Reinforcement Learning Fine-tuning of Language Models is Biased Towards More Extractable Features

Diogo Cruz[1] **Edoardo Pona**[1]
Alex Holness-Tofts[1] **Elias Schmied**[1] **Victor Abia Alonso**[1]
Charlie Griffin[1] **Bogdan-Ionuţ Cîrstea**[1]
[1]AI Safety Hub Labs

Abstract

Many capable large language models (LLMs) are developed via self-supervised pre-training followed by a reinforcement-learning fine-tuning phase, often based on human or AI feedback. During this stage, models may be guided by their inductive biases to rely on simpler features which may be easier to extract, at a cost to robustness and generalisation. We investigate whether principles governing inductive biases in the supervised fine-tuning of LLMs also apply when the fine-tuning process uses reinforcement learning. Following Lovering et al. [2021], we test two hypotheses: that features more *extractable* after pre-training are more likely to be utilised by the final policy, and that the evidence for/against a feature predicts whether it will be utilised. Through controlled experiments on synthetic and natural language tasks, we find statistically significant correlations which constitute strong evidence for these hypotheses.

1 Introduction

Most capable large language models (LLMs) are developed using self-supervised pre-training, where they learn representations of various features, followed by a reinforcement-learning (RL) fine-tuning phase, during which they learn to utilise these features to perform a specific task according to human preferences [Christiano et al., Ziegler et al., 2020, Stiennon et al., 2020, Ouyang et al., 2022]. The reward signal provided during the fine-tuning process under-determines the behaviour of the learned policy on data outside the training distribution [D'Amour et al., 2022, Jayawardana et al., 2022].

[Lovering et al., 2021] demonstrated that LLM supervised fine-tuning exhibits the following inductive bias: fine-tuned models are more likely to rely on features that are more extractable after pre-training, even if these features have less predictive power. We examine whether this inductive bias also holds for LLMs fine-tuned via RL and when the reward is provided by another learned model [Ziegler et al., 2020, Stiennon et al., 2020]. Specifically, our contribution is to test the following hypotheses about the policy of the fine-tuned model:

Extractability hypothesis: features which score higher in extractability for the pre-trained model are relied upon more by the RL fine-tuned model (policy).

Evidence hypothesis: the more evidence there is for/against a feature during RL fine-tuning, the more likely the model learns a policy that relies on that feature to get a high reward (policy).

After providing key terminology in Section 2, we explain the experimental setup (Section 3) and present evidence in support of the extractability hypothesis in Section 4. Our key result is Fig. 3. We then discuss our results (Section 5) and contrast them against existing work (Section 6).

Socially Responsible Language Modelling Research (SoLaR) Workshop at 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Background

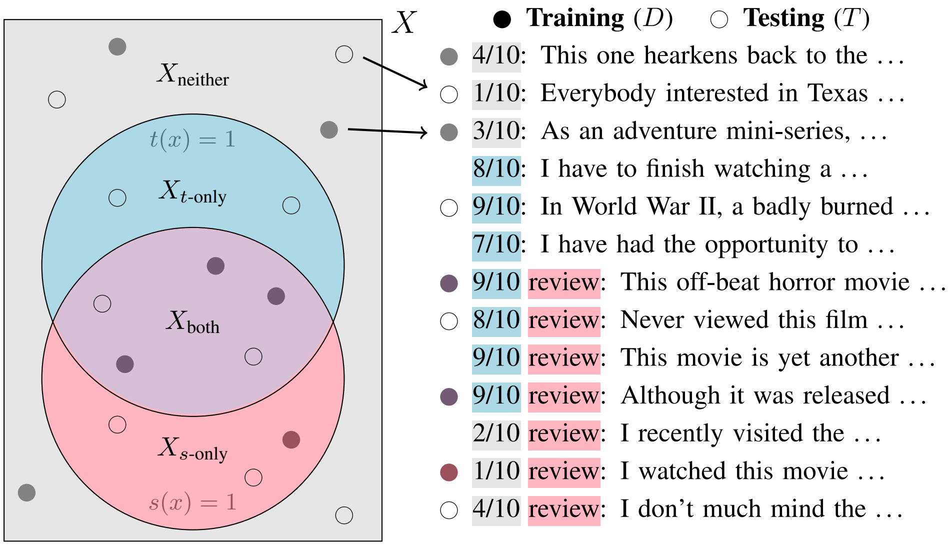
We modify the supervised fine-tuning setup of Lovering et al. [2021] so that - instead of a binary classification task - the pre-trained language model receives a reward signal from a reward model trained using human labels [Ziegler et al., 2020, Stiennon et al., 2020, Christiano et al., 2018, Bai et al., 2022]. We adopt the definitions of **evidence** and **extractability** from [Lovering et al., 2021], while adapting the definitions of **target** and **spurious** features to suit a reinforcement learning setting.

The reinforcement learning problems we consider vary in the reward functions but share common state and action spaces. Let X be the set of all possible natural language prompts for the task of interest. Consider the setup where an LLM takes a prompt text $x \in X$ and produces a response y. The initial state distribution is made by sampling from a training dataset. During the RL fine-tuning process, there may be a **target feature** $t : X \mapsto \{0, 1\}$ in the training data whose presence ($t(x) = 1$) or absence ($t(x) = 0$) determines the goal for that prompt. That is, the reward function scoring the prompt-response pair (x, y) can be described as

$$R(x,y) = \begin{cases} R_0(x,y), & \text{if } t(x) = 0 \\ R_1(x,y), & \text{if } t(x) = 1 \end{cases} \tag{1}$$

where R_0 and R_1 do not depend on t. Along with the target feature, there may be **spurious features** s in the training prompts, whose presence and absence correlate with that of the target feature. The LLM may then learn to get high reward by relying on s instead of t during RL fine-tuning. Our study considers the simplified scenario where only one spurious feature may be present in the prompt.

3 Experimental setup



● **Training** (D) ○ **Testing** (T)

● 4/10: This one hearkens back to the ...
○ 1/10: Everybody interested in Texas ...
● 3/10: As an adventure mini-series, ...
8/10: I have to finish watching a ...
○ 9/10: In World War II, a badly burned ...
7/10: I have had the opportunity to ...
● 9/10 review: This off-beat horror movie ...
○ 8/10 review: Never viewed this film ...
9/10 review: This movie is yet another ...
● 9/10 review: Although it was released ...
2/10 review: I recently visited the ...
● 1/10 review: I watched this movie ...
○ 4/10 review: I don't much mind the ...

Figure 1: We partition X into four sets, defined by which features (target t and spurious s) apply for each prompt. We partition the training data D (filled dots) analogously into $D_{s\text{-only}}$, $D_{t\text{-only}}$, D_{neither} and D_{both}, and similarly for the testing data T (hollow dots). This example presents the controlled sentiment task `score`. We prepend a rating out of 10 at the beginning. The target feature is present if the rating is more than 6/10 and absent otherwise. The spurious feature is the presence of the word "review" prepended to the rating. Note that t-only examples only appear during testing, never training.

For our experiments, we use a GPT-2 model. For clarity, here we present the results for the smaller `gpt2` variant, for a single setup based on controlled sentiment generation. Similar results, obtained with `gpt2-large` and for other setups, can be seen in Appendix E. As the pre-trained GPT-2 model is biased towards generating positive sentiment text, we start with an unbiased warmed-up GPT-2 model (from `https://huggingface.co/lvwerra/gpt2-imdb`), which we fine-tune on controlled sentiment generation tasks using proximal policy optimisation (PPO). To produce the

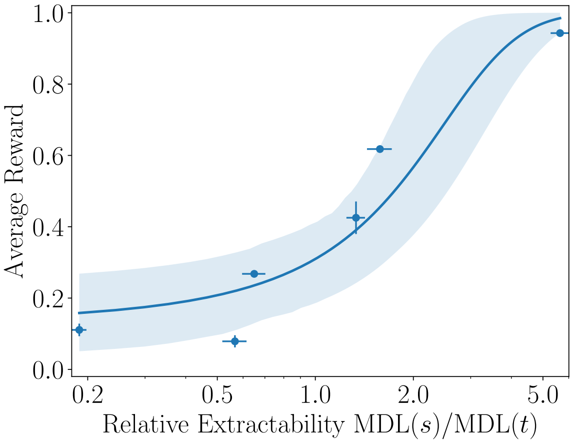


Figure 2: **Extractability hypothesis**. The average reward in $T_{s\text{-only}}$ and $T_{t\text{-only}}$ (for $p = 0$) is positively correlated with the relative MDL of the studied tasks. The blue line marks a logistic regression with a 95% confidence interval.

training prompts, we modify the IMDb dataset [Maas et al., 2011] by introducing target-spurious feature pairs chosen to cover a wide variety of relative extractabilities (see Appendices A and B). The training (resp. testing) prompts are a sampled subset D (resp. T) of the whole dataset X (see Fig. 1).

The **evidence** against a spurious feature is equivalent to the s-only example ratio $p = |D_{s\text{-only}}|/|D|$: the proportion of training examples in which s occurs without t. For a given p, the training data D will be composed of $p|D|$ examples from $D_{s\text{-only}}$, $\frac{1-p}{2}|D|$ from D_{both}, and $\frac{1-p}{2}|D|$ from D_{neither}.

The **extractability** of a feature refers to how simply-represented a feature is by a model and, therefore, how easy it is for the model to detect the feature's occurrence in the input during fine-tuning. We quantify this as the **minimum description length (MDL)** following the methodology in [Voita and Titov, 2020, Lovering et al., 2021]. To compute the MDL of a feature for a given model, a classifier is trained on a dataset labelled $y = \{0, 1\}$, denoting the presence or absence of the feature. MDL measures the number of bits needed to transmit the labels and model given the inputs. A smaller MDL value implies that the probe has quickly converged to high accuracy, suggesting that the feature is easily detectable and, therefore, has high extractability. Our results primarily focus on the *relative extractability* of the target vs. the spurious feature, given by the ratio MDL(s)/MDL(t).

During RL fine-tuning, the model is rewarded for generating positive sentiment text if the target feature is present, and for producing negative sentiment otherwise. The reward signal for each generated sequence comes from a model (from `https://huggingface.co/lvwerra/distilbert-imdb`) which is a fine-tuned LLM on sentiment classification that produces a score $\mathcal{M}(w)$ of how positive a text w is. We use the reward function

$$R(x, y) = \begin{cases} \mathcal{M}(x + y), & \text{if } t(x) = 1 \\ -\mathcal{M}(x + y), & \text{if } t(x) = 0 \end{cases} \tag{2}$$

where $+$ stands for string concatenation. In practice, $\mathcal{M}$ produces a bounded score, so we rescale the resulting R to be in $[0, 1]$.

4 Results

The extractability hypothesis predicts that models trained on tasks with more extractable target features are more likely to learn to rely on these features. As a result, these models will receive higher reward in training, specifically on s-only and t-only prompts. Our experiments support this hypothesis.

To study the feature extractability in these tasks, we may analyse the reward obtained for $T_{s\text{-only}}$ and $T_{t\text{-only}}$ when there is no evidence in the training data to distinguish the target from the spurious feature (i.e. $p = 0$, and all training prompts are from D_{both} or D_{neither}). In this case, we expect that tasks

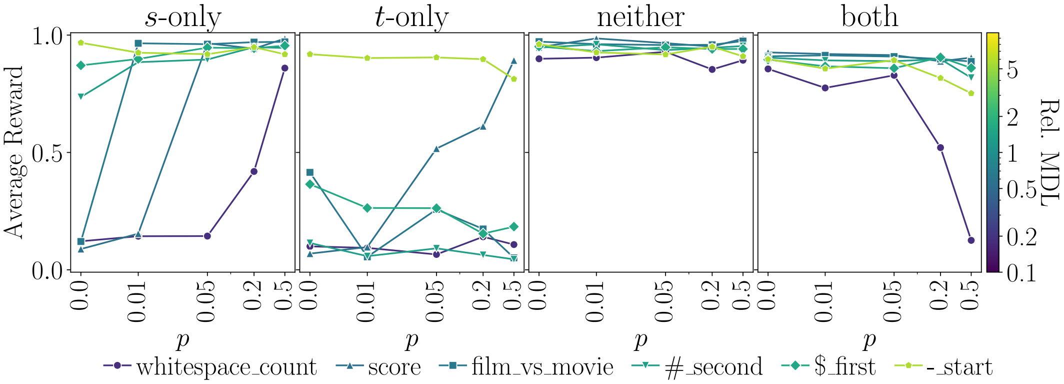


Figure 3: **GPT-2 performance on controlled sentiment tasks**. Average reward obtained by the fine-tuned model (on $T_{s\text{-only}}$,$T_{t\text{-only}}$,T_{neither},T_{both}) as the evidence $p = |D_{s\text{-only}}|/|D|$ varies. The tasks are ordered from left to right in increasing relative extractability of the target feature (i.e. increasing relative MDL). The task descriptions and MDL values can be found in Appendices A and B.

where the target feature t is easier to extract will lead the model to rely on t to distinguish prompts in D_{both} from prompts in D_{neither}, thereby obtaining high reward when testing prompts in $T_{s\text{-only}}$ and $T_{t\text{-only}}$. However, where t is harder to extract, the model may instead rely on the spurious feature to distinguish D_{both} from D_{neither}, consequently obtaining low reward for $T_{s\text{-only}}$ and $T_{t\text{-only}}$.

As seen in Fig. 2, this extractability hypothesis is, in fact, observed for the tasks considered, with the average reward in $T_{s\text{-only}}$ and $T_{t\text{-only}}$ increasing as the relative MDL increases, which is a proxy for how (relatively) easy the target feature is to extract.

In Fig. 3, we also observe data consistent with the evidence hypothesis: the more evidence p against the spurious feature, the more likely the RL fine-tuned GPT-2 is to learn a policy leading to higher reward for examples in $T_{s\text{-only}}$. That is, the model has learned that the spurious feature is irrelevant for the task at hand. The $T_{t\text{-only}}$ case is mixed, and the observed behaviour seems to depend on the specific task. For task `score`, higher p leads to higher reward in $T_{t\text{-only}}$, as expected from the evidence hypothesis. The model has inferred from the training data that the target feature is the relevant one to get maximum reward, and disregarded the spurious feature. However, for the remaining tasks, mostly with hard-to-extract target features, the RL fine-tuning procedure doesn't seem to lead the model to infer the optimal policy for $T_{t\text{-only}}$ solely from observing $D_{s\text{-only}}$, D_{neither} and D_{both}. The task `score` is possibly the only one tested where the target and spurious features are associated with specific tokens at specific prompt locations across the 4 training subsets, making it easier for the model to infer the target feature as it is exposed to more evidence. For the remaining tasks, the model may have learned instead to treat the "both", "neither", and "s-only" datasets separately, and fail to generalise to "t-only", or it may have inferred the wrong feature. For the easiest-to-extract target feature in task `-_start`, consistent with the extractability hypothesis, the model learned to rely on the target feature to get high reward for $T_{t\text{-only}}$, even for $p = 0$.

We also note a distinct behaviour when the target feature has low relative extractability and the training data has high evidence against the spurious feature. For task `whitespace_count`, we observe a sharp drop in reward in the T_{both} dataset. Although higher error rates were similarly observed in [Lovering et al., 2021] for high p, their magnitude was much lower. One explanation is that RL fine-tuning is less likely to lead to an optimal policy than supervised fine-tuning [Lovering et al., 2021]. In particular, when p is high, having a low fraction of training data showcasing the hard-to-extract target feature may lead the model to learn the suboptimal policy of behaving as if the target feature is never present. In this case, the model consequently learns to always generate negative sentiment completions, regardless of the prompt.

Combining the extractability hypothesis with the evidence hypothesis, we note that, the harder a target feature is to extract, the more evidence against the spurious feature is needed for the model to get high reward in $T_{s\text{-only}}$ (and $T_{t\text{-only}}$). The most extractable target features get high reward in $T_{s\text{-only}}$ regardless of the p value. Worsening extractability then requires more evidence, with tasks `film_vs_movie`, `score` and `whitespace_count` respectively requiring $p = 0.01, 0.05$ and 0.5 to get high reward in $T_{s\text{-only}}$.

5 Discussion

Our results align with findings on supervised fine-tuning [Lovering et al., 2021] - the relative extractability of the target and spurious features strongly predict inductive biases of reinforcement learning fine-tuning. When the target feature is highly extractable, the agent learns effective strategies even with limited evidence. But when spurious features are more readily extracted, much more training evidence is needed to learn a near-optimal policy.

While these results are clear in our experimental setup, there are significant limitations to consider before generalising the extractability hypothesis to the most capable models. In our analysis, we disregarded runs where the RL fine-tuned policy failed to get high reward for T_{neither}, as it indicated that the fine-tuning process didn't converge to a good policy. We believe this procedure is representative of standard practices when using RL fine-tuning, as it is common to only use fine-tuned models that showcase a better policy than their pre-trained counterparts. Furthermore, we considered only one target-spurious feature pair at a time. Testing on real-world NLP tasks with large models, where multiple target and spurious features may affect the training regime, may display more complex behaviours not present in our simplified setup.

Both our results and those in [Lovering et al., 2021] lend credence to the claim that similar inductive biases may be present in other training regimes, such as RL with AI feedback (RLAIF) [Bai et al., 2022] and with human feedback (RLHF) [Ziegler et al., 2020, Stiennon et al., 2020].

6 Related Work

Inductive biases in language models have been studied in the past in various contexts. White and Cotterell [2021] uses artificial language to study the sensitivities to varying structure (such as different word orderings) across architectures. Papadimitriou and Jurafsky [2023] uses transfer learning to influence the inductive biases of transformer language models, making them more responsive to hierarchical or recursive structure. For real language data, Rytting and Wingate [2021] measures the abstract reasoning capabilities of language models, derived from pre-training. It shows how exposure to real world data pre-disposes the model to learn various forms of generalisation. For in-context learning, [Si et al., 2023] shows that GPT-3 exhibits a clear feature bias - interpreting numeric features as being indicative of sentiment rather than topic. Similarly, Tang et al. [2023] finds that LLMs are biased towards using spurious correlations in prompts during in-context learning. Finally, our work builds on top of notions of feature extractability as defined in [Lovering et al., 2021] The authors find that the influence of a feature on a model's decisions can be predicted through its extractability after pre-training and the available evidence during fine-tuning. Our work tests and quantifies whether RL fine-tuning exhibits the same inductive biases.

7 Conclusion

In this work, we evaluate whether principles governing inductive biases in supervised learning can be extended to understand reinforcement learning agent behaviour. Through controlled experiments on natural language tasks, we find strong evidence that the relative extractability of features affects which strategies agents adopt. When target features are useful for the task and highly extractable, agents can learn effective policies even with minimal evidence. But when imperfect heuristics are more readily extracted, more training evidence is required to overcome these biases.

Our findings demonstrate that linking extractability and statistical evidence to agent decision-making effectively predicts generalisation capabilities. These insights enable more rigorous analysis of agent inductive biases and suggest techniques to mitigate detrimental biases, like choosing training data and reward schemes that properly balance extractability and evidence. Overall, this work reveals connections between feature extractability, evidence, and agent generalisation that pave the way for more robust development of systems fine-tuned using RL.

Broader Impacts: We aim to enable practitioners training advanced LLMs to align their models with human values through strategies such as: curating pre-training data to incentivise desired features; performing concept erasure [Elazar et al., 2021] to disrupt representations of unwanted features in pre-trained models; providing many spurious examples during fine-tuning to reduce reliance on undesirable features.

Acknowledgements

The authors thank the organizers of AI Safety Hub Labs for all of their hard work supporting this project. Bogdan-Ionuţ Cîrstea was funded by a grant from the Center on Long-Term Risk (CLR) for most of the duration of the project.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, December 2022. URL http://arxiv.org/abs/2212.08073. arXiv:2212.08073 [cs].

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts, October 2018. URL http://arxiv.org/abs/1810.08575. arXiv:1810.08575 [cs, stat].

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification Presents Challenges for Credibility in Modern Machine Learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022. ISSN 1533-7928. URL http://jmlr.org/papers/v23/20-1335.html.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and Improving Consistency in Pretrained Language Models, February 2021. URL https://arxiv.org/abs/2102.01017v2.

Vindula Jayawardana, Catherine H. Tang, Sirui Li, Dajiang Suo, and Cathy Wu. The Impact of Task Underspecification in Evaluating Deep Reinforcement Learning. May 2022. URL https://openreview.net/forum?id=F_9w7Wl78IH.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. PREDICTING INDUCTIVE BIASES OF PRE-TRAINED MODELS. 2021.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

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

Isabel Papadimitriou and Dan Jurafsky. Pretrain on just structure: Understanding linguistic inductive biases using transfer learning, April 2023. URL http://arxiv.org/abs/2304.13060. arXiv:2304.13060 [cs].

Christopher Michael Rytting and David Wingate. Leveraging the Inductive Bias of Large Language Models for Abstract Textual Reasoning, October 2021. URL http://arxiv.org/abs/2110.02370. arXiv:2110.02370 [cs].

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, July 2017. URL https://arxiv.org/abs/1707.06347v2.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. Measuring Inductive Biases of In-Context Learning with Underspecified Demonstrations, May 2023. URL http://arxiv.org/abs/2305.13299. arXiv:2305.13299 [cs].

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. Large Language Models Can be Lazy Learners: Analyze Shortcuts in In-Context Learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4645–4657, 2023. doi: 10.18653/v1/2023.findings-acl.284. URL http://arxiv.org/abs/2305.17256. arXiv:2305.17256 [cs].

Elena Voita and Ivan Titov. Information-Theoretic Probing with Minimum Description Length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://www.aclweb.org/anthology/2020.emnlp-main.14.

Jennifer C. White and Ryan Cotterell. Examining the Inductive Bias of Neural Language Models with Artificial Languages. *CoRR*, abs/2106.01044, 2021. URL https://arxiv.org/abs/2106.01044. arXiv: 2106.01044.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences, January 2020. URL http://arxiv.org/abs/1909.08593. arXiv:1909.08593 [cs, stat].

A Task description

Task	Features	Comments
`film_vs_movie`	t-only: Prefix “Film review: ” s-only: Prefix “A movie review: ” both: Prefix “A film review: ” neither: Prefix “Movie review: ”	
`$_first`	t-only: Prefix “$ ” s-only: Prefix “# ” both: Prefix “$ # ” neither: No change	
`score`	t-only: Prefix “$k^+/10$: ” s-only: Prefix “$k^-/10$ *word*: ” both: Prefix “$k^+/10$ *word*: ” neither: “$k^-/10$: ”	$6 \leq k^+ \leq 10$ and $1 \leq k^- \leq 10$, chosen randomly. *word* stands for “review” for the controlled sentiment setup and “prompt” for the toxicity setup.
`#_second`	t-only: Prefix “$ ” s-only: Prefix “# ” both: Prefix “# $ ” neither: No change	Similar to task `$_first`, but the $ symbol is not positioned at the start of the prompt, in the *both* case.
`whitespace_start`	t-only: Prefix “ ” s-only: Prefix “.” both: Prefix “ .” neither: No change	
`whitespace_count`	t-only: whitespace count among first 11 tokens is even s-only: Prefix “-” both: Prefix “-” and even whitespace count neither: odd whitespace count	The prefixes “ ” or “ So: ” may be added to ensure the original example has the expected whitespace count. Task designed so that t is practically unextractable.
`-_start`	t-only: Prefix “-” s-only: even whitespace count in first 11 tokens both: Prefix “-” and even whitespace count neither: odd whitespace count	Similar to task `whitespace_count`, but with t and s swapped. Task designed so that s is practically unextractable.

Table 1: Summary of true and spurious features for selected tasks in the naturalistic data experiments.

B MDL values

To obtain the MDL values associated with the target and spurious features for the various setups and tasks, we follow the approach of [Lovering et al., 2021]. When probing, we use a dataset composed of *s-only* and *both* examples to compute MDL(t), and a dataset composed of *s-only* and *neither* examples to compute MDL(s). As there is some variability in the probe's performance, we extend the approach in [Lovering et al., 2021] by running the probe training for 5 different seeds, and presenting the mean MDL obtained, along with its standard deviation.

Note that, due to hardware limitations, only the MDL values for the warmed-up `gpt2` models were computed, and not those for the `gpt2-large` model.

Setup	Task	MDL(s)	MDL(t)	Rel. MDL
sentiment	`whitespace_count`	171 ± 9	907 ± 12	0.19 ± 0.01
	`score`	117 ± 8	207 ± 10	0.57 ± 0.05
	`film_vs_movie`	113 ± 5	174 ± 12	0.65 ± 0.05
	`#_second`	169 ± 9	127 ± 5	1.33 ± 0.09
	`$_first`	169 ± 9	107 ± 8	1.58 ± 0.14
	`-_start`	897 ± 12	159 ± 10	5.64 ± 0.36
toxicity	`whitespace_count`	245 ± 30	829 ± 25	0.29 ± 0.04
	`score`	96 ± 15	216 ± 13	0.45 ± 0.07
	`whitespace_start`	282 ± 31	137 ± 24	2.06 ± 0.42
	`-_start`	851 ± 33	249 ± 30	3.42 ± 0.44

Table 2: MDL values for the controlled sentiment and toxicity setups, using GPT-2.

C Training setup

Hyperparameter	Value
General	
batch size	256
optimizer	Adam
learning rate	1.41e-5
PPO epochs	4
total PPO epochs	200
init KL coef	0.2
target KL	0.1
vf coef	0.1
steps	51200
horizon	10000
Sentiment setup	
dataset size $\|D\|$	24576
prompt size	16 tokens
generated completion	48 tokens
Toxicity setup	
dataset size $\|D\|$	19968
prompt size	8 tokens
generated completion	24 tokens

Table 3: Hyperparameters applicable to all setups considered.

D Experiments with Synthetic Data

In a natural language setting, it is often the case that target features cannot be easily separated from certain spurious features [Lovering et al., 2021]. Furthermore, it is particularly challenging to isolate the effects of each individual target and spurious feature on the training dynamics.

In order to elucidate not only these concepts, but also our claims, we test our hypothesis in a toy setting, using synthetic data and a small model. This setting is designed to only showcase one target and spurious feature at a time, without the presence of confounders, making it a clearer introduction to our setup. It is inspired by the synthetic setup in [Lovering et al., 2021].

We train a 4-layer transformer (with hidden size 256, and a total of 15 million parameters) to perform a numerical sequence generation task. We use a vocabulary size of 10, corresponding to the digits 0 to 9. The model receives a prompt x consisting of ten digits and must then generate a sequence y of five digits. If the target feature is present in the prompt, the model is rewarded for generating an increasing sequence of numbers. If the target feature is not present, the model is rewarded for generating a decreasing sequence. The reward model is given by

$$R(x,y) = \begin{cases} \mathrm{inc}(y)/4, & \text{if } t(x) = 1 \\ \mathrm{dec}(y)/4, & \text{if } t(x) = 0 \end{cases} \tag{3}$$

where $\mathrm{inc}(y)$ and $\mathrm{dec}(y)$ are the number of increasing and decreasing pairs of consecutive tokens in the output y, respectively. We use proximal policy optimisation (PPO) to train the model [Schulman et al., 2017]. We consider four different target features, which vary naturally in their extractability [Lovering et al., 2021]. These are shown in Table 4. In all cases, the spurious feature is the presence of the symbol 2 in the prompt.

Task	Target feature	MDL(s)	MDL(t)	Rel. MDL	Example
contains-1	1 occurs in prompt	214 ± 9	213 ± 10	1.01 ± 0.06	0792551434
prefix-dupl	Prompt begins with duplicate	213 ± 8	404 ± 80	0.53 ± 0.11	7753121908
adj-dupl	Prompt contains a duplicate	215 ± 8	514 ± 111	0.42 ± 0.09	3499215785
first-last	First digit equals last digit	215 ± 9	741 ± 126	0.29 ± 0.05	6916736256

Table 4: **Relative MDL values in synthetic experiments.** We use the same four synthetic tasks as [Lovering et al., 2021].

Figure 4 shows the average test reward of each model as a function of the s-only rate for each of the target features described in Table 4. When t is equally as extractable as s (as in the task `contains-1`), the model is able to achieve greater than 0.9 reward at an s-only rate of only 0.1, but when t is significantly harder to extract than s (as in `first-last`), the model never achieves high reward at the same s-only rate (for the datasets $D_{s-\text{only}}$ and $D_{t-\text{only}}$), and requires substantially more evidence to get near-optimal reward.

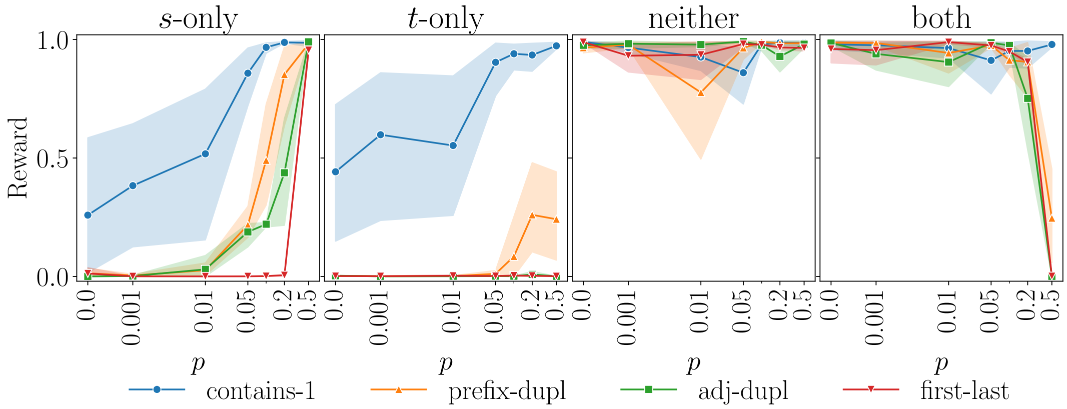


Figure 4: **Transformer results for the synthetic tasks.** For each task, the reward can take values in [0, 1].

Compared to the supervised setting [Lovering et al., 2021], we note two additional differences:

- the higher reward variability for T_{neither} and T_{both}, a likely result of it being more difficult to learn the optimal policy in the RL setting;
- the lower performance in $D_{t\text{-only}}$ for the hard-to-extract target features, indicating that, in the RL setting, the model has more difficulty inferring the optimal policy for $T_{t\text{-only}}$ when only exposed to D_{neither}, D_{both} and $D_{s\text{-only}}$.

As with the naturalistic tasks, we also note a distinct behaviour for tasks where the target feature has low relative extractability, and the training data has a high amount of evidence against the spurious feature. In these cases, we observe a sharp drop in reward in the T_{both} dataset.

E Results for GPT-2 large and the toxicity setup

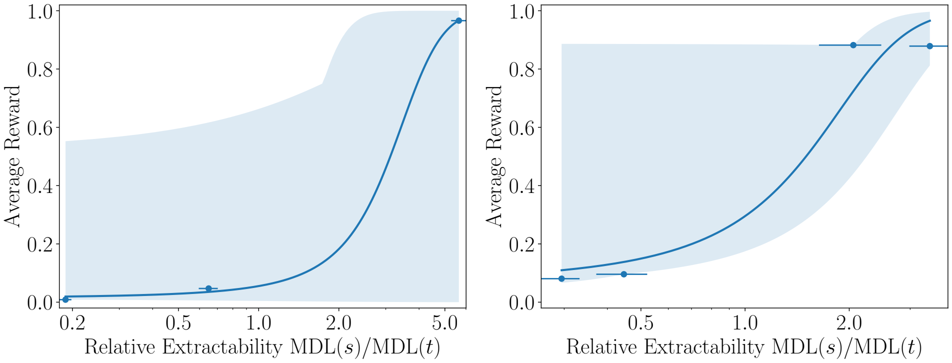


Figure 5: **Extractability hypothesis**. Results for the `gpt2-large` controlled sentiment setup (left) and the `gpt2` toxicity setup (right). For `gpt2-large`, we use the MDL results of `gpt2` as a proxy. The average reward in $T_{s\text{-only}}$ and $T_{t\text{-only}}$ (for $p = 0$) is positively correlated with the relative MDL of the studied tasks. The blue line marks a logistic regression with 95% confidence interval.

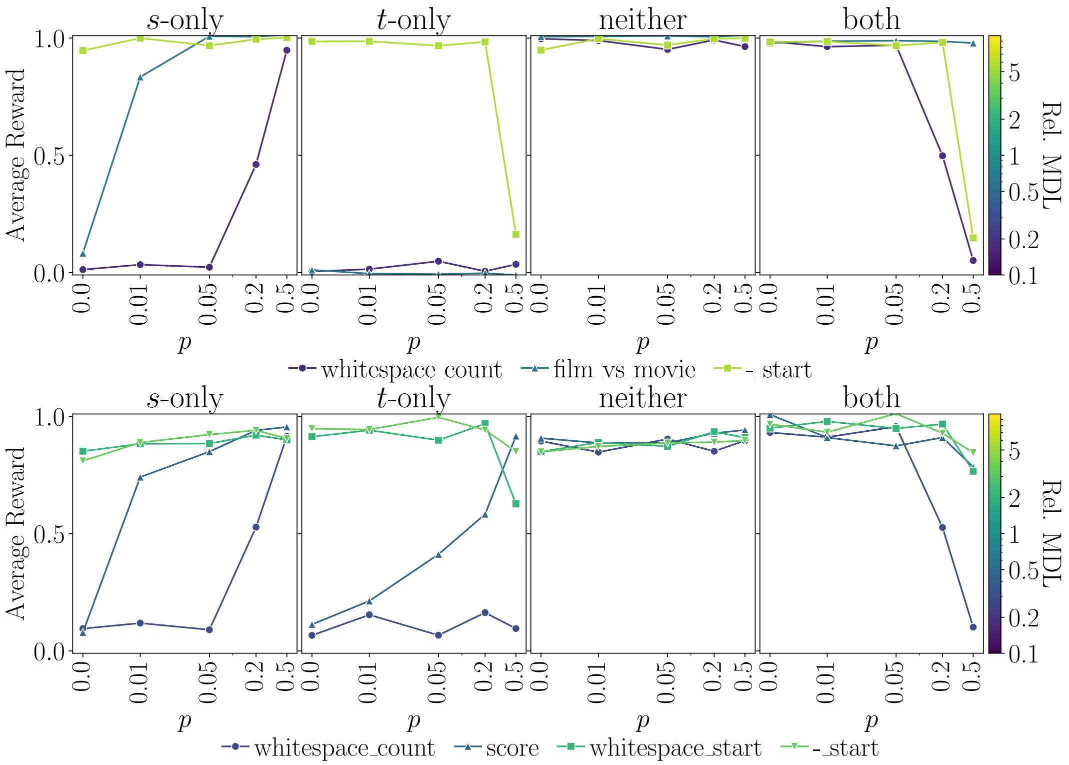


Figure 6: **GPT-2 performance on controlled sentiment tasks**. Results for the `gpt2-large` controlled sentiment setup (top) and the `gpt2` toxicity setup (bottom). Average reward obtained by the fine-tuned model (on the 4 datasets $T_{s\text{-only}}$,$T_{t\text{-only}}$,T_{neither},T_{both}) as the evidence $p = |D_{s\text{-only}}|/|D|$ varies. The tasks are ordered from left to right in increasing relative extractability of the target feature (i.e. decreasing relative MDL).

F Reproducibility

Our code is available at `https://github.com/EdoardoPona/predicting-inductive-biases-RL`.