Unshuffling Data for Improved Generalization

Damien Teney Ehsan Abbasnejad Anton van den Hengel
Australian Institute for Machine Learning
The University of Adelaide
Adelaide, Australia
{damien.teney,ehsan.abbasnejad,anton.vandenhengel}@adelaide.edu.au

Abstract

Generalization beyond the training distribution is a core challenge in machine learning. The common practice of mixing and shuffling examples when training neural networks may not be optimal in this regard. We show that partitioning the data into well-chosen, non-i.i.d. subsets treated as multiple training environments can guide the learning of models with better out-of-distribution generalization. We describe a training procedure to capture the patterns that are stable across environments while discarding spurious ones. The method makes a step beyond correlation-based learning: the choice of the partitioning allows injecting information about the task that cannot be otherwise recovered from the joint distribution of the training data.

We demonstrate multiple use cases with the task of visual question answering, which is notorious for dataset biases. We obtain significant improvements on VQA-CP, using environments built from prior knowledge, existing meta data, or unsupervised clustering. We also get improvements on GQA using annotations of "equivalent questions", and on multi-dataset training (VQA v2 / Visual Genome) by treating them as distinct environments.

1. Introduction

The best of machine learning models can sometimes be right for the wrong reasons [2, 20, 25, 64]. The ubiquitous paradigm of empirical risk minimization (ERM) produces models that capture all statistical patterns present in the training data[1]. However, not all of these patterns are reliable and reflective of the task of interest. Some of them result from confounding factors, sampling biases, and other annotation artifacts specific to a given dataset. We call these patterns *spurious*[2] and a model that relies on them will generalize poorly to test data obtained in different conditions (*i.e.* out-of-distribution or OOD data).

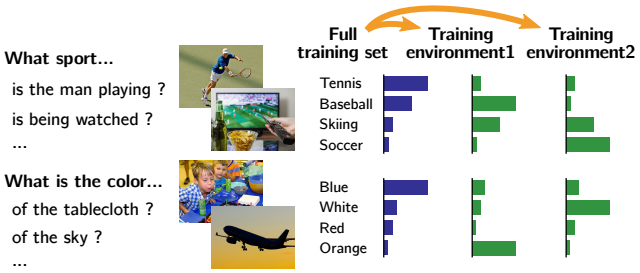

Figure 1. Datasets for visual question answering contain biases and spurious correlations: the first few words of a question are associated with a peaky distribution over answers (blue histograms). Models that guess their answers using these correlations generalize poorly. We improve by **partitioning the data** into multiple training environments across which the spurious correlations vary (green histograms) while reliable correlations are stable. Our training procedure produces a model that relies on these stable correlations such that it generalizes much better at test time.

The **limits of ERM** on OOD data are oftentimes overlooked and eclipsed by the common practice of evaluating on test data i.i.d. to the training data – a central assumption of classical learning theory [65]. The awareness of these limits has grown with that of their practical implications, from poor transfer across datasets [64] to biases and fairness issues [1] and vulnerability to adversarial inputs [26]. As a result, benchmarks with OOD test sets are becoming increasingly common in vision and NLP [2, 4, 9, 28, 36, 43]. This paper presents a training paradigm to improve OOD generalization.

The study of **generalization in computer vision** has a long history [19, 40, 47]. The rise in popularity of high-level tasks like visual question answering (VQA) [7], visual dialogue [18], or vision-and-language navigation [6] has made the topic even more important. The complexity of these tasks and the combinatorial explosion of the size of their input domain make it impossible to process

[1] We use *patterns* and *correlations* interchangeably to refer to statistical relationships between observed random variables, typically the input(s) and output(s) of a supervised learning task.

[2] The literature also uses *dataset biases* to refer to spurious correlations between inputs and outputs that are dataset-specific [64].

training data densely spanning this space. Models trained with ERM are then more likely to latch on spurious correlations (because they are often easier to fit [55]) rather than on the true reasoning process that underlies the task [35]. VQA was shown empirically to be a prime example of this issue. Many methods have been proposed to address it [2, 12, 16, 26, 28, 46, 53].

We propose a **general method to improve OOD generalization**. We discourage the model from using spurious correlations that only appear in subsets of the training data, and rather ensure that it uses reliable ones that are more likely to generalize at test time. More precisely, we first partition the data into multiple training environments [8] such that spurious correlations vary across environments while reliable ones remain stable. We later describe multiple strategies to build such environments, using unsupervised clustering, prior knowledge, and auxiliary annotations in existing datasets. Second, we train multiple copies of a neural network, one per environment. Some of their weights are shared across environments, while others are subject to a variance regularizer in parameter space. This leads the model to extract features that are stable across environments (*i.e*. features that do not represent environment-specific properties) since they are optimized to be predictive under a classifier common to all environments (as encouraged by the variance regularizer). Additional intuitions and reasons why this approach is superior to ERM are discussed in Section 3.3.

We provide **empirical evidence of improvements** in three distinct use cases on the task of VQA. First, we demonstrate improved resilience to language biases with VQA-CP [2]. Second, on GQA [33], we show how to use annotations of equivalent questions (some training questions being rephrasings of others). We obtain substantial gains over simple data augmentation with these equivalent questions in a small-data regime. Third, we show a small benefit in training a model on multiple datasets by treating VQA v2 [25] and Visual Genome QA [41] as two environments rather than one aggregated dataset. Of these three use cases, the first is the most well-studied but **our method has a much wider scope than the VQA-CP dataset**.

The contributions of this paper are summarized as follows.

1. We propose to partition existing datasets into training environments to improve generalization. We describe the requirements for the partitioning and a procedure to train neural networks to rely on stable correlations across environments while ignoring spurious ones.
2. We apply the method to three use cases with the task of VQA: (1) resilience to language biases, (2) leveraging known relations of equivalence between specific training questions, and (3) multi-dataset training.
3. We provide empirical evidence of clear improvements and a extensive sensitivity analysis to hyperparameters and implementation choices.

2. Related work

This paper touches on fundamental aspects of machine learning and so is related to many existing works.

Dataset biases in vision-and-language tasks. Several popular datasets used in vision-and-language [24] and natural language processing [71] have been shown to exhibit strong biases. A model trained naively on these datasets can exhibit surprisingly good performance by relying on dataset-specific biases without capturing the true mechanisms of the task. On the evaluation side, there is a trend towards out-of-distribution test data to better identify this behaviour (*e.g*. [2, 4, 9, 28, 36, 43, 71]).

Invariances and generalization. Resilience to dataset biases cannot be solved by simply collecting more data from the same distribution, since it would still contain the same unreliable patterns. The data collection process can be improved [24, 71, 72] but this option only addresses precisely identified biases and confounders. Improving generalization requires to bring in information (often implicitly) about the mechanisms of the task of interest that go beyond what is represented by the joint distribution of the training data (see discussion Section 3.3). Common methods include architecture design and data augmentation [42, 58] to specify input transformations the model should be invariant to. This helps ignoring spurious correlations and improves generalization but it requires explicit knowledge of desirable invariances. In comparison, our method discovers invariances implicitly. We train a model to rely on input features that are similarly predictive (*i.e*. invariant) across environments. The required expert knowledge is displaced to the specification of the partitioning into environments.

Aggregating datasets. Using training data collected in different conditions is often beneficial for generalization (*e.g*. in [61] for VQA) because biases in each dataset are likely different and they can "cancel each other out". We argue however that treating aggregated datasets as a collection of samples from a unique distribution loses valuable information. Our method treats them as distinct training environments and seeks to identify patterns that are similarly predictive across them while discarding those that are dataset-specific. Khosla *et al*. [40] also showed that accounting for bias when combining datasets was beneficial for generalization.

Domain adaptation and domain generalization. Training a model under multiple environments is reminiscent of domain adaptation [21]. But our objective is not to adapt to one particular target domain but rather to generalizes across a range of unknown conditions. Our setup is more similar to the recently-introduced terminology of domain generalization [67, 27, 67]. These methods specifically target image recognition benchmarks like PACS and VLCS.

Ensembles. Our method trains multiple copies of a model in parallel, which is superficially similar to ensembling [74] and bootstrap aggregation a.k.a. bagging [11]. Traditional ensembles however combine models in output space. We combine models in parameter space[3]. We show empirically that the improvements from our approach are distinct from (and complementary to) those of traditional ensembling. Bagging uses uniform sampling, whereas the point of our method is to exploit prior knowledge to build the training environments.

Robustness in VQA. State-of-the-art models for VQA have been shown to be strongly reliant language biases. Benchmarks have been designed to better study the issue [2, 33, 37]. VQA-CP [2] allows out-of-distribution evaluation, where the joint distribution of questions and answers is different in the training and test sets. Methods have been proposed with strong improvements on VQA-CP ([12, 16, 26, 28, 46, 53] among others). However, many were been shown to cheat the evaluation by exploiting knowledge of the construction of VQA-CP that should have been kept private [62]. In [60], the authors use counterfactual examples to learn which input features to focus on. This is comparable to our method in that we learn by contrasting training environments, whereas [60] proceeds at the instance level. In [59], the authors exploited auxiliary annotations of the GQA dataset to improve robustness, which we also use in some of our experiments.

Fair and bias-resilient machine learning. Addressing dataset biases is also motivated by issues of fairness [1, 30, 68, 73]. These works aim to build predictive models that are invariant to specific attributes of the input, such as gender or ethnicity. These attributes need to be specified and annotated, which is very limiting. For example in VQA, there is a known desired invariance to some linguistic patterns in the question, but their exact form is not known and cannot be annotated as a discrete attribute.

Invariant risk minimization. This paper is strongly inspired the principle of invariant risk minimization (IRM) [8]. Arjovsky *et al*. showed how training under multiple environments can improve OOD generalization. Our contributions over [8] are threefold: an alternative, easy-to-train implementation, an application to a real large-scale dataset, and demonstrations of how to obtain training environments. Other implementations were proposed [3, 13, 15, 34] but mostly demonstrated on toy data. Theoretical and empirical comparisons of variants of IRM are missing pieces to address in the future. Creager *et al*. [17] attempted unsuccessfully to discover environments automatically for IRM. Their failure, even on toy data, supports the view that the true source of improvements is in the information used to obtain or create the environments.

[3] Averaging predictions or final weights is equivalent with a linear classifier. It is not in our case of a non-linear output.

Learning from groupings of data has been studied from statistical [10, 29] and causal [48, 54] points of view. Heinze-Deml and Meinshausen [29] used a variance regularizer on predictions across versions of an example, such as multiple photos of a same individual. Our variance regularizer, in comparison, acts on the parameters of the model. And we do not require correspondences between specific training examples. [19] is another classic work using a variance regularizer for multi-task learning. And Vasilescu *et al*. [66] were among the first to compute invariant representations for the causal factors of image formation, which they did for face recognition and human motion signatures.

3. Proposed approach

3.1. Partitioning data into training environments

The main intuition behind our method is that the training data contains both reliable and spurious correlations between inputs and labels, and that it is sometimes possible to partition the data into "training environments" in which the strength of the spurious ones is affected more than the reliable ones. We then train a model to rely on the correlations that are stable across environments. The corollary is that it ignores the environment-specific spurious ones.

As an example in VQA, let's imagine the question *What animal is in the picture ?* A reliable correlation is the presence of canines in images that have dog as answer. An example of a spurious correlation is that most questions starting with *What sport...* have *tennis* as answer. A model that relies on this correlation irrespective of image contents or of the rest of the question will generalize poorly. This spurious correlation results from annotator and selection biases. Conceivably, data collected from two groups of annotators will exhibit different biases, *e.g*. one group mostly picking images with tennis as the answer, the other football. Our approach identifies such groupings of the data as training environments then trains a model that uses the correlations that are stable (*i.e*. similarly predictive) across environments.

Concretely, we partition the training set $\mathcal{T}=\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i$ of inputs $\boldsymbol{x}_i$ and labels $\boldsymbol{y}_i$ (one-hot vectors in a classification task) into E disjoint training environments $\mathcal{T}_e$ such that $\bigcup_{e=1}^{E} \mathcal{T}_e = \mathcal{T}$. The environments are built to isolate the effect of spurious correlations, such that only the strength of reliable correlations remains stable across all environments. We provide additional justification for the principle in Section 3.3 and we describe strategies to build environments from existing datasets in Section 4. We show that they can be built by unsupervised clustering, by injecting prior knowledge, and by leveraging auxiliary annotations from existing datasets. Next, we describe how to train a model across environments to rely on stable correlations while ignoring spurious ones.

3.2. Training over multiple environments

Our goal is to learn a predictive model Φ that maps an input $\boldsymbol{x}$ to an output $\hat{\boldsymbol{y}} = \Phi(\boldsymbol{x})$ such as a vector of class probabilities in a classification task. We represent the model as the combination of a feature extractor and a subsequent linear classifier. The feature extractor $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ (typically with a deep neural network) uses parameters $\boldsymbol{\theta}$ to extract a vector $\boldsymbol{h} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$. The subsequent linear classifier is a matrix of weights $\mathbf{W}$ and the whole model is described as $\Phi(\boldsymbol{x}) = \mathbf{W} f_{\boldsymbol{\theta}}(\boldsymbol{x})$. The standard training procedure is to optimize $\boldsymbol{\theta}$ and $\mathbf{W}$ for maximum likelihood on the training set $\mathcal{T}$ under a loss $\mathcal{L}$, *i.e.* solving the following optimization problem:

$$\underset{\boldsymbol{\theta},\mathbf{W}}{\arg\min} \; \Sigma_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \, \mathcal{L}\big(\mathbf{W} f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}\big) \; . \tag{1}$$

In our method, assuming a prior definition of training environments $\mathcal{T}_e$ (see Section 3.1), we want to train the model to be highly predictive on the training environments as well as on an OOD test set, in which only the input/output correlations *common to all training environments* can be assumed to hold. We train a different model $\Phi_e(\boldsymbol{x}) = \mathbf{W}_e \, f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for each environment. The feature extractor $f_{\boldsymbol{\theta}}(\cdot)$ is shared, such that it identifies features common to all environments. But a different classifier $\mathbf{W}_e$ is optimized for each environment. We want the features extracted by $f_{\boldsymbol{\theta}}(\cdot)$ to be stable, *i.e.* similarly predictive across environments. For this, we choose to encourage the parameters of the classifiers $\mathbf{W}_e$ to converge to a common value. This is naturally implemented by minimizing their variance over $e = 1..E$.

At test time, we use $\Phi^\star(\boldsymbol{x}) = \bar{\mathbf{W}} \, f(\boldsymbol{x})$, where $\bar{\mathbf{W}}$ is the arithmetic mean of $\mathbf{W}_e$ over e. Since the variance regularizer brings all $\mathbf{W}_e$ toward a common value during training, the arithmetic mean is a natural choice. The complete optimization task is defined as:

$$\underset{\boldsymbol{\theta},\mathcal{W}}{\arg\min} \, \Sigma_e \; \Sigma_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}_e} \; \mathcal{L}\big(\mathbf{W}_e f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}\big) \; + \; \lambda \operatorname*{Var}_e(\mathbf{W}_e) \tag{2}$$

where λ is a scalar hyperparameter, $\mathcal{W} = \{\mathbf{W}_e\}_{e=1}^{E}$, and $\operatorname{Var}_e(\mathbf{W}_e)$ is the variance of classifier weights. The standard definition of the variance gives

$$\operatorname*{Var}_e(\mathbf{W}_e) \; = \; (1/E) \; \Sigma_e \, ||\mathbf{W}_e - \bar{\mathbf{W}}||^2 \tag{3}$$

$$\text{with} \quad \bar{\mathbf{W}} = (1/E) \; \Sigma_e \mathbf{W}_e \; . \tag{4}$$

We refer to this definition as the "absolute variance" in our experiments. Finding a unique best value for λ in Eq. 2 proved difficult because the magnitude of the weights can vary widely during the early stages of the optimization. As a remedy, we use a relative measure of variance that rescales each term by the weights' magnitude:

$$\operatorname*{Var}_e(\mathbf{W}_e) \; = \; (1/E) \; \Sigma_e \, \left(||\mathbf{W}_e - \bar{\mathbf{W}}||_2 / \|\mathbf{W}_e\|_1\right)^2 \tag{5}$$

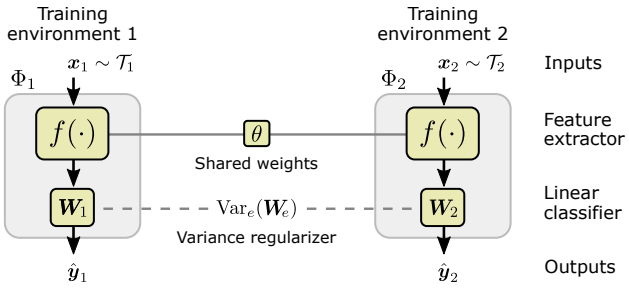

Figure 2. During training, we optimize a different copy of the model under each environment (each sees a different subset of the data). Two environments are pictured but our experiments use up to 18. The objective is to have the model rely on statistical patterns that are stable across environments. The weights of the feature extractor ($\boldsymbol{\theta}$) are shared across environments, those of the classifier ($\mathbf{W}_e$) are not. A regularizer encourages the latter to converge to a unique solution simultaneously optimal across environments. At test time, we use the arithmetic mean of these weights $\bar{\mathbf{W}}$.

It slightly improves our results (see Table 1) and makes λ easier to tune. We also found small improvements in optimizing Eq. 2 alternatively: one mini-batch serves to update $\boldsymbol{\theta}$, another one to update $\mathbf{W}$, repeating until convergence. It slightly improves the final accuracy but it is not crucial to the method and was only used in select experiments reported in Table 1.

3.3. Why it works

This section discusses further connections with existing methods and formal interpretations of generalization.

Causal view of generalization. Improving generalization requires distinguishing dataset-specific artifacts from the actual mechanisms of the task of interest. A causal view of generalization formalizes this as identifying causal properties[4] of the data-generating process behind the observed data [8, 51]. A critical outcome of this formalization is that such causal properties are provably impossible to recover solely from the joint distribution over inputs and outputs that a standard training set represents. In other words, spurious and reliable correlations cannot be distinguished from one another.

Common avenues for bringing the missing information include architecture design (setting a prior on the causal structure), data augmentation (defining invariances in the input domain) and other task-specific inductive biases. Our method falls in another category that seeks to exploit implicit training signals found unexploited in existing datasets [8, 60, 29, 38]. More specifically, we use multiple

[4] We do not claim to identify a causal model, we only aim at producing a predictive model. The causal formalization only serves to highlight that properties relevant to OOD generalization cannot be recovered from the joint distribution represented by a standard dataset, but can be informed by training across well-chosen environments.

training environments. These emerge naturally in data collected from multiple annotators, sites, or annotation interfaces. Data points are usually collated into a single dataset, but this loses information. Our premise is that such environments elicit different spurious correlations without affecting the reliable patterns inherent to the task of interest.

The fact that observational data from multiple environments can be informative about causal properties of the data generating process may seem at odds with basic principles of causal reasoning [49]. This is resolved when considering each environment as an intervention on the data generating process. Assuming these interventions only act on variables spuriously correlated with the output (hence the importance of a well-chosen partitioning), the causal mechanisms between non-intervened variables and the output will remain unchanged by the principle of independent causal mechanisms [52]. The statistical patterns inherent to the task thus remain stable across environments.

Source of improvements. The improvements obtained through our method therefore hinge on the information used to obtain training environments (*e.g.* multiple existing datasets) or to create them by "unshuffling" an existing dataset. The latter requires the practitioner to bring in meta knowledge about the task or dataset, such as axes along which to cluster the data. Conversely, environments made as random partitions are not expected to bring any benefit, as verified in our experiments (see Table 1).

Invariant risk minimization. Our training procedure is inspired by the principle of invariant risk minimization (IRM) [8]. IRM proposes to identify a representation of data such that the optimal classifier, on top of this representation, is identical across environments. Formally, using our notations, this amounts to optimizing the feature extractor $f_{\boldsymbol{\theta}}(\cdot)$ and linear classifier $\mathbf{W}$ for the following objective:

$$\min_{\boldsymbol{\theta},\mathcal{W}} \; \Sigma_e \; \Sigma_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}_e} \; \mathcal{L}\big(\mathbf{W}^\star f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}\big) \tag{6}$$

$$\text{s.t. } \mathbf{W}^\star \in \arg\min_{\mathbf{W}} \Sigma_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}_e} \; \mathcal{L}\big(\mathbf{W} f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}\big), \; \forall \; e. \tag{7}$$

The constraint on $\mathbf{W}^\star$ is the crux of the principle. A classifier that is optimal in a given environment can only use the features that are reliable predictors in that environment. Requiring the classifier $\mathbf{W}^\star$ to be simultaneously optimal across all environments (*i.e.* at the intersection of all environment-specific optima) means that it can only use stable features. In other words, consider a spurious correlation, specific to an environment e, between the output labels and a feature $\tilde{\boldsymbol{x}}$. A model (feature extractor and classifier) trained in isolation on e would use this feature $\tilde{\boldsymbol{x}}$. However, this spurious correlation does not hold in another environment e'. Even though the shared feature extractor could extract some semblance of the feature $\tilde{\boldsymbol{x}}$ in e', this feature will not be predictive in the same way as in e. Therefore, the optimal classifier in e' will not use $\tilde{\boldsymbol{x}}$ in the same way. Since we are looking for a unique classifier that is simultaneously optimal in e and e', the shared feature extractor must ignore this unreliable feature, and only extract those that are *similarly predictive* across environments.

Proposed method and IRM. The objective of Eq. 7 involves an impractical nested optimization. The approximation proposed in [8] replaces the constraint with a regularizing term in the objective that uses the gradient of the environment-specific risk with respect to the classifier. The resulting objective is highly convex and has been reported to be difficult to train in practice. Our method (Eq. 2) uses the variance of $\mathbf{W}_e$ as a regularizer. The gradient of the risk in [8] is motivated as a measure of "how optimal" a classifier is. Our version operates directly in the parameter space of the classifier. As explained in Section 3.2, our version intuitively leads to stationary points that satisfy the IRM principle, but further work is warranted to study its convergence properties and possible guarantees discussed in [8]. Our results are only empirical but showed it to be very stable during training and highly effective in our use cases.

Finally, recent breakthrough theoretical [39] and empirical results [56] proved identifiability in non-linear ICA when the generating process is conditional distribution of which the conditioning variable is observed. This setting is analogous to training across environments with the environment ID interpreted as the conditioning variable. Further work is needed to establish formal connections with these results.

4. Experiments

We present three applications on the task of VQA, which is notorious for dataset biases. Our strongest results are with the VQA-CP dataset [2] which is designed to test out-of-distribution generalization. The other two application use GQA [33], and VQA v2 [24] combined with Visual Genome QA [41]. The quantitative improvements in these other two applications are smaller but they demonstrate the wider applicability of the method. Most other methods in Table 2 are specific to VQA-CP.

Implementation. We implemented the method on top of the "bottom-up and top-down attention" model [61] (details in supp. mat.) since it serves as the baseline of most competing techniques on VQA-CP [12, 16, 26, 28, 46, 53]. Our method should readily apply to recent models [14, 22, 23, 44, 45, 57, 70] including those with stronger baseline performance on GQA [31, 32]. Evaluations on VQA-CP follow the guidelines of Teney *et al.* [62], including performing ablations on "Other" questions only, and reporting both in-domain and OOD performance without retraining. Our results are also averaged across multiple runs (thus not to

outcome of one possibly-lucky random seed) unlike most results reported in the literature.

4.1. Robustness to language biases (VQA-CP)

Experimental setup. The VQA-CP dataset [2] was constructed by reorganizing VQA v2 [24] such that the correlation between the question type and correct answer differs in the training and test splits. For example, the most common answer to questions starting with *What sport...* is *tennis* in the training set, but *skiing* in the test set. A model that guesses an answer primarily from the question will perform poorly. In our experiments, we report the accuracy on the official test set, but also on a validation set that we built by holding out 8,000 random instances from the training set. This serves as to measure "in-distribution" performance, while the test set serves to measure generalization to out-of-distribution data. As discussed in [62], evaluation on the 'yes/no' and 'number' categories of VQA-CP have unintuitive issues (for example, randomly guessing yes/no on the former category achieves 72.9% while a method like [2] only gets 65.5%; thus, a random, untrained model is usually better than a trained one). For these reasons, our ablation study uses only the 'other' type of questions.

Environments from ground truth question types. We first present experiments for which we built training environments with the ground truth type of questions (provided with the dataset). Each training question has one label among 65. This label serves as a natural clustering of the data. We assign the 65 clusters randomly to E environments, splitting clusters as needed to obtain the same number of training questions per environment. We trained our method with a different number of environments (see Fig. 3b). The point E=1 corresponds a standard training of the model with the whole dataset. The plot shows a clear improvement with multiple environments, with a peak performance with E=15. Why does the accuracy decrease with more environments ? We believe that the diversity and amount of data in each environment then gets too low. We experimented with other strategies (not reported in plots and tables) to assign clusters to environments other than randomly, by maximizing or minimizing the variation in the answer distribution in each environment (compared to the whole dataset). We found that the random assignment performed best. It keeps the distribution of answers relatively similar across environments, unless E is too large, which further explains the slight decrease in accuracy then.

Environments by clustering questions. We now present experiments where the environments are built through unsupervised clustering of the questions. We do not use the ground truth question types here. We rely on our prior knowledge that a model should not be overly reliant on the general form of a question. We represent the questions as binary bag-of-words vectors (details in supp. mat.) and cluster them with K-means. As above, we then assign the clusters randomly to E environments ($E < K$). We plot in Fig. 3c the accuracy of the model against the number of clusters K. There is a distinct broad optimum. The best accuracy is close but still inferior to the strategy that uses the ground truth question types (compare the peaks in Fig. 3b and c). We measured the similarity of the unsupervised clustering with the ground truth type in terms of Rand index, and noted that it was positively correlated with the accuracy. This shows that using ground truth types is the better strategy and the clustering approximates it.

Ablative analysis. We provide an ablation study in Table 1. The performance substantially increases on the test set with the proposed method compared to all baselines. The variance regularizer is crucial to the success of the method. We plot in Fig. 3a the accuracy as a function of the regularizer weight (λ in Eq. 2). There is a clear optimum, with higher values being generally better (the plot uses a log scale). In Table 1, we also observe that the relative variance performs slightly better than the absolute variance. We also note that the alternating optimization scheme performs slightly better. It works best after a a few epochs of non-alternating "warm-up", during which all parameters are updated together. The use of the alternating optimization is not crucial to the overall success of the method, and it is not used in any other experiment.

Comparison to existing methods. We trained our method on the whole VQA-CP dataset, including 'yes/no' and 'number' questions to compare it against existing methods (see Table 2). Our method surpasses all others on 'other', most of them by a large margin. The method of Clark *et al*. [16] gets better results on the 'yes/no' and 'number' questions, but its results on the standard splits of VQA v2 are also down to baseline levels (*i.e*. similar to a random guess out of the subset of answers used in each category). In comparison, our performance on the standard splits remains higher. Note that some competing methods admittedly use the test set as a validation set (!) for hyperparameter selection and/or model selection [2, 26]. We rather hold out 8k instances from the training set to serve as a validation set as recommended in [62]. They serve for example to monitor training and determine the epoch for early stopping.

4.2. Invariance to equivalent questions (GQA)

Experimental setup. The GQA dataset [33] is a VQA dataset built with images of the Visual Genome project [41] and questions generated from the scene graphs of these images. The questions are generated from a large number of templates and hand-coded rules, such that they are of high linguistic quality and variety. We present experiments that the annotations of "equivalent questions" that are provided

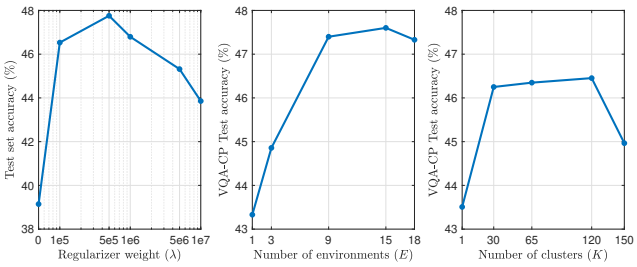

Figure 3. Sensitivity to hyperparameters on VQA-CP, using environments built from question groups (left and middle) or by clustering questions (right). See discussion in Section 4.1.

	VQA-CP v2	
	Val. set	Test set
	Other	Other
Baseline	54.74	43.33
Environments: random; rel. var., no alt. opt.	53.34	43.51
Environments: clustered questions; rel. var., no alt. opt.	54.10	46.35
Environments: question groups; rel. var., no alt. opt.	53.87	47.60
+ Alternating optimization (0 warm-up epoch)	54.00	47.71
+ Alternating optimization (2 warm-up epochs)	53.90	47.82
+ Alternating optimization (4 warm-up epochs)	53.98	**48.06**
+ Alternating optimization (6 warm-up epochs)	53.86	47.38
Without variance regularizer	40.76	39.14
With absolute variance regularizer	51.44	46.17

Table 1. Ablative study on VQA-CP (accuracy in percent, training on 'Other' questions only). Our method brings a significant gain over the baseline, both with environments built using the ground truth question types, and with environments built by unsupervised clustering of the questions. As a sanity check, we run the method with random environments, which gives results essentially identical to the baseline, as expected. The alternating optimization scheme brings a additional small improvement, although it is not crucial to the success of the method.

with the dataset. These annotations are not used in any existing model, to our knowledge. A small fraction of training questions (∼17.4% in the balanced training set) are annotated with up to three alternative forms. They involve a different word order or represent a different way of asking about a same thing. For example:

– *Is there a fence in the scene ?*
Do you see a fence ?
– *Which size is the green salad, small or large ?*
Does the green salad look large or small ?
– *Are there airplanes or cars ?*
Are there any cars or airplanes in this photo ?

Some alternative forms are already part of the dataset as other training questions, others are not. The straightforward way to use these annotations is by data augmentation, *i.e.* aggregating the equivalent forms with original training set.

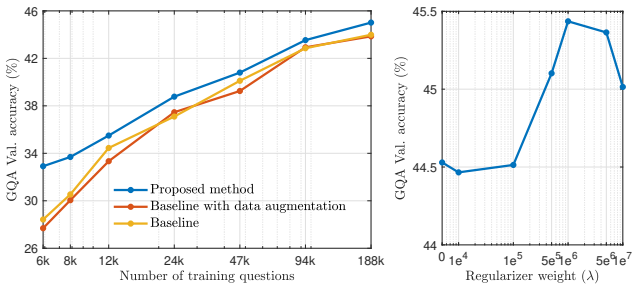

Figure 4. Experiments on GQA using equivalent questions to build environments. Our method provides consistent gains over the baseline, especially in the low-data regime. The improvement diminishes as more data is available, and is essentially imperceptible when the model is training on the full 2M training examples (∼10× than shown on this plot). A naive use of the equivalent questions for data augmentation has a negative effect because it shifts the distribution of the training set away from the test set.

Training environments with equivalent questions. We use our method to help learning invariance to the linguistic patterns of equivalent questions. We use E=4 environments where we replace, in each, a question by its e^{th} equivalent form if available, or the original form otherwise. Each environment thus contains a single form of each question.

Results. We compare in Fig. 4a the accuracy of our method with same model trained on the standard training set, and with the data augmentation baseline described above. The data augmentation does not help despite the additional training examples, because it modifies the distribution of training examples away from the distribution of test questions. Our method, in comparison, brings a clear improvement. For a fair comparison, we made sure that the data augmentation uses the exact same questions (original and equivalent forms) in every mini-batch, such that the improvement is strictly brought on by the architectural differences of our method. The improvement with our method is greatest with low amounts of training data (we use random subsets of the full training set). The full dataset provides a massive 14M examples (about 1M in its balanced version), at which point the impact of our method is imperceptible. The training set then essentially covers the variety of linguistic forms and concepts exhaustively enough such that there is no benefit from the additional annotations.

It is worth noting that all improvements brought by our method come from only a small fraction of questions being annotated with equivalent forms. It would therefore be realistic to annotate a real VQA dataset with similar equivalent forms, and investigate possible gains with our method, which we hope to do in the future.

In Fig. 4b, we plot the accuracy as a function of the regularizer weight. The clear optimum confirms again that the regularizer is a crucial component of the method.

	VQA-CP v2, Test set				VQA v2, Validation set			
	Overall	Yes/no	Numbers	Other↓	Overall	Yes/no	Numbers	Other
SAN [69]	24.96	38.35	11.14	21.74	52.02	–	–	–
GVQA [2]	31.30	57.99	13.68	22.14	48.24	–	–	–
Ramakrishnan *et al*., 2018 [53]	42.04	65.49	15.87	36.60	62.75	79.84	42.35	55.16
Grand and Belinkov, 2019 [26]	42.33	59.74	14.78	40.76	51.92	–	–	–
RUBi [12]	47.11 ±0.51	68.65	20.28	43.18	61.16	–	–	–
Teney *et al*., 2019 [63]	46.00	58.24	29.49	44.33	–	–	–	–
Product of experts [16]	40.04	43.39	12.32	45.89	63.21	81.02	42.30	55.20
Clark *et al*., 2019 [16]	**52.01**	**72.58**	**31.12**	46.97	56.35	65.06	37.63	54.69
Our baseline model	37.87 ±0.24	41.62	10.87	44.02	61.09 ±0.26	80.23	42.25	53.97
Proposed method	42.39 ±1.32	47.72	14.43	**47.24**	61.08 ±0.12	78.32	42.16	52.81
Our baseline model (×4 ensemble)	39.30	40.72	11.18	46.44	64.26	82.07	44.56	56.33
Proposed method (×4 ensemble)	43.37	47.82	14.35	**49.18**	63.47	81.99	43.07	55.21

Table 2. Comparison with existing methods designed to improve generalization on VQA-CP (accuracy in percents). The evaluation on 'yes/no' and 'number' questions is highly unreliable (see Section 4.1 and [62]). On the 'Other' questions however, our method surpasses all others. Our improvements on VQA-CP come only with a slight decrease in performance when trained and evaluated on the standard splits of VQA v2 (right columns). Reassuringly, the benefits of our method are cumulative with those of an ensemble (obtained by averaging the predictions of four models trained independently). The proposed method evaluated here uses environments built with question groups, E=15 environments, the relative variance regularizer, and no alternating optimization.

4.3. Multi-dataset training (VQA v2 and VG QA)

Experimental setup. These experiments apply our method to the training of a model on multiple datasets simultaneously. The VQA v2 dataset has previously been aggregated with Visual Genome QA (VG) [41] as a simple way to use more training data. The datasets contain similar types of questions, but it is reasonable to assume that they have slightly different distributions. We use E=2 environments, the first one containing the VQA v2 training data, the second the VG data.

Results. In Table 3, we compare our method with a model trained on VQA v2, another trained on VG, and one trained on the aggregation of the two datasets. The improvement is small but was verified over multiple training runs. We also ruled out explanation of the improvement as merely an ensembling effect, by comparing an ensemble of the baseline with one of the proposed method. The benefits of our method are cumulative with those of an ensemble, which suggests that our method should also apply to higher-capacity models. A number of such models have been described with a higher performance on VQA v2 [14, 22, 23, 44, 45, 57, 70] and it will be interesting to combine them with our method in the future.

	VQA v2, Validation set					VG
	Overall	Overall	Yes/no	Numbers	Other	Val.
	Ens. ×4	Single model				
Baseline model						
Trained on VQA v2	64.86	63.07 ±0.23	81.40	42.09	54.21	49.67
Trained on VG	28.48	27.58 ±0.22	0.11	36.03	47.11	60.17
Trained on Aggregated data	65.47	63.32 ±0.35	**82.27**	40.99	**55.98**	**61.20**
Proposed method						
Without variance reg.	64.33	62.18 ±0.27	78.95	41.68	54.42	59.68
With variance reg.	**65.73**	**63.80** ±0.17	81.00	**42.35**	55.97	60.54

Table 3. Multi-dataset training with VQA v2 and Visual Genome. The standard practice is to aggregate the two datasets. Our method treats them as two distinct training environments. The improvement is very small, but it comes at zero extra cost, and it was verified over multiple runs (mean and standard deviation are reported), as well as in an ensemble (first column). It was also verified on two different implementations of the baseline model (not in table).

5. Conclusions

We presented a method to train a deep models to better capture the mechanism of a task of interest, rather than blindly absorbing all statistical patterns from a training set. The method is based on the identification of correlations that are stable across multiple training environments, *i.e.* subsets of the training data. We described several strategies to build these environments using different forms of prior knowledge and auxiliary annotations. We showed benefits in various conditions including out-of-distribution test data, low-data training, and multi-dataset training.

An exciting challenge in computer vision is to design models solving tasks rather than datasets. Our strong results on VQA, which is known for its challenges in generalization and data scarcity, give us confidence that suitable tools like this method are emerging to make progress in this direction.

The proposed method is related to a series of works on IRM that appeared concurrently with the preparation of this paper [8, 3, 13, 15, 34]. Much remains to be done in terms of theoretical and empirical comparisons of these methods and their application to real (non-toy) data beyond VQA.

References

[1] E. Adeli, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, L. Fei-Fei, J. C. Niebles, and K. M. Pohl. Bias-resilient neural network. *arXiv preprint arXiv:1910.03676*, 2019. 1, 3

[2] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4971–4980, 2018. 1, 2, 3, 5, 6, 8, 12

[3] K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar. Invariant risk minimization games. *arXiv preprint arXiv:2002.04692*, 2020. 3, 8

[4] A. R. Akula, S. Gella, Y. Al-Onaizan, S.-C. Zhu, and S. Reddy. Words aren't enough, their order matters: On the robustness of grounding visual referring expressions. *arXiv preprint arXiv:2005.01655*, 2020. 1, 2

[5] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. *CVPR*, 2018. 12

[6] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018. 1

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 1

[8] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2, 3, 4, 5, 8

[9] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 9448–9458, 2019. 1, 2

[10] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *arXiv preprint arXiv:1705.08841*, 2017. 3

[11] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug 1996. 3

[12] R. Cadene, C. Dancette, H. Ben-younes, M. Cord, and D. Parikh. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, 2019. 2, 3, 5, 8

[13] S. Chang, Y. Zhang, M. Yu, and T. S. Jaakkola. Invariant rationalization. *arXiv preprint arXiv:2003.09772*, 2020. 3, 8

[14] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 5, 8

[15] Y. J. Choe, J. Ham, and K. Park. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020. 3, 8

[16] C. Clark, M. Yatskar, and L. Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019. 2, 3, 5, 6, 8, 12

[17] E. Creager, J.-H. Jacobsen, and R. Zemel. Environment inference for invariant learning. In *ICML Workshop on Uncertainty and Robustness*, 2020. 3

[18] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *CVPR*, 2017. 1

[19] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004. 1, 3

[20] S. Feng, E. Wallace, and J. Boyd-Graber. Misleading failures of partial-input baselines. *arXiv preprint arXiv:1905.05778*, 2019. 1

[21] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[22] P. Gao, Z. Jiang, H. You, P. Lu, S. C. Hoi, X. Wang, and H. Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6639–6648, 2019. 5, 8

[23] P. Gao, H. You, Z. Zhang, X. Wang, and H. Li. Multi-modality latent interaction network for visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5825–5835, 2019. 5, 8

[24] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. *arXiv preprint arXiv:1612.00837*, 2016. 2, 5, 6

[25] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 1, 2

[26] G. Grand and Y. Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. *arXiv preprint arXiv:1906.08430*, 2019. 1, 2, 3, 5, 6, 8

[27] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 2

[28] Y. Guo, Z. Cheng, L. Nie, Y. Liu, Y. Wang, and M. Kankanhalli. Quantifying and alleviating the language prior problem in visual question answering. *arXiv preprint arXiv:1905.04877*, 2019. 1, 2, 3, 5

[29] C. Heinze-Deml and N. Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017. 3, 4

[30] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision*, pages 793–811. Springer, 2018. 3

[31] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. Language-conditioned graph networks for relational reasoning. *arXiv preprint arXiv:1905.04405*, 2019. 5

[32] D. A. Hudson and C. D. Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018. 5

[33] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. 2, 3, 5, 6, 13

[34] D. Idnani and J. C. Kao. Learning robust representations with score invariant learning. 3, 8

[35] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. 2016. 2

[36] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017. 1, 2

[37] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*, 2016. 3

[38] D. Kaushik, E. Hovy, and Z. C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019. 4

[39] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217, 2020. 5

[40] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *Proc. Eur. Conf. Comp. Vis.*, 2012. 1, 2

[41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 2, 5, 6, 8

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2012. 2

[43] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 1, 2

[44] G. Li, N. Duan, Y. Fang, D. Jiang, and M. Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019. 5, 8

[45] B. Liu, Z. Huang, Z. Zeng, Z. Chen, and J. Fu. Learning rich image region representation for visual question answering. *arXiv preprint arXiv:1910.13077*, 2019. 5, 8

[46] R. K. Mahabadi and J. Henderson. Simple but effective techniques to reduce biases. *arXiv preprint arXiv:1909.06321*, 2019. 2, 3, 5

[47] T. M. Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . . , 1980. 1

[48] S. Pashami, A. Holst, J. Bae, and S. Nowaczyk. Causal discovery using clusters from observational data. In *FAIM'18 Workshop on CausalML, Stockholm, Sweden, July 15, 2018*, 2018. 3

[49] J. Pearl. *Causality: models, reasoning and inference*, volume 29. Springer, 2000. 5

[50] J. Pennington, R. Socher, and C. Manning. Glove: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014. 12

[51] J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *arXiv preprint arXiv:1501.01332*, 2015. 4

[52] J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference*. The MIT Press, 2017. 5

[53] S. Ramakrishnan, A. Agrawal, and S. Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, pages 1541–1551, 2018. 2, 3, 5, 8

[54] E. Sgouritsa, D. Janzing, J. Peters, and B. Schölkopf. Identifying finite mixtures of nonparametric product distributions and causal inference of confounders. *arXiv preprint arXiv:1309.6860*, 2013. 3

[55] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2020. 2

[56] P. Sorrenson, C. Rother, and U. Köthe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *arXiv preprint arXiv:2001.04872*, 2020. 5

[57] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 5, 8

[58] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987. 2

[59] D. Teney, E. Abbasnejad, and A. van den Hengel. On incorporating semantic prior knowledge in deep learning through embedding-space constraints. *arXiv preprint arXiv:1909.13471*, 2019. 3

[60] D. Teney, E. Abbasnejad, and A. van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. *arXiv preprint arXiv:2004.09034*, 2020. 3, 4

[61] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *CVPR*, 2018. 2, 5, 12

[62] D. Teney, K. Kafle, R. Shrestha, E. Abbasnejad, C. Kanan, and A. van den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. In *Proc. Advances in Neural Inf. Process. Syst.*, 2020. 3, 5, 6, 8

[63] D. Teney and A. van den Hengel. Actively seeking and learning from live data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[64] A. Torralba, A. A. Efros, et al. Unbiased look at dataset bias. In *CVPR*, volume 1, page 7, 2011. 1

[65] V. Vapnik. Statistical learning theory. john wiley&sons. *Inc., New York*, 1998. 1

[66] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. Eur. Conf. Comp. Vis.* Springer, 2002. 3

[67] H. Wang, S. Ge, Z. Lipton, and E. P. Xing. Learning robust global representations by penalizing local predictive power. In *Proc. Advances in Neural Inf. Process. Syst.*, 2019. 2

[68] T. Wang, J. Zhao, K.-W. Chang, M. Yatskar, and V. Ordonez. Adversarial removal of gender from deep image representations. *arXiv preprint arXiv:1811.08489*, 2018. 3

[69] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked Attention Networks for Image Question Answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 8

[70] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6281–6290, 2019. 5, 8

[71] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018. 2

[72] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 2

[73] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017. 3

[74] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012. 3

Supplementary material

A. Implementation of the VQA model

The VQA model used in our experiment follows the general description of Teney *et al*. [61]. We use the "bottom-up attention" features [5] of size 36×2048, pre-extracted and provided by Anderson *et al*.[5] The non-linear operations in the network use gated hyperbolic tangent units. The word embeddings are initialized as GloVe vectors [50] of dimension 300, then optimized with the same learning rate as other weights of the network. All activations except the word embeddings and their average are of dimension 512. The answer candidates are those appearing at least 20 times in the VQA v2 training set, *i.e*. a set of about 2000 answers. The output of the network is passed through a logistic function to produce scores in $[0, 1]$. The final classifier is trained from a random initialization. The model is trained with backpropagating a binary cross-entropy loss, and updating all weights with AdaDelta.

We use early stopping in all experiments to prevent overfitting. When using a distinct validation and test set, we report the accuracy on the test set, at the epoch of highest accuracy on the validation set.

B. Implementation of the proposed method

In our experiments with VQA-CP, the environments are built using either the ground truth question types, or an unsupervised clustering of the training questions. In the latter case, we use the k-means algorithm on a bag-of-words representations of questions. These representations are binary vectors whose length is equal to the size of the vocabulary of words that appear $\geq$10 times in the training set. Each component of the vector is equal to one if the corresponding word is present in the question, or zero otherwise. The clustering algorithm uses the cosine similarity as a metric. We also experimented with clustering representation of the questions made of their average GloVE embeddings [50] but the results were slightly worse.

The alternating optimization scheme showed a slight improvement in accuracy on VQA-CP. However, it brings another tunable hyperparameter (the number of warm-up epochs). We did not use it in most experiments because of the low potential return compared to the added expense in compute for tuning this hyperparameter. We have not verified whether the improvement holds on datasets other than VQA-CP.

C. Additional experiments and negative results

This section provides some insights on the timeline of the experiments presented in the paper, and of others that brought negative results.

Our initial, most encouraging results were obtained with VQA-CP, using the ground truth annotations of question types. The question types are known to be spuriously correlated with the answers across the training and test sets of VQA-CP, by construction of the dataset [2]. The use of this very fact is specific to the VQA-CP dataset, and it somewhat defeats the very objective of VQA-CP of encouraging generalizable models. Other recent works have used these annotations however [16], so it seemed fair game to do so as well. Nonetheless, we wanted to demonstrate a more general usage of our method that did not rely on these annotations. We experimented with various strategies to build environments by clustering the training data. The one presented in the paper simply uses the questions, which essentially approximates the labeling of the question groups. We tested other strategies, all of which proved unsuccessful, both on in- and out-of-distribution test data. We tried to cluster the training data based on the answers, the question words, the image features, and all combinations thereof.

With the GQA dataset, we experimented with using two environments, where we would sample, in the first, from the standard balanced training set, and in the second, from the larger unbalanced training set. The accuracy did however decrease on the standard balanced validation set.

The experiments we considered for this paper focused on VQA, but we believe there are a lot of possible other applications worth exploring, well beyond tasks in vision-and-language.

At test time, we use the average of the classifier weights learned across the training environments. We tried other strategies, such as using the median values, but the difference was insignificant. The variance regularizer already brings the weights to very similar values across environments.

[5] https://github.com/peteanderson80/bottom-up-attention

Table 4. Accuracy per question type on the GQA dataset [33]. Almost all categories benefit equally from the proposed method.

	Overall	Verify	Query	Choose	Logical	Compare	Object	Attribute	Category	Rel.	Global
With 6k Training examples (leftmost points on Fig. 4a)											
Baseline	28.42	50.00	14.61	22.67	**51.58**	45.67	**52.31**	32.84	17.93	23.02	23.57
Baseline with data augmentation	27.69	51.07	13.48	23.91	49.03	44.31	47.17	33.03	15.32	22.60	17.20
Proposed method	**32.91**	**52.26**	**20.89**	**29.50**	50.42	**50.76**	**52.31**	**37.64**	**25.85**	**26.91**	**35.03**
With 188k Training examples (rightmost points on Fig. 4a)											
Baseline	43.99	60.48	32.73	52.97	57.68	**51.95**	67.87	47.05	37.86	38.56	**52.87**
Baseline with data augmentation	43.85	60.17	32.74	52.52	**58.18**	49.41	**69.67**	46.97	37.68	38.18	49.68
Proposed method	**45.01**	**61.55**	**34.15**	**55.18**	57.74	48.90	68.64	**48.28**	**41.60**	**38.97**	49.04