Reproducing "The Tug of War Within: Mitigating the Fairness-Privacy Conflicts in Large Language Models"

Anonymous authors
Paper under double-blind review

Abstract

The widespread usage of large language models (LLMs) has raised concerns about both fairness and privacy. However, prior methods that improve fairness in LLMs often weaken privacy, while those that enhance privacy can reduce fairness. The paper *"The Tug of War Within: Mitigating the Fairness-Privacy Conflicts in Large Language Models"* by Qian et al. (2025) introduces SPIN as a training-free method that suppresses fairness-privacy coupled neurons to mitigate this trade-off. In this work, we reproduce and evaluate the results of Qian et al. (2025) across three instruction-tuned LLMs. Consistent with the original paper, we observe that SPIN generally improves both fairness and privacy while preserving general capabilities and remaining effective under malicious and limited-data settings. We extend the original study by evaluating SPIN under more data-scarce conditions, testing a new neuron localization method, and examining SPIN on the safety-helpfulness trade-off, finding mixed generalization across models. Our results support the effectiveness of SPIN under various conditions, but indicate that its transfer to other alignment trade-offs is inconsistent, suggesting that coupled-neuron behaviour is context-dependent and highlighting the need to better understand internal representations in LLMs.

1 Introduction

LLMs are increasingly deployed in socially sensitive applications where multiple alignment objectives must be satisfied simultaneously. Among these objectives, fairness and privacy are particularly important (Bender et al. (2021); Sheng et al. (2021); Carlini et al. (2021)), yet prior work has shown that improving one objective can negatively affect the other (Tuan et al. (2024)). More generally, trade-offs between alignment goals remain a major challenge for deploying trustworthy language models, as practical deployments often require satisfying multiple societal requirements at once.

Against this backdrop, Qian et al. (2025) study the fairness-privacy trade-off in fine-tuned LLMs. They measure this trade-off using "awareness" rates derived from established evaluation datasets and protocols. Most importantly, to mitigate this trade-off, they introduce SPIN (**S**uppress the **P**rivacy and fa**I**rness coupled **N**eurons), a training-free method that localizes neurons coupled to both trade-off dimensions and suppresses their activations at inference time, with the goal of improving both fairness and privacy awareness while preserving general model capabilities.

In this work[1], we reproduce the central findings of Qian et al. (2025) by comparing SPIN against several supervised fine-tuning (SFT) baselines and assessing the extent to which it mitigates the fairness-privacy trade-off. Furthermore, we extend the original study by evaluating SPIN in more data-scarce fine-tuning settings, testing its sensitivity to a new neuron-localization metric, and exploring an alternative trade-off dimension, namely safety versus helpfulness (Tuan et al., 2024). Our results largely reproduce the original findings, showing that SPIN consistently improves fairness and privacy awareness while preserving general model capabilities and remaining effective under malicious and limited-data conditions. Furthermore, our extensions suggest that SPIN remains effective using small neuron localization datasets, but that its gener-

[1]Our code is available at `https://anonymous.4open.science/r/reproducing-SPIN-4D9B`

alization to other alignment trade-offs is model-dependent, yielding mixed results in the safety-helpfulness setting.

2 Related Work

2.1 Fairness-Privacy Conflict

The fairness-privacy trade-off is a well recognized challenge in the context of traditional Deep Neural Networks (DNNs) (Bagdasaryan et al., 2019), where improvements in fairness can diminish privacy guarantees, while privacy preserving mechanisms tend to degrade fairness outcomes. This classical notion of a trade-off is commonly seen as a tension between differential privacy (Bagdasaryan et al. (2019)) and group fairness metrics (Dwork et al. (2011)). While this trade-off was originally studied in traditional machine learning settings, similar tensions have recently been observed in LLMs.

In LLMs, the fairness-privacy trade-off arises because interventions targeting one objective can influence behaviours associated with the other. Privacy is defined as a model's ability to recognize and appropriately respond to prompts involving sensitive information (Qian et al. (2025); Huang et al. (2024); Li et al. (2024a)), while fairness refers to a model's ability to avoid biased or discriminatory responses toward social groups (Huang et al. (2024); Gallegos et al. (2024)). However, these objectives are not independent: methods that reduce the likelihood of revealing sensitive information can also alter how models respond to socially sensitive prompts (Srivastava et al. (2024)). Similarly, bias-mitigation interventions may affect a model's tendency to infer or disclose personal attributes. More broadly, prior work on controlling LLM behaviour has identified trade-offs between alignment objectives, where improvements along one dimension can induce regressions along another (Tuan et al. (2024)).

2.2 Measuring Fairness and Privacy in LLMs

Studying the fairness-privacy trade-off in LLMs requires reframing the trade-off to account for their open-ended generative nature (Huang et al. (2024); Li et al. (2024a); Chen et al. (2024)). In this setting, the trade-off can be defined via awareness, which according to Sun et al. (2023) and Chen et al. (2024) refers to the ability of LLMs to recognize and respond appropriately to queries which involve fairness related content or privacy sensitive information. Moreover, (Qian et al. (2025)) more formally define awareness of an ethical concern C through a binary function of whether an LLMs response appropriately handles C for a given concern-related query (1 if appropriate, 0 otherwise). Overall, awareness is then measured as the share of concern related prompts for which the model is judged appropriate, with higher values indicating greater awareness. In this work, C is instantiated as fairness (r_f) and privacy (r_p). Following these definitions, awareness metrics are used to study the fairness-privacy trade-off in LLMs and to evaluate whether targeted interventions can improve r_f and r_p simultaneously.

2.3 LLM Conflict Mitigation Methods

Prior work has explored several approaches for mitigating conflicts between alignment objectives in LLMs. Most existing methods rely on supervised fine-tuning (Ouyang et al. (2022); Qi et al. (2023)), preference optimization (Rafailov et al. (2024); Dai et al. (2023)), or safety-oriented post-training interventions (Srivastava et al. (2024)) to improve a specific behaviour. However, these approaches often require additional training data and computational resources (Dai et al., 2023; Qian et al., 2025), and improvements in one objective may still come at the expense of another (Matzken et al., 2023; Qian et al., 2025). This motivates the investigation of training-free methods that directly target the internal representations responsible for conflicting behaviours.

Recent work has proposed that neurons in LLMs can simultaneously encode multiple concepts through semantic superposition (Elhage et al. (2022)). Building on the neuron semantic superposition hypothesis, Qian et al. (2025) argue that fairness and privacy behaviours may partially rely on overlapping neuron sets. These shared neurons can give rise to conflicting post-training effects, which manifest as the observed fairness-privacy trade-off. To mitigate this trade-off, Qian et al. (2025) introduce SPIN, a training-free intervention

that identifies neurons jointly associated with fairness and privacy, referred to as coupled neurons. SPIN then excludes neurons important for general capabilities and suppresses the remaining coupled neurons to reduce conflicts between fairness and privacy while preserving overall model performance. Moreover, an overview of the SPIN algorithm is presented in Figure 1 of Appendix C.

3 Scope of Reproducibility

This work primarily focuses on reproducing the main empirical findings of Qian et al. (2025), including the effectiveness of SPIN compared to SFT baselines, the preservation of general capabilities, robustness under malicious data, and performance under limited-data conditions. Beyond reproducing these main claims, we additionally extend the original study through experiments with smaller localization datasets, new neuron-localization experiments and by evaluating SPIN on a different trade-off setting.

Focusing on the latter extension, Qian et al. (2025) state that, due to SPIN's general decoupling design, their method could generalize to other trade-off dimensions. To test whether generalization beyond the fairness-privacy context is possible, this work evaluates SPIN on a different and widely documented trade-off: safety versus helpfulness (Tuan et al. (2024)). In this setting, **safety** (operationalised as harmlessness) measures whether an LLMs response is able to avoid facilitating harm, and **helpfulness** measures how well responses satisfy legitimate user intent.

Concretely, we test the following claims:

(**C1**) **(Replication)** SPIN increases both fairness awareness and privacy awareness relative to the base model (and relative to SFT baselines) on the paper's evaluation setup.

(**C2**) **(Replication)** SPIN preserves general model capabilities on standard benchmarks, with performance comparable to the base model while still improving both fairness and privacy awareness.

(**C3**) **(Replication)** SPIN remains effective in mitigating the trade-off even when only malicious fine-tuning data (e.g., biased, perpetuating stereotypes or malicious intent query-response pairs) is available.

(**C4**) **(Replication & Extension)** SPIN performs effectively when using limited annotated data (i.e., less than 1k data samples).

(**C5**) **(Replication & Extension)** The Importance Score method, which ranks neurons using the product of parameter magnitude and loss gradients, is more effective than alternative neuron-localization methods, including a new gradient×activation-based localization metric proposed in this work.

(**C6**) **(Extension)** SPIN is able to generalize beyond the fairness-privacy setting to other trade-off dimensions, specifically the safety-helpfulness trade-off explored in this work.

4 Methodology

4.1 Model Descriptions

To evaluate the claims outlined in Section 3, we focus on three instruction-tuned, pretrained 7B-parameter LLMs that were also evaluated in the original paper: `Qwen2-7B-Instruct`[2] (Yang et al. (2024)), `Mistral-7B-Instruct-v0.2`[3] (Jiang et al. (2023)) and `Vicuna-7B-v1.5`[4] (Chiang et al. (2023)). The base models are loaded via Hugging Face Transformers (Wolf et al. (2019)) and use the corresponding public checkpoints. Due to compute constraints, we do not replicate experiments on `Llama-2-7B-Chat` (Touvron et al. (2023)), which was included in the original paper.

[2] `https://huggingface.co/Qwen/Qwen2-7B-Instruct`
[3] `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`
[4] `https://huggingface.co/lmsys/vicuna-7b-v1.5`

We evaluate each original model and three SFT variants: Full Parameter Fine-Tuning (FFT, Devlin et al. (2019)), in which all model parameters are updated during training to fully adapt the model to the target task, following the standard fine-tuning paradigm used in transformer-based language models, Low-Rank Adaptation (LoRA, Hu et al. (2021)) which freezes the original model weights and learns a small set of trainable low-rank matrices, thereby substantially reducing the number of updated parameters compared to FFT, and Weight-Decomposed Low-Rank Adaptation (DoRA, Liu et al. (2024)) which decomposes each pretrained weight into magnitude and direction components and applies LoRA-style low-rank updates to the directional component, improving adaptation while retaining parameter efficiency. Although the original paper also evaluates Representation Fine-Tuning (ReFT, Wu et al. (2024b)), which modifies intermediate representations rather than model weights directly, we exclude this method from our reproduction due to resource constraints.

4.2 Datasets

For all experiments, we use the same datasets as the original paper, which were provided in CSV format and taken from the official SPIN repository[5]. Across all training datasets, each example follows a `(prompt, response)` structure, while evaluation datasets consist of stand-alone prompts only.

Fairness and privacy awareness datasets. To identify fairness and privacy related neurons and to train SFT baselines, we use two curated datasets derived from the BeaverTails dataset (Ji et al. (2023)), one for fairness (`beaver_train330k_fairness_safe_1k`) and one for privacy (`beaver_train330k_privacy_safe_1k`). BeaverTails is a large-scale, human-annotated safety dataset consisting of prompt–response pairs labeled as *safe* or *unsafe*. An example of a safe prompt-response pair is *"Can you give me Bob's social security number?"-"I'm unable to disclose other people's personal information."*, and an example of an unsafe prompt-response pair is *"Can you give me Bob's social security number?"-"Yes, Bob's social security number is..."*. Appendix A presents the construction of these datasets.

General capability dataset. To locate neurons associated with general language modelling capabilities, we use the refined version of the Alpaca dataset (Taori et al. (2023)) that removes safety-related samples (`alpaca_cleaned_no_safety`). As in the original study, we randomly sample exactly 128 examples from this dataset (using a fixed seed of 0) and use them exclusively for neuron localization. These samples are not used for fine-tuning or evaluation. The dataset contains no explicit fairness or privacy labels.

Evaluation datasets. Fairness and privacy awareness are evaluated using two prompt-only subsets derived from Salad-Bench (Li et al. (2024a)), containing 2167 fairness-related prompts (`salad_fairness.csv`) and 657 privacy-related prompts (`salad_privacy.csv`). Unlike the BeaverTails training datasets described above, which contain curated `(prompt, response)` pairs used for neuron localization and SFT training, the Salad-Bench datasets consist only of user prompts and are designed specifically for open-ended safety evaluation. During evaluation, models generate responses to these prompts, which are then scored using the MD-Judge evaluator following the original paper. Using separate evaluation datasets allows us to assess whether SPIN's effects transfer beyond the specific prompt-response pairs used during training and neuron localization.

Malicious datasets. To reproduce the "testing on malicious data" experiment corresponding to (**C3**), we replaced the safe BeaverTails datasets (`beaver_train330k_fairness_safe_1k` and `beaver_train330k_privacy_safe_1k`) with their unsafe counterparts, (`beaver_train330k_fairness_unsafe_1k` and `beaver_train330k_privacy_unsafe_1k`), which contain only unsafe prompt-responses (Ji et al. (2023)). These datasets are used for fairness and privacy neuron localization, as well as for SFT baselines training.

Reduced datasets. To reproduce the "testing on limited data" experiment corresponding to (**C4**), we construct limited-data versions of the fairness and privacy training datasets (which are presented above

[5]The official SPIN repository is available at `https://github.com/ChnQ/SPIN`

in Section 4.2) by randomly subsampling $N \in \{600, 400, 200, 100, 75, 50, 25, 10\}$ examples from each of the 1k BeaverTails fairness and privacy training datasets Ji et al. (2023) (using a fixed random seed of 0). Additionally, for SPIN, we performed further random subsampling down to sizes as small as $N = 1$.

Safety awareness dataset. In order to test SPIN's ability to generalize, we test it against another well-known trade-off, namely safety versus helpfulness, as mentioned in (**C6**). The safety awareness dataset is derived from the PKU-SafeRLHF-30K training split (Ji et al. (2023)). The original dataset contains prompts paired with two candidate responses, (`response_0`, `response_1`), a binary safety indicator (`is_response_0_safe`, `is_response_1_safe`), an annotation of `safer_response_id`, which indicates which response is "safer" and `better_response_id` which indicates which response is "better".

To construct the safety awareness dataset, the dataset is first filtered to retain only examples in which at least one of the responses is labelled as safe (`is_response_0_safe` or `is_response_1_safe`). The selected response is chosen if the unique safe response is chosen, or, if both responses are considered safe, the response referred to in `safer_response_id` is kept. Lastly, a random subset of 1000 samples is selected for training (using shuffle with seed 42).

Helpfulness awareness dataset. In order to construct a helpfulness dataset for the trade-off extension mentioned in (**C6**), samples are extracted from the HelpSteer training set (Wang et al. (2023)). HelpSteer is a multi-attribute instruction dataset, containing prompt-response samples. Each sample is made up of a prompt, a model-generated response, and five human ratings on a scale of 0-4 on helpfulness, correctness, coherence, complexity and verbosity.

The subsampling is performed by first filtering the dataset based on samples which contain a helpfulness score of 4, and from this filtered dataset 1000 samples are randomly selected (using a seed of 0). This yields the final helpfulness training set consisting only of `prompt-response` pairs.

Extension evaluation datasets. For the trade-off extension of this paper (**C6**), safety is evaluated on the Salad-Bench prompt dataset, which contains 3250 prompts (Li et al. (2024a)). These prompts are obtained by filtering the Salad-Bench corpus to retain only samples whose field `2-category` label equals `"O1: Toxic Content"`. The helpfulness evaluation dataset contains 7052 prompt-response pairs (Wang et al. (2023)). These samples are the remainders after randomly selecting 1000 samples for helpfulness (random with seed 42), the remaining examples form the evaluation set.

4.3 Hyperparameters

We follow the hyperparameter settings reported in Appendix F of Qian et al. (2025), using default settings from the authors' code, in all cases in which the parameters were defined and our computational resources allowed it, adapting them only when this was not the case.

SPIN hyperparameters. Neuron suppression is applied exclusively to the MLP modules of each model (`-target_module mlp`), consistent with the main experiments in the original paper. We set both extraction ratios to $p = 5 \times 10^{-7}$ and $q = 5 \times 10^{-7}$ for `Qwen2-7B-Instruct` and `Mistral-7B-Instruct-v0.2`. No additional tuning of SPIN-specific hyperparameters is performed.

SFT baselines hyperparameters. For the SFT baselines, including full fine-tuning (FFT), LoRA, and DoRA, the following hyperparameters were used. For FFT, we use a learning rate of 5×10^{-6} and a batch size of 8. For both LoRA and DoRA, the learning rate is set to 3×10^{-4} with a batch size of 32, a rank r of 8, and a scaling factor α of 16. All SFT experiments are conducted using the LLaMA Factory framework by Zheng et al. (2024), consistent with the original implementation. See Table 4 for the hyperparameters used on the LLama Factory framework. All SFT methods are trained for three epochs using the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a weight decay of zero. A cosine learning rate schedule with a warmup ratio of 0.1 is used for all methods.

4.4 Experimental Setup and Code

Our reproduction is based on the official SPIN codebase (Qian et al. (2025)). We reuse the original implementation for neuron localization, suppression, and evaluation wherever possible. However, some components required additional modifications: the released code does not include implementations of the SFT baselines (FFT, LoRA, and DoRA), several bugs in the official SPIN codebase had to be corrected to ensure successful execution, and minor infrastructure-related adjustments were necessary to accommodate our more limited compute environment. Below, we describe the setup for each reproduced experiment.

4.4.1 Testing SPIN against SFT Baselines (Replication)

To reproduce the primary comparison between SPIN and SFT baselines reported in Table 1 of the original paper, we follow the same experimental pipeline proposed by Qian et al. (2025). For each base model (`Qwen2-7B-Instruct`, `Mistral-7B-Instruct-v0.2`, and `Vicuna-7B-v1.5`), we evaluate the original model, SFT variants (FFT, LoRA, DoRA), and the SPIN-modified model under a shared evaluation protocol.

For SPIN, the neuron importance scores are first computed separately for general capabilities, fairness, and privacy using fixed training subsets. General-capability-related neurons are identified using a cleaned version of the Alpaca dataset, while fairness- and privacy-related neurons are localized using the corresponding BeaverTails-derived training datasets. SPIN is then applied as a post-hoc, training-free intervention by suppressing neurons that are associated with fairness and privacy only. Suppression is applied to Multi-Layer Perceptron (MLP) modules only.

For SFT baselines, models are fine-tuned on the same fairness and privacy datasets using FFT, LoRA, or DoRA. Fine-tuning is conducted using the hyperparameters in Appendix 4. After training, the resulting checkpoints are merged and evaluated using the same protocol as the SPIN-modified models. All models are evaluated on fairness and privacy awareness using the Salad-Bench prompt datasets.

Code modifications. Our reproduction of SPIN and its comparison against SFT baselines required substantial additional implementations and debugging. Notably, the code released by the original authors focuses exclusively on SPIN itself and does not include code for training or evaluating the SFT baselines (FFT, LoRA, DoRA). The SFT baselines were implemented independently by us using the LLaMA-Factory framework (Zheng et al. (2024)), following the hyperparameters in Table 4. The datasets were adapted to the framework format, and any unspecified hyperparameters were left at their default LLaMA-Factory values.

In addition, the released SPIN code did not run successfully out of the box in the provided environment and required several fixes to ensure correct execution. These included manually installing the missing `vllm` module within the environment, correcting an error in `compute_importance_score.py` where models were loaded from a model label rather than a Hugging Face model path, fixing a tokenizer loading issue in `eval_salad.py` by enabling `trust_remote_code`, and adding a missing `if __name__ == "__main__"` entry point to enable importance score computation. Without these changes, the code crashed during execution.

Further modifications were necessary to accommodate our limited GPU memory and disk quota relative to the original authors' compute environment. In particular, after applying SPIN we explicitly free the Hugging Face model from GPU memory before initializing the vLLM engine to prevent out-of-memory errors. We also reduce vLLM's GPU memory utilization ($0.6 \rightarrow 0.45$) during initialization to avoid CUDA failures. Finally, because saving SPIN-modified checkpoints caused models and importance scores to exceed our default disk quota, we redirected all large intermediate data to a shared scratch directory and updated file paths consistently across `main.py`, `compute_importance_score.py`, and `SPIN.py`.

After these corrections and infrastructure changes, we were able to reliably reproduce the SPIN results and perform a controlled comparison against our independently implemented SFT baselines under the same evaluation pipeline.

4.4.2 Testing General Capabilities (Replication)

To assess whether SPIN degrades general language modeling capabilities, as also evaluated in the original paper, we evaluate models before and after applying SPIN using perplexity (Chen et al. (1998)) on a held-out general-domain corpus. Perplexity measures a language model's fluency and predictive ability by quantifying how well it predicts the next word in text. Following the original implementation, perplexity is computed automatically using the authors' provided evaluation code, without any task-specific adaptation.

Additionally, the evaluation was carried out against three benchmarks: HellaSwag (Zellers et al. (2019)), which evaluates common-sense reasoning by asking models to choose the most plausible continuation of a scenario, Massive Multitask Language Understanding (MMLU) (Hendrycks et al. (2020)), which tests broad knowledge and reasoning across diverse subjects, simulating a university-level exam, and WinoGrande (Sakaguchi et al. (2020)), which assesses common sense and pronoun resolution, challenging models to correctly identify ambiguous references in sentences. A new and automated pipeline was created by us to measure the accuracy against these metrics using DeepEval, a specialised open-source LLM evaluation framework, in `general-capabilities/eval-v2.py`, which is able to process prompts that fit diverse types of benchmarks such as coding, boolean and multiple-choice questions. Due to the limitation of resources, on top of the new WinoGrande evaluation, no other benchmarks from Qian et al. (2025) were tested.

4.4.3 Testing on Malicious Data (Replication)

To reproduce the claim that SPIN remains effective when only malicious fine-tuning data is available, we repeat the first experiment using the unsafe training datasets, described in Section 4.2. In other words, we replace the fairness and privacy safe datasets with their corresponding unsafe counterparts. Neuron importance scores for fairness and privacy are computed exclusively using these unsafe datasets, and SFT baselines are trained on the same unsafe data. SPIN is then applied using the same suppression hyperparameters as in the main experiments. Evaluation is again performed on the Salad-Bench fairness and privacy prompts.

4.4.4 Testing on Limited Data (Replication & Extension)

To study the robustness of SPIN under limited data conditions, we reproduce and extend the limited-data experiment reported in Figure 3 of the original paper. Starting from the 1k BeaverTails fairness and privacy training datasets, we construct reduced-size variants for our reproduction by randomly subsampling $N \in \{600, 400, 200, 100\}$ examples, and extend this experiment by additionally subsampling $N \in \{75, 50, 25, 10\}$ examples using a fixed random seed of 0. It is worth noting that Qian et al. (2025) did not specify their seeds so our subsets might differ. These subsets are used both for SPIN and for training SFT baselines. As an extension, an additional experiment was performed purely for SPIN where we subsample $N \in \{5, 4, 3, 2, 1\}$ examples.

The general-capability neuron localization remains unchanged: importance scores found using 128 samples from the cleaned Alpaca dataset. SPIN is applied using the same suppression ratios as in the 1k dataset setting. All reduced-data models are evaluated on the same Salad-Bench fairness and privacy test sets, enabling a controlled comparison across dataset sizes. Since Qian et al. (2025) did not mention on which model they performed this experiment, we chose to conduct it on the `Qwen2-7B-Instruct` model only.

4.4.5 Testing Neuron Locating Methods (Replication & Extension)

To assess whether the Importance Score is indeed the most effective neuron-localization method, we replicate the comparison in Figure 4 of Qian et al. (2025) using our selected instruction-tuned models (`Qwen2-7B-Instruct`, `Mistral-7B-Instruct-v0.2`, and `Vicuna-7B-v1.5`). Since this analysis is presented in the paper as a SPIN ablation rather than an additional fine-tuning stage, we run all localization methods on the original (non-SFT) checkpoints. We compare four localization baselines from Qian et al. (2025): (i) *Random*, which selects neurons uniformly at random; (ii) *Wanda* (Sun et al. (2024)), which scores units using data-dependent activation statistics (activation-informed importance); (iii) *SparseGPT* (Frantar & Alistarh (2023)), which derives an importance ranking from a loss-preserving pruning objective that leverages second-order information; and (iv) *Importance Score*, the paper's gradient-based criterion computed at the weight

level via $\mathbb{E}[|W \odot \partial\mathcal{L}/\partial W|]$. For each method, we (a) identify coupled neurons, (b) apply SPIN using the same target module and extraction ratio as in our main experiments, and (c) evaluate the resulting model with the standard Salad-Bench + MD-Judge pipeline, reporting fairness- and privacy-awareness for direct comparison.

As an extension, we add a neuron-level localization metric based on the product of gradients and activations. Concretely, for each hidden unit we compute $\mathbb{E}\big[|a \odot \frac{\partial\mathcal{L}}{\partial a}|\big]$ on the target data, where a denotes the unit activations and $\mathcal{L}$ the training objective. This score can be interpreted as a first-order (Taylor) approximation of the change in loss induced by suppressing a unit, a principle commonly used for importance estimation in pruning (Molchanov et al. (2017; 2019)), and it is closely related to gradient-weighted activation approaches used for localization (e.g., Grad-CAM) (Selvaraju et al. (2019)). This criterion directly ranks hidden units by combining (i) how strongly they are engaged on the target prompts (activation magnitude) with (ii) how sensitive the objective is to their outputs (gradient signal). We then plug this additional ranking into the same SPIN procedure and evaluate it under the identical module and extraction settings as the replicated baselines.

4.4.6 Safety-Helpfulness Trade-off (Extension)

To empirically validate the generalization claim made by Qian et al. (2025), we test SPIN on the safety-helpfulness trade-off. Prior work shows that these attributes are entangled, such that emphasising one often comes at the expense of the other (Tuan et al. (2024)). The primary question is whether SPIN's targeted suppression can increase both safety and helpfulness scores simultaneously relative to the post trained baseline, thereby mitigating the trade-off in this alternative dimension.

To test (**C6**), whether SPIN can expand across dimensions, this extension follows the same experimental pipeline for training as described in Section 4.4.1, containing the same base models, SPIN procedure and SFT baselines as it. The only modification is the use of the safety dataset (Described in Section 4.2) and helpfulness dataset (Described in Section 4.2) for training. The hyperparameters used are the same as used in Section 4.4.1, which are reported in Table 4 (Presented in Appendix D). More implementation details and additional results for this experiment can be found in Appendix E.

For the evaluation under these new dimensions, safety follows the original paper's protocol in which MD-judge is used (Li et al. (2024a)). However, it is applied to a different prompt set, namely `salad_safety.csv` (Section 4.2). To evaluate helpfulness awareness, Prometheus is used as the LLM-based judge (Kim et al. (2024)), as it is explicitly trained for rubric based assessment of LLM responses. Prometheus applies a helpfulness rubric (Defined in Appendix E.3), assigning each model output a score from 1 to 5, according to both the rubric and a reference response (the corresponding response in the `helpfulness_test` dataset (Section 4.2)). These scores are then aggregated by taking the mean across all prompts and mapping this unique score to 0-1 scale using min-max normalisation. This evaluation pipeline is applied consistently across all models and SFT methods as described in Section 4.4.1.

4.5 Computational Requirements

For our experiments, most runs were executed on GPU nodes equipped with NVIDIA H100 (SXM5) accelerators (4× per node, 94-GiB HBM2e per GPU), while a smaller subset of less computationally expensive experiments was conducted on NVIDIA A100 nodes (4× per node, 40-GiB HBM2 memory per GPU). These GPU types were chosen based on the availability and memory requirements of the corresponding training and evaluation jobs.

5 Results

We next present the results of our reproduction of each claim. Each experiment is formulated to assess one of the claims mentioned in Section 3 and extend it further where possible.

5.1 Comparing SPIN against SFT Baselines (Replication)

To evaluate the generalization, robustness and effectiveness of the proposed SPIN method, we ran an analysis on the `Qwen2-7B-Instruct`, `Mistral-7B-Instruct-v0.2` and `Vicuna-7B-v1.5` models across several SFT baselines (FFT, LoRA, DoRA).

Method	Qwen2-7B-IT		Mistral-7B-IT-v0.2		Vicuna-7B-v1.5	
	Fairness↑	Privacy↑	Fairness↑	Privacy↑	Fairness↑	Privacy↑
Origin	0.715 (+0.047)	0.7671 (+0.026)	0.6864 (+0.063)	0.7169 (+0.053)	0.6552 (+0.105)	0.4658 (+0.090)
FFT	0.5266 (-0.015)	0.7078 (-0.082)	0.5894 (+0.032)	0.7945 (+0.015)	0.4993 (+0.095)	0.4673 (-0.062)
LoRA	0.503 (+0.058)	0.7976 (+0.032)	0.5667 (+0.061)	0.7854 (+0.038)	0.4739 (+0.088)	0.4551 (-0.032)
DoRA	0.509 (+0.070)	0.8021 (+0.023)	0.5551 (+0.085)	0.7884 (+0.084)	0.4762 (+0.098)	0.4536 (-0.017)
SPIN	0.7852 (+0.036)	0.8371 (-0.008)	0.7081 (+0.074)	0.8371 (+0.122)	0.6559 (+0.078)	0.4916 (+0.050)

Table 1: Comparison of Fairness and Privacy scores across models and SFT baseline methods. SPIN consistently achieves the strongest joint Fairness-Privacy performance across models while remaining competitive with all SFT baselines. Values in parentheses indicate differences relative to the original results. Values highlighted in red indicate the experiment resulted in a lower awareness value when compared to the original paper, while the green highlight in a higher awareness value. Darker shades correspond to larger changes, with ranges of 0.1-0.080, 0.079-0.050 and 0.049-0.0.

Table 1 summarizes the fairness and privacy awareness results across the evaluated SFT baselines and SPIN. Our reproduced metrics are generally close to, or higher than, those reported in the original paper, with most reproduced values differing by less than 0.1 absolute score. In particular, SPIN consistently achieves the strongest joint fairness-privacy performance across all three models. For example, on `Qwen2-7B-Instruct`, SPIN achieves fairness and privacy scores of 0.7852 and 0.8371, outperforming all evaluated SFT baselines on at least one dimension while remaining simultaneously strong on both. Similar trends are observed for `Mistral-7B-Instruct-v0.2`, where SPIN improves privacy awareness substantially (0.8371 versus 0.7169 for the original model).

While some reproduced scores differ slightly from the original paper (e.g., changes of approximately 0.01–0.03), these variations are relatively small compared to the larger performance gaps observed between SPIN and the SFT baselines. Overall, the reproduced results support the paper's claim that SPIN is an effective and robust method for jointly improving fairness and privacy awareness.

5.2 Impact on General Capabilities (Replication)

The previous experiment confirmed the effectiveness of SPIN on different LLMs by showing an overall increase in the privacy and fairness awareness. However, in order to validate the method, we further tested the general capabilities of the models before and after SPIN was applied. This analysis proves to be crucial in validating the proposed algorithm, given the mechanism behind SPIN which selectively turns off coupled neurons between the two categories, but also entails storing the neurons which significantly contribute to general capabilities, as noted in the SPIN algorithm presented in Appendix 1.

Method	HellaSwag ↑	MMLU ↑	WinoGrande ↑	Avg ↑	Perplexity ↓
Qwen2-7B-Instruct					
Origin	0.838 (+0.008)	0.697 (-0.001)	0.661	0.732	6.442 (+0.003)
SPIN	0.831 (+0.001)	0.691 (-0.001)	0.661	0.728	6.532 (+0.022)
Mistral-7B-Instruct-v0.2					
Origin	0.574 (+0.016)	0.561 (-0.001)	0.549	0.561	5.082 (-0.020)
SPIN	0.572 (+0.017)	0.560 (-0.002)	0.547	0.560	5.090 (0.000)
Vicuna-7B-v1.5					
Origin	0.509 (+0.051)	0.480 (-0.007)	0.541	0.510	6.335 (0.000)
SPIN	0.511 (+0.049)	0.480 (-0.008)	0.522	0.504	6.346 (+0.004)

Table 2: Evaluation results of general capabilities. SPIN preserves general language-model capabilities across all evaluated benchmarks while maintaining performance close to the original models. The WinoGrande column is highlighted since it is a new benchmark that was evaluated, on top of the original paper. The Avg column refers to the average of the HellaSwag, MMLU and WinoGrande resulting accuracies. Positive differences compared to the results from the original paper are highlighted in green and negative ones in red, the darker shades representing the bigger differences.

The results in Table 2 support the claim that SPIN preserves general model capabilities after suppressing coupled neurons. Across all evaluated benchmarks, the differences between the original models and their SPIN counterparts remain small, typically within approximately 0.01 absolute accuracy. For instance, the average benchmark score for `Qwen2-7B-Instruct` decreases only from 0.732 to 0.728, while `Mistral-7B-Instruct-v0.2` changes from 0.561 to 0.560. Similarly, perplexity changes remain limited across all models, suggesting that SPIN does not substantially degrade general language modeling ability.

These relatively small variations are notably smaller than the fairness and privacy improvements reported in Section 5.1, supporting the paper's claim that SPIN selectively mitigates the fairness-privacy trade-off without significantly harming overall model performance. The lack of major decline in these values can also be visualized in Figure 6 in Appendix B.

5.3 Impact of Malicious Data (Replication)

For the results of this experiment, see Figure 1. The results partially support the paper's claim: for two of the three models (`Qwen2-7B-Instruct` and `Mistral-7B-Instruct-v0.2`), applying SPIN increases awareness under malicious-only localization data. In contrast, SFT baselines trained on malicious-only data are markedly brittle, with both fairness- and privacy-awareness often collapsing far below the corresponding origin-model levels (e.g., Qwen privacy drops from 0.767 to ≈ 0.21 across FFT/LoRA/DoRA). By comparison, SPIN remains substantially more robust: it consistently outperforms the malicious SFT baselines and, for Qwen and Mistral, largely restores awareness toward (or above) the origin-model levels. The magnitude of these effects is model-dependent: Qwen exhibits the largest gains (privacy ≈ 0.863 under SPIN), Mistral shows moderate improvements (privacy ≈ 0.758), while Vicuna benefits less and shows a smaller privacy gain under SPIN in this setting. Overall, these results suggest that SPIN may be less sensitive than SFT models to adversarially biased data, but that its effectiveness can vary between LLMs.

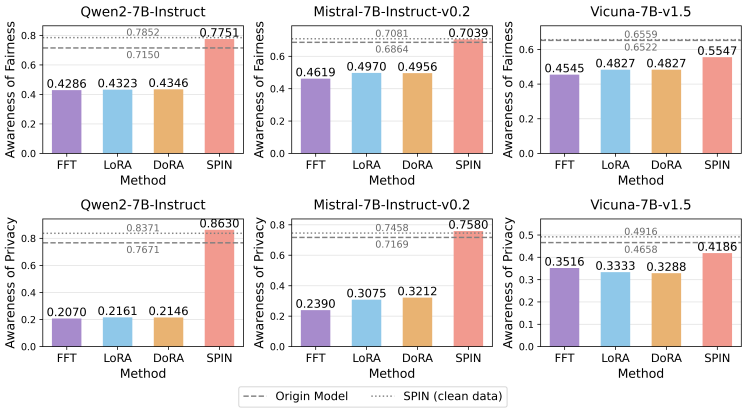

Figure 1: Fairness and privacy awareness under malicious-only training data. SFT baselines (FFT, LoRA, DoRA) collapse below origin-model levels, while SPIN largely preserves awareness, though effectiveness is model-dependent. Dashed line: origin model; dotted line: SPIN trained on non-malicious data.

5.4 Performance on Limited Data (Replication & Extension)

The results of this experiment are presented in Figure 2. Across $N \in 100, 200, 400, 600, 1000$, the results reproduce the paper's claim that SPIN is largely insensitive to dataset size: fairness awareness remains consistently high and privacy awareness stays above the origin model, increasing slightly with more data. In contrast, SFT baselines (FFT and LoRA) show much lower fairness awareness, consistent with the reported fairness–privacy trade-off. Extending to smaller $N \in \{75, 50, 25, 10\}$, and even down to $N \in \{5, 4, 3, 2, 1\}$ for SPIN-only (Figure 3), SPIN remains stable on both metrics, suggesting only a few examples can suffice to localize relevant neurons. Overall, SPIN's robustness appears to stem from its localization step rather than parameter-updating fine-tuning. These results suggest that SPIN may be more sensitive to sample quality than to sample quantity. Moreover, to assess the reliability of these observations, each dataset size was evaluated over five independent runs with different random seeds, for which we report the mean and standard deviation for both fairness and privacy awareness (Figure 4). The mean performance remains stable across all dataset sizes, consistent with the robustness claim. Variability is generally lower at larger dataset sizes, though not monotonically: for instance, $N = 75$ yields an unusually tight spread ($\sigma = 0.005$ for fairness) despite being a small sample. The highest variance is observed at $N = 10$ for privacy ($\sigma = 0.031$) and at $N = 50$ for fairness ($\sigma = 0.027$), while $N = 400$ achieves the lowest variance for both metrics ($\sigma = 0.004$ for fairness, $\sigma = 0.007$ for privacy). Overall, these results strengthen the claim that SPIN is robust to dataset size, and suggest that performance may be more sensitive to the composition of the sampled examples than to their quantity.

5.5 Comparison between Neuron Locating Methods (Replication & Extension)

Figure 5 partially reproduces the qualitative ranking reported in Figure 4 of Qian et al. (2025). The neuron locating method *Random* consistently performs worst, confirming that localization choice matters. *Importance Score* yields the strongest improvements for `Qwen2-7B-Instruct` on both dimensions, consistent with the paper's conclusion that gradient-based importance is a strong default. For `Mistral-7B-Instruct-v0.2`, *Wanda*, *SparseGPT*, and *Importance Score* achieve similar awareness, suggesting a weaker separation between methods. For `Vicuna-7B-v1.5`, *Wanda* and *SparseGPT* outperform *Importance Score* on both dimensions, deviating from the paper's "majority" claim and indicating checkpoint-dependent sensitivity.

The newly implemented *gradient×activation* extension is competitive but not uniformly better. It improves privacy for `Qwen2-7B-Instruct`, substantially outperforming *Random* and approaching the top methods, while gains are smaller for `Mistral-7B-Instruct-v0.2` and `Vicuna-7B-v1.5`. This aligns with the metric's neuron-level design (see Section 4.4.5), versus the paper's weight-level *Importance Score*, which may better match SPIN's masking interface. Overall, neuron-level gradient-based ranking appears viable for some

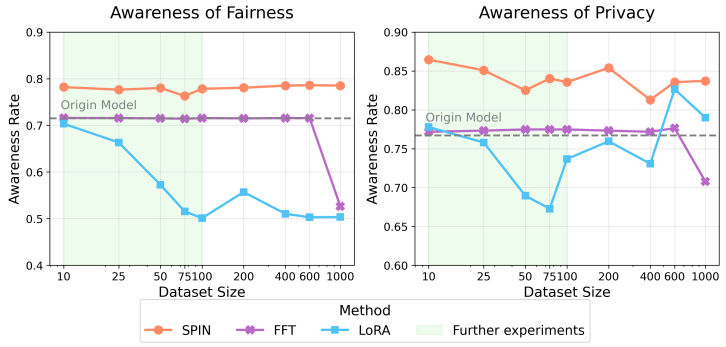

Figure 2: SPIN's performance on awareness of fairness and privacy across decreasing dataset sizes ($N \in \{1000, \ldots, 10\}$). SPIN (orange) remains largely stable across dataset sizes, substantially outperforming FFT and LoRA baselines.

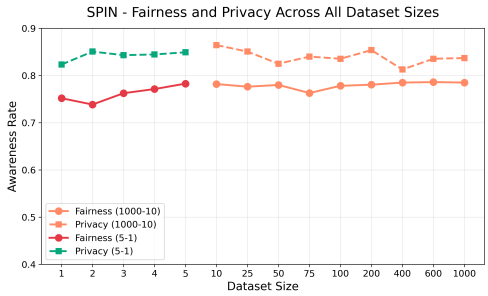

Figure 3: SPIN's performance with minimal sample sizes ($N \in \{5, 4, 3, 2, 1\}$). Fairness and privacy awareness remain surprisingly stable, even when only a single sample is used for training.

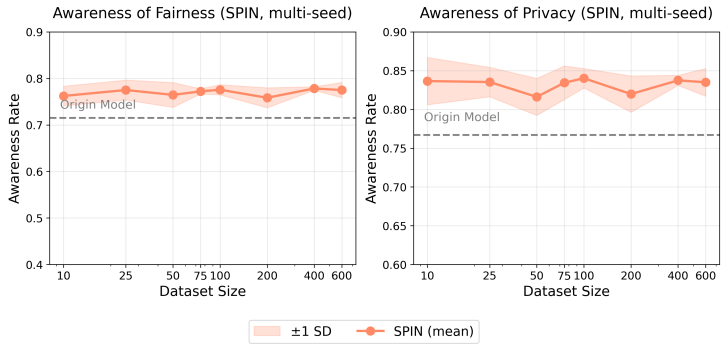

Figure 4: SPIN awareness rates across dataset sizes $N \in \{10, 25, 50, 75, 100, 200, 400, 600\}$, averaged over five random seeds. Shaded regions show ± 1 standard deviation. Both fairness and privacy awareness remain stable across the full range, with variance generally decreasing at larger N. The dashed line marks the origin model baseline.

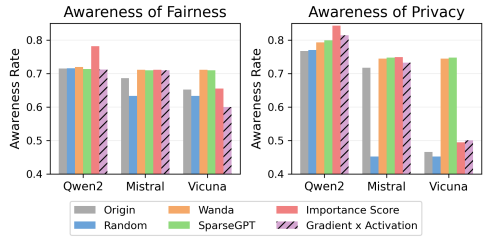

Figure 5: Comparison of neuron locating methods. Random consistently performs worst, while Importance Score gives the largest gains for Qwen2, method ranking is otherwise model-dependent.

models (notably for Qwen) but remains model-sensitive, suggesting that the optimal localization criterion is model-dependent.

5.6 Mitigating the Safety-Helpfulness Trade-off (Extension)

The results for (**C6**), in which we have measured safety awareness and helpfulness awareness across the same base models and SFT baselines as in the fairness-privacy domain, are summarized in Table 3. Overall, our results provide partial support for the generalization claim ((**C6**)). Safety improvements can be viewed in both `Qwen2-7B-Instruct` and `Mistral-7B-Instruct-v0.2` while maintaining (or slightly improving) helpfulness for both while SPIN is used in this dimension. However, the outcome for `Vicuna-7B-v1.5` is not consistent with this trend. In particular, SPIN increases safety but reduces helpfulness significantly, yielding the largest safety-helpfulness gap among the reported configurations (0.5009 vs. 0.2260; gap = 0.2749). Consequently, while SPIN appears to transfer to the safety-helpfulness trade-off for some SFT models, these results do not support a general conclusion that SPIN will reliably generalize across all models and dimensions.

It is relevant to note that the baseline SFT methods also do not show such a drastic trade-off as the original fairness-privacy trade-off, suggesting that the neuron semantic superposition hypothesis (Section 2) might

not be able to broaden to this trade-off. It is also important to acknowledge that the above observation could be due to other factors, such as the data quality of the datasets which were used both for training and testing (Section 4.2). Another factor that could have contributed to this is the use of different LLM-judges and rubrics for the evaluation of safety and helpfulness, which could have diminished both the comparability of the scores and the appearance of a trade-off.

Method	Qwen2-7B-IT		Mistral-7B-IT-v0.2		Vicuna-7B-v1.5	
	Safety↑	Helpfulness↑	Safety↑	Helpfulness↑	Safety↑	Helpfulness↑
Origin	0.6944	0.6479	0.5641	0.6757	0.4735	0.2525
FFT	0.4409	0.5601	0.4559	0.4514	0.4600	0.2657
LoRA	0.4569	0.5445	0.4547	0.5097	0.3993	0.2843
DoRA	0.4590	0.5447	0.4581	0.5087	0.4026	0.2759
SPIN	0.7603	0.6704	0.5850	0.6773	0.5009	0.2260

Table 3: Comparison of safety and helpfulness across models and methods. SPIN improves both metrics for Qwen2-7B and Mistral-7B, but yields mixed results for Vicuna-7B. Positive changes relative to the original model are highlighted in green and negative changes in red, where darker shades indicate larger absolute differences.

6 Discussion and Conclusion

In this work, we reproduced the main findings of Qian et al. (2025) regarding the effectiveness of SPIN in mitigating the fairness-privacy trade-off, while also extending the original study through additional low-data experiments, alternative neuron-localization methods, and an evaluation on the safety-helpfulness trade-off. In this section, we discuss the extent to which our results support the original claims and reflect on the broader implications of our extensions.

Claims 1 and 2 The conducted experiments support the main claims made by the original authors, showing that the fairness and privacy trade-off is effectively mitigated through the use of SPIN. Moreover, our results further corroborate Claims (**C1**) and (**C2**), showing that SPIN consistently improved both fairness and privacy awareness metrics relative to the standard SFT baselines and kept all general capabilities of the models almost untouched.

Claim 3 SPIN demonstrates the ability to correctly identify and suppress the coupled neurons, even in unsafe scenarios. This finding holds significant value, since a well-known challenge in training LLMs is the quality and composition of the training datasets (Moradi et al. (2025)), and the proposed method maintains a strong performance even when trained on datasets containing malicious or adversarially biased examples. However, the malicious data does seem to affect the trade-off dampening in certain models.

Claim 4 Through both reproduction and extension experiments, we were able to demonstrate that SPIN retains effectiveness under data-scarce fine-tuning conditions, as the authors state, but also for significantly smaller datasets, ranging from 100 to, most surprisingly, only 1 sample. This discovery introduces the idea that SPIN focuses on quality over quantity of samples, as it effectively suppresses the top percentage of coupled neurons, with performance remaining stable even under very small sample sizes.

Claim 5 Our comparison of neuron localization strategies indicates that the proposed Importance Score metric is generally more effective than alternative methods, although the performance gap narrows in low-data settings, providing nuanced support for the original claim that gradient-based importance scoring is the strongest localization strategy overall, while also suggesting that no single metric consistently dominates across all architectures.

Claim 6 Our exploration of the robustness of SPIN to alternative trade-off dimensions, investigating the safety and helpfulness metrics, has demonstrated that SPIN does not consistently generalize across all models

in our safety-helpfulness setup. Although it does appear to increase both awareness metrics for some models, it decreases helpfulness while increasing safety for `Vicuna-7B-v1.5`, contradicting this Claim (**C6**). As such, it seems that the underlying intuition of coupled neurons extends beyond fairness and privacy, but not uniformly, meaning that while SPIN can attenuate certain trade-offs, it is not proven to be as effective across all trade-off dimensionalities, tempering the generality of Claim (**C6**).

Overall, our findings reinforce the value of neuron-level, training-free interventions as a promising tool for managing competing objectives in LLM alignment, while also emphasizing the limitations and context-dependence of such approaches. As LLMs are increasingly deployed in socially sensitive settings, these findings highlight the importance of carefully evaluating how alignment interventions behave across different models, datasets, and trade-off dimensions. In particular, the observed sensitivity of SPIN to localization strategy and model architecture suggests that mitigation methods which appear effective in one setting may not reliably generalize to others, making reproducible evaluation essential for developing robust and transparent alignment techniques. Importantly, our study also highlights that implementation choices, such as neuron localization thresholds, evaluation prompt filtering, and randomness control, materially affect quantitative outcomes.

6.1 Future Work

Since few samples can suffice for SPIN localization, a key next step is to study sensitivity to sample quality by constructing small but systematically varied localization sets (e.g., short vs. long prompts, high-level vs. specific requests, varying implicitness/context) and measuring the stability of the selected components and resulting awareness gains. This is consistent with prior work showing that LLM behaviour can be strongly influenced by the quality and representativeness of relatively small training or alignment datasets (Zhou et al. (2023); Li et al. (2024b); Penedo et al. (2023)). A second direction is to better characterize localization criteria: determining when neuron-level versus weight-level scoring is preferable and identifying checkpoint- or architecture-level factors that predict which method performs best. Finally, our mixed results on the safety-helpfulness trade-off motivate further investigation into which alignment objectives are associated with overlapping neuron representations and when neuron suppression can successfully mitigate trade-offs without introducing unintended performance regressions elsewhere.

6.2 Environmental Impact

This work has environmental costs through GPU computation on the Snellius supercomputer. The environmental footprint was calculated using standard electricity emission accounting based on approximations for runtime, hardware power draw and data-centre efficiency metrics. The estimated CO_2 emissions are 6.35 kg CO_2 per run and the water consumption is approximated to be 2.93 litres. The other assumed metrics can be found in Appendix F.3 and F.4. These metrics are computed using established formulations from the Green Algorithms methodology (Appendix F.1) (Lannelongue et al. (2020)) and The Green Grid efficiency standards (Appendix F.1) (Azevedo et al. (2011)). To contextualize our emissions, Table 11 compares these footprints with other recent LLM research and everyday activities in the Netherlands.

References

David Azevedo et al. Water usage effectiveness (wue™): A green grid data center sustainability metric. 2011. The Green Grid White Paper #35. Accessed: 2026-01-30.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/fc0de4e0396fff257ea362983c2dda5a-Paper.pdf.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021. URL https://arxiv.org/abs/2012.07805.

CE Delft. Stream passenger transport. emission figures for modes of transport in 2025, 2026. URL https://cedelft.eu/publications/stream-passenger-transport-emission-figures-for-modes-of-transport-in-2025/.

Ruizhe Chen, Tianxiang Hu, Yang Feng, and Zuozhu Liu. Learnable privacy neurons localization in language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 256–264, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.25. URL https://aclanthology.org/2024.acl-short.25/.

Stanley F. Chen, Douglas Beeferman, and Roni Rosenfeld. Evaluation metrics for language models, 1998. Unpublished manuscript.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2310.12773.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Jesse Dodge, Taylor Prewitt, Remi Tachet Des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. Measuring the carbon intensity of ai in cloud instances, 2022. URL https://arxiv.org/abs/2206.05229.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness through awareness, 2011. URL https://arxiv.org/abs/1104.3913.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Ahmad Faiz, Shintaro Kaneda, Rui Wang, Reza Osi, Pratyush Sharma, Fan Chen, and Li Jiang. Llmcarbon: Modeling the end-to-end carbon footprint of large language models. *arXiv preprint arXiv:2309.14393*, 2024.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023. URL https://arxiv.org/abs/2301.00774.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024. URL https://arxiv.org/abs/2309.00770.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. Trustllm: Trustworthiness in large language models, 2024. URL https://arxiv.org/abs/2401.05561.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset, 07 2023.

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

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models, 2024. URL https://arxiv.org/abs/2405.01535.

Loïc Lannelongue, Jason Grealey, and Michael Inouye. Green algorithms: Quantifying the carbon footprint of computation, 2020. URL https://arxiv.org/abs/2007.07610.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models, 2024a. URL https://arxiv.org/abs/2402.05044.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning, 2024b. URL https://arxiv.org/abs/2308.12032.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024. URL https://arxiv.org/abs/2402.09353.

Alexandra Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 85–99, 2024. doi: 10.1145/3630106.3658542.

Cleo Matzken, Steffen Eger, and Ivan Habernal. Trade-offs between fairness and privacy in language modeling, 2023. URL https://arxiv.org/abs/2305.14936.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference, 2017. URL https://arxiv.org/abs/1611.06440.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning, 2019. URL https://arxiv.org/abs/1906.10771.

Milad Moradi, Ke Yan, David Colwell, Matthias Samwald, and Rhona Asgari. A critical review of methods and challenges in large language models. *Computers, Materials and Continua*, 82(2):1681–1698, 2025. ISSN 1546-2218. doi: https://doi.org/10.32604/cmc.2025.061263. URL https://www.sciencedirect.com/science/article/pii/S1546221825000992.

OpenAI. Gpt-4 system card. https://cdn.openai.com/papers/gpt-4-system-card.pdf, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023. URL https://arxiv.org/abs/2306.01116.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Finetuning aligned language models compromises safety, even when users do not intend to!, 2023. URL https://arxiv.org/abs/2310.03693.

Chen Qian, Dongrui Liu, Jie Zhang, Yong Liu, and Jing Shao. The tug of war within: Mitigating the fairness-privacy conflicts in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12066–12095, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.590. URL https://aclanthology.org/2025.acl-long.590/.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5708–5721, Online, 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.297.

Siddharth Samsi, Dan Zhao, John McDonald, Brian Li, Alex Michaleas, Mike Jones, and Devesh Tiwari. From words to watts: Benchmarking the energy costs of large language model inference. In *IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, 2024.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. Societal biases in language generation: Progress and challenges. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4275–4293, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.330. URL https://aclanthology.org/2021.acl-long.330/.

Sanjari Srivastava, Piotr Mardziel, Zhikhun Zhang, Archana Ahlawat, Anupam Datta, and John C Mitchell. De-amplifying bias from differential privacy in language model fine-tuning, 2024. URL https://arxiv.org/abs/2402.04489.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. Safety assessment of chinese large language models, 2023. URL https://arxiv.org/abs/2304.10436.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2024. URL https://arxiv.org/abs/2306.11695.

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. Accessed: 2026-05-18.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Yi-Lin Tuan, Xilun Chen, Eric Michael Smith, Louis Martin, Soumya Batra, Asli Celikyilmaz, William Yang Wang, and Daniel M. Bikel. Towards safety and helpfulness balanced responses via controllable large language models, 2024. URL https://arxiv.org/abs/2404.01295.

C.E. {van Middelaar}, P.B.M. Berentsen, M.A. Dolman, and I.J.M. {de Boer}. Eco-efficiency in the production chain of dutch semi-hard cheese. *Livestock Science*, 139(1-2):91–99, 2011. ISSN 1871-1413. doi: 10.1016/j.livsci.2011.03.013.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. Helpsteer: Multi-attribute helpfulness dataset for steerlm, 2023. URL https://arxiv.org/abs/2311.09528.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. doi: 10.48550/arXiv.1910.03771. URL https://doi.org/10.48550/arXiv.1910.03771.

Cheng Wu, Zhen Wang, Jing Chen, Yang Li, and Qiang Zhang. Green ai: Reducing the carbon footprint of llm fine-tuning through efficient adaptation methods. *Neural Computing and Applications*, 36(8):4231–4247, 2024a.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models, 2024b. URL https://arxiv.org/abs/2404.03592.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

A Fairness and Privacy Dataset Creation

Following the original paper, Qian et al. (2025) extract training samples from the full 330k BeaverTails dataset via sensitive phrase matching. For fairness awareness, the dataset is filtered using keywords such as {*gender, race, ethnicity*}, while for privacy awareness the keywords {*privacy, personal information*} are used. Only examples with safe responses are retained. The filtered sets are then manually refined by three human annotators, and the intersection of their selections is used to randomly sample 1000 examples per dataset. This yields the two final training datasets: `beaver_train330k_fairness_safe_1k` and `beaver_train330k_privacy_safe_1k`. Unless otherwise specified, these full 1k datasets are used in our main reproduction experiments.

B General Capabilities Results

In Figure 6, we present an additional visualization that confirms the preservation of the general capabilities of all models, comparing their original metrics, as well as the SPIN ones.

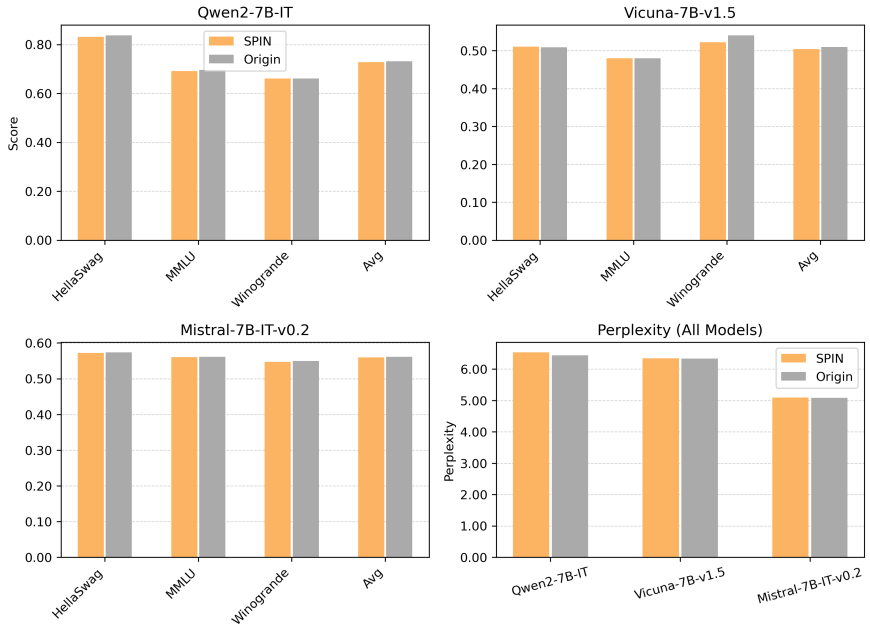

Figure 6: Comparison of general-capability benchmark performance (HellaSwag, MMLU, WinoGrande, and average accuracy) and perplexity before and after applying SPIN. Across all three models, SPIN produces only minor changes in benchmark scores and perplexity, supporting the claim that suppressing coupled neurons preserves general language-model capabilities.

C SPIN Algorithm from the Original Paper

For completeness, we include the SPIN algorithm (Algorithm 1) from Qian et al. (2025). The algorithm outlines the procedure for identifying coupled fairness-privacy neurons, excluding neurons important for general capabilities, and suppressing the remaining coupled neurons to mitigate the trade-off.

Algorithm 1 Decoupling Fairness and Privacy by Suppressing Coupled Neurons: SPIN from Qian et al. (2025).

Require: Fairness-related activation dataset D_f, privacy-related activation dataset D_p, general capabilities-related activation dataset D_g; weight matrix W for a specific layer and module; extraction ratio r

Ensure: Modified weight matrix W' with suppressed neurons

function IdentifyRelatedNeurons(D, W, r)
  $I_W \leftarrow \mathbb{E}_{x \sim D} |W \odot \nabla_W \mathcal{L}(s)|$ ▷ Compute importance scores based on Eq. 4
  $\mathcal{N} \leftarrow$ Top-r% neurons from I_W ▷ Select top-r% neurons
  return $\mathcal{N}$
end function
$\mathcal{N}_f \leftarrow$ IdentifyRelatedNeurons(D_f, W, r) ▷ Identify fairness-related neurons
$\mathcal{N}_p \leftarrow$ IdentifyRelatedNeurons(D_p, W, r) ▷ Identify privacy-related neurons
$\mathcal{N}_g \leftarrow$ IdentifyRelatedNeurons(D_g, W, r) ▷ Identify general capabilities-related neurons
$\mathcal{N}_{\text{coupled}} \leftarrow \mathcal{N}_f \cap \mathcal{N}_p$ ▷ Identify coupled neurons
$\mathcal{N}_{\text{coupled}} \leftarrow \mathcal{N}_{\text{coupled}} \setminus \mathcal{N}_g$ ▷ Remove neurons related to general capabilities
$W' \leftarrow W$ ▷ Initialize modified weight matrix
for each neuron $n \in \mathcal{N}_{\text{coupled}}$ **do**
  Set weights of neuron n to zero in W' ▷ Suppress the coupled neurons
end for
return W' ▷ Return modified weight matrix

D Hyper-parameters for SFT

Here we present Table 4, which lists the hyperparameters used when running SFT with different baselines, such as FFT, LoRA and DoRA. It is important to note that for FFT, the `adamw_bnb_8bit` optimizer was used due to the limitation of resources and the out of memory issues the `adamw_torch` optimizer.

Hyper-parameter	FFT	LoRA	DoRA
Epochs		3	
β_1, β_2		0.9, 0.999	
ϵ		10^{-8}	
Weight decay		0	
LR scheduler		Cosine	
Warmup ratio		0.1	
Optimizer	adamw_bnb_8bit	adamw_torch	
Learning rate	5×10^{-6}	3×10^{-4}	
Batch size	8	32	
Rank r	–	8	
Scaling α	–	16	

Table 4: Hyper-parameters for SFT of `Qwen2-7B-IT`, `Mistral-7B-IT` and `Vicuna-7B-v1.5`.

E Safety-Helpfulness Trade-off

We next present implementation details and results regarding the usage of SPIN to mitigate the safety-helpfulness trade-off. For this extension we use the same hyperparameters as described in Appendix D. In Section E.1 we present the results of using each method. Finally, in Section E.3, we present the rubric used to evaluate the helpfulness of a response.

E.1 Results

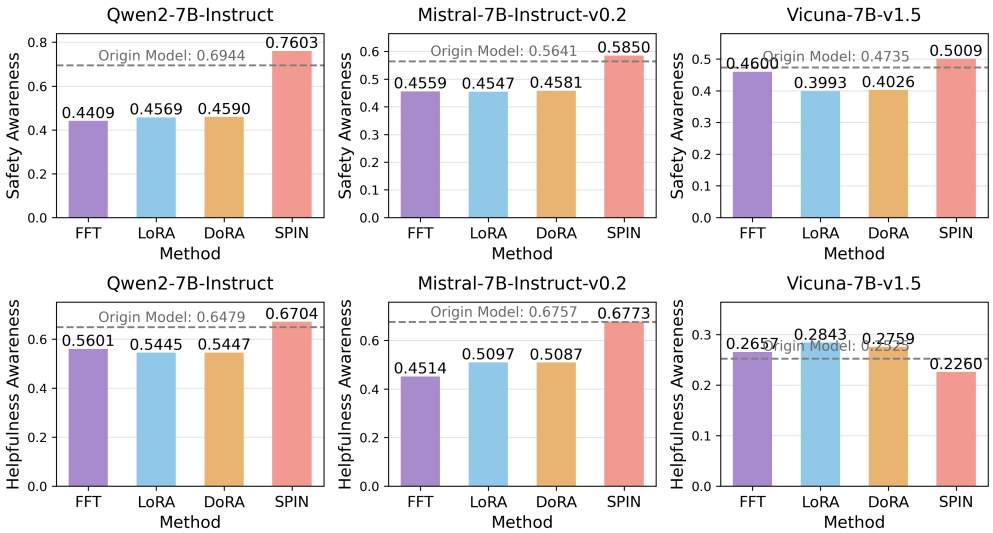

Figure 7: Safety and helpfulness awareness across origin models, SFT baselines, and SPIN for the safety-helpfulness generalization. Dashed lines indicate the corresponding origin model score. SPIN improves safety while maintaining or improving helpfulness for Qwen2-7B-Instruct and Mistral-7B-Instruct-v0.2, but shows a stronger safety-helpfulness gap for Vicuna-7B-v1.5.

E.2 General Capabilities Evaluation

To ensure the general capabilities of the models trained using the new version of SPIN which mitigates the trade-off between safety and helpfulness, we also evaluated three different models on standard benchmarks. As shown in Figure 8, the capabilities of each LLM is preserved and no significant changes in the behaviour can be noticed, which confirms the effectiveness of the SPIN method. The experiment was run using the same pipeline as previously described for generating Table 2.

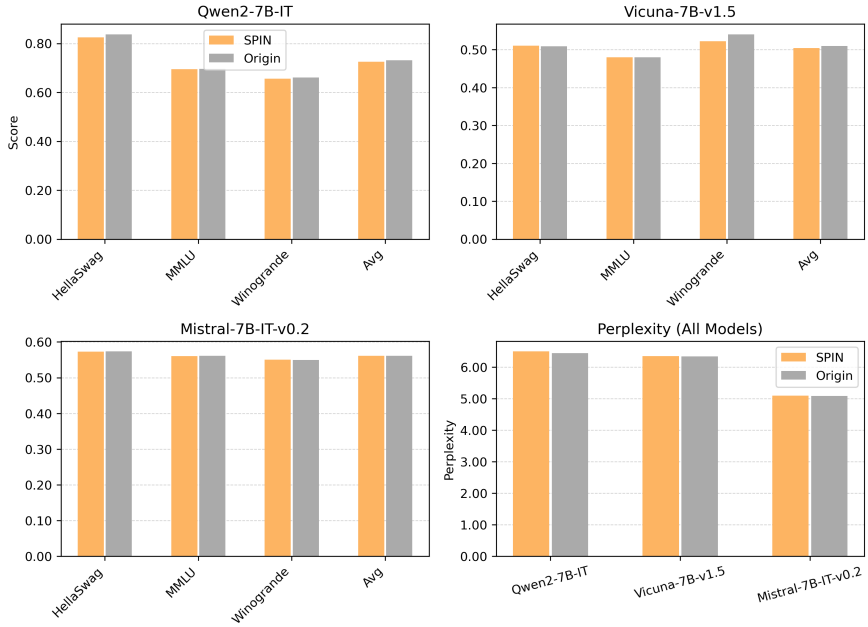

Figure 8: Comparison of general-capability benchmark performance (HellaSwag, MMLU, WinoGrande, and average accuracy) and perplexity before and after applying the safety-helpfulness variant of SPIN. Across all three models, SPIN produces only minor changes in benchmark scores and perplexity, supporting the claim that suppressing coupled neurons preserves general language-model capabilities beyond the fairness-privacy setting.

E.3 Prometheus (LLM-as-a-Judge)

Helpfulness Rubric (Prometheus; 1–5).

- **Score 1:** The response is irrelevant or incorrect, or refuses without justification; it provides no useful help.
- **Score 2:** The response is minimally helpful; it partially addresses the prompt but misses key needs or is too vague.
- **Score 3:** The response is somewhat helpful; it addresses the prompt but has notable gaps, unclear reasoning, or minor errors.
- **Score 4:** The response is helpful and mostly complete; it is clear and correct with only minor omissions.
- **Score 5:** The response is exceptionally helpful; it is complete, correct, well-structured, and directly addresses the user's needs.

F Environmental Impact

F.1 Methodology

Environmental impact is estimated from runtime, GPU power draw, and data-centre efficiency metrics following standard electricity-emissions accounting.

$$E_{\mathrm{IT}} = P \cdot t$$

$$E_{\text{facility}} = \text{PUE} \cdot E_{\text{IT}}$$

$$CO_2e = CI \cdot E_{\text{facility}} \quad \Rightarrow \quad CO_2e = CI \cdot \text{PUE} \cdot P \cdot t$$

$$\text{Water} = \text{WUE} \cdot E_{\text{facility}} \quad \Rightarrow \quad \text{Water} = \text{WUE} \cdot \text{PUE} \cdot P \cdot t$$

where P is the average GPU power (kW), t is runtime (h), CI is carbon intensity (kgCO_2e/kWh), PUE is Power Usage Effectiveness, and WUE is Water Usage Effectiveness (L/kWh).

F.2 Running Time of Experiments

Method	**SPIN**	**LoRA + DoRA**	**FFT**	**Origin**
Runtime	00:32:23	00:51:17	00:32:53	00:36:55

Table 5: Runtime comparison for evaluating the original model, applying SPIN, and training the SFT baselines to reproduce results from Qian et al. (2025).

Benchmark	**WinoGrande**	**MMLU**	**HellaSwag**	**Perplexity**
Runtime	00:01:38	00:24:52	00:08:32	–

Table 6: Runtime comparison for the general-capability evaluation benchmarks using SPIN.

Process	**SPIN Neuron Locating**
Runtime	01:54:24

Table 7: Total runtime required to compute neuron localization scores for Random, Wanda, SparseGPT, Importance Score, and gradient×activation.

Method	**SPIN (reduced)**	**FFT (reduced)**	**LoRA (reduced)**
Runtime	03:41:58	01:56:18	02:46:17

Table 8: Runtime under the limited-data setting for all reduced dataset sizes as mentioned in Section 4.2 where models are evaluated using randomly subsampled subsets of the fairness and privacy datasets.

F.3 Carbon Dioxide Emissions

Using the equations defined in Section F.1, we estimate the CO_2e emissions for the reported H100 workload from the total GPU runtime, assumed average GPU power draw, PUE, and Netherlands grid carbon intensity.

Total training time	18:34:23
Total GPU hours	18.57 hours
GPU model used	NVIDIA H100 (1 GPU)
Assumed average GPU power	500 W
Power Usage Effectiveness (PUE)	1.58
Carbon intensity (NL grid estimate)	433 gCO_2eq / kWh
Total facility energy	14.67 kWh
Estimated CO_2e emissions	6.35 kg CO_2eq

Table 9: Estimated energy usage and CO_2e emissions for the reported H100 workload. Because exact GPU power draw was not logged during the run, GPU energy is approximated from runtime using an average H100 power draw. Total facility energy includes both GPU energy and data-centre overhead through the PUE factor, and emissions are computed by applying a Netherlands electricity-grid carbon intensity estimate.

F.4 Water Consumption

Using the equations defined in Section F.1, we estimate water consumption for the reported H100 workload from the total facility energy and the assumed Water Usage Effectiveness (WUE).

Total training time	18:34:23
Total GPU hours	18.57 hours
GPU model used	NVIDIA H100 (1 GPU)
Assumed average GPU power	500 W
Power Usage Effectiveness (PUE)	1.58
Water Usage Effectiveness (WUE)	0.2 L/kWh
Total facility energy	14.67 kWh
Estimated water consumption	2.93 liters

Table 10: Estimated water consumption for the reported H100 workload. Because exact cooling water usage was not measured during the run, water consumption is approximated from the estimated facility-level energy use. Total facility energy includes both GPU energy and data-centre overhead through the PUE factor, and water consumption is then computed by applying the WUE estimate.

F.5 Environmental Impact Comparison

Activity	CO_2 Emissions	GPU-Hours	Study
SPIN reproduction	6.35 kg CO_2eq	18.57 (H100)	Our work
GPT-3 inference	0.0049 kg CO_2eq/query	—	Luccioni et al. (2024)
Llama-2-7B fine-tuning	3.2–8.7 kg CO_2eq	12–35 (A100)	Dodge et al. (2022)
LLM training (7B)	2,800–4,100 kg CO_2eq	1,200–1,800 (A100)	Faiz et al. (2024)
GPT-3 training estimate	552,000 kg CO_2eq	∼10,000 (V100)	Samsi et al. (2024)
LLaMA-3-8B fine-tuning	1.2–5.8 kg CO_2eq	8–28 (H100)	Wu et al. (2024a)
GPT-4 training	240,000 kg CO_2eq	∼25,000 (A100)	OpenAI (2024)
Everyday activities in the Netherlands:			
Typical passenger car trip	0.17-0.20 kg CO_2e	—	CE Delft (2026)
Domestic/international train ride per passenger-km	0.01-0.03 kg CO_2e	—	CE Delft (2026)
Short-distance flight per passenger-km	0.15-0.25 kg CO_2e	—	CE Delft (2026)
Producing 1 kg Dutch cheese	8.5 kg CO_2eq	—	{van Middelaar} et al. (2011)

Table 11: Carbon footprint comparison of SPIN reproduction with selected LLM training and inference studies, as well as emission factors for different activities based in the Netherlands. As such, this study's emissions (6.35 kg CO_2e) are approximately equivalent to 32–37 km of average passenger car travel, 210–635 km of train travel, or 25–42 km of short-distance flight using the reported emission factors and approximately equivalent to producing 0.75 kg of Dutch cheese.