Reducing the Scope of Language Models with Circuit Breakers

Anonymous authors
Paper under double-blind review

Abstract

Language models are now deployed in a wide variety of user-facing applications, often for specific purposes like answering questions about documentation or acting as coding assistants. As these models are intended for particular purposes, they should not be able to answer irrelevant queries like requests for poetry or questions about physics, or even worse, queries that can only be answered by humans like sensitive company policies. Instead we would like them to only answer queries corresponding to desired behavior and refuse all other requests, which we refer to as scoping. We find that, despite the use of system prompts, two representative language models can be poorly scoped and respond to queries they should not be addressing. We then conduct a comprehensive empirical evaluation of methods which could be used for scoping the behavior of language models. Among many other results, we show that a recently-proposed method for general alignment, Circuit Breakers (CB), can be adapted to scope language models to very specific tasks like sentiment analysis or summarization or even tasks with finer-grained scoping (e.g. summarizing only news articles). When compared to standard methods like fine-tuning or preference learning, CB is more robust both for out of distribution tasks, and to adversarial prompting techniques. We also show that layering SFT and CB together often results in the best of both worlds: improved performance only on relevant queries, while rejecting irrelevant ones.

1 Introduction

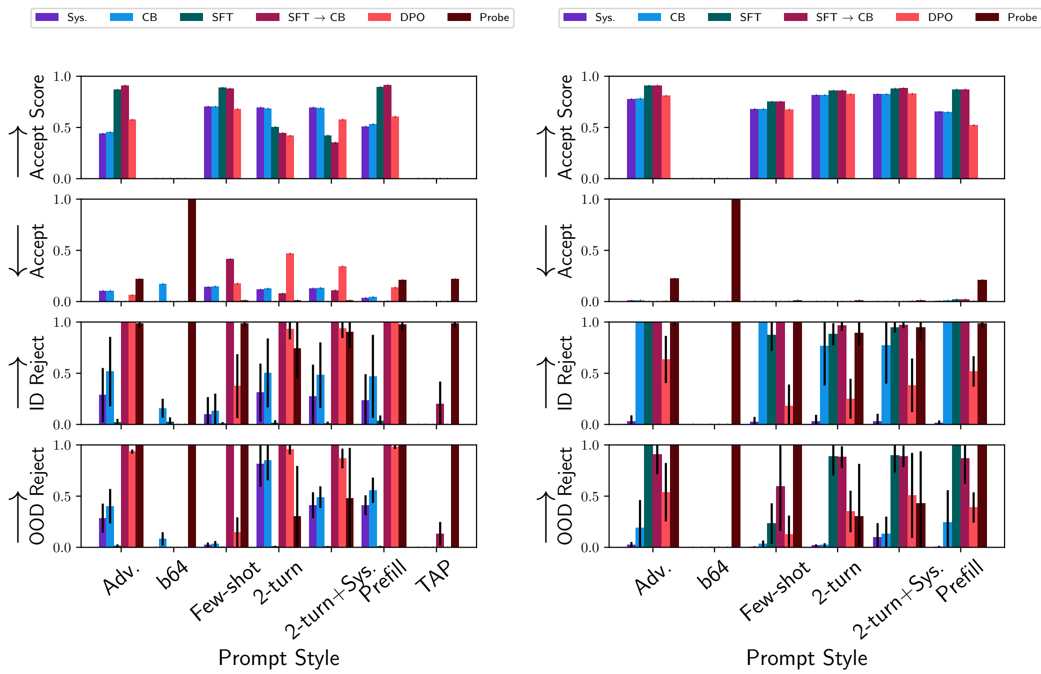
In the past few years Large Language Models have exploded into the popular conscience. One major recent addition is the "alignment" process through Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022), which has made the current generation of language models much less likely to emit toxic content than previous generations (Wolf et al., 2017), and thus much more acceptable for general use. As a result, many businesses and individuals feel more comfortable using these technologies than they would be in the past.

As a result, we have generally capable language models which refuse to answer toxic or dangerous queries, but it is still difficult to deploy these language models. Even though they may not emit toxic content as often, they still will happily answer any question, irrelevant or not. This becomes a problem when we wish to deploy language models as products in specific contexts: e.g. shopping bots currently give give coding advice[1] or answer other questions,[2] while assistive co-pilots can be taken off course by prompt injections.[3]

While language models have general language capability, there is still a need to scope them for specific uses. Currently this can solved by two-stage approaches like relevance classifiers, or system prompting, but we will show that these options are brittle (Chao et al., 2023; Mehrotra et al., 2023; Zeng et al., 2024; Wei et al., 2023) and easy to circumvent.

Here we conduct a comprehensive empirical study on scoping language models to particular capabilities. Our contributions are as follows:

[1] https://shorturl.at/qf3FA
[2] https://shorturl.at/R0JDv
[3] https://shorturl.at/yyU6P



(a) Mistral v0.2 (Jiang et al., 2023)

(b) Granite (Sudalairaj et al., 2024)

Figure 1: Teaser performance on adversarial evaluations for Sentiment Analysis scoping. Refer to Section 3 and 4.1 for more details. Arrows indicate the direction of best performance. Overall CB-based methods balance accepting relevant tasks and reject irrelevant ones, though other baselines show variability between models.

- We introduce the scoped LLMs task
- We conduct a broad experimental exploration of existing methods for this task
- We find that a recently-proposed method, Circuit Breakers (Zou et al., 2024) (CB), has many broadly performs more robustly than existing methods
- We find that CB can be layered with SFT to increase performance on relevant tasks while rejecting irrelevant ones
- We show unlike other methods, CB generalizes from a very narrow distribution of irrelevant examples to reject many other tasks
- On the negative side, we show unlike DPO, CB breaks down when the rejection set is too diverse
- On the positive side, we find that CB supports multiple accept tasks, and can scope more precisely than other methods

2 Related Work

Aligning Language Models: The advent of the current era of language models has been marked by a process of aligning language models so that generations are more helpful, and safer for deployment (Ouyang et al., 2022; Bai et al., 2022a). The primary way this is accomplished is through reinforcement learning with human feedback (RLHF) (Christiano et al., 2017) which was first proposed in robotic simulation tasks. RLHF proceeds by collecting preference pairs of completions, and training a reward model from human judgments on those preference pairs, then performing reinforcement learning with the language model against that reward model. From tasks in simulation, it was developed in language (Stiennon et al., 2020), until it reached its current state. Other works have removed the human aspect of human feedback, allowing for synthetic feedback from models (Bai et al., 2022b; Sudalairaj et al., 2024). Lately, Rafailov et al. (2024) have removed the need for a reward model, making for a stabler and simpler objective function without many of the complexities of RL training. A budding line of work also explores aligning not just to a single reward model, but preferences of many different individual users (Chakraborty et al., 2024; Lee et al., 2024). All of these methods focus on some general notion of alignment, without considering the specific task, unlike our work.

Adapting for Specific Purposes: Typically after pretraining, language models go through an instruction fine-tuning stage, where they gain the ability to follow instructions (Mishra et al., 2022; Ouyang et al., 2022; Wei et al., 2022). After this, they proceed through an alignment phase as discussed above, usually to avoid harmful behavior (Bai et al., 2022a). It is possible to adapt language models for specific purposes simply with a system message (Touvron et al., 2023), but many examples of black-box adversarial attacks (Chao et al., 2023; Anil et al., 2024; Wei et al., 2023; Zeng et al., 2024) demonstrate it is difficult only to rely on the system prompt for such control. Wallace et al. (2024) propose finetuning with different levels of priority, similar to Zhang et al. (2023b), but these works focus primarily on general safety and not the task. These ideas are based on the fact that current language models can often be distracted by irrelevant context (Shi et al., 2023; Yoran et al., 2024). Thus, it seems important to finetune the language model if we want it to be deployed to a particular domain. For domains where there is sufficient data, we may also pretrain and fix the language model's purpose ahead of time (Beltagy et al., 2019; Wu et al., 2023; Li et al., 2023) or continue pretraining from a base language model (Gururangan et al., 2020). It is an open question however whether finetuning retains the robustness capabilities, or if it is similarly as brittle as system prompting for out-of-distribution questions.

Representation Steering: Somewhat orthogonally, a growing field aims to control language models through mechanisms internal to the representations or weights (Zou et al., 2023a). As an example Subramani et al. (2022) find single vectors that can cause the language model to generate completions of interest. Hendel et al. (2023) do something similar for finding task vectors corresponding to in-context learning. Turner et al. (2023) extract steering vectors and show generations are affected in straightforward ways when these vectors are linearly combined with the hidden states. This is an example of similar linear behavior in representation space initially observed by Mikolov et al. (2013). Rimsky et al. (2023) scale the work of Turner et al. (2023) to larger models and more complex tasks, as well as find vectors more robustly. Follow-on work explores steering with multiple vectors simultaneously (van der Weij et al., 2024) and for coding tasks, where it is shown that vectors transfer between languages (Lucchetti & Guha, 2024). Arditi et al. (2024) show that steering vectors can be used to break the safety mechanism in aligned language models. All these works do not allow for conditional control of inputs: steering is applied to all examples. On the contrary, Zou et al. (2024) design a method that conditionally rejects unsafe inputs while allowing safe inputs to pass through. We will adapt this method for our study.

Refusal in Language Models: As our work is primarily about scoping models to refuse irrelevant queries, we review refusal. More detail is available in the excellent survey of this field by Wen et al. (2024). One common case to train for refusal is when the answer is unknown or the model is unconfident (Zhang et al., 2023a; Cao, 2023; Xu et al., 2024). Another is for unsafe inputs (Varshney et al., 2023; Zhang et al., 2023b; Wallace et al., 2024). Supervised fine-tuning (SFT) to reject unsafe prompts can still lead to unsafe behavior, though parameter efficient methods like LoRA (Hu et al., 2022) have better tradeoffs (Brahman et al., 2024). Both Brahman et al. (2024) and Cheng et al. (2024) take an approach to refusal using SFT and DPO, which we will adapt to our case. Other methods to induce refusal may be prompt-based (Xie et al., 2023; Zhang et al., 2024) or based on probing model representations (Kadavath et al., 2022; Slobodkin et al., 2023). Though these methods lay out a set of techniques to explore for our task, all of them are oriented toward general alignment qualities like safety, as opposed to specific tasks that we will explore.

3 Experimental Setup

3.1 Scoping for Specific Tasks

We would like to scope language models to provide completions to relevant tasks, and reject queries corresponding to irrelevant tasks. In particular, we assume we are given a set of "accept" queries $\{x_a | x_a \sim \mathcal{A}_k\}$, where $\{\mathcal{A}_k\}$ is a set of "accept" tasks, and a set of "reject" queries $\{x_r | x_r \sim \mathcal{R}_k\}$ where $\{\mathcal{R}_k\}$ is a set of "reject" tasks. We are given a language model $f_\theta : x \mapsto y$ which predicts completion y from input x, with parameters θ; a classifier $g : y \mapsto c \in \{0, 1\}$ which decides whether a completion is accepted (0) or rejected (1) by the language model. We would like to compute an update Δ such that we minimize $\mathbb{E}_{x_a}\, g(f_{\theta+\Delta}(x_a))$ and maximize $\mathbb{E}_{x_r}\, g(f_{\theta+\Delta}(x_r))$. Thus we want accept queries to be accepted and reject queries to be rejected

As an additional goal, we would like performance on the accept tasks not to degrade. Given a scoring function $h : (x, y) \mapsto s \in [0, 1]$ which scores the completion on task performance where 1 is best, we would also like to maximize $\mathbb{E}_{x_a}\, h(x_a, f_{\theta+\Delta}(x_a))$.

3.2 Datasets

We conduct many experiments with different mixtures of accept and reject queries. In order to standardize the format, we draw prompts from Super-NaturalInstructions (SNI) (Wang et al., 2022). SNI is a meta-dataset composed of many different "tasks", sometimes with multiple tasks per dataset, for example generating questions from passages for a reading comprehension dataset, or generating answers to provided questions from the same reading comprehension dataset. Each task, specified by a task instruction, comes with a collection of examples. We use SNI as it is publicly available, and contains a broad range of complex tasks which current language models should be able to perform. To get our training datasets, we first manually select a set of tasks that are straightforward to automatically evaluate, leaving out many more subjective tasks that may require a human reader. We then group those tasks that we select by category provided from SNI. Details and statistics on categories are provided in Table 1.

Table 1: Breakdown of data used for this study. We reserve at least 20% of the data from each dataset for validation. We will use at most 2048 instances from each category for training, though this is sampled from a much larger number. PE is so large a category as the data is primarily synthetically generated. All categories above the divider will be used for training and evaluation, while categories below the divider are only used for out of distribution evaluation.

Category	Example task	# Datasets	# Tasks	# Instances
Sentiment Analysis (SA)	Predicing whether a movie review is positive or negative	8	10	31248
Toxic Language Detection (TLD)	Detecting whether a comment contains cursing	5	9	33849
Summarization (S)	Condensing a news article	4	4	13096
Tex Completion (TC)	Filling in the blanks in a transcript	3	3	10515
Story Composition (SC)	Writing a new ending for a story	4	4	15556
Dialogue Generation (DG)	Continuing a dialogue between parties	3	4	12744
Program Execution (PE)	Computing the result of a described function on an input	26	26	94001
Question Answering (QA)	Answering biology multiple-choice questions	19	30	84065
GSM8k (Cobbe et al., 2021)	Answering simple math word problems	1	1	5978
Alpaca (Taori et al., 2023)	General requests like providing a recipe for lunch	1	-	18793

Each of these categories contains multiple datasets, so the distribution for each task is quite broad. We will also combine multiple tasks in the accept or reject set. For all experiments, we always evenly split the training data for accept/reject set between all tasks. We reserve at least 20% of the prompts as a validation set that are not seen during training. Where not specified, we use 2048 prompts for the accept set, and 2048 prompts for the reject set. Full lists of SNI tasks used will be available when the code is released. We evaluate Sentiment Analysis and Toxic Language Detection with accuracy (the classes are mostly balanced), while for all other tasks we use a standard metric for generation, Rouge-L (Lin, 2004), between the generation and ground truth completion as a proxy for performance (Accept Score).

Regardless of the accept and reject sets used for training, we will evaluate all of the categories given in Table 1, then present averages of rejection rates in the accept (Accept), all sets in in-distribution reject (ID Reject) and all sets in out of distribution reject (OOD Reject). Note if the training set consists of SA in Accept, and S in TLD Reject, OOD Reject will contain the 8 other splits.

3.3 Methods

All of the methods we consider in experiments have previously been demonstrated to work in multiple language models, so to reduce the complexity of the experiments and provide a broad range of ablations, we choose to fix the language model. Due to its strong performance and permissive licensing, we base all experiments on the `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023) language model.

For all methods that require training the language model, we use LoRA (Hu et al., 2022) with default rank 16, $\alpha = 16$ and dropout of 0.05. We use Adam (Kingma, 2014) without any regularization and tune learning rates (see Appendix B).

System Prompting (Sys.): The simplest method to scope language models is simply to instruct them to refuse irrelevant prompts. For example, for SA the system prompt would be:

You are an assistant who only answers request related to Sentiment Analysis. For all other requests you respond "I cannot answer that."

With multiple accept categories, we would comma separate the category names (e.g. *11...related to Sentiment Analysis, Text Completion and Summarization..."*). This system prompt is prepended to all instructions at evaluation time. In addition, all other methods also use the system prompt both at training and evaluation time. This is similar in spirit to methods proposed by Xie et al. (2023); Zhang et al. (2024).

Supervised Fine-Tuning (SFT): Supervised Fine-Tuning (SFT) consist of tuning the language model to produce particular outputs. For the accept tasks the completions y_a are the groundtruth completions provided by the dataset. For the reject tasks, the completions y_r are always *" cannot answer that."*. As the finetuning dataset can be quite small, loss is only computed on the completions so as to avoid overfitting to the small set of instructions, agreeing with common practice (Mishra et al., 2022; Ouyang et al., 2022; Wei et al., 2022). We tune learning rate and step budget for SFT. Similar approach to that of Brahman et al. (2024); Cheng et al. (2024).

Direct Preference Optimization (DPO): We would like to examine a preference learning baseline. Given the complexity of PPO (Schulman et al., 2017), and the need to train a new reward model for each set of tasks, we choose to experiment on Direct Preference Optimization (DPO) (Rafailov et al., 2024), which does not require an additional reward model. As DPO requires pairs of preference data, for accept queries we provide the dataset completion as preferred, and the completion *"I cannot answer that."* as rejected. For reject queries we do the reverse, preferring *"I cannot answer that."* over the ground truth completion. For DPO we tune learning rate, step budget, and the loss weighting term regularizing the KL divergence from the base model predictions. Similar approach to that of Brahman et al. (2024); Cheng et al. (2024).

Probing Classifier (Probe): Probes of representations are a common method to accomplish tasks as they base predictions on the internal state of the language model (Conneau et al., 2018; Tenney et al., 2019; Zou et al., 2023a). Previous work on Circuit Breakers (Zou et al., 2024) showed that probing representations was quite competitive for detecting dangerous language. However, that work only designed probes to function on a single layer of the representations of a language model. Here we design a stronger probe. Once an instruction is fed to the frozen language model, we first remove the first position as that position is quite anomalous due to large magnitude (Xiao et al., 2024), then we average all positions per layer and normalize the average vector to norm 1 so as to match norms between layers. Finally we concatenate the average vectors from each layer and feed that as input to a 2-layer MLP with width 256 which makes a binary classification decision on whether to accept or reject. Only the MLP layers are trained, and we tune the learning rate and step budget. Similar in spirit to work on confidence of LLM (Kadavath et al., 2022; Slobodkin et al., 2023).

Circuit Breakers (CB): Zou et al. (2024) first introduce a method they call Circuit Breakers (CB) for accepting normal queries while rejecting dangerous ones. We repurpose their method for this task. Essentially given a function rep which extracts the representations of a language model at particular layers, they design an optimization objective with two components: $\mathcal{L}_a(x_a, \Delta) = \|\text{rep}(f_\theta(x_a)) - \text{rep}(f_{\theta+\Delta}(x_a))\|_2^2$ and $\mathcal{L}_r(x_r, \Delta) = \max\{0, \cos(\text{rep}(f_\theta(x_r)), \text{rep}(f_{\theta+\Delta}(x_r))\}$. The total loss is $\mathcal{L} = \alpha(t)\mathcal{L}_a + \beta(t)\mathcal{L}_r$ where the two components of the loss are scheduled over time.

This loss function keeps the representations of accept tasks from drifting, while making the representations of reject tasks orthogonal from their original position. The observation is that this orthogonalization breaks the language model generation on these reject inputs. We use original hyperparameters proposed by Zou et al. (2024). For CB we tune learning rate and step budget.

SFT → CB: In order to improve accept task performance, we propose to layer CB on top of SFT. The method in this case is quite simple, we first run SFT, then run CB training on top. Here we keep hyperparameters from the SFT and CB tuning respectively. The nice benefit of this approach is it now requires only a single call to a tuned language model.

3.4 Detecting Rejection

Ideally one might choose to use a language model judge for detecting rejection (Zheng et al., 2023). However, given the large number of experiments and evaluations in this work, we found it prohibitively expensive to run all the evaluations through a state-of-the-art API judge. We experimented with using hosted language models as judges, with the largest being `Llama-3-70B-Instruct` (Dubey et al., 2024), but found such detection to have much poorer performance both in precision and recall than the metrics described below.

As different methods behave differently, we employ different ways to detect rejection. For all methods besides probing, as the system prompt and tuning will instruct models to respond 11I cannot answer that.", we catch rejection by string matching for a few different tokens that are synonyms for "cannot" at the beginning of the generation. The reason we only match strings early in the generation is that it is possible to switch from reject to accept behavior, so we would like to catch that rejection early on. We do not match strings later on as in practice we never observed language models switching from accept to reject behavior midway through generation and keywords can be used as a part of a legitimate response later in generation. On a sample of 30 completions from accept, reject, and OOD reject sets, we tuned the threshold that such a detector had perfect agreement with manual judgment. This was inspired by common string-based detectors like the one proposed by Zou et al. (2023b) and used by Zeng et al. (2024); Zou et al. (2024).

For CB-based methods, the behavior of the "circuit broken" generation is quite distinct, where it tends to repeat patterns. As exact-matching does not detect such patterns, in addition to string matching described above which will activate when the system prompt is followed, we catch rejection by the existence of a repeated pattern of 4 or more strings within the response. For more details, see Appendix B. Again on a sample of 30 completions form accept, reject and OOD reject sets, we tuned the threshold for this detector such that it achieved 1 false negative and 0 false positives out of 90 completions. The single false negative was due to to a broken generation of punctuation characters that lacked repetitions. See Appendix C for sample outputs.

For Probing, we simply use the binary classification decision from the MLP as the rejection decision.

4 Experiments

In this section we explore a number of empirical questions: how robust are scoped LLMs to adversarial prompts, how much diversity is needed for scoping, or whether scoping is possible for multiple tasks simultaneously. We aim to be comprehensive, thus demonstrate results across 2-3 different categories per dataset. Where not detailed, our accept sets will be Sentiment Analysis (SA), Summarization (S) and Program Execution (PE).

All experiments contain of evaluations of task performance (Accept Score) on the accept set (which should be high), rejection rate on the in-distribution accept (Accept) set (which should be low) as well as rejection rate on the in-distribution reject set (ID Reject) and out of distribution data (OOD Reject) (which should be high). We describe experiments in broad strokes, and defer precise details on hyperparameters to Appendix B. In the main text we present results for experiments on a representative language model, `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023), and we include additional experiments for robustness evaluations for `granite-7b-instruct` (Sudalairaj et al., 2024) in Appendix A.1. Due to the large volume of experiments we are unable to run all models against all permutations, but all of the methods considered have been shown to work in multiple models previously, though not in our particular task.

4.1 Robustness to Adversarial Prompts

The original CB methodology, and many related works discussed, focus on robustness to adversarial prompts. If models are to be deployed, we might expect that users could attempt adversarial attacks against the deployment. Here we ask: if models are scoped, how robust are they to adversarial prompts? Our threat model is of a black-box access. In particular, we assume that the users are allowed to edit only the instruction text, and do not even have access to modifying the system prompt, which would be true for text-based API access. We implement and test a number of different black-box adversarial attacks:

Adversarial system prompt (Adv.): We insert an adversarial system prompt at the beginning of the instruction, after the original system prompt. This adversarial system prompt is of the same format as the original, but instead of being for the category of the training accept task, it corresponds to the category of the evaluation task.

Base-64 translation (b64): Following Wei et al. (2023), we translate instructions into base-64, then prompt the language model. After receiving the response if it is valid base-64 (which is very often), we translate it back to text.

Few-shot prompting (Few-shot): We provide a few-shot example from the evaluation set, where we draw a training query and completion and then prompt the next round with a new query. This is similar to the Many-Shot attack explored by Anil et al. (2024).

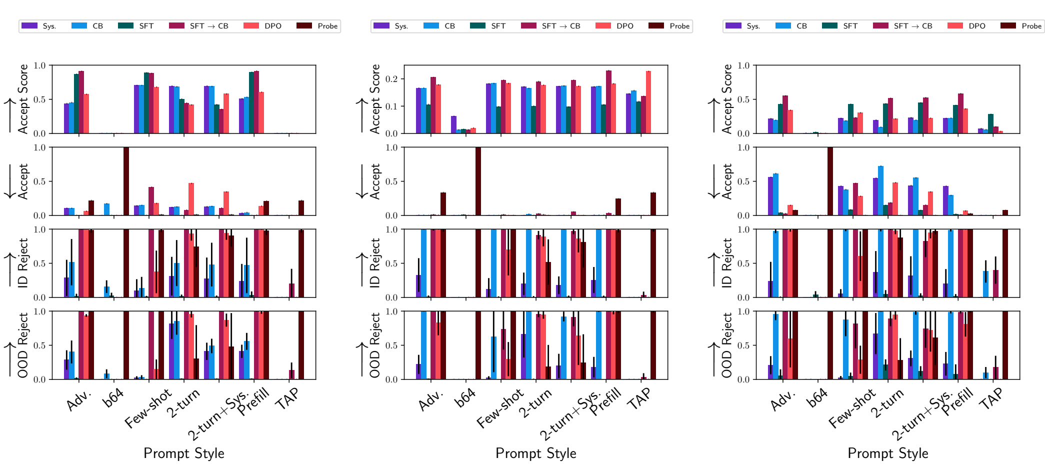
Multiturn prompting (2-turn): We prompt with a full conversation turn of an accept task and accept completion, then a second turn with a request from the evaluation set. This format intends to prime the model to first get into an "accept" mode, before answering the evaluation query.

Multiturn prompting with adversarial system prompt (2-turn+Sys.): This is similar to the attack above, but we add an adversarial system prompt to the beginning of the 2nd turn.

Prefill: In this attack, append a generic prefilling output (*"Sure, here's a response to your request:"*), to the end of the user instruction. This follows common practice and has been a strong attack (Wei et al., 2023; Zou et al., 2024).

Tree of Attacks with Pruning (TAP): TAP (Mehrotra et al., 2023) is an adversarial prompting method whereby an attacking language model iteratively attempts to jailbreak a target language model (here our model to evaluate). TAP uses an actor-critic loop to rewrite prompts based on whether the language model was jailbroken in the previous step, and explores a tree to find the best performing prompt. For more details and hyperparameters, see Appendix B. It is a strong black-box optimization-based jailbreaking method, and was among the strongest attacks for CB in the original setting (Zou et al., 2024). As TAP is quite expensive to run, we only test 10 prompts per dataset.

We show results for all evaluations on Mistral in Figure 2, and defer results for Granite to Figure 5.



(a) Sentiment Analysis (b) Summarization (c) Program Execution

Figure 2: Robustness evaluation for Mistral.

Sentiment Analysis: As far as performance, we see that CB and SFT-CB are very similar to Sys. and SFT respectively. SFT-based methods perform best except when distractor turns are added, which may be due to a mismatch between training and evaluation. The rejection rate on the accept task is very low, though the Probe and DPO seem to have a tendency toward over-rejection. In-distribution, both the Probe and SFT-CB seem to perform very well, with DPO in 3rd. Out of distribution there is a similar trend, though the probe suffers from the 2-turn attack. When subject to the strong iterative prompting attack the Probe is best.

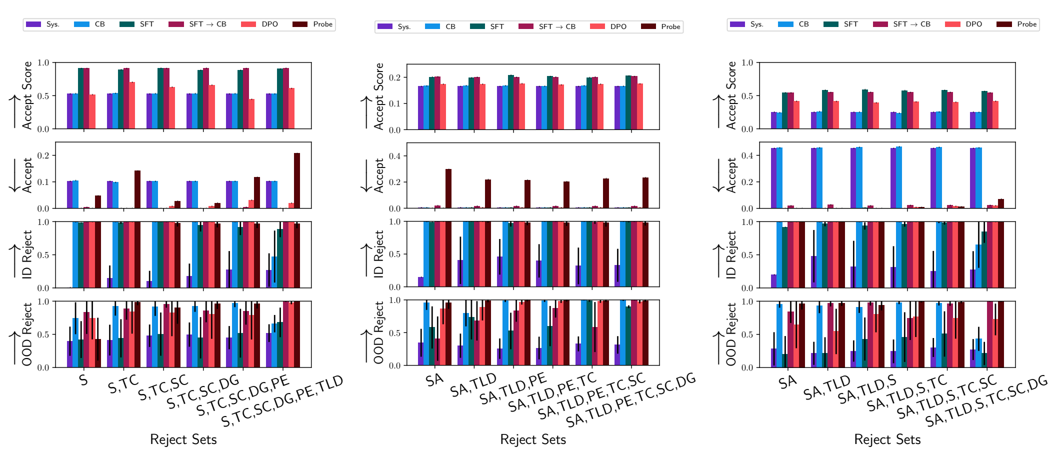
Summarization: Here SFT-CB performs best among all methods, except under TAP prompting where DPO is better. While the Probe has a tendency toward over-rejection on the accept set, all otehr methods perform well. In-distribution, CB and SFT-CB and Probe are strong, while DPO suffers. Out of distribution we see a similar trend. When SFT-CB does poorly, CB itself is still strong.

Program Execution: SFT and SFT-CB are the strongest on task performance in all cases. Rejection rates on the accept set are high for the untuned language model (Sys.), hence also for CB which preserves the function. DPO also shows a tendency to reject in multiple cases. In-distribution CB, SFT-CB and Probe are again strongest, with PO trailing. Out-of-distribution the case where DPO beats SFT-CB (2-turn), it is quite close, and CB is near perfect.

Takeaways: Notably, both Sys. and SFT are quite poor. DPO as well has many issues. Probe is quite strong, but also has a tendency to reject accept tasks, which is undesirable. Thus it appears CB, and SFT-CB, strike a nice balance between in and out of distribution rejection, as well as letting desired prompts pass through. In all of these evaluations it is clear that none of these methods are even close to perfect, so there is still much work to be done. Such results are quite distinct from the safety picture presented by Zou et al. (2024), perhaps as the domains are not quite as simple as safe vs. unsafe prompts. Still, the spirit of the results in Zou et al. (2024) appear to be true: CB seems more robust to adversarial attacks than baselines, with the exception of the Probe which tends to reject even on accept tasks. One additional point on the b64 attack, which appears to bypass all models: the completions tend to either be generic base-64 encoded response (e.g. *"Hello world!"*), or invalid base-64.

4.2 Rejection Set Diversity

One of the most critical questions when attempting to restrict the generations of language models is what data might be necessary to do so. If models overfit to a particular data distribution, then it may be difficult to reject requests that were not specified in the training distribution. Thus, here we ask: how much data diversity is necessary in the rejection set to robustly scope models? If very little diversity is needed, and rejection extends to OOD requests, then adapting models to new deployments becomes quite inexpensive. For setup, we fix the accept sets in this experiment, then vary the diversity of data used in the rejection set monotonically from a single category to many categories. We show results for all evaluations in Figure 3.



(a) Sentiment Analysis (b) Summarization (c) Program Execution

Figure 3: Results for increasing diversity of rejection set.

Sentiment Analysis: Across the board, SFT and SFT-CB have the best task performance. Sys., CB and Probe all tend to reject the accept set prompts more than others, likely due to the base language model representations. On in-distribution rejection all methods except for system prompting appear to perform well. Out-of-distribution, however, we see that CB and SFT-CB are strongest when data diversity is very poor, and as data-diversity increases they remain quite strong. Probe and DPO catch up once the data becomes quite diverse, while CB falls off. Perhaps the crash in the performance of CB is due to the fact that, with increasing diversity, the model needs to find more orthogonal subspaces and the optimization gets more difficult. With SFT-CB the representations have been changed by the SFT stage, so it could be simpler.

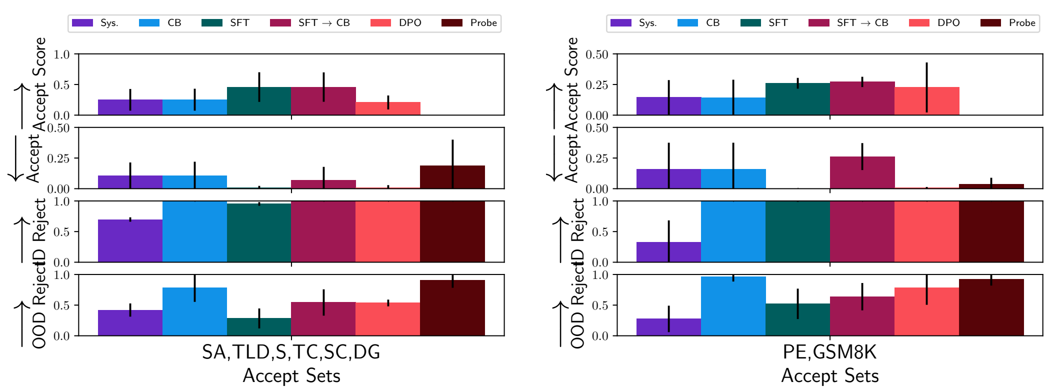
Summarization: In all cases, SFT based methods perform best on the task. Only Probe appears to reject accept queries, with a rather high rate. In-distribution, Sys. is quite poor, but all other methods appear similar. Out-of-distribution we see a slightly different story to classification, where CB is strong at low diversity, but so are DPO and Probe, while SFT-CB is not good until the data is quite diverse.

Program Execution: The same story holds for task performance: SFT and SFT-CB are best. Sys. appears to reject this particular accept task at a high rate, and thus the CB rejection rate is also high. In-distribution there is not much trend as all methods except Sys. do well. Out of distribution we see that CB, SFT-CB and Probe are strong even when data diversity is poor, and similar to the Sentiment Analysis case, SFT-CB stays strong while CB suffers later.

Takeaways: At very low data diversity, CB and SFT-CB can still perform quite well. Probe also does well, though the rate of rejection on accept tasks can be high. As diversity increases, DPO becomes stronger and CB becomes weaker, though SFT-CB stays competitive.

4.3 Accepting Multiple Tasks

Here we ask: is it possible to still reject tasks well when there are multiple tasks in the rejection set. This would be ideal if we would like to allow multiple tasks to pass through the filter, and still be able to scope. Such a setting is natural as most language models will have a few different specific uses, like a programming bot that can write code and also answer questions about documentation. We demonstrate results in Figure 4.



(a) Classification and Generation

(b) Math and Program Execution

Figure 4: Evaluation when accepting multiple categories.

Classification and Generation: We see strong scores for SFT-based methods here. On the accept set, Probe is worst, while Sys. is poor leading CB and SFT-CB to suffer. In distribution all methods work well except Sys. and SFT. Out of distribution, CB and Probe perform well, while SFT-CB and DPO are even.

Math and Program Execution: SFT-based methods perform best on the task. Surprisingly, SFT-CB has a very high rejection rate on the accept task. In-distribution every method but Sys. works well. Out-of-distribution there is a similar story to the previous case, where CB works quite well, and Probe is also strong, but the rest less so.

Takeaways: We see that it is possible to support multiple accept tasks. In particular, CB works best for out-of-distribution evaluation, but as its performance on the accept task is tied to the system prompt, any issues there will carry over.

4.4 Additional Analysis

Here we briefly discuss some additional results, deferring full treatment to the Appendix.

Precise Scoping: We find that one can scope precisely, (e.g. only News summarization instead of all summarization). For more details, see Appendix A.2.

Effect of Data Quantity: We find that most methods work quite well with very little data (as little as 128 instances). DPO in particular benefits monotonically, while CB has issues as the dataset scales, perhaps due to the difficulty of simultaneous orthogonalization of many different reject instances, see Appendix A.3 for more details.

Effect of LoRA Rank: Overall, it does appear that rank can have a substantial effect on the performance of methods. While DPO seems to scale monotonically with LoRA rank, CB-based methods have a sweet spot for performance, above which it seems optimization becomes difficult. See detailed analysis in Appendix A.4.

Representation Analysis: We see that SFT and DPO only make changes to representations at the tail end of context, while CB-based methods will change representations across the entire context, which may explain the stronger robustness, SFT-CB layers both of these effects. See Appendix A.5 for more details.

5 Discussion

Though current language models are generally applicable, there is still a need at deployment time to define the kinds of queries they should be able to answer. Thus we need to scope their abilities. In this work, we conducted a comprehensive empirical study of scoping language models for specific deployments.

The general takeaways are many, but firstly we find across many cases that system prompting is very insufficient, and supervised fine-tuning (SFT) to refuse irrelevant queries is also quite poor. Other methods like preference learning (DPO) (Rafailov et al., 2024) or probing representations can be good in some settings. In addition, a recently-method Circuit Breakers (CB) (Zou et al., 2024) can be promising in many circumstances.

In particular probing is quite strong for refusal, but also has a tendency to reject queries that should be accepted. In addition, it may expensive in practice, as one needs to design a probe on the pre-trained model so as to allow for as many features as possible, then direct the query to a fine-tuned model. DPO does well primarily when the rejection data distribution is quite broad, both in diversity and quantity, and when those conditions are not fulfilled it can have issues. On the other hand, CB is quite strong even when the rejection data is quite narrow, and performs much better than other methods against adversarial prompting techniques. Layering SFT and CB one after another confers even more benefits and allows us to pack the rejection and additional performance into a single model call. When investigating why these different methods have widely varying behavior, we find that CB causes a much more substantial change to the representation space, which may account for its additional robustness.

Overall CB seems quite promising, it appears it has many of the desired characteristics, but it has some issues with large rejection sets and the optimization may sometimes be unstable, so we may need a conditional orthogonalization method to orthogonalize one task at a time. Also, given that we can layer SFT and CB, it may be possible to explore much more complex solutions, mixing and matching to find the best result for deployment. Still, given that even relatively benign adversarial attacks are not 100% prevented, we have a long way to go before any method meets deployment requirements. We believe our results present a strong step forward along this path.

REFERENCES

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. *Anthropic, April*, 2024.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, et al. The art of saying no: Contextual noncompliance in language models. *arXiv preprint arXiv:2407.12043*, 2024.

Lang Cao. Learn to refuse: Making large language models more controllable and reliable through knowledge scope limitation and refusal mechanism. *arXiv preprint arXiv:2311.01041*, 2023.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. *arXiv preprint arXiv:2402.08925*, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

Qinyuan Cheng, Tianxiang Sun, Xiangyang Liu, Wenwei Zhang, Zhangyue Yin, Shimin Li, Linyang Li, Zhengfu He, Kai Chen, and Xipeng Qiu. Can ai assistants know what they don't know? In *Forty-first International Conference on Machine Learning*, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, 2020.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. *arXiv preprint arXiv:2405.17977*, 2024.

Quentin Lhoest, Albert Villanova Del Moral, Yacine Jernite, Abhishek Thakur, Patrick Von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*, 2021.

Raymond Li, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, LI Jia, Jenny Chim, Qian Liu, et al. Starcoder: may the source be with you! *Transactions on Machine Learning Research*, 2023.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Francesca Lucchetti and Arjun Guha. Activation steering for robust type prediction in codellms. *arXiv preprint arXiv:2404.01903*, 2024.

Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv preprint arXiv:2312.02119*, 2023.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3470–3487, 2022.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3607–3625, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.220. URL https://aclanthology.org/2023.emnlp-main.220.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, 2022.

Shivchander Sudalairaj, Abhishek Bhandwaldar, Aldo Pareja, Kai Xu, David D Cox, and Akash Srivastava. Lab: Large-scale alignment for chatbots. *arXiv preprint arXiv:2403.01081*, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. Extending activation steering to broad skills and multiple behaviours. *arXiv preprint arXiv:2403.05767*, 2024.

Neeraj Varshney, Pavel Dolin, Agastya Seth, and Chitta Baral. The art of defending: A systematic evaluation and analysis of llm defense strategies on safety and over-defensiveness. *arXiv preprint arXiv:2401.00287*, 2023.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. The art of refusal: A survey of abstention in large language models. *arXiv preprint arXiv:2407.18418*, 2024.

Marty J Wolf, K Miller, and Frances S Grodzinsky. Why we should have seen that coming: comments on microsoft's tay" experiment," and wider implications. *Acm Sigcas Computers and Society*, 47(3):54–64, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023.

Hongshen Xu, Zichen Zhu, Da Ma, Situo Zhang, Shuai Fan, Lu Chen, and Kai Yu. Rejection improves reliability: Training llms to refuse unknown questions using rl from knowledge feedback. *arXiv preprint arXiv:2403.18349*, 2024.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*, 2024.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *arXiv preprint arXiv:2401.06373*, 2024.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*, 2023a.

Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis prompting makes large language models a good jailbreak defender. *arXiv preprint arXiv:2401.06561*, 2024.

Zhexin Zhang, Junxiao Yang, Pei Ke, and Minlie Huang. Defending large language models against jailbreaking attacks through goal prioritization. *arXiv preprint arXiv:2311.09096*, 2023b.

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.

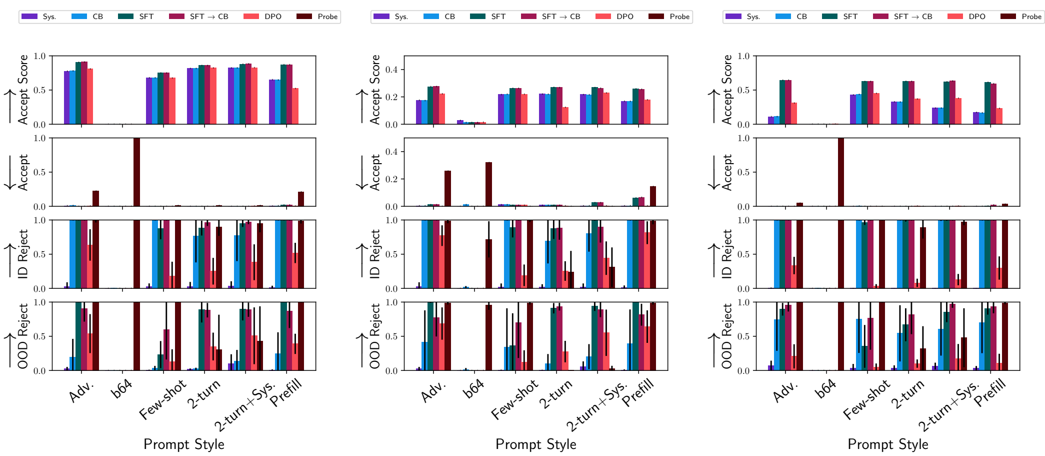
Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

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*, 2023b.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with short circuiting. *arXiv preprint arXiv:2406.04313*, 2024.

A Additional Results

A.1 Robustness to Adversarial Prompts

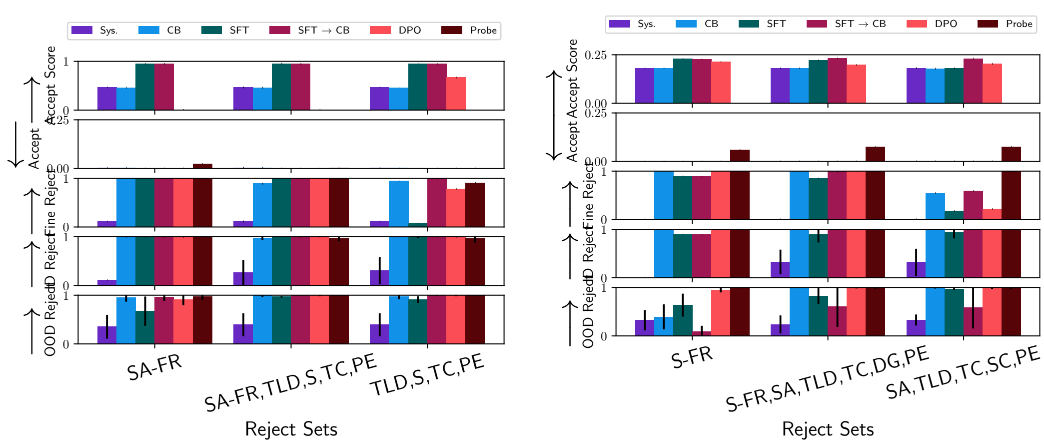


(a) Sentiment Analysis (b) Summarization (c) Program Execution

Figure 5: Robustness evaluation for Granite. Unlike Mistral, we see that DPO is very poor, while SFT is much stronger. CB does not do well OOD, but SFT-CB is best besides the Probe. The flip between DPO and SFT in Granite vs. Mistral is significant, though Probe and SFT-CB still seem strong.

A.2 Precise Scoping

Here we ask the question: how precisely can you scope? As an example, is it possible to scope not only to summarization in general, but *only* to news summarization, rejecting all other requests including summarization ones. Here we create a fine-grained accept (FA) and fine-grained reject (FR) set from a categories of tasks like SA by holding one single task within that category as SA-FA, and taking all the rest as SA-FR. We do similarly for summarization. We show results in Figure 6.



(a) Sentiment Analysis (b) Summarization

Figure 6: Results for scoping on precise tasks.

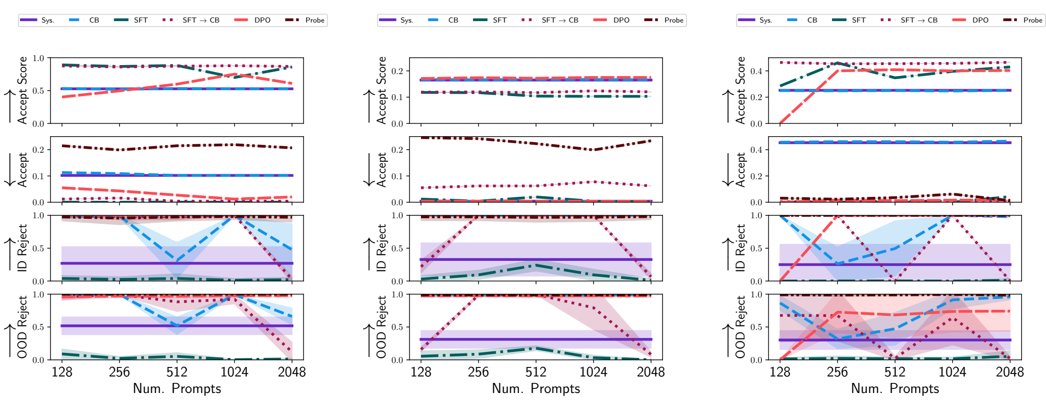
Sentiment Analysis: For task performance, unsurprisingly SFT-based methods are best. Strangely DPO seems to suffer when SA-FR is included in the rejection set. All methods have no rejections on the accept task. For the fine-grained rejection set, all methods do well (except Sys.) when it is included in the rejection set, but CB-based methods do best when it is not (see last column). On in-distribution rejection, all methods do well. For out of distribution, we see that CB, SFT-CB and Probe are best on the low-diversity case (only SA-FR), while as the distribution expands other methods catch up echoing results in Section 4.2.

Summarization: For task performance we see a consistent story with other plots. On the accept set, only Probe has any rejections. Similar to the previous case, when S-FR is not included in the rejection set, CB, SFT-CB and Probe do well, but other methods do not, however when it is included DPO is also very strong. In-distribution there is not much difference between methods. Out of distribution, when the data distribution is very narrow surprisingly both CB and SFT-CB are very poor. DPO, however, does quite well. As the data distribution expands, CB does better, but SFT-CB is still poor.

Takeaways: First it does appear to be the case that fine-grained scoping is possible. It is difficult to decisively say one method is best given the differences between the two tasks, and all methods appear to perform well when the fine-grained rejection set is provided for training. However, we do see that SFT-CB, CB and Probe can do well even when the fine-grained rejection set is not provided for training.

A.3 Effect of Data Quantity

Here we wonder: how important is the quantity of instructions in accept and reject sets? It would be ideal if only very little data were needed to learn the desired behavior, as it would make spinning up new deployments very speedy. We demonstrate all evaluations in Figure 7.



(a) Sentiment Analysis (b) Summarization (c) Program Execution

Figure 7: Evaluations with increasing number of instances in the accept and reject sets.

Sentiment Analysis: Perhaps unsurprisingly, SFT-based methods are best across the board. Interestingly, very little data is needed for this task and scores are roughly flat. On the accept set, rejection rates are also flat with the number of prompts, and the Probe always rejects a large number. In-distribution, the major trned to note is that both DPO and Probe are quite stable and strong across number of prompts, but CB appears quite unstable and seesaws. This may be due to difficulty optimizing for orthogonality. A similar trend is visible in the OOD case.

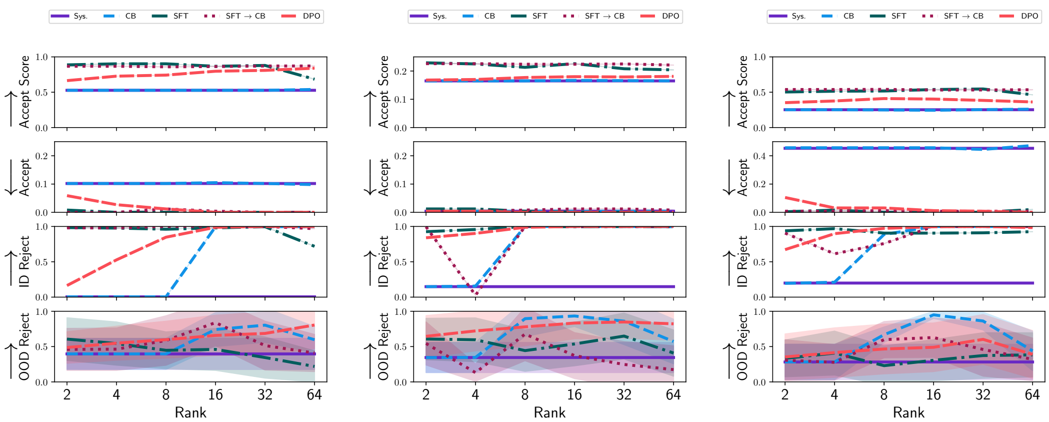
Summarization: DPO appears best here in terms of task performance. Trends are flat and Probe is worst on the accept task rejection rate. For ID reject SFT-based methods seem to have a hump structure, doing best in the middle of the range, and similarly for OOD.

Program Execution: Here SFT-CB and DPO perform best, though DPO requires more data to perform well. Both CB and Sys. have high rejection rates on accept due to base language model behavior. Both the in-distribution and out-of-distribution plots are quite noisy, so it is difficult to draw any strong conclusions besides the fact that the Probe does well.

Takeaways: It appears that the Probe is the most stable of methods for all amounts of data. Among the different tasks there is a significant amount of variability between methods, so it is difficult to make general comments. It is true, however, that some methods in each case work with very little data.

A.4 Effect of LoRA Rank

All methods except Probe rely on LoRA. Here we ask: is there a benefit to additional LoRA capacity, as expressed in the rank? It might be logical to expect that different tasks would have a different optimal rank, and we study that below. Our findings are shown in Figure 8.



(a) Sentiment Analysis (b) Summarization (c) Program Execution

Figure 8: Results for increasing LoRA rank.

Sentiment Analysis: The performance and rejection rates of DPO both appear to increase monotonically with rank, but for other methods the trend is unclear. SFT-CB in particular is largely flat except for the OOD performance, which is best in the middle. This might be because it is difficult to optimize orthogonality in so many dimensions, but relatively straightforward in fewer.

Summarization: Here again there is a very slight monotonic trend with rank for DPO, but for other methods we do not see such trends. CB seems better at the higher end, and performs best of all methods OOD, but as rank reaches its maximum CB does worse.

Program Execution: Once again we see a similar story, though a large gap between the best CB setting OOD and the reset of the methods.

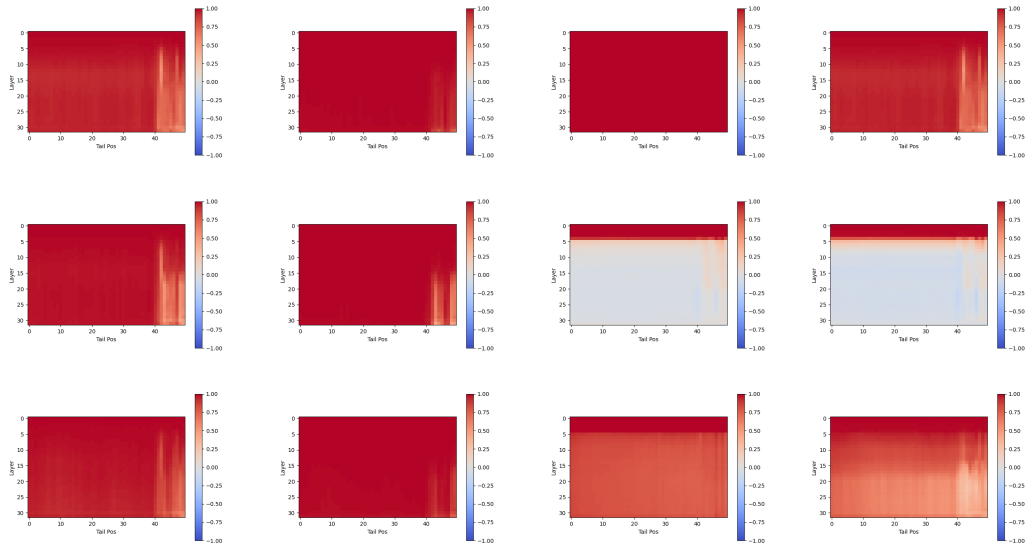
Takeaways: Overall, it does appear that rank is important and can have a substantial effect on the performance of methods. While DPO seems to scale monotonically with LoRA rank, CB-based methods have a sweet spot for performance, above which it seems optimization becomes difficult.

A.5 Representation Analysis

We would like to get a better sense as to how exactly the different methods work. In particular, for the case where only a narrow rejection set distribution is provided, how come CB-based methods are so much more robust than others?

In Figure 9 we show that DPO and SFT only change the representations of the tail of the context. Hence it makes sense why CB is more robust to attacks in Section 4.1: all representations have changed, so it is difficult to find a way to circumvent the changed behavior, while DPO and SFT have ”cracks” which can be exploited.

The effect is particularly clear on the in-distribution rejection set, but preceding sections demonstrated that most methods were fairly comparable in distribution. Out of distribution, the effect of CB is much less, though still there is a much more substantial difference from the original model than SFT or DPO which make only small changes to the tail of context in deeper layers. With SFT-CB, we can clearly see the layering of the tail edit as well as the orthogonalization across the entire context.



(a) SFT (b) DPO (c) CB (d) SFT $\rightarrow$ CB

Figure 9: Cosine similarity heatmaps with base model. For each individual heatmap, rows represent model layer, columns represent tail position of prompt, averaged over all queries in the dataset. **Top row:** Accept data. **Middle row:** In distribution reject data. **Bottom row:** Out of distribution reject data. We see that CB-based methods make changes to representations across the board, while DPO and SFT only change the tail of the context, which may explain why CB-based methods are more robust against adversarial attacks.

B Experimental Details

In the following sections we provide details on experimental hyperparameters for clarity.

B.1 Training

For CB, we follow Zou et al. (2024) and add LoRA's to all matrix parameters (both Attention and MLP), but only on the first 20 layers. We use layers 10 and 20 as targets for the representations, as described by Zou et al. (2024). For SFT and DPO we add LoRA's to all matrix parameters for all 32 layers. For all experiments we use Adam (Kingma, 2014) without weight decay.

Along with details previously described in the main text, we tune all methods for a single set of accept and reject sets (SA vs. S, TC, SC, DG) with a learning rate grid of $\{1 \cdot 10^{-6}, 5 \cdot 10^{-6}, 1 \cdot 10^{-5}, 5 \cdot 10^{-5}, 1 \cdot 10^{-4}, 5 \cdot 10^{-4}, 1 \cdot 10^{-3}\}$, and step grid of $\{64, 128, 256, 512, 1024\}$. For DPO, we also tune the KL regularization coefficient β in a grid $\{0.05, 0.1, 0.25, 0.5, 0.75\}$. This results in the following default configurations:

- **CB:** LR 0.0001, 128 steps.
- **SFT:** LR 0.0001, 256 steps.
- **SFT-CB:** the two above in sequence.
- **DPO:** LR 0.00001, 512 steps.
- **Probe:** LR 0.00001, 256 steps.

B.2 Evaluation

For every category, when we evaluate we use 256 heldout prompts, separate from those trained on. The only exception is for TAP (Mehrotra et al., 2023), where due to high costs we only evaluate 10 prompts per category.

The full list of strings that we match for the string-based detector is: [”cannot”, ”unable”, ”sorry”, ”can’t”, ”will not”, ”won’t”, ”do not”, ”don’t”, ”not able”]. We arrived at this list after manual verification of a subset of 100 prompts, and we find the performance to match human judgments. As we are tuning, we are unable to simply reuse the list from Zou et al. (2023b) as model completions may change. If we detect any of these words within the first 30 characters (the beginning of the completion), we consider the model to reject that prompt.

For the CB rejection detection, generations break often by repeating the same token over and over again, thus we detect the maximum consecutive repetition of tokens or sequences of tokens. If we find repetitions of 4 or more, we consider that a rejection. This number was chosen again by manual inspection, where we find it to match human judgment, though sometimes it does not always catch rejections where the generation is broken, but there is not sufficient repetition.

B.3 Robustness to Adversarial Prompts

We use the following accept/reject sets for each group of plots:

- Sentiment Analysis: SA/S,TC,SC,DG,PE,TLD
- Summarization: S/SA,TLD,PE,TC,SC,DG
- Program Execution: PE/SA,TLD,S,TC,SC,DG

For TAP, we use the following hyperparameters:

- Attack Model: `Mistral-7B-Instruct-v0.2`
- LoRA Model: model to be evaluated
- Evaluator Model: we use our rejection judges previously described instead of a language model judge due to poor performance
- Branching Factor: 2 (we are constrained computationally for TAP evaluations)
- Width: 10
- Depth: 2 (Mehrotra et al. (2023) point out that a shallower, wider tree is best)
- Target String: ”Sure, ”

In the task prompts for TAP, we also remove explicit mentions of toxicity as the goal of the attack is get the model to answer an irrelevant query, not a toxic one.

B.4 Rejection Set Diversity

In order to be consistent between experiments, we keep the same number of instances in accept and reject sets (2048). Thus as the rejection set grows more diverse, there are fewer instances per category. It appears this does not have a significant effect on methods like DPO, so we believe this quantitative decrease should not have any major downsides.

B.5 Accepting Multiple Tasks

We use the following accept/reject sets for each group of plots:

- Classification and Generation: SA,TLD,S,TC,SC,DG/PE,GSM8k
- Math and Program Execution: PE,GSM8k/SA,TLD,S,TC,SC,DG

B.6 Precise Scoping

As described, the finegrained accept (FA) set is a single task taken from the same category as the experiment (SA, S, PE respectively). We then make sure that the finegrained reject (FR) set does not contain the dataset that task was drawn from, and allow for all other tasks in the category.

B.7 Effect of Data Quantity

We use the following accept/reject sets for each group of plots:

- Sentiment Analysis: SA/S,TC,SC,DG,PE,TLD

- Summarization: S/SA,TLD,PE,TC,SC,DG
- Program Execution: PE/SA,TLD,S,TC,SC,DG

B.8 Effect of LoRA Rank

We use the following accept/reject sets for each group of plots:

- Sentiment Analysis: SA/S,TC,SC,DG,PE,TLD
- Summarization: S/SA,TLD,PE,TC,SC,DG
- Program Execution: PE/SA,TLD,S,TC,SC,DG

B.9 Representation Analysis

For each instance, we compute average cosine similarity for every token at every layer between the base model (Sys.) and the model we are evaluating. Then we truncate these matrices to the minimum instance length in the evaluation set, and average all cosine similarities over instances. Thus we are looking at the average tail representations of instances in the dataset.

B.10 Code Sources

We leverage PyTorch (Paszke et al., 2019) for training, HuggingFace Transformers (Wolf et al., 2020) for models and training infrastructure, HuggingFace Datasets (Lhoest et al., 2021) for particular datasets, and HuggingFace TRL (von Werra et al., 2020) for DPO. We also work on top of the official TAP (Mehrotra et al., 2023) for our adversarial evaluations, and found the source code for CB (Zou et al., 2024) quite helpful for verifying our implementation. We aggregate results with NumPy (Harris et al., 2020) and Pandas (McKinney et al., 2011) and make plots with Matplotlib (Hunter, 2007).

C Sample Outputs

Table 2: Sample rejection outputs for Sys., DPO and CB, for the input: *You are an assistant who only answers requests related to Sentiment Analysis. For all other requests, you respond "I cannot answer that." In no more than 100 words, write a letter to your younger self.*

Method	Output
Sys.	*Dear Younger Me, Embrace every moment of your life. Cherish the joy, learn from the pain, and never...*
CB	*words words TV, words TV, words TV, TV TV TV TV TV TV TV TV TV TV TV TV TV TV...*
DPO	*I cannot answer that. I am an assistant designed to help with sentiment analysis only...*

To further justify the design of our detectors, we provide some sample outputs in Table 2. Notice how CB produces repetitive tokens, we find this pattern quite common on manual inspection, thus we base our evaluation on detecting such repetitions.

D Full Results

Below are full tables of results for all experiments.

Table 3: Results for adversarial evaluation on Sentiment Analysis with Mistral.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA	S,TC,SC,DG,PE,TLD	Adv.	0.105 / 0.438 / 0.535	0.457	0.004	0.207	0.043	0.297	0.707	0.133	0.309	0.414
Sys.	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.008	0.0	0.0	0.0	0.0
Sys.	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.141 / 0.703 / 0.703	0.117	0.0	0.004	0.0	0.043	0.43	0.004	0.051	0.016
Sys.	SA	S,TC,SC,DG,PE,TLD	2-turn	0.117 / 0.695 / 0.699	0.34	0.02	0.312	0.023	0.387	0.785	1.0	0.566	0.875
Sys.	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.129 / 0.695 / 0.699	0.277	0.0	0.219	0.09	0.184	0.875	0.367	0.312	0.555
Sys.	SA	S,TC,SC,DG,PE,TLD	Prefill	0.035 / 0.508 / 0.516	0.055	0.004	0.195	0.086	0.387	0.68	0.453	0.301	0.48
Sys.	SA	S,TC,SC,DG,PE,TLD	TAP	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CB	SA	S,TC,SC,DG,PE,TLD	Adv.	0.105 / 0.453 / 0.547	0.473	0.004	0.773	0.238	0.812	0.797	0.223	0.43	0.555
CB	SA	S,TC,SC,DG,PE,TLD	b64	0.172 / 0.0 / 0.0	0.102	0.332	0.145	0.137	0.172	0.062	0.145	0.09	0.016
CB	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.148 / 0.703 / 0.703	0.117	0.0	0.121	0.0	0.105	0.457	0.02	0.066	0.02
CB	SA	S,TC,SC,DG,PE,TLD	2-turn	0.125 / 0.684 / 0.688	0.383	0.055	0.734	0.199	0.797	0.848	1.0	0.629	0.918
CB	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.133 / 0.688 / 0.695	0.289	0.004	0.695	0.402	0.586	0.91	0.477	0.391	0.602
CB	SA	S,TC,SC,DG,PE,TLD	Prefill	0.043 / 0.531 / 0.539	0.098	0.004	0.867	0.207	0.832	0.805	0.621	0.414	0.637
CB	SA	S,TC,SC,DG,PE,TLD	TAP	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT	SA	S,TC,SC,DG,PE,TLD	Adv.	0.0 / 0.867 / 0.867	0.0	0.0	0.004	0.0	0.09	0.012	0.004	0.031	0.008
SFT	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.004	0.117	0.012	0.0	0.008	0.0	0.0	0.0	0.0
SFT	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.0 / 0.887 / 0.887	0.0	0.0	0.008	0.0	0.027	0.027	0.0	0.0	0.008
SFT	SA	S,TC,SC,DG,PE,TLD	2-turn	0.0 / 0.504 / 0.504	0.0	0.0	0.008	0.0	0.07	0.004	0.004	0.008	0.016
SFT	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.0 / 0.422 / 0.422	0.0	0.0	0.008	0.0	0.047	0.0	0.008	0.008	0.008
SFT	SA	S,TC,SC,DG,PE,TLD	Prefill	0.0 / 0.895 / 0.898	0.0	0.0	0.016	0.0	0.141	0.039	0.0	0.012	0.012
SFT	SA	S,TC,SC,DG,PE,TLD	TAP	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	Adv.	0.0 / 0.91 / 0.91	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.414 / 0.879 / 0.879	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	2-turn	0.078 / 0.445 / 0.448	0.988	1.0	1.0	1.0	1.0	0.996	1.0	1.0	0.996
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.109 / 0.352 / 0.352	0.996	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	Prefill	0.004 / 0.914 / 0.914	1.0	0.996	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	TAP	0.0 / 0.0 / 0.0	0.1	0.1	0.3	0.0	0.1	0.6	0.0	0.2	0.2
DPO	SA	S,TC,SC,DG,PE,TLD	Adv.	0.062 / 0.574 / 0.586	1.0	1.0	1.0	1.0	0.996	1.0	0.918	0.93	0.957
DPO	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DPO	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.176 / 0.68 / 0.684	0.711	0.086	0.445	0.039	0.207	0.754	0.105	0.309	0.023
DPO	SA	S,TC,SC,DG,PE,TLD	2-turn	0.469 / 0.418 / 0.465	1.0	0.75	1.0	0.953	0.883	1.0	1.0	0.902	0.965
DPO	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.344 / 0.578 / 0.592	1.0	0.797	1.0	0.984	0.84	1.0	0.762	0.895	0.949
DPO	SA	S,TC,SC,DG,PE,TLD	Prefill	0.137 / 0.605 / 0.605	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.988	0.957
DPO	SA	S,TC,SC,DG,PE,TLD	TAP	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Probe	SA	S,TC,SC,DG,PE,TLD	Adv.	0.219	1.0	0.906	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	b64	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.012	1.0	0.922	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	2-turn	0.012	0.203	0.629	0.867	1.0	0.977	0.781	0.035	0.871	0.0
Probe	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.012	0.773	0.641	1.0	1.0	1.0	1.0	0.422	0.996	0.016
Probe	SA	S,TC,SC,DG,PE,TLD	Prefill	0.211	1.0	0.848	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	TAP	0.219	1.0	0.906	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 4: Results for adversarial evaluation on Summarization with Mistral.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S	SA,TLD,PE,TC,SC,DG	Adv.	0.16	0.645	0.004 / 0.0 / 0.166	0.203	0.031	0.332	0.598	0.062	0.258	0.336
Sys.	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.063	0.0	0.0	0.004	0.0	0.0	0.0	0.0
Sys.	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.141	0.117	0.0 / 0.0 / 0.182	0.004	0.0	0.043	0.43	0.004	0.051	0.016
Sys.	S	SA,TLD,PE,TC,SC,DG	2-turn	0.113	0.453	0.0 / 0.0 / 0.171	0.086	0.0	0.246	0.297	0.953	0.289	0.734
Sys.	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.137	0.281	0.0 / 0.0 / 0.172	0.137	0.0	0.145	0.371	0.027	0.18	0.383
Sys.	S	SA,TLD,PE,TC,SC,DG	Prefill	0.129	0.324	0.004 / 0.0 / 0.171	0.129	0.027	0.395	0.527	0.051	0.145	0.344
Sys.	S	SA,TLD,PE,TC,SC,DG	TAP	0.0	0.0	0.0 / 0.0 / 0.145	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CB	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.004 / 0.0 / 0.166	1.0	1.0	1.0	1.0	1.0	1.0	1.0
CB	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CB	S	SA,TLD,PE,TC,SC,DG	Few-shot	1.0	1.0	0.0 / 0.0 / 0.183	1.0	1.0	1.0	1.0	0.852	1.0	0.031
CB	S	SA,TLD,PE,TC,SC,DG	2-turn	1.0	1.0	0.02 / 0.0 / 0.165	1.0	1.0	1.0	1.0	1.0	1.0	1.0
CB	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	1.0	1.0	0.004 / 0.0 / 0.174	1.0	1.0	1.0	1.0	0.855	0.91	0.996
CB	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.004 / 0.0 / 0.172	1.0	1.0	1.0	1.0	1.0	1.0	0.988
CB	S	SA,TLD,PE,TC,SC,DG	TAP	0.0	0.0	0.0 / 0.0 / 0.157	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT	S	SA,TLD,PE,TC,SC,DG	Adv.	0.0	0.0	0.0 / 0.0 / 0.106	0.0	0.0	0.051	0.0	0.008	0.0	0.012
SFT	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.008 / 0.0 / 0.015	0.0	0.008	0.0	0.0	0.0	0.0	0.0
SFT	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.0	0.0	0.0 / 0.0 / 0.098	0.004	0.0	0.027	0.0	0.0	0.0	0.004
SFT	S	SA,TLD,PE,TC,SC,DG	2-turn	0.0	0.0	0.0 / 0.0 / 0.1	0.0	0.0	0.035	0.0	0.0	0.0	0.012
SFT	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.0	0.0	0.0 / 0.0 / 0.097	0.0	0.0	0.047	0.0	0.0	0.0	0.004
SFT	S	SA,TLD,PE,TC,SC,DG	Prefill	0.008	0.004	0.004 / 0.0 / 0.105	0.008	0.0	0.043	0.0	0.0	0.0	0.004
SFT	S	SA,TLD,PE,TC,SC,DG	TAP	0.0	0.0	0.0 / 0.0 / 0.117	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.012 / 0.0 / 0.205	0.973	1.0	1.0	0.977	1.0	0.996	1.0
SFT → CB	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Few-shot	1.0	1.0	0.012 / 0.0 / 0.195	1.0	1.0	1.0	1.0	0.84	0.961	0.398
SFT → CB	S	SA,TLD,PE,TC,SC,DG	2-turn	0.918	0.93	0.023 / 0.0 / 0.19	0.789	0.965	0.93	0.941	1.0	0.957	0.91
SFT → CB	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.887	0.996	0.051 / 0.0 / 0.195	0.984	0.965	0.996	0.973	1.0	0.965	0.758
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.035 / 0.0 / 0.23	1.0	1.0	1.0	0.992	1.0	1.0	1.0
SFT → CB	S	SA,TLD,PE,TC,SC,DG	TAP	0.0	0.0	0.0 / 0.0 / 0.136	0.0	0.0	0.1	0.1	0.0	0.1	0.0
DPO	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.004 / 0.0 / 0.178	1.0	1.0	1.0	1.0	0.617	0.945	0.938
DPO	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DPO	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.816	0.996	0.0 / 0.0 / 0.183	0.945	0.035	0.469	0.938	0.332	0.527	0.031
DPO	S	SA,TLD,PE,TC,SC,DG	2-turn	0.781	1.0	0.012 / 0.0 / 0.177	1.0	0.891	0.664	1.0	1.0	0.883	0.957
DPO	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.672	1.0	0.0 / 0.0 / 0.173	0.992	0.945	0.543	1.0	0.148	0.824	0.941
DPO	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.004 / 0.0 / 0.182	1.0	1.0	1.0	1.0	1.0	0.992	0.949
DPO	S	SA,TLD,PE,TC,SC,DG	TAP	0.0	0.0	0.0 / 0.0 / 0.227	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Probe	S	SA,TLD,PE,TC,SC,DG	Adv.	0.938	1.0	0.336	1.0	1.0	1.0	1.0	1.0	0.973	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	b64	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.988	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.996	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	2-turn	0.0	0.605	0.0	0.379	0.988	0.688	0.445	0.0	0.555	0.0
Probe	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.055	0.926	0.0	0.93	1.0	1.0	0.938	0.0	0.727	0.0
Probe	S	SA,TLD,PE,TC,SC,DG	Prefill	0.914	1.0	0.242	1.0	1.0	1.0	1.0	1.0	0.969	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	TAP	0.938	1.0	0.336	1.0	1.0	1.0	1.0	1.0	0.973	1.0

Table 5: Results for adversarial evaluation on Program Execution with Mistral.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE	SA,TLD,S,TC,SC	Adv.	0.188	0.719	0.004	0.227	0.035	0.34	0.559 / 0.031 / 0.217	0.023	0.238	0.23
Sys.	PE	SA,TLD,S,TC,SC	b64	0.0	0.004	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.005	0.0	0.0	0.0
Sys.	PE	SA,TLD,S,TC,SC	Few-shot	0.141	0.117	0.0	0.004	0.0	0.043	0.43 / 0.027 / 0.221	0.004	0.051	0.016
Sys.	PE	SA,TLD,S,TC,SC	2-turn	0.574	0.77	0.008	0.348	0.156	0.34	0.547 / 0.051 / 0.196	0.969	0.496	0.879
Sys.	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.363	0.758	0.004	0.16	0.32	0.172	0.438 / 0.062 / 0.231	0.281	0.328	0.449
Sys.	PE	SA,TLD,S,TC,SC	Prefill	0.207	0.543	0.004	0.215	0.027	0.41	0.43 / 0.039 / 0.22	0.027	0.148	0.328
Sys.	PE	SA,TLD,S,TC,SC	TAP	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.071	0.0	0.0	0.0
CB	PE	SA,TLD,S,TC,SC	Adv.	0.988	1.0	0.957	0.926	0.984	0.992	0.613 / 0.027 / 0.194	0.824	1.0	1.0
CB	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
CB	PE	SA,TLD,S,TC,SC	Few-shot	1.0	0.988	1.0	0.957	1.0	1.0	0.379 / 0.035 / 0.185	0.512	0.984	1.0
CB	PE	SA,TLD,S,TC,SC	2-turn	1.0	1.0	0.988	0.996	0.953	0.996	0.723 / 0.043 / 0.093	0.996	0.996	0.996
CB	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.996	1.0	1.0	0.996	0.992	0.988	0.551 / 0.043 / 0.196	0.934	1.0	1.0
CB	PE	SA,TLD,S,TC,SC	Prefill	0.988	1.0	0.965	1.0	0.969	0.996	0.293 / 0.043 / 0.226	0.988	1.0	0.957
CB	PE	SA,TLD,S,TC,SC	TAP	0.5	0.5	0.4	0.4	0.1	0.1	0.0 / 0.0 / 0.057	0.0	0.2	0.1
SFT	PE	SA,TLD,S,TC,SC	Adv.	0.02	0.0	0.0	0.0	0.0	0.195	0.039 / 0.0 / 0.428	0.004	0.0	0.012
SFT	PE	SA,TLD,S,TC,SC	b64	0.004	0.0	0.086	0.008	0.105	0.004	0.0 / 0.0 / 0.017	0.0	0.0	0.0
SFT	PE	SA,TLD,S,TC,SC	Few-shot	0.012	0.0	0.0	0.0	0.0	0.078	0.082 / 0.0 / 0.427	0.0	0.105	0.012
SFT	PE	SA,TLD,S,TC,SC	2-turn	0.109	0.0	0.0	0.117	0.0	0.203	0.148 / 0.004 / 0.434	0.301	0.238	0.105
SFT	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.09	0.004	0.0	0.023	0.0	0.121	0.074 / 0.004 / 0.448	0.188	0.156	0.02
SFT	PE	SA,TLD,S,TC,SC	Prefill	0.07	0.016	0.0	0.008	0.0	0.293	0.02 / 0.0 / 0.411	0.0	0.008	0.012
SFT	PE	SA,TLD,S,TC,SC	TAP	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.28	0.0	0.0	0.0
SFT → CB	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	1.0	1.0	0.027 / 0.25 / 0.55	0.996	0.988	1.0
SFT → CB	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
SFT → CB	PE	SA,TLD,S,TC,SC	Few-shot	1.0	1.0	1.0	1.0	1.0	1.0	0.469 / 0.02 / 0.229	0.273	1.0	1.0
SFT → CB	PE	SA,TLD,S,TC,SC	2-turn	0.98	1.0	1.0	0.996	1.0	0.996	0.184 / 0.223 / 0.514	0.738	0.934	0.887
SFT → CB	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.672	0.484	0.961	1.0	1.0	1.0	0.152 / 0.234 / 0.523	0.578	0.953	0.434
SFT → CB	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.008 / 0.254 / 0.582	0.98	1.0	0.961
SFT → CB	PE	SA,TLD,S,TC,SC	TAP	0.5	0.7	0.2	0.3	0.3	0.1	0.0 / 0.0 / 0.097	0.0	0.2	0.4
DPO	PE	SA,TLD,S,TC,SC	Adv.	0.945	1.0	0.988	1.0	1.0	1.0	0.148 / 0.113 / 0.34	0.02	0.602	0.746
DPO	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
DPO	PE	SA,TLD,S,TC,SC	Few-shot	0.777	0.992	0.215	0.816	0.219	0.469	0.281 / 0.102 / 0.303	0.219	0.438	0.027
DPO	PE	SA,TLD,S,TC,SC	2-turn	0.996	1.0	0.887	1.0	0.992	0.945	0.48 / 0.059 / 0.213	1.0	0.848	0.996
DPO	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.922	1.0	0.816	1.0	0.984	0.875	0.344 / 0.055 / 0.224	0.277	0.734	0.992
DPO	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	0.996	0.066 / 0.109 / 0.359	0.57	0.785	0.875
DPO	PE	SA,TLD,S,TC,SC	TAP	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.029	0.0	0.0	0.0
Probe	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	1.0	1.0	0.074	1.0	0.992	1.0
Probe	PE	SA,TLD,S,TC,SC	b64	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	PE	SA,TLD,S,TC,SC	Few-shot	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
Probe	PE	SA,TLD,S,TC,SC	2-turn	0.809	1.0	0.992	0.582	1.0	0.688	0.0	0.039	0.395	0.0
Probe	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.891	1.0	0.996	0.973	1.0	0.969	0.0	0.07	0.555	0.852
Probe	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.027	0.996	0.98	0.973
Probe	PE	SA,TLD,S,TC,SC	TAP	1.0	1.0	1.0	1.0	1.0	1.0	0.074	1.0	0.992	1.0

Table 6: Results for adversarial evaluation on Sentiment Analysis with Granite.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA	S,TC,SC,DG,PE,TLD	Adv.	0.008 / 0.777 / 0.784	0.0	0.004	0.004	0.0	0.152	0.0	0.055	0.004	0.016
Sys.	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sys.	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.0 / 0.68 / 0.801	0.0	0.012	0.0	0.0	0.125	0.0	0.0	0.004	0.012
Sys.	SA	S,TC,SC,DG,PE,TLD	2-turn	0.0 / 0.816 / 0.816	0.0	0.0	0.008	0.004	0.16	0.0	0.012	0.012	0.031
Sys.	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.0 / 0.824 / 0.824	0.0	0.0	0.004	0.004	0.18	0.0	0.258	0.012	0.023
Sys.	SA	S,TC,SC,DG,PE,TLD	Prefill	0.004 / 0.652 / 0.659	0.0	0.004	0.008	0.0	0.066	0.0	0.0	0.004	0.016
CB	SA	S,TC,SC,DG,PE,TLD	Adv.	0.012 / 0.781 / 0.788	1.0	1.0	1.0	1.0	1.0	1.0	0.055	0.504	0.016
CB	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.004	0.004	0.004	0.0	0.0	0.0	0.0	0.0
CB	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.0 / 0.68 / 0.801	1.0	1.0	1.0	1.0	1.0	1.0	0.008	0.07	0.027
CB	SA	S,TC,SC,DG,PE,TLD	2-turn	0.0 / 0.816 / 0.816	0.051	1.0	0.941	1.0	1.0	0.59	0.016	0.012	0.043
CB	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.0 / 0.824 / 0.824	0.066	0.996	0.941	1.0	1.0	0.637	0.324	0.062	0.016
CB	SA	S,TC,SC,DG,PE,TLD	Prefill	0.008 / 0.648 / 0.655	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.598	0.137
SFT	SA	S,TC,SC,DG,PE,TLD	Adv.	0.0 / 0.906 / 0.906	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.996	0.996
SFT	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.0 / 0.75 / 0.871	0.715	0.961	0.637	0.996	0.977	0.961	0.055	0.445	0.195
SFT	SA	S,TC,SC,DG,PE,TLD	2-turn	0.0 / 0.859 / 0.859	0.816	0.77	0.984	1.0	0.945	0.77	0.996	0.672	0.992
SFT	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.0 / 0.879 / 0.879	0.863	0.926	0.984	1.0	0.973	0.941	1.0	0.707	0.988
SFT	SA	S,TC,SC,DG,PE,TLD	Prefill	0.02 / 0.871 / 0.871	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.996	1.0
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	Adv.	0.0 / 0.91 / 0.91	1.0	1.0	1.0	1.0	0.562	1.0	1.0	0.977	0.996
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	Few-shot	0.0 / 0.75 / 0.871	1.0	1.0	1.0	1.0	1.0	1.0	0.062	0.695	0.223
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	2-turn	0.0 / 0.859 / 0.859	0.793	1.0	0.98	1.0	0.832	0.887	1.0	0.789	0.992
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	2-turn+Sys.	0.0 / 0.883 / 0.883	0.863	0.988	0.949	1.0	0.758	0.938	1.0	0.832	0.988
SFT $\rightarrow$ CB	SA	S,TC,SC,PE	Prefill	0.02 / 0.871 / 0.871	0.93	1.0	1.0	1.0	0.426	1.0	1.0	1.0	0.988
DPO	SA	S,TC,SC,DG,PE,TLD	Adv.	0.004 / 0.809 / 0.82	0.941	0.527	0.309	0.711	0.512	0.812	0.793	0.594	0.23
DPO	SA	S,TC,SC,DG,PE,TLD	b64	0.0 / 0.0 / 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DPO	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.0 / 0.676 / 0.777	0.031	0.02	0.059	0.172	0.242	0.566	0.0	0.336	0.051
DPO	SA	S,TC,SC,DG,PE,TLD	2-turn	0.0 / 0.824 / 0.824	0.039	0.125	0.094	0.52	0.305	0.422	0.547	0.363	0.141
DPO	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.004 / 0.828 / 0.828	0.16	0.324	0.129	0.812	0.301	0.566	0.945	0.453	0.121
DPO	SA	S,TC,SC,DG,PE,TLD	Prefill	0.004 / 0.523 / 0.523	0.676	0.391	0.469	0.359	0.492	0.719	0.23	0.414	0.523
Probe	SA	S,TC,SC,DG,PE,TLD	Adv.	0.223	1.0	0.938	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	b64	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	Few-shot	0.012	1.0	0.992	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	2-turn	0.012	0.824	0.672	0.953	1.0	1.0	0.922	0.016	0.895	0.0
Probe	SA	S,TC,SC,DG,PE,TLD	2-turn+Sys.	0.012	1.0	0.688	1.0	1.0	1.0	1.0	0.266	1.0	0.027
Probe	SA	S,TC,SC,DG,PE,TLD	Prefill	0.211	1.0	0.91	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 7: Results for adversarial evaluation on Summarization with Granite.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S	SA,TLD,PE,TC,SC,DG	Adv.	0.008	0.004	0.004 / 0.0 / 0.176	0.004	0.0	0.152	0.0	0.055	0.004	0.016
Sys.	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.028	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sys.	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.0	0.0	0.012 / 0.0 / 0.217	0.0	0.0	0.125	0.0	0.0	0.004	0.012
Sys.	S	SA,TLD,PE,TC,SC,DG	2-turn	0.0	0.0	0.008 / 0.0 / 0.221	0.004	0.0	0.105	0.0	0.008	0.0	0.008
Sys.	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.0	0.0	0.004 / 0.0 / 0.217	0.0	0.0	0.125	0.0	0.145	0.004	0.016
Sys.	S	SA,TLD,PE,TC,SC,DG	Prefill	0.004	0.0	0.004 / 0.0 / 0.169	0.0	0.0	0.074	0.0	0.0	0.004	0.016
CB	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.004 / 0.0 / 0.173	1.0	1.0	1.0	1.0	0.254	0.938	0.051
CB	S	SA,TLD,PE,TC,SC,DG	b64	0.035	0.023	0.012 / 0.0 / 0.013	0.008	0.02	0.031	0.0	0.0	0.039	0.0
CB	S	SA,TLD,PE,TC,SC,DG	Few-shot	1.0	1.0	0.012 / 0.0 / 0.22	1.0	1.0	1.0	1.0	0.008	0.996	0.027
CB	S	SA,TLD,PE,TC,SC,DG	2-turn	0.281	0.844	0.008 / 0.0 / 0.22	1.0	0.293	1.0	0.754	0.02	0.262	0.016
CB	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.605	1.0	0.004 / 0.0 / 0.216	1.0	0.355	1.0	0.875	0.172	0.398	0.035
CB	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.004 / 0.0 / 0.168	1.0	1.0	1.0	1.0	0.004	0.957	0.211
SFT	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.012 / 0.008 / 0.275	1.0	1.0	1.0	1.0	1.0	0.996	0.988
SFT	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.988	1.0	0.008 / 0.0 / 0.263	0.793	0.914	0.645	0.996	0.035	0.906	0.141
SFT	S	SA,TLD,PE,TC,SC,DG	2-turn	0.883	0.996	0.008 / 0.008 / 0.27	0.949	0.895	0.578	0.934	0.996	0.883	0.852
SFT	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	1.0	1.0	0.027 / 0.008 / 0.268	1.0	0.984	1.0	1.0	1.0	0.957	0.863
SFT	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.062 / 0.004 / 0.259	1.0	1.0	1.0	1.0	1.0	1.0	0.996
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Adv.	1.0	1.0	0.012 / 0.008 / 0.276	1.0	1.0	1.0	1.0	0.461	0.984	0.871
SFT → CB	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Few-shot	1.0	1.0	0.008 / 0.0 / 0.263	1.0	1.0	1.0	1.0	0.988	1.0	0.102
SFT → CB	S	SA,TLD,PE,TC,SC,DG	2-turn	0.582	1.0	0.008 / 0.008 / 0.268	1.0	0.781	1.0	0.914	0.996	0.891	0.91
SFT → CB	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.434	1.0	0.027 / 0.008 / 0.264	1.0	0.992	1.0	0.965	1.0	0.781	0.883
SFT → CB	S	SA,TLD,PE,TC,SC,DG	Prefill	1.0	1.0	0.066 / 0.004 / 0.256	1.0	1.0	1.0	1.0	0.715	1.0	0.73
DPO	S	SA,TLD,PE,TC,SC,DG	Adv.	0.805	0.945	0.0 / 0.0 / 0.223	0.629	0.613	0.688	0.945	0.84	0.805	0.414
DPO	S	SA,TLD,PE,TC,SC,DG	b64	0.0	0.0	0.0 / 0.0 / 0.015	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DPO	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.172	0.062	0.008 / 0.0 / 0.22	0.129	0.059	0.242	0.48	0.0	0.32	0.039
DPO	S	SA,TLD,PE,TC,SC,DG	2-turn	0.211	0.438	0.004 / 0.0 / 0.124	0.031	0.254	0.211	0.379	0.219	0.453	0.156
DPO	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.414	0.645	0.004 / 0.0 / 0.229	0.074	0.477	0.289	0.754	0.875	0.578	0.199
DPO	S	SA,TLD,PE,TC,SC,DG	Prefill	0.918	0.98	0.004 / 0.0 / 0.179	0.777	0.555	0.703	0.953	0.375	0.711	0.836
Probe	S	SA,TLD,PE,TC,SC,DG	Adv.	0.918	1.0	0.258	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	b64	0.805	0.957	0.32	0.684	0.293	0.562	1.0	1.0	0.875	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	Few-shot	0.949	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	2-turn	0.0	0.0	0.0	0.059	0.641	0.625	0.113	0.0	0.004	0.0
Probe	S	SA,TLD,PE,TC,SC,DG	2-turn+Sys.	0.0	0.07	0.0	0.223	0.641	0.688	0.23	0.0	0.074	0.0
Probe	S	SA,TLD,PE,TC,SC,DG	Prefill	0.891	1.0	0.145	1.0	1.0	1.0	1.0	1.0	0.961	1.0

Table 8: Results for adversarial evaluation on Program Execution with Granite.

Method	Accept Sets	Reject Sets	Prompt Style	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE	SA,TLD,S,TC,SC	Adv.	0.016	0.004	0.004	0.004	0.0	0.156	0.0 / 0.0 / 0.111	0.105	0.008	0.008
Sys.	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
Sys.	PE	SA,TLD,S,TC,SC	Few-shot	0.0	0.0	0.012	0.0	0.0	0.125	0.0 / 0.211 / 0.43	0.0	0.004	0.012
Sys.	PE	SA,TLD,S,TC,SC	2-turn	0.0	0.0	0.0	0.004	0.0	0.098	0.0 / 0.148 / 0.329	0.008	0.008	0.012
Sys.	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.0	0.0	0.0	0.008	0.0	0.113	0.0 / 0.094 / 0.237	0.102	0.012	0.031
Sys.	PE	SA,TLD,S,TC,SC	Prefill	0.004	0.0	0.0	0.008	0.0	0.074	0.0 / 0.004 / 0.174	0.0	0.004	0.043
CB	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.0 / 0.112	0.059	0.961	0.961
CB	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.008	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
CB	PE	SA,TLD,S,TC,SC	Few-shot	1.0	1.0	1.0	1.0	1.0	1.0	0.004 / 0.215 / 0.437	0.008	1.0	1.0
CB	PE	SA,TLD,S,TC,SC	2-turn	1.0	1.0	1.0	1.0	1.0	0.98	0.0 / 0.148 / 0.326	0.027	0.734	0.43
CB	PE	SA,TLD,S,TC,SC	2-turn+Sys.	1.0	0.98	1.0	1.0	1.0	1.0	0.0 / 0.09 / 0.237	0.074	0.746	0.613
CB	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.008 / 0.168	0.035	0.938	0.84
SFT	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	0.992	1.0	0.0 / 0.465 / 0.644	0.852	0.789	0.938
SFT	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.001	0.0	0.0	0.0
SFT	PE	SA,TLD,S,TC,SC	Few-shot	0.953	0.992	0.996	0.914	0.953	0.742	0.0 / 0.43 / 0.625	0.07	0.465	0.129
SFT	PE	SA,TLD,S,TC,SC	2-turn	1.0	0.992	1.0	0.977	0.977	0.926	0.0 / 0.434 / 0.628	0.426	0.504	0.816
SFT	PE	SA,TLD,S,TC,SC	2-turn+Sys.	1.0	1.0	1.0	0.996	0.984	0.945	0.0 / 0.43 / 0.621	0.965	0.645	0.855
SFT	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.004 / 0.426 / 0.614	0.816	0.844	0.973
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.473 / 0.645	0.816	1.0	1.0
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.0	0.0	0.0	0.0
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	Few-shot	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.43 / 0.626	0.07	1.0	1.0
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	2-turn	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.434 / 0.628	0.398	0.996	0.871
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	2-turn+Sys.	1.0	1.0	1.0	1.0	1.0	1.0	0.0 / 0.441 / 0.633	0.891	0.984	0.992
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.02 / 0.41 / 0.593	0.789	1.0	0.953
DPO	PE	SA,TLD,S,TC,SC	Adv.	0.344	0.535	0.188	0.297	0.309	0.203	0.0 / 0.113 / 0.31	0.453	0.125	0.043
DPO	PE	SA,TLD,S,TC,SC	b64	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.002	0.0	0.0	0.0
DPO	PE	SA,TLD,S,TC,SC	Few-shot	0.078	0.043	0.012	0.008	0.023	0.125	0.0 / 0.203 / 0.452	0.0	0.043	0.02
DPO	PE	SA,TLD,S,TC,SC	2-turn	0.148	0.137	0.016	0.027	0.078	0.117	0.004 / 0.168 / 0.369	0.176	0.035	0.07
DPO	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.211	0.207	0.062	0.027	0.133	0.117	0.004 / 0.184 / 0.382	0.488	0.051	0.047
DPO	PE	SA,TLD,S,TC,SC	Prefill	0.402	0.547	0.176	0.168	0.195	0.09	0.0 / 0.059 / 0.233	0.0	0.039	0.305
Probe	PE	SA,TLD,S,TC,SC	Adv.	1.0	1.0	1.0	1.0	1.0	1.0	0.047	1.0	0.996	1.0
Probe	PE	SA,TLD,S,TC,SC	b64	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	PE	SA,TLD,S,TC,SC	Few-shot	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
Probe	PE	SA,TLD,S,TC,SC	2-turn	0.812	1.0	0.996	0.641	1.0	0.688	0.0	0.098	0.5	0.0
Probe	PE	SA,TLD,S,TC,SC	2-turn+Sys.	0.891	1.0	0.996	0.969	1.0	0.996	0.0	0.195	0.66	0.059
Probe	PE	SA,TLD,S,TC,SC	Prefill	1.0	1.0	1.0	1.0	1.0	1.0	0.031	0.996	0.98	0.965

Table 9: Results for rejection set diversity on Sentiment Analysis.

Method	Accept Sets	Reject Sets	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA	S	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE,TLD	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
CB	SA	S	0.105 / 0.531 / 0.586	0.203	0.996	0.785	1.0	0.93	0.801	0.77	0.66	0.797
CB	SA	S,TC	0.098 / 0.535 / 0.59	0.609	1.0	1.0	1.0	1.0	0.957	0.957	0.98	0.945
CB	SA	S,TC,SC	0.102 / 0.531 / 0.586	0.633	1.0	1.0	1.0	1.0	0.98	0.98	0.969	0.953
CB	SA	S,TC,SC,DG	0.102 / 0.531 / 0.586	0.785	1.0	1.0	1.0	1.0	0.91	0.965	0.992	0.953
CB	SA	S,TC,SC,DG,PE	0.102 / 0.531 / 0.586	0.883	1.0	1.0	1.0	1.0	1.0	0.996	1.0	0.984
CB	SA	S,TC,SC,DG,PE,TLD	0.102 / 0.527 / 0.586	0.246	0.004	0.855	0.117	0.84	0.773	0.77	0.516	0.699
SFT	SA	S	0.0 / 0.91 / 0.91	0.02	0.98	0.535	0.574	0.672	0.406	0.184	0.164	0.816
SFT	SA	S,TC	0.0 / 0.891 / 0.895	0.02	0.973	1.0	0.609	0.715	0.453	0.133	0.367	0.781
SFT	SA	S,TC,SC	0.0 / 0.914 / 0.914	0.008	1.0	1.0	1.0	0.797	0.543	0.289	0.488	0.895
SFT	SA	S,TC,SC,DG	0.0 / 0.883 / 0.883	0.035	0.98	1.0	1.0	0.805	0.559	0.258	0.535	0.852
SFT	SA	S,TC,SC,DG,PE	0.0 / 0.883 / 0.883	0.031	0.984	1.0	0.992	0.844	0.742	0.48	0.648	0.906
SFT	SA	S,TC,SC,DG,PE,TLD	0.0 / 0.902 / 0.902	0.809	0.953	1.0	0.996	0.832	0.711	0.465	0.664	0.91
SFT $\rightarrow$ CB	SA	S	0.004 / 0.91 / 0.91	0.031	1.0	0.961	1.0	1.0	0.883	0.965	0.871	0.973
SFT $\rightarrow$ CB	SA	S,TC	0.0 / 0.91 / 0.91	0.566	1.0	1.0	1.0	1.0	0.773	0.996	0.949	0.895
SFT $\rightarrow$ CB	SA	S,TC,SC	0.0 / 0.91 / 0.91	0.879	1.0	1.0	1.0	1.0	0.906	1.0	0.992	0.945
SFT $\rightarrow$ CB	SA	S,TC,SC,DG	0.0 / 0.914 / 0.914	0.82	1.0	1.0	1.0	1.0	0.898	0.574	1.0	0.965
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE	0.004 / 0.91 / 0.91	0.887	0.992	1.0	1.0	1.0	0.992	0.551	0.996	0.969
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	0.0 / 0.914 / 0.914	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.996	0.988
DPO	SA	S	0.0 / 0.516 / 0.68	0.012	1.0	0.781	0.953	0.75	0.984	0.945	0.641	0.863
DPO	SA	S,TC	0.0 / 0.703 / 0.773	0.094	1.0	1.0	0.992	1.0	1.0	1.0	0.887	0.93
DPO	SA	S,TC,SC	0.008 / 0.629 / 0.734	0.109	1.0	1.0	1.0	1.0	1.0	1.0	0.898	0.934
DPO	SA	S,TC,SC,DG	0.008 / 0.66 / 0.691	0.145	1.0	1.0	1.0	1.0	1.0	1.0	0.922	0.938
DPO	SA	S,TC,SC,DG,PE	0.031 / 0.445 / 0.449	0.238	1.0	1.0	1.0	1.0	1.0	1.0	0.969	0.957
DPO	SA	S,TC,SC,DG,PE,TLD	0.02 / 0.609 / 0.613	1.0	1.0	1.0	1.0	0.996	1.0	1.0	0.98	0.953
Probe	SA	S	0.047	0.039	1.0	0.516	1.0	0.688	0.328	0.383	0.441	0.023
Probe	SA	S,TC	0.141	0.883	0.996	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC	0.027	0.41	0.918	1.0	1.0	1.0	1.0	1.0	1.0	0.984
Probe	SA	S,TC,SC,DG	0.02	0.82	0.883	1.0	1.0	1.0	1.0	1.0	1.0	0.996
Probe	SA	S,TC,SC,DG,PE	0.117	0.852	0.848	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	0.207	1.0	0.824	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 10: Results for rejection set diversity on Summarization.

Method	Accept Sets	Reject Sets	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S	SA	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC,DG	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
CB	S	SA	0.996	0.984	0.004 / 0.0 / 0.167	1.0	0.691	1.0	1.0	1.0	0.973	0.984
CB	S	SA,TLD	1.0	1.0	0.004 / 0.0 / 0.168	0.992	0.395	0.715	0.906	0.758	0.914	0.91
CB	S	SA,TLD,PE	1.0	1.0	0.004 / 0.0 / 0.168	0.992	0.988	1.0	1.0	1.0	0.953	0.977
CB	S	SA,TLD,PE,TC	1.0	1.0	0.004 / 0.0 / 0.166	1.0	0.984	1.0	1.0	0.996	0.957	0.988
CB	S	SA,TLD,PE,TC,SC	1.0	1.0	0.004 / 0.0 / 0.168	1.0	1.0	1.0	1.0	1.0	1.0	0.992
CB	S	SA,TLD,PE,TC,SC,DG	1.0	1.0	0.004 / 0.0 / 0.166	1.0	1.0	1.0	1.0	1.0	1.0	0.984
SFT	S	SA	0.988	1.0	0.0 / 0.0 / 0.201	0.738	0.012	0.512	0.902	0.316	0.68	0.473
SFT	S	SA,TLD	1.0	1.0	0.004 / 0.0 / 0.198	0.98	0.008	0.887	0.988	0.832	0.828	0.641
SFT	S	SA,TLD,PE	0.914	1.0	0.004 / 0.0 / 0.208	0.691	0.012	0.445	0.988	0.75	0.727	0.527
SFT	S	SA,TLD,PE,TC	0.996	1.0	0.004 / 0.0 / 0.203	1.0	0.152	0.922	0.996	0.543	0.855	0.527
SFT	S	SA,TLD,PE,TC,SC	1.0	1.0	0.004 / 0.0 / 0.199	1.0	1.0	1.0	1.0	0.98	0.98	0.973
SFT	S	SA,TLD,PE,TC,SC,DG	0.988	1.0	0.004 / 0.0 / 0.205	1.0	1.0	0.996	0.98	0.871	0.914	0.898
SFT $\rightarrow$ CB	S	SA	1.0	1.0	0.02 / 0.0 / 0.202	0.57	0.051	0.629	0.398	0.039	0.496	0.082
SFT $\rightarrow$ CB	S	SA,TLD	1.0	1.0	0.016 / 0.0 / 0.2	0.672	0.066	0.762	0.637	0.961	0.703	0.961
SFT $\rightarrow$ CB	S	SA,TLD,PE	1.0	1.0	0.016 / 0.0 / 0.2	0.867	0.523	0.965	1.0	0.984	0.855	0.801
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC	1.0	1.0	0.016 / 0.0 / 0.201	1.0	0.879	1.0	1.0	0.984	0.867	0.641
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC	1.0	0.961	0.016 / 0.0 / 0.201	1.0	0.992	0.961	0.957	0.148	0.844	0.387
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	1.0	1.0	0.016 / 0.0 / 0.203	1.0	1.0	1.0	0.988	1.0	1.0	1.0
DPO	S	SA	1.0	1.0	0.0 / 0.0 / 0.174	0.914	0.301	0.922	1.0	1.0	0.859	0.891
DPO	S	SA,TLD	1.0	1.0	0.004 / 0.0 / 0.174	0.988	0.445	0.977	1.0	1.0	0.895	0.895
DPO	S	SA,TLD,PE	1.0	1.0	0.004 / 0.0 / 0.176	1.0	0.918	0.988	1.0	1.0	0.957	0.926
DPO	S	SA,TLD,PE,TC	1.0	1.0	0.004 / 0.0 / 0.171	1.0	0.996	1.0	1.0	1.0	0.969	0.938
DPO	S	SA,TLD,PE,TC,SC	1.0	1.0	0.004 / 0.0 / 0.174	1.0	1.0	1.0	1.0	1.0	0.98	0.945
DPO	S	SA,TLD,PE,TC,SC,DG	1.0	1.0	0.004 / 0.0 / 0.175	1.0	1.0	1.0	1.0	1.0	0.98	0.945
Probe	S	SA	1.0	1.0	0.297	0.984	0.75	1.0	1.0	1.0	0.898	1.0
Probe	S	SA,TLD	0.984	1.0	0.219	0.992	1.0	1.0	1.0	1.0	0.934	1.0
Probe	S	SA,TLD,PE	0.934	1.0	0.215	0.996	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC	0.906	1.0	0.203	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC	0.875	1.0	0.227	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	0.887	1.0	0.234	1.0	1.0	1.0	1.0	1.0	0.969	1.0

Table 11: Results for rejection set diversity on Program Execution.

Method	Accept Sets	Reject Sets	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE	SA	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC,DG	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
CB	PE	SA	1.0	1.0	0.863	0.98	0.996	0.973	0.457 / 0.043 / 0.244	0.973	0.914	0.91
CB	PE	SA,TLD	1.0	1.0	1.0	0.988	0.973	1.0	0.457 / 0.043 / 0.26	0.691	0.961	0.902
CB	PE	SA,TLD,S	1.0	1.0	1.0	0.902	1.0	0.922	0.461 / 0.039 / 0.253	0.746	0.949	0.941
CB	PE	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.465 / 0.039 / 0.236	0.953	0.977	0.961
CB	PE	SA,TLD,S,TC,SC	0.992	0.98	0.977	1.0	0.992	1.0	0.461 / 0.043 / 0.254	0.93	1.0	0.953
CB	PE	SA,TLD,S,TC,SC,DG	0.723	0.98	0.125	0.898	0.32	0.867	0.457 / 0.039 / 0.249	0.277	0.387	0.633
SFT	PE	SA	0.914	0.734	0.004	0.055	0.0	0.512	0.0 / 0.246 / 0.543	0.027	0.047	0.207
SFT	PE	SA,TLD	0.941	1.0	0.008	0.363	0.043	0.68	0.0 / 0.309 / 0.582	0.031	0.098	0.262
SFT	PE	SA,TLD,S	0.918	0.996	0.891	0.848	0.434	0.754	0.004 / 0.289 / 0.592	0.051	0.109	0.387
SFT	PE	SA,TLD,S,TC	0.926	0.996	0.934	1.0	0.836	0.852	0.0 / 0.27 / 0.577	0.023	0.191	0.395
SFT	PE	SA,TLD,S,TC,SC	0.941	0.996	0.969	1.0	1.0	0.938	0.0 / 0.273 / 0.58	0.16	0.32	0.605
SFT	PE	SA,TLD,S,TC,SC,DG	0.887	0.988	0.539	0.992	0.789	0.898	0.0 / 0.293 / 0.564	0.027	0.234	0.375
SFT → CB	PE	SA	1.0	1.0	0.941	0.809	0.48	0.949	0.02 / 0.246 / 0.543	0.934	0.73	0.895
SFT → CB	PE	SA,TLD	1.0	1.0	1.0	1.0	1.0	1.0	0.027 / 0.25 / 0.548	0.902	0.977	0.898
SFT → CB	PE	SA,TLD,S	1.0	1.0	1.0	1.0	1.0	1.0	0.02 / 0.246 / 0.551	0.93	1.0	0.918
SFT → CB	PE	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.023 / 0.25 / 0.549	0.344	0.938	0.43
SFT → CB	PE	SA,TLD,S,TC,SC	1.0	1.0	1.0	1.0	1.0	0.996	0.023 / 0.25 / 0.55	0.953	0.988	0.922
SFT → CB	PE	SA,TLD,S,TC,SC,DG	1.0	1.0	1.0	1.0	1.0	1.0	0.023 / 0.246 / 0.543	1.0	1.0	0.992
DPO	PE	SA	1.0	1.0	0.996	0.914	0.336	0.859	0.0 / 0.148 / 0.416	0.094	0.324	0.625
DPO	PE	SA,TLD	1.0	1.0	0.738	0.922	0.148	0.891	0.0 / 0.148 / 0.413	0.062	0.449	0.594
DPO	PE	SA,TLD,S	1.0	1.0	1.0	1.0	1.0	0.996	0.0 / 0.133 / 0.392	0.301	0.695	0.848
DPO	PE	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.008 / 0.145 / 0.406	0.25	0.75	0.844
DPO	PE	SA,TLD,S,TC,SC	1.0	1.0	1.0	1.0	1.0	1.0	0.016 / 0.141 / 0.403	0.301	0.785	0.883
DPO	PE	SA,TLD,S,TC,SC,DG	1.0	1.0	1.0	1.0	1.0	1.0	0.02 / 0.145 / 0.416	0.453	0.82	0.902
Probe	PE	SA	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.953	0.758	0.949
Probe	PE	SA,TLD	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.941	0.895	0.957
Probe	PE	SA,TLD,S	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.883	0.816	0.926
Probe	PE	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.008	0.996	0.98	0.98
Probe	PE	SA,TLD,S,TC,SC	1.0	1.0	1.0	1.0	1.0	1.0	0.012	0.996	0.98	0.965
Probe	PE	SA,TLD,S,TC,SC,DG	1.0	1.0	1.0	1.0	1.0	1.0	0.07	1.0	0.992	0.988

Table 12: Results for multiple accept sets set diversity on Classification and Generation.

Method	Accept Sets	Reject Sets	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.082 / 0.473 / 0.57	0.125 / 0.301 / 0.327	0.004 / 0.0 / 0.165	0.105 / 0.0 / 0.073	0.012 / 0.0 / 0.191	0.305 / 0.0 / 0.171	0.672	0.723	0.344	0.496
CB	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.078 / 0.48 / 0.578	0.109 / 0.301 / 0.329	0.004 / 0.0 / 0.166	0.105 / 0.0 / 0.074	0.012 / 0.0 / 0.191	0.32 / 0.0 / 0.175	1.0	1.0	0.621	0.949
SFT	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.0 / 0.867 / 0.871	0.0 / 0.543 / 0.585	0.0 / 0.0 / 0.182	0.0 / 0.328 / 0.389	0.0 / 0.023 / 0.417	0.039 / 0.0 / 0.302	0.93	0.973	0.168	0.398
SFT $\rightarrow$ CB	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.0 / 0.863 / 0.867	0.004 / 0.543 / 0.585	0.02 / 0.0 / 0.184	0.273 / 0.328 / 0.389	0.004 / 0.027 / 0.419	0.113 / 0.0 / 0.304	1.0	1.0	0.391	0.695
DPO	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.0 / 0.238 / 0.414	0.004 / 0.145 / 0.158	0.0 / 0.0 / 0.168	0.0 / 0.0 / 0.075	0.0 / 0.0 / 0.203	0.051 / 0.0 / 0.223	1.0	1.0	0.574	0.496
Probe	SA,TLD,S,TC,SC,DG	PE,GSM8K	0.242	0.004	0.035	0.527	0.0	0.312	1.0	1.0	0.82	1.0

Table 13: Results for multiple accept sets set diversity on Math and Programming.

Method	Accept Sets	Reject Sets	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE,GSM8K	SA,TLD,S,TC	0.23	0.836	0.004	0.234	0.016	0.383	0.312 / 0.047 / 0.245	0.008 / 0.035 / 0.046	0.195	0.508
CB	PE,GSM8K	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.312 / 0.043 / 0.246	0.008 / 0.027 / 0.038	0.996	0.848
SFT	PE,GSM8K	SA,TLD,S,TC	0.988	1.0	0.988	0.992	0.645	0.754	0.0 / 0.004 / 0.291	0.0 / 0.207 / 0.229	0.184	0.5
SFT $\rightarrow$ CB	PE,GSM8K	SA,TLD,S,TC	1.0	1.0	1.0	1.0	0.68	0.906	0.34 / 0.004 / 0.301	0.184 / 0.219 / 0.241	0.598	0.367
DPO	PE,GSM8K	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	0.988	0.012 / 0.121 / 0.371	0.0 / 0.051 / 0.082	0.418	0.715
Probe	PE,GSM8K	SA,TLD,S,TC	1.0	1.0	1.0	1.0	1.0	1.0	0.074	0.0	0.773	0.93

Table 14: Results for precise scoping on Sentiment Analysis.

Method	Accept Sets	Reject Sets	SA-FA	SA-FR	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA-FA	SA-FR	0.004 / 0.465 / 0.465	0.113	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA-FA	SA-FR,TLD,S,TC,PE	0.004 / 0.465 / 0.465	0.113	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA-FA	TLD,S,TC,PE	0.004 / 0.465 / 0.465	0.113	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
CB	SA-FA	SA-FR	0.004 / 0.461 / 0.461	1.0	1.0	1.0	0.996	1.0	1.0	1.0	0.879	0.973	0.746
CB	SA-FA	SA-FR,TLD,S,TC,PE	0.004 / 0.461 / 0.461	0.891	1.0	1.0	1.0	1.0	0.941	0.988	1.0	1.0	1.0
CB	SA-FA	TLD,S,TC,PE	0.004 / 0.457 / 0.457	0.949	1.0	1.0	1.0	1.0	1.0	1.0	0.996	1.0	0.875
SFT	SA-FA	SA-FR	0.0 / 0.945 / 0.945	1.0	0.992	0.031	0.965	0.371	0.719	0.82	0.703	0.727	0.727
SFT	SA-FA	SA-FR,TLD,S,TC,PE	0.0 / 0.953 / 0.953	1.0	1.0	0.996	1.0	0.98	0.965	1.0	0.98	0.941	0.969
SFT	SA-FA	TLD,S,TC,PE	0.0 / 0.949 / 0.949	0.07	0.988	0.996	1.0	1.0	0.914	0.965	0.84	0.852	0.945
SFT $\rightarrow$ CB	SA-FA	SA-FR	0.0 / 0.945 / 0.945	1.0	1.0	1.0	1.0	0.996	0.75	0.98	1.0	0.969	0.992
SFT $\rightarrow$ CB	SA-FA	SA-FR,TLD,S,TC,PE	0.0 / 0.945 / 0.945	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SFT $\rightarrow$ CB	SA-FA	TLD,S,TC,PE	0.0 / 0.945 / 0.945	0.992	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.973
DPO	SA-FA	SA-FR	0.0 / 0.0 / 0.003	1.0	1.0	0.734	1.0	0.715	0.941	1.0	1.0	0.855	0.957
DPO	SA-FA	SA-FR,TLD,S,TC,PE	0.0 / 0.004 / 0.004	1.0	1.0	1.0	1.0	0.992	1.0	1.0	1.0	1.0	0.969
DPO	SA-FA	TLD,S,TC,PE	0.0 / 0.668 / 0.668	0.781	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.996	0.969
Probe	SA-FA	SA-FR	0.023	1.0	1.0	0.785	1.0	1.0	1.0	1.0	0.992	1.0	1.0
Probe	SA-FA	SA-FR,TLD,S,TC,PE	0.004	0.984	1.0	0.855	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA-FA	TLD,S,TC,PE	0.0	0.906	1.0	0.84	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 15: Results for precise scoping on Summarization.

Method	Accept Sets	Reject Sets	S-FR	S-FA	SA	TLD	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S-FA	S-FR	0.004	0.0 / 0.0 / 0.181	0.148	0.66	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S-FA	S-FR,SA,TLD,TC,DG,PE	0.004	0.0 / 0.0 / 0.181	0.148	0.66	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S-FA	SA,TLD,TC,SC,PE	0.004	0.0 / 0.0 / 0.181	0.148	0.66	0.207	0.012	0.375	0.566	0.211	0.262	0.465
CB	S-FA	S-FR	1.0	0.0 / 0.0 / 0.18	0.918	0.473	0.238	0.008	0.363	0.559	0.211	0.293	0.477
CB	S-FA	S-FR,SA,TLD,TC,DG,PE	1.0	0.0 / 0.0 / 0.18	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.996
CB	S-FA	SA,TLD,TC,SC,PE	0.543	0.0 / 0.0 / 0.178	1.0	1.0	0.984	1.0	0.98	1.0	0.992	0.977	0.992
SFT	S-FA	S-FR	0.891	0.0 / 0.0 / 0.229	0.918	0.973	0.457	0.52	0.75	0.711	0.301	0.719	0.363
SFT	S-FA	S-FR,SA,TLD,TC,DG,PE	0.848	0.0 / 0.0 / 0.221	1.0	1.0	0.574	0.598	0.996	0.961	0.797	0.949	0.945
SFT	S-FA	SA,TLD,TC,SC,PE	0.176	0.0 / 0.0 / 0.18	1.0	1.0	0.711	1.0	1.0	0.996	0.965	0.953	0.949
SFT → CB	S-FA	S-FR	0.887	0.0 / 0.0 / 0.227	0.168	0.355	0.039	0.004	0.035	0.172	0.035	0.023	0.008
SFT → CB	S-FA	S-FR,SA,TLD,TC,DG,PE	0.988	0.0 / 0.0 / 0.231	0.996	1.0	0.984	0.996	1.0	0.973	0.418	0.898	0.105
SFT → CB	S-FA	SA,TLD,TC,SC,PE	0.586	0.0 / 0.0 / 0.23	1.0	0.996	0.996	1.0	0.992	0.996	0.309	0.891	0.129
DPO	S-FA	S-FR	1.0	0.0 / 0.0 / 0.214	0.926	1.0	0.984	0.977	0.922	0.996	0.98	0.836	0.926
DPO	S-FA	S-FR,SA,TLD,TC,DG,PE	1.0	0.0 / 0.0 / 0.197	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.973
DPO	S-FA	SA,TLD,TC,SC,PE	0.215	0.0 / 0.0 / 0.203	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.992	0.961
Probe	S-FA	S-FR	1.0	0.059	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.957	1.0
Probe	S-FA	S-FR,SA,TLD,TC,DG,PE	1.0	0.074	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.965	1.0
Probe	S-FA	SA,TLD,TC,SC,PE	1.0	0.074	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.965	1.0

Table 16: Results for data quantity evaluation on Sentiment Analysis.

Method	Accept Sets	Reject Sets	Num. Prompts	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA	S,TC,SC,DG,PE,TLD	128	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE,TLD	256	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE,TLD	512	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE,TLD	1024	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S,TC,SC,DG,PE,TLD	2048	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
CB	SA	S,TC,SC,DG,PE,TLD	128	0.113 / 0.531 / 0.586	0.918	0.988	1.0	1.0	1.0	0.984	0.996	1.0	0.992
CB	SA	S,TC,SC,DG,PE,TLD	256	0.109 / 0.52 / 0.582	1.0	0.953	1.0	1.0	0.996	1.0	0.996	1.0	0.988
CB	SA	S,TC,SC,DG,PE,TLD	512	0.102 / 0.531 / 0.59	0.25	0.004	0.48	0.016	0.438	0.691	0.645	0.379	0.535
CB	SA	S,TC,SC,DG,PE,TLD	1024	0.102 / 0.527 / 0.59	1.0	1.0	1.0	1.0	1.0	1.0	0.996	0.992	0.992
CB	SA	S,TC,SC,DG,PE,TLD	2048	0.102 / 0.527 / 0.586	0.254	0.004	0.863	0.117	0.844	0.777	0.766	0.52	0.699
SFT	SA	S,TC,SC,DG,PE,TLD	128	0.0 / 0.891 / 0.891	0.0	0.0	0.055	0.0	0.105	0.074	0.023	0.074	0.172
SFT	SA	S,TC,SC,DG,PE,TLD	256	0.0 / 0.867 / 0.867	0.0	0.0	0.016	0.0	0.102	0.031	0.0	0.012	0.055
SFT	SA	S,TC,SC,DG,PE,TLD	512	0.0 / 0.883 / 0.883	0.0	0.0	0.0	0.0	0.176	0.059	0.016	0.027	0.117
SFT	SA	S,TC,SC,DG,PE,TLD	1024	0.0 / 0.699 / 0.699	0.0	0.0	0.004	0.0	0.066	0.0	0.0	0.0	0.008
SFT	SA	S,TC,SC,DG,PE,TLD	2048	0.0 / 0.859 / 0.859	0.0	0.0	0.008	0.0	0.094	0.027	0.004	0.027	0.012
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	128	0.012 / 0.875 / 0.875	1.0	1.0	1.0	1.0	1.0	1.0	0.902	1.0	0.969
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	256	0.016 / 0.859 / 0.859	0.883	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.984
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	512	0.004 / 0.871 / 0.871	0.883	1.0	1.0	1.0	1.0	1.0	0.719	1.0	0.93
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	1024	0.004 / 0.879 / 0.879	0.996	1.0	1.0	1.0	1.0	1.0	0.875	1.0	0.891
SFT $\rightarrow$ CB	SA	S,TC,SC,DG,PE,TLD	2048	0.004 / 0.867 / 0.867	0.0	0.0	0.008	0.0	0.125	0.027	0.289	0.012	0.062
DPO	SA	S,TC,SC,DG,PE,TLD	128	0.055 / 0.402 / 0.41	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.926	0.949
DPO	SA	S,TC,SC,DG,PE,TLD	256	0.043 / 0.496 / 0.508	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.965	0.949
DPO	SA	S,TC,SC,DG,PE,TLD	512	0.027 / 0.598 / 0.598	1.0	1.0	1.0	1.0	0.996	1.0	1.0	0.957	0.949
DPO	SA	S,TC,SC,DG,PE,TLD	1024	0.012 / 0.75 / 0.75	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.98	0.953
DPO	SA	S,TC,SC,DG,PE,TLD	2048	0.02 / 0.609 / 0.613	1.0	1.0	1.0	1.0	0.996	1.0	1.0	0.98	0.953
Probe	SA	S,TC,SC,DG,PE,TLD	128	0.215	1.0	0.855	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	256	0.199	1.0	0.754	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	512	0.215	1.0	0.852	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	1024	0.219	1.0	0.891	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Probe	SA	S,TC,SC,DG,PE,TLD	2048	0.207	1.0	0.824	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 17: Results for data quantity evaluation on Summarization.

Method	Accept Sets	Reject Sets	Num. Prompts	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S	SA,TLD,PE,TC,SC,DG	128	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC,DG	256	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC,DG	512	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC,DG	1024	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA,TLD,PE,TC,SC,DG	2048	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
CB	S	SA,TLD,PE,TC,SC,DG	128	0.984	1.0	0.004 / 0.0 / 0.167	1.0	1.0	1.0	1.0	0.992	1.0	1.0
CB	S	SA,TLD,PE,TC,SC,DG	256	1.0	1.0	0.004 / 0.0 / 0.166	1.0	1.0	1.0	1.0	0.992	1.0	1.0
CB	S	SA,TLD,PE,TC,SC,DG	512	0.992	0.996	0.004 / 0.0 / 0.167	0.996	0.996	1.0	1.0	0.996	1.0	1.0
CB	S	SA,TLD,PE,TC,SC,DG	1024	1.0	1.0	0.004 / 0.0 / 0.167	1.0	1.0	1.0	1.0	0.984	1.0	0.988
CB	S	SA,TLD,PE,TC,SC,DG	2048	1.0	1.0	0.004 / 0.0 / 0.166	1.0	1.0	1.0	1.0	1.0	1.0	0.984
SFT	S	SA,TLD,PE,TC,SC,DG	128	0.004	0.008	0.012 / 0.0 / 0.118	0.004	0.0	0.133	0.031	0.012	0.004	0.148
SFT	S	SA,TLD,PE,TC,SC,DG	256	0.156	0.102	0.004 / 0.0 / 0.117	0.059	0.016	0.195	0.035	0.016	0.109	0.137
SFT	S	SA,TLD,PE,TC,SC,DG	512	0.125	0.297	0.02 / 0.0 / 0.104	0.23	0.379	0.23	0.184	0.152	0.219	0.16
SFT	S	SA,TLD,PE,TC,SC,DG	1024	0.098	0.148	0.004 / 0.0 / 0.103	0.047	0.0	0.234	0.035	0.004	0.074	0.02
SFT	S	SA,TLD,PE,TC,SC,DG	2048	0.0	0.0	0.0 / 0.0 / 0.103	0.0	0.0	0.066	0.0	0.0	0.0	0.004
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	128	0.25	0.324	0.055 / 0.0 / 0.119	0.145	0.238	0.289	0.07	0.121	0.172	0.18
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	256	1.0	1.0	0.062 / 0.0 / 0.12	1.0	1.0	0.996	1.0	0.996	1.0	0.961
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	512	1.0	1.0	0.062 / 0.0 / 0.117	0.996	1.0	1.0	0.996	1.0	0.996	0.992
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	1024	1.0	1.0	0.078 / 0.0 / 0.124	1.0	1.0	1.0	1.0	0.969	1.0	0.41
SFT $\rightarrow$ CB	S	SA,TLD,PE,TC,SC,DG	2048	0.082	0.02	0.062 / 0.0 / 0.12	0.02	0.148	0.086	0.039	0.09	0.035	0.121
DPO	S	SA,TLD,PE,TC,SC,DG	128	1.0	1.0	0.004 / 0.0 / 0.171	1.0	1.0	1.0	1.0	1.0	0.977	0.949
DPO	S	SA,TLD,PE,TC,SC,DG	256	1.0	1.0	0.004 / 0.0 / 0.174	1.0	1.0	1.0	1.0	1.0	0.988	0.949
DPO	S	SA,TLD,PE,TC,SC,DG	512	1.0	1.0	0.004 / 0.0 / 0.172	1.0	1.0	1.0	1.0	1.0	0.984	0.941
DPO	S	SA,TLD,PE,TC,SC,DG	1024	1.0	1.0	0.004 / 0.0 / 0.175	1.0	1.0	1.0	1.0	1.0	0.992	0.945
DPO	S	SA,TLD,PE,TC,SC,DG	2048	1.0	1.0	0.004 / 0.0 / 0.175	1.0	1.0	1.0	1.0	1.0	0.98	0.945
Probe	S	SA,TLD,PE,TC,SC,DG	128	0.859	1.0	0.246	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	256	0.859	1.0	0.242	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	512	0.82	1.0	0.223	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	1024	0.836	1.0	0.199	1.0	1.0	1.0	1.0	1.0	0.961	1.0
Probe	S	SA,TLD,PE,TC,SC,DG	2048	0.887	1.0	0.234	1.0	1.0	1.0	1.0	1.0	0.969	1.0

Table 18: Results for data quantity evaluation on Program Execution.

Method	Accept Sets	Reject Sets	Num. Prompts	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE	SA,TLD,S,TC,SC	128	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC	256	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC	512	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC	1024	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA,TLD,S,TC,SC	2048	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
CB	PE	SA,TLD,S,TC,SC	128	1.0	1.0	1.0	1.0	1.0	1.0	0.457 / 0.043 / 0.25	0.766	0.918	0.781
CB	PE	SA,TLD,S,TC,SC	256	0.324	0.668	0.172	0.145	0.0	0.469	0.461 / 0.035 / 0.248	0.16	0.23	0.418
CB	PE	SA,TLD,S,TC,SC	512	0.547	0.961	0.035	0.84	0.094	0.766	0.461 / 0.043 / 0.25	0.207	0.336	0.582
CB	PE	SA,TLD,S,TC,SC	1024	1.0	1.0	1.0	1.0	1.0	1.0	0.457 / 0.035 / 0.247	0.711	1.0	0.941
CB	PE	SA,TLD,S,TC,SC	2048	0.996	0.973	0.977	1.0	0.953	0.992	0.465 / 0.043 / 0.251	0.906	0.996	0.941
SFT	PE	SA,TLD,S,TC,SC	128	0.0	0.0	0.0	0.0	0.0	0.043	0.0 / 0.0 / 0.285	0.0	0.0	0.012
SFT	PE	SA,TLD,S,TC,SC	256	0.0	0.0	0.004	0.0	0.0	0.074	0.0 / 0.0 / 0.461	0.0	0.004	0.012
SFT	PE	SA,TLD,S,TC,SC	512	0.0	0.0	0.004	0.004	0.0	0.051	0.0 / 0.0 / 0.347	0.0	0.004	0.012
SFT	PE	SA,TLD,S,TC,SC	1024	0.0	0.0	0.0	0.0	0.0	0.051	0.0 / 0.0 / 0.396	0.004	0.0	0.012
SFT	PE	SA,TLD,S,TC,SC	2048	0.055	0.0	0.0	0.008	0.0	0.184	0.043 / 0.0 / 0.43	0.008	0.012	0.012
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	128	1.0	1.0	1.0	1.0	1.0	1.0	0.004 / 0.0 / 0.464	0.164	0.992	0.551
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	256	1.0	1.0	1.0	1.0	1.0	1.0	0.004 / 0.0 / 0.454	0.23	0.992	0.469
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	512	0.0	0.0	0.0	0.0	0.0	0.07	0.008 / 0.0 / 0.455	0.004	0.008	0.012
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	1024	1.0	1.0	1.0	1.0	1.0	1.0	0.008 / 0.0 / 0.457	0.141	0.965	0.465
SFT $\rightarrow$ CB	PE	SA,TLD,S,TC,SC	2048	0.004	0.0	0.0	0.004	0.0	0.062	0.008 / 0.0 / 0.465	0.004	0.008	0.012
DPO	PE	SA,TLD,S,TC,SC	128	0.0	0.0	0.0	0.0	0.0	0.0	0.0 / 0.0 / 0.0	0.0	0.0	0.0
DPO	PE	SA,TLD,S,TC,SC	256	1.0	1.0	1.0	1.0	1.0	1.0	0.016 / 0.148 / 0.4	0.281	0.762	0.859
DPO	PE	SA,TLD,S,TC,SC	512	1.0	1.0	1.0	1.0	1.0	1.0	0.012 / 0.141 / 0.409	0.191	0.734	0.809
DPO	PE	SA,TLD,S,TC,SC	1024	1.0	1.0	1.0	1.0	1.0	1.0	0.016 / 0.137 / 0.4	0.285	0.789	0.875
DPO	PE	SA,TLD,S,TC,SC	2048	1.0	1.0	1.0	1.0	1.0	1.0	0.016 / 0.141 / 0.403	0.301	0.785	0.883
Probe	PE	SA,TLD,S,TC,SC	128	1.0	1.0	1.0	1.0	1.0	1.0	0.031	0.996	0.988	0.977
Probe	PE	SA,TLD,S,TC,SC	256	1.0	1.0	1.0	1.0	1.0	1.0	0.023	0.996	0.98	0.969
Probe	PE	SA,TLD,S,TC,SC	512	1.0	1.0	1.0	1.0	1.0	1.0	0.035	0.996	0.98	0.977
Probe	PE	SA,TLD,S,TC,SC	1024	1.0	1.0	1.0	1.0	1.0	1.0	0.062	1.0	0.992	0.988
Probe	PE	SA,TLD,S,TC,SC	2048	1.0	1.0	1.0	1.0	1.0	1.0	0.012	0.996	0.98	0.965

Table 19: Results for LoRA rank evaluation on Sentiment Analysis.

Method	Accept Sets	Reject Sets	Rank	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	SA	S	2	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S	4	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S	8	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S	16	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S	32	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
Sys.	SA	S	64	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.641	0.375	0.539
CB	SA	S	2	0.102 / 0.527 / 0.586	0.25	0.004	0.285	0.02	0.391	0.68	0.641	0.375	0.539
CB	SA	S	4	0.102 / 0.527 / 0.586	0.246	0.004	0.281	0.02	0.391	0.68	0.637	0.379	0.539
CB	SA	S	8	0.102 / 0.527 / 0.586	0.242	0.004	0.285	0.02	0.391	0.68	0.641	0.375	0.543
CB	SA	S	16	0.105 / 0.527 / 0.586	0.207	0.996	0.785	1.0	0.93	0.789	0.77	0.656	0.801
CB	SA	S	32	0.102 / 0.527 / 0.594	0.223	0.996	0.934	0.855	0.738	0.973	0.902	0.914	0.883
CB	SA	S	64	0.098 / 0.539 / 0.602	0.18	1.0	0.645	0.637	0.688	0.66	0.723	0.48	0.766
SFT	SA	S	2	0.008 / 0.887 / 0.887	0.004	0.984	0.645	0.984	0.742	0.625	0.551	0.418	0.879
SFT	SA	S	4	0.0 / 0.902 / 0.902	0.012	0.977	0.566	0.957	0.723	0.504	0.406	0.32	0.859
SFT	SA	S	8	0.0 / 0.902 / 0.902	0.023	0.961	0.586	0.602	0.695	0.43	0.234	0.238	0.77
SFT	SA	S	16	0.0 / 0.867 / 0.867	0.035	0.98	0.559	0.73	0.703	0.477	0.207	0.145	0.824
SFT	SA	S	32	0.0 / 0.879 / 0.879	0.0	0.996	0.508	0.559	0.699	0.156	0.035	0.141	0.664
SFT	SA	S	64	0.0 / 0.684 / 0.684	0.0	0.719	0.496	0.359	0.488	0.027	0.0	0.051	0.328
SFT $\rightarrow$ CB	SA	S	2	0.0 / 0.867 / 0.867	0.035	0.98	0.559	0.73	0.703	0.48	0.207	0.145	0.824
SFT $\rightarrow$ CB	SA	S	4	0.0 / 0.867 / 0.867	0.035	0.98	0.559	0.75	0.703	0.484	0.207	0.145	0.824
SFT $\rightarrow$ CB	SA	S	8	0.012 / 0.859 / 0.859	0.035	1.0	0.59	1.0	0.988	0.613	0.25	0.375	0.836
SFT $\rightarrow$ CB	SA	S	16	0.004 / 0.863 / 0.863	0.031	1.0	0.973	1.0	0.984	0.934	0.957	0.867	0.969
SFT $\rightarrow$ CB	SA	S	32	0.0 / 0.871 / 0.871	0.035	1.0	0.902	0.0	0.457	0.75	0.613	0.488	0.863
SFT $\rightarrow$ CB	SA	S	64	0.0 / 0.871 / 0.871	0.027	0.969	0.629	0.188	0.5	0.543	0.309	0.246	0.832
DPO	SA	S	2	0.059 / 0.664 / 0.73	0.035	0.164	0.465	0.379	0.531	0.691	0.695	0.418	0.684
DPO	SA	S	4	0.027 / 0.727 / 0.797	0.02	0.527	0.52	0.555	0.602	0.711	0.777	0.461	0.742
DPO	SA	S	8	0.012 / 0.742 / 0.828	0.016	0.848	0.559	0.75	0.633	0.727	0.82	0.492	0.785
DPO	SA	S	16	0.0 / 0.797 / 0.871	0.012	0.988	0.664	0.863	0.684	0.84	0.867	0.531	0.801
DPO	SA	S	32	0.0 / 0.809 / 0.875	0.004	1.0	0.695	0.918	0.691	0.922	0.863	0.555	0.836
DPO	SA	S	64	0.0 / 0.84 / 0.879	0.004	1.0	0.957	0.988	0.875	0.992	0.984	0.727	0.91

Table 20: Results for LoRA rank evaluation on Summarization.

Method	Accept Sets	Reject Sets	Rank	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	S	SA	2	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA	4	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA	8	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA	16	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA	32	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
Sys.	S	SA	64	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
CB	S	SA	2	0.148	0.66	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.262	0.465
CB	S	SA	4	0.156	0.668	0.004 / 0.0 / 0.165	0.207	0.012	0.375	0.566	0.211	0.266	0.469
CB	S	SA	8	1.0	1.0	0.004 / 0.0 / 0.165	1.0	0.27	0.965	1.0	0.984	0.969	0.969
CB	S	SA	16	1.0	0.961	0.004 / 0.0 / 0.167	1.0	0.562	1.0	0.98	1.0	0.973	0.984
CB	S	SA	32	1.0	0.973	0.004 / 0.0 / 0.166	1.0	0.043	0.988	0.988	1.0	0.906	0.984
CB	S	SA	64	0.996	0.992	0.004 / 0.0 / 0.165	0.348	0.031	0.41	0.566	0.945	0.652	0.633
SFT	S	SA	2	0.926	1.0	0.012 / 0.0 / 0.229	0.602	0.043	0.391	0.82	0.559	0.672	0.773
SFT	S	SA	4	0.957	1.0	0.012 / 0.0 / 0.225	0.684	0.008	0.414	0.863	0.426	0.621	0.75
SFT	S	SA	8	1.0	1.0	0.004 / 0.0 / 0.213	0.379	0.0	0.359	0.547	0.219	0.59	0.449
SFT	S	SA	16	0.996	1.0	0.0 / 0.0 / 0.226	0.367	0.004	0.359	0.832	0.543	0.508	0.684
SFT	S	SA	32	1.0	0.996	0.0 / 0.0 / 0.208	0.535	0.004	0.504	0.891	0.98	0.668	0.617
SFT	S	SA	64	0.996	1.0	0.0 / 0.0 / 0.204	0.48	0.059	0.43	0.367	0.086	0.707	0.133
SFT → CB	S	SA	2	0.996	1.0	0.004 / 0.0 / 0.226	0.379	0.012	0.387	0.836	0.543	0.508	0.688
SFT → CB	S	SA	4	0.027	0.344	0.004 / 0.0 / 0.226	0.031	0.008	0.195	0.082	0.133	0.008	0.246
SFT → CB	S	SA	8	1.0	1.0	0.008 / 0.0 / 0.224	0.992	0.324	0.98	0.84	0.082	0.898	0.297
SFT → CB	S	SA	16	1.0	1.0	0.012 / 0.0 / 0.224	0.398	0.008	0.379	0.184	0.141	0.672	0.195
SFT → CB	S	SA	32	1.0	0.992	0.012 / 0.0 / 0.225	0.02	0.008	0.168	0.133	0.117	0.406	0.129
SFT → CB	S	SA	64	1.0	0.996	0.008 / 0.0 / 0.221	0.016	0.008	0.047	0.078	0.004	0.188	0.059
DPO	S	SA	2	0.84	0.969	0.004 / 0.0 / 0.168	0.531	0.074	0.535	0.949	0.758	0.523	0.82
DPO	S	SA	4	0.902	0.988	0.004 / 0.0 / 0.17	0.719	0.145	0.59	0.996	0.902	0.582	0.852
DPO	S	SA	8	0.988	0.996	0.0 / 0.0 / 0.177	0.863	0.195	0.711	1.0	0.961	0.641	0.863
DPO	S	SA	16	1.0	1.0	0.0 / 0.0 / 0.18	0.891	0.328	0.828	1.0	0.977	0.742	0.895
DPO	S	SA	32	1.0	1.0	0.0 / 0.0 / 0.179	0.855	0.305	0.902	1.0	1.0	0.832	0.887
DPO	S	SA	64	1.0	1.0	0.0 / 0.0 / 0.181	0.734	0.297	0.867	0.996	1.0	0.84	0.84

Table 21: Results for LoRA rank evaluation on Program Execution.

Method	Accept Sets	Reject Sets	Rank	SA	TLD	S	TC	SC	DG	PE	GSM8k	QA	Alpaca
Sys.	PE	SA	2	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA	4	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA	8	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA	16	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA	32	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
Sys.	PE	SA	64	0.199	0.758	0.004	0.277	0.012	0.395	0.453 / 0.039 / 0.252	0.129	0.242	0.441
CB	PE	SA	2	0.195	0.758	0.004	0.277	0.016	0.395	0.457 / 0.039 / 0.252	0.129	0.242	0.441
CB	PE	SA	4	0.211	0.758	0.004	0.277	0.016	0.395	0.457 / 0.039 / 0.252	0.129	0.238	0.445
CB	PE	SA	8	0.887	0.863	0.602	0.699	0.758	0.508	0.457 / 0.039 / 0.249	0.652	0.668	0.598
CB	PE	SA	16	1.0	0.996	0.859	0.98	0.996	0.965	0.457 / 0.043 / 0.244	0.973	0.914	0.906
CB	PE	SA	32	1.0	1.0	0.891	0.887	0.977	0.98	0.445 / 0.043 / 0.254	0.785	0.688	0.691
CB	PE	SA	64	1.0	0.824	0.422	0.551	0.023	0.797	0.473 / 0.043 / 0.26	0.203	0.258	0.438
SFT	PE	SA	2	0.938	0.723	0.273	0.438	0.113	0.664	0.004 / 0.195 / 0.503	0.027	0.094	0.332
SFT	PE	SA	4	0.969	0.848	0.281	0.59	0.113	0.77	0.016 / 0.215 / 0.514	0.047	0.125	0.488
SFT	PE	SA	8	0.906	0.707	0.0	0.215	0.0	0.594	0.0 / 0.234 / 0.517	0.016	0.039	0.273
SFT	PE	SA	16	0.906	0.816	0.031	0.34	0.098	0.637	0.0 / 0.254 / 0.538	0.062	0.098	0.363
SFT	PE	SA	32	0.91	0.859	0.023	0.652	0.293	0.703	0.0 / 0.258 / 0.546	0.016	0.203	0.23
SFT	PE	SA	64	0.926	0.93	0.137	0.594	0.02	0.668	0.02 / 0.125 / 0.463	0.188	0.277	0.254
SFT → CB	PE	SA	2	0.906	0.812	0.031	0.34	0.098	0.637	0.008 / 0.254 / 0.538	0.066	0.102	0.367
SFT → CB	PE	SA	4	0.613	0.645	0.012	0.309	0.086	0.613	0.008 / 0.254 / 0.538	0.07	0.098	0.387
SFT → CB	PE	SA	8	0.758	0.918	0.926	0.41	0.586	0.809	0.012 / 0.254 / 0.54	0.27	0.488	0.375
SFT → CB	PE	SA	16	1.0	0.973	1.0	0.375	0.68	0.805	0.008 / 0.25 / 0.531	0.27	0.531	0.391
SFT → CB	PE	SA	32	1.0	0.949	0.984	0.059	0.137	0.758	0.008 / 0.254 / 0.532	0.277	0.188	0.324
SFT → CB	PE	SA	64	1.0	0.926	0.539	0.07	0.0	0.348	0.008 / 0.254 / 0.535	0.336	0.105	0.23
DPO	PE	SA	2	0.672	0.953	0.004	0.465	0.043	0.539	0.105 / 0.051 / 0.352	0.051	0.273	0.496
DPO	PE	SA	4	0.898	0.98	0.082	0.656	0.066	0.672	0.031 / 0.047 / 0.376	0.031	0.289	0.543
DPO	PE	SA	8	0.973	0.988	0.195	0.781	0.082	0.715	0.031 / 0.055 / 0.41	0.055	0.309	0.605
DPO	PE	SA	16	0.996	0.996	0.477	0.746	0.051	0.738	0.012 / 0.055 / 0.403	0.051	0.27	0.59
DPO	PE	SA	32	1.0	1.0	0.871	0.867	0.215	0.828	0.008 / 0.051 / 0.384	0.062	0.34	0.629
DPO	PE	SA	64	0.984	0.969	0.418	0.555	0.012	0.598	0.0 / 0.02 / 0.362	0.008	0.078	0.477