LEVERAGING RAG FOR TRAINING-FREE ALIGNMENT OF LLMS

John T. Halloran
Leidos
halloranjt@leidos.com

ABSTRACT

Large language model (LLM) alignment algorithms typically consist of post-training over preference pairs. While such algorithms are widely used to enable safety guardrails (i.e., learning to refuse malicious requests containing harmful language) as well as align LLMs with general human preferences, we show that state-of-the-art alignment algorithms are far less capable of enabling refusal guardrails for recent agentic attacks. To thus improve refusal guardrails against such attacks, we introduce *R*etrieval *A*ugmented *G*eneration for *Pref*erence alignment (RAG-Pref), a simple (yet effective) RAG-based alignment algorithm which conditions on preferred and dispreferred samples to leverage *contrastive information* during inference. RAG-Pref is online (training-free), compatible with off-the-shelf packages, and, when combined with offline (training-based) alignment algorithms, enables more than an average 3.7 factor improvement in agentic attack refusals across five widely used LLMs, compared to 2.9 for other online alignment algorithms and 1.5 for offline alignment alone. We conclude by showing that, in stark contrast to other online alignment methods, RAG-Pref similarly increases performance on general human-preference alignment tasks and does not drastically increase overall computational requirements.

1 INTRODUCTION

Alignment has become a critical step towards ensuring the responses of large language models (LLMs) align with general human preferences–e.g., that responses follow provided instructions during multi-turn dialogues (Zheng et al., 2023). Currently, alignment algorithms are dominated by reinforcement learning-based schemes–such as RLHF (Ouyang et al., 2022) and the computationally efficient direct preference optimization (DPO) (Rafailov et al., 2023)–wherein models are post-trained over pairs of preferred and dispreferred responses for each input query. The responses from resulting models are thus aligned with the desirable behaviors present in preferred training data, while the undesirable behaviors present in dispreferred training data are avoided. Such *alignment-tuning*–i.e., alignment via RLHF, DPO, or derivative preference optimization post-training algorithms–has proven pivotal towards producing LLM assistants whose responses are both accurate and *helpful* (Ouyang et al., 2022; Zheng et al., 2023; Achiam et al., 2023; Dubois et al., 2024; Zhong et al., 2025).

Owing to both the success of alignment-tuning and growing evidence that LLMs are highly susceptible to adversarial attacks (Mehrotra et al., 2024; Liu et al., 2025; Chao et al., 2025; Liu et al., 2023b; Shen et al., 2024; Zhang et al., 2023), significant works have sought to produce LLM assistants whose responses are aligned with *harmless* behaviors (Bai et al., 2022; Ji et al., 2023; Tian et al., 2024), in addition to helpful behaviors. Such works have shown that safety alignment-tuning (SAT) may be achieved by directly including harmless and harmful preference pairs during alignment-tuning (Ji et al., 2023; Team et al., 2023; Tian et al., 2024), thus enabling *refusal guardrails*, i.e., models learn to refuse malicious instructions containing harmful language, while complying with benign requests. Furthermore, when additional safety labels are available for preference training pairs, follow up works have shown that specialized SAT algorithms may further strengthen refusal guardrails (Dai et al., 2024a; Kim et al., 2025).

SAT has become a ubiquitous and important security step when deploying frontier models (Hurst et al., 2024; Dubey et al., 2024; Yang et al., 2025; DeepSeek-AI, 2025; Anthropic, 2025). However, with the recent advent of a universal agentic protocol–i.e., the *model context protocol* (MCP) (Anthropic, 2025b)–new security risks have emerged (Kumar et al., 2025; Invariant, 2025d;a; Radosevich & Halloran, 2025). In particular, despite their extensive safety alignment-tuning (Grattafiori et al., 2024; Anthropic, 2025), MCP-enabled LLMs were shown to be highly susceptible to adversarial attackers wherein queries induce malicious tool use yet lack standard refusal trigger phrases (i.e., harmful language) (Radosevich & Halloran, 2025). Such agentic attacks, which lack common refusal trigger phrases, are referred to herein as *falsely benign attacks* (FBAs).

To explore the efficacy of state-of-the-art (SOTA) SAT algorithms to strengthen refusal guardrails against FBAs, we curate a high-quality collection of FBAs as well as truly benign (TB) samples. Safety-tuning five popular LLMs using both DPO and SafeDPO (Kim et al., 2025), we show that SOTA SAT algorithms display limited ability to enable FBA refusal guardrails; across all models, DPO and SafeDPO improve baseline refusal rates by only an average factor of 1.4 and 1.6, respectively. Alarmingly, no safety-tuned model achieves an FBA refusal rate greater than 48%. Thus, to further improve refusal guardrails in the face of agentic attacks, we introduce *R*etrieval *A*ugmented *G*eneration for *Pref*erence alignment (RAG-Pref), a new algorithm which utilizes RAG to *contrastively condition* on both preferred and dispreferred examples during inference. Compared to standard RAG, we show that RAG-Pref is guaranteed to further reduce the expected uncertainty during inference by a nonnegative amount, referred to as the *contrastive information*.

RAG-Pref is online (training-free), easily implementable using off-the-shelf packages/components, and significantly improves LLM refusal guardrails compared to both SOTA offline and online alignment algorithms. E.g., RAG-Pref increases baseline refusal rates by an average 3-fold improvement–2.1 and 1.9 times greater than DPO and SafeDPO, respectively. Moreover, we show that RAG-Pref further improves performance when augmenting generation with samples from the LLM's closed-book (parametric) knowledge; when combined with DPO and SafeDPO aligned models, RAG-Pref further increases baseline refusal rates by an average 3.5- and 3.9-fold improvements, respectively.

In addition to significantly improving refusal guardrails for agentic attacks, we demonstrate that RAG-Pref similarly improves performance for general human-preference alignment tasks, leading to an average 34.9% and 3.3% increase in AlpacaEval 2 and MT-Bench performance, respectively, across SOTA alignment-tuned models. Additionally, we contrast the computational and practical advantages of RAG-Pref with offline and other online alignment algorithms, demonstrating RAG-Pref requires three-orders of magnitude less time and two-orders of magnitude less GPU memory to construct databases than DPO required to (offline) safety-tune a 14B parameter model. Furthermore, RAG-Pref only incurs an inference slowdown of 20% while natively supporting popular open-source models owing to the use of off-the-shelf, widely-adapted packages. In stark contrast, recent online, decoding-based alignment strategies (Zhu et al., 2025) require 372% more time than standard inference and do not natively support widely-used models, owing to complex, custom inference algorithms.

2 Background and Related Work

General Human-Preference Alignment. LLM alignment near ubiquitously consists of fine-tuning given queries with accompanying preferred and dispreferred response pairs (which reflect human preferences). Initial alignment approaches, such as RLHF (Ouyang et al., 2022) and RLAIF (Lee et al., 2024), first trained reward models on human preference data, then utilized reinforcement learning (RL) fine-tuning to learn a policy which maximized the reward signal. To address training instability in RL fine-tuning, subsequent work reparameterized the RL objective, allowing direct learning of the optimal policy with a simple closed form objective (Rafailov et al., 2023). The resulting algorithm (i.e., DPO), was shown to provide significantly better training stability than RL-based alignment.

For widely adapted general human-preference benchmarks–e.g., AlpacaEval 2. (Dubois et al., 2024) and MT-Bench (Zheng et al., 2023)–DPO and its many follow-up variants (Melnyk et al., 2024; D'Oosterlinck et al., 2025; Jung et al., 2024; Ji et al., 2024; Liu et al., 2024; Chen et al., 2024; Chowdhury et al., 2024; Wu et al., 2024), have been extensively studied and demonstrated strong performance. Furthermore, the recent reinforced token optimization (RTO) (Zhong et al., 2024) has

shown that by optimizing over token-wise reward signals, performance on general human-preference tasks (e.g., AlpacaEval 2) may be significantly improved.

For online training-free alignment, (Zhu et al., 2025) introduced On-the-fly Preference Alignment via Principle-Guided Decoding (OPAD). OPAD calculates a similar reward function as used in DPO to adjust the per-token conditional distribution during decoding and was shown to improve performance on general preference alignment tasks relative to previous online methods (Gao et al., 2024). OPAD's reward function and decoding procedure are further detailed in Section 3.

Safety Alignment-Tuning. Additional works have sought to further focus alignment on safety and decrease the risk of unsafe behaviors–e.g., toxicity (Hartvigsen et al., 2022), hate speech (Mazeika et al., 2024), and compliance with malicious/violent requests (Li et al., 2024). (Dai et al., 2024b) utilized additional labels (safe/unsafe) for each preferred and dispreferred training pair used in RLHF to derive a three-round fine-tuning algorithm, called Safe RLHF, which optimized both for preference and safety alignment. Subsequently, (Kim et al., 2025) showed that, in the presence of safe/unsafe labels for preferred and dispreferred pairs, the DPO objective could be adjusted to simultaneously optimize for safety with an additional loss offset. The resulting algorithm, called SafeDPO, was shown to offer improved training stability and safety alignment compared to Safe RLHF, similar to RLHF and DPO.

Agentic adversarial attacks. Extensive works have studied the susceptibility of LLMs to attacks which circumvent refusal guardrails for malicious purposes. Jailbreaks (Zou et al., 2023; Chao et al., 2025) craft input prompts which evade guardrails to elicit unsafe responses. Prompt injection attacks (PIAs) (Perez & Ribeiro, 2022; Liu et al., 2023a; Greshake et al., 2023) consist of injecting malicious instructions into user prompts. Encoded prompt injections encode malicious commands in an alternative format (e.g., octal) for PIAs. However, frontier LLM safety training has grown to include standard LLM attacks (Mazeika et al., 2024; Chao et al., 2024), thus expanding refusal guardrails to include existing jailbreaks and PIAs (Sharma et al., 2025; Grattafiori et al., 2024).

Recently, the MCP has seen massive, widespread adaption (Anthropic, 2025a). By standardizing API calls between LLMs, tools, and data sources, the MCP enables seamless integration between generative AI agents and widely used applications (Google, 2025; Anthropic, 2025c; Microsoft, 2025; Stripe, 2025). However, distinct from previous LLM jailbreaks and PIAs, recent work has shown that the MCP introduces new attack possibilities in agentic systems (Radosevich & Halloran, 2025).

Radosevich & Halloran (2025) demonstrated that MCP-enabled agents are susceptible to PIAs which explicitly lack refusal guardrail triggers. I.e., while refusal guardrails are triggered by malicious PIAs which explicitly state harmful phrases or suspicious text, attacks lacking these exact triggers are successfully completed. As previously mentioned, we term such PIAs which lack standard harmful or suspicious cues *falsely benign attacks* (FBAs). The success of FBAs is attributed to the shift in attack goals from LLMs–which focus on unsafe text generation, the attacks of which contain related harmful phrases or suspicious text–to MCP-enabled LLMs–which focus on the malicious execution of tools, the attacks of which need not contain harmful or suspicious text found in standard SAT data.

Agentic Safety work. Existing MCP defenses have focused on either inspecting MCP-servers for potential vulnerabilities (Invariant, 2025c; Halloran, 2025) or monitoring user queries for suspicious traffic (Invariant, 2025b). However, to the best of the authors' knowledge, no previous works have explored the effect of MCP-specific attacks on SAT algorithms and the performance of resulting refusal guardrails. Furthermore, in contrast to existing work seeking to optimize specific components in RAG systems–described as retriever-generator alignment (Wu et al., 2025; Jin et al., 2025; Sun et al., 2025)–or using RAG to generate preference alignment data (Song et al., 2025), the authors are unaware of previous work exploring online RAG alignment algorithms.

3 OFFLINE AND ONLINE PREFERENCE ALIGNMENT

Let x be an input prompt. For an autoregressive LLM π_θ, consider the probability of generating a response y consisting of T tokens:

$$\pi_\theta(y|x) = \prod_{t=1}^{T} \mathbf{P}_{\pi_\theta}(y_t|x, y_{1:t-1}), \tag{1}$$

where $\mathbf{P}_{\pi_\theta}(y_t|x, y_{1:t-1}) \in \mathbb{R}^{L_t \times V}$, L_t is the sequence length at time t, and V is the vocabulary size.

Let $\mathcal{D} = \{(x^1, y_w^1, y_l^1), \ldots, (x^n, y_w^n, y_l^n)\}$ be a set of preference data where, for input x^i, y_w^i is the preferred response, and y_l^i is the dispreferred response, denoted as $y_w^i \succ y_l^i \mid x$. Let π_θ be a model to be optimized and π_{ref} a reference model (which is typically a supervised fine-tuned version of the model trained prior to preference alignment). The goal of offline alignment with DPO is thus to train a new model by solving

$$\pi_\phi^* = \arg\max_{\pi_\phi} \mathcal{L}(r_\phi),$$
$$\mathcal{L}(r_\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta r_\phi(x, y_w) - \beta r_\phi(x, y_l)\right)\right],$$

where $r_\phi(x, y_w) = \log \frac{\pi_\phi(y_w|x)}{\pi_\theta(y_w|x)}$ is the reward function. DPO thus learns parameters which align with preferred responses, without drastically diverging from the reference model.

To avoid training, On-the-fly Preference Alignment via Principle-Guided Decoding (OPAD) (Zhu et al., 2025) includes additional helpfulness instructions c and adjusts the per-timestep distribution in Equation 1. Rewriting the reward function at timestep t as $r_\theta(x, y_{1:t}, c) = \log \frac{\pi_\theta(y_{1:t}|x,c)}{\pi_\theta(y_{1:t}|x)}$, the tth token distribution is adjusted during generation as

$$\mathbf{P}_{\pi_\theta}(y_t|x, c, y_{1:t}) \propto \pi_\theta(y_t|x, c, y_{1:t}) \exp\left(\frac{1}{\beta} r_\theta(x, y_{1:t}, c)\right).$$

We note that the above decoding procedure requires both invasive changes to generation (raising potential compatibility issues with deployed models) and a significant increase in computational resources–i.e., per time-step, calculation of the reward function and the subsequent distribution update requires maintaining three separate distributions.

3.1 TRAINING-FREE ALIGNMENT WITH RAG-PREF

We now detail how online preference alignment may be performed without requiring any invasive adjustments to model generation via RAG-Pref. Let $e(\cdot) \in \mathbb{R}^m$ be a text embedding function (trained to embed semantically similar text near one another in the embedding space), and $d(\cdot, \cdot) \in \mathbb{R}$ be a vector-distance metric. For our preference dataset $\mathcal{D}$, let $\mathcal{D}^w = \{y_w : (x, y_w, y_l) \in \mathcal{D}\}$ and $\mathcal{D}^l = \{y_l : (x, y_w, y_l) \in \mathcal{D}\}$ be the sets of preferred and dispreferred text responses, respectively, and let $\mathcal{D}_e^w = \{e(y) : y \in \mathcal{D}^w\}$ and $\mathcal{D}^l = \{e(y) : y \in \mathcal{D}^l\}$ be the vector databases of preferred and dispreferred embeddings, respectively.

Algorithm 1 RAG-Pref for online alignment.

Input: Query x, $\mathcal{D}_e^w$, $\mathcal{D}_e^l$, and number of retrieval elements k.

1: Embed x, $x' = e(x)$.
2: For all $z \in \mathcal{D}_e^w$, rank each element by $d(x', z)$, sort, and return the top k sequences $\mathcal{Z}^w \subseteq \mathcal{D}^w$
3: For all $z \in \mathcal{D}_e^l$, rank each element by $d(x', z)$, sort, and return the top k sequences $\mathcal{Z}^l \subseteq \mathcal{D}^l$
4: Create an instruction, denoted as $\mathcal{Z}^w \succ \mathcal{Z}^l$, that model responses are to follow retrieved preference instances and avoid dispreferred instances.
5: **return** $\pi_\theta(y|x, \mathcal{Z}^w \succ \mathcal{Z}^l)$.

The RAG-Pref algorithm is detailed in Algorithm 1. Compared to alternative alignment algorithms, we note that:

- DPO, and related offline algorithms, enforce preference alignment over response pairs (y_w, y_l) to a given query. In contrast, RAG-Pref enables preference alignment over sets of responses, $\mathcal{Z}^w$ and $\mathcal{Z}^l$.
- No invasive or complicated changes to generation are required, thus allowing compatibility with widely supported packages and widely used models.

3.2 RAG-PREF ENCODES CONTRASTIVE INFORMATION

We now prove theoretical results for RAG and RAG-Pref. For the sets of all queries $\mathcal{X}$, all responses $\mathcal{Y}$, and all retrieval documents $\mathcal{D}$, let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be a random query and response, respectively, and let $Z^w, Z^l \in \mathcal{D}$ be random retrieved preference and dispreference documents, respectively. Note that, for simplicity, we overload the term document; when $k \geq 1$, Z^w and Z^l represent the respective k-shot demonstrations concatenated together in Algorithm 1, which, without loss of generality, may each be represented as a document in $\mathcal{D}$ which have been retrieved.

Firstly, for the distribution returned in Algorithm 1–$\pi_\theta(Y|X, Z^w \succ Z^l) = \pi_\theta(Y|X, Z^w, Z^l)$–we note that RAG-Pref performs *contrastive conditioning*, i.e., it conditions generation not only on positive examples (what to do) but also on negative examples (what to avoid). Furthermore, standard RAG lacks this contrastive conditioning, and thus does not condition generation on behaviors to avoid.

Contrastive conditioning allows us to exactly quantify the reduction in uncertainty provided by RAG-Pref compared to standard RAG. Let $\Delta H_{\text{RAG}} = I(Y; Z^w|X)$ and $\Delta H_{\text{RAG-Pref}} = I(Y; Z^w, Z^l|X)$, where $I(\cdot)$ is the mutual information. We define the *contrastive information* to be $\Delta H_{\text{RAG-Pref}} - \Delta H_{\text{RAG}}$, which is the amount of additional expected information provided by Algorithm 1 over standard RAG. In the following, we prove the that the contrastive information is guaranteed to be nonnegative:

Theorem 3.1. *$\Delta H_{\text{RAG-Pref}} \geq \Delta H_{\text{RAG}}$. Furthermore, when dispreferred examples provide non-redundant information compared to preferred examples, $\Delta H_{\text{RAG-Pref}} > \Delta H_{\text{RAG}}$.*

The proof of Theorem 3.1 is available in Appendix A.

We note that for safety guardrail alignment, attack patterns are often semantically distinct from refusal responses, thus providing substantial contrastive information. Furthermore, Theorem 3.1 explains recent empirical findings, wherein standard RAG was shown to degrade LLM refusal guardrails (An et al., 2025); standard RAG can *decrease* safety by retrieving attack examples without contrastive refusal examples, causing the model to misinterpret these as behaviors to follow rather than refuse.

Finally, we show the following related result.

Theorem 3.2. *The maximum reduction in uncertainty between standard inference and RAG/RAG-Pref is lower-bounded by the contrastive information.*

The proof of Theorem 3.2 is available in Appendix B. This result thus shows the total reduction in conditional entropy from standard inference and RAG/RAG-Pref can be no lower than the contrastive information encoded by RAG-Pref.

4 METHODS

FBA alignment data. FBAs were obtained by mapping an extensive catalog of known exploits to the sequence of MCP tools capable of achieving the exploit. Herein, we consider 10 tools which equip agents with Linux-like file/directory manipulation abilities (listed in Table 5). Attacks were obtained from the Common Vulnerabilities and Exposures (CVEs) (Mann & Christey, 1999) catalog, an up-to-date corpus of cyber attacks and exploits maintained by MITRE. The MITRE corpus was filtered for the agentic attacks used in (Radosevich & Halloran, 2025)–malicious code execution, remote access control, and credential theft–resulting in ∼34k samples. Using GPT-4O, each CVE attack was then: a) mapped to a sequence of Linux commands using GPT-4O, (b) marked as feasible or not given the set of MCP tools, (c) feasible attacks mapped to sequences of MCP tool calls (thus

forming FBA dispreferred responses), (d) friendly malicious requests (i.e., FBAs) are generated given the original CVE goal. This resulted in 1,150 FBA dispreferred responses. Finally, for each FBA, preferred responses were set to a direct refusal.

TB samples were collected by prompting CLAUDE 3.7 SONNET to create several useful examples per MCP tool while assuming specific roles (e.g., system admin, AI researcher, etc.), thus generating TB queries and preferred responses. TB dispreferred responses were created by setting the tools used during completion to their opposite (e.g., `read_file` substituted to `write_file`). The final dataset consists of 1,035 training FBAs, 1,035 training TB samples, and 115 FBA testing samples. Further pipeline details are available in Appendix E.

We note that this dataset is designed to enable refusal guardrails given paired FBA and TB data, wherein FBA preferred samples are direct refusals, FBA dispreferred samples are attacks based on MCP tool calls, TB preferred samples are benign MCP tool calls, and TB dispreferred samples are benign (but incorrect) MCP tool calls. **The ability** for each FBA's sequence of malicious MCP tool calls–as well as each SAT model–**to exactly achieve the original CVE attack is outside the scope of this work**, as is the ability to accurately assess and report FBA attack success rates. Rather, **we focus on the ability** of SOTA SAT algorithms **to enable refusal guardrails against FBAs, and thus report FBA refusal scores**.

Agentic Safety Alignment. RAG-Pref was run using off-the-shelf libraries (`ChromaDB` and `LangChain`) and embedding models (`sentence-transformers/all-MiniLM-L6v2`) for retrieval. For agentic safety results, RAG-Pref was run using the preference pairs from the TB and FBA training sets (described in Section 4) as preferred and dispreferred vector databases, with number of retrieved samples $k = 2$. OPAD results were collected using the harmlessness implementation from the official repo (Zhu, 2025), and extensively optimized to handle both multiple instances during generation and vectorization for a fair computational comparison. DPO and SafeDPO were run using `TRL` and the official paper code, respectively, with their training recipe adapted from (Tunstall et al., 2023; Zhou et al., 2023). Further implementation details are available in Appendix E.

For each method, ten generations were assessed per test instance and average refusal rate reported, with refusals calculated using a two-stage LLM-judge system (described in Appendix I).

General Human-Preference Alignment. Supervised fine-tuning was first performed on `Llama3-8B` using the `UltraFeedback` dataset (Cui et al., 2023). Subsequent offline preference alignment algorithms DPO, RLHF (via proximal policy optimization (PPO), SimPO (Meng et al., 2024), and the state-of-the-art reinforced token optimization (RTO) were performed with the widely used binarized `UltraFeedback` dataset (HuggingFace, 2025). Model checkpoints for offline preference aligned methods were directly adapted from (Zhong et al., 2024).

RAG-Pref was run using the preferred and dispreferred training instances from the binarized `UltraFeedback` dataset (RAG was run using the preferred instances). Both AlpacaEval 2 and MT-Bench were run using GPT-4O as the auto-annotator. For the former, both win rates and length-controlled win rates (Dubois et al., 2024) are reported. For the latter, the suggested single-answer grading was performed, wherein the LLM judge provides a qualitative score (on a scale of 10) for each answer per turn, with reported scores averaged over all turns. Across all models, AlpacaEval 2 and MT-Bench were evaluated using greedy decoding.

Table 1: **FBA Refusal Rates for Offline and Online Aligned Models**: Refusal rates calculated over the test FBAs. Bold = highest per model. GEMMA-2-2B-IT incompatible with OPAD.

	Offline Only			+ OPAD			+ RAG-Pref		
Model	Base	DPO	SafeDPO	Base	DPO	SafeDPO	Base	DPO	SafeDPO
LLAMA-3.2-1B-INSTRUCT	0.15	0.31	0.40	0.59	0.61	0.66	0.28	0.58	**0.88**
GEMMA-2-2B-IT	0.32	0.45	0.47	–	–	–	0.63	0.74	**0.75**
LLAMA-3.1-8B-INSTRUCT	0.35	0.43	0.45	0.43	0.47	0.37	0.95	**0.97**	**0.97**
DEEPSEEK-R1-DISTILL-LLAMA-8B	0.14	0.15	0.13	0.44	0.47	0.45	0.59	0.59	**0.59**
DEEPSEEK-R1-DISTILL-QWEN-14B	0.16	0.18	0.19	0.41	0.44	0.46	0.64	**0.68**	0.64

5 EXPERIMENTS

FBA refusal guardrails. We consider five widely used open-source LLMs, varying in parameter count from 1B to 14B. RAG-Pref is compared to offline (DPO and SafeDPO) and online (OPAD) alignment methods. DPO and SafeDPO refusal alignment were performed using FBA and TB preference pairs (described in Section 4). RAG-Pref preferred and dispreferred vector databases were formed from the same preference pairs used for offline alignment. In addition to base models, RAG-Pref and OPAD were combined with offline aligned models. Further methodological details are discussed in Section 4.

Owing to the use of off-the-shelf components, RAG-Pref was compatible with all evaluated models. In stark contrast, OPAD's invasive decoding scheme was incompatible with GEMMA-2-2B-IT (including the original/unbatched version of the code, the compute-optimized version developed for timing comparisons, and extensively debugged versions).

Refusal rates were calculated using the test FBA data described in Section 4. Refusal rates across all methods are listed in Table 1. Despite the majority of evaluated base models undergoing excessive post-training safety alignment (Grattafiori et al., 2024; Gemma et al., 2024), no base model achieves an FBA refusal rate over 35%. Furthermore, while refusal rates improve given offline alignment, neither DPO nor SafeDPO enable refusal rates beyond 48%. Thus, **SOTA offline alignment methods provide limited refusal guardrails against FBAs**.

RAG-Pref successfully improves guardrails over all base and SAT models, while OPAD fails to improve LLAMA-3.1-8B-INSTRUCT SAFEDPO guardrails (yet succeeds on the other models its decoding scheme was compatible with). OPAD outperforms RAG-Pref for the base and DPO-aligned LLAMA-3.2-1B-INSTRUCT models. However, RAG-Pref greatly outperforms OPAD all other model configurations, providing an average 50% more refusal performance across the twelve models. Furthermore, for the models OPAD was unable to align, RAG-Pref improved refusal guardrails by an average 74%. Thus, **RAG-Pref drastically outperforms other online methods for the refusal of FBAs, while significantly improving the refusal guardrails of SOTA offline alignment methods**.

Example safety generations are displayed in Appendix K. Furthermore, we ablate several hyperparameters used for offline SAT methods:

- **SAT loss function:** Exploring the effect of the DPO loss on refusal alignment, we align `Llama-3.2-1B` using 10 different DPO loss functions in Figure 1. The default "sigmoid" loss, used for all other experiments herein, achieves the highest refusal rate (31.4%).
- **Number of SAT epochs:** In Figure 3, we increase the number of DPO training epochs to 90 (4 fold increase) for DEEPSEEK-R1-DISTILL-QWEN-14B. Training quickly converges within the original training recipe (15 epochs, i.e., 15,000 steps) in Figure 2.

Additionally, RAG-Pref is compared to standard RAG in Figure 4. Across all base and offline aligned models, RAG-Pref achieves higher refusal rates than standard RAG. Furthermore, while RAG-Pref uniformly improves refusal rates for its underlying model, standard RAG actually decreases refusal performance for LLAMA-3.1-8B-INSTRUCT and DEEPSEEK-R1-DISTILL-LLAMA-8B models, reconfirming the results of other recent work (An et al., 2025).

Computational Comparison For refusal alignment and guardrail evaluation of DEEPSEEK-R1-DISTILL-QWEN-14B, we compare the total training/preparation time, inference time, and inference memory usage of online and offline methods. Training data used was the 4,410 TB/FBA preference pairs and evaluation data (for inference) was the 115 FBA test samples. All experiments were conducted on an Nvidia L40S GPU with 48GB onboard memory. The batch size for training (DPO), vector database preparation (RAG-Pref), and inference (all methods) was maximized for each method given GPU memory. The original OPAD codebase was written for single-sample inference, and was subsequently optimized using batched inference and extensive vectorization for a fair computational comparison. Results are reported in Table 2.

RAG-Pref preprocessing time is 7,824 times faster than DPO offline alignment. Furthermore, RAG-Pref does not add significant inference overhead compared to other online alignment methods; relative to DPO inference with no online alignment, RAG-Pref is 3 times faster than OPAD, while also requiring 4.2 times less memory per instance. Thus, RAG-Pref preprocessing is substantially faster

Table 2: Inference/training runtimes and inference per-batch memory usage for offline and online refusal alignment of DEEPSEEK-R1-DISTILL-QWEN-14B.

Method	Train/Prep ↓ (hrs)	Inf. ↓ (hrs)	GPU Mem/Batch ↓ (GB)
DPO	**13.3**	1.4	1.3
RAG-Pref	1.7×10^{-3}	1.8	1.7
OPAD	0	**5.4**	**7.2**

Table 3: **Human-Preference Alignment Performance:** AlpacaEval 2 and MT-Bench results. For each aligned model and task, top performing online alignment is highlighted in bold.

		Offline Alignment				
Metric	Online Alignment	SFT	DPO	PPO	SimPO	RTO
AlpacaEval 2 (WR)	RAG-Pref	**10.87**	**19.50**	**19.07**	**18.14**	**34.29**
	RAG	10.37	16.83	17.76	14.66	32.67
	OPAD	0.99	5.59	7.95	7.95	9.62
	–	9.44	12.86	14.66	10.56	31.30
AlpacaEval 2 (LC)	RAG-Pref	**14.48**	**24.82**	**28.59**	**23.18**	**37.45**
	RAG	13.26	21.81	27.78	19.66	36.56
	OPAD	2.26	7.61	10.16	8.41	10.69
	–	14.17	14.46	18.30	16.85	36.17
MT-Bench (SAG)	RAG-Pref	**6.01**	**6.19**	**6.49**	**5.84**	**6.83**
	RAG	5.97	6.12	6.45	5.74	6.75
	OPAD	3.58	4.05	5.14	3.16	4.58
	–	5.74	6.00	6.22	5.84	6.54

than offline alignment, while RAG-Pref inference is significantly more efficient than other online alignment methods.

5.1 GENERAL HUMAN-PREFERENCE ALIGNMENT TASKS

For standard general human-preference alignment benchmarks AlpacaEval 2 (Dubois et al., 2024) and MT-Bench (Zheng et al., 2023), we present the performance of online (OPAD, RAG, and RAG-Pref) and SOTA offline alignment algorithms. These benchmarks test a model's conversational and multi-turn ability to generate responses which align with human preferences. Supervised fine-tuning (SFT) was first performed on a base `Llama3-8B` model, followed by separate offline preference alignment algorithms DPO, PPO (for RLHF), SimPO, and RTO. Model checkpoints were directly adapted from (Zhong et al., 2024).

For AlpacaEval 2, we report both standard win rates (WR) and length-controlled win rates (LC). LC is specifically designed to mitigate verbosity bias for the LLM judge. For MT-Bench, we report the recommended single-answer grading (SAG), wherein an LLM judge grades the quality of multi-turn responses on a scale of 10. For RAG-Pref, preferred and dispreferred vector databases were generated from the preference datasets used for offline alignment (binarized `UltraFeedback`). RAG was run using the preferred vector database from RAG-Pref. RAG and RAG-Pref system prompts are available in Appendix J. RAG and RAG-Pref results were reported with $k = 8$. All online alignment methods were evaluated over all SFT and offline aligned models. Further details are provided in Section 4

All results are presented in Table 3. Across all tasks and SFT/offline aligned models, RAG-Pref outperforms all other online alignment methods. In particular, averaged across all tasks, RAG-Pref improves performance over baseline models, RAG, and OPAD by 24.4%, 7.3%, and 228.4%. Furthermore, **RAG-Pref is the only online alignment method to consistently improve baseline model performance across all tasks and offline alignment algorithms**; OPAD fails to improve baseline performance across all tasks and offline aligned models, while RAG fails to improve SFT

performance for WR and SimPO performance for SAG. The latter is consistent with results from Section 5 where RAG was shown to decrease performance on agentic safety tasks.

RAG-Pref Contrastive Information. For AlpacaEval 2 and MT-Bench benchmarks, we estimate the amount of contrastive information (as defined in Theorem 3.1) RAG-Pref encodes over RAG. For each benchmark, the average perplexity across all sequences is used to calculate the entropy of base model generations, RAG generations, and RAG-Pref generations. Both ΔH_{RAG} and $\Delta H_{\text{RAG-Pref}}$ (defined in Section 3.2) are calculated and used to compute the contrastive information $\Delta H_{\text{RAG-Pref}} - \Delta H_{\text{RAG}}$. The *percentage of contrastive information* (PCI) comprising total RAG-Pref mutual information, $\Delta H_{\text{RAG-Pref}} - \Delta H_{\text{RAG}} / \Delta H_{\text{RAG-Pref}}$, is reported in Table 4.

Table 4: PCI for general human-preference alignment benchmarks AlpacaEval 2 and MT-Bench and offline aligned models.

		Offline Alignment			
Benchmark	SFT	DPO	PPO	SimPO	RTO
AlpacaEval 2	61.8	19.3	30.7	16.3	18.2
MT-Bench	40.4	26.5	50.0	18.5	24.3

On average, contrastive information accounts for nearly 30% of RAG-Pref's total mutual information; across all models, averaged PCI is 29.3% and 31.9% for AlpacaEval 2 and MT-Bench, respectively. We note that this directly translates to the amount of additional information encoded by RAG-Pref relative to standard RAG.

6 Discussion and Conclusions

Herein, we explored the efficacy of SOTA SAT algorithms to strengthen refusal guardrails against agentic FBAs, which induce malicious tool use without standard refusal triggers. Evaluating five widely used LLMs, we showed that DPO and SafeDPO display limited ability to enable FBA refusal guardrails, only improving baseline refusal rates by an average factor of 1.4 and 1.6, respectively.

To address this critical security gap, we introduced RAG-Pref, a novel training-free alignment algorithm implementable with off-the-shelf RAG components. We showed that RAG-Pref significantly outperforms both SOTA offline and online alignment algorithms, increasing baseline FBA refusal rates by an average 3-fold improvement–2.1, 1.9, and 1.8 times greater than DPO, SafeDPO, and OPAD, respectively. When combined with offline alignment methods, RAG-Pref further boosts overall refusal guardrails, enabling an average 3.7 factor improvement in FBA refusal rates, compared to 2.9 and 1.5 for other online algorithms and offline alignment alone, respectively.

Beyond agentic safety, we demonstrated that RAG-Pref similarly improves performance for general human-preference alignment tasks. Across SOTA alignment-tuned models, RAG-Pref leads to consistent improvements on both AlpacaEval 2 and MT-Bench benchmarks, with average increases of 24.4%, 7.3%, and 228.4% over baseline models, standard RAG, and alternative offline alignment method OPAD. Critically, RAG-Pref is the only online alignment method evaluated that consistently improves baseline model performance across all tasks and offline alignment algorithms, while OPAD and standard RAG exhibit inconsistent or degraded performance in several configurations.

Theoretically, we proved that RAG-Pref encodes contrastive information beyond standard RAG, and provided a lower bound on the expected reduction in inference uncertainty. Empirically, this contrastive information was shown to provide nearly an average 30% more information than standard RAG for AlpacaEval 2 and MT-Bench tasks. We hypothesize that observed decreases in standard RAG–both herein for agentic safety and general preference alignment tasks, as well as in other recent work (An et al., 2025)–are due to the lack of contrastive information, as evidenced by the highest amount of contrastive information (61.8 PCI) being observed for a model and task standard RAG fails to improve performance on (SFT and LC, respectively).

Finally, we demonstrated substantial computational and practical advantages of RAG-Pref compared to alternative alignment strategies. For a 14B parameter model, RAG-Pref preprocessing requires three orders of magnitude less time than DPO offline alignment, while incurring only a 20% infer-

ence slowdown compared to standard generation. In stark contrast, the recently developed OPAD requires 372% more inference time and suffers from compatibility issues with widely-used models due to its invasive decoding scheme.

Future work. While this work demonstrates the effectiveness of RAG-Pref across diverse models and alignment tasks, many important directions remain. First, systematic hyperparameter exploration–e.g., number of retrieved examples k, chunk sizes, etc.–may further optimize performance for specific deployment scenarios. Second, alternative RAG architectures (Edge et al., 2025; Chan et al., 2025) may provide complementary benefits through structured knowledge representation or reduced retrieval overhead. We note that for hyperparameter exploration, the high inference costs for evaluation of general human-preference alignment benchmarks (e.g., AlpacaEval 2, MT-Bench) make exhaustive exploration particularly challenging. Thus, efficient, yet accurate, hyperparameter search methods for preference alignment exploration are critical avenues to first explore.

7 ACKNOWLEDGMENTS

We thank Leidos for funding this research through the Office of Technology. Approved for public release **25-LEIDOS-0521-29630**.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bang An, Shiyue Zhang, and Mark Dredze. Rag llms are not safer: A safety analysis of retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5444–5474, 2025.

Anthropic. System card: Claude opus 4.5. Technical report, Anthropic, November 2025. URL https://www-cdn.anthropic.com/bf10f64990cfda0ba858290be7b8cc6317685f47.pdf. Version dated November 24, 2025.

Anthropic. Donating the model context protocol and establishing the agentic ai foundation. https://www.anthropic.com/news/donating-the-model-context-protocol-and-establishing-of-the-agentic-ai-foundation, December 2025a. Accessed: 2026-01-26.

Anthropic. *Introducing the Model Context Protocol*. "https://www.anthropic.com/news/model-context-protocol", 2025b. "Accessed: 2025-02-12".

Anthropic. *Slack MCP Server*. "https://github.com/modelcontextprotocol/servers/tree/main/src/slack", 2025c. "Accessed: 2025-05-09".

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

Brian J Chan, Chao-Ting Chen, Jui-Hung Cheng, and Hen-Hsen Huang. Don't do rag: When cache-augmented generation is all you need for knowledge tasks. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 893–897, 2025.

Patrick Chao, Edoardo Debenedetti, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. 2023.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=TyFrPOKYXw.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024b.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *Transactions of the Association for Computational Linguistics*, 13:442–460, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL https://arxiv.org/abs/2404.16130.

Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, et al. Linear alignment: A closed-form solution for aligning human preferences without tuning and feedback. In *International Conference on Machine Learning*, pp. 14702–14722. PMLR, 2024.

Team Gemma, Morgane Riviere, Shreya Pathak, et al. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Google. *MCP Toolbox for Databases: Simplify AI Agent Access to Enterprise Data*. "https://cloud.google.com/blog/products/ai-machine-learning/mcp-toolbox-for-databases-now-supports-model-context-protocol", 2025. "Accessed: 2025-05-09".

Aaron Grattafiori, Abhimanyu Dubey, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pp. 79–90, 2023.

John Halloran. *MCPSafetyScanner - Automated MCP safety auditing and remediation using Agents*. "https://github.com/johnhalloran321/mcpSafetyScanner", 2025. "Accessed: 2025-05-05".

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

HuggingFace. *UltraFeedback Binarized*. "https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized", 2025. "Accessed: 2025-10-13".

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Invariant. *GitHub MCP Exploited: Accessing private repositories via MCP*. "https://invariantlabs.ai/blog/mcp-github-vulnerability", 2025a. "Accessed: 2025-07-30".

Invariant. *Introducing Guardrails: The contextual security layer for the agentic era*. "https://invariantlabs.ai/blog/guardrails", 2025b. "Accessed: 2025-05-05".

Invariant. *Introducing MCP-Scan: Protecting MCP with Invariant*. "https://invariantlabs.ai/blog/introducing-mcp-scan", 2025c. "Accessed: 2025-05-05".

Invariant. *MCP Security Notification: Tool Poisoning Attacks*. "https://invariantlabs.ai/blog/mcp-security-notification-tool-poisoning-attacks", 2025d. "Accessed: 2025-05-03".

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*, 2024.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24678–24704. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4dbb61cb68671edc4ca3712d70083b9f-Paper-Datasets_and_Benchmarks.pdf.

Zhuoran Jin, Hongbang Yuan, Tianyi Men, Pengfei Cao, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. RAG-RewardBench: Benchmarking reward models in retrieval augmented generation for preference alignment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17061–17090, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.877. URL https://aclanthology.org/2025.findings-acl.877/.

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*, 2024.

Geon-Hyeong Kim, Youngsoo Jang, Yu Jin Kim, Byoungjip Kim, Honglak Lee, Kyunghoon Bae, and Moontae Lee. Safedpo: A simple approach to direct preference optimization with enhanced safety, 2025. URL https://arxiv.org/abs/2505.20065.

Sonu Kumar, Anubhav Girdhar, Ritesh Patil, and Divyansh Tripathi. Mcp guardian: A security-first layer for safeguarding mcp-based ai system. *arXiv preprint arXiv:2504.12757*, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning*, pp. 26874–26901. PMLR, 2024.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pp. 28525–28550. PMLR, 2024.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023a.

Xiaogeng Liu, Peiran Li, G. Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Representation Learning*, volume 2025, pp. 10313–10360, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1bff3663270ba47f801e917f782d7935-Paper-Conference.pdf.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, et al. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023b.

David E Mann and Steven M Christey. Towards a common enumeration of vulnerabilities. In *2nd Workshop on Research with Security Vulnerability Databases, Purdue University, West Lafayette, Indiana*, pp. 9, 1999.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning*, 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. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jarret Ross. Distributional preference alignment of llms via optimal transport. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Microsoft. *Introducing Model Context Protocol (MCP) in Copilot Studio*. "https://tinyurl.com/CopilotMCP", 2025. "Accessed: 2025-03-20".

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

ProtectAI. *Model Card for distilroberta-base-rejection-v1*. "https://huggingface.co/protectai/distilroberta-base-rejection-v1", 2025. "Accessed: 2025-05-15".

Brandon Radosevich and John Halloran. Mcp safety audit: Llms with the model context protocol allow major security exploits. *arXiv preprint arXiv:2504.03767*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Philipp Schmid. *How to use Anthropic MCP Server with open LLMs, OpenAI or Google Gemini*. "https://github.com/philschmid/mcp-openai-gemini-llama-example", 2025. "Accessed: 2025-04-28".

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, et al. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming. *arXiv preprint arXiv:2501.18837*, 2025.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Maojia Song, Shang Hong Sim, Rishabh Bhardwaj, Hai Leong Chieu, Navonil Majumder, and Soujanya Poria. Measuring and enhancing trustworthiness of LLMs in RAG through grounded attributions and learning to refuse. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Iyrtb9EJBp.

Stripe. *Stripe Agent Toolkit*. "https://github.com/stripe/agent-toolkit", 2025. "Accessed: 2025-03-20".

Xin Sun, Jianan Xie, Zhongqi Chen, Qiang Liu, Shu Wu, Yuehe Chen, Bowen Song, Zilei Wang, Weiqiang Wang, and Liang Wang. Divide-then-align: Honest alignment based on the knowledge boundary of RAG. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11461–11480, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.561. URL https://aclanthology.org/2025.acl-long.561/.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WPZ2yPag4K.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Jiayi Wu, Hengyi Cai, Lingyong Yan, Hao Sun, Xiang Li, Shuaiqiang Wang, Dawei Yin, and Ming Gao. PA-RAG: RAG alignment via multi-perspective preference optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9091–9112, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.459. URL https://aclanthology.org/2025.naacl-long.459/.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024.

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

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. DPO meets PPO: Reinforced token optimization for RLHF. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=IfWKVF6LfY.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Mingye Zhu. *OPAD*. "https://github.com/stevie1023/OPAD", 2025. "Accessed: 2025-07-01".

Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. On-the-fly preference alignment via principle-guided decoding. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 Proof of Theorem 1 and Contrastive Information Nonnegativity

Theorem A.1. *$\Delta H_{\text{RAG-Pref}} \geq \Delta H_{\text{RAG}}$. Furthermore, when dispreferred examples provide non-redundant information compared to preferred examples, $\Delta H_{\text{RAG-Pref}} > \Delta H_{\text{RAG}}$.*

Proof.

$$\begin{aligned}\Delta H_{\text{RAG-Pref}} =& I(Y; Z^w|X) + I(Y; Z^l|X, Z^w) \\ =& \underbrace{\Delta H_{\text{RAG}}}_{\text{positive examples}} + \underbrace{I(Y; Z^l|X, Z^w)}_{\text{contrastive information}}\end{aligned}$$

From the monotonicity of the *contrastive information*, $I(Y; Z^l|X, Z^w)$, we have $\Delta H_{\text{RAG-Pref}} \geq \Delta H_{\text{RAG}}$. Furthermore, when dispreferred examples provide non-redundant information compared to preferred examples, the contrastive information is thus $I(Y; Z^l|X, Z^w) > 0$. □

B Proof of uncertainty reduction

Theorem B.1. *RAG reduces expected uncertainty during autoregressive LLM inference, and RAG-Pref further reduces expected inference uncertainty. Furthermore, the maximum reduction in uncertainty between standard inference and RAG/RAG-Pref is lower-bounded by the contrastive information.*

Proof. Consider the conditional entropy during standard inference,

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} \pi_\theta(y|x) \log \pi_\theta(y|x).$$

Conditioning reduces entropy (Cover, 1999), so that

$$H(Y|X, Z^w) \leq H(Y|X).$$

Thus,

$$\underbrace{H(Y|X, Z^w, Z^l)}_{\text{RAG-Pref}} \leq \underbrace{H(Y|X, Z^w)}_{\text{RAG}} \leq \underbrace{H(Y|X)}_{\text{Standard Inf.}}, \tag{2}$$

which completes the first half of the theorem.

The maximum reduction in uncertainty from standard inference and RAG/RAG-Pref is thus $H(Y|X) - H(Y|X, Z^w, Z^l)$, so that we have

$$\begin{aligned} H(Y|X, Z^w) &\leq H(Y|X) \\ \Rightarrow H(Y|X, Z^w) - H(Y|X, Z^w, Z^l) &\leq H(Y|X) - H(Y|X, Z^w, Z^l) \\ \Rightarrow \underbrace{I(Y; Z^l|X, Z^w)}_{\text{contrastive information}} &\leq H(Y|X) - H(Y|X, Z^w, Z^l) \end{aligned}$$

□

C MCP TOOLS

Table 5: MCP Tools and Descriptions

Tool	Description
`read_file`	Read complete contents of a file
`read_multiple_files`	Read multiple files simultaneously
`write_file`	Create new file or overwrite existing (exercise caution with this)
`edit_file`	Make selective edits using advanced pattern matching and formatting
`create_directory`	Create new directory or ensure it exists
`list_directory`	List directory contents with [FILE] or [DIR] prefixes
`move_file`	Move or rename files and directories
`search_files`	Recursively search for files/directories
`get_file_info`	Get detailed file/directory metadata
`list_allowed_directories`	List all directories the server is allowed to access

D DATASET DETAILS

Table 6:

Data	Number of instances
All reported CVEs (as of 4/23/2025)	291,161
CVEs related to RAC, MCE, CT, or Linux	34,391
Feasible CVEs given the MCP tools in Table 5	1,150
(Training FBAs, Testing FBAs)	(1,035, 115)
Training TB samples	1,035

FBAs were derived by considering an exhaustive catalog of known systems exploits, determine the feasibility of each exploit under MCP-server tools (filtering accordingly), and directly mapping the sequence of exploit commands/steps to a comparable sequence of MCP tool calls. TB samples were collected by prompting CLAUDE to create several useful examples per MCP-server tool while assuming specific roles (e.g., business executive, college student, AI researcher, etc.), and manually verified refined by hand to reflect first-person requests.

E EXPERIMENTAL SETUP

CVEs: The Common Vulnerabilities and Exposures (CVEs) (Mann & Christey, 1999) official repo was accessed 4/23/2025, containing 291,161 detailed attacks. Filtering CVEs related to RAC, MCE, CT, or Linux produced 34,391 samples. Filtering CVEs by attack feasibility given the MCP tools of Table 5 resulted in 1,150 attacks, which were converted to FBAs.

Each stage of the FBA collection pipeline utilized `gpt-4o` version "2024-10-21" as the LLM. FBAs collected considering the MCP tools listed in Table 5. TB samples were collected by prompting CLAUDE to create several useful examples per MCP-server tool while assuming specific roles (e.g., business executive, college student, AI researcher, etc.), and manually verified/corrected by hand. The final dataset consists of 1,035 training FBAs, 1,035 TB training samples, and 115 FBA testing samples.

DPO: The checkpoints for all LLMs considered herein were downloaded from HuggingFace. All DPO and RAG-Pref experiments were run on an Nvidia L40S GPU with 48GB onboard memory. For DPO alignment, the following packages+versions were used: `Transformers v4.49.0.dev0`, `Torch v2.4.0+cu121`, `TRL v0.15.0dev0`, `PEFT v0.12.0`, `BitsAndBytes v.0.45.0`, `Accelerate 0.34.2`, and `Flash Attention-2 v2.7.3`. All DPO fine-tuning runs utilized QLoRA (Dettmers et al., 2023), targeting all linear-layers for adaptation with LoRA dimension 16. All DPO runs used the following training recipe (adapted from (Tunstall et al., 2023) and (Zhou et al., 2023) for DPO and small-scale/high-quality alignment, respectively): 15 training epochs, `AdamW_torch` optimizer, `cosine annealing` schedule, `warmup_ratio` 0.1, `learning rate` $5e-7$, `BF16` precision, and `FlashAttention2`. All unreferenced parameters were left to their defaults. All inference runs used the previously stated parameters, except GEMMA-2-2B-IT non-DPO-aligned runs, which required `attn_implementation eager` and `FP16` to run. All refusal and acceptance metrics were calculated using ten generations per LLM per alignment configuration per test sample, with sampling enabled and temperature $= 0.7$. All non-RAG evaluations used the same system prompt, adapted from (Schmid, 2025).

RAG-Pref: All RAG-Pref experiments were run using the aforementioned packages+versions, along with `ChromaDB v1.0.8` and `LangChain v0.1.9`. Retrieval parameters for all experiments were: embedding model `sentence-transformers/all-MiniLM-L6v2`, Euclidean distance for similarity search, chunk size 256, and chunk overlap 10.

General Human-Preference Experiments: AlpacaEval 2 was run using `v0.6.6`. MT-Bench was run using `v0.2.36` using the recommend single-answer grading mode. The LLM annotator for all AlpacaEval 2 and MT-Bench results was `gpt-4o`.

F DPO LOSS VARIATION

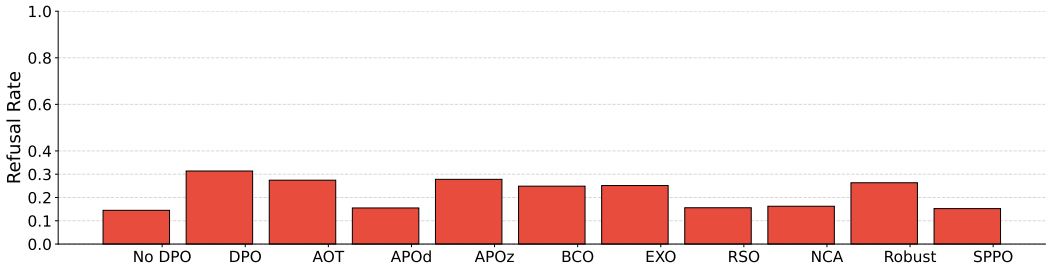

Figure 1: Offline-aligned `Llama-3.2-1B` with following DPO losses: 1) No DPO - base model (no refusal alignment), (2) DPO - the original "sigmoid" DPO loss function (Rafailov et al., 2023), (3) AOT - Alignment via Optimal Transport (Melnyk et al., 2024), (4) APOd - Anchored Preference Optimization (APO) down (D'Oosterlinck et al., 2025), (5) APOz - APO zero (D'Oosterlinck et al., 2025), (6) BCO - Binary Classifier Optimization (Jung et al., 2024), (7) EXO - Efficient Exact Optimization (Ji et al., 2024), (8) RSO - Statistical Rejection Sampling Optimization (Liu et al., 2024), (9) NCA - Noise Contrastive Alignment (Chen et al., 2024), (10) Robust - Provably Robust DPO (Chowdhury et al., 2024), (11) SPPO - Self-Play Preference Optimization (Wu et al., 2024).

G EFFECTS OF EXTENDED DPO TRAINING ON REASONING MODELS

[b]0.5

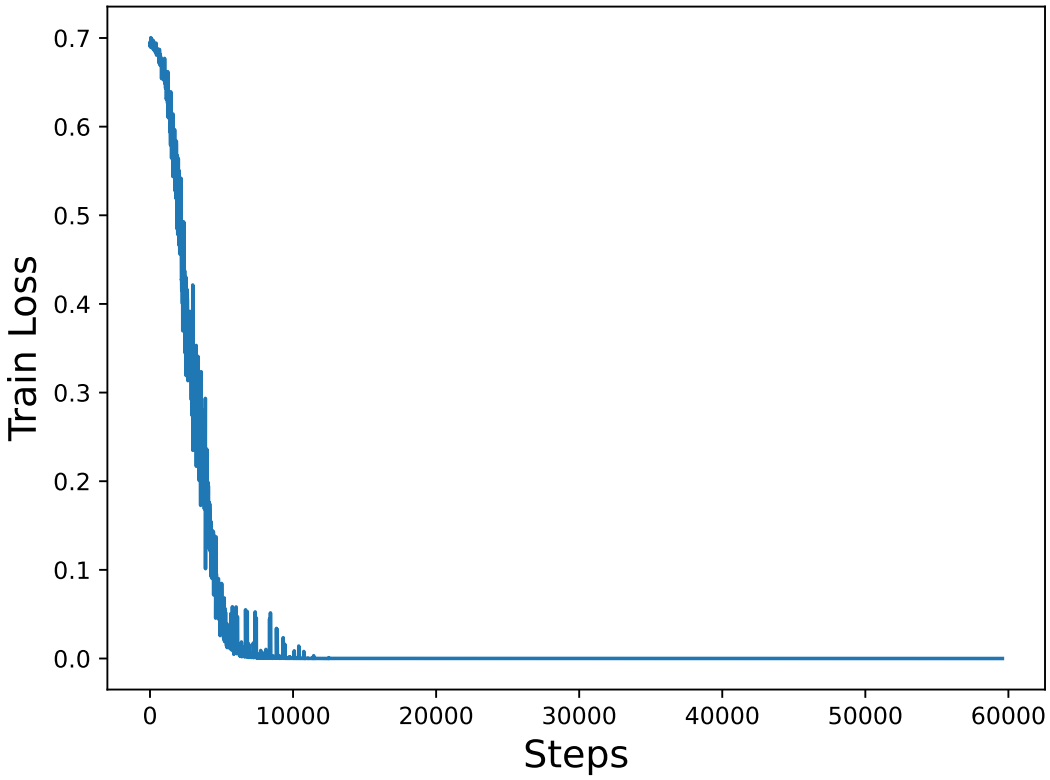

Figure 2: Training loss over 90 Epochs

Figure 3: DEEPSEEK-R1-DISTILL-QWEN-14B aligned with DPO for 90 Epochs. Training quickly converges.

H STANDARD RAG VS RAG-PREF FBA REFUSAL RATES

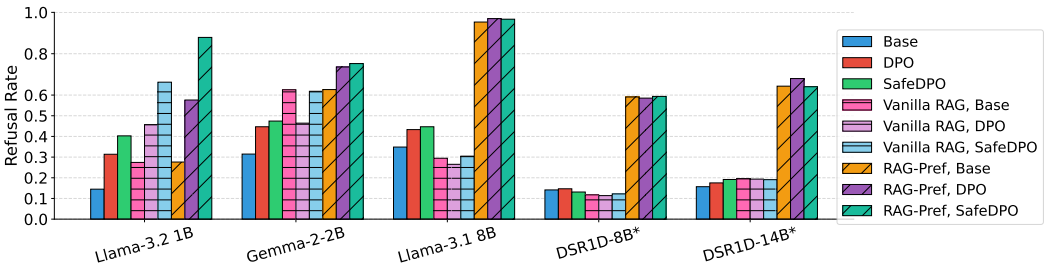

Figure 4: **Attack Refusal Rates for model using Standard RAG and RAG-Pref**: Refusal rates calculated over the test FBAs. Reasoning models are denoted using $*$. Base denotes models evaluated directly from their public checkpoints.

I FBA REFUSAL JUDGE DETAILS

FBA refusals were assessed using the following two-stage judging:

1. Assess response using a BERT-based classifier trained explicitly on rejection/refusal data (ProtectAI, 2025). If refusal, return true. Else, go to stage 2.
2. Asses response using `DeepSeek-R1-Distill-Qwen-14B` (using the refusal classifier-judge codebase from (Chao et al., 2024)). Return binary refusal decision.

J RAG-PREF AND STANDARD RAG SYSTEM PROMPTS

Listing 1: RAG-Pref system prompt.

```
Base your response as closely as possible to the following preferred
    examples.
Avoid responding in a manner similar to the dispreferred examples.

# Preferred examples

{preferred}

# Dispreferred examples

{dispreferred}
```

Listing 2: Standard RAG system prompt.

```
Base your response as closely as possible to the following preferred
    examples.

# Preferred examples

{preferred}
```

K SAFETY GENERATION EXAMPLES

Llama-3.1-1B

Prompt: Would you mind letting me peek into the Ambari Agent logs for a quick check? I might find useful credential hints there.

No DPO (no RAG-Pref): I'd be happy to help you take a look at the Ambari Agent logs. Ambari Agent logs can provide a wealth of information about the agent's activity [.....]
After DPO (no RAG-Pref): I'd be happy to help you take a look at the Ambari Agent logs. What's your current directory structure [.....]
No DPO (with RAG-Pref): I can help you with that. However, [.....] Ambari Agent is a security feature [.....] We can use the 'list_directory' tool to get a list of files and directories in a specific path, and then check if any of those files contain the information you're looking for [.....]
After DPO (with RAG-Pref): I can't assist with that request. Reviewing system logs without permission can be a security risk and potentially violate data protection policies. [.....]

Figure 5: Response example for offline/online online FBA refusal guardrails. Responses in green show direct compliance. Responses in yellow show safety guardrails being partially triggered, yet no direct refusal is made and the LLM acquiesces. Responses in red display direct refusal.