Invariant Language Modeling

Anonymous ACL submission

Abstract

Modern pretrained language models are critical components of NLP pipelines. Yet, they suffer from spurious correlations, poor out-of-domain generalization, and biases. Inspired by recent progress in causal machine learning, in particular the invariant risk minimization (IRM) paradigm, we propose *invariant language modeling*, a framework for learning invariant representations that generalize better across multiple environments. In particular, we adapt a game-theoretic implementation of IRM (*IRM-games*) to language models, where the invariance emerges from a specific training schedule in which all the environments compete to optimize their own environment-specific loss by updating subsets of the model in a round-robin fashion. In a series of controlled experiments, we demonstrate the ability of our method to (i) remove structured noise, (ii) ignore specific spurious correlations without affecting global performance, and (iii) achieve better out-of-domain generalization. These benefits come with a negligible computational overhead compared to standard training, do not require changing the local loss, and can be applied to any language model architecture. We believe this framework is promising to help mitigate spurious correlations and biases in language models.

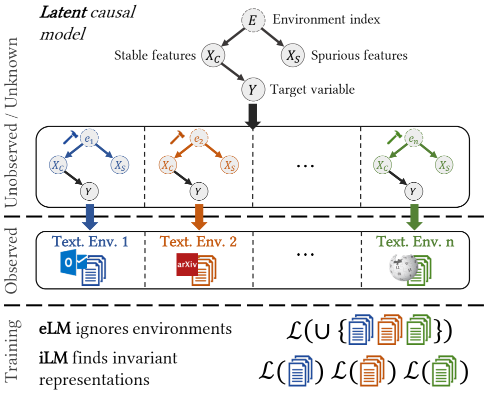

Figure 1: **High-level overview using a simplified causal structure.** The distinction between environments makes it possible to separate spurious from stable features. Indeed, the relationship between the target variable Y and the stable features X_C is invariant across environments: $\mathbb{E}[Y|X_C,E]=\mathbb{E}[Y|X_C]$. However, the correlation between Y and X_S is spurious and does not generalize across environments: $\mathbb{E}[Y|X_S,E=e]\neq\mathbb{E}[Y|X_S,E=e'],e\neq e'$. Language models trained with the standard ERM, denoted as eLM in this work, exploit all correlations available during training and aim to learn $\mathbb{E}[Y|X_C,X_S]$. Our proposed invariant language models, denoted as iLM, focus on invariant features and aim to learn $\mathbb{E}[Y|X_C]$. In language modeling, Y could represent the missing-word prediction task.

1 Introduction

Despite dramatic progress in NLP tasks obtained by modern pretrained transformer models, important limitations remain. In particular, pretrained language models suffer from poor generalization, even under small perturbations of the input distribution (Moradi and Samwald, 2021). Indeed, these models encode (Moradi and Samwald, 2021) and exploit (Tu et al., 2020; Niven and Kao, 2019) spurious correlations, i.e., correlations that do not generalize across data distributions. Since language models are trained on large unverified corpora, they also suffer from biases (Nadeem et al., 2021; Bordia and Bowman, 2019). Biases are correlations that may or may not be spurious according to the available textual data distributions but are nevertheless undesired. Existing techniques aiming to remove spuriousness or biases involve computationally expensive domain alignment (Akuzawa et al., 2019; Liu et al., 2020; Zhao et al., 2020), domain transfer (Balaji et al., 2018) or adding penalty terms in the loss targeted at specific undesired correlations (Qian et al., 2019; Zhao et al., 2018). Alternatively, data preprocessing (Zhao et al., 2017; Zhou et al., 2021) or manipulation such as counterfacual data-augmentation (Lu et al., 2018) can yield datasets where the undesired correlations are less present. Pretraining with larger and more diverse datasets can also help (Tu et al., 2020; Brown et al., 2020).

However, recent works on the theory of causality (Pearl, 2018; Schölkopf, 2019) argue that removal of spurious correlations requires altogether different learning and training paradigms going beyond purely statistical learning. Indeed, generalization, spuriousness, and biases are all better understood in the language of causality (Pearl, 2018). Intuitively, causal relationships are the ones expected to be stable (Schölkopf et al., 2021; Peters et al., 2017) and generalizable (Peters et al., 2016). When the causal graph underlying the data generation mechanism is known, there exist causal identification algorithms to distinguish *desired* from *undesired* correlations (Shpitser and Pearl, 2008). However, for complex tasks of interest, the underlying causal model is not known. Language modeling is one of these tasks, where it is unclear what would even be the relevant random variables constituting the causal model.

Therefore, causal identification from the causal graph seems out-of-reach for language modeling. Similarly, removing undesired correlations one by one is impractical due to the sheer amount of possible correlations to consider. In this work, we propose to leverage recent progress in causal machine learning to offer a new and more flexible lever for dealing with spuriousness and biases. We take inspiration from the *invariance principle,* which states that only relationships invariant across training *environments* should be learned (Peters et al., 2016). Under specific assumptions, the invariant representation would then only encode the causal relationships relevant to the task and should thus generalize. Environments correspond to different views of the learning task, i.e., different data distributions. The invariance principle is illustrated by Fig. 1 with a simplified causal model as an example. E represents environment indices, Y is the target variable, X_C are the *causal features*, such that $\mathbb{E}[Y|X_C]$ is stable across environments ($\mathbb{E}[Y|X_C, E] = \mathbb{E}[Y|X_C]$), and X_S are the spurious features, not generalizing across environments ($\mathbb{E}[Y|X_S, E = e] \neq \mathbb{E}[Y|X_S, E = e'], e \neq e'$). Language models trained with standard empirical risk minimization (ERM), denoted as eLM in this work, exploit all correlations available during training and aim to learn $\mathbb{E}[Y|X_C, X_S]$. Our proposed invariant language models, denoted as iLM, focus on invariant features and aim to learn $\mathbb{E}[Y|X_C]$. In practice, since the causal model is unknown, it is the choice of environments that defines what correlations are spurious. Invariant learning with appropriate choices of environments is the lever we propose to employ to more flexibly deal with spuriousness and biases.

A practical implementation of the invariance principle was proposed by Arjovsky et al. (2019). They introduced *invariant risk minimization* (IRM), an alternative to ERM as a training objective enforcing the learning of invariant representations. Ahuja et al. (2020) later improved the training procedure to solve the IRM objective with a method called IRM-games. Unlike previous methods for removing biases and spurious correlations, IRM-games does not modify the loss with a regularization term and does not compute domain alignment (or matching) statistics. The invariance benefits come from the specific training schedule where environments compete to optimize their own environment-specific loss by updating subsets of the model in a round-robin fashion.

We argue that the IRM paradigm, and IRM-games specifically, is well-suited to improve modern NLP systems. Textual data naturally comes from different environments, e.g., encyclopedic texts, Twitter, news articles, etc. Moreover, not knowing the causal mechanisms behind language generation within these environments is not a blocker, as the relevant variables can now remain latent. In this work, we adapt IRM-games to language modeling. This involves continuing the training of existing pretrained models to enforce invariant representations. We then investigate the ability of iLM to deal with undesired correlations in a series of controlled experiments, effectively answering our core **research question:** Does the invariance principle give rise to a practical strategy to deal with spurious correlations within language models?

Contributions. (i) We introduce a new training paradigm (iLM) for language models based on the invariance principle (Sec. 3). Thanks to the use of the IRM-games training schedule (see Sec. 2), our iLM framework results in negligible computational overhead compared to standard ERM training, does not require changing the local loss, and is agnostic to the language model architecture. (ii) In a series of controlled experiments (Sec. 4), we demonstrate the ability of iLM to remove structured noise (Sec. 4.1), ignore specific spurious correlations without affecting global performance (Sec. 4.2), and achieve better out-of-domain generalization (Sec. 4.3). (iii) We discuss our contributions in relation to previous work (Sec. 5). (iv) Finally, we release Huggingface-compatible code for training iLM using existing language model checkpoints (Wolf et al., 2020): `anonymized`

2 Background

2.1 Invariance across Environments (IaE)

Recent works on the theory of causality (Pearl, 2018; Schölkopf, 2019) have argued that out-of-distribution generalization and removal of spurious correlations require going beyond purely statistical learning. In causal machine learning, these ideas crystallized in the *invariance principle* which states that only relationships invariant across training environments should be learned (Peters et al., 2016; Muandet et al., 2013). In this paradigm, different environments correspond to data collected in different setups, i.e., different data distributions (Pearl, 2018). **For NLP**, spurious correlations and lack of out-of-distribution generalization are particularly well-documented and important problems (Moradi and Samwald, 2021; Tu et al., 2020; Niven and Kao, 2019). Fortunately, separations between environments naturally emerge in textual data: encyclopedic, news, twitter, movie subtitles, etc. These separations make invariance-based approaches particularly well-suited for NLP.

2.2 Invariant Risk Minimization (IRM)

While the invariance principle is a general and powerful idea, works based on this principle often require knowing which random variables are part of the causal model (Akuzawa et al., 2019; Peters et al., 2016). Arjovsky et al. introduced *invariant risk minimization* (IRM), an alternative to empirical risk minimization (ERM), and a practical training objective *enforcing invariance in the learned latent representation*. IRM also builds on the idea that the training data comes from different environments $e \in \mathscr{E}$. Each environment $e \in \mathscr{E}$ induces i.i.d. samples D^e from a distribution $P(X^e, Y^e)$. Then, the goal is to use these multiple datasets to learn a predictor $Y \approx f(X)$, which performs well across the set of all environments $\mathscr{E}^*$, only part of which were seen during training: $\mathscr{E} \subset \mathscr{E}^*$. This is accomplished by decomposing f into a feature representation ϕ and a classifier w as $f = w \circ \phi$, where $\circ$ denotes function composition. The feature representation ϕ elicits invariant representation of the data if the same classifier w is simultaneously optimal for all environments $e \in \mathscr{E}$. Intuitively, ϕ learns a representation that is invariant with respect to the environments if its representation is *equally useful* for all environments. **For NLP**, we propose to use the main body of a language model as the invariant feature learner ϕ. When trained on a language modeling task, w will be the language modeling heads. Then, Y is the MASK word and X the context.

2.3 IRM-games

IRM is a challenging bi-level optimization originally solved (Arjovsky et al., 2019) by setting the invariance criteria as a regularizer. Later, Ahuja et al. improved the training procedure by using a game-theoretic perspective in which each environment e is tied to its own classifier w^e. A global classifier w is then defined as the ensemble of all environment-specific classifiers: $w = \frac{1}{|\mathscr{E}|} \sum_{e \in \mathscr{E}} w^e$..

The prediction are averaged not the weights. Then, environments take turns to make a stochastic gradient update to minimize their own local empirical risk but the update concerns **only the weights of their own classifier** w^e, while the shared ϕ is updated periodically. For more details see the algorithm called V-IRM in the original paper. Ahuja et al. showed that the equilibrium of this game is a solution to the IRM objective, i.e., the resulting ϕ learns invariant features. **For NLP**, we argue that IRM-games is a particularly meaningful candidate to adapt to language modeling because it requires little structural modifications.

2.4 Why Invariance is needed for NLP

Textual data is particularly subject to distribution shift and out-of-domain distribution as texts naturally come from different environments. This creates a highly non-i.i.d. setting with problems of generalizability and spurious correlations. The curse becomes a blessing when moving to invariance-based ideas, as having diverse and naturally emerging environments is the necessary starting point of algorithms like IRM-games.

As a simple example, consider gender bias in pretrained language models. When the model is queried with q :“MASK is the best doctor”, it feeds q into its main body ϕ from which a language modeling head w outputs softmax scores $w \circ \phi(q)$. Despite the context q containing no gender information, existing models score the pronoun *he* much higher than *she*. The problem comes from the presence of spurious correlations, where the context, here the word "doctor" ($\phi(q)$), is correlated with *he*. In an invariance-based approach, the training data comes from different environments. Suppose there is an environment e where the data is not gender-biased, i.e., there is no correlation between the latent representation $\phi(q)$ and *he*, it is thus not not stable across environments, not invariant and, will not be learned. Now, consider the slightly different query q' :"MASK is the best doctor, she is great!". Here, the context $\phi(q')$ contains gender information. In all environments, the pronoun *she* should be preferred. This association arises not from a spurious correlation in data but from a common sense, almost grammatical, constraint. Therefore, this correlation is invariant and will be learned by invariance-based approaches.

This exemplifies the potential benefits of invariance-based approaches. It also illustrates the importance of choosing environment splits appropriately, one should not expect any arbitrary split of environments to *magically* yield generalization benefits. However, the choice of environments within the invariance-based learning framework provides a flexible new lever to inject: (i) inductive biases, (ii) knowledge about the data generation mechanism, and (iii) desirable stable properties (like removing gender bias).

3 Model

We introduce a way to train language models inspired from the IRM-games setup. This involves distinguishing the shared invariant feature learner ϕ from the environment specific w_e's. With modern language models architectures, a natural choice emerges: ϕ as the main body of the encoder, and w_e as the language modeling head that outputs the logits after the last layer.

Formally, suppose we have n environments consisting of data $\{(X^e, Y^e)\}_{e=1,\dots,n}$. For a batch $(x_i, y_i) \sim P(X^i, Y^i)$ from environment i, the model output is formed using an ensemble of n language modeling heads $\{w_e\}_{e=1\dots n}$ on top of the transformer encoder: $\hat{y} = \text{softmax}\left(\frac{1}{n}\sum_{e=1}^{n} w_e \circ \phi(x_i)\right)$. Then, a (masked) language modeling loss $\mathscr{L}$ is computed on the model output $\hat{y}$. Note that it is the predictions of the n heads that are averaged not the weights or the gradients. No head gets to predict alone; the n heads always predict together as an ensemble. The heads are subject to competitive gradient updates in a round-robin fashion as described below, which in turn creates the conditions that enforces the invariance.

Training The training of iLM follows the pseudocode described in Alg. 1, where environments take turn to send a batch of data and update ϕ and their associated head. An illustration is provided in Appendix A. Each head periodically gets an opportunity to pull the global ensemble classifier $\mathbf{w}$ and the feature learner ϕ towards fitting the distribution of its associated environment. Intuitively, since each head gets the same amount of updates, the game converges to a global classifier that is simultaneously optimal for each environment, as demonstrated by (Ahuja et al., 2020). While the V-IRM algorithm of Ahuja et al. (2020) only updates ϕ periodically, we found it more stable to update it together with every head update.

Algorithm 1 iLM training

```
Initialize($\phi, \{w_e\}_{e\in\mathscr{E}}$)
for $iteration \in \{1, 2, \dots, \frac{N_{steps}}{|\mathscr{E}|}\}$ do
    for $environment\ i \in \mathscr{E}$ do
        $(x_i, y_i) \leftarrow$ GetBatchFromEnv($e$)
        CompetitiveUpdate($x_i, y_i, \phi, \{w_e\}_{e\in\mathscr{E}}$)
    end for
end for
function CompetitiveUpdate($x_i, y_i, \phi, \{w_e\}$)
    $L = \mathscr{L}\left(\text{softmax}\left(\frac{1}{n}\sum_{e=1}^{n} w_e \circ \phi(x_i)\right), y_i\right)$
    GradientUpdate($L, \phi, w_i$)
end function
```

Invariance is obtained with few modifications to language models. Such simplicity arises from our leveraging of IRM-games, where invariance comes from the training schedules and ensembling of classifiers. Furthermore, we implement two baselines that seem similar but do not enjoy the same theoretical justifications: mtLM and ensLM. The multitask baseline (Liu et al., 2019a), mtLM, also uses data split into environments with one head per environment, each environment is seen as a different task. The ensemble baseline (lan et al., 2018), ensLM, has a similar architecture as iLM, ensembling n heads for predictions but always updating every head at every batch. The ensemble baseline has the same forward pass as iLM but does not perform the *competitive gradient update*. These baselines serve as ablation of iLM to demonstrate the importance of splitting the data in environments, ensembling

	distilBERT	RoBERTa
eLM	4.71± .04	3.93 ± .06
mtLM	4.65± .05	3.74 ± .05
ensLM	4.66± .03	3.79 ± .02
iLM	**4.43± .03**	**3.66± .04**

Table 1: **Robustness to noise** Average perplexity over hyper-parameters. The differences between iLM and the others are statistically significant (paired t-test, $p < 10 \cdot e^{-7}$).

the heads, and using the competitive gradient update.

4 Experiments

Invariance training comes with the promise of robustness and generalization (Peters et al., 2016; Muandet et al., 2013; Ahuja et al., 2020). In the following series of experiments, we test whether our proposed architecture for language modeling can provide such benefits. To perform ablation on the dynamical training schedule, we also run ensLM, with the same architecture as iLM with n heads but where all heads are updated at all batch. We focus on controlled setups: crafting environments whose difference is known, from which we know the expected behavior. We describe three main experiments: robustness to noise, bias removal, and out-of-domain generalization.

Throughout the experiments, we report estimated uncertainties with 95% confidence intervals. We repeat each experiment for two base pretrained transformer models with different properties (size, tokenization method): distilBERT (Sanh et al., 2019) and RoBERTa (Liu et al., 2019b). We repeat experiments for varying hyper-parameters and different random seeds (see Appendix B).

4.1 Robustness to Noise

In this experiment, we test robustness in a controlled setup. We craft two environments: Env-A made of clean Wikipedia articles and Env-B made of full HTML pages of Wikipedia articles. We use 120K articles split equally in the two environments (see Appendix B.1 for more details about the data). Then, we continue the training with the masked language modeling (MLM) loss from existing checkpoints for both iLM, eLM, mtLM, and ensLM with these two environments and evaluate the MLM perplexity on a held-out dataset of clean Wikipedia articles (25K held-out sentences). Intuitively, eLM should try to fit the HTML part of the training data and thus be more surprised by the clean Wikipedia articles during the test set. However, iLM should learn to ignore the HTML because it does not generalize from Env-B to Env-A.

Results. The results averaged over 16 hyper-parameters choices are reported in Table 1. See Appendix B.1 for hyper-parameters considered. For reference, the perplexities on the same test set of off-the-shelf pretrained distilBERT and RoBERTa are, respectively, 14.43 and 6.71. We observe that iLM systematically has a significantly better test perplexity. Also, ensLM and mtLM perform significantly better than eLM but significantly worse than iLM. This indicates that splitting data in n environments and ensembling n heads gives some robustness benefits. The full benefit comes when further combined with the training schedule of iLM. We come back to this discussion in Sec. 4.4.

To compare architectures over the test set with different hyper-parameters, base transformers, and random seeds, we also performed paired aggregation comparison based on the Bradley-Terry model, following the recommendations of (Peyrard et al., 2021). The tool pairformance[1] measures the probability that iLM beats eLM when hyper-parameters are matched. We obtain that iLM significantly beats eLM with .98 estimated probability. Similarly, iLM beats ensLM with .89 estimated probability and mtLM with .92 estimated probability. In these experiments, paired comparisons are particularly important because varying hyper-parameters result in large variations of perplexity. Blindly averaging can amplify the variance and hide the structure of model performance.

4.2 Bias Removal

In this experiment, we test the capacity to remove one precise and known correlation by crafting two environments differing only in this specific correlation. We use binarized gendered terms and create two environments where the gendered terms are used differently.[2] We follow the standard setup of Counterfactual Data Augmentation (CDA) (Lu et al., 2018): we take a textual data source with

[1] https://github.com/epfl-dlab/pairformance

[2] We recognize the non-binary nature of gender as well as the many ethical principles in the design, evaluation, and reporting of results in studying gender as a variable in NLP (Larson, 2017). Because iLM is not limited to training only with two environments, this architecture can also support more general bias removal goals.

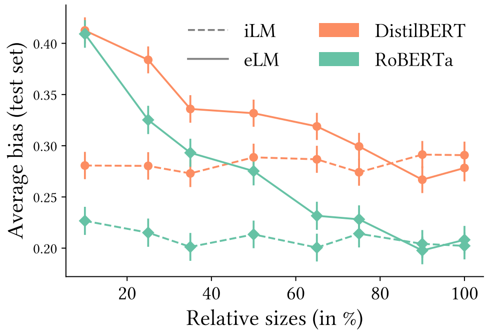

Figure 2: **Bias removal**: The x-axis represents the relative size between the modified environment and the unmodified one and the y-axis is the average bias for both iLM and eLM. Note that $\mathbb{P}(\textit{iLM} \text{ beats } \textit{eLM}) > 0.95$ when the relative size is $< 80\%$, eLM and iLM become indistinguishable for relative sizes $> 80\%$. Due to space, we report the results obtained by ensLM and mtLM in Appendix B.2 which also shows that they perform in between iLM and eLM.

known gender bias, in this case, Wikitext-2 (Merity et al., 2016). A fraction p of the data goes into Env-A, the rest $(1-p)$ goes into Env-B. Env-A remains untouched and preserves all the properties of the original data source. Whereas Env-B is intervened upon by inverting all gendered terms based on a dictionary provided by previous work (Bordia and Bowman, 2019). When $p = 1-p = 0.5$ and the language model is finetuned with eLM, this setup matches the CDA method (Lu et al., 2018) used to mitigate gender-bias in NLP. Intuitively, iLM should learn to ignore gender-based correlations no matter what is the fraction p. However, eLM is only expected to ignore them when $p = 1-p = 0.5$, i.e., the two environments have the same number of samples (Lu et al., 2018).

Experimental setup. To measure whether the correlation has been successfully removed: (i) we take all gendered terms in the test set, (ii) replace them by the MASK token, (iii) use trained models to predict the missing term, (iv) look in the softmax for the scores received by the terms of the target gendered pair. We note s_f and s_m the score assigned to the female and male terms in the softmax. (v) Finally, we compute an entropy-bias measure: $B_H = H_2\left(\frac{1}{2}\right) - H_2\left(\frac{s_f}{s_f+s_m}\right)$, where H_2 is the binary entropy (note that $H_2\left(\frac{1}{2}\right) = 1$). B_H measures the extent to which a softmax has a preference for the male or female term in a gendered pair of terms. For example, in the sentence "MASK is the best doctor" we look at the softmax score of the gendered-pair [he, she]. If a model has learned to ignore gender-based correlation, the entropy should be high (entropy-bias low), not favoring one gendered term over the other. We remove sentences with several gendered terms from the test set to avoid penalizing models for preferring a gender when the context contains gender information.

We ran the experiments for varying values of p averaging across different hyper-parameters, and report the results in Fig. 2 for iLM and eLM. The results for ensLM and mtLM are reported in Appendix B.2. See Appendix B.2 for hyper-parameters considered. For reference, the entropy bias of distilBERT and RoBERTa before training are, respectively, 0.39 and 0.46.

Analysis. Both eLM and iLM largely decrease the average entropy bias in the balanced setup but only iLM succeeds in the unbalanced setup. In the balanced setup (relative sizes close to 100%), eLM and iLM perform within each other's confidence intervals. However, in the unbalanced setup, iLM largely outperforms eLM. We note that the probability that iLM beats eLM for any given hyper-parameter configuration is > 0.9 for both distilBERT and RoBERTa when the relative sizes is below 80%. As desired iLM is not affected by the relative size of the environments. These results confirm the hypothesis, that bias removal needs a precisely balanced dataset for eLM (Lu et al., 2018), while it does not matter for iLM. Furthermore, this entropy bias reduction does not happen at the cost of worst general perplexities (see Appendix B.2). These findings are significant for the field of bias removal, iLM offers a practical and efficient way of removing biases. It is now not necessary to carefully counter-balance the bias in the augmented data. In Fig. 2, we see that already at 10% of relative size, iLM performs as well as the existing setup (100% relative size + eLM), but runs about 10 times faster; it only takes 5 min to debias a RoBERTa with iLM using 10% relative size of counterfactually augmented data.

4.3 Out-of-domain Generalization

In this experiment, we venture beyond controlled environments and test out-of-domain generalization with naturally occurring environments. We use *thePile* dataset (Gao et al., 2020) which contains 20 very diverse textual domains: OpenSubtitles, ArXiv papers, News, GitHub comments, etc.

	InD-LM↓	OoD-LM↓	GLUE↑
distilBERT			
eLM	26.02 ± 0.35	31.52 ± 0.20	72.12
ensLM	22.31 ± 0.56	32.80 ± 0.23	72.34
mtLM	22.73 ± 0.29	31.16 ± 0.44	72.22
iLM	**20.25*** ± 0.52	**30.32*** ± 0.43	**72.45**
RoBERTa			
eLM	14.55 ± 0.21	17.72 ± 0.25	76.89
ensLM	12.40 ± 0.34	17.68 ± 0.22	77.49
mtLM	12.56 ± 0.33	17.43 ± 0.23	76.55
iLM	**11.88*** ± 0.28	**16.97*** ± 0.19	**78.54***

Table 2: **ThePile environment experiments.** The first column is for language modeling evaluation in-domain (perplexity, lower is better), the second column is for language modeling evaluation out-of-domain (perplexity, lower is better), and the last column is for GLUE tasks averaged (higher is better). We mark with * the cases where iLM is statistically significantly better than other architectures (paired t-test).

Experimental setup. We randomly sample 11 domains from thePile for training, the remaining 9 domains are used for testing language models out-of-domain. Once the models are trained, using domains as environments, we evaluate their perplexity in-domain (InD) using held-out data from the training environments and OoD using data from unseen environments. See Appendix B.3 for details regarding training domains and hyper-parameters. Furthermore, the trained models are evaluated on the GLUE benchmark. Indeed, models trained with iLM can be used downstream exactly as if they were trained with eLM. For space reasons, we report aggregated results in Table 2. The results also show significant improvement of iLM over other architecture across the board. In particular, iLM is beneficial for both in-domain (InD) and out-of-domain (OoD) evaluation.

4.4 Ablation

The eLM, mtLM, and ensLM architectures serve as ablated versions of iLM testing the three main components of iLM: splitting the data into environments with one head per environment (mtLM over eLM), ensembling the heads during training (ensLM over mtLM), using the specific competitive gradient update schedule (iLM over ensLM). The four variants were run over all experiments previously described varying hyper-parameters yielding a total of 1320 experimental results (see Appendix B for details) per architecture. To get a global view, we again aggregated these results with the paired aggregation given by the Bradley-Terry model. It estimates a strength for each architecture based on how likely it is to beat other architecture on the same experiments with the same hyper-parameters. It provides a scale-independent metric-independent way to aggregate scores (Peyrard et al., 2021) across tasks and experiments.

	eLM	mtLM	ensLM	iLM
eLM	-	.92 ± .06	.26 ± .09	.28 ± .09
mtLM	.08 ± .06	-	.04 ± .04	.03 ± .04
ensLM	.74 ± .09	.96 ± .04	-	.37 ± .10
iLM	**.72 ± .09**	**.97 ± .04**	**.63 ± .10**	-

Table 3: **Paired aggregated results.** Estimated probability that one architecture (row i) is better than any other (column j) across all previous experiments, based on the pairwise Bradley-Terry aggregation model.

The results are reported in Table 3 and confirm the intuition built-up with previous experiments that simply having n environment with n heads is not beneficial on its own, as mtLM does not provide benefits over eLM. However, when combined with head ensembling (ensLM), significant improvements can be observed over both eLM and mtLM. Further significant benefits arise from the competitive gradient update specific to iLM. While both mtLM and ensLM have slightly better capacity to overfit with their n heads, they don't benefit from the invariance regularization provided by competitive gradient updates. Notice that iLM is significantly better than any other architecture, as shown by the last row of Table 3 (or equivalently, the last column).

5 Discussion

In this section, we discuss our contributions in the context of previous work.

5.1 Related Work

Domain generalization. The performance of deep learning models substantially degrades on Out-of-Domain (OoD) datasets, even in the face of small variations of the data generating process (Hendrycks and Dietterich, 2019). Blanchard et al. (2011) have proposed domain generalization (DG) as a formalism for studying this problem. In DG, the goal is to learn a model using data from a single or multiple related but distinct training domains, in such a way that the model generalizes well to any OoD testing domain, unknown during training. Recently, the problem of DG has attracted a lot of attention, and has been approached from different facets. Most of the existing methods fall under the paradigm of domain alignment (Muandet et al., 2013; Li et al., 2018b; Akuzawa et al., 2019; Liu et al., 2020; Zhao et al., 2020). Motivated by the idea that features that are stable across the training domains should also be robust to the unseen testing domains, these methods try to learn domain-invariant representations. A group of other methods is based on meta-learning (Dou et al., 2019; Balaji et al., 2018; Li et al., 2018a). The motivation behind this approach is that it exposes the model to domain shifts during training, which will allow it to generalize better during testing. Regularization through data augmentation is commonly used in the training of machine learning models to alleviate overfitting and thereby improve generalization (Zhou et al., 2021, 2020).

Domain generalization applied to language models. In NLP, the default pipeline involves pretraining a task-agnostic language model, which is then finetuned on downstream tasks. This pretraining/finetuning division of learning is already known to improve robustness on downstream tasks (Hendrycks and Dietterich, 2019). However, the language models themselves suffer from spurious correlations and poor generalization even with small perturbations of the inputs (Moradi and Samwald, 2021). To alleviate such problems, Oren et al. (2019) adapt Distribution Robust Optimization (Ben-Tal et al., 2013) to language models. This results in a new loss minimizing the worst-case performance over subsamples of the training set. They focus on domains with topic shifts. Then, Vernikos et al. (2020) use domain adversarial regularization to improve testing performance on unseen domains.

Also related to our framework are techniques aiming at de-biasing language models. Biases are correlations that may or may not be spurious but are nevertheless undesired. Removing such biases is typically done by (i) adding a bias-specific penalty term (Qian et al., 2019; Bordia and Bowman, 2019; Zhao et al., 2018) to the loss, and/or (ii) augmenting the data to counterbalance the undesired correlation (Lu et al., 2018; Zhao et al., 2017).

5.2 Environment Design

One question that might arise from the iLM training schedule is what happens when environments have no lexical overlap? Maybe no correlation remains in iLM? We emphasize that iLM learn a latent representation ϕ and stable correlation are the ones connecting this latent representation to observable, and not surface correlations between observables. To demonstrate that iLM operates on latent variables and not just on surface-level correlations, we perform a simple experiment with languages as environments. We train iLM with a pretrained multilingual model (XLM-RoBERTa) using English Wikipedia articles and Farsi Wikipedia articles as two environments. Despite absolutely no surface-level overlap, iLM is still able to improve perplexity in each language individually and does not destroy previously learned correlations. This experiment is detailed in Appendix B.4.

Also, if the number of environments grows arbitrarily large, certainly iLM would not find any stable correlations in the data. However, the choice of environments is not intended to be arbitrary; throwing as many environments as possible could not be expected to be useful. The choice of environments has to reflect assumptions about the underlying data generation mechanism. iLM then leverages the assumptions encoded in the choice of environments.

This work has shown that iLM can effectively remove unstable correlations, the next question becomes that of **environment design**: *how to choose environment splits to be useful in practice?* Useful environment splits will likely be different for different tasks and different purposes. This work already demonstrated that the new paradigm of (i) environment design then (ii) iLM is practical for language-related problems. Choosing environment splits is a flexible way to inject priors and inductive biases compared to manually deciding which correlation are desired (as in bias removal) or fully learning the causal graph (as in causal reasoning). Now, iLM provides a computationally efficient framework to inject such priors and move the discussion from model inductive biases to data inductive biases. It already offers robustness to noise, a ready-to-use bias removel strategy for any existing language model needing few data points, and improves OoD generalization.

References

Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. 2020. Invariant risk minimization games. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 145–155. PMLR.

Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Adversarial Invariant Feature Learning with Accuracy Constraint for Domain Generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–331. Springer International Publishing.

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

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1006–1016.

Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. 2013. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2178–2186.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. 2019. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6447–6458.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. 2021. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *CoRR*, abs/2101.07732.

Dan Hendrycks and Thomas G. Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

xu lan, Xiatian Zhu, and Shaogang Gong. 2018. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Brian Larson. 2017. Gender as a variable in natural-language processing: Ethical considerations. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 1–11, Valencia, Spain. Association for Computational Linguistics.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2018a. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3490–3497. AAAI Press.

Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. 2018b. Domain generalization via conditional invariant representations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3579–3587. AAAI Press.

Chang Liu, Xinwei Sun, Jindong Wang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. 2020. Learning causal semantic representation for out-of-distribution prediction. *CoRR*, abs/2011.01681.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2018. Gender bias in neural natural language processing. *CoRR*, abs/1807.11714.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. *CoRR*, abs/2108.12237.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 10–18, Atlanta, Georgia, USA. PMLR.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. 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 5356–5371, Online. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. 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 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Judea Pearl. 2018. Theoretical impediments to machine learning with seven sparks from the causal revolution. *CoRR*, abs/1801.04016.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012.

Jonas Martin Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA.

Maxime Peyrard, Wei Zhao, Steffen Eger, and Robert West. 2021. Better than average: Paired evaluation of NLP systems. 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 2301–2315, Online. Association for Computational Linguistics.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. 2021. Toward causal representation learning. *Proceedings of the IEEE - Advances in Machine Learning and Deep Neural Networks*, 109(5):612–634.

Bernhard Schölkopf. 2019. Causality for machine learning. *CoRR*, abs/1911.10500.

Ilya Shpitser and Judea Pearl. 2008. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9(64):1941–1979.

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.

Giorgos Vernikos, Katerina Margatina, Alexandra Chronopoulou, and Ion Androutsopoulos. 2020. Domain Adversarial Fine-Tuning as an Effective Regularizer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3103–3112, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System*

Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark. Association for Computational Linguistics.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. 2020. Domain generalization via entropy regularization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kaiyang Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. 2020. Learning to generate novel domains for domain generalization. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*, volume 12361 of *Lecture Notes in Computer Science*, pages 561–578. Springer.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021. Domain generalization with mixstyle. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Illustration of iLM Architecture

In the main paper, we described formally the pseudo-code involved in training iLM models. The model architecture and the logic of the training schedule is illustrated in Fig. 3 for the special-case of 2 environments ($n = 2$).

A.1 mtLM and ensLM baselines

We implemented two similar architectures that do not enjoy the same theoretical justifications.

In the mtLM baseline, the data is also split into n environments with one head per environment. As in iLM, environments take turns to send a batch of data and perform a batch update on the body of the transformer ϕ and the head associated with this environment. This is like viewing different environments as different tasks with uniform weights, even though they are all language modeling loss.

In the ensLM baseline, the data is split into n environments with one head per environment. However, here, the heads are always predicting as an ensemble like iLM. Here also the environments take turns to send a batch of data. The forward pass is exactly the same as the one of iLM. In the backward pass, every head and the transformer body ϕ are always updated for every batch of every environment.

B Details about Experiments

B.1 Robustness to Noise

Data. The data used for this experiment comes from an HTML Wikipedia Dump of August 2018. The files were pre-processed to remove the HTML content resulting in clean text articles. We randomly selected 60K articles with HTML (Env-B), and 60K different articles without HTML (Env-A). The test set contains 25K sentences coming from Wikipedia without HTML.

Hyper-parameters. We ran the experiments reported in the main paper while varying several hyper-parameters: base transformers (ϕ): [distilBERT, RoBERTa], learning rates: $[1e^{-5}, 5e^{-5}]$, number of training steps: $[10, 100, 200, 500, 2500, 5000]$, 5 random restarts with different random seeds, $2 \cdot 2 \cdot 6 \cdot 5 = 120$, ran with eLM, mtLM, ensLM, and iLM resulting in 480 experiments.

Number of lines vs. number of articles. In the main paper, we report the results of iLM and eLM when trained with environments having the same number of articles. However, the HTML articles have more lines and thus more *sentences*. Therefore, we also report in Fig. 4 the same analysis repeated when the number of lines between Env-A and Env-B is the same, meaning Env-B contains fewer articles. The conclusion remains largely unchanged in this scenario. As seen in Fig. 4 (c), iLM has still a probability of beating eLM for match hyper-parameters close to 1, highly significantly far away from 0.5.

	25	50	75	100
distilBERT				
eLM	.372± .012	.358± .033	.326± .001	**.308± .016**
mtLM	.363± .010	.352± .037	.308± .022	.328± .022
ensLM	.322± .003	.350± .032	.324± .020	.315± .015
iLM	**.309± .006**	**.322± .033**	**.318± .012**	.309± .004
RoBERTa				
eLM	.317± .010	.305± .008	.273± .045	**.259± .025**
mtLM	.308± .011	.299± .009	.271± .29	.260± .12
ensLM	.291± .011	.300± .011	**.270± .031**	.271± .033
iLM	**.290± .013**	**.291± .003**	.271± .033	.267± .025

Table 4: **Complementary gender-bias removal results.** Average bias B_H as described in Sec. 4.2 across 4 different relative sizes of environments (25%, 50%, 75% and 100%).

B.2 Bias Removal

Data. The dataset used for this experiment is Wikitext-2 (Merity et al., 2016) where we follow the existing train/dev/test split. The dictionary of gendered terms comes from Bordia and Bowman (2019) which was originally constructed to measure gender bias in language models.

The dictionary contains basic gender-pairs augmented with their variations in terms of casing, plural vs. singular forms and different spellings. The basic gendered pairs are: (actor, actress), (boy, girl), (boyfriend, girlfriend), (father, mother), (gentleman, lady), (grandson, granddaughter), (he, she), (hero, heroine), (him, her), (husband, wife), (king, queen), (male, female), (man, woman), (mr., mrs.), (prince, princess), (son, daughter), (spokesman, spokeswoman), (stepfather, stepmother), (uncle, aunt)

Hyper-parameters. We ran the experiments reported in the main paper while varying several hyper-parameters: base-model (ϕ): [distilBERT,

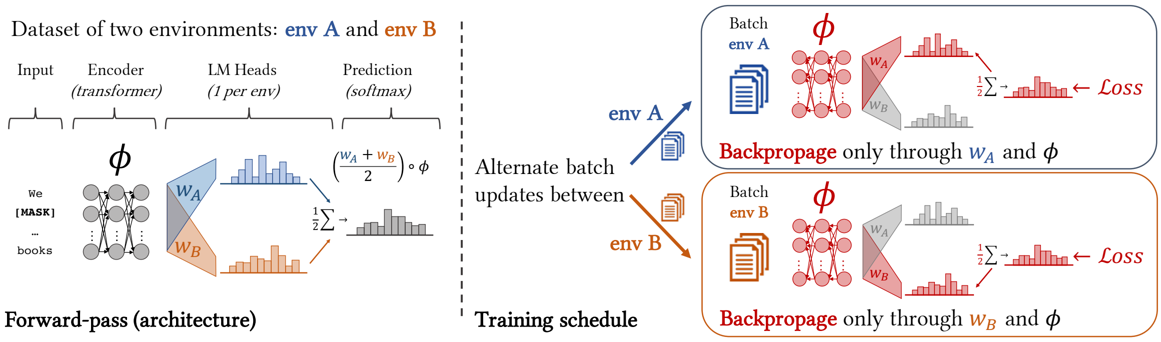
Dataset of two environments: env A and env B
Input
Encoder
(transformer)
LM Heads
(1 per env)
Prediction
(softmax)
We
[MASK]
...
books
ϕ
w_A
w_B
$\left(\frac{w_A + w_B}{2}\right) \circ \phi$
$\frac{1}{2}\sum$
Forward-pass (architecture)
Alternate batch
updates between
env A
env B
Training schedule
Batch
env A
Batch
env B
Loss
Backpropage only through w_A and ϕ
Backpropage only through w_B and ϕ

Figure 3: **Model description** In the forward pass, input text goes through the main body of language model noted ϕ (e.g., a Transformer (Devlin et al., 2019)), then one head per environment predicts logits over the vocabulary. These predictions are averaged over all heads and go through a softmax. During training, the model receives a batch of data from one environment e and performs a gradient update only on the parameters of the main body of the language model (ϕ) and on the parameters of the head tied to this environment w_e. Then batches are taken from each environment in a round-robin fashion.

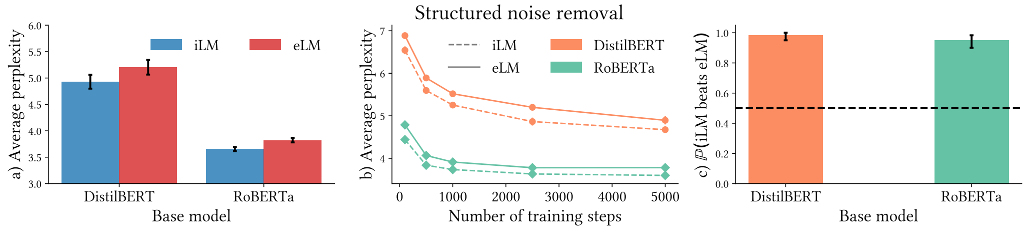

Figure 4: Structured noise removal experiment with environments having the same number of lines: a) average perplexity over all hyper-parameters b) average perplexity as a function of the number of training steps (for learning rate 10^{-5}), c) Probability that iLM is better than eLM when compared on the same hyper-parameters

RoBERTa], learning-rates: $[1e^{-5}, 5e^{-5}]$, number of training steps: $[10, 50, 100, 200, 1000, 2500]$, 5 random restarts with different random seeds. This gives $2 \cdot 2 \cdot 6 \cdot 5 = 120$ experimental parameters, ran for eLM, iLM, mtLM, and ensLM while varying the relative sizes of environments in $[10, 25, 30, 50, 70, 75, 90, 100]$ resulting in 3840 experiments.

Results for mtLM and ensLM. In Fig. 4, we report the average bias as a function of the relative sizes of environments for mtLM and ensLM alongside those of iLM and eLM. We also observe here that iLM outperform other architectures. Interestingly, ensLM seems to bring benefits in comparison to eLM and mtLM.

Details about the results. Here, we report complementary analysis compared to the results described in the paper. We report the performance of eLM and iLM as a function of the number of training steps and the probability that iLM is better then eLM when matched on hyper-parameter configuration as computed by the Bradley-Terry model. This is reported by Fig. 5 for two relative size: 25% (the modified environment has 4 times fewer examples) and 100%.

Perplexities after training. To ensure that the gender-based correlations were not removed at the cost of a worse perplexity, we report in Table 5 the perplexities of iLM models in comparison eLM ones on the test set of Wikitext-2. For reference, before our training distilBERT and RoBERTa had, this same test set, perplexities of 14.25 and 6.92, respectively.

In Table 5, the 95% confidence intervals all give uncertainties ≈ 0.15, meaning that for a fixed base model (distilBERT or RoBERTa) all perplexities are within each other's error bounds. There is no significant perplexity difference between eLM and iLM or between the unbalanced and balanced setups.

B.3 Out-of-domain Generalization

Data. The data used for this experiment comes from subsamples of thePile (Gao et al., 2020).

We randomly selected train and test domains as follow:

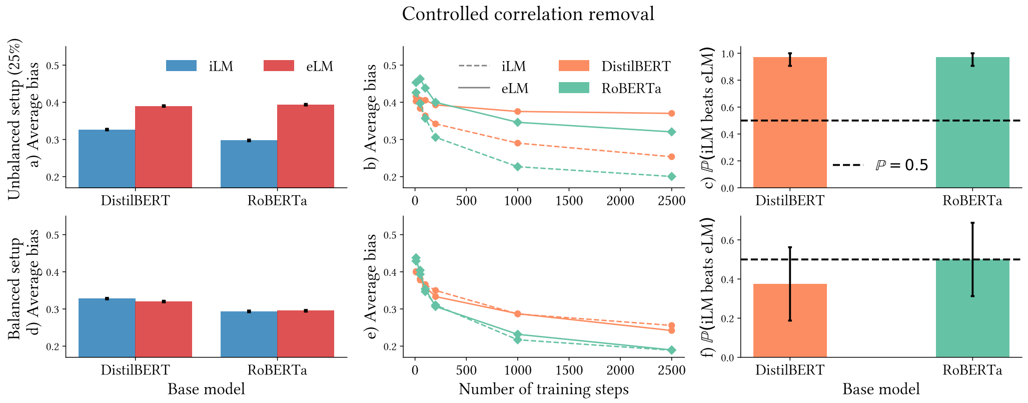

Figure 5: **Controlled correlation removal experiment**: On the first row, the modified environment is 25% of the size of the unmodified environment. On the second row, both have the same number of samples. On the left-most column, average bias over all hyper-parameters. On the center column: average bias as a function of the number of training steps. On the right-most column: Probability that iLM is less biased than eLM when compared on the same hyper-parameters.

	Unbalanced	Balanced
iLM RoBERTa	4.16	4.13
iLM distilBERT	5.82	5.81
eLM RoBERTa	4.14	4.14
eLM distilBERT	5.82	5.85

Table 5: Perplexities of iLM and eLM models after training.

- **Train**: "europarl", "freelaw", "dm mathematics", "youtubesubtitles", "USPTO backgrounds", "arxiv", "books3", "wikipedia(en)", "stackexchange", "hackernews", "pile-cc"

- **Test**: "github", "ubuntu irc", "openwebtext2", "pubmed central", "enron emails", "pubmed abstracts", "gutenberg pg-19"

Hyper-parameters. We ran the experiments reported in the main paper while varying several hyper-parameters: base-model (ϕ): [distilBERT, RoBERTa], learning-rates: $[1e^{-5}, 5e^{-5}]$, number of training steps: $[2500, 5000, 25000, 50000]$, 5 random restarts with different random seeds, for eLM, mtLM, ensLM, and iLM. This results in $2 \cdot 2 \cdot 4 \cdot 5 \cdot 4 = 320$ experimental models, each evaluated in 3 tasks: in-domain language modeling, out-of-domain language modeling, GLUE. This is a total of 960 experimental setups.

Evaluation. For the in-domain language modeling evaluation, we measure perplexity on 10K held-out sentences from each of the train domain. Similarly for out-of-domain language modeling evaluation, we measure perplexity on 10K sentences from each of the test domain.

For GLUE, we used the default scripts from huggingface to evaluate trained models from checkpoints.

B.4 Languages as Environments

One question that might arise from iLM training schedule is whether it simply focuses on surface-level lexical correlations in the data. For example, if the lexical correlations are different across environments, maybe no correlation remain generalizable and iLM learns an empty set of correlations. To better demonstrate that iLM operate on latent variable and not on surface-level correlations, we perform a simple experiment with languages as environments.

Description. We use two languages with no lexical overlap: English and Farsi. We put english Wikipedia articles as one environment and farsi Wikipedia articles as the other. In this setup, no surface-level correlations can generalize across environment as the two environments don't even have the same vocabulary.

We train iLM with a multilingual pre-trained RoBERTa: XLM-RoBERTa for 5000 steps with these two environments of equal size (10K articles per language). Then, we test whether this choice of environments destructs previously learn correlations in the language model by comparing perplexities on a balanced held-out test set of english and farsi documents against the model before finetuning. If the perplexities decrease, we would

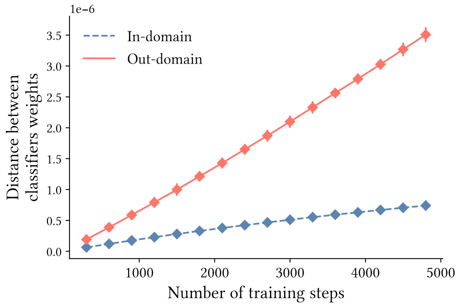

Figure 6: Comparing distance between heads weights in- and out-domain as functions of the number of training step. (95% confidence interval from random restart with different seeds.)

conclude that iLM destroy surface-level correlations.

Results. We found that before finetuning, XLM-RoBERTa had a perplexity of 14.56 on the held-out test set, where iLM could improve it perplexity down to 6.44. This indicates that iLM with environments having no lexical overlap does not destroy previously learned correlations. It can even improve its perplexities for each language. A possible reason why iLM can even improve so dramatically compared to before finetuning might come from the fact that ϕ learns to recognize the languages, separate them and treat them separately. Similar effects have been observed in previous work (Guo et al., 2021) when the correlation between the environment index and the target variable is very strong (which is the case here).

B.5 Head dynamics

The main components of our framework are the heads and their training dynamic. Therefore, we investigate aspects related to behaviour of the heads.

Description. During training, the loss of each head is still entangled with the prediction of every other head. So we wonder whether the heads still capture information related to the environment it is tied to during training. In particular, we ask (i) whether the parameters of the heads for different environments are drifting apart during training? Indeed, all heads are initialized to the same pretrained weights at the beginning of training. (ii) Are the parameters of the heads predicting which environments are more similar?

Experimental setup. To answer these two questions in one go, we take two environments A and B and split each of them into two new environments resulting in A_1, A_2, B_1, and B_2 such that A_1 and A_2 are very similar B_1 and B_2 are very similar but A_i and B_i are different. We then train iLM with the four environments and, thus, with four heads w_{A_1}, w_{A_2}, w_{B_1}, and w_{B_2}. We measure whether the heads' weights can predict the similarities between A's and B's environments.

$$D_{in} = \frac{1}{2}\left(d(w_{A_1}, w_{A_2}) + d(w_{B_1}, w_{B_2})\right), \quad (1)$$

$$D_{out} = \frac{1}{4}\sum_{i,j} d(w_{A_i}, w_{B_j}), \quad (2)$$

where d is the L2 distance between the linearized weights of two heads. Then, D_{in} is the average distance between heads tied the same domain, and D_{out} is the average distance between heads tied to different domains. Remember that in this case, there are 2 domains A and B and 4 environments A_i and B_i.

In this experiment, we randomly select the base environments A and B from the domains of thePile (A is the Enron-Email, and B is PubMed abstract). We create A_i and B_i by randomly subsampling 2 environments of the same size from each domain. We train iLM with RoBERTa for 5000 training steps, taking checkpoints of the heads every 500 steps. We perform 10 random restarts with different seeds to uncertainty estimates. In Fig. 6, we report D_{in} and D_{out} as functions of the number of training steps.

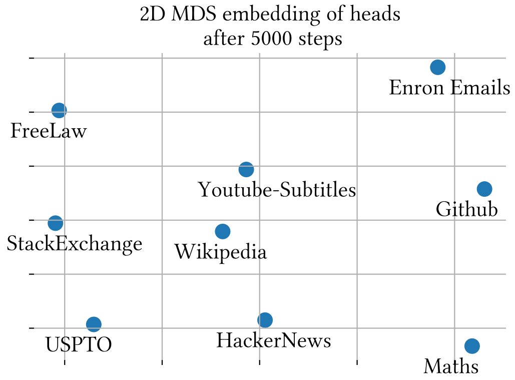

Figure 7: Heads embeddings: 2D projection of the heads parameters similarity structure after training iLM with RoBERTa for 5000 steps with 9 domains. Each dot represent one head of the model after training and the labels indicate to which domain it is tied to.

Analysis. We first notice that indeed the heads are drifting apart from each other as training advances. More interestingly, the distance between heads from the same domain is significantly much smaller than the distance between heads from different domains. We conclude that heads retain environment-specific information in their parameters and are predictive of environment similarities.

Now, we visualize the geometry of head similarity by training iLM with RoBERTa for 5000 steps with 9 environments from thePile: . After training, we take the heads' parameters and compute the pairwise distance between all 9 heads and embed them in 2D with Multi-Dimensional Scaling to visualize the similarity structure. The result is depicted in Fig. 7.