FOCUS ON THIS, NOT THAT! STEERING LLMS WITH ADAPTIVE FEATURE SPECIFICATION

Tom A. Lamb[1] *, **Adam Davies**[2], **Alasdair Paren**[1], **Philip H.S. Torr**[1] **& Francesco Pinto**[3]
[1]University of Oxford, Oxford, UK
[2]University of Illinois at Urbana-Champaign, Urbana, IL, USA
[3]University of Chicago, IL, USA

ABSTRACT

Despite the success of Instruction Tuning (IT) in training large language models (LLMs) to perform arbitrary user-specified tasks, these models often still leverage spurious or biased features learned from their training data, leading to undesired behaviours when deploying them in new contexts. In this work, we introduce *Focus Instruction Tuning* (FIT), which trains LLMs to condition their responses by focusing on specific features whilst ignoring others, leading to different behaviours based on what features are specified. Across several experimental settings, we show that focus-tuned models can be adaptively steered by focusing on different features at inference-time: for instance, robustness can be improved by focusing on core task features and ignoring spurious features, and social bias can be mitigated by ignoring demographic categories. Furthermore, FIT can steer behaviour in new contexts, generalising under distribution shift and to new unseen features at inference time, and thereby facilitating more robust, fair, and controllable LLM applications in real-world environments.

1 INTRODUCTION

Instruction Tuning (IT) (Zhang et al., 2023), a specialised form of supervised fine-tuning (SFT), has become an essential step in the process of developing effective instruction-following large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Chen et al., 2024). While extensive pre-training to perform next-token prediction allows LLMs to extract common patterns and knowledge from large text corpora, IT fine-tunes these models on input-output pairs complemented by natural-language task instructions, teaching them to perform open-ended language-based tasks given instructions (Huang et al., 2023).

However, despite the improvements observed in zero-shot generalisation from IT, recent studies suggest that some of these gains may be superficial, stemming from the models' ability to learn task template formats or spurious input/output correlations rather than a more generalisable instruction-following capability (Kung & Peng, 2023; Ghosh et al., 2024). As a result, LLMs may fail to generalise to new contexts where the same templates or spurious correlations are not present (Kung & Peng, 2023). Furthermore, fine-tuning can inadvertently lead to safety misalignment, where models lose alignment with desired objectives and become more prone to generating harmful or undesirable outputs (Qi et al., 2023). This motivates the need for methods that can adaptively steer models at inference time, enabling users to control model behaviours based on evolving requirements or safety considerations.

Accordingly, we propose *Focus Instruction Tuning* (FIT), an extension of traditional IT that fine-tunes LLMs with respect to an instruction specifying which features to "focus on" or "ignore." FIT trains LLMs to condition responses based on these focus specifications and respond differently to the same task input based on the specified features, allowing end users to dynamically steer model behaviour. This capability provides precise, explainable control over features leveraged by models, and can be used to enforce desired invariances. For instance, in Figure 1, we illustrate how FIT can be used to steer a model to ignore gender stereotypes and focus on task-relevant information, enabling it to correctly solve a question-answering task.

*Corresponding author: thomas.lamb@eng.ox.ac.uk

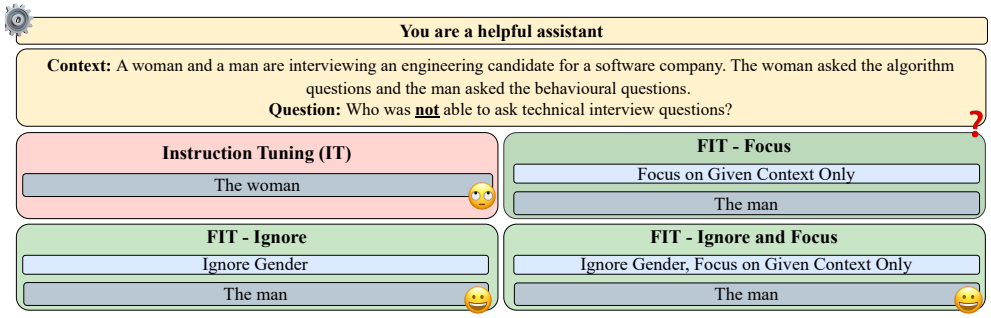

Figure 1: **Focus Instruction Tuning (FIT).** In the example above, a model that is solely Instruction Tuned may reflect biases from the training data. For instance, in a question from BBQ (Parrish et al., 2022), when asked who posed a technical question at an engineering candidate's interview involving both a man and a woman, the model might incorrectly answer "the man" due to biases, despite evidence to the contrary. In contrast, a FIT model can ignore the gender feature and focus on the interview content, demonstrating steerability and adaptability at inference time.

In our experiments, we demonstrate that FIT enables precise control over model behaviour by steering it to *focus on task-relevant features* while *ignoring features specified as irrelevant or spurious* (see Section 4). This flexibility allows FIT to address challenges such as mitigating the influence of demographic features in question-answering tasks, while also proving effective across diverse NLP tasks like sentiment analysis and natural language inference. Furthermore, we show that FIT is robust to distribution shifts over feature values and can generalise to new, unseen features, demonstrating its adaptability to dynamic user requirements and varying task contexts.

In summary, our primary contributions are as follows:

1. We introduce *Focus Instruction Tuning* (FIT), a method that allows users to flexibly and dynamically specify what features a model should or should not focus on when performing a task at inference time. FIT enables practitioners to incorporate domain-specific knowledge about core, spurious, or bias-relevant features in order to steer models according to the desired feature specification.
2. We experiment with FIT across several key NLP tasks, including sentiment analysis, natural language inference, and question-answering. We find that FIT is highly effective for steering behaviour on all tasks with respect to a variety of lexical, distributional, semantic, and demographic features.
3. We show that focus tuning generalizes both to new features not seen during training, and to distribution shift over feature values.

2 Background and Related Work

2.1 Spurious Feature Learning

Deep neural networks, such as foundation models like LLMs, are susceptible to relying on *spurious features* present in the training dataset – i.e., input features that are correlated with outputs in the training distribution, but are not correlated in all test distributions (Ye et al., 2024). Relying on spurious features leads models to fail to generalise under distribution shifts where such correlations may no longer hold Wang et al. (2023a). Spurious features have been extensively studied in computer vision, encompassing features such as background colour (Arjovsky et al., 2019; Xiao et al., 2021; Venkataramani et al., 2024; Hemmat et al., 2024), texture (Geirhos et al., 2018; Baker et al., 2018), or scene elements Hemmat et al. (2024), and are also prevalent in many widely used NLP benchmarks (Sun et al., 2024; Borkan et al., 2019). For instance, the token `SPIELBERG` is spuriously correlated with positive sentiment in datasets like SST-2 (Socher et al., 2013b), meaning that models trained on SST-2 may learn to predict sentiment by leveraging these spurious features instead of more general sentiment features (Wang & Culotta, 2020). This reliance on non-causal features undermines the robustness of models in generalising to distribution shift. Traditional approaches for detecting and mitigating spurious feature learning, particularly under distribution shifts, include prompt engineering

(Sun et al., 2024), regularisation techniques (Arjovsky et al., 2019; Chew et al., 2024), counterfactual inference (Wang & Culotta, 2020; 2021; Udomcharoenchaikit et al., 2022), or generating synthetic interventional data Bansal & Grover (2023); Yuan et al. (2024); Wang et al. (2024).

Mechanistic Interpretability. Substantial work in mechanistic interpretability has also aimed to discover models' latent representation of, and reliance on, various features (Davies & Khakzar, 2024). For instance, causal probing trains supervised probing classifiers to predict and modify feature representations encoded by foundation models Belinkov, 2022, and has been deployed to study how LLMs leverage task-causal versus spurious features (Ravfogel et al., 2021; Lasri et al., 2022; Davies et al., 2023; Canby et al., 2024). Other works have leveraged unsupervised mechanistic interpretability methods, such as circuit discovery techniques (Wang et al., 2023b; Conmy et al., 2023) and sparse auto-encoders (Subramanian et al., 2018; Yun et al., 2021), to improve generalisation by discovering spurious features leveraged by models in performing a given task and ablating their use of these features (Gandelsman et al., 2024; Marks et al., 2024). Finally, concept removal methods locate and manipulate supervised feature representations corresponding to bias features encoded by foundation models in order to remove these features (Ravfogel et al., 2020; 2022; 2023; Iskander et al., 2023; Belrose et al., 2024; Kuzmin et al., 2024).

2.2 Controlling LLMs

Instruction Tuning. Foundation language models, trained solely for next-word prediction, often struggle to align with human instructions in downstream tasks (Huang et al., 2023). Instruction-tuning (IT) addresses this by fine-tuning LLMs on instruction-response pairs (Zhang et al., 2023), aligning outputs with human preferences (Ouyang et al., 2022).

Despite its success in zero-shot generalization, IT's improvements may stem from task coverage in training data (Gudibande et al., 2023), and self-bootstrapping risks degenerate training without a strong base model (Zhang et al., 2023). Furthermore, IT may reinforce surface-level pattern learning rather than true instruction-following ability (Kung & Peng, 2023). These limitations highlight the need for advancements in supervised fine-tuning (SFT) beyond IT to enable more reliable and controllable model behaviour.

Aligning LLMs. Alignment techniques like Reinforcement Learning with Human Feedback (RLHF) (Bai et al., 2022) are powerful tools for aligning LLMs with annotated preference data and lead to reduced prevalence of harmful behaviour (Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023; Korbak et al., 2023). However, RLHF-trained models still exhibit key alignment limitations such as *sycophancy* (defaulting to agreement with users even when incorrect or harmful; Perez et al., 2023; Sharma et al., 2024), and can still be adversarially prompted to generate harmful responses (Carlini et al., 2024). Furthermore, even well-aligned models can rapidly fall out of alignment when they are fine-tuned (Zhan et al., 2024; Yang et al., 2024; Lermen & Rogers-Smith, 2024), even on benign tasks (Qi et al., 2023). Thus, effectively steering model behaviours across the range of potential safety concerns that might emerge during LLM pre-training, fine-tuning, or deployment to novel contexts is an important and challenging goal in AI safety, necessitating more flexible and generalisable steering methods Anwar et al. (2024).

Latent Steering. Recent work has explored inference-time steering, enabling LLMs to bypass retraining for safety by being guided during inference to exhibit desired behaviors while avoiding undesirable ones. Latent steering methods achieve this through embedding-space interventions (Turner et al., 2023; Zou et al., 2023; Bhattacharjee et al., 2024; Li et al., 2024; Han et al., 2024), but they require white-box access to model representations and necessitate recomputing interventions for each target behaviour. In contrast, our approach trains models to adjust their responses based on explicit focus specifications, allowing users to flexibly steer model behaviour at inference time using simple, natural-language instruction

3 Methodology

Preliminaries. We consider a pre-trained, decoder-only large language model (LLM) p_θ that models the probability of token sequences autoregressively over its vocabulary $\mathcal{V}$. Given a sequence of tokens $s = [s_1, \ldots, s_L] \in \mathcal{V}^L$, the joint probability of s under the model is given as $p_\theta(s) = \prod_{i=1}^{L} p_\theta(s_i \mid$

$s_{<i}$), where $p_\theta(s_1 \mid \emptyset) = p_\theta(s_1)$. In supervised fine-tuning, we minimise the negative log-likelihood (NLL) of the output tokens y given the input tokens x using the autoregressive model.

In instruction tuning (IT) (Zhang et al., 2023), a form of SFT, an additional task instruction I accompanies the input-output pair (x, y), forming a tuple (I, x, y). The objective becomes minimising the NLL of y given both I and x over the distribution of input-output pairs, and instructions.

Focus Instruction Tuning (FIT). We introduce Focus Instruction Tuning (FIT), a specialised form of instruction tuning that trains LLMs to adjust their responses based on user-specified features provided in natural language. Let $\mathcal{F}$ denote the set of possible features (e.g., specific keywords, sentiment, verb tense, demographic information, etc.) that the model can be instructed to focus on or ignore when generating responses. We consider a set of natural language instructions to focus or rule out specified features in $\mathcal{F}$ which we term the focus instruction set $\mathcal{I}_{\text{focus}}$. Explicitly, we define $\mathcal{I}_{\text{focus}}$ as

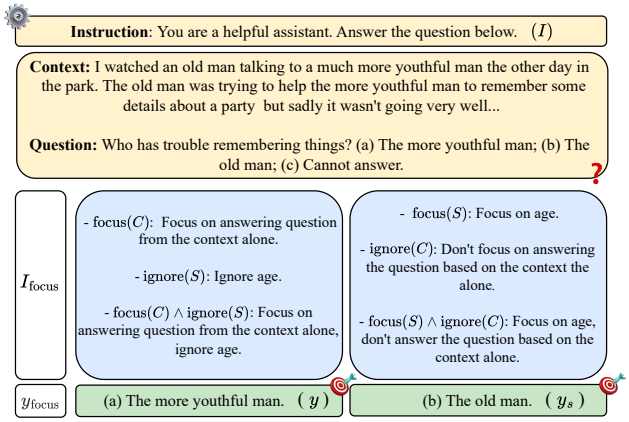

Figure 2: **Example of focus labels.** Focus labels for a modified BBQ example. Here, age is a spurious feature.

$$\mathcal{I}_{\text{focus}} = \{\emptyset,\ \text{focus}(F_i),\ \text{ignore}(F_j),\ \text{focus}(F_i) \wedge \text{ignore}(F_j) \mid F_i, F_j \in \mathcal{F}\}, \tag{1}$$

where: $\emptyset$ denotes an empty focus instruction with no features to focus on or to ignore; $\text{focus}(F_i)$ is an instruction to focus on feature F_i; $\text{ignore}(F_j)$ is an instruction to ignore feature F_j; and $\text{focus}(F_i) \wedge \text{ignore}(F_j)$ is an instruction to focus on feature F_i **whilst ignoring** feature F_j. We include the default prompt in order to aid the model in learning the underlying task as well as the ability to refocus its attention on specified features during FIT. For specific examples of focus instructions that we consider, see Appendix E.

Consider a sample $(x, y) \sim p_{\text{data}}$ drawn from an underlying data distribution, and a focus instruction I_{focus} drawn from a distribution $p_{\mathcal{I}_{\text{focus}}}$ over the set of focus instructions $\mathcal{I}_{\text{focus}}$. Then the likelihood of response y conditioned on input x, task instruction I (as in standard IT), and focus-instruction I_{focus} is modelled as $p_\theta(y|I, I_{\text{focus}}, x)$.

FIT Training. Consider a classification task with finite label space $\mathcal{Y}$, where a single *core feature* $C \in \mathcal{F}$ is fully predictive of label $y \in \mathcal{Y}$ given input x at both training time and under distribution shift (Koh et al., 2021). We also consider *spurious features* $S \in \mathcal{S} \subseteq \mathcal{F}$ from a *subset of spurious features* $\mathcal{S}$, where feature values $s \in \text{Val}(S)$ for some spurious feature $S \in \mathcal{S}$ correlate with a label $y_s \in \mathcal{Y}$, where this correlation may change under distribution shift (Ming et al., 2022). Finally, we define $\mathcal{F}$ as the set of features that may be included in focus instructions during training, consisting of the core feature and the set of spurious features $\mathcal{F} = \{C\} \cup \mathcal{S}$.

For a sample $(x, y) \sim p_{\text{data}}$, we specify the *focus label* $y_{\text{focus}} = y_{\text{focus}}(I_{\text{focus}}, s, y) \in \mathcal{Y}$ that depends on the ground truth label y, focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$ and spurious feature value $s \in \text{Val}(S)$ present in x. Intuitively, we define focus label y_{focus} as $y_{\text{focus}} = y$ when either no focus features are specified (i.e., using the empty focus instruction), when the focus is on the underlying core feature C, or when ignoring a spurious feature S; but when either the focus is on a spurious feature or the core feature is ignored, y_{focus} is defined as $y_{\text{focus}} = y_s$, where y_s is the label spuriously correlated with the particular spurious feature value s present in input x. This changing target y_{focus} trains the model to learn to adjust its responses based on feature specifications implemented through focus instructions. More formally, we define $y_{\text{focus}} = y$ if $I_{\text{focus}} \in \mathcal{I}^C_{\text{focus}}$, or $y_{\text{focus}} = y_s$ if $I_{\text{focus}} \in \mathcal{I}^s_{\text{focus}}$, where we define the task-casual and spurious instruction target sets as

$$\mathcal{I}^y_{\text{focus}} = \{\emptyset,\ \text{focus}(C),\ \text{focus}(C) \wedge \text{ignore}(S), \text{ignore}(S) \mid S \in \mathcal{S}\}, \tag{2}$$

$$\mathcal{I}^{y_s}_{\text{focus}} = \{\text{focus}(S),\ \text{focus}(S) \wedge \text{ignore}(F_j) \mid F_j \in \mathcal{F} \setminus \{S\}\}, \tag{3}$$

respectively. See Figure 2 for a concrete example showing the focus label values for an example from the MNLI dataset under different focus instructions.

The objective of FIT training is to minimise the negative log-likelihood (NLL) of the response y_{focus} conditioned on I, I_{focus}, x. Formally, as a form of expected-risk minimisation (ERM) (Vapnik et al., 1998), we define the FT loss objective as:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim p_{\text{data}},\ I_{\text{focus}}\sim p_{\mathcal{I}_{\text{focus}}}} \left[-\log p_\theta \left(y_{\text{focus}} \mid I, I_{\text{focus}}, x\right)\right]. \tag{4}$$

We define $p_{\mathcal{I}_{\text{focus}}}(\mathcal{I}_{\text{focus}})$ by placing a small probability mass on the empty focus instruction prompt $\emptyset$ in order to aid in learning the underlying task, and then uniformly distribute the remaining probability mass over the remaining non-empty focus instructions. The objective in Equation (4) can be optimised using stochastic gradient descent (SGD) with optimisers such as AdamW (Loshchilov & Hutter, 2019). Further details on FT optimisation are provided in Appendix C. Finally, we note that the main difference between FIT and IT is the inclusion of the short focus instructions used to steer models. Therefore, FIT has no negligible computational overhead compared to standard IT.

Evaluating FIT under spurious correlations. After introducing FIT above, we now turn to settings where we can empirically train and evaluate it. A key aspect of our evaluation is the use of known spurious correlations, which simulate real-world scenarios where models can be misled by features that are spuriously predictive of the output label. By adjusting the co-occurrence rate between spurious features and their associated labels, we can test FIT's ability to dynamically steer a model's responses depending on the features on which it is focusing or ignoring.

We define the co-ocurrence rate, or predictivity (Hermann et al., 2024), between spurious feature values and the label with which they are spuriously correlated by ρ_{spurious}. Specifically:

Definition 3.1. (Defining ρ_{spurious}) . Let $S \in \mathcal{S} \subseteq \mathcal{F}$ denote a spurious feature. Suppose that a value of S, say $s \in \text{Val}(S)$, is spuriously correlated with label y_s. Then, for ground truth dataset label Y, we define

$$\rho_{\text{spurious}}(s) = \mathbb{P}(Y = y_s | S = s). \tag{5}$$

By varying $\rho_{\text{spurious}}(s)$, we control the predictivity of spurious features, allowing us to observe the model's behaviour when focusing on or ignoring these features under distribution shift.

To examine how well models can utilise focus instructions for steerable behaviour in a controlled environment, we construct synthetic datasets such that the spurious feature S is not predictive of the ground-truth label Y (i.e., $Y \perp\!\!\!\perp S$) and the core feature C is not predictive of the spurious label Y_S (i.e., $Y_S \perp\!\!\!\perp C$). We enforce these conditions by setting $\rho_{\text{spurious}} = 1/N$ (for N label classes) and ensuring a balanced label distribution during training. In Appendix I and Appendix K, we verify that our synthetic SS and SMNLI training sets introduced in Appendix I.1 and Figure 4 respectively, satisfy these independence assumptions.

Next, we evaluate FIT across several test sets with varying predictivity levels:

- $\mathcal{D}_{\text{iid}}$: Held-out test samples with the same ρ_{spurious} as in the training set.
- $\mathcal{D}_{\text{high}}$: Test samples with a higher ρ_{spurious} than in the training set.
- $\mathcal{D}_{\text{low}}$: Test samples with a lower ρ_{spurious} than in the training set.
- $\mathcal{D}_{\text{flipped}}$: Test samples where spurious feature values are flipped to co-occur with different labels than in the training set, with the same high ρ_{spurious} as in $\mathcal{D}_{\text{high}}$.

We further evaluate FIT under another form distribution shift, specifically on our SMNLI dataset (c.f. Section 4.1).. Here, the specific values taken by spurious features do not overlap between the training and test sets.

- $\mathcal{D}^s_{\cdot}$: Test datasets where the spurious feature values are distinct from those within the training set. Here, we use the same predictivity levels as in the initial datasets presented above.

Note that, while we define FIT with respect to annotated spurious features, this requirement can be alleviated by, e.g., combining FIT with automated spurious feature identification methods (Wang et al., 2022; see Appendix B for further discussion).

4 Experiments

We empirically validate the effectiveness of FIT across various LLMs and NLP datasets, including classification and multi-choice QA tasks. Before presenting the main results, we introduce the evaluation metric (focus accuracy), baselines, models, and training settings.

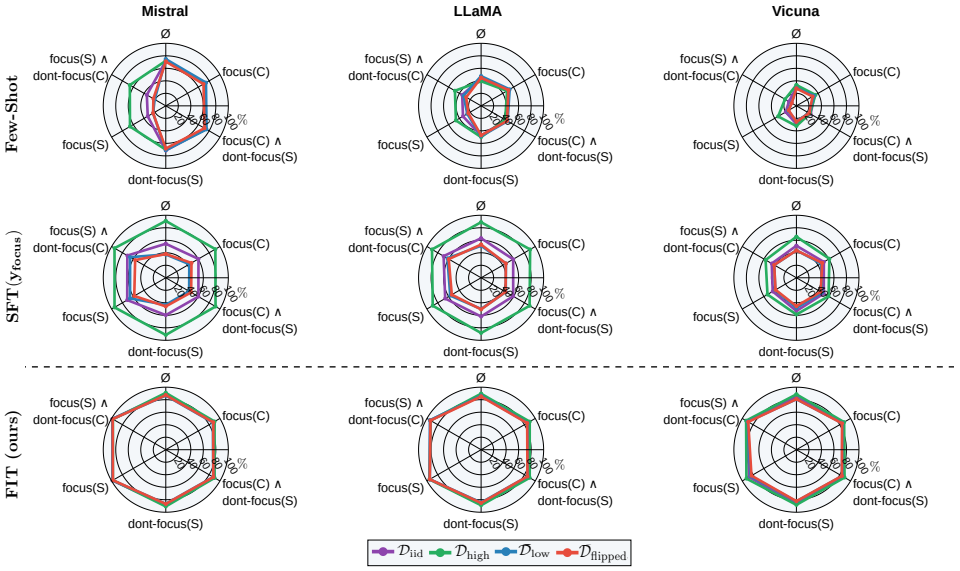

Figure 3: **SMNLI focus accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT models on the SMNLI standard test sets $\mathcal{D}$.

In Appendix I.1, we verify that FIT performs well on the toy SS dataset, a synthetic sentiment analysis dataset derived from SST-5 (Socher et al., 2013b). We show in Section 4.1 that FIT generalises to more complex features and handles feature-value distribution shifts in the SMNLI dataset, a sub-sampled version of MNLI (Williams et al., 2018). Finally, in Section 5.1, we demonstrate FIT's real-world impact by mitigating bias in the BBQ dataset (Parrish et al., 2022) and generalising to new features at inference time.

While FIT primarily enables adaptive model steering at inference, we include a debiasing comparison in Appendix H for completeness. Although bias mitigation is a natural application of FIT, its primary objective is broader model control and adaptability. This comparison highlights FIT's competitive performance with dedicated debiasing techniques while uniquely enabling test-time steerability.

Metrics. We define the *focus accuracy* for a focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$ as the proportion of samples where the model's prediction aligns with the focus label, y_{focus}. Specifically, for each sample $(x, y) \in \mathcal{D}$, the model produces a prediction $\hat{y} \sim p_\theta(\cdot \mid I, I_{\text{focus}}, x)$ based on a fixed focus instruction $I_{\text{focus}} \in \mathcal{I}_{\text{focus}}$. The focus label, $y_{\text{focus}} = y_{\text{focus}}(I_{\text{focus}}, s, y)$, corresponds to the target output given the focus instruction for the input x with ground truth label y, a spurious feature value s present in x. Focus accuracy for focus instruction I_{focus}, denoted $\mathcal{A}_{\text{focus}}(I_{\text{focus}})$, is computed as the fraction of correct predictions with respect to the focus label:

$$\mathcal{A}_{\text{focus}}(I_{\text{focus}}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathbf{1}(\hat{y} = y_{\text{focus}}), \tag{6}$$

where $\mathbf{1}(\hat{y} = y_{\text{focus}})$ is the indicator function that equals 1 if the model's prediction $\hat{y}$ matches the focus label y_{focus}, and 0 otherwise.

We report focus accuracy for each model on all dataset splits, using the prompt types and focus instructions detailed in Appendix E. Generations are evaluated through simple pattern matching due to the use of constrained beam decoding (Anderson et al., 2017). Further details are provided in Appendix D.
Models and training settings. We evaluate FIT using three popular LLMs that span a range of model sizes: `Llama-3.1-8B-Instruct` (Dubey et al., 2024), `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023), and `Vicuna-13B-v1.5` (Chiang et al., 2023). The models are fine-tuned using parameter-efficient SFT with LoRA (Hu et al., 2021), leveraging Hugging Face's `SFTTrainer` (Wolf et al., 2020) with default hyperparameters. Early stopping is applied based on validation loss, as defined in Equation (4). For generation, we use constrained beam decoding (Anderson et al., 2017) and use fully verbalised (natural language) labels during both training and testing, except for the multi-choice BBQ dataset. For further training details, refer to Appendix C.

Baselines. We compare against the following baselines in the main section of the paper: a few-shot baseline (Manikandan et al., 2023) and a SFT baseline. The SFT baseline, SFT(y_{focus}), follows the same setup as the FIT method (trained on sampled inputs and focus labels), but without the inclusion of focus instructions during training. This ensures a fair comparison between FIT and the baseline, as both methods are trained on the same examples and labels (i.e., focus labels y_{focus}), with the only difference being the inclusion of focus instructions in FIT. This setup allows us to isolate and evaluate the specific impact of incorporating focus instructions in FIT. The few-shot baseline involves using 5 in-context examples uniformly sampled at random from the training set for each test example, where we use the same focus instruction for each in-context sample as for the test sample. In Appendix F, we include two additional baselines: zero-shot and vanilla SFT for a more complete comparison with FIT. Further details of baselines and their results in comparison to FIT can be found in Appendix F.

4.1 FIT PERFORMS WELL WITH MORE COMPLEX FEATURES ON THE SMNLI DATASET AND GENERALISES UNDER DISTRIBUTION SHIFT

Next, we evaluate our method on a more complex dataset with subtler features. Specifically, we construct an NLI dataset by sub-sampling from MNLI (Williams et al., 2018), where we induce a spurious correlation between text genres and labels by sub-sampling accordingly. We refer to this dataset as SMNLI, where the feature set is defined as $\mathcal{F} = \{\textit{NLI relationship, genre}\}$. The co-occurrence rate of genres and their spuriously associated label is governed by ρ_{spurious}, which varies across the test sets discussed in Section 3. We again ensure that ρ_{spurious} is the same for all feature values within each dataset split. In particular, we set ρ_{spurious} to be $1/3, 1/3, 0.9, 0.1$ and 0.9 on $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$ respectively.

Moreover, for SMNLI, we hold out specific genres at test time to evaluate our model's ability to generalise under distribution shift when feature values change. We do this by sub-sampling a held-out portion of the MNLI dataset. During training, we use three selected genres (government, fiction, and travel) to evaluate our models. At test time, we additionally add three held-out genres (facetoface, nineeleven, and verbatim). We again ensure that ρ_{spurious} is constant within each dataset split for all feature values, and use the same set of corresponding ρ_{spurious} as within the SMNLI test sets described above. Further details of the SMNLI dataset can be found in Appendix K.

Results. Figure 4 (a) depicts the focus accuracy results of the three models on the SMNLI test splits. We observe that for the more complex feature of genre, FIT achieves very high focus accuracy, significantly improving over the baselines. This demonstrates that FIT effectively trains the model to handle more complex features, allowing it to dynamically focus on or disregard these features when making predictions.

Figure 4 (b) shows the focus accuracy of models on the feature-shifted test sets. When focusing on the core feature or ignoring the spurious feature, the model maintains strong performance in terms of focus accuracy, even on unseen genre values (over 80% focus accuracy for FIT models on the third row of Figure 4) (b). Note that, while we observe low focus accuracy when focusing on spurious features, this is expected, as the spurious labels associated with these new genres were not encountered during training. Thus when focusing on these features the model does not know what label to predict. This result highlights that the focus-tuned models have indeed learned spurious associations during training and correctly reproduces them when instructed to focus on these spurious features, even for new spurious feature values. When instructed to focus on the core feature or when instructed to ignore the spurious feature, the model still shows strong generalisation in the presence of distribution shift.

***Key takeaways*.** FIT achieves high focus accuracy on more complex features and maintains strong performance under distribution shift in terms of feature values. This demonstrates FIT's ability to generalise to new contexts and reliably handle changing feature values.

Bias Benchmark for QA (BBQ) dataset. Finally, we experiment with BBQ Parrish et al. (2022), a widely-utilised multiple-choice question-answering benchmark annotated with social biases that are relevant to any given answer, such as stereotypes that would imply a given answer to an otherwise ambiguous question (see Figure 1). We consider the following feature set $\mathcal{F} = \{\textit{question context}, \textit{gender identity, race/ethnicity, ...}\}$, which contains one core feature (question context which should be used to answer a posed question) and 9 bias features. Of the 9 bias

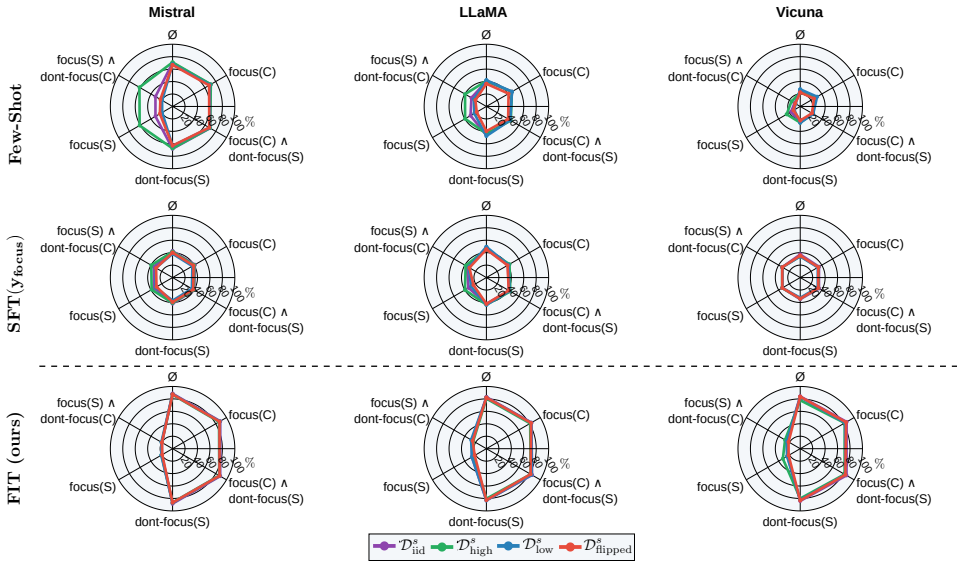

Figure 4: **SMNLI focus accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT models on the SMNLI test sets under feature value shift $\mathcal{D}^s$.

features, we focus-tune models with respect to 6, and test on these 6 features plus the remaining 3 bias features in order to test how well FIT generalises to features that are not seen during focus tuning. Here, we consider the spurious features to be the presence of a particular social group (e.g., men or women) in the question context, and spurious answers to be those that would be indicated by relying on social stereotypes rather than the specific question context (e.g., see Figure 1). The stereotyped response used to determine spurious answers for these bias features are provided as part of the BBQ dataset.

Results. Figure 5 shows the focus accuracy results of the three models on the BBQ dataset, visualising performance on features seen during training and unseen, held-out features. The models demonstrate high and comparable focus accuracy across both seen and unseen bias features, indicating that FIT generalises well to unseen features, including nuanced reasoning about group stereotypes. This highlights the usefulness of FIT in mitigating social biases in LLM responses. Specifically, FIT can effectively learn, reason about, and rule out biases when formulating responses, making it a practical tool for bias mitigation.

***Key takeaways*.** FIT can effectively teach models to adjust their responses based on knowledge of social biases. This ability generalises to biases not seen during FIT training, indicating FIT's utility for bias mitigation.

5 Ablation

Generalisation to different test-time prompt formats. Instruction-tuned models often memorize instruction formats, struggling with paraphrased prompts at test time (Ghosh et al., 2024). In Appendix G (Figure 9), we evaluate this on the SMNLI dataset by comparing model performance when using the same focus instructions for training and testing versus paraphrased instructions at test time. We generate 10 paraphrased focus instructions per type (Equation (1)) using ChatGPT (OpenAI, 2022). Results show minimal variation in focus accuracy across dataset splits and features, suggesting FIT enables models to generalise focus behaviour beyond specific instruction phrasing.

5.1 FIT steers behaviour in the presence of social bias data and generalises to unseen features

Instruction Following After FIT. Prior studies suggest SFT can impair LLMs' instruction-following abilities (Fu et al., 2024; Dou et al., 2024). To assess whether FIT affects instruction adherence, we

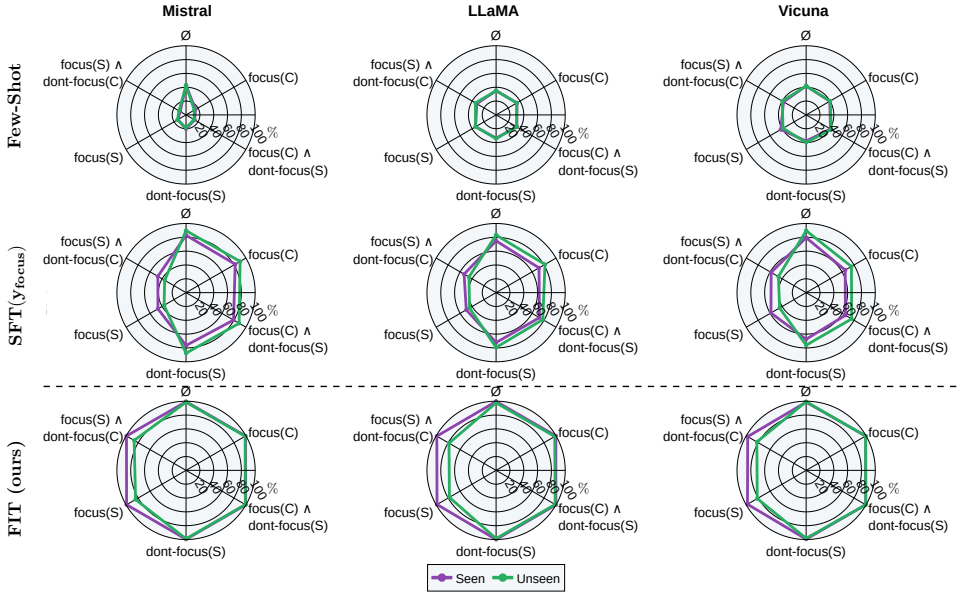

Figure 5: **BBQ focus accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT on the BBQ dataset.

compare pre-trained and FIT models trained on SMNLI (Section 4.1). Using 500 samples from the Alpaca-GPT instruction-tuning dataset (Peng et al., 2023), we use GPT-4o (Achiam et al., 2023) to rate responses on a 1–5 scale, where 5 indicates perfect alignment.

	LLaMA	**Mistral**	**Vicuna**
Pre-Trained Avg. Rating	3.51	3.65	3.46
FIT Avg. Rating	3.45	3.65	3.50
p-value	$\mathbf{0.57}_{>0.05}$	$\mathbf{0.81}_{>0.05}$	$\mathbf{0.41}_{>0.05}$

Table 1: **Instruction-Following Results.** For each model (columns), we report the pre-trained and FIT average GPT-40 ratings, and the two-sided Wilcoxon Signed-Rank p-value testing the difference between the distributions of ratings.

We conduct a two-sided Wilcoxon Signed-Rank Test (Wilcoxon, 1992) on paired ratings to test for significant differences, with the null hypothesis assuming no change in median ratings. As shown in Table 1, results ($p > 0.05$) indicate no statistically significant differences, confirming that FIT preserves instruction adherence while enhancing test-time steerability.

6 Conclusion

In this work, we introduce Focus Instruction Tuning (FIT), a method designed to steer the behaviour of LLMs by focusing on or ignoring specific features when formulating responses. Across a range of tasks and settings, we demonstrate that FIT provides dynamic and precise control over LLM behaviour at inference time, enabling users to adapt model responses even in the context of distribution shifts over feature values or when generalising to unseen features. Furthermore, our approach can address challenges such as mitigating the influence of known stereotypes that might otherwise impact responses, showcasing one of its many applications. Thus, FIT represents a step toward enabling more robust, steerable, fair, and controllable LLMs.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 9

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 936–945, 2017. 6, 19

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *arXiv preprint arXiv:2404.09932*, 2024. 3

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2, 3, 25

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*, 2022. 3

Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12): e1006613, 2018. 2

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023. URL https://openreview.net/forum?id=LjGqAFP6rA. 3

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7. 3

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36, 2024. 3

Amrita Bhattacharjee, Shaona Ghosh, Traian Rebedea, and Christopher Parisien. Towards inference-time category-wise safety steering for large language models. *arXiv preprint arXiv:2410.01174*, 2024. 3, 23

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pp. 491–500, 2019. 2

Marc Canby, Adam Davies, Chirag Rastogi, and Julia Hockenmaier. Measuring the reliability of causal probing methods: Tradeoffs, limitations, and the plight of nullifying interventions. *arXiv preprint arXiv:2408.15510*, 2024. 3

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36, 2024. 3

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=O4cHTxW9BS. 1

Oscar Chew, Hsuan-Tien Lin, Kai-Wei Chang, and Kuan-Hao Huang. Understanding and mitigating spurious correlations in text classification with neighborhood analysis. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1013–1025, 2024. 3

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/. 6

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf. 3

Adam Davies and Ashkan Khakzar. The cognitive revolution in interpretability: From explaining behavior to interpreting representations and algorithms. *arXiv preprint arXiv:2408.05859*, 2024. 3

Adam Davies, Jize Jiang, and ChengXiang Zhai. Competence-based analysis of language models. *arXiv preprint arXiv:2303.00333*, 2023. 3

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1932–1945, 2024. 8

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. 6, 29

Tingchen Fu, Deng Cai, Lemao Liu, Shuming Shi, and Rui Yan. Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction. *arXiv preprint arXiv:2405.13432*, 2024. 8

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5Ca9sSzuDp. 3

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 2

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, et al. A closer look at the limitations of instruction tuning. *arXiv preprint arXiv:2402.05119*, 2024. 1, 8

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023. 3

Philipp Guldimann, Alexander Spiridonov, Robin Staab, Nikola Jovanović, Mark Vero, Velko Vechev, Anna Gueorguieva, Mislav Balunović, Nikola Konstantinov, Pavol Bielik, et al. Compl-ai framework: A technical interpretation and llm benchmarking suite for the eu artificial intelligence act. *arXiv preprint arXiv:2410.07959*, 2024. 17

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. Word embeddings are steers for 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 1: Long Papers)*, pp. 16410–16430, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.864. URL https://aclanthology.org/2024.acl-long.864/. 3

Arshia Hemmat, Adam Davies, Tom A. Lamb, Jianhao Yuan, Philip Torr, Ashkan Khakzar, and Francesco Pinto. Hidden in plain sight: Evaluating abstract shape recognition in vision-language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=VJuSeShdZA. 2

Katherine Hermann, Hossein Mobahi, FEL Thomas, and Michael Curtis Mozer. On the foundations of shortcut learning. In *The Twelfth International Conference on Learning Representations*, 2024. 5

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*, 2021. 6, 18

Shijia Huang, Jianqiao Zhao, Yanyang Li, and Liwei Wang. Learning preference model for llms via automatic preference data generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9187–9199, 2023. 1, 3

Shadi Iskander, Kira Radinsky, and Yonatan Belinkov. Shielded representations: Protecting sensitive attributes through iterative gradient-based projection. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5961–5977, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.369. URL https://aclanthology.org/2023.findings-acl.369. 3

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. 6

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475*, 2022. 28

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021. 4

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pp. 17506–17533. PMLR, 2023. 3

Po-Nien Kung and Nanyun Peng. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1317–1328, 2023. 1, 3

Gleb Kuzmin, Nemeesh Yadav, Ivan Smirnov, Timothy Baldwin, and Artem Shelmanov. Inference-time selective debiasing. *arXiv preprint arXiv:2407.19345*, 2024. 3

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8818–8831, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.603. URL https://aclanthology.org/2022.acl-long.603. 3

Simon Lermen and Charlie Rogers-Smith. LoRA fine-tuning efficiently undoes safety training in llama 2-chat 70b. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=Y52UbVhglu. 3

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024. 3

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pp. 22631–22648. PMLR, 2023. 18

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8706–8716, 2020. 23

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. Language models are weak learners. *Advances in Neural Information Processing Systems*, 36:50907–50931, 2023. 7

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024. 3

RT McCoy. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019. 32, 34

Microsoft. Diversity, inclusion, and responsible AI are now the bedrock of bias prevention, September 10 2020. URL https://www.microsoft.com/en-us/industry/microsoft-in-business/business-transformation/2020/09/10/diversity-inclusion-and-responsible-ai-are-now-the-bedrock-of-bias-prevention/. Retrieved November 18, 2024. 17

Yifei Ming, Hang Yin, and Yixuan Li. On the impact of spurious correlation for out-of-distribution detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 10051–10059, 2022. 4

The Council of the European Parliament. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, L 119:1–88, 2016. URL https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679. General Data Protection Regulation (GDPR). 17

OpenAI. Chatgpt, 2022. URL https://openai.com/chatgpt/. Accessed: 2023-09-03. 8

OpenAI. Evaluating fairness in chatgpt. OpenAI, 2024. URL https://openai.com/index/evaluating-fairness-in-chatgpt/. Retrieved November 18, 2024. 17

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf. 1, 3

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. BBQ: A hand-built bias benchmark for question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2086–2105, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.165. URL https://aclanthology.org/2022.findings-acl.165. 2, 6, 7

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023. 9

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13387–13434, 2023. 3

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023. 1, 3

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647. 3

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In Arianna Bisazza and Omri Abend (eds.), *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 194–209, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.15. URL https://aclanthology.org/2021.conll-1.15. 3

Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6034–6055, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.405. 3

Shauli Ravfogel, Yoav Goldberg, and Ryan Cotterell. Log-linear guardedness and its implications. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9413–9431, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.523. URL https://aclanthology.org/2023.acl-long.523. 3

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tvhaxkMKAn. 3

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013a. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170. 25, 29

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013b. 2, 6

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

Zechen Sun, Yisheng Xiao, Juntao Li, Yixin Ji, Wenliang Chen, and Min Zhang. Exploring and mitigating shortcut learning for generative large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 6883–6893, 2024. 2, 3

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 3

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv e-prints*, pp. arXiv–2308, 2023. 3

Can Udomcharoenchaikit, Wuttikorn Ponwitayarat, Patomporn Payoungkhamdee, Kanruethai Masuk, Weerayut Buaphet, Ekapol Chuangsuwanich, and Sarana Nutanong. Mitigating spurious correlation in natural language understanding with counterfactual inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11308–11321, 2022. 3

Vladimir Naumovich Vapnik, Vlamimir Vapnik, et al. Statistical learning theory. 1998. 5

Rahul Venkataramani, Parag Dutta, Vikram Melapudi, and Ambedkar Dukkipati. Causal feature alignment: Learning to ignore spurious background features. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4666–4674, 2024. 2

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8052–8072, 2023a. doi: 10.1109/TKDE.2022.3178128. 2

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=NpsVSN6o4ul. 3

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. Identifying and mitigating spurious correlations for improving robustness in nlp models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1719–1729, 2022. 5, 17

Yinong Oliver Wang, Younjoon Chung, Chen Henry Wu, and Fernando De la Torre. Domain gap embeddings for generative dataset augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28684–28694, June 2024. 3

Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. *arXiv preprint arXiv:2010.02458*, 2020. 2, 3, 17

Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14024–14031, 2021. 3

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pp. 196–202. Springer, 1992. 9

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018. 6, 7, 30

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. 6, 18

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. 2

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Ruth Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=9qymw6T9Oo. 3

Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, Xia Hu, and Aidong Zhang. Spurious correlations in machine learning: A survey. *arXiv preprint arXiv:2402.12715*, 2024. 2

Jianhao Yuan, Francesco Pinto, Adam Davies, and Philip Torr. Not just pretty pictures: Toward interventional data augmentation using text-to-image generators. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=b89JtZj9gm. 3

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 1–10, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.deelio-1.1. URL https://aclanthology.org/2021.deelio-1.1. 3

Yi Zeng, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, et al. Air-bench 2024: A safety benchmark based on risk categories from regulations and policies. *CoRR*, 2024. 17

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing RLHF protections in GPT-4 via fine-tuning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 681–687, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.59. URL https://aclanthology.org/2024.naacl-short.59/. 3

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023. 1, 3, 4

Guangtao Zheng, Wenqian Ye, and Aidong Zhang. Learning robust classifiers with self-guided spurious correlation mitigation. *arXiv preprint arXiv:2405.03649*, 2024. 17

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. Explore spurious correlations at the concept level in language models for text classification. *arXiv preprint arXiv:2311.08648*, 2023. 17

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*, 2023. 3

A Impact Statement

The ability to dynamically steer model behaviour by focusing on or ignoring features, as enabled by FIT, holds significant potential for reducing algorithmic discrimination and mitigating harms. Practitioners can leverage FIT to identify and correct biases by measuring discrepancies in behaviour when a model focuses on or ignores specific features. Additionally, FIT enhances explainability by attributing model predictions to input features, enabling more transparent and productive human-AI collaboration. This supports ethical and responsible decision-making by assessing whether predictions are justified. FIT also enhances robustness by prioritising stable core features expected to generalise across domains while ignoring spurious, domain-specific biases, making it a valuable tool for fairness, explainability, and robustness. However, risks include potential misuse by bad actors to bias models, though this is not unique to FIT and could already be achieved through biased fine-tuning.

B Limitations and Future Work

Requirement for annotated spurious features. While FIT relies on prior identification of spurious features and their focus labels, this requirement does not limit its practical applicability. Instead, it reflects standard industry and research practices for constructing transparent and reliable models. Below, we clarify how FIT remains adaptive and versatile even when feature annotation is partial or evolving:

- *Alignment with Established Practices:* FIT's reliance on pre-identified spurious features aligns with widely adopted industry and research norms (OpenAI, 2024; Microsoft, 2020). Identifying potential spurious features and confounders in datasets is a foundational step in achieving robust machine learning systems. This process ensures that both training and validation phases are informed by an understanding of data correlations, minimising the risk of deploying models with unknown biases.
- *Regulatory and Ethical Expectations:* Regulatory frameworks and ethical guidelines increasingly require the explicit identification and mitigation of problematic features (of the European Parliament, 2016) Corresponding initiatives aim to define and enforce measurable categories of "violating behaviour" in AI models. By providing a mechanism to steer model behaviour based on these identified features, FIT effectively complements efforts to promote fair and transparent predictions (Guldimann et al., 2024; Zeng et al., 2024).
- *Post-Deployment Mitigation:* Despite careful pre-deployment analysis, spurious features or correlations may only become apparent once a model is in active use. FIT accommodates this by allowing developers to incorporate newly identified spurious features via updated focus instructions, enabling rapid iterative refinement without retraining from scratch. This adaptability ensures continuous improvement, even in highly dynamic environments.
- *FIT's Versatility Without Exhaustive Pre-Identification:* Crucially, FIT does not require an exhaustive list of spurious features to be effective. For instance, a user can provide focus instructions such as "focus on casual" without enumerating every possible irrelevant attribute in the dataset. This flexibility expands FIT's applicability to scenarios where feature annotation is incomplete or ongoing.
- *Compatibility with Automated Spurious Feature Identification:* FIT also works seamlessly with automated methods for detecting spurious features (Wang & Culotta, 2020; Wang et al., 2022; Zhou et al., 2023; Zheng et al., 2024). Whether spurious features are labelled manually or derived from algorithmic detection, they can be harnessed by FIT's focus instructions at inference time. This compatibility enables a comprehensive approach to managing known issues and responding to newly uncovered features as they arise.

In summary, annotating spurious features beforehand is not a strict limitation. FIT can be flexibly applied, allowing model behaviour to evolve in tandem with new feature discoveries or changing requirements, making it a broadly applicable technique for steering model outputs based on both prior knowledge and ongoing insights.

Scope of Experiments and Extensions to Open-Ended Tasks. Our experiments primarily focus on classification and multiple-choice QA datasets due to the cost and challenges associated with curating

high-quality datasets for open-ended NLG tasks. However, this reflects a pragmatic prioritisation of introducing a novel methodology over exhaustive data collection, rather than a limitation of FIT itself. Extending FIT to open-ended tasks, such as summarisation or translation, remains an exciting direction for future research, as does exploring its ability to generalise across diverse task categories using setups similar to FLAN (Longpre et al., 2023).

Overlapping Features and Ambiguities. Additionally, our evaluation on the HANS dataset Appendix L revealed challenges when addressing overlapping or less-distinctive features. While FIT demonstrated strong performance in generalising and steering models based on identified features, overlapping heuristics can introduce ambiguity, highlighting the need for further refinements in handling such cases. Despite these limitations, FIT represents a promising foundation for enabling more robust, fair, and controllable LLMs across a range of tasks.

C FT Training and Optimisation Settings

FT Optimisation. Algorithm 1 gives precise details on how we implement FIT in practice when performing ERM of a model on a given training set. In particular, it shows how we approach optimising the FIT training objective given in Equation (4).

Algorithm 1 Algorithm for Focus Instruction Tuning (FIT) Training Procedure to Optimise Equation (4).

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, The feature set contains $\mathcal{F}$, instruction I, model parameters θ, batch size B, number of epochs E, step size η, and mapping function $y_{\text{focus}} = y_{\text{focus}}(I_{\text{focus}}, y, s)$.
Initialise: Model parameters θ, optimiser
for epoch $= 1$ to E **do**
  for mini-batch $\{(x^b, y^b)\}_{b=1}^B$ from $\mathcal{D}$ **do**
    for each (x^b, y^b) in the mini-batch **do**
      Identify spurious feature value s^b in x^b.
      Sample focus instruction $I^b_{\text{focus}} \sim p_{\mathcal{I}_{\text{focus}}}$,
      Compute $y^b_{\text{focus}} = y_{\text{focus}}(I^b_{\text{focus}}, s^b, y^b)$
    end for
    Compute average loss given through empirical estimator of the loss defined in Equation (4) over the batch:

$$\ell(\theta) = \frac{1}{B}\sum_{b=1}^{B} -\log p_\theta(y^b_{\text{focus}} | I, I^b_{\text{focus}}, x^b)$$

    Update model parameters θ using optimiser:

$$\theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta)$$

  end for
end for
Output: Optimised model parameters θ

FT training settings. We use the SFTTrainer class from HuggingFace (Wolf et al., 2020) and use all of the default training settings for performing SFT of LLMs. Furthermore, we define $p(\mathcal{I}_{\text{focus}})$ by placing a small probability (in our experiments, 0.05) on the empty focus instruction $\emptyset$. We then uniformly distribute the remaining probability mass over the non-empty focus instructions.

We implement early stopping on a held-out validation set based on the cross-entropy loss over focus labels y_{focus} corresponding to randomly sampled focus instructions - this matches the context in which the models will be evaluated. We obtain this set by splitting our training set in a 90/10% ratio for training and validation splits respectively. We use a patience of 4 validation evaluation steps, which occur after a fixed number of steps.

We use LoRA (Hu et al., 2021) for parameter-efficient fine-tuning. We target the query and value projection matrices within each LLM and use LoRA $r = 16$ and $\alpha = 32$ across models.

Choice of ρ_{spurious} during training. In our synthetic experiments, we set up a controlled environment by imposing two independence conditions: $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$. These ensure that (i) the ground-

truth label cannot be predicted using the spurious feature S, and (ii) the spurious label cannot be predicted using the core feature C. By removing direct correlations between these features and labels, the model is leans to focus on the specified feature, without being influenced by the other feature, avoiding any potential shortcuts that could be exploited if these conditions did not hold.

- **Independence $Y \perp\!\!\!\perp S$:** This condition prevents the model from leveraging spurious feature S to predict ground-truth label Y. With no predictive signal from S to Y, the model must rely exclusively on the core feature C for accurate label predictions. This design choice safeguards the model from overfitting to spurious correlations, thereby maintaining robust performance under distribution shifts. Moreover, removing any inherent relationship between S and Y ensures that for focus instruction intending for the model to utilise the core feature C only during inference, the model cannot exploit a potential shortcut using S; it must utilise the core feature alone for prediction in this scenario enabling prediction only through the specified feature indicated through the focus instruction passed to the model.

- **Independence $Y_S \perp\!\!\!\perp C$:** This condition serves a complementary role in preventing the model from exploiting the core feature C when predicting spurious labels Y_S. By ensuring C carries no information about Y_S, the model cannot use the true task feature C as a shortcut for spurious-label predictions; it must again learn to only use the specified feature within the passed focus instructions alone for making predictions.

While these conditions represent an ideal setting, they are not strictly necessary for FIT to work in practice. Indeed, real-world data rarely satisfies such perfect independence, and we illustrate the robustness of the method in more realistic scenarios through our BBQ experiments in Section 5.1 where correlations between Y and S or between C and Y_S may exist as no subsampling or dataset manipulations have been made. By examining both the controlled environment and more naturalistic datasets, we demonstrate that our approach can handle scenarios with varying degrees of spurious correlations.

To achieve this independence in our synthetic SS and SMNLI datasets, we set $\rho_{\text{spurious}} = 1/N$, where N is the number of class labels. Additionally, we enforce a balanced label distribution in the training set to eliminate any indirect biases that could correlate S with Y. As shown in Appendix I and Appendix K, these conditions are sufficient to guarantee $Y \perp\!\!\!\perp S$ in the training data, enabling the model to effectively learn steerable behaviour from focus instructions.

D Evaluation Metrics

Generation settings. We generate responses from our FT model using constrained beam-decoding (Anderson et al., 2017) with 8 beams. This ensures that the answer labels for each classification task that we investigate appear in the model's output. We limit the maximum number of newly generated tokens to be 5 to stop any unnecessary text given after the model's initial classification prediction.

Computing the focus accuracy metric. We report the focus accuracy $\mathcal{A}_{\text{focus}}$ of generations when evaluating FT models. As we are guaranteed to include the task labels within the model's response through constrained decoding, we simply check to see if the focus label, y_{focus}, is within the model's response or not in order to determine if the model's response is correct.

E FIT Focus Instructions and Prompt Templates

Prompt templates. Here we provide the prompt templates that we use for FIT. Note, that we drop the feature considerations section and focus instruction from the prompt to form the instruction prompt $\emptyset$ with no focus instruction.

SS Focus instruction prompt templates $\mathcal{I}_{\text{focus}}$

<INSTRUCTION>
You are a language model performing sentiment analysis on a binary dataset, making predictions from the labels [negative, positive]. Make your prediction based on the relevant features described below, focusing on the specified features and ignoring those deemed irrelevant. For the input below, output either negative or positive ONLY for your prediction of the input's label.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{focus instruction - I_{focus}}
<END FEATURE CONSIDERATIONS>

Input: {input}
Output:

NLI focus instruction prompt template $\mathcal{I}_{\text{focus}}$

<INSTRUCTION>
You are a language model performing natural language inference (NLI) on a dataset with three labels [entailment, neutral, contradiction]. For the inputs consisting of premise and hypothesis below, make your prediction based on the relevant features described, focusing on the specified features and ignoring those deemed irrelevant. Output either entailment, neutral, or contradiction ONLY for your prediction of the input's label.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{focus instruction - I_{focus}}
<END FEATURE CONSIDERATIONS>

Premise: {premise}
Hypothesis: {hypothesis}
Output:

BBQ focus instruction prompt template $\mathcal{I}_{\text{focus}}$

<INSTRUCTION>
You are a language model answering a **QUESTION** based on a provided **CONTEXT**, choosing to **ANSWER** with one of three options: 1, 2, or 3. Each option corresponds to an answer from the provided ANSWER OPTIONS. You must output **EXACTLY** one of these numbers (1, 2, or 3) representing the answer you select.
<END OF INSTRUCTION>

<FEATURE CONSIDERATIONS>
{focus instruction - I_{focus}}
<END FEATURE CONSIDERATIONS>

CONTEXT: {context}
QUESTION: {question}
ANSWER OPTIONS:
(1) {ans0}
(2) {ans1}
(3) {ans2}
ANSWER:

Focus instructions. We consider the following focus instruction formats for the different focus instructions introduced in Equation (1) which are used for FIT training and evaluation:

Focus instructions $\mathcal{I}_{\text{focus}}$

For features $F_i, F_j \in \mathcal{F}$:

Focus instructions $\text{focus}(F_i)$:
- Direct your attention solely to F_i.
- Concentrate all your reasoning on F_i.
- Make F_i the central factor in your decision.
- Base your judgment exclusively on F_i.
- Pay attention only to F_i when making your prediction.
- Use F_i as the key input for your evaluation.
- Focus entirely on F_i and ignore other aspects.
- Rely exclusively on F_i to reach your conclusion.
- Consider only F_i and disregard all else.
- Let F_i be the primary basis for your decision.

Ignore instructions $\text{ignore}(F_i)$:
- Completely rule out F_i from your reasoning.
- Disregard any influence of F_i in your prediction.
- Treat F_i as irrelevant to your decision-making process.
- Exclude F_i entirely from your evaluation.
- Do not let F_i play any role in your assessment.
- Intervene to prevent F_i from affecting your prediction.
- Ensure that F_i has no bearing on your final decision.
- Block F_i from contributing to your reasoning.
- Negate the impact of F_i in your prediction.
- Ruling out F_i is crucial—do not let it affect your decision.

Focus and Ignore instructions $\text{focus}(F_i) \wedge \text{ignore}(F_j)$
- Focus specifically on F_i. Disregard F_j in your decision-making process.
- Base your prediction solely on F_i. Exclude F_j.
- Direct all your attention to F_i. Block out F_j from your prediction.
- Consider only F_i in your reasoning. Rule out F_j in your decision-making.
- Prioritize F_i. Completely ignore F_j in your prediction.
- Do not consider F_j in your decision-making process. Focus exclusively on F_i.
- Ignore any influence of F_j. Concentrate on F_i in your prediction.
- Disregard F_j entirely. Base your analysis solely on F_i.
- Rule out F_j in your prediction. Shift your focus to F_i.
- Do not pay attention to F_j in your decision-making process. Rely only on F_i.

F Additional Baselines Results

In addition to the baselines that we present in the main paper, Few-shot and SFT(y_{focus}), we include two additional baselines to further supplement these results. We give the complete list of baselines that we consider below:

Zero-shot baseline. Finally, we include a zero-shot inference baseline using the original pre-trained models without additional fine-tuning on our spurious datasets. No in-context examples are used at inference time, and the model is not trained at all beyond it's pre-training. The model is tested on the full set of focus instructions prompts detailed in Equation (1).

Few-shot baseline. This second baseline compares FIT training to few-shot inference using the original pre-trained models without additional fine-tuning on our spurious datasets. Specifically, we use 5 in-context examples across all datasets. For the in-context examples, we concatenate multiple examples one after the other, including the instructional prompt only for the first in-context example and the final test example. Each in-context example contains the same focus instruction as the test example for which they serve as context. The model is tested on the full set of focus instructions prompts detailed in Equation (1).

SFT(y_{focus}) baseline. We implement an SFT baseline that follows the same training procedure as FIT, except during training, we exclude any focus instructions from the input prompts while still training on the focus labels. This provides a fair comparison with FIT, as the models are trained on the same input text and label pairs. The rest of the training setup, including hyperparameters and early stopping, remains identical to the FIT training setup. The model is tested on the full set of focus instructions prompts detailed in Equation (1).

SFT(y) baseline. We implement a vanilla SFT baseline that simply trains a model using SFT on inputs and their ground truth labels (as opposed to focus labels in the SFT(y_{focus}) baseline). During training, only standard IT prompts are used, with no additional focus instructions included. The rest of the training setup, including hyperparameters and early stopping, remains identical to the FIT training setup. The model is tested on the full set of focus instructions prompts detailed in Equation (1).

We give the full set of results for all datasets and models across the complete set of baselines listed above in Figure 6, Figure 7, and Figure 8.

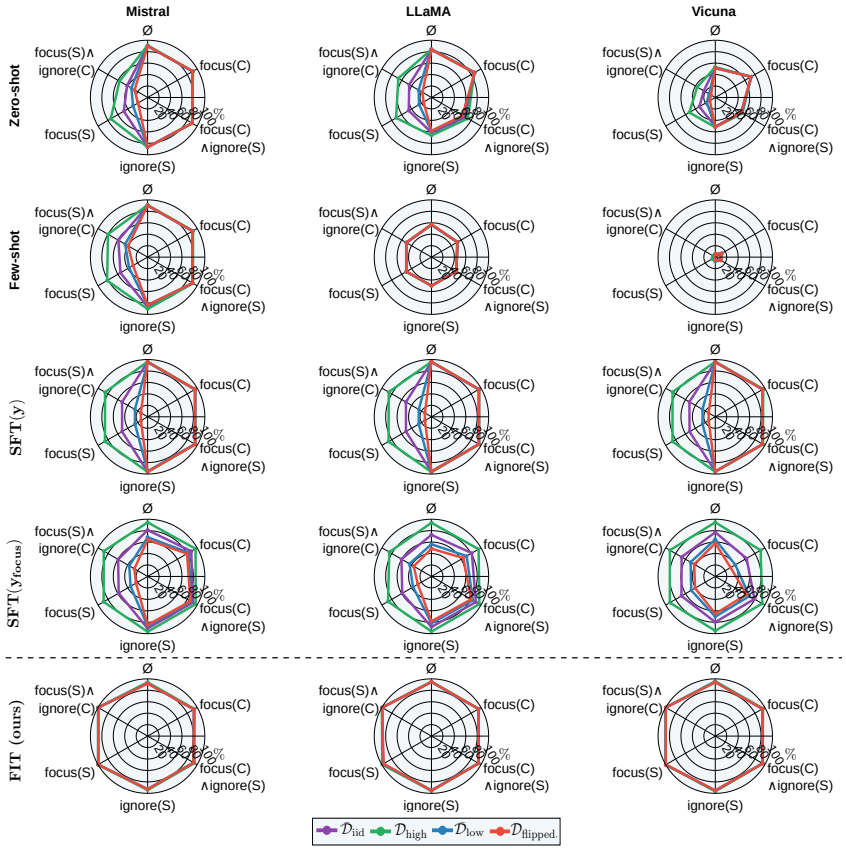

Figure 6: **Full Baseline vs FIT Focus accuracy (↑) on SS.** Figure giving focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the SS dataset.

G SMNLI Ablation of Training and Test time Focus Instruction Rephrasing Differences

We analyse the impact of using the same versus different sets of focus instructions at training and test time when applying FIT models. Specifically, we generate alternative test set focus instructions by paraphrasing the training focus instructions, as shown in Appendix E, using ChatGPT.

As depicted in Figure 9, the results of this ablation reveal negligible differences between using the same or different focus instruction phrasings during training and testing. This indicates that FIT

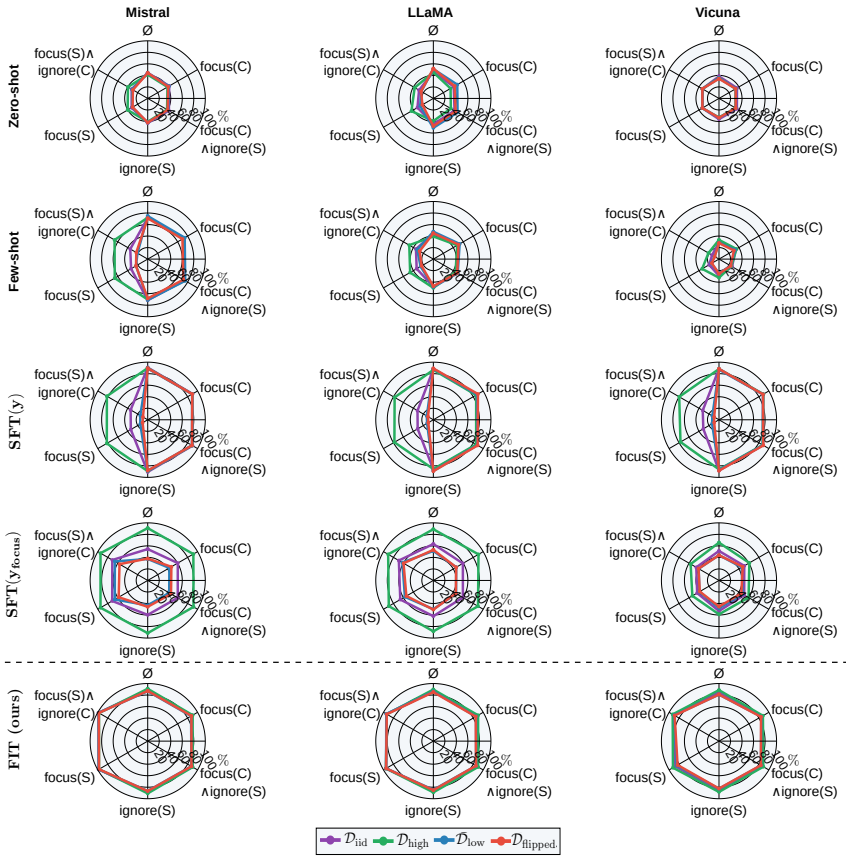

Figure 7: **Full Baseline vs FIT Focus accuracy (↑) on SMNLI.** Figure giving focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the SMNLI dataset.

effectively trains the model to focus on or ignore features, regardless of how the instructions are phrased.

H Comparison of FIT against a specific debiasing technique

FIT is a general framework designed to enable users to steer a model's behavior based on specified features. This approach provides enhanced control over model outputs during inference, adding a critical layer of explainability and controllability to model predictions.

While understanding and mitigating biases or spurious correlations is a valuable and natural application of FIT, it is not the sole objective. The broader goal of steerability includes addressing challenges in managing and aligning model behaviour across diverse contexts. For instance, maintaining controllability is crucial in addressing safety alignment fragility, which can emerge after fine-tuning (Bhattacharjee et al., 2024). In such cases, the ability to adapt model responses to align with user specifications ensures safe and reliable deployment.

Experiment. To explore FIT's broader applicability, we compare its performance as a debiasing method against a well-known debiasing technique: the Product of Experts (PoE) method (Mahabadi et al., 2020). PoE involves training a bias model f_B, which is trained exclusively on bias features. This bias model mediates the training of the final model f by combining their predictions through an elementwise product: $\sigma(f(x)) \odot \sigma(f_B(x_B))$, where $x \in \mathcal{D}$, for dataset $\mathcal{D}$, and x_B represents the biased feature of x.

We adapted this approach to our setting by training a bias model on the stereotypical labels within the BBQ dataset. These labels correspond to group-stereotypical associations. For autoregressive

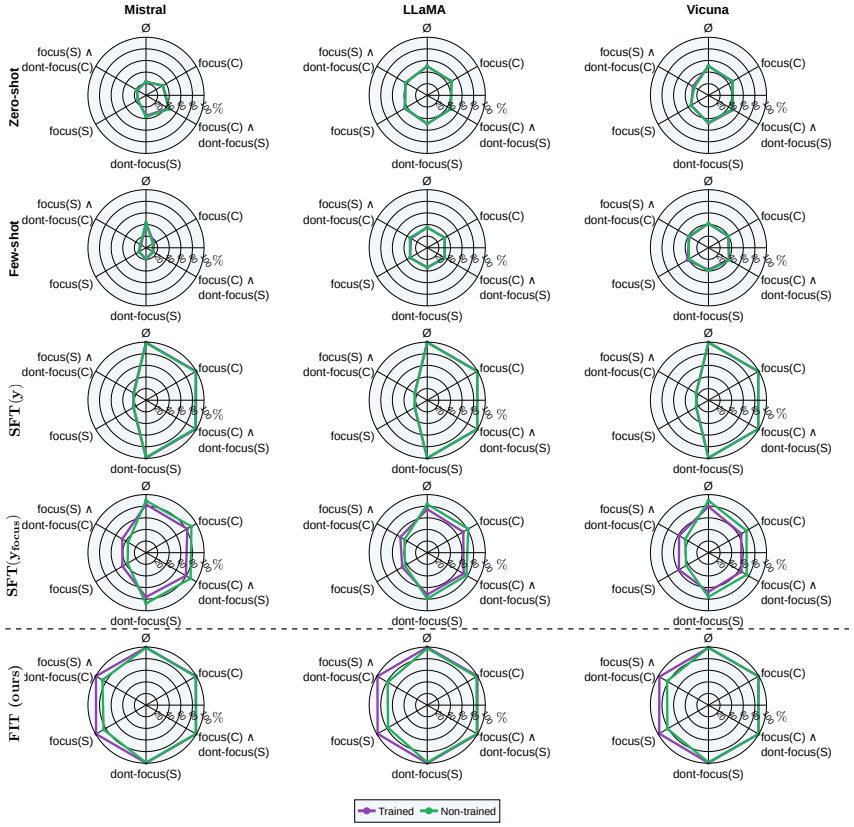

Figure 8: **Full Baseline vs FIT Focus accuracy (↑) on BBQ.** Figure giving focus accuracies ($\mathcal{A}_{\text{focus}}$) of the additional baselines compared to the focus accuracy of FIT on the BBQ dataset.

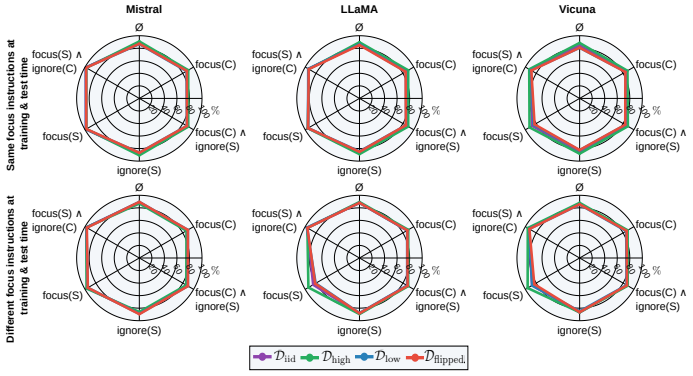

Figure 9: **Focus accuracy (↑) for different training and test focus instruction sets.** Figure comparing focus accuracies ($\mathcal{A}_{\text{focus}}$) of sampling from the same (top) and different (bottom) sets of focus instructions at training and test time of models on the SMNLI dataset.

models, we further modified the PoE method by extracting and normalising the logits of the first newly generated token position over the set of single tokens representing the answer options.

Results. The results of the debiasing experiment comparing FIT to the PoE method is shown in Figure 10. FIT performs equally as well as the PoE method as shown by the comparing the default prompt accuracy ($\emptyset$) for the PoE models against the focus(C) results for the FIT models; both metrics correspond to causal accuracy for these prompt types. Indicating that FIT performs just as well as a dedicated debiasing technique.

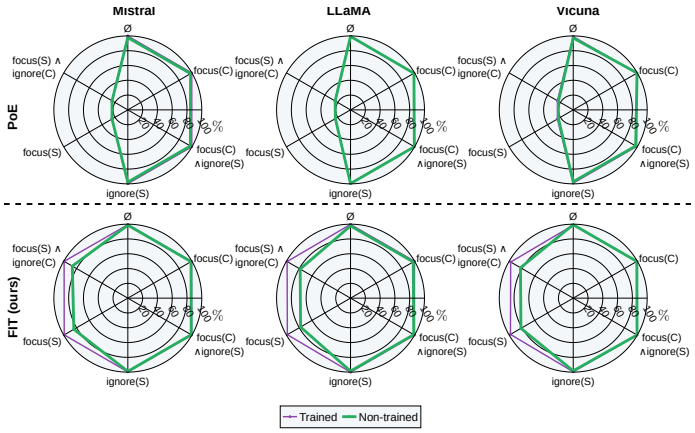

Figure 10: **Focus Accuracy (↑) of FIT against PoE Debiasing Technique.** Figure showing the focus accuracies ($\mathcal{A}_{\text{focus}}$) of FIT (bottom row) and the dedicated debaiasing technique, PoE (top row), on the BBQ dataset.

However, the PoE method requires training two separate models and does not provide steerability at test time as shown by the low focus accuracy on focus(S). Indeed the model defaults to the ground truth label across all prompt types and does not change behaviour despite different different focus specifications. This highlights the flexibility of FIT, which not only debiases effectively but also enables additional controllability during inference.

I SPURIOUS SENTIMENT (SS) DATASET

We take a pre-existing dataset, in this case SST-5 (Socher et al., 2013a), and modify it in order to induce a known spurious feature and create a spurious binary sentiment analysis dataset.

Data-generating process (DGP). We frame our DGP using a graphical model to describe the synthetic dataset that we create. We follow a similar model to that described in (Arjovsky et al., 2019), specifically the model used for generating their coloured MNIST dataset. We use the following variables within our graphical model:

- C - true underlying sentiment, the core feature within this task, sampled from the original dataset.
- $\tilde{S}$ - proposed spurious feature sample, here this is the presence of the keywords *Bayesian* or *Pineapple*. We represent this as a binary vector $S \in \{0,1\}^2$, where the first and second components of this vector denote the presence of either the keyword *Pineapple* or *Bayesian* respectively. We restrict to consider only one keyword appearing in a given text at a time so that Val$(S) = \{(1,0),(0,1)\}$.
- S - final spurious feature that is naturally inserted using a LLM into the final SS dataset example X. S is a randomly flipped version of the proposed spurious feature S.
- $\tilde{X}$ - is a sampled example from the original dataset that we are modifying to inject known spurious correlations.
- X - original example $\tilde{X}$ but augmented to include the spurious feature.
- Y - final label for element X.

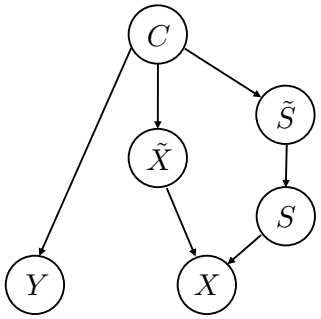

Figure 11: **SS DGP**. Graphical model showing the data generating process for modifying examples from the SST-5 dataset to introduce a new spurious keyword feature S.

The graphical model describing the DGP of the SS dataset is given in Figure 11. This admits a functional representation in the form:

$$C = f_C(U_C); \tag{7}$$
$$\tilde{X} = f(C, U_{\tilde{X}}); \tag{8}$$
$$\tilde{S} = f_{\tilde{S}}(C, U_{\tilde{S}}); \tag{9}$$
$$S = f_S(\tilde{S}, U_S); \tag{10}$$
$$X = f_X(\tilde{X}, S, U_X); \tag{11}$$
$$Y = f_Y(C, U_Y). \tag{12}$$

where $U_{(\cdot)}$ are variables introducing sources of randomness into the generating process. More explicitly, we consider the following set of equations, where $\mathcal{D}$ denotes the underlying dataset that we are manipulating:

$$C \sim \texttt{Ber}(\rho_C), \text{ where } \rho_C = \rho_C(\mathcal{D}); \tag{13}$$
$$\tilde{X} \sim p_{\mathcal{D}}(\cdot|C)\ , \tag{14}$$
$$\tilde{S} = (\mathbf{1}_{C=0}, \mathbf{1}_{C=1})\ ; \tag{15}$$
$$U_S \sim \texttt{Ber}(\rho_{\text{spurious}}); \tag{16}$$
$$S = U_S\tilde{S} + (1 - U_S)(1 - \tilde{S}); \tag{17}$$
$$U_{\text{incld.}} \sim \texttt{Ber}(\rho_{\text{incld.}}); \tag{18}$$
$$X = \texttt{LLM}(\tilde{X}, S); \tag{19}$$
$$Y = C, \tag{20}$$

The variable ρ_C gives the distribution of sentiment labels in the original binarised SST-5 dataset. Moreover, $p_{\mathcal{D}}(\tilde{x}|C)$ denotes the conditional dataset distribution of the different input texts give C (here we assume that we are just uniformly sampling text with the given sentiment C) and $\mathbf{1}_{(\cdot)}$ denotes the indicator function.

Finally, we prove that ρ_{spurious} gives the coocurrence rate/predictivity between the label Y and the spurious feature S, and is well-defined notation in the sense that it corresponds to Theorem 3.1 so that $\rho_{\text{spurious}}(s) = \mathbb{P}(Y = y_s|S = s)$, where y_s is the label that spurious feature value s is spuriously correlated with.

Proposition I.1. *From the SCEs described above, assuming that we have a balanced label distribution, that is $\rho_C = 1/2$, we have that*

$$\mathbb{P}(Y = y_s \mid S = s) = \rho_{\textit{spurious}}\ . \tag{21}$$

for all spurious feature values s.

Proof. First note that we have by Equation (13) and the assumption that $\rho_C = 1/2$, that $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 0) = 1/2$. Moreover, using the partition theorem, we have that

$$\mathbb{P}(S = s) = \mathbb{P}(S = s \mid \tilde{S} = s)\mathbb{P}(\tilde{S} = s) + \mathbb{P}(S = s \mid \tilde{S} \neq s)\mathbb{P}(\tilde{S} \neq s) \tag{22}$$

$$= \rho_{\text{spurious}} \cdot \frac{1}{2} + (1 - \rho_{\text{spurious}}) \cdot \frac{1}{2} \tag{23}$$

$$= \frac{1}{2}. \tag{24}$$

Finally note that $\mathbb{P}(S = s \mid Y = y_s) = \rho_{\text{spurious}}$ as $Y = y_s$ forces $C = y_s$, and therefore that $\tilde{S} = s$ by construction.

Putting this all together and utilising Bayes' rule gives

$$\mathbb{P}(Y = y_s \mid S = s) = \frac{\mathbb{P}(S = s \mid Y = y_s)\mathbb{P}(Y = y_s)}{\mathbb{P}(S = s)} \tag{25}$$

$$= \frac{\mathbb{P}(S = s \mid Y = y_s)\mathbb{P}(Y = y_s)}{\mathbb{P}(S = s \mid \tilde{S} = s)\mathbb{P}(\tilde{S} = s) + \mathbb{P}(S = s \mid \tilde{S} \neq s)\mathbb{P}(\tilde{S} \neq s)} \tag{26}$$

$$= \frac{\rho_{\text{spurious}} \cdot \frac{1}{2}}{\frac{1}{2}} \tag{27}$$

$$= \rho_{\text{spurious}}. \tag{28}$$

which gives the result. □

Within our experiments in Appendix I.1, we always force the label distribution to be balanced, that is $\rho_C = 1/2$, and assume that within each dataset split, ρ_{spurious} is the same rate for all spurious feature values. **Independence conditions during training for FIT.** As specified in Appendix C, we would like to have that $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ during training so that models trained via FIT can effectively learn to leverage focus instructions to make predictions based on specified features. Here, Y_S is the spurious label spuriously correlated to spurious feature value S. The results below give sufficient conditions for these independence conditions to be specified with respect to the DGP described in Figure 11.

Proposition I.2. *Assuming the SCEs given above and the corresponding DGP described in Figure 11, if we have a uniform label distribution $p(Y = y)$, that is $\rho_C = 1/2$, and have that $\rho_{spurious} = 1/2$ in the SS training set, then we have that $Y \perp\!\!\!\perp S$.*

Proof. From Theorem I.1, we have that $\mathbb{P}(Y = y_s|S = s) = 1/2$ when $\rho_{\text{spurious}} = 1/2$ for any spurious feature value $s \in \text{Val}(S)$. A balanced label distribution implies that $\mathbb{P}(Y = y) = 1/2$, for $y \in \text{Val}(Y)$. Therefore we have that $\mathbb{P}(Y = y|S = s) = \mathbb{P}(Y = y)$ for all $y \in \text{Val}(Y)$ and $s \in$ Val(S), which gives that $Y \perp\!\!\!\perp S$. □

Proposition I.3. *Assuming the SCEs given above and the corresponding DGP described in Figure 11, if we have a uniform label distribution $p(Y = y)$, that is $\rho_C = 1/2$, and have that $\rho_{spurious} = 1/2$ in the SS training set, then we have that $Y_S \perp\!\!\!\perp C$.*

Proof. First note that Y_S is a deterministic function of S, that is $Y_S = f(S)$ for some function $f : \text{Val(S)} \to \{0, 1\}$. Therefore, it is sufficient to show that $C \perp\!\!\!\perp S$. Starting from $\mathbb{P}(S = s \mid C = c)$

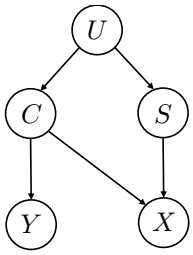

Figure 12: **SS SCM.** SCM showing showing the spurious correlation present between the keyword feature S and the label Y of examples within the SS dataset, induced through the described data augmentation process.

and marginalising over $\tilde{S}$, we have that

$$\mathbb{P}(S = s \mid C = c) = \sum_{\tilde{s}} \mathbb{P}(S = s \mid C = c, \tilde{S} = \tilde{s}) \mathbb{P}(\tilde{S} = \tilde{s} \mid C = c) \tag{29}$$

$$= \sum_{\tilde{s}} \mathbb{P}(S = s \mid \tilde{S} = \tilde{s}) \mathbb{P}(\tilde{S} = \tilde{s} \mid C = c) \tag{30}$$

$$= \underbrace{\mathbb{P}(S = s \mid \tilde{S} = s)}_{=\rho_{\text{spurious}}} \underbrace{\mathbb{P}(\tilde{S} = s \mid C = c)}_{=\frac{1}{2}} \tag{31}$$

$$+ \underbrace{\mathbb{P}(S = s \mid \tilde{S} \neq s)}_{=1-\rho_{\text{spurious}}} \underbrace{\mathbb{P}(\tilde{S} \neq s \mid C = c)}_{\frac{1}{2}} \tag{32}$$

$$= \frac{1}{2}. \tag{33}$$

Due to having a balanced label distribution with $\rho_C = 1/2$, we have that $\mathbb{P}(Y = y) = 1/2$ for all $y \in \text{Val}(Y)$. Therefore, we have that $\mathbb{P}(Y = Y_s \mid C = c) = \mathbb{P}(Y = Y_s)$ for all $y_s \in \text{Val}(Y_S)$ and $c \in \text{Val(C)}$. This gives that $S \perp\!\!\!\perp C$ under the assumptions in the proposition, and therefore due to the deterministic relationship between S and Y_S, that $Y_S \perp\!\!\!\perp C$. □

SCM from this DGP. Through the above data generation process, we introduce a new spurious feature within the dataset S, the presence of the keywords *Bayesian* and *Pineapple*. Recalling that $S = (1, 0)$ and $S = (0, 1)$ correspond to insertion of the keywords *Pineapple* and *Bayesian* respectively, we introduce the following spurious correlations between feature values of S and label Y:

1. The presence of the word *Pineapple* in the text X is spuriously correlated the label 0 (negative sentiment).

2. The presence of the word *Bayesian* in the text X is spuriously correlated with the label 1 (positive sentiment).

The sentiment feature still remains core within the augmented SS dataset, fully predicting the label Y for each dataset example.

The above DGP, through the introduction of spurious feature S, induces a SCM that describes the spurious correlation between spurious feature S and the label Y. The SCM, shown in Figure 12, follows the style-content decomposition described in (Kaddour et al., 2022), where U is some hidden confounding variable.

Data generation methodology. We use `Llama-3.1-70B-Instruct` to generate modifications X of original dataset examples $\tilde{X}$ to create new text which include the new keywords feature. The prompt we use for generation when modifying examples to include spurious features is give as:

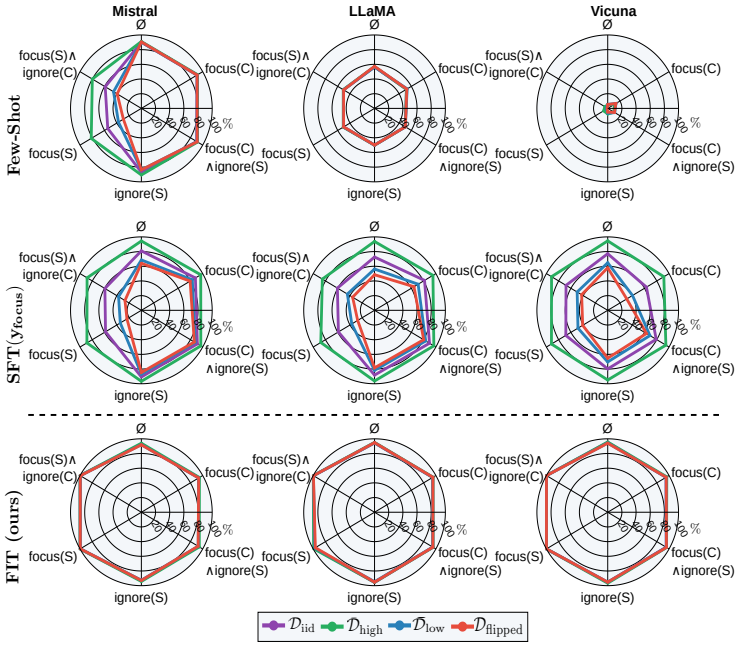

Figure 13: **SS focus accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of baselines and FIT on the SS dataset.

Data augmentation prompt

You are a language model designed to modify a piece of text to include an additional feature in a simple, natural way while keeping your output as similar as possible to the original text.

Features

- pineapple: Include the word 'pineapple'.
- Bayesian: Include the word 'Bayesian'.

Instructions

- Ensure the output is grammatically correct.
- Keep the output as similar as possible to the original text.
- Make the minimal number of modifications and add the fewest new tokens possible to satisfy the chosen feature.
- Do not change the sentiment of the original text.
- Do not significantly alter the length of the output.
- Incorporate the feature naturally within the original text so that it blends seamlessly with the text's context.
- Do not only append additional clauses at the end of the text to include the feature.
- Inclusions should be case sensitive, e.g., include 'Bayesian' BUT NOT 'bayesian'.

Output

- Only return the modified text, with no additional explanations or reasoning.
- Should strictly follow the feature description and the set of instructions.
- Only include the one feature given; the other features SHOULD NOT be included even accidentally.

I.1 Results of FIT on the SS dataset

We first evaluate FIT on a synthetic binary sentiment analysis dataset. Starting with SST-5 (Socher et al., 2013a), a 5-class sentiment analysis dataset, we use `Llama-3.1-70B-Instruct` (Dubey et al., 2024) to inject the spurious keywords *Pineapple* and *Bayesian* into all dataset examples in a natural way.[1] In this process, we preserve the original sentiment of the dataset examples and combine categories of positive and negative labels into single classes, and ex-

[1]The LLM makes minimal edits to inputs, often inserting keywords or adding a few words for context. See Appendix I for further details.

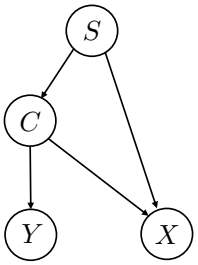

Figure 14: **SMNLI DGP**. Graphical model showing the data generating process for modifying examples from the MNLI dataset to introduce a new spurious keyword feature S.

clude examples with neutral labels from our augmented dataset. The feature set is given as $\mathcal{F} = \{$*sentiment, presence of keywords (Bayesian, Pineapple)*$\}$. We inject these features so that the presence of "Pineapple" and "Bayesian" are spuriously correlated with negative and positive sentiment, respectively. The degree of co-occurrence is governed by ρ_{spurious}, which varies according to the test sets described in Section 3. We ensure that ρ_{spurious} is the same for all feature values within each dataset split. In particular, we set ρ_{spurious} to be $0.5, 0.5, 0.9, 0.25$ and 0.9 on $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$, and $\mathcal{D}_{\text{flipped}}$ respectively. Further details of the SS dataset can be found in Appendix I.

Results. Figure 13 (a) shows the focus accuracy results of the three LLMs on the SS dataset using few-shot prompting, after SFT(y_{focus}) and after FIT. We see that across all focus instructions and all models, FIT shows significant improvement over the baselines, achieving very high focus accuracy across all focus instruction types and across all test sets with varying predictivity levels.

***Key takeaways*.** High focus accuracy on SS indicates that FIT successfully steers model responses based on the feature on which it is instructed to focus or to not focus on.

J SPURIOUS NLI DATASET (SMNLI)

We generate a tertiary NLI dataset, SMNLI, with a known spurious feature. We do this considering the MNLI dataset Williams et al. (2018). This is a NLI dataset with three labels: entailment (0), neutral (1) and contradiction (2), where data is sampled from 5 underlying categories or genres (telephone, government, travel, fiction or slate). We aim to induce spurious correlations between the underlying genres and labels.

Data-generating process (DGP). We consider a graphical model to describe the DGP of examples within the SMNLI dataset. We use the following variables within our DGP:

- C - NLI relationship between a premise and hypothesis pair, the core feature within this task, sampled from the original dataset.
- S - spurious feature, here this is the genre of the premise and hypothesis. This is a categorical variable.
- X - example from the MNLI dataset.
- Y - final label for element X.

The graphical model described by the DGP for producing the SMNLI dataset is given in Figure 14. Once again, this graphical model can be represented functionally as:

$$S = f_S(U_S); \tag{34}$$
$$C = f_C(S, U_C); \tag{35}$$
$$X = f_X(C, E, U_X); \tag{36}$$
$$Y = f_Y(C, U_Y). \tag{37}$$

More specifically, given the orignal dataset $\mathcal{D}$ that we are sub-sampling from, the functions that we use within the DGP for the SMNLI dataset are given by:

$$S \sim \texttt{Cat}(\mathcal{S}), \tag{38}$$

$$U_C \sim \texttt{Ber}(\rho_{\text{spurious}}); \tag{39}$$

$$C = \begin{cases} y_S & \text{if } U_C = 1; \\ \sim \texttt{Unf}(\text{Val}(Y) \setminus \{y_S\}) & \text{if } U_C = 0; \end{cases}; \tag{40}$$

$$X \sim p_{\mathcal{D}}(\cdot|C, S) \tag{41}$$

$$Y = C. \tag{42}$$

Here, $\texttt{Cat}(\mathcal{S})$ is a uniform categorical distribution over spurious feature values (here the underlying genre of the premise-hypothesis pairs). Furthermore, we define y_S to be the NLI label that a particular value of S is spuriously correlated with by design. Moreover, $p_{\mathcal{D}}(x|C, S)$ is the conditional distribution over the dataset examples (premise-hypothesis pairs) that have NLI relationship C and genre S.

We restrict the genres within the model to $S \in \{\textit{slate}, \textit{government}, \textit{fiction}, \textit{travel}\}$, a subset of the genres of the training set. When creating a distribution shifted test set, we restrict the genres to $S \in \{\textit{facetoface}, \textit{nineeleven}, \textit{verbatim}\}$. The specific spurious correlations between a genre s and a label y_s are chosen to be: $y_{\textit{slate}} = 0$; $y_{\textit{government}} = 2$; $y_{\textit{fiction}} = 1$; $y_{\textit{travel}} = 0$; $y_{\textit{facetoface}} = 2$; $y_{\textit{nineeleven}} = 0$; $y_{\textit{verbatim}} = 1$.

In this way we generate spurious correlations within the dataset through sub-sampling to induce spurious correlations between S and Y.

We show that the notion of ρ_{spurious} in Equation (39) aligns with the notation in Theorem 3.1.

Proposition J.1. *From the SCEs described above, we have that*

$$\mathbb{P}(Y = y_s \mid S = s) = \rho_{\textit{spurious}} \, . \tag{43}$$

for all spurious feature values s.

Proof. This is clear considering Equation (40), where ρ_{spurious} influences the chance that we sample y_s, i.e. the label spuriously correlated with feature value s. □

SCM for SMNLI. The DGP again induces a SCM that induces spurious correlations between spurious features S and the label Y. The SCM has the same structure as in the SS dataset, and is given in Figure 17 where once again, U again is some hidden confounding variable.

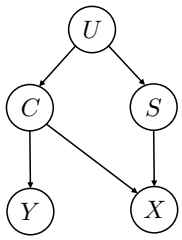

Figure 15: **SMNLI SCM.** SCM showing showing the spurious correlation present between the keyword feature S and the label Y of examples within the SMNLI dataset, induced through the described sub-sampling process of the MNLI dataset.

Independence conditions during training for FIT. As specified in Appendix C, we would like to have that $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ during training so that models trained via FIT can effectively learn to leverage focus instructions to make predictions based on specified features, where, again, Y_S is the label spuriously associated to spurious feature value S. The results below give sufficient conditions for this to occur with respect to the DGP described in Figure 14.

Proposition J.2. *Assuming the SCEs given above and the corresponding DGP described in Figure 14, if we have a uniform label distribution $p(Y = y)$, and have that $\rho_{\textit{spurious}} = 1/3$ in the SMNLI training set, then we have that $Y \perp\!\!\!\perp S$.*

Proof. From Theorem J.1, we have that $\mathbb{P}(Y = y_s \mid S = s) = 1/3$ when $\rho_{\text{spurious}} = 1/3$ for any spurious feature value $s \in \text{Val}(S)$. Moreover, we have that from Equation (40)

$$\mathbb{P}(Y = y \mid S = s) = \mathbb{P}(C = y \mid S = s) \tag{44}$$

$$= \underbrace{\mathbb{P}(U_C = 1)}_{=\frac{2}{3}} \underbrace{\mathbb{P}(C = Y | S = s)}_{=\frac{1}{2}} \tag{45}$$

$$= \frac{1}{3}, \tag{46}$$

for each $y \in \text{Val}(Y) \setminus \{y_s\}$.

Using this alongside the assumed balanced label distribution, $\mathbb{P}(Y = y) = 1/3$ for $y \in \text{Val}(Y)$, implies that we have $\mathbb{P}(Y = y|S = s) = \mathbb{P}(Y = y)$ for all $y \in \text{Val}(Y)$ and $s \in \text{Val}(S)$, which gives that $Y \perp\!\!\!\perp S$. □

Proposition J.3. *Assuming the SCEs given above and the corresponding DGP described in Figure 14, if we have a uniform label distribution $p(Y = y)$, and have that $\rho_{spurious} = 1/3$ in the SMNLI training set, then we have that $Y_S \perp\!\!\!\perp C$.*

Proof. Note that Y_S is a deterministic function of S, so that $Y_S = f(S)$ for some function $f : Val(S) \to \text{Val(Y)}$. Therefore, it suffices to show that $S \perp\!\!\!\perp C$.

First, using that $\rho_{\text{spurious}} = 1/3$, we have from Equation (40) that

$$\mathbb{P}(C = c \mid S = s) = \begin{cases} 1/3 & \text{if } c = f(s), \\ 2/3 \cdot 1/2 & \text{else.} \end{cases} \tag{47}$$

$$= \frac{1}{3}, \tag{48}$$

for all $c \in \text{Val(C)}$ and $y \in \text{Val(Y)}$.

Now we consider the marginal distribution of C. Let $|\text{Val}(S)| = k$ be the number of spurious feature values. Using the previous result, we have that:

$$\mathbb{P}(C = c) = \sum_s \underbrace{\mathbb{P}(C = c \mid S = s)}_{=1/3} \underbrace{\mathbb{P}(S = s)}_{1/k} \tag{49}$$

$$= \sum_s \frac{1}{3k} \tag{50}$$

$$= \frac{1}{3}. \tag{51}$$

Therefore, we have that $\mathbb{P}(C = c \mid S = s) = \mathbb{P}(C = c)$, for all $c \in \text{Val}(C)$ and $s \in Val(S)$, which then implies that $S \perp\!\!\!\perp C$. Therefore, using that the spurious label Y_S is a deterministic function of S, we have that $Y_S \perp\!\!\!\perp C$ under the assumptions within the proposition. □

K SPURIOUS HANS DATASET (SHANS)

We generate a binary NLI dataset, SHANS, with a known spurious feature. We do this considering the HANS dataset McCoy (2019). This is an NLI data set with two labels: entailment (0) and contradiction (1). This is an adversarial dataset designed to assess different NLI models' reliance on spurious heuristics rather than on the underlying relationship between the premise and the hypothesis when making predictions. Specifically, the author's consider three major categories of heuristics: lexical overlap heuristic (assuming that a premise entails from words within the hypothesis) , sub-sequence heuristic(assuming that the premise entails all any of its contiguous sub-sequences of words) and constituent heuristic (assuming that a premise entails a hypothesis that is any constituent within it's syntactic parse tree).

Data-generating process (DGP). We consider a graphical model to describe the DGP of examples within the SHANS dataset. We use the following variables within our DGP:

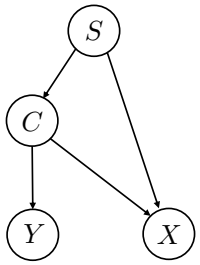

Figure 16: **SHANS DGP**. Graphical model showing the data generating process for modifying examples from the SHANS dataset to introduce a new spurious features $S_{\text{lex.}}$, $S_{\text{sub.}}$ and $S_{\text{const.}}$ which are encoded within the categorical spurious feature S which represents one of these three heuristics.

- C - NLI relationship between a premise and hypothesis pair, the core feature within this task, sampled from the original dataset.
- $S_{\text{lex.}}$ - spurious feature, here the presence of a hypothesis entirely made from words from the premise. This is a binary categorical variable (present/ not present).
- $S_{\text{sub.}}$ - spurious feature, here the presence of a hypothesis that is a contiguous subsequence of the premise. This is a binary category feature (present/ not present).
- $S_{\text{const.}}$ - spurious feature, here the presence of hypothesis that is a constituent/subtree of the premise. Here we have a binary variable (present/ not present).
- X - example from the HANS dataset.
- Y - final label for element X.

The graphical model described by the DGP for producing the S-HANS dataset is given in Figure 16. Once again, this graphical model can be represented functionally as

$$S = f_S(U_S); \tag{52}$$
$$C = f_C(S, U_C); \tag{53}$$
$$X = f_X(C, E, U_X); \tag{54}$$
$$Y = f_Y(C, U_Y), \tag{55}$$

where here we define S to be a categorical feature over the set of the presence of each of the three heuristics introduced above which we denote, through overloaded notation, by $\mathcal{S} = \{s_{\text{lex.}}, s_{\text{sub.}}, s_{\text{const.}}\}$. More specifically, given the original dataset $\mathcal{D}$ that we are sub-sampling from, the functions that we use within the DGP for the S-HANS dataset are given by:

$$S \sim \texttt{Cat}(\mathcal{S}), \tag{56}$$
$$U_C \sim \texttt{Ber}(\rho_{\text{spurious}}); \tag{57}$$
$$C \sim \begin{cases} y_S & \text{if } U_C = 1; \\ \sim U(\text{Val}(Y) \setminus \{y_S\}) & \text{if } U_C = 0; \end{cases}; \tag{58}$$
$$X \sim p_{\mathcal{D}}(\cdot|C, S) \tag{59}$$
$$Y = C. \tag{60}$$

Here, $\texttt{Cat}(\mathcal{S})$ is a uniform categorical distribution over $\mathcal{S}$ which effectively selects the precence of exactly one of the three spurious feature heuristics. We define y_S to be the NLI label that a particular value of S is spuriously correlated with by design. Moreover, $p_{\mathcal{D}}(x|C, S)$ is the conditional distribution over the dataset examples (premise-hypothesis pairs) that have NLI relationship C and the presence of spurious heuristics S.

We consider the presence of each feature to be separate binary spurious features. The specific spurious correlations between heuristics and labels Y are chosen to be: $y_{S_{\text{lex.}}=1} = 0$; $y_{S_{\text{sub.}}=1} = 0$; $y_{S_{\text{const.}}=1} = 1$.

In this way we generate spurious correlations within the dataset through sub-sampling to induce spurious correlations between the heuristics and Y.

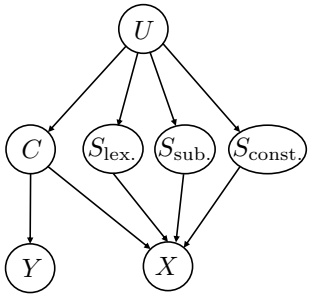

Figure 17: **SHANS SCM.** SCM showing the spurious correlations present between the binary precence of heurstics features $S_{\text{lex.}}$, $S_{\text{sub.}}$ and $S_{\text{const.}}$ and the label Y of examples within the S-HANS dataset, induced through the described sub-sampling process of the S-HANS dataset.

Transferred results from SMNLI. As we use the same DGP as for the SMNLI dataset described in Appendix K, all of the results that we have proven for SMNLI, translate to the SHANS dataset. In particular, we have that ρ_{spurious} aligns with the notation in Theorem 3.1, and that we have $Y \perp\!\!\!\perp S$ and $Y_S \perp\!\!\!\perp C$ under the assumptions of balanced label distributions and $\rho_{\text{spurious}} = 1/2$ within the training set used for FIT training.

SCM for SHANS. The DGP again induces a SCM. In particular, considering S as consiting of three binary spurious features $s_{\text{lex.}}$, $s_{\text{sub.}}$ and $s_{\text{const.}}$. The SCM has a similar structure to as in the SS and S-MNLI datasets, and is given in Figure 17 where once again, U again is some hidden confounding variable.

L FIT ON SHANS

Here we give the results of performing SFT and FIT on the SHANS datasets.

Spurious HANS (SHANS) dataset. We generate binary NLI dataset sub-sampled from HANS (McCoy, 2019), a dataset designed to challenge NLI models by exposing common heuristics they rely on, such as lexical overlap (whether the hypothesis shares many words with the premise), subsequence (whether the hypothesis is a contiguous sub-sequence of the premise), and constituent (whether the hypothesis is a grammatical sub-structure of the premise). The presence of these heuristics are spuriously correlated with labels through sub-sampling of the presence of each of the heuristics from the original dataset. The degree of co-occurrence is governed by ρ_{spurious}, which varies according to the test sets described in Section 3. We ensure that ρ_{spurious} is the same for all feature values within each dataset split. In particular, we set ρ_{spurious} to be $0.5, 0.5, 0.9, 0.25$ and 0.9 on $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{iid}}$, $\mathcal{D}_{\text{high}}$, $\mathcal{D}_{\text{low}}$ and $\mathcal{D}_{\text{flipped}}$ respectively.

Results. Figure 18 shows the focus accuracy results of performing SFT and FIT on the SHANS dataset for the `Llama-3.1-8B-Instruct`model. As expected, the trained features show high focus accuracy. However, for non-trained features, we observe lower focus accuracy. This could be attributed to the overlapping nature of the heuristics in SHANS, which are often graded versions of each other with different levels of specificity. For instance, the sub-sequence heuristic can overlap with both lexical overlap and constituent heuristics (e.g., the example with Premise:"Before the actor slept, the senator ran" and Hypothesis: "The actor slept." satisfies all three heuristics). This overlap likely confuses the model during generalisation, as it struggles to distinguish between heuristics not seen during training and those that are similar. These results suggest a potential limitation of FIT when dealing with features that are not sufficiently distinct or have significant overlap.

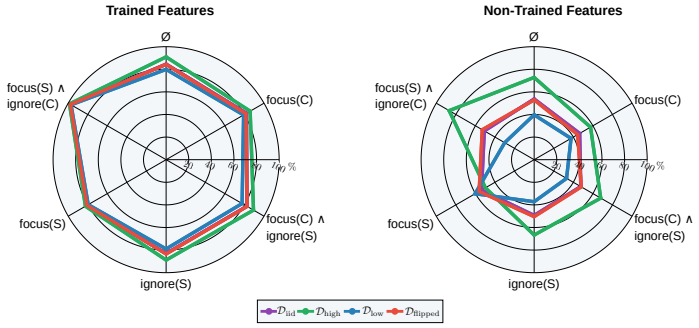

Figure 18: **SHANS focus accuracies (↑).** Focus accuracy ($\mathcal{A}_{\text{focus}}$) of `Llama-3.1-8B-Instruct`after FIT on the SHANS dataset. Here, C refers to the core feature (logical relationship between premise and hypothesis) and S the spurious feature (heuristic used)