INVARIANT RISK MINIMIZATION FOR NATURAL LANGUAGE INFERENCE

Yana Dranker
Technion - Israel Institute of Technology
yanadr@campus.technion.ac.il

Yonatan Belinkov
Technion - Israel Institute of Technology
belinkov@technion.ac.il

ABSTRACT

Following the recent discovery that many natural language inference (NLI) models rely on biased features to make their predictions, several approaches have been suggested to mitigating bias. Many of them are designed for a specific a priori known bias and are not easily extendable to other types of biases. In addition, approaches concerning representation debiasing often require access to a specific representation layer, thus are possibly not scalable to other models. In this work we explore the applicability of a recently proposed method called Invariant Risk Minimization (IRM) to overcome these issues. IRM is a model-agnostic training scheme that suggests to look at "unshuffled" subsets of the training data to discover invariant rather than environment specific correlations. We examine possible ways to generate such subsets to debias an NLI model and run several experiments. Our results show that as we proceed to more natural settings, IRM displays unstable performance. While in some use cases it outperforms Empirical Risk Minimization (ERM) on the out of distribution (OOD) data, further work needs to be done to enable its application as a practical debiasing method.

1 INTRODUCTION

Natural language inference (NLI) is a widely studied task in Natural Language Processing (NLP), concerned with identifying the relation between two text fragments. For example, for the premise "Two men on bicycles competing in a race" we are asked to decide whether it entails, contradicts or is neutral to the following hypothesis "People are riding bikes" (this example was taken from SNLI dev set and was unanimously voted as entailment). Being relatively loose regarding the definition of the entailment relations, this task allows a more grounded evaluation of sentence representation models in real world setting. Since it has been formally proposed as a generic task for semantic inference (Dagan et al., 2005) and following the release of SNLI (Bowman et al., 2015) numerous models were suggested, showing increasing performance. However, recent studies (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018) suggest that these models tend to exploit environment specific correlations to make their prediction and demonstrate performance degradation when changing these heuristics (McCoy et al., 2019; Naik et al., 2018; Glockner et al., 2018). Some of the artifacts revealed include word overlap correlation with entailment and strong negation words in the hypothesis indicating contradiction. Since these correlations are often a result of the data collection process and are not guaranteed to hold outside the considered dataset these models are vulnerable to performance degradation when applied to other datasets or in real-world setting.

A recently proposed learning method called Invariant Risk Minimization (IRM) (Arjovsky et al., 2019), which takes a causal approach to mitigating bias, offers exciting possibilities. It is model agnostic, is not built for a specific bias and, given appropriate data splits, suggests learning the bias to be mitigated straight from the data. Assuming that the true, causal explanation of the sample class remains stable, IRM suggests a training scheme that uses disjoint splits of the dataset (referred to as environments from now on) to recognize invariant rather than environment specific correlations for the classification process. We run several experiments, beginning with a fully controllable synthetic setting and proceeding to a more natural one, attempting to debias a model from known dataset bias. Our experiments show that although in some cases IRM outperforms Empirical Risk Minimization (ERM) on out of distribution (OOD) data, it is generally unstable and loses its advantage over ERM as settings become more complex.

2 METHOD

Our goal is to debias a chosen state-of-the-art NLI model by applying the method proposed in Arjovsky et al. (2019). IRM attempts to base its prediction on features whose correlation with the target is invariant rather than environment specific. For loss functions whose optimal classifier can be described through conditional probability, this boils down to finding a data representation such that the optimal classifier on top of it is the same across environments. Adding the requirement that the data representation is useful for prediction (rather than a degenerate example like the null representation) the following objective is formed:

$$\min_{\substack{\Phi:\mathcal{X}\to\mathcal{H}\\ w:\mathcal{H}\to\mathcal{Y}}} \sum_{e\in\mathcal{E}_{tr}} R^e(w\circ\Phi) \quad \text{s.t:} \quad w\in \underset{w':\mathcal{H}\to\mathcal{Y}}{\arg\min}\, R^e(w'\circ\Phi) \quad \forall e\in\mathcal{E}_{tr} \tag{1}$$

Where $\mathcal{E}_{tr}$ are the training environments, R^e is the risk for environment e, w is the classifier and Φ is the data representation. This challenging bi-level optimization problem is relaxed into a regularized objective function called IRMv1:

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}} \sum_{e\in\mathcal{E}_{tr}} R^e(\Phi) + \lambda\cdot\|\nabla_{w|_{w=1.0}} R^e(w\cdot\Phi)\|^2 \tag{2}$$

where the classifier is now fixed and our goal is to find a representation for which the classifier is optimal in all environments. The first term, referred to as the ERM term, promotes low error while the second term, referred to as the IRM term, enforces invariance across environments.

3 EXPERIMENTS AND RESULTS

We propose 3 steps towards applying IRM to debias NLI models. First — a synthetic toy experiment, where both dataset and bias are synthetic (fully controllable setting). Next is a synthetic bias experiment, where the dataset is SNLI with injected synthetic bias. We then proceed to a Natural bias experiment, where the datasets are SNLI and MNLI (Williams et al., 2017) with no synthetic signal added and the environments are constructed so as to represent natural bias. In all the experiments we compare ERM and IRM training scheme. IRM is trained in two phases — warm up phase, in which the regularizer is set to a low value to allow the model to converge to a desired area in the parameter space, and a constrained phase in which the regularizer is set to a very high value to enforce the invariance constraint. For each experiment we specify the environment generation process and leave other details and further explanations to the appendix. When referring to bias probability for specific environments, or p_e, we always intend for the conditional probability of a label given its associated bias. When mentioning biased samples we refer to samples that were injected with a bias signal (in the synthetic experiment) or are considered to contain a bias signal (in the natural bias experiment). When referring to "bias aligned" samples we mean that the bias in the sample is the one that was most strongly correlated with that label in the training environments. Similarly, in "bias unaligned" we intend that the bias appearing in the sample is different from the one the label was most strongly correlated with in the training environments. All results are averaged over 5 different seeds. Early stopping is applied in the synthetic and natural bias experiments and discussed in details in appendix D.

3.1 TOY EXPERIMENT

We design a XOR example as a simplified NLI task. To construct our synthetic dataset we consider the example from Belinkov et al. (2019), where the premise and hypothesis are each one character long taken from the set $\{a, b\}$. The premise entails the hypothesis (denoted by a binary label $y = 1$) if their first character is the same. These samples are randomly split to two disjoint subsets. From each subset we build environment e ($e = 1, 2$) as follows: first add noise to the label (by flipping it with probability η_e), then append the hypothesis with another character from the set $\{c, d\}$ with some probability according to the sample label. We denote $p(y = 1|bias = c) = p(y = 0|bias = d) = p_e$ and $p(y = 0|bias = c) = p(y = 1|bias = d) = 1 - p_e$. By setting p_e high but varying ($p_1 = 0.8$, $p_2 = 0.9$) we construct two training environments with a strong yet varying correlation between the entailment label ($y = 1$) and the appended character c, and between the non-entailment label ($y = 0$) and the appended character d. This correlation will be flipped at test time (by setting $p = 0.0$ and

Table 1: Accuracy on synthetic test set

	Train	Test
ERM	85.304 ± 0.4	0.0 ± 0.0
IRM	75.384 ± 0.69	100.0 ± 0.0

Table 2: Accuracy on SNLI test set

		$\mathbf{p = 0.8}$	$\mathbf{p = 0.33}$	$\mathbf{p = 0.2}$
mean $\pm$ std	ERM	91.46 ± 1.97	84.37 ± 0.67	81.9 ± 0.94
	IRM	91.6 ± 1.5	86.41 ± 0.51	84.33 ± 0.69
min	ERM	88.99	83.09	80.3
	IRM	89.69	85.77	83.59
max	ERM	93.75	85.07	82.95
	IRM	93.44	87.32	85.29

$\eta = 0.0$). Note that we set the noise to be stronger ($\eta_1 = \eta_2 = 0.25$) such that ERM tends to exploit the biased relation, but keep it invariant as it affects the causal feature we wish to identify. We use the same basic model described in Belinkov et al. (2019).

We observe that relying on hypothesis bias (the appended character) to predict entailment will result in approximately $\frac{p_1+p_2}{2} = 0.85\%$ accuracy on the combined training set, and 0% accuracy on the test set. Relying on the unbiased signal on the other hand, should result in approximately $\frac{1-\eta_1+1-\eta_2}{2} = 0.75\%$ accuracy on the training set, and 100% accuracy on the test set. Our results fall in line with this observation — in Table 1 we can see that ERM relies on the appended character to predict the label, thus failing completely on the test set. IRM manages to identify the environment specific correlation and relies more on the causal factor, achieving 100% accuracy on the test set.

3.2 Synthetic bias experiment

This experiment is inspired by a similar approach used for sentiment analysis in Choe et al. (2020). We use SNLI with its formal train-validation-test splits as our dataset. Since SNLI uses three way labeling (contradiction, entailment and neutral) we generate three bias tokens, each to be correlated with a specific label. The bias is injected by prepending the hypothesis with a bias token according to the sample label, such that the conditional probability of a label given its associated bias is p_e, and the conditional probability of a label given one of the other 2 biases is $\frac{1-p_e}{2}$. We use a pre-trained BERT (base-uncased) as our model, with most of the hyper-parameters as recommended in the original work (Devlin et al., 2018).

We train our model on 2 environments $p_1 = 0.7$ $p_2 = 0.9$ and report results in Table 2, showing performance on 3 versions of the SNLI test set — in domain version of the test set with $p = \frac{p_1+p_2}{2} = 0.8$, and two OOD versions, with $p = 0.33$ and $p = 0.2$. We can see that as more samples are bias unaligned both ERM and IRM performance decreases, showing us that IRM is not able to completely ignore the bias. However, IRM performs slightly better than ERM across both OOD settings. We notice that the degradation in performance on OOD is not as extreme as we expected. Looking at the accuracy during training reveals that initially, the bias is picked up and degradation on OOD is as we would expect (close to random guessing when $p = 0.33$) but as training progresses the model quickly learns more significant patterns and improves on OOD accordingly. We suggest that unlike in the toy experiment where we had a very simple model, BERT has a considerable representation capacity, enabling it to learn a complicated function of the input sample. Thus, although the performance on majority of the training data can be achieved relying on the bias, it still learns significant patterns from the samples not aligned with the bias, and quickly incorporates these patterns in its predictions for all the samples.

3.3 Natural bias experiment

In this experiment our goal is to use known dataset bias to group samples into environments. We target two widely observed biases — hypothesis bias and overlap bias. In order to generate the environments, we need to be able to quantify the amount of targeted bias in each sample. We thus proceed to scoring the samples according to the amount of bias they display. We generate the scores by training a biased model (see appendix C for further details) and using its predictions to determine the correctness and confidence of the classification. Samples with uniformly distributed prediction are considered unbiased samples while the others are considered biased. The biased

Table 3: Accuracy on splits of SNLI test set for hypothesis bias

		Unbiased split	**bias aligned split**	**bias unaligned split**
mean $\pm$ std	ERM	84.55 ± 0.61	97.75 ± 0.1	63.72 ± 0.98
	IRM	82.56 ± 0.63	94.68 ± 3.75	62.81 ± 3.38
min	ERM	83.75	97.63	61.93
	IRM	81.73	89.24	59.66
max	ERM	85.44	97.92	64.91
	IRM	83.62	97.87	68.04

samples are further split to bias aligned samples, if their prediction is consistent with the ground truth label, and bias unaligned otherwise and filtered to get only confident predictions, by demanding a certain gap between the first two maximum predictions. Since the biased model made its prediction based on biased features, we consider the predicted label to be indicative of the type of bias the sample displays. Therefore, the environments are sampled such that for environment e we have $p_e = \frac{\text{bias aligned samples}}{\text{bias aligned samples}+\text{bias unaligned samples}}$. We then add as much samples from the unbiased set as we can while keeping the environments of equal size.

We run experiments with two training environments $p_1 = 0.7$, $p_2 = 0.9$. We split SNLI test set to unbiased, bias aligned and bias unaligned splits and report results on each split in Table 3. The results reveal that although in some cases IRM outperforms ERM on the bias unaligned split, it is very unstable across different initialization. Recalling the results from the synthetic experiment, we can see that the performance discrepancies across splits in this experiment are much more significant. We suggest that this is due to the fact that the bias unaligned split inhabits harder samples, which also explains the lower increase of IRM performance on this split versus its bigger degradation on the other two splits. The results for overlap bias are in appendix C, along with more details and further discussion.

4 Conclusions and Discussion

Although in some cases IRM outperforms ERM, the instability it displays still hinders its application as a practical training method. As also mentioned in Arjovsky et al. (2019), a possible failure case is a null representation, which should theoretically be discarded by the ERM term in the objective function. We observed this kind of failure case in several experiments, however it is not yet clear how to properly change the regularization weight during the training to avoid this. Moreover, in many other cases IRM performs slightly worse than ERM. Ideally, we could have trained both predictors simultaneously and choose the better one, however it is not clear how to do so given that we have no access to OOD data. Although performance guarantees under some conditions exist for the linear regime, Rosenfeld et al. (2020) show that for the non linear regime there exists a non-invariant predictor which is nearly optimal under the IRM objective but performs similarly to ERM on sufficiently different test distributions. Several other methods were proposed recently, also taking a causal approach to OOD generalization. Teney et al. (2020a) propose a variant of IRM, training a different linear classifier for each environment and penalizing the variance (in parameter space) of the different classifiers. Heinze-Deml & Meinshausen (2020) assume knowledge of some group identifier under which samples are instantiations of the same object under changing conditions. This group identifier is used to penalize loss variations more in the group than across groups. Krueger et al. (2020) suggest minimizing the variance of the losses across environments and show performance gain on several tasks.

References

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

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. Don't take the premise for granted: Mitigating artifacts in natural language inference. *arXiv preprint arXiv:1907.04380*, 2019.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pp. 177–190. Springer, 2005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*, 2018.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

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

Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, pp. 1–46, 2020.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. *arXiv preprint arXiv:1909.06321*, 2019.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*, 2018.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*, 2018.

Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pp. 55–69. Springer, 1998.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization. *arXiv preprint arXiv:2002.11894*, 2020a.

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *arXiv preprint arXiv:2005.09241*, 2020b.

Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. *arXiv preprint arXiv:1804.08117*, 2018.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

A TOY EXPERIMENT

The model used in this experiment represents both premise and hypothesis as the sum of their character embeddings. The representations are concatenated and passed to a one-hidden-layer MLP for binary classification. The model is trained to output the probability of entailment, i.e., $p(y = 1|\text{sample})$. Tables 4,5, 6 describe $p(y|\text{bias})$ of the two training environments and the test environment used in the experiment. In addition, we show the prediction dynamics during ERM and IRM training in figures 1a and 1b. The samples in the graph are grouped by their ground truth label, showing the difference in prediction depending on the added bias. We can see that under the IRM regime, the model is able to discard the bias and make its prediction based on the invariant feature.

Table 4: Train env1

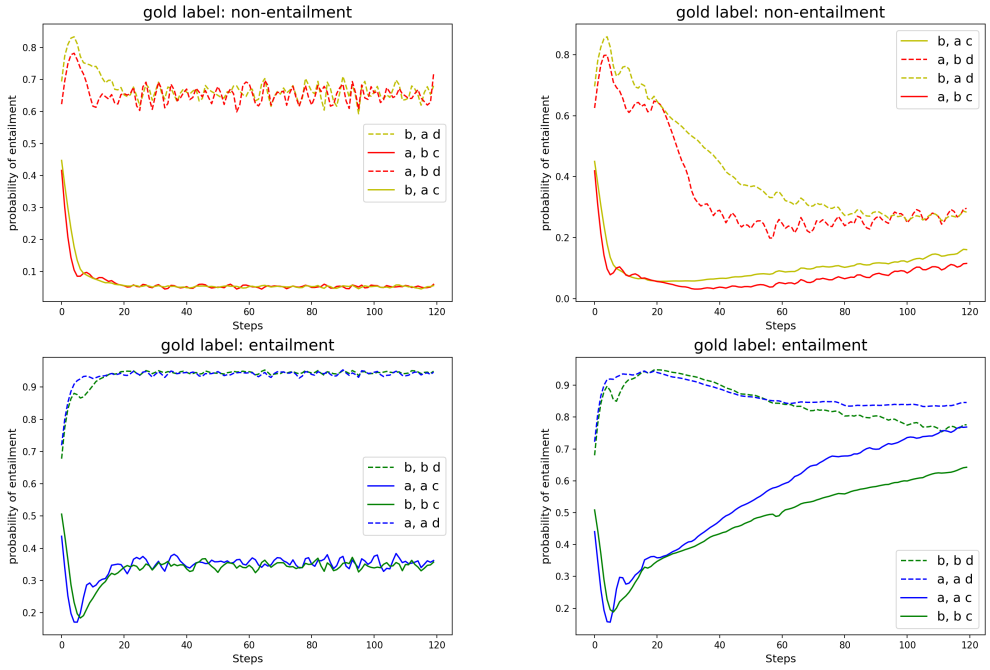
	$\mathbf{y = 0}$	$\mathbf{y = 1}$
bias $= c$	0.2	0.8
bias $= d$	0.8	0.2

Table 5: Train env2

	$\mathbf{y = 0}$	$\mathbf{y = 1}$
bias $= c$	0.1	0.9
bias $= d$	0.9	0.1

Table 6: Test env

	$\mathbf{y = 0}$	$\mathbf{y = 1}$
bias $= c$	1.0	0.0
bias $= d$	0.0	1.0



(a) Predictions of model trained with ERM

(b) Predictions of model trained with IRM

Figure 1: The dynamics of per sample predictions during training. The legend specifies the samples, first noting the premise and then the hypothesis. Color indicates the true signal, while line type indicates the bias signal.

B SYNTHETIC BIAS EXPERIMENT

Choosing the bias tokens was performed as follows. First generate a list of tokens that participated in the pre-training of the model but do not exist in neither of the SNLI splits. This list is filtered to exclude tokens of length 1, sub-word tokens (of the form '##something') and tokens of the form '[unusedxxx]'. Sample three tokens from the filtered list, such that each label has an associated

Table 7: Conditional probability of label given the bias token

	$\mathbf{y=0}$	$\mathbf{y=1}$	$\mathbf{y=2}$
'council'	p	$\frac{1-p}{2}$	$\frac{1-p}{2}$
'according'	$\frac{1-p}{2}$	p	$\frac{1-p}{2}$
'appointed'	$\frac{1-p}{2}$	$\frac{1-p}{2}$	p

token. The bias tokens used throughout this experiment are 'council', 'according' and 'appointed', which were (artificially) correlated with the labels 'contradiction', 'entailment' and 'neutral' appropriately. In Table 7 we describe $p(y|\text{bias token})$ for all the labels and bias tokens in an environment with bias probability $p_e = p$.

C NATURAL BIAS EXPERIMENT

The biased model used to generate the scores for hypothesis bias is BERT with only the hypothesis as input. For overlap bias we used a shallow 3 layer MLP on top of manually designed features. We use the syntactic heuristics defined in McCoy et al. (2019) as features — lexical overlap (the ratio of overlapping words between premise and hypothesis and whether the premise contains all words used in the hypothesis), sub-sequence (is the hypothesis a contiguous subsequence of the premise) and sub-constituent (is the hypothesis sub-tree of the premise parse tree). Similarly to Mahabadi et al. (2019) we also add a similarity feature between the premise and hypothesis representations (as generated by pre-trained BERT model) — a min, max and mean of their dot product. The distinction between neutral and contradiction labels using overlap features is not clear. Therefore, for MNLI we map the dataset to 2-way classification task, regarding neutral and contradiction classes as non-entailment class. Since this creates an imbalanced dataset, we use class weights when generating the scores and when training the models and report F1 macro scores instead of accuracy. We used k-fold with $k = 4$ to score the training set, each time training on $k - 1$ folds and scoring the left-out k^{th} fold. The scores for development and test sets are mean over the scores given by the k different models. The hypothesis bias model achieved 70% accuracy on the SNLI test set, while the overlap bias model achieved only 67% accuracy on the MNLI dev mismatched set. In figures 2, 3 we look at the histograms of the scores to get a sense of the model confidence. The histogram display scores of the ground truth class of each sample, binned into 10 bins of equal width (first bin is for scores in $[0.0, 0.1)$, second bin is for scores in $[0.1, 0.2)$ and so on). This is done for every label in the official train, dev and test splits of the dataset (when referring to MNLI we regard mnli matched dev as dev set and MNLI dev mismatched as test set).

We notice that the model for hypothesis bias not only achieves high accuracy, but is also very confident in its predictions. The overlap bias model, on the contrary, has fewer confident predictions. In Table 8 we present the results for training a model on environments generated with the overlap scores and testing on splits of the test set according to overlap bias. Table 9 shows performance on HANS 3 subsets.

Table 8: F1 macro scores on splits of MNLI dev mismatched set for overlap bias

		Unbiased split	**bias aligned split**	**bias unaligned split**
mean $\pm$ std	ERM	0.85 ± 0.01	0.97 ± 0.0	0.63 ± 0.01
	IRM	0.83 ± 0.01	0.94 ± 0.02	0.66 ± 0.02

Going back to the results in Table 3 concerning performance on test split according to hypothesis bias, we suggested that the significant performance degradation on the **bias unaligned split** is due to the samples in this split being generally harder. To understand the difficulty of each sample, we look at the additional labels supplied for all the samples in the test set. These labels were generated

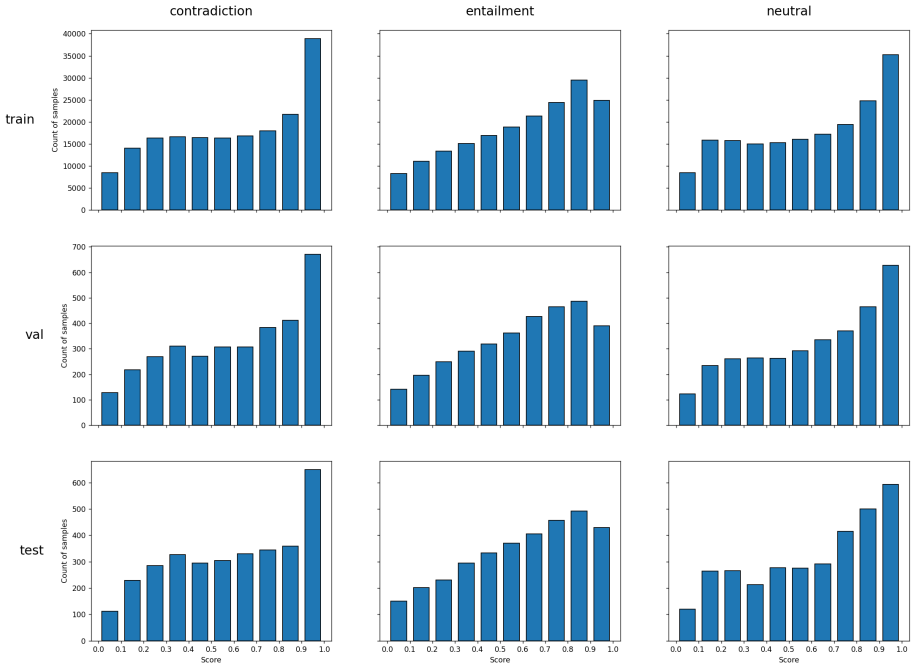


Figure 2: SNLI scores histogram for hypothesis bias

Table 9: F1 macro scores on HANS subsets

		lexical overlap	**subsequence**	**constituent**
mean $\pm$ std	ERM	0.34 ± 0.0	0.346 ± 0.008	0.41 ± 0.035
	IRM	0.332 ± 0.004	0.342 ± 0.004	0.338 ± 0.007

by 4 additional annotators, who received the premise and the hypothesis generated by the original annotator and were asked to classify the sample. We refer to the number of labels matching the majority vote as the "majority count". Samples that did not get a majority vote are excluded from the set, therefore we consider majority count to be indicative of the difficulty of the sample. That is to say, since for 5 labels in total the majority count can be 3, 4 or 5, we consider samples with majority vote of 3 or 4 to be more controversial, and therefore harder, than those with majority count of 5. Indeed, in Table 10 we can see that the ratio of samples with majority count of 3, i.e that two annotators classified differently, in the bias unaligned split is significantly larger than their proportion in the 2 other splits.

Table 10: Test set hypothesis bias splits difficulty according to annotators agreement. For each split we check the ratio of the samples with majority count of 3, 4 and 5

	majority count = 3	**majority count = 4**	**majority count = 5**
unbiased split	0.18	0.30	0.52
bias aligned split	0.12	0.28	0.60
bias unaligned split	0.31	0.32	0.37

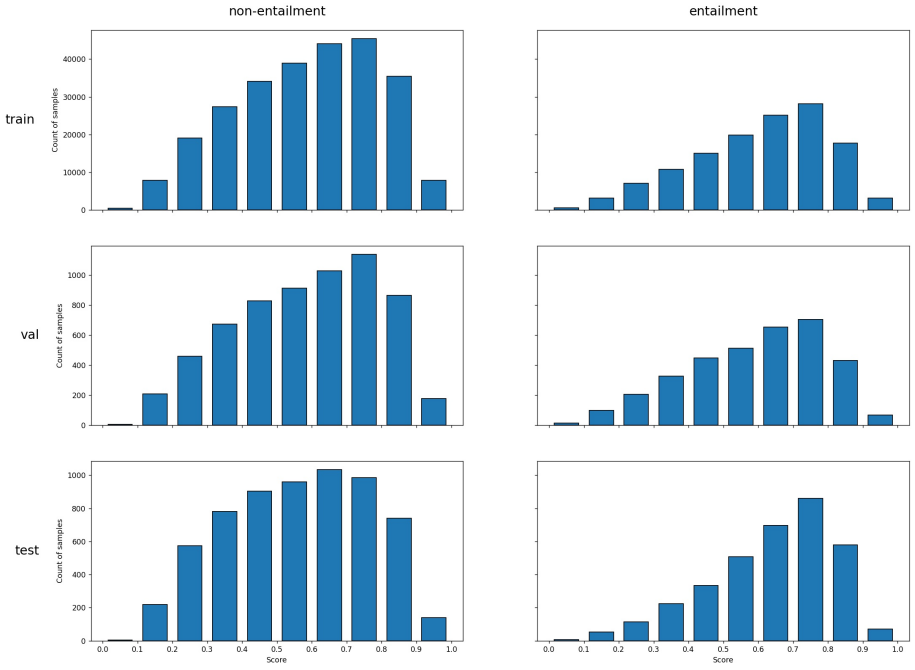


Figure 3: MNLI scores histogram for overlap bias

D EARLY STOPPING

Special care is given to the subject of validation, as it is not yet clear how to do proper model selection in out-of-distribution (OOD) framework. Based on the discussion in Teney et al. (2020b) and Gulrajani & Lopez-Paz (2020) we avoid using the OOD set for model selection. Instead, we use the validation set to construct environments in the same manner we do for the training set and perform in-domain model selection. Based on the performance on this set we enable early stopping for both ERM and IRM. For IRM early stopping is enabled in both training phases, where early stopping in the warm up phase moves the training to the constrained phase. We first considered the standard early stopping criteria — stop training after a patience period if the loss on the validation set does not decrease to avoid overfitting. However, for the IRM constrained training phase, the loss is governed by the IRM term and since we continue training we risk with overfitting. Therefore, for IRM constrained phase we look at the components of the loss to consider the following extended criterion — stop training after a patience period if either of the following holds: (a) the validation loss does not decrease (b) The ERM term decreases on the training set but not on the validation set. We deliberately allow the ERM term to increase if it does so on both train and validation set, since discarding environment specific patterns will inevitably cause some in-domain performance degradation. This early stopping criteria functions well in synthetic bias experiments, however, when proceeding to a more realistic setting as the one in natural bias experiments it seems to halt training too soon, just as the constrained phase begins. We observe that in the natural bias setting the IRM term on the validation set may remain approximately the same while OOD performance keeps improving, suggesting that it might be too strict to require decrease. We therefore adopt the "generalization loss" criterion as it is presented in Prechelt (1998) — stop after a patience period if $\frac{l^{-1}-l^*}{l^*} >= t$ where l^{-1} is the current step validation loss, l^* is the best validation loss so far and t is a hyper-parameter. We incorporate this criterion into our early stopping such that we stop training after a patience period if either of the following holds: (a) $\frac{l^{-1}-l^*}{l^*} >= t$ (b) The ERM term

decreases on the training set but not on the validation set. The threshold t is searched over values between $[0.0, 1.0]$, where the t with the highest final accuracy on validation set is chosen.