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

The inability to generalize to test data outside the distribution of a training set is at the core of practical limits of machine learning. We show that mixing and shuffling training examples when training deep neural networks is not an optimal practice. On the opposite, partitioning the training data into non-i.i.d. subsets can guide the model to use reliable statistical patterns, while ignoring spurious correlations in the training data. We demonstrate multiple use cases where these subsets are built using unsupervised clustering, prior knowledge, or other meta-data available in existing datasets. The approach is supported by recent results on a causal view of generalization, it is simple to apply, and demonstrably improves generalization. Applied to the task of visual question answering, we obtain state-of-the-art performance on VQA-CP. We also show improvement over data augmentation using equivalent questions on GQA. Finally, we show a small improvement when training a model simultaneously on VQA v2 and Visual Genome, treating them as two distinct environments rather than one aggregated training set.

1. Introduction

The best of machine learning models can sometimes be right for the wrong reasons [2, 19, 14, 47]. A deep neural network trained for maximum likelihood on a given training set will reflect all statistical patterns present in the data. However, not all of these patterns may hold on the test data, limiting the generalization capabilities of the model. This paper presents a training paradigm that improves generalization to out-of-distribution data. This issue is critical from both theoretical and practical aspects, although it is often eclipsed when evaluating models on test data drawn from the same distribution as the training data [48]. Generalization is also manageable on tasks that are simple enough, or on domains that are reasonably circumscribed (*e.g.* classification of ImageNet-type photographs). As the task of interest grows more complex, for example in vision-and-language [40, 45, 46, 50], the combinatorial explosion in the input domain makes it impossible to process enough training data to span the domain. On tasks that require long chains of reasoning, the likelihood increases that spurious correlations in the training data will overshadow those that reflect the true reasoning process that underlies the task [27]. The demonstrations in this paper focus on visual question answering (VQA) as it is a prime example of these issues.

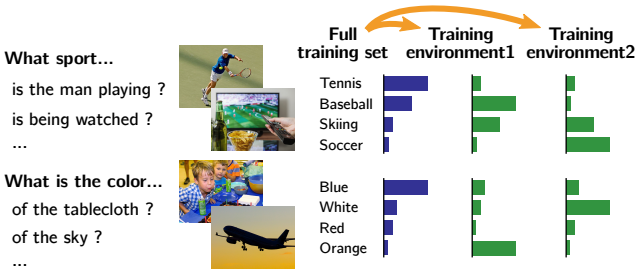

Figure 1. A dataset of labeled examples often contain biases and spurious correlations. For example in visual question answering, the first few words of a question are associated with a peaky distribution over answers (blue histograms). It is easy for a model to guess the answer by using such correlations, but the model will generalize poorly. We show how to improve generalization by partitioning a dataset into multiple training environments, in which the spurious correlations vary (green histograms) and only the reliable ones are invariant across environments. Our method trains a model to rely on these invariant correlations to make it generalize much better at test time.

The study of generalization has a long history in computer vision [34, 10]. The growing popularity of high-level tasks like VQA [5], but also visual dialogue [13], or vision-and-language navigation [4] for example, have brought the issue to the forefront as it stands clearly in the way of progress, more than it did for classical perceptual tasks. In VQA for example, the collection of training images, questions, and answers, brings out biases resulting from a propensity of annotators to focus on certain visual ele-

ments or propose questions about particular concepts. As a consequence, the model trained with a naive supervised approach have been shown to be overly reliant on the presence of particular words in a question. To develop a VQA system useful beyond a particular dataset and group of annotators, one must ensure that the model relies on statistical patterns that are not specific to one particular data collection process. This objective remains a major challenge, and multiple datasets [2, 26] and methods [8, 11, 20, 21, 33, 39] have recently been proposed to address the problem.

In this paper, we propose a general method to implicitly identify stable, invariant statistical patterns in a training set, and leverage them to train a robust predictive model. The resulting model is more likely to capture the underlying mechanisms of the task of interest, *i.e*. its actual causal structure, rather than superficial biases and spurious correlations present in a particular training set. In our method, we first partition the training set into multiple training environments, such that the spurious correlations vary, while only the reliable ones remain invariant across environments. We demonstrate multiple strategies to build such environments, using unsupervised clustering, prior knowledge, and by leveraging auxiliary annotations and meta-data from the data collection process. Second, we train multiple copies of a deep neural network, one in each environment. Some of its weights are shared across environments, while others are subject to a variance regularizer in the parameter space. Ultimately, the model learns to rely on the reliable, invariant correlations in the training data, while ignoring spurious correlations. This makes it able to generalize much better at test time.

Our approach follows a long line of work that aims to improve generalization and robustness in machine learning [21, 37, 41, 49]. Of particular relevance is the paradigm of invariant risk minimization (IRM) recently proposed by Arjovsky *et al*. [6]. IRM trains a model to capture invariances across multiple environments and improve generalization on out-of-distribution test data. The principle of IRM is to find a data representation such that the optimal classifier over this representation is identical in every environment (see Section 3.3). Our technical realization differs from that proposed in [6], but our work shows that the general idea of IRM brings substantial benefits to a range of use cases.

Our experiments focus on the task of VQA. We demonstrate three use cases with three different existing datasets. First, we improve resilience to language biases, and obtain state-of-the-art performance on the non-i.i.d. training/test splits of VQA-CP [2]. Second, on GQA [26], we demonstrate how to use annotations of equivalent questions (one question being a rephrasing of another). We obtain substantial gains over simple data augmentation in regimes with small amounts of training data. Third, we show a small benefit for training a model on multiple datasets, by treating the VQA v2 [19] and Visual Genome QA [29] datasets as two training environments rather than aggregating and shuffling the two training sets.

The contributions of this paper are summarized as follows.

1. We propose a method to implicitly identify stable and invariant correlations in a training set, and train a deep neural network that relies on these reliable patterns while ignoring spurious correlations.
2. We apply the method to three distinct use cases on the task of visual question answering: (1) resilience to language biases (*i.e*. leveraging prior knowledge of partial invariance to question words), (2) leveraging known relations of equivalence between specific training questions, and (3) multiple-dataset training.
3. We provide an extensive empirical study of the method and of its behaviour with respect to many hyperparameters and implementation choices. We obtain state-of-the-art performance on VQA-CP, and small but reliable improvements in particular training conditions on GQA and VQA v2.

2. Related work

There is a growing interest in building machine learning models resilient to **dataset biases**. Several popular datasets used in vision-and-language [18] and natural language processing [54] have been shown to exhibit strong biases. A model trained naively for maximum likelihood on these datasets can exhibit surprisingly good performance by capturing unreliable statistical patterns in the training set, without necessarily capturing the true mechanisms of the task. There is a trend toward evaluating models on out-of-distribution data [2, 54] to better identify this behaviour. Building models that generalize cannot be achieved by simply collecting more data from the same distribution, since it would contain the same unreliable patterns. Improving the data collection process can help [18, 54, 55] but it will only address precisely identified biases and confounders.

The general effectiveness of **data augmentation** [43] can be appreciated in light of generalization. The procedure essentially amounts to hard-coding known invariances in the input domain (*e.g*. geometrical transformation on an image [30]). Learning these invariances helps a model to ignore spurious correlations and to generalize better. It requires however a precise knowledge of the mechanisms of these invariances.The method in this paper, in comparison, allows leveraging invariances in the data implicitly, without attempting to produce their explicit definition.

The data used to train a model is usually considered as a collection of samples from a unique distribution. **Aggregating multiple datasets** to use more training data is not unusual (*e.g*. in [44] for VQA). But if datasets were collected in different conditions, we then loose valuable information.

Our method takes the opposed approach. We show that a suitable partitioning of the data can reveal which statistical correlations are reliable *vs* spurious.

The idea of training a model under multiple environments is reminiscent of **domain adaptation** [15]. Our objective is not to adapt to a particular new domain, but rather to learn a model that generalizes across a wide range of conditions. In domain adaptation, the idea of finding a data representation that is invariant across domains is limiting. This is because the true causal factors that we wish our model to rely on may differ in their distribution across training (see [6] for a formal discussion of these issues).

In our approach, we train multiple copies of a model in parallel, which is superficially similar to **ensembling** [57] and bootstrap aggregation a.k.a. bagging [7]. In an ensemble however, the models are combined in the space of their outputs. In our case, they are combined in parameter-space[1], and regularized during training in that space also. We show experimentally below that the improvements from this approach are distinct (and in fact complementary) to those of traditional ensembling. Bagging uses uniform sampling, whereas the point of our method is to exploit some prior knowledge for building the training environments.

Robustness in visual question answering is an increasingly popular concern, following the exposure of strong biases in existing datasets. New benchmarks have been designed to better study the issue [2, 26, 28]. 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 developed that brought substantial progress on VQA-CP (*e.g.* [8, 11, 20, 21, 33, 39]. In [12], the authors showed how to exploit additional annotations specific to the GQA dataset to make their model more robust.

Some of the above methods are related to **fair and bias-resilient machine learning** [1, 23, 51, 56]. The objective of the field is to build predictive models that are invariant to specific attributes of their inputs, such as gender or ethnicity. The attributes often to be specified and annotated, which is limiting for many applications. 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 like a discrete attribute.

The work that inspired the approach of this paper is the **invariant risk minimization** (IRM) by Arjovsky *et al.* [6]. The authors proposed to train a model under multiple training environments. They train a shared feature extractor such that a subsequent, environment-specific classifier is simultaneously optimal across all environments. The objective involves an expensive nested optimization, so they derive a practical objective that involves the magnitude of the gradient of the loss with respect to the classifier. This objective is still highly non-convex, and it proved difficult to use in our early investigations. In this paper, we describe a different realization of the same general principle of IRM. We use a simple variance regularizer to encourage the classifiers trained across environments to converge to a common optimum. We demonstrate applications on real large-scale datasets for multiple practical use cases, whereas [6] was limited to a toy example.

[1] In the case of a purely linear output, averaging the predictions, or the weights of the final layer is mathematically equivalent. In our case, the last linear is followed by a non-linearity, in which case the equivalence does not hold.

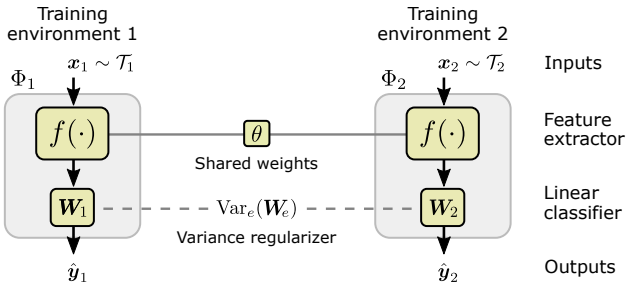

Figure 2. During training, we optimize a different copy of the model under each environment (each copy only sees a different subset of training data). Two environments are pictured, although our experiments use up to 18. The objective is to make the model rely on statistical patterns that are invariant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

Other relevant recent works include learning using statistical invariants [49] and [22]. In the latter, the authors use a variance regularizer on the predictions of a model trained on multiple versions of training examples, such as multiple photos of a same individual. Although superficially similar, the variance regularizer in our approach is on the parameters of the model rather than its predictions, and we do not require correspondences between specific training examples across environments.

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 might be possible to partition the data so that the reliable ones are more uniformly distributed than the spurious ones. We train a model to rely on the correlations that are common to all of the training environments. The corollary is that it ignores the environment-specific spurious correlations.

For example, a reliable correlation in VQA could be the presence of a dog in an image (supposing the question *What*

is the animal in the picture ?) and the answer dog. An unreliable correlation could be between questions starting with *What sport...* and the answer *tennis*, irrespective of the contents of the image. This particular unreliable correlation is a consequence of the data collection process, because a large amount of photos of tennis games were available, or because tennis was the first sport that would spring to the mind of annotators. It is conceivable however, that in non-i.i.d. subsets of the training data, for using distinct clusters in the input space, these unreliable correlations will vary in strength, vanish, or be replaced by other ones. The key to our approach is then to identifying which correlations remain stable across these subsets (which we call "training environments").

Concretely, we propose to partition a given 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 so as to isolate the effect of spurious correlations, while the reliable correlations remain roughly evenly distributed, because they reflect the underlying truth about the data. We provide in Section 3.3 additional justification for this principle, and we describe in Section 4 multiple strategies to build environments from existing datasets. We show that environments can be built by unsupervised clustering, by injecting prior knowledge, and by leveraging auxiliary annotations and meta-data from the data collection process. Next, we describe how to train a model across these 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 inputs $\boldsymbol{x}$ to a prediction $\hat{\boldsymbol{y}} = \Phi(\boldsymbol{x})$ 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 $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{h}$ uses parameters $\boldsymbol{\theta}$ and extracts a feature vector $\boldsymbol{h}$, typically with a deep neural network. The subsequent classifier is simply a matrix of weights $\mathbf{W}$, such that the whole model is $\Phi(\boldsymbol{x}) = \mathbf{W} f_{\boldsymbol{\theta}}(\boldsymbol{x})$. 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 the prior definition of training environments $\mathcal{T}_e$, we train the model Φ such that it is highly predictive across these environments, as well as on the test set, where we assume that only the input/output correlations *common to all training environments* will hold. For this purpose, we train a different $\Phi_e(\boldsymbol{x}) = \mathbf{W}_e f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for each environment. The feature extractor $f_{\boldsymbol{\theta}}(\cdot)$ is shared, such that it identifies features common to all environments (see additional justifications in Section 3.3). A different matrix of classifier weights $\mathbf{W}_e$ is optimized for each environment. To ensure that the features extracted by $f_{\boldsymbol{\theta}}(\cdot)$ are also *stable* across environments, we add a variance regularizer on the parameters of the classifiers $\mathbf{W}_e$, encouraging them to converge toward a common value.

At test time, we must essentially apply the model to a new, unknown environment for which we do not have a corresponding $\mathbf{W}_e$. We then use $\Phi^\star(\boldsymbol{x}) = \bar{\mathbf{W}} f(\boldsymbol{x})$, where $\bar{\mathbf{W}}$ is the arithmetic mean of $\mathbf{W}_e$ over $e = 1..E$. Even though the distances in the parameter space are difficult to interpret, the variance regularizer brings all vectors $\mathbf{W}_e$ to a similar value upon convergence of the whole model, such that using the arithmetic average is rational and practically effective. 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 proves difficult because the magnitude of the weights varies widely during the early stages of the optimization. As a remedy, we use an alternative definition of the variance, where the weights are scaled by the inverse of their average magnitude:

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

We refer to this definition as the "relative variance" in our experiments. It gives slightly better results and makes the optimal λ easier to find and more stable across environments.

We found a small empirical advantage in optimizing Eq. 2 with alternating updates. We use one mini-batch to update $\boldsymbol{\theta}$, then one to update $\mathbf{W}$, alternatively until convergence. This scheme slightly improves the final accuracy, but it is not crucial to the success of the method. It is only used in a few select experiments (see Table 1).

3.3. Why it works

Invariant risk minimization. Our training procedure was designed to approximate the objective of invariant risk minimization (IRM) [6]. In summary, the principle of IRM is to identify a representation of data such that the optimal classifier, on top of this representation, is identical in every environment. Formally, using our notations, this amounts to optimizing the feature extractor $f_{\boldsymbol{\theta}}(\cdot)$ and linear classi-

fier $\mathbf{W}$ 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.

The objective in Eq. 7 involves a nested optimization that is impractical to implement in practice. An approximation was proposed in [6] that replaces the constraint with a regularizer term in the objective that involves the gradient of the environment-specific risk with respect to the classifier. The resulting objective is highly convex and we were not able to apply it to any practical task. Our version (Eq. 2) rather uses the variance of $\mathbf{W}_e$ as a regularizer. The gradient of the risk in [6] is motivated as a measure of how optimal a classifier is. Our version operates directly in the parameter space of the classifier, which is dependent on other factors such as the magnitude of the activations of earlier layers. Consequently, our version does not provide all the guarantees discussed in [6] but it proved very stable to train and highly effective in our use cases.

The improvements in generalization brought by our method do not come out of thin air. The additional training signal comes from the information used to "unshuffle" the training data into multiple environments. If the environments are made as random partitions of the training data, no benefit is to be expected (as verified in our experiments, Table 1). In one of the use cases demonstrated in Section 4, we build environments by clustering chosen attributes of the data. This practice allows us to inject some of our prior knowledge about the task into the model. This brings us to a complementary interpretation of our technique trough the lens of causal reasoning.

Causal view of generalization. A robust model must essentially mirror the causal model of the task of interest. It must use, to produce its predictions, only the direct causal parents of the variable of interest. A spurious correlation shows up as a statistical dependence between the random variables representing the target (Y) and the input (X), such that an intervention on X does *not* change the distribution on Y: $P(Y|\,\mathrm{do}\,(X)) = P(Y)$ [35]. For example, imagine training a VQA model on question/answer examples from two datasets (Fig. 3). In the first, the annotators provided mostly short questions, with the most frequent answer being *yes*. In the second, they provided mostly longer questions, with the most frequent answer being *no*. There is a strong dependence between the answer (Y) and the length of the questions (a function of the input $\boldsymbol{x}$, which we represent as a latent random variable L). However, there is no causal relation between L and Y: reassigning a long question from the second dataset to the first dataset will not change its answer. The image however (another function of the input, represented as the variable I) is a direct causal factor for the answer Y, since intervening on the image will generally change the distribution of the answer, *i.e.* $P(Y|\,\mathrm{do}\,(I)) \neq P(Y)$. After suspecting this spurious correlation between L and Y, one could use our method and build environments where the joint distribution $P(Y, L)$ varies, *e.g.* by clustering the values of L while maintaining $P(A)$ similar. A model trained with our method will then learn to be invariant to the unreliable feature L.

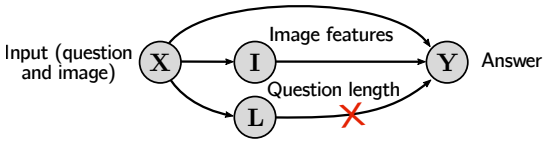

Figure 3. Causal model for the hypothetical example of Section 3.3. The question length is correlated with the answer, but L is not a causal factor of Y, making this a spurious correlation.

The identification of a causal model from purely observational data is known to be impossible outside of particular cases [35, 38]. What our method allows, however, is to inject prior knowledge about the causal structure of the task. It can be obtained from our own experience, from controlled experiments (*e.g.* data gathered in different conditions), or other task-specific knowledge. In contrast to most works on causal reasoning and causal discovery from the field of statistics, we are not interested in the causal model of the task per se. We only identify implicitly invariances that result from causal relations, and that can improve generalization of a predictive model.

4. Experiments

We implemented the method on top of the "bottom-up and top-down attention" model for VQA [44] (see supp. mat. for implementation details). It is well studied, relatively simple, and serves as the underlying model of many other techniques for bias-resilient learning [8, 11,

20, 21, 33, 39]. Our strongest results are with the VQA-CP dataset [2], which is designed to test out-of-distribution generalization. We present two other use cases, one with GQA [26], and another with VQA v2[18] combined with Visual Genome [29]. The quantitative improvements over baselines are smaller, but they demonstrate the wide applicability of the method. We have no reason to suspect a synergy between our method and the chosen VQA model, so it should readily apply to other recent models [9, 16, 17, 31, 32, 42, 53] including strong models designed for GQA [24, 25].

4.1. Robustness to language biases (VQA-CP)

Experimental setup The VQA-CP dataset [2] was constructed by reorganizing VQA v2 [18] 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 [46], 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. 4b). 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. 4c the accuracy of the model against the number of clusters K. There is a clear but 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. 4b 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 the ground truth types is the better strategy, and that the clustering essentially 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. 4a 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 "warm-up", during which we update all parameters 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*. [11] 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 method admittedly use the test set as a validation set (!) for hyperparameter selection and/or model selection [2, 20]. We rather hold out 8k instances from the training set to serve as a validation set. They serve for example to monitor training and determine the epoch for early stopping.

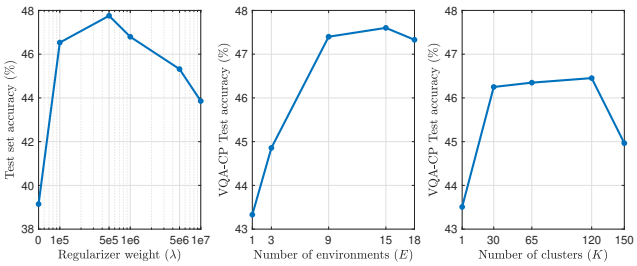

Figure 4. Sensitivity to hyperparameters on VQA-CP, using environments built from question groups (left and middle) or by clustering questions (right). See discussion in Section 4.1.

	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.

4.2. Invariance to equivalent questions (GQA)

Experimental setup The GQA dataset [26] is a VQA dataset built with images of the Visual Genome project [29] 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 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 of the 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.

Training environments with equivalent questions We use our method to help the model to learn 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 will thus use a single form of each training question.

Results We compare in Fig. 5a 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. 5b, we plot the accuracy as a function of the regularizer weight. We observe a clear optimum, which confirms again that the regularizer is a crucial component of the method.

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) [29] 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

	VQA-CP v2, Test set				VQA v2, Validation set			
	Overall	Yes/no	Numbers	Other↓	Overall	Yes/no	Numbers	Other
SAN [52]	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 [39]	42.04	65.49	15.87	36.60	62.75	79.84	42.35	55.16
Grand and Belinkov, 2019 [20]	42.33	59.74	14.78	40.76	51.92	–	–	–
RUBi [8]	47.11 ±0.51	68.65	20.28	43.18	61.16	–	–	–
Teney *et al.*, 2019 [46]	46.00	58.24	29.49	44.33	–	–	–	–
Product of experts [11]	40.04	43.39	12.32	45.89	63.21	81.02	42.30	55.20
Clark *et al.*, 2019 [11]	**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 [46]). 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.

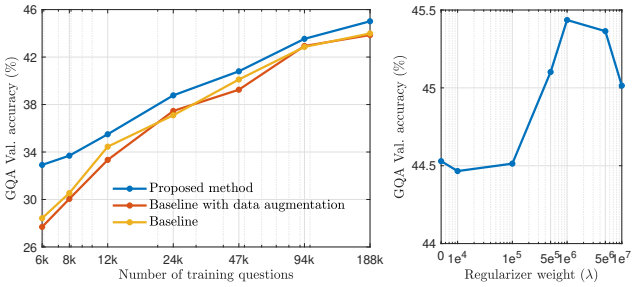

Figure 5. Experiments on GQA using equivalent questions to build environments. Our method provides consistent gains over the baseline, especially in the low-data regime. The improvement diminishes as more data is available, and is essentially imperceptible when the model is training on the full 2M training 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.

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 [9, 16, 17, 31, 32, 42, 53] 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 invariant across multiple training environments, *i.e.* subsets of the training data. We described several strategies to build these environments by 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.

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. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018. 1, 2, 3, 6, 8, 11

[3] 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. 11

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 1

[5] 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

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

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

[8] 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

[9] 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. 6, 8

[10] W. Chojnacki, M. J. Brooks, A. Van Den Hengel, and D. Gawley. On the fitting of surfaces to data with covariances. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1294–1303, 2000. 1

[11] 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, 11

[12] A. v. d. H. Damien Teney, Ehsan Abbasnejad. On incorporating semantic prior knowledge in deep learning through embedding-space constraints. *arXiv preprint arXiv:1909.13471*, 2019. 3

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

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

[15] 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. 3

[16] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6639–6648, 2019. 6, 8

[17] P. Gao, H. You, Z. Zhang, X. Wang, and H. Li. Multi-modality latent interaction network for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5825–5835, 2019. 6, 8

[18] 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, 6

[19] 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

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

[21] 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. 2, 3, 5

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

[23] 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

[24] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. Language-conditioned graph networks for relational reasoning. *arXiv preprint arXiv:1905.04405*, 2019. 6

[25] D. A. Hudson and C. D. Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018. 6

[26] 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, 6, 7, 12

[27] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. 2016. 1

[28] 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

[29] 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, 6, 7

[30] 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

[31] 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. 6, 8

[32] 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. 6, 8

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

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

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

[36] J. Pennington, R. Socher, and C. Manning. Glove: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014. 11

[37] J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 2

[38] J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1):2009–2053, 2014. 5

[39] 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

[40] S. K. Ramakrishnan, A. Pal, G. Sharma, and A. Mittal. An empirical evaluation of visual question answering for novel objects. *arXiv preprint arXiv:1704.02516*, 2017. 1

[41] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018. 2

[42] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 6, 8

[43] 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

[44] 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, 11

[45] D. Teney and A. van den Hengel. Visual question answering as a meta learning task. 2017. 1

[46] 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. 1, 6, 8

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

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

[49] V. Vapnik and R. Izmailov. Rethinking statistical learning theory: learning using statistical invariants. *Machine Learning*, 108(3):381–423, 2019. 2, 3

[50] S. Venugopalan, L. Anne Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. Captioning images with diverse objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5753–5761, 2017. 1

[51] 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

[52] 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

[53] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019. 6, 8

[54] 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

[55] 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

[56] 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

[57] 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*. [44]. We use the "bottom-up attention" features [3] of size 36×2048, pre-extracted and provided by Anderson *et al*.[2] The non-linear operations in the network use gated hyperbolic tangent units. The word embeddings are initialized as GloVe vectors [36] 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 [36] 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 [11], 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.

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

Table 4. Additional results (accuracy per question type) on the GQA dataset [26]. Most categories benefit similarly from the proposed method.

	Overall	Verify	Query	Choose	Logical	Compare	Object	Attribute	Category	Rel.	Global
With 6k Training examples (leftmost points on Fig. 5a)											
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. 5a)											
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