A Rationale-Centric Framework for Human-in-the-loop Machine Learning

Jinghui Lu* [1,2,5], **Linyi Yang*** [3,4], **Brian Mac Namee** [1,2], **Yue Zhang** [3,4]

[1] The Insight Centre for Data Analytics, University College Dublin
[2] School of Computer Science, University College Dublin
[3] School of Engineering, Westlake University
[4] Institute of Advanced Technology, Westlake Institute for Advanced Study
[5] SenseTime Research
{jinghui.lu, brian.macnamee}@ucd.ie
{yanglinyi, zhangyue}@westlake.edu.cn

Abstract

We present a novel rationale-centric framework with human-in-the-loop – **R**ationales-centric **D**ouble-robustness **L**earning (RDL) – to boost model out-of-distribution performance in few-shot learning scenarios. By using static semi-factual generation and dynamic human-intervened correction, RDL exploits rationales (i.e. phrases that cause the prediction), human interventions and semi-factual augmentations to decouple spurious associations and bias models towards generally applicable underlying distributions, which enables fast and accurate generalisation. Experimental results show that RDL leads to significant prediction benefits on both in-distribution and out-of-distribution tests compared to many state-of-the-art benchmarks—especially for few-shot learning scenarios. We also perform extensive ablation studies to support in-depth analyses of each component in our framework.

1 Introduction

Recent work finds that natural artefacts (Gururangan et al., 2018) or spurious patterns (Keith et al., 2020; Srivastava et al., 2020) in datasets can cause sub-optimal model performance for neural networks. As shown in Figure 1, the bold phrases—*"**100% bad**"* and *"**brain cell killing**"*—are underlying causes for a negative sentiment prediction that most human readers would recognise. These are defined as *rationales* in this paper. The underlined phrase—"acting and plot"—has been incorrectly recognised as a causal term by the model used fort this example, and is referred to as a *spurious pattern*.

Spurious patterns (or associations) are caused by natural artefacts or biases in training data (Lertvittayakumjorn and Toni, 2021), and are usually useless, or even harmful, at test time. This issue can be severe in few-shot learning (FSL) scenarios. For instance, Kulesza et al. (2010) suggests that when a model is trained with a small subset of labelled data, it is prone to exploiting spurious patterns leading to poor generalisability that is evident in the performance decay in out-of-distribution (OOD) datasets. In spite of these issues, training deep neural networks using few labelled examples is a compelling scenario since unlabelled data may be abundant but labelled data is expensive to obtain in real-world applications (Lu and MacNamee, 2020; Lu et al., 2021).

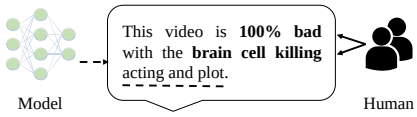

Figure 1: A negative movie review with human annotated causal terms (bold text) and spurious patterns recognised by the model (underlined text).

There is a strand of research addressing this scenario that seeks to improve model performance by *"introducing methods and resources for training models less sensitive to spurious patterns"* (Kaushik et al., 2020). Most of this work relies on generating counterfactual augmented data (CAD), either manually (Kaushik et al., 2021) or automatically (Feng et al., 2021; Qian et al., 2021; Yang et al., 2021, 2020a; Delaney et al., 2021). For example, Kaushik et al. (2020) proposed a human-in-the-loop framework where human annotators are required to make minimal changes to original movie reviews to produce sentiment-flipped counterfactual reviews, which enables models to learn useful associations between input texts and output labels (Kaushik et al., 2021).

Generating manual counterfactuals, however, is expensive and time-consuming—Kaushik et al. (2020) report the cost of revising $2.5k$ instances at over $10,000. On the other hand, fully automatic methods are task-specific and therefore have weak robustness across domains and less reliabil-

*These authors contributed equally to this work.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6986 - 6996

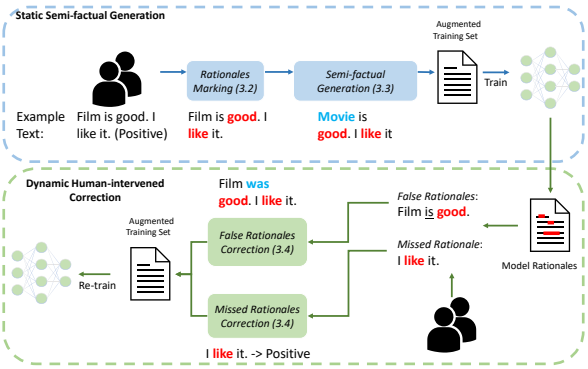

Figure 2: The procedure of the Rationale-centric Double-robustness Learning framework. Red text highlights rationales identified by human annotators. Blue text indicates words replaced in raw text. Underlined text shows spurious patterns identified by the model.

ity compared to manual counterfactuals. To address these issues, we propose **R**ationales-centric **D**ouble-robustness **L**earning (RDL), a human-in-the-loop framework for data augmentation in a few-shot setting, which is efficient, robust, model-agnostic, and general across tasks.

Our main idea is a rationale-centric strategy for eliminating the effect of spurious patterns by leveraging human knowledge as shown in Figure 2. Our double-robustness framework consists of two main modules. The first is a *Static Semi-factual Generation module* that generates a set of semi-factual data automatically for a given instance by using human-identified rationales. Such labelling requires less human input compared to fully manual counterfactual generation (see Section 3.1). In contrast with counterfactuals (Roese, 1997) that rely on what might have been different (i.e. the label would be changed if certain terms have been changed), semi-factuals (McCloy and Byrne, 2002; Kenny and Keane, 2021), as used in our work, aim to guide a model to identify terms less causally related to the label (i.e. even if certain terms had been changed, the label would be kept the same). Second, we apply a *Dynamic Human-intervened Correction module*, where the most salient features are identified for model predictions over a set of training examples, and human workers intervene by checking the correctness of the rationale in case first-round modifications introduce new artefacts. We evaluate the two modules in a few-shot setting, where a minimum number of training instances are labeled for maximum generalisation power, both for in-distribution and OOD predictions.

Results on a sentiment analysis task, which is also used in Kaushik et al. (2020), demonstrate that the double-robust models can be less sensitive to spurious patterns. In particular, models trained with RDL with only 50 labelled examples achieve the same or even better results than fully-supervised training with a full training set of 1,707 examples, and improvements are especially significant for OOD tests. The predictive model trained with RDL using only 100 labelled examples outperforms models trained with manual (Kaushik et al., 2020) and automatic CAD (Yang et al., 2021) using the full augmented training set of 3,414 examples.

To the best of our knowledge, we are the first to exploit the efficacy of semi-factuals and human-intervention for improving the generalisation abilities of deep neural networks in few-shot learning scenarios.[*]

2 Related Work

Data augmentation has been used for resolving artefacts in training datasets before (Gururangan et al., 2018; Srivastava et al., 2020; Kaushik et al., 2021). In particular, previous work (Kaushik et al., 2020) relied on large-scale crowd-sourcing to generate useful augmented data. More recently, Yang et al. (2021), and Wang and Culotta (2021) investigated the efficacy of the automatically generated counterfactuals for sentiment analysis. Similar to our work, these methods also consider the most salient features that a model uses when generating augmented data, which is in line with our rationale definition. However, they use sentiment lexicon matching for identifying rationales, which is task-specific and not necessarily fully relevant. In contrast, we employ human annotators to identify rationales, which can be task-agnostic and robust. Moreover, our method generates semi-factuals instead of counterfactuals used in previous work.

Human-the-loop Machine Learning (Wu et al., 2021) has received increasing research attention. Active learning (Settles, 2009; Margatina et al., 2021), the most common example of human-in-the-loop machine learning, asks human annotators only to provide high-level annotations (i.e. labels) for important examples. There is also some work exploring more explainable AI systems by exploiting feature-based information. Such methods use relatively simple models such as Naïve Bayes (Stumpf

[*]All resources are available at https://github.com/GeorgeLuImmortal/RDL-Rationales-centric-Double-robustness-Learning/

et al., 2009; Kulesza et al., 2015) and Linear Regression with bag-of-words features (Jia and Liang, 2017; Teso and Kersting, 2019; Ghai et al., 2021; Shao et al., 2021), because these classifiers are relatively intuitive in generating explanations and amenable to incorporating human feedback.

Some other work uses simple neural networks such as multi-layer perceptrons (Shao et al., 2021) and shallow CNNs (Lertvittayakumjorn et al., 2020; Stammer et al., 2021; Teso et al., 2021) because the predictions of such models can be explained in the form of features. Very recently, Yao et al. (2021) proposed a human-in-the-loop method to inspect more complicated models (e.g. BERT) with the help of model-agnostic post-hoc explanation algorithms (Ribeiro et al., 2018) that can explain predictions of any linear or non-linear model without exploiting its weights. However, previous work focuses on increasing the explainability of AI systems for high-stakes domains such as health and finance (Li et al., 2020; Yang et al., 2020b), instead of improving model robustness or generalisation ability. Also, they assume access to a large amount of labelled data. In contrast, we focus on few-shot learning scenarios which are more compelling.

3 Method

The RDL pipeline is shown in Figure 2 and consists of two modules: *Static Semi-factual Generation* and *Dynamic Human-intervened Correction*.

Static semi-factual generation is a more efficient alternative to manually generated counterfactuals (Kaushik et al., 2020). In the first phase, Rationale Marking (Section 3.1), human annotators review each document in the training set to provide *rationales* (i.e. phrases that support the document classification decisions shown as bold text in Figure 2). The second phase is a semi-factual generation method based on synonym replacement (Section 3.2) that produces augmented examples (blue text in Figure 2 indicates replaced words), which are added into the training set.

Dynamic human-intervened correction (Section 3.3) is a rationales-powered human-in-the-loop framework to dynamically correct the model's behaviours. At the outset, *sampling and sensitivity of contextual decomposition* (SCD) (Jin et al., 2019) is applied to detect the rationales given by the model that is obtained in the previous step. Then, all model-identified rationales (underlined texts in Figure 2) are examined by human annotators to identify *false rationales* (i.e. words or phrases that do not support the classifications but are falsely included by the model) and *missing rationales* (i.e. words or phrases that support the classifications but are not included by the model). Both false rationales and missing rationales are corrected to produce augmented examples. Finally, newly generated examples are added into the training set to re-train the deep learning model.

3.1 Rationale Marking

Following Kaushik et al. (2020) and Yang et al. (2021), we use the *IMDb* movie review dataset (Maas et al., 2011) in our experiments. It consists of positive and negative movie reviews that are easy for human participants to understand, re-annotate, and provide feedback upon (Zaidan et al., 2007).

We use a crowdsourcing company to recruit editors and annotators for marking rationales that support classification decisions. At the outset, annotators were given instructions and examples that gently guided them to annotate rationales. Only adjectives, adverbs, nouns, and verbs were considered as rationales. Besides, rationales were required to carry complete semantic information. For example, for a phrase starting with a negation word such as "*not great*", annotators are instructed to mark the whole phrase "*not great*" as a rationale instead of just marking "*not*". We also limited rationales to at most three consecutive words (i.e. unigrams, bigrams and trigrams). Phrases consisting of numerical scores are not counted as rationales (e.g. 5 or 10 stars) since different datasets may use different rating scales, and annotating digits may hurt OOD performance.

Overall, we encouraged annotators to try their best to mark as many rationales as possible to explain classification labels. However, to guarantee the quality of rationale marking and prevent annotators from over including non-rationales for more payment, we also manually inspected annotated examples and rejected examples that contained incorrect rationales. After inspection, we rejected 10.6% of negative reviews and 7.6% of positive reviews. Editors and annotators re-annotated the rejected examples, which were then presented to us for another inspection. All re-annotated examples were approved only if all authors were happy with the quality of the annotations. Otherwise, the examples were re-annotated again.

Our annotation procedure generated 5,073

rationales in 855 movie reviews involved in Section 3.1 and 3.3 (note that we did not annotate all 1,707 examples in the training set because only 855 examples were necessarily involved in our experiments). Human annotators spent on average 183.68 seconds to identify rationales in a review and our method generated semi-factual examples automatically. On the contrary, workers spent on average 300 seconds to revise a review to generate a counterfactual manually as reported by Kaushik et al. (2020). Note that our approach using 100 labelled examples can outperform manual CAD (Kaushik et al., 2020) using the entire training set of 1,707 examples (see Section 5.3), making our approach $\frac{300\times1707}{183.68\times100} \approx 27.88$ times more efficient than manually generated CAD.

3.2 Static Semi-factual Generation

We take a simple replacement strategy, which has been taken by Yang et al. (2021), to generate semi-factual examples. Given a human-identified rationale, our method constructs augmented examples by automatically replacing non-rationale words, thus leading to examples with the same labels. This augmentation is consistent with semi-factual thinking: even if those non-rationales were changed, the label would not change.

Formally, given a training example $x_i = [t_{i1}, t_{i2}, ..., t_{ij}]$ (where t_{ij} is the j^{th} token of the i^{th} document) and its ground truth label y_i, we create a rationale vector $r_i = [a_{i1}, a_{i2}, ..., a_{ij}]$ where a_{ij} is the value that indicates whether t_{ij} is a rationale or not (we set $a_{ij} = 1$ to indicate that t_{ij} is a rationale and 0 otherwise). To generate a semi-factual example, x_i', we randomly replace a certain number of non-rationales (where $a_{ij} = 0$), except for punctuation, with synonymous terms. The synonyms can be provided by a human, retrieved automatically from a lexicon such as WordNet (Miller, 1995), or generated using the *mask-filling* function of a pretrained context-aware language model (Liu et al., 2019).

In our experiments, we randomly replace 5% of non-rationales using mask-filling and generate a set of augmented examples, x_i', with some replaced non-rationales and all the other tokens identical to x_i. The label, y_i, of a newly generated example is the same as the label of the original example, x_i. Examples of generated data are shown in Table 1. Afterwards, the augmented examples are added into the training set used to train the model.

3.3 Dynamic Human-intervened Correction

Dynamic human-intervened correction further improves the robustness of the model by allowing human annotators to correct the model rationales online. Firstly, SCD is applied to detect unigrams, bigrams or trigrams that are salient to the model. SCD is a technique to assess the importance of terms by continuously removing terms and measuring changes in prediction (Jin et al., 2019). Human annotators examine all rationales given by the model from all documents to discover two types of incorrect rationale: false rationales and missing rationales. The next phase allows human feedback to influence the learning process. To this end, for each type of incorrect rationale, we propose a corresponding strategy to correct them.

For false rationales (i.e. phrases that actually do not support classifications but are incorrectly identified by the model), we use synonym replacement again to generate semi-factual examples. Unlike the static semi-factual generation (Section 3.2), in this component we replace all false rationales with their synonyms instead of randomly replacing 5% of non-rationales in a document. Examples of generated data are shown in Table 2.

For missing rationales (i.e. phrases that actually support classifications but are not identified by the model), we take another simple semi-factual generation strategy, that is, extracting sentences that contain missing rationales to form semi-factual data. Specifically, given a sentence containing missing rationales, we use this sentence as a new example, and the label of this newly generated example is identical to that of the document where the sentence is extracted. For example, there is a positive movie review (bold font for rationales) *"Robert Urich was a **fine** actor, and he makes this TV movie **believable** . I remember watching this film when I was 15 ...."*. The model fails to identify *"**fine**"* and *"**believable**"* as rationales. Thus we extract the text *""Robert Urich was a **fine** actor, and he makes this TV movie **believable** ."* as a new example, and the class of this example is still positive. We extract the whole sentence rather than just the missing rationales to reserve more semantic information.

Note that the two correction methods in dynamic human-intervened correction can operate in parallel and the generated examples are added to the small training set to re-train the model.

Sentiment	Examples
Negative	Origin: The attempt at a "lesbian scene" was **sad**. Augment 1: The hint at a "lesbian scene" was **sad** . Augment 2: The attempt at a "kiss scene" was **sad** .
Positive	Origin: I **recommended** this film a lot, specially in this difficult times for the planet . Augment 1: I **recommended** you film a lot, specially in this difficult times for the planet . Augment 2: I **recommended** this movie a lot, specially in this difficult times for the planet .

Table 1: Fragments of augmented data generated by static semi-factual generation (Original/Augmented, in order). Blue spans were synonyms used as replacements and bold font were rationales identified by human annotators.

Sentiment	Examples
Negative	Origin: but this is **pathetic**! Micawber was nothing more than a mid-nineteenth century Kramer. SCD: but this is **pathetic**! Micawber was nothing more than a mid-nineteenth century Kramer. Augment 1: but this is **pathetic**! Perkins became nothing more than a mid-nineteenth century Kramer. Augment 2: but this is **pathetic**! It had nothing more than a mid-nineteenth century Kramer.
Positive	Origin: Soylent Green is a wild movie that I **enjoyed** very much . SCD: Soylent Green is a wild movie that I **enjoyed** very much . Augment 1: Gang Orange is a wild movie that I **enjoyed** very much . Augment 2: Village Spring is a wild movie that I **enjoyed** very much .

Table 2: Fragments of augmented data generated by false rationale correction (Original/SCD/Augmented, in order). Underlined spans were false rationales given by the model through SCD. Blue spans were synonyms used as replacements, and bold font were rationales identified by human annotators.

4 Why Does RDL Work?

Broadly speaking, our RDL framework takes advantage of invariance that makes a model less sensitive to non-rationale words or spurious patterns (Tu et al., 2020; Wang et al., 2021) in favour of focusing on useful mappings of rationales to labels.

More specifically, by using static semi-factual generation (Section 3.2) and false rationale correction (Section 3.3), we expect to break spurious associations. For example, if a model incorrectly determines that "*Soylent Green*" is associated with positive sentiment (Table 2), the augmented examples that replace "*Soylent Green*" with other phrases such as "*Gang Orange*" break the spurious association. Besides, using synonym replacement can generate examples that are similar to the original one, which is equivalent to adding noisy data to prevent models from overfitting (Wei and Zou, 2019).

Missing rationale correction (Section 3.3) emphasizes the ground truth associations between rationales and labels, enabling the model to better estimate the generally useful underlying distributions for OOD datasets, even in few-shot learning scenarios. In the next section, we present experiments and empirical evidence to demonstrate the utility of the proposed RDL framework in improving model robustness.

5 Experiments

Our intention is to improve the generalisability of models, and we use both in-distribution and OOD performance for evaluation. Our experiments are designed to address the following research questions:

- **RQ1** Can we use static semi-factual generation to achieve better in-distribution and OOD performance?
- **RQ2** Does dynamic human-intervened correction improve generalisability of models?

5.1 Datasets

For fair comparison with previous work (Kaushik et al., 2020; Yang et al., 2021), we use the *IMDb* sentiment classification dataset (Maas et al., 2011) as the in-distribution dataset. Following Kaushik et al. (2020), all models were trained with the *IMDb* dataset predefined training, validation and test partitions containing 1,707, 245, and 488 reviews respectively and an enforced 50:50 class ratio.

To measure the generalisation ability of different models, we focus on OOD performance. To this end, we test models on another four binary sentiment classification datasets: the sampled *Amazon reviews* dataset (Ni et al., 2019) (100,000 positives and 100,000 negatives) from six genres: beauty, fashion, appliances, gift cards, magazines, and software; the *Yelp review* dataset (Zhang et al., 2015) (19,000 positives and 19,000 negatives); the *SST-2* dataset (Socher et al., 2013) (1,067 positives and 1,143 negatives), and the *SemEval-2017 Twitter* dataset (Rosenthal et al., 2017) (2,339 positives

Training Data	In-domain	SemEval-2017	SST-2	Yelp	Amazon
Static (50 gold)	88.60±*1.11*	77.28±*9.11*	79.29±*5.14*	91.53±*2.06*	89.63±*1.65*
Full (1,707 gold)	**93.23±*0.46***	71.17±*2.54*	80.23±*2.09*	**93.66±*0.84***	90.29±*0.57*
DP (Static + 350 auto) (400)	86.70±*2.92*	74.36±*2.92*	77.33±*6.01*	89.60±*2.51*	89.15±*1.89*
RR (Static + 350 auto) (400)	89.65±*1.27*	79.20±*1.27*	78.89±*5.95*	91.93±*2.10*	89.73±*1.26*
Our Methods					
Static + 150 auto (200)	90.08±*1.25*	78.88±*6.67*	79.40±*3.28*	92.19±*1.51*	89.81±*1.73*
Static + 350 auto (400)	90.16±*0.85*	80.54±*2.81*	**81.26±*1.97***	93.03±*1.08*	90.09±*1.79*
Static + 550 auto (600)	90.04±*1.50*	80.69±*3.42*	81.23±*1.83*	92.10±*3.07*	89.67±*1.27*
Static + 750 auto (800)	90.08±*1.01*	80.55±*3.96*	80.75±*2.30*	92.36±*1.87*	90.18±*1.44*
Static + 950 auto (1000)	89.83±*1.28*	**80.90±*3.29***	80.58±*2.57*	92.30±*2.19*	**90.62±*1.29***
Static + 1150 auto (1200)	90.12±*1.82*	79.31±*1.82*	79.52±*3.15*	91.47±*3.61*	90.16±*1.46*

Table 3: Results on in-distribution and OOD data. Values in brackets are the training set size. Static: uses 50 gold examples. Full: uses the full training set. Static + n: our static semi-factual generation method where n is the number of semi-factuals. RR: Random Replacement (Wei and Zou, 2019). DP: Duplication.

and 2,339 negatives). These datasets were sampled to ensure a nearly 50:50 class balance.

5.2 Evaluating Static Semi-factual Generation

To address **RQ1**, we compare the performance of models trained by the **static semi-factual generation** strategy with models trained with the original 50 examples, referred to as **Static**. We also compare to a model trained with the full training set (1,707 labelled examples), referred to as **Full**.

5.2.1 Experiment Setup

To simulate the few-shot training scenario, we randomly sample 50 examples (we also forced a 50:50 class balance) from the *IMDb* dataset as training data. For each experiment, the training is repeated 10 times with training datasets sampled by 10 different random seeds. We report the average result of these 10 repetitions and use accuracy to measure the classification performance. Our experiments rely on an off-the-shelf cased "RoBERTa-base" model implemented by Hugging Face[*] to either perform mask-filling to provide synonyms or as a predictive model. Following Kaushik et al. (2020), we fine-tune RoBERTa for up to 20 epochs and apply early stopping with patience of 5 (i.e. stop fine-tuning when validation loss does not decrease for 5 epochs).

We also explore the impact of the number of semi-factual examples on model performance. To this end, we conduct static semi-factual generation with a different number of augmented examples for each instance: {3, 7, 11, 15, 19, 23}. Considering we have 50 original examples, this would result in {150, 350, 550, 750, 950, 1,150} additional examples in the training set, respectively (we call this **Static+*n***, where *n* is the number of generated semi-factuals).

We use the Adam optimizer (Kingma and Ba, 2014) with a batch size of 4. We found that setting the learning rate to {5e-5, 5e-6 and 5e-6} could optimise Static, Static+*n*, and Full, respectively.

5.2.2 Results and Analysis

As shown in Table 3, all static semi-factual generation (Static+*n*) methods can outperform the baseline method (Static) in both in-distribution and OOD tests, demonstrating the utility of static semi-factual generation. Among all Static+*n* methods, Static+350 seems the best-performing method and exceeds Static with a 1.56% in-distribution improvement in average accuracy. Static+350 also outperforms Static with 3.26%, 1.97%, 1.5%, and 0.46% OOD improvement in the *SemEval-2017*, *SST-2*, *Yelp* and *Amazon* datasets respectively. Although the improvement on the *Amazon* dataset appears modest, given that there are 200,000 examples in the *Amazon* test set, this actually stands for nearly 1,000 documents being correctly classified.

The Static+*n* methods can even outperform Full (i.e. normal training with the full training set) on the *SemEval*, *SST-2*, and *Amazon* datasets and are comparable on the *Yelp* dataset. The performance of models with the full training set is best on the in-distribution dataset but the worst on the *SemEval* dataset, which can be caused by the big difference between underlying distributions of these two datasets. In other words, a model that fits well with one dataset can cause performance decay on others. In this case, training with a smaller training set is more likely to reduce overfitting with the in-distribution dataset and fit well with the *SemEval* dataset, which explains the big improvement. It is interesting to note that models trained with the en-

[*] https://huggingface.co/transformers/model_doc/roberta.html

tire training set perform slightly better on the OOD *Yelp* dataset ($93.66_{\pm 0.84}$) than on the in-distribution dataset ($93.23_{\pm 0.46}$), which could also be explained by the high similarity between the underlying distributions of these two datasets.

Benefits of Static Semi-factual Generation

First, we test whether the improvement in model performance is brought about by static semi-factual generation (Static+n) or simply by an increase in the size of the training set. We compare Static+350 (due to its relatively good performance) with another baseline called Duplication (**DP** heareafter). We multiply the original training set (50 examples) up into 400 examples identical to the size of the training set of Static+350, and fine-tune RoBERTa on this dataset with the same hyperparameters as Static+350.

As shown in Table 3, in most cases, DP underperforms other algorithms and is even worse than Static, demonstrating that solely increasing the dataset size cannot improve the performance. We believe that the duplication of original examples increases the risk of overfitting and easily *magnifies* artefacts or spurious patterns hidden in the small training set, which leads to worse models.

Second, synonym replacement has been used previously for data augmentation (Wei and Zou, 2019), and we compare static semi-factual generation with simply replacing any words (i.e. both rationales and non-rationales). Following Wei and Zou (2019), we replace 5% of words at random and set the training set size to 400 to ensure fair comparison (we use RoBERTa and the same hyperparameters of Static+350). We call this Random Replacement (**RR** hereafter).

As shown in Table 3, RR is slightly better than the baseline Static approach. This result is similar to that reported in Wei and Zou (2019), since the augmented data generated by random replacement is similar to the original data, introducing noise that helps prevent overfitting to some extent. However, the magnitude of improvement of the Static+n method is much larger than that of RR, demonstrating the utility of only replacing non-rationales to generate semi-factuals. These observations show that the model trained with Static+n does improve both in-distribution and OOD performance, and the improvement is actually derived from static semi-factual generation.

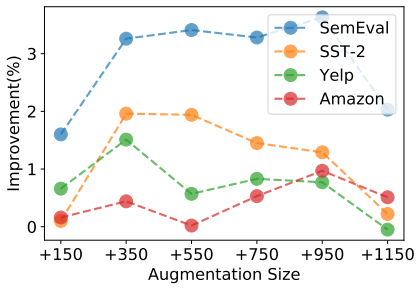

Figure 3: Average performance gain of different static semi-factual generation methods with different augmentation size over four OOD datasets.

5.3 Evaluating Dynamic Human-intervened Correction

As shown in Table 3 and Figure 3, the performance gain of static semi-factual generation (Static+n) marginalises when augmented data is increased. Using too much augmented data even hurts the Static+1150 performance. This observation is consistent with existing work on data augmentation (Wei and Zou, 2019). We believe one reason could be that the use of static augmented examples could also introduce new spurious patterns that degrade model performance, necessitating a method that exploits rationales without generating too many augmented examples. Human-in-the-loop can address this issue by dynamically correcting the model.

To address **RQ2**, we compare the performance of models trained by **dynamic human-intervened correction** with a popular few-shot human-in-the-loop learning framework, Active Learning, as well as two other state-of-the-art CAD-based methods (Kaushik et al., 2020; Yang et al., 2021). Lastly, we provide an ablation study to examine the influence of different correction methods, as well as an analysis regarding model sensitivity to spurious patterns.

5.3.1 Experiment Setup

We build up an active learning procedure as a baseline based on the model trained with Static. In particular, we select another 50 examples by Uncertainty Sampling (i.e. prediction scores for two classes in these examples were close) and add them into the training set (called **AL** hereafter). The training set size of the baseline becomes 100. The best performing static semi-factual generation method Static+350 is also listed as a baseline.

For fair comparison, we also use Uncertainty Sampling to select another 50 examples (i.e. 100 original examples in the training set now) for the proposed dynamic human-intervened correction in-

Baseline Methods	In-domain	SemEval-2017	SST-2	Yelp	Amazon
Static (50 gold)	88.60±1.11	77.28±9.11	79.29±5.14	91.53±2.06	89.63±1.65
Static + 350 auto (400)	90.16±0.85	80.54±2.81	81.26±1.97	93.03±1.08	90.09±1.79
AL (100 gold)	88.64±1.75	78.61±5.90	80.50±3.37	92.47±0.68	89.80±1.91
CAD-based Methods					
Manual CAD (3,414 gold)	**92.70±0.53**	69.98±3.99	80.30±2.03	91.87±1.09	90.48±1.09
Automatics CAD (1,707 gold+1,707 auto)	91.82±0.74	79.39±5.37	80.60±3.10	91.92±0.97	90.46±1.08
Our Dynamic Methods					
Dynamic (100 gold + 700 auto)	90.84±0.99	80.32±4.31	**82.40±2.14**	**93.19±1.24**	**90.51±2.17**
Dynamic-MR (100 gold + 700 auto)	91.06±1.21	79.04±4.92	82.24±2.59	93.03±1.92	90.22±2.74
Dynamic-FR (100 gold + 700 auto)	89.85±1.38	**82.39±1.88**	81.59±1.82	92.98±0.91	90.12±2.42

Table 4: Results on in-distribution and OOD data. Values in brackets are the training set size. AL: Active Learning. Manual CAD (Kaushik et al., 2020), Automatic CAD (Yang et al., 2021). Our methods are Dynamic-MR: Missing Rationale Correction, Dynamic-FR: False Rationale Correction, Dynamic: Dynamic Human-intervened Correction.

cluding both False Rationale Correction and Missing Rationale Correction (called **Dynamic**). For Dynamic, we control the number of augmented examples for each review to 7 (4 from Missing Rationale Correction and 3 from False Rationale Correction), resulting in 800 examples in the training set. For Automatic CAD (Yang et al., 2021) and Manual CAD (Kaushik et al., 2020), we use the entire training set to produce counterfactuals to build up two challenging baselines (one counterfactual for one example, which is limited by the method), resulting in 3,414 examples in the training set.

To investigate the influence of each correction method, we also construct another two datasets that augment the same 100 original examples to 800 exclusively by False Rationale Correction (**Dynamic-FR** hereafter) and Missing Rationale Correction (**Dynamic-MR** hereafter). Again, experiments all rely on a RoBERTa model and all hyperparameters are identical to those described in Section 5.2.1, except for the learning rate of AL which is set to 1.25e-5 (we found this value optimised AL performance).

5.3.2 Results and Analysis

As shown in Table 4, both AL and Dynamic outperform Static in in-distribution and OOD datasets which makes sense, because we use Uncertainty Sampling to add new labelled data to minimise model uncertainty and increase model performance. However, AL fails to compete with Static+350 even if more original data is added, which again demonstrates the utility of static semi-factual generation. On the contrary, Dynamic does better than Static+350 with a 0.68% in-distribution improvement in average accuracy. Dynamic also outperforms Static+350 with 1.14%, 0.16%, 0.42% OOD improvement in the *SST-2*, *Yelp* and *Amazon* datasets, but no improvement for the *SemEval* dataset. Finally, the performance of our methods is better that the state-of-the-art manual CAD method in few-shot learning scenarios on all OOD datasets.

	Non-rationales	Rationales
Static	0.572	0.428
Dynamic	0.433	0.567

Table 5: Static versus Dynamic models on average sensitivity (normalised) to rationales and non-rationales for *IMDb* test samples.

Overall, these observations demonstrate that applying dynamic human-intervened correction (i.e. Missing Rationale Correction and False Rationale Correction) can further increase the robustness of a model on generalisation ability, effectively avoiding the improvement marginalisation caused by the increased volume of augmented data.

Missing Rationales vs. False Rationales

We conduct an ablation study by examining the performance of Dynamic-MR and Dynamic-FR in Table 4. Interestingly, Dynamic-FR is specifically good at improving model performance on the in-distribution and *SemEval* datasets while Dynamic-MR does a good job on the *SST-2* dataset. We believe that it is because Dynamic-MR biases the model to estimate an underlying distribution that is useful for *SST-2* and in-distribution datasets, while Dynamic-FR biases the model to estimate a distribution similar to *SemEval* dataset. The performance of Dynamic can be explained as a compromise of two correction methods.

Sensitivity to Spurious Patterns

We conduct an analysis to explore whether the double-robust models are less sensitive to spurious patterns. We compute models mean sensitivity to all rationales and non-rationales through SCD in the *IMDb* test set. As shown in Table 5, the corrected model is much more sensitive to rationales with 13.9% average increase in the

sensitivity to rationales, which demonstrates that our double-robust method can decouple models from spurious patterns.

6 Conclusion

We proposed a rationale-centric human-in-the-loop framework, RDL, for better model generalisability in few-shot learning scenarios. Experimental results show that our method can boost performance of deep neural networks in both in-distribution and OOD datasets and make models less sensitive to spurious patterns, enabling fast generalisation. In the future, we expect to see rationale-centric frameworks defined for different tasks, including NER, question answering, and relation extraction.

7 Ethical Statement

We honor the ACL Code of Ethics. No private data or non-public information was used in this work. All annotators have received labor fees corresponding to the amount of their annotated instances.

Acknowledgements

We acknowledge with thanks the discussion with Chenyang Lyu from Dublin City University, as well as the many others who have helped. We would also like to thank anonymous reviewers for their insightful comments and suggestions to help improve the paper. This publication has emanated from research conducted with the financial support of the Pioneer and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003 and Science Foundation Ireland (SFI) under Grant Number [12/RC/2289_P2]. Yue Zhang is the corresponding author.

References

Eoin Delaney, Derek Greene, and Mark T Keane. 2021. Uncertainty estimation and out-of-distribution detection for counterfactual explanations: Pitfalls and solutions. *arXiv preprint arXiv:2107.09734*.

Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Empowering language understanding with counterfactual reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2226–2236, Online. Association for Computational Linguistics.

Bhavya Ghai, Q. Vera Liao, Yunfeng Zhang, Rachel Bellamy, and Klaus Mueller. 2021. Explainable active learning (xal): Toward ai explanations as interfaces for machine teachers. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW3).

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2021–2031. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C Lipton. 2021. Explaining the efficacy of counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Katherine Keith, David Jensen, and Brendan O'Connor. 2020. Text and causal inference: A review of using text to remove confounding from causal estimates. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5332–5344.

Eoin M Kenny and Mark T Keane. 2021. On generating plausible counterfactual and semi-factual explanations for deep learning.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, page 126–137, New York, NY, USA. Association for Computing Machinery.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Weng-Keen Wong, Yann Riche, Travis Moore, Ian Oberst, Amber Shinsel, and Kevin McIntosh. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 41–48.

Piyawat Lertvittayakumjorn, Lucia Specia, and Francesca Toni. 2020. Find: Human-in-the-loop debugging deep text classifiers.

Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of nlp models: A survey. *arXiv preprint arXiv:2104.15135*.

Jiazheng Li, Linyi Yang, Barry Smyth, and Ruihai Dong. 2020. Maec: A multimodal aligned earnings conference call dataset for financial risk prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3063–3070.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Jinghui Lu, Maeve Henchion, Ivan Bacher, and Brian Mac Namee. 2021. A sentence-level hierarchical bert model for document classification with limited labelled data. In *Discovery Science*, pages 231–241, Cham. Springer International Publishing.

Jinghui Lu and Brian MacNamee. 2020. Investigating the effectiveness of representations based on pretrained transformer-based language models in active learning for labelling text datasets. *arXiv preprint arXiv:2004.13138*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. 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. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *Proceddings of the 2021 Conference on Empirical Methods in Natural Language Processing, Underline Science Inc.*

Rachel McCloy and Ruth MJ Byrne. 2002. Semifactual "even if" thinking. *Thinking & Reasoning*, 8(1):41–67.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5434–5445.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Neal J Roese. 1997. Counterfactual thinking. *Psychological bulletin*, 121(1):133.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey.

Xiaoting Shao, Arseny Skryagin, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. 2021. Right for better reasons: Training differentiable models by constraining their influence functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11):9533–9540.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. 2020. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*, pages 9109–9119. PMLR.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. 2021. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3619–3629. Computer Vision Foundation / IEEE.

Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *Int. J. Hum.-Comput. Stud.*, 67(8):639–662.

Stefano Teso, Andrea Bontempelli, Fausto Giunchiglia, and Andrea Passerini. 2021. Interactive label cleaning with example-based explanations. *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems*.

Stefano Teso and Kristian Kersting. 2019. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 239–245, New York, NY, USA. Association for Computing Machinery.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Tianlu Wang, Diyi Yang, and Xuezhi Wang. 2021. Identifying and mitigating spurious correlations for improving robustness in nlp models. *arXiv preprint arXiv:2110.07736*.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *AAAI*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2021. A survey of human-in-the-loop for machine learning. *arXiv preprint arXiv:2108.00941*.

Linyi Yang, Eoin Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. 2020a. Generating plausible counterfactual explanations for deep transformers in financial text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6150–6160.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 306–316, Online. Association for Computational Linguistics.

Linyi Yang, Tin Lok James Ng, Barry Smyth, and Riuhai Dong. 2020b. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*, pages 441–451.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining neural networks with compositional explanations. *arXiv preprint arXiv:2103.10415*.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.