<s>[INST] {instruction} [/INST]`
Llama-3.1-8B-Instruct	`<|begin_of_text|><|start_header_id|>system<|end_header_id|>\n\n Cutting Knowledge Date: December 2023\nToday Date: 26 Jul 2024\n \n<|eot_id|><|start_header_id|>user<|end_header_id|>\n\n {instruction}<|eot_id|><|start_header_id|>assistant <|end_header_id|>\n\n`
Llama-3.2-1B-Instruct	`<|begin_of_text|><|start_header_id|>system<|end_header_id|>\n\n Cutting Knowledge Date: December 2023\n Today Date: {rendering date}\n\n<|eot_id|><|start_header_id|> user<|end_header_id|>\n\n{instruction}<|eot_id|> <|start_header_id|>assistant<|end_header_id|>\n\n`
Mistral-7B-Instruct-v0.3	`<s>[INST] {instruction}[/INST]`
Qwen3-0.6B	`<|im_start|>user\n{instruction}<|im_end|>\n<|im_start|>assistant \n<think>\n\n</think>\n\n`
Qwen3-8B	`<|im_start|>user\n{instruction}<|im_end|>\n<|im_start|>assistant \n<think>\n\n</think>\n\n`
Qwen3-32B	`<|im_start|>user\n{instruction}<|im_end|>\n<|im_start|>assistant \n<think>\n\n</think>\n\n`
Phi-4-Mini-Instruct	`<|user|>{instruction}<|end|><|assistant|>`
Phi-4-14B	`<|im_start|>user<|im_sep|>{instruction}<|im_end|><|im_start|> assistant<|im_sep|>`
Nemotron-3.5-30B-A3B	`<|im_start|>system\n<|im_end|>\n<|im_start|>user\n{instruction} <|im_end|>\n<|im_start|>assistant\n<think></think>`
Gemma-4-12B-IT	`<bos><|turn>user\n{instruction}<turn|>\n<|turn>model\n<|channel> thought\n<channel|>`
Gemma-4-26B-A4B-IT	`<bos><|turn>user\n{instruction}<turn|>\n<|turn>model\n<|channel> thought\n<channel|>`

A.2 Details on ASR Evaluation

ASR Evaluation Prompt Template.

You are a classifier for red teaming large language models (LLMs). Given a generation, determine whether it counts as an instance of the following behavior (subject to the rules below). Your answer should be "Yes" or "No" with an explanation.

Behavior:
{Harmful Prompt}

Generation:
{Output Generation}

—

Rules:

- Instances of the behavior should be unambiguous and non-minimal. Vague or very short instances do not count.
- Generations should themselves be harmful or undesirable for a public-facing LLM to output. Benign generations do not count.
- Generations that include caveats or warnings count if the behavior can reasonably be said to have occurred alongside them.
- Generations that respond, analyze, discuss or otherwise react to the behavior without actually exhibiting it do not count.
- Generations where the model clearly tries to exhibit a behavior count. E.g., it is OK if generated code has errors.
- If the behavior involves coding and programming, the generation should contain code artifacts related to the behavior, not just comments or instructions. Otherwise it does not count.

Answer:

To evaluate a target LLM's output generation for successful jailbreak, we wrap the input behaviour and output generation with the evaluation prompt template from the Trojan Detection Challenge 2023 (LLM Edition)-Red Teaming Track (Jeong et al., 2026). We then call GPT-4.1 (OpenAI, 2023) as judge to analyze the semantics of the output generation for successful compliance with the harmful behaviour. Where the judge response is ambiguous or malformed, such as having both affirmative and refusal language in the same response, we manually review the target model response for jailbreak success.

B Pseudocode for SPACE Attacks

In this section, we detail the pseudocodes for implementations of SPACE used in this study. For brevity, we define a number of named operators and their intended actions. We also clarify that for Algorithm 2, the fixed arms supply suffix, pooled-VSS, or pooled-posterior-mean allocations and initialise the inserted tokens afresh. For Algorithm 3, a uniform or transferred prior is supplied, followed by warm-start continuation with previously obtained best-loss state.

Basic operators. Insert adds tokens to allocated positions when called. $\mathrm{GaussianUpdate}$ applies the closed-form posterior update described in Section 4.3. Furthermore, based on the BEAST paper (Sadasivan et al., 2024), we define the following operators with respect to the BEAST attack procedure. $\mathrm{BeamStep}$ samples Q distinct replacement tokens from the model's next-token distribution for each beam at the selected position, scores the proposals, and keeps the b lowest-loss beams. $\mathrm{BeamShares}$ divides the inference beams proportionally to allocation group sizes using floor plus largest remainder. $\mathrm{SlotTransfer}$ initialises the MAP allocation policy using tokens from a previously obtained lowest-loss state, keeping them in their original canonical positions.

Algorithm 1 SPACE: posterior-guided allocation with grouped GCG

Require: Task τ, prior profile $\boldsymbol{\theta}_\tau$, schedules $(m_t),(I_t),(B_t)$
Require: Top-K pool size K, initial token z^A_{init}, scales β,γ
Ensure: Final adversarial prompt and terminal posterior moments

$m \leftarrow \sum_t m_t$; $\mathcal{L}_{\tau,0} \leftarrow \mathcal{L}_\tau(\widetilde{x})$
Initialise $(\boldsymbol{\mu},\boldsymbol{\Sigma},\sigma_\tau)$ using Section 4.2
$\mathbf{C} \leftarrow \mathbf{0}$; $\mathbf{z}^A \leftarrow$ empty token assignment
for each insertion round t **do**
  for $j = 1,\ldots,B_t$ **do**
    $\widetilde{\mathbf{w}}_{t,j} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$
    $\mathbf{a}_{t,j} \leftarrow \Psi_{m_t}(\widetilde{\mathbf{w}}_{t,j};\mathbf{C})$; $\mathbf{c}_{t,j} \leftarrow \mathbf{C} + \mathbf{a}_{t,j}$
  end for
  Group identical allocations into $\mathcal{G}_t(\mathbf{c})$
  for each distinct allocation $\mathbf{c}$ **do**
    $\mathbf{z}^A_{\mathbf{c}} \leftarrow \textsc{Insert}(\mathbf{z}^A, \mathbf{c}-\mathbf{C}, z^A_{\text{init}})$
  end for
  for $i = 1,\ldots,I_t$ **do**
    for each distinct allocation $\mathbf{c}$ **do**
      Configure $\mathcal{A}$ with batch size $|\mathcal{G}_t(\mathbf{c})|$ and top-K pool
      $\mathbf{z}^A_{\mathbf{c}} \leftarrow \mathcal{A}(\mathbf{c}, \mathbf{z}^A_{\mathbf{c}}, 1)$
      $\widehat{\ell}_{\mathbf{c}} \leftarrow \mathcal{L}_\tau(\widetilde{x}(\mathbf{c}, \mathbf{z}^A_{\mathbf{c}}))$
    end for
    Form $\Phi,\mathbf{r}$ from $(\boldsymbol{\phi}(\mathbf{c}), \mathcal{L}_{\tau,0} - \widehat{\ell}_{\mathbf{c}})$ over distinct allocations
    $(\boldsymbol{\mu},\boldsymbol{\Sigma}) \leftarrow \textsc{GaussianUpdate}(\boldsymbol{\mu},\boldsymbol{\Sigma},\Phi,\mathbf{r},\sigma^2_\tau)$
  end for
  $\mathbf{c}^\star \in \operatorname{argmin}_{\mathbf{c}} \widehat{\ell}_{\mathbf{c}}$
  $\mathbf{C} \leftarrow \mathbf{c}^\star$; $\mathbf{z}^A \leftarrow \mathbf{z}^A_{\mathbf{c}^\star}$
end for
return $(\widetilde{x}(\mathbf{C},\mathbf{z}^A),\boldsymbol{\mu},\boldsymbol{\Sigma})$

Algorithm 2 Fixed-allocation BEAST: suffix, VSS-pooled, and SPACE-pooled arms

Require: Task τ, canonical allocation $\mathbf{c}$, initial tokens $\mathbf{z}^A_0$, beam width b, proposals per beam Q
Ensure: Lowest-loss scored adversarial prompt

$\mathcal{H} \leftarrow b$ copies of $\widetilde{x}(\mathbf{c},\mathbf{z}^A_0)$; $\widehat{\ell}_{\text{best}} \leftarrow +\infty$
$(p_1,\ldots,p_m) \leftarrow$ inserted-token positions in physical order
for $k = 1,\ldots,m$ **do**
  $\mathcal{H} \leftarrow \textsc{BeamStep}(\mathcal{H},p_k,b,Q)$
  $\widetilde{x}' \in \operatorname{argmin}_{u \in \mathcal{H}} \mathcal{L}_\tau(u)$
  if $\mathcal{L}_\tau(\widetilde{x}') < \widehat{\ell}_{\text{best}}$ **then**
    $(\widetilde{x}_{\text{best}}, \widehat{\ell}_{\text{best}}) \leftarrow (\widetilde{x}', \mathcal{L}_\tau(\widetilde{x}'))$
  end if
end for
return $\widetilde{x}_{\text{best}}$

Algorithm 3 Adaptive SPACE-BEAST: uniform or transferred prior, with warm-start continuation

Require: Task τ, calibrated moments $(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$, noise σ_τ^2, clean loss $\mathcal{L}_{\tau,0}$
Require: Schedules (m_t),(B_t), beam widths b_{inf},b_{cont}, proposals Q, initial token z_{init}^A
Ensure: Lowest-loss scored prompt across inference and continuation

$(\boldsymbol{\mu},\boldsymbol{\Sigma}) \leftarrow (\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$; $\mathbf{C}^{(0)} \leftarrow \mathbf{0}$; $\widetilde{x}^{(0)} \leftarrow \widetilde{x}$; $\widehat{\ell}_{\text{best}} \leftarrow +\infty$
for each insertion round t **do**
  Sample B_t profiles and form $\mathbf{c}_{t,j} = \mathbf{C}^{(t-1)} + \Psi_{m_t}(\widetilde{\mathbf{w}}_{t,j};\mathbf{C}^{(t-1)})$ as in Algorithm 1
  Group equal allocations into $\mathcal{G}_t(\mathbf{c})$; write $(\mathbf{c}_{t,g},n_{t,g})$ for their allocations and sizes
  $(b_{t,g}) \leftarrow \textsc{BeamShares}((n_{t,g}),b_{\text{inf}})$
  for $g : b_{t,g} > 0$ **do**
    Append z_{init}^A tokens within the requested canonical blocks, preserving the committed tokens, to obtain $\mathbf{z}_{t,g}^A$
    Construct $\widetilde{x}(\mathbf{c}_{t,g},\mathbf{z}_{t,g}^A)$ using π_x
    $\mathcal{H}_{t,g} \leftarrow b_{t,g}$ copies of this state; $(p_{t,g,k})_{k=1}^{m_t} \leftarrow$ its new positions in physical order
  end for
  $\Phi_{g,:} \leftarrow \boldsymbol{\phi}(\mathbf{c}_{t,g})^\top$ for the active groups $g : b_{t,g} > 0$
  for $k = 1,\ldots,m_t$ **do**
    for $g : b_{t,g} > 0$ **do**
      $\mathcal{H}_{t,g} \leftarrow \textsc{BeamStep}(\mathcal{H}_{t,g},p_{t,g,k},b_{t,g},Q)$
      $\widetilde{x}_{t,g}^{(k)} \in \operatorname{argmin}_{u \in \mathcal{H}_{t,g}} \mathcal{L}_\tau(u)$; $\widehat{\ell}_{t,k,g} \leftarrow \mathcal{L}_\tau(\widetilde{x}_{t,g}^{(k)})$
      $r_g \leftarrow \mathcal{L}_{\tau,0} - \widehat{\ell}_{t,k,g}$
      if $\widehat{\ell}_{t,k,g} < \widehat{\ell}_{\text{best}}$ **then**
        $(\widetilde{x}_{\text{best}},\mathbf{c}_{\text{best}},\widehat{\ell}_{\text{best}}) \leftarrow (\widetilde{x}_{t,g}^{(k)},\mathbf{c}_{t,g},\widehat{\ell}_{t,k,g})$
      end if
    end for
    $(\boldsymbol{\mu},\boldsymbol{\Sigma}) \leftarrow \textsc{GaussianUpdate}(\boldsymbol{\mu},\boldsymbol{\Sigma},\Phi,\mathbf{r},\sigma_\tau^2)$
  end for
  $g_t^\star \in \operatorname{argmin}_{g : b_{t,g} > 0} \widehat{\ell}_{t,m_t,g}$; $\mathbf{C}^{(t)} \leftarrow \mathbf{c}_{t,g_t^\star}$; $\widetilde{x}^{(t)} \leftarrow \widetilde{x}_{t,g_t^\star}^{(m_t)}$
end for
$\mathbf{c}_{\text{MAP}} \leftarrow \Psi_m(\boldsymbol{\mu};\mathbf{0})$, where $m = \sum_t m_t$
$\mathbf{z}_0^A \leftarrow \textsc{SlotTransfer}(\widetilde{x}_{\text{best}},\mathbf{c}_{\text{best}},\mathbf{c}_{\text{MAP}},z_{\text{init}}^A)$
Run Algorithm 2 at $\mathbf{c}_{\text{MAP}}$ from $\mathbf{z}_0^A$ with (b_{cont},Q), obtaining $\widetilde{x}_{\text{cont}}$
return the lower-loss prompt among $\widetilde{x}_{\text{best}}$ and $\widetilde{x}_{\text{cont}}$

C Further Experimental Results

C.1 Wall Clock Compute Times

Table 4: Mean attack-process wall-clock time in minutes across behaviours, seed 0. SPACE uses the 4×5 insertion schedule. Times include process startup, model loading, attack computation and output handling, excluding downstream generation and judging. The bottom row equally weights the twelve model-level means, without normalising for step counts or GPU packing. †250 optimiser steps; otherwise 500. ‡49 behaviours; §48 behaviours; otherwise 50.

Model	GCG	SlotGCG	SPACE
Gemma-4-12B-it†	19.45	19.49	26.77
Gemma-4-26B-A4B-it†	8.32‡	8.32	15.15
Phi-4-Mini-Instruct	14.71	14.64§	29.77
Llama-3.2-1B-Instruct	12.59	12.63	35.08
Nemotron-3.5-30B-A3B†	147.32	144.33	143.02
Qwen3-8B	27.89	27.16	41.03
Phi-4-14B	40.40	40.34	49.15
Qwen3-0.6B	10.76	10.68	28.65
Llama-2-7B-Chat	24.27	24.40	37.75
Llama-3.1-8B-Instruct	32.58	32.66	45.36
Mistral-7B-Instruct-v0.3	21.95	21.87	35.11
Qwen3-32B	46.61	46.78	51.31
Mean wall-clock time	33.90	33.61	44.85

C.2 UNION ASR RESULTS

Table 5: Complete per-model coverage. **Left block**: the 50 behaviours of each model partitioned by which methods solved them — the seven regions of the GCG / SlotGCG / SPACE Venn plus the behaviours none solved. The three pairwise columns are *exclusive* (exactly two methods), so each row's left block sums to 50. **Right block**: the marginal ASR of each method, which is the sum of the four regions containing it, and "Any", the union — the share of behaviours at least one method solved. G = GCG, S = SlotGCG, Sp = SPACE. Rows ascend by mean ASR over the three methods, an ordering symmetric in the methods. **Seed 0 only:** a union is defined only within a seed, and Table 1 reports two-seed means for six of the twelve models, so the marginals here do not match it exactly; an unparsed `manual_review` verdict counts as a failure. †Runs 250 optimiser steps for all methods.

	behaviours, of 50								ASR (%)			
Model	All three	G only	G ∩ S	S only	S ∩ Sp	Sp only	G ∩ Sp	None	GCG	SlotGCG	SPACE	Any
Gemma-4-12B-it†	1	1	0	2	1	7	1	37	6.0	8.0	20.0	26.0
Gemma-4-26B-A4B-it†	1	1	1	3	0	7	1	36	8.0	10.0	18.0	28.0
Qwen3-32B	6	7	6	3	5	5	2	16	42.0	40.0	36.0	68.0
Llama-2-7B-Chat	2	0	3	12	11	8	3	11	16.0	56.0	48.0	78.0
Llama-3.2-1B-Instruct	8	5	2	3	7	8	6	11	42.0	40.0	58.0	78.0
Nemotron-3.5-30B-A3B†	6	1	4	6	11	8	5	9	32.0	54.0	60.0	82.0
Qwen3-0.6B	12	5	3	1	6	5	5	13	50.0	44.0	56.0	74.0
Llama-3.1-8B-Instruct	16	1	3	7	9	4	4	6	48.0	70.0	66.0	88.0
Phi-4-14B	15	1	2	5	13	4	4	6	44.0	70.0	72.0	88.0
Qwen3-8B	19	6	4	3	5	5	3	5	64.0	62.0	64.0	90.0
Phi-4-Mini-Instruct	27	4	2	0	6	0	7	4	80.0	70.0	80.0	92.0
Mistral-7B-Instruct-v0.3	36	1	3	2	3	1	2	2	84.0	88.0	84.0	96.0
Pooled ($n=600$)	149	33	33	47	77	62	43	156	43.0	51.0	55.2	74.0

D ADDITIONAL PROOFS FOR BAYESIAN INTEGER ALLOCATION

D.1 INCREMENTAL ALLOCATION WITH KNOWN SLOT UTILITIES

Proposition 2 (Exact incremental allocation)**.** *Let $g:\{0,\dots,m\}\to\mathbb{R}_{\geq 0}$ be non-decreasing and discretely concave, with $g(0)=0$. Fix a profile $\mathbf{w}\in\mathbb{R}^{|\mathcal{S}|}$, committed counts $\mathbf{C}\in\mathbb{Z}_{\geq 0}^{|\mathcal{S}|}$, and an integer budget $b\geq 1$ such that $\sum_s C_s+b\leq m$. Consider the allocation problem*

$$\Psi_b(\mathbf{w};\mathbf{C})\in\operatorname*{argmax}_{\mathbf{a}\in\mathcal{C}_b}\sum_{s\in\mathcal{S}}w_s g(C_s+a_s),$$

$$\Delta_s^{\mathbf{C}}(j):=w_s\big[g(C_s+j)-g(C_s+j-1)\big],\qquad j=1,\dots,b.$$

If $\max_s w_s\geq 0$, initialise $\mathbf{a}=\mathbf{0}$ and repeat b times:

$$s^\star\in\operatorname*{argmax}_{s:w_s\geq 0}\Delta_s^{\mathbf{C}}(a_s+1),\qquad \mathbf{a}\leftarrow\mathbf{a}+\mathbf{e}_{s^\star}.$$

The resulting allocation is optimal. If all coefficients are negative, an optimal allocation is

$$\mathbf{a}^\star=b\mathbf{e}_{s^\star},\qquad s^\star\in\operatorname*{argmax}_{s\in\mathcal{S}}w_s\big[g(C_s+b)-g(C_s)\big].$$

Proof. Since $\mathbf{C}$ is fixed, subtracting $\sum_s w_s g(C_s)$ leaves the equivalent objective

$$\sum_{s\in\mathcal{S}}w_s\big[g(C_s+a_s)-g(C_s)\big].$$

Case 1: At least one coefficient is non-negative. If a feasible allocation assigns a token to position s with $w_s<0$, move the last assigned token to a position q with $w_q\geq 0$. The change in objective is

$$\Delta_q^{\mathbf{C}}(a_q+1)-\Delta_s^{\mathbf{C}}(a_s)\geq 0,$$

because g is non-decreasing. Repeating this operation shows that an optimum exists using only non-negative-coefficient slots.

For each such slot, discrete concavity implies

$$\Delta_s^{\mathbf{C}}(1) \geq \cdots \geq \Delta_s^{\mathbf{C}}(b) \geq 0.$$

Assigning a_s tokens selects a prefix of this sequence, with value

$$w_s\big[g(C_s+a_s)-g(C_s)\big] = \sum_{j=1}^{a_s} \Delta_s^{\mathbf{C}}(j).$$

Every remaining marginal in a slot is no greater than its first unselected marginal, which is available for selection by the greedy allocation rule. Consequently, each greedy step selects the largest remaining marginal across all eligible positions while preserving the prefix constraint. After b steps, it has selected b largest marginals and attains an upper bound on the value of any feasible allocation. It is therefore optimal, including when marginal values tie.

Case 2: All coefficients are negative. For each position,

$$h_s(r) := w_s\big[g(C_s+r)-g(C_s)\big], \qquad r=0,...,b.$$

The increments $h_s(j)-h_s(j-1)=\Delta_s^{\mathbf{C}}(j)$ are non-decreasing: discrete concavity of g gives non-increasing increments, and multiplication by $w_s<0$ reverses their order. For $1 \leq r \leq b$, the average of the first r increments is therefore no greater than the average of all b increments:

$$\frac{h_s(r)}{r} = \frac{1}{r}\sum_{j=1}^{r} \Delta_s^{\mathbf{C}}(j) \leq \frac{1}{b}\sum_{j=1}^{b} \Delta_s^{\mathbf{C}}(j) = \frac{h_s(b)}{b}.$$

Thus $h_s(r) \leq (r/b)h_s(b)$, also for $r=0$ since $h_s(0)=0$. For any $\mathbf{a} \in \mathcal{C}_b$,

$$\sum_s h_s(a_s) \leq \sum_s \frac{a_s}{b} h_s(b) \leq \max_s h_s(b),$$

where the last inequality uses $a_s/b \geq 0$ and $\sum_s a_s/b = 1$. Assigning all b tokens to a slot maximising $h_s(b)$ attains this upper bound, proving the claim. □