Adversarial Regularization for Visual Question Answering: Strengths, Shortcomings, and Side Effects

Gabriel Grand[1] and **Yonatan Belinkov**[1,2]
[1]Harvard John A. Paulson School of Engineering and Applied Sciences
[2]MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA, USA
ggrand@alumni.harvard.edu, belinkov@seas.harvard.edu

Abstract

Visual question answering (VQA) models have been shown to over-rely on linguistic biases in VQA datasets, answering questions "blindly" without considering visual context. Adversarial regularization (AdvReg) aims to address this issue via an adversary sub-network that encourages the main model to learn a bias-free representation of the question. In this work, we investigate the strengths and shortcomings of AdvReg with the goal of better understanding how it affects inference in VQA models. Despite achieving a new state-of-the-art on VQA-CP, we find that AdvReg yields several undesirable side-effects, including unstable gradients and sharply reduced performance on in-domain examples. We demonstrate that gradual introduction of regularization during training helps to alleviate, but not completely solve, these issues. Through error analyses, we observe that AdvReg improves generalization to binary questions, but impairs performance on questions with heterogeneous answer distributions. Qualitatively, we also find that regularized models tend to over-rely on visual features, while ignoring important linguistic cues in the question. Our results suggest that AdvReg requires further refinement before it can be considered a viable bias mitigation technique for VQA.

1 Introduction

In recent years, the Visual Question Answering (VQA) community has grown increasingly cognizant of the confounding role that bias plays in VQA research. Many popular VQA datasets have been shown to contain systematic language biases that enable models to cheat by answering questions "blindly" without considering visual context (Agrawal et al., 2016; Zhang et al., 2016; Goyal et al., 2017; Agrawal et al., 2018).

Efforts to address this problem have mainly focused on constructing more balanced datasets (Zhang et al., 2016; Goyal et al., 2017; Johnson et al., 2017; Chao et al., 2018). However, any benchmark that involves crowdsourced data is likely to encode certain cognitive and/or social biases (van Miltenburg, 2016; Misra et al., 2016; Eickhoff, 2018). An alternate approach is to develop models that can generalize to novel domains with different biases. In this spirit, Agrawal et al. (2018) introduced VQA under Changing Priors (VQA-CP), a new benchmark in which the distribution of answers varies significantly between train and test splits. Existing models, which tend to rely heavily on the distribution of answers in the training set, perform poorly on VQA-CP (Agrawal et al., 2018).

One approach to mitigating bias that has recently gained interest is a technique called adversarial regularization (AdvReg). In AdvReg, an adversary sub-network performs an inference task based on a subset of the input features; in this case, the adversary attempts to predict answers based only on the question. Successful performance by the adversary indicates that the main network has learned a biased input representation. Negated gradient updates from the adversary are backpropagated to a shared encoder to encourage the main network to learn a bias-neutral representation of the question. Recently, Ramakrishnan et al. (2018) applied AdvReg to VQA and found that it improves generalization to out-of-domain examples on VQA-CP test.

Despite this initial success, AdvReg is still a relatively new methodology, and its effects on representation learning in neural networks remain largely unknown. In this study, we explore AdvReg with the goal of better understanding how this technique affects inference in VQA models. We apply AdvReg to the Pythia VQA architec-

ture (Jiang et al., 2018b), achieving a new state-of-the-art on VQA-CP v1 and v2. However, we find that AdvReg yields a number of previously unreported and undesirable side-effects. We first observe that AdvReg introduces significant noise into gradient updates that creates instability during training. This finding motivates the introduction of a new scheduling technique that gradually introduces regularization over the course of training. We find that scheduling improves gradient stability in the early phases of adversarial training and improves performance on VQA-CP v2. However, even with scheduling, AdvReg significantly reduces performance on in-domain examples. This side-effect suggests that like many statistical regularization methods, AdvReg offers a trade-off between in-domain and out-of-domain performance.

To investigate the strengths and weaknesses of regularized models, we perform quantitative and qualitative error analyses. We find that AdvReg is especially helpful with Yes/No questions, but reduces performance on questions with heterogeneous answers. We also visualize a number of successes and failures of AdvReg, revealing that regularized models often ignore linguistic cues in the question and are heavily swayed by salient visual features. These findings suggest an under-utilization of key information in the question.

The contributions of this work are two-fold. First, we share practical tips for dealing with the idiosyncrasies of AdvReg. Second, we highlight some core drawbacks of AdvReg that have not previously been reported in the literature. By drawing attention to these shortcomings, we hope to motivate future efforts to refine AdvReg.

2 Related Work

Biases in VQA datasets A growing body of work points to the existence of biases in popular VQA datasets (Agrawal et al., 2016; Zhang et al., 2016; Jabri et al., 2016; Goyal et al., 2017; Johnson et al., 2017; Chao et al., 2018; Agrawal et al., 2018; Thomason et al., 2018). In VQA v1 (Antol et al., 2015), for instance, for questions of the form, "What sport is...?", the correct answer is "tennis" 41% of the time, and for questions beginning with "Do you see a...?" the correct answer is "yes" 87% of the time (Zhang et al., 2016). By exploiting these biases, models can disregard the image and still achieve high VQA scores.

Biases in other language tasks Language biases have also been reported in natural language inference (NLI) (Gururangan et al., 2018; Tsuchiya, 2018; Poliak et al., 2018), reading comprehension (Kaushik and Lipton, 2018), and story cloze completion (Schwartz et al., 2017). Many of these tasks are concerned with inferring the relationship between two objects. As in VQA, models can often succeed by learning biases associated with one of these objects, while ignoring the other.

Biases in other vision tasks Images can also encode certain associative biases. For instance, the Commmon Objects in Context (COCO) image dataset (Lin et al., 2014), which is used in VQA, has been shown to contain prominent gender biases (Zhao et al., 2017; Hendricks et al., 2018). Recently, Hendricks et al. (2018) introduced a technique that encourages the assignment of equal gender probability when gender information is occluded from an image. Their Appearance Confusion Loss can be viewed as a vision captioning analogue to AdvReg for VQA.

Mitigating bias Initial efforts to address bias in VQA focused on debiasing existing datasets. VQA v2 introduced complimentary examples with different answers to every question (Goyal et al., 2017). While VQA v2 resulted in a near 50/50 balance for Yes/No questions, the distribution for non-binary questions (e.g., "What type of...?"; "What sport is...?") remains skewed towards a handful of top answers (Goyal et al., 2017).

Given the difficulty of isolating bias from crowdsourced data, researchers have instead begun to emphasize generalization to new domains with different biases. In this line, Agrawal et al. (2018) introduced VQA-CP, a re-division of the existing VQA datasets in which the distribution of answers per question type is inverted between train and test splits. For instance, in the VQA-CP v1 train split, "tennis" is the most frequent answer for the question "What sport is...?", while "skiing" is very uncommon; in the test split, this prior is reversed. Most relevant to our work, Ramakrishnan et al. (2018) applied AdvReg to VQA-CP, and found that it improved test performance over a non-regularized model. Similarly, Belinkov et al. (2019) analyzed the effects of using AdvReg to address bias in NLI. In this work, we analyze the effects of AdvReg on VQA models in further detail, complement AdvReg with a scheduling scheme, and point to remaining limitations in its behavior.

3 Methods

3.1 Adversarial Regularization

Many modern VQA architectures adhere to a common modular design (Jiang et al., 2018b) consisting of the following four components:

- $f_v(I; \theta_v) : I \mapsto v$ Image encoder
- $f_q(Q; \theta_q) : Q \mapsto q$ Question encoder
- $f_z(v, q; \theta_z) : v, q \mapsto z$ Multimodal fusion
- $g_{\text{VQA}}(z; \theta_{\text{VQA}}) : z \mapsto P(a)$ Answer classifier

Composing these components, we obtain the following expression for the base VQA model. This model is trained to minimize cross entropy loss:[1]

$$P(a|I, Q) = g_{\text{VQA}}(f_z(f_v(I), f_q(Q))) \quad (1)$$

$$\mathcal{L}_{\text{VQA}} = -\sum_i a_i \log P(a_i|I, Q) \quad (2)$$

In AdvReg, we introduce an adversarial classifier $g_{\text{ADV}}(q; \theta_{\text{ADV}})$, which attempts to infer the correct answer from only the question features. g_{ADV} shares the same question feature extractor f_q as the base VQA model. However, f_q and g_{ADV} are separated by a gradient reversal layer (GRL). The GRL is a pseudo-function that negates gradients on the backward pass; otherwise, it leaves inputs unchanged:

$$\text{GRL}_\lambda(x) = x \qquad \frac{\partial \text{GRL}_\lambda}{\partial x} = -\lambda_{\text{GRL}} \quad (3)$$

where λ_{GRL} is a hyperparameter. As above, the adversary is trained to minimize the cross entropy loss $\mathcal{L}_{\text{ADV}}$:

$$P(a|Q) = g_{\text{ADV}}(\text{GRL}_\lambda(f_q(Q))) \quad (4)$$

$$\mathcal{L}_{\text{ADV}} = -\sum_i a_i \log P(a_i|Q) \quad (5)$$

The adversarial relationship between the main model and the adversary can be expressed as:

$$\min_{\theta_{v,q,z,\text{VQA}}} \max_{\theta_{q,\text{ADV}}} \mathcal{L} = \mathcal{L}_{\text{VQA}} - \lambda_{\text{ADV}}\mathcal{L}_{\text{ADV}} \quad (6)$$

where the regularization coefficient $\lambda_{\text{ADV}} \geq 0$ controls the trade-off between performance on VQA and robustness to language bias. Additionally, $\lambda_{\text{GRL}} \geq 0$ (from Eq. 3) scales the reversed

---

[1]Since the VQA evaluation metric includes ground truth answers from 10 different subjects, we follow the top-performing models in using a soft target, multi-label variant of the cross entropy objective (see Teney et al. 2018).
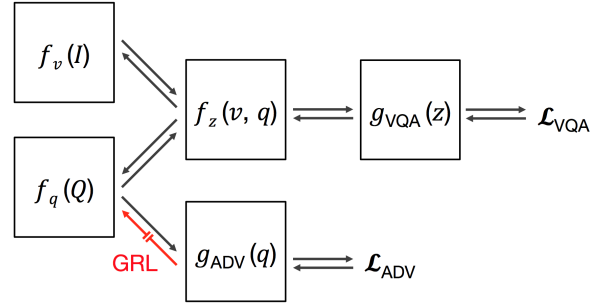

Figure 1: Schematic diagram of adversarial VQA architecture. Right and left arrows represent forward and backward propagation, respectively. The red arrow indicates the gradient reversal layer.

gradients. These two hyperparameters perform related, but different, functions. Setting either or both to zero disables the regularization, since f_q receives no gradients from the adversary. This combination is equivalent to the baseline model. Meanwhile, setting $\lambda_{\text{ADV}} > 0, \lambda_{\text{GRL}} > 0$ enables AdvReg. This setting is the main focus of our experiments.

3.2 Gradient Reversal Layer Scheduling

Because the GRL counteracts the main gradient updates, AdvReg produces noisy gradients that can interfere with learning, as we observe in the experiments below (Fig. 4). To improve stability during the early stages of training, we experiment with a scheduling regime for the gradient reversal layer similar to that used in domain-adversarial neural networks (Ganin et al., 2016). During training, we delay the introduction of regularization for the first μ iterations, which allows f_q to receive clean gradients from the VQA model. Next, we have a warmup phase for w iterations, in which we increase λ_{GRL} linearly from 0 to some constant c:

$$\lambda_{\text{GRL}}(t) = \begin{cases} 0 & t \leq \mu \\ \frac{c(t-\mu)}{w} & \mu \leq t \leq \mu + w \\ c & t > \mu + w \end{cases} \quad (7)$$

GRL scheduling introduces two new hyperparameters, μ and w, which we set by grid search; further details are given in Appendix A.2.

4 Experimental Setup

4.1 Data

We evaluated the performance of our AdvReg setup on VQA-CP v1 and v2 (Agrawal et al., 2018). We also retrained our best-performing models with the same hyperparameter settings on

Model	λ_{ADV}	λ_{GRL}	VQA-CP v1 (test)				VQA-CP v1 (val)				VQA v1 (val)
			Overall	Yes/No	Num.	Other	Overall	Yes/No	Num.	Other	Overall
Baseline	0	0	37.87	42.58	14.16	42.71	**65.79**	86.98	40.06	56.41	**62.68**
+ AdvReg	0.01	0.1	**45.69**	77.64	13.21	26.97	46.94	65.32	32.95	37.22	46.34
+ GRL Sch.	0.01	0.1	44.09	75.01	13.40	25.67	46.45	67.28	29.11	35.71	46.71
			VQA-CP v2 (test)				VQA-CP v2 (val)				VQA v2 (val)
Baseline	0	0	38.80	41.70	12.17	44.59	**67.76**	84.76	49.22	57.04	**63.27**
+ AdvReg	0.005	1	36.33	59.33	14.01	30.41	50.63	67.39	38.81	38.37	48.78
+ GRL Sch.	0.005	1	**42.33**	59.74	14.78	40.76	56.90	69.23	42.50	49.36	51.92

Table 1: Performance comparison of baseline and adversarially-trained models on VQA-CP/VQA v1 and v2 datasets using the best-performing hyperparameters.

VQA v1 (Antol et al., 2015) and v2 (Goyal et al., 2017) in order to evaluate performance on datasets without changing priors.

One difficulty of working with VQA-CP is the lack of validation sets. Ramakrishnan et al. (2018) explain that VQA-CP does not provide validation sets due to the difficulty in varying the answer distributions of binary questions across more than two splits. The authors note that, in place of early stopping, they train their models "until convergence."[2] Although the nonstandard structure of VQA-CP makes validation tricky, we believe it is important to have some mechanism to distinguish between overfitting to language priors and overfitting to the examples in the training set (the latter may occur regardless of the presence of language biases). Our solution is to train models on 90% of the training data and reserve the remaining 10% (sampled randomly) for validation. Score on the val split is useful as an early stopping metric, but does not forecast test performance. In this way, we are able to prevent our models from overfitting to the training data, while remaining agnostic to the distribution of priors in the test set.

While the addition of a VQA-CP val set enables early stopping, models that perform best on the val set will tend to be under-regularized, since AdvReg reduces in-domain performance. We considered creating a second val set derived from VQA-CP test for model selection. However, in addition to introducing additional complexity, this approach would both compromise our ability to remain agnostic to the test set and make our results incomparable with prior work. Therefore, we follow Ramakrishnan et al. (2018) and per-

form model selection on VQA-CP test. However, to increase transparency, we report results across a broad range of hyperparameters. We hope that recognition of these challenges will motivate the introduction of a standard val set for VQA-CP.

4.2 Implementation

Our experimental setup is based on the Pythia implementation of the Bottom-Up / Top-Down VQA model (Jiang et al., 2018a; Anderson et al., 2018).[3] The adversarial classifier g_{ADV} is implemented as a two-layer fully-connected network with 512 hidden units and ReLU activation. Unless otherwise noted, we use the default hyperparameters from Pythia. Additional details are available in Appendix A.1.

5 Results

5.1 Strengths of AdvReg

Table 1 summarizes the results of the baseline model and the best performing adversarially regularized models. On the VQA-CP v1 test set, our best AdvReg model outperforms the baseline by 7.82%, attaining a new state-of-the-art for this task. On the VQA-CP v2 test set, our best AdvReg model performs worse than the baseline; however, with GRL scheduling, it surpasses the baseline by 3.53%, again setting a new state-of-the-art. Note that in both cases, our models perform better than Ramakrishnan et al. (2018), who report scores of 43.43% and 41.17% on VQA-CP v1 and v2 test, despite the fact that we use only 90% of the available training data. This result indicates that allocating 10% for validation helps prevent overfitting to the training examples.

To highlight how AdvReg mitigates overfitting, Fig. 2 plots loss curves of the baseline (blue)

---

[2]In correspondence, the authors clarified that they trained for a fixed interval determined by the number of iterations to reach peak performance on VQA v2. Since overfitting tends to occur more rapidly on VQA-CP, we view an in-domain val split as a more reliable early stopping metric.

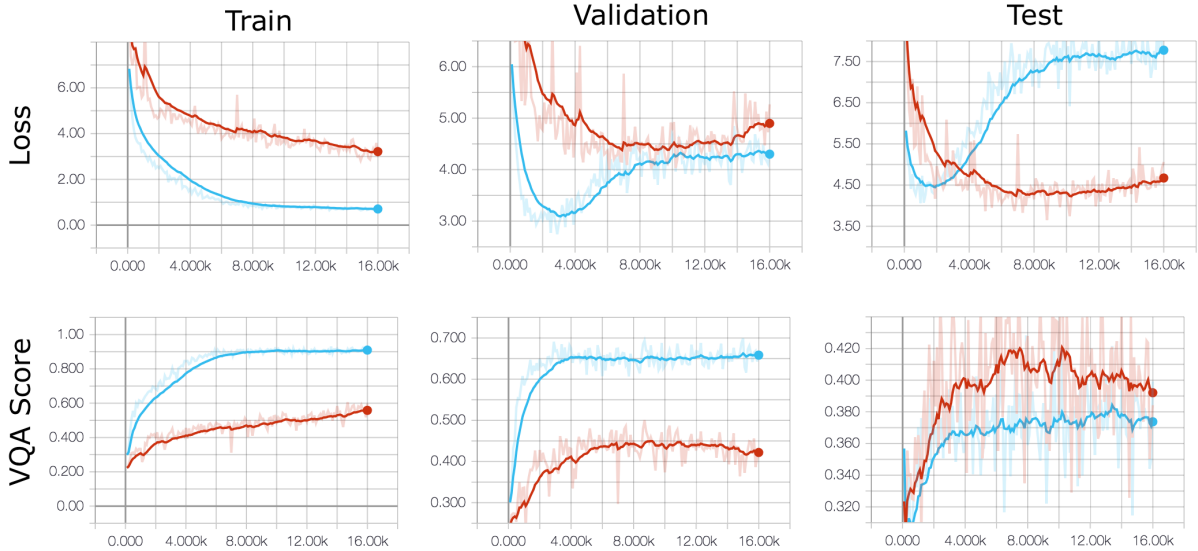[3]Our code is available at https://github.com/gabegrand/adversarial-vqa

Figure 2: Comparison of regularized (red) and baseline (blue) models on VQA-CP v1 train, val, and test. The baseline model exhibits severe overfitting on both the val and test splits. In contrast, the regularized model overfits much less and achieves a higher score on VQA-CP test.

and regularized (red) models during training. The baseline model exhibits severe overfitting on both VQA-CP v1 val and test. Note that overfitting on the test set appears around 2000 iterations as the model begins to over-rely on language priors. In contrast, overfitting on the val set appears later (around 3500 iterations) as the model begins to memorize the training examples.

In general, AdvReg works well out-of-box on VQA-CP v1. Many of the hyperparameter combinations we tested (Fig. 3) outperform the baseline on VQA-CP v1 test. The key to successful regularization appears to be balancing λ_{ADV} and λ_{GRL}. As Fig. 3 reveals, large values of λ_{ADV} perform better with small values of λ_{GRL}, and vice-versa. However, when λ_{ADV} is too small, AdvReg fails to improve performance; none of the models we tested with $\lambda_{ADV} = 0.001$ outperformed the baseline. On the other hand, when λ_{ADV} is too large, training becomes unstable; for $\lambda_{ADV} > 1$ (not shown), we observed many training runs failing to converge due to exploding gradient values.

5.2 Shortcomings of AdvReg

The improved performance on the out-of-domain test sets comes at the expense of performance on the in-domain validation sets. As Table 1 shows, on both VQA-CP v1 and v2 val, AdvReg models significantly under-performed the baseline (-18.85% and -10.66%, respectively). Retraining with the same hyperparameters on the original VQA v1 and v2 datasets yielded similar results.

Notably, these findings differ from Ramakrishnan et al. (2018), who report only minimal reductions in performance on VQA v1 and v2 from AdvReg. One explanation is that the gains we observed on VQA-CP test relative to Ramakrishnan et al. resulted in diminished performance on VQA-CP val. Indeed, across all runs of our experiments, we found that score on VQA-CP v1 test correlated negatively with score on the val split (r^2 = -0.355, p = 0.013).[4] In their work, Ramakrishnan et al. also introduce a secondary "difference of entropies" (DoE) regularizer, which they find improves in-domain performance and helps to stabilize adversarial training. However, even without DoE, they report margins of only 1-4% between their AdvReg and baseline models. Ultimately, these unaccounted differences may be due to implementation details, suggesting the need for a closer comparison.[5]

Our results also highlight interesting differences between VQA-CP v1 and v2. On VQA-CP test, the gains due to AdvReg were more significant on v1 as compared to v2. However, on the validation sets, the losses were also greater. This pattern also applied with respect to the original versions of these datasets (i.e., VQA v1 and v2). These findings support the notion that VQA v2 is indeed less biased than v1.

---

[4]We did not find a significant correlation between test and val performance on VQA-CP v2 (r^2 = 0.237, p = 0.141).

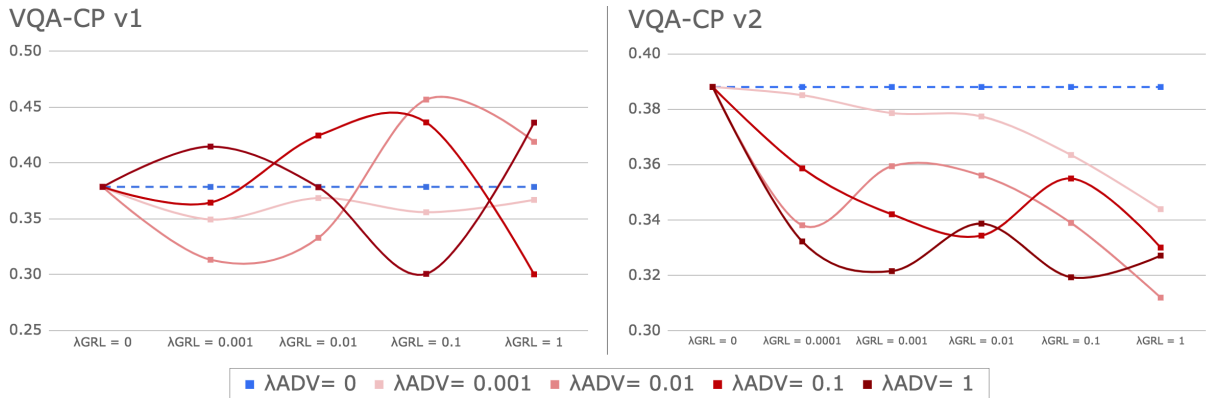[5]To our knowledge, code from (Ramakrishnan et al., 2018) is not public at present.

Figure 3: Hyperparameter sweep on VQA-CP v1 and v2 test. Each line is a different setting of λ_{ADV}; lighter/darker red indicates less/more regularization, respectively. λ_{GRL} is varied along the x-axis. Blue dashes: baseline score.

5.3 Effect of GRL scheduling

Without GRL scheduling, none of the AdvReg hyperparameter combinations we tested outperformed the baseline on VQA-CP v2 test (see Fig. 3). This finding may be attributed to the substantial amount of noise that the adversary injects into the gradient updates for the question encoder, as demonstrated by recording gradient norms throughout training.

As Fig. 4 illustrates, on VQA-CP v2, GRL scheduling reduces gradient instability early in training, allowing the model to converge to a lower loss value. In the best-performing schedule, regularization was delayed until $\mu = 2000$ iterations, and slowly warmed up for the following $w = 4000$ steps. This schedule resulted in a 6.00% performance increase on VQA-CP v2 test compared to using the same regularization coefficients without GRL scheduling, and a 3.53% improvement over the baseline (see Table 1).

On VQA-CP v1, we did not observe commensurate improvements from GRL scheduling. We hypothesize that introducing AdvReg on a delay may not be as effective on v1 due to the more prominent biases in this dataset. Note that the baseline model begins to overfit roughly twice as quickly on VQA-CP v1 as on VQA-CP v2 (Fig. 4, Baseline loss). Accordingly, in addition to sweeping the same hyperparameters tested on VQA-CP v2, we experimented with accelerated GRL schedules for VQA-CP v1. While five of the runs outperformed the baseline, three of these were with no start delay. Moreover, all of the runs with GRL scheduling performed worse than a model with the same regularization coefficients with static λ_{GRL}. Finally, many of the runs on VQA-CP v1, and es-
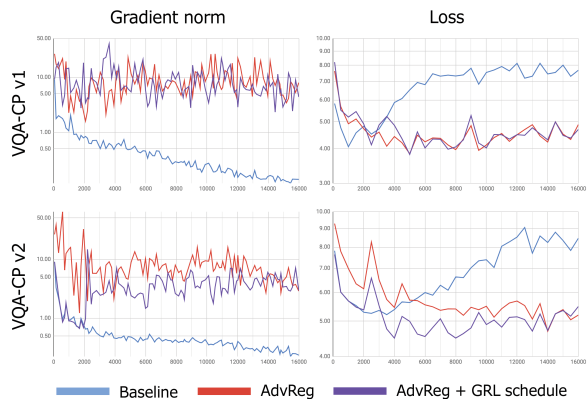
Figure 4: Gradient norms and loss during adversarial training. On VQA-CP v2, GRL scheduling helps to reduce gradient noise early in training (bottom left), leading to lower loss values (bottom right). On VQA-CP v1, the baseline (blue, top right) overfits more quickly; hence, delaying the regularization is less effective.

pecially those with fewer warm-up iterations, diverged due to exploding gradients. These findings suggest that the stronger the biases in a dataset, the earlier AdvReg must be introduced in order to counter overfitting effectively.

6 Error Analysis

We performed quantitative and qualitative error analyses to understand how AdvReg affects model inferences on different kinds of examples. To best highlight the effect of AdvReg, both analyses were performed on VQA-CP v1 test, where the change in priors is more pronounced. In both analyses, we compare our best AdvReg model (which did not use GRL scheduling) and the baseline model.

6.1 Quantitative Analysis

We first explore how model performance differs by question type. In the VQA datasets, each ques-

| | AdvReg >> Baseline | | | | | | AdvReg << Baseline | | | | |
Question type	Ans.	N	Base.	Reg.	Δ	Question type	Ans.	N	Base.	Reg.	Δ
is there a	Yes/No	6501	16.75	93.41	49.83	is this	Yes/No	13063	76.96	64.85	-15.82
is this a	Yes/No	7177	29.70	86.27	40.60	what color is the	Other	4418	47.71	21.36	-11.64
are the	Yes/No	5037	24.99	87.07	31.27	what	Other	8646	38.48	25.28	-11.42
does the	Yes/No	3525	24.02	94.34	24.79	what is the	Other	6363	41.49	28.51	-8.26
is	Yes/No	3154	32.84	92.38	18.78	is the	Other	1148	50.44	4.40	-5.29
are they	Yes/No	1577	27.96	89.40	9.69	what kind of	Other	3141	51.43	35.51	-5.00
do you	Yes/No	1083	26.14	92.32	7.17	how many	Number	15917	15.90	13.01	-4.60
is there	Yes/No	5265	68.83	78.45	5.06	what type of	Other	1995	54.74	36.30	-3.68
is the person	Yes/No	757	41.64	92.46	3.85	none of the above	Other	2057	29.65	13.66	-3.29
how many people are	Number	2118	11.96	21.08	1.93	what color are the	Other	1435	56.93	35.74	-3.04

Table 2: Comparison of relative strengths and weaknesses of regularized and baseline models. The top 10 question types for which the regularized model outperforms the baseline are shown on the left, and vice versa on the right.

tion is assigned a type corresponding to the 64 most common prefixes (e.g., "Is there a...?") or "none of the above." Additionally, each example is given an answer type (Yes/No, Number, Other).[6]

To quantify the relative performance of the AdvReg and baseline models, we computed a difference metric, weighted by the number of questions N of the given type:

$$\Delta = \tfrac{N}{100}\left(\text{score}_{\text{baseline}} - \text{score}_{\text{regularized}}\right)$$

Table 2 shows the question types with the largest and smallest Δ values, respectively. Compared to the baseline, the AdvReg model excels at Yes/No examples, but suffers on Other examples. Overall, AdvReg improves Yes/No test performance by 35.06 points, but reduces Other performance by 15.74 points (Table 1). Additionally, AdvReg reduces Number test performance by 0.95%, though in general both models score poorly on counting questions—a known shortcoming of many VQA models (Chattopadhyay et al., 2017; Trott et al., 2018; Zhang et al., 2018).

These results suggest that much of the observed advantage of AdvReg on VQA-CP test is due to the extreme biases present in the dataset. In VQA-CP, Yes/No questions encode very strong priors (e.g., "no" is the answer to roughly 90% of the questions beginning with "Is there a...?" in the v1 training set). Because this prior is inverted, any learned association between question prefixes and answers becomes harmful at test time. That AdvReg scores well above chance (77.64%) on Yes/No examples suggests that this model has, to a certain degree, learned to answer binary questions without relying on language priors.

---

[6]Note that the mapping between question types and answer types is not exactly one-to-one. However, for a given question type, a single answer type typically predominates; therefore, we are able to draw an approximate correspondence between question and answer types.

In contrast, the 15.74% drop on Other-type examples implies that AdvReg impairs the model's ability to make inferences about questions with heterogeneous answers. Other-type questions typically have 3–20 top answers. This finding suggests that AdvReg interferes with learning of language cues in the question that yield key information about the answer.

6.2 Qualitative Analysis

In this section, we examine individual examples to highlight common success and failure modes of AdvReg. We consider different question types and compare the prior answer distribution in the train/test sets to the posterior distribution assigned by the AdvReg and baseline models. Expanding on the visualization format introduced by Ramakrishnan et al. (2018, Fig. 3), Fig. 5 shows examples where the AdvReg model successfully answered the question while the baseline model was wrong. In these cases, the baseline model prediction relies on the prior answer distribution in the train set, while the AdvReg model is able to overcome these priors to infer the correct answer.

Turning to failures, we investigate what kinds of errors the AdvReg model makes on Other-type examples—the largest source of errors according to Section 6.1. We randomly selected instances where the regularized model produced an incorrect answer, and manually grouped these examples into four approximate categories corresponding to different failure modes. Fig. 6 shows representative examples for each of these failure modes; more examples are available in Appendix A.3.

Fig. 6a shows an example where the regularized model fails to infer the correct form of the answer from the question, answering "beach" to a question that entails animal answers. In Fig. 6b, the regularized model struggles with a question that
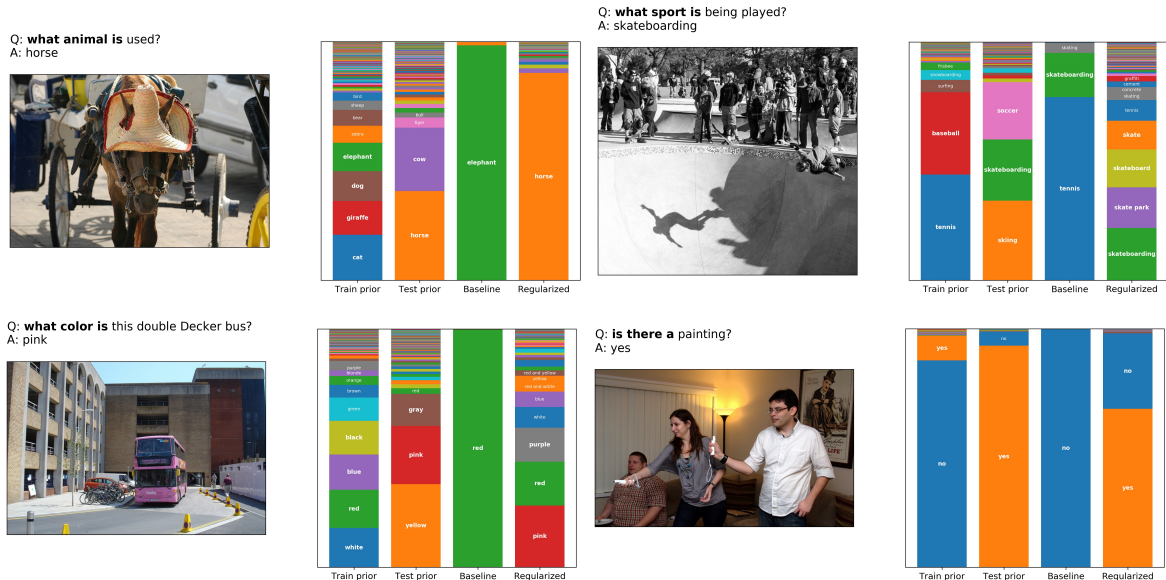
Figure 5: Visualization of AdvReg success cases. In each example, the leftmost two bars show the prior distribution over answers for the given question type (in bold). The rightmost two bars show the scores assigned to different answers by the baseline and AdvReg models for a particular example of the given type. The baseline model frequently assigns high probability to incorrect answers that are prominent in the training distribution. In contrast, the regularized model is able to make correct inferences in cases where the ground truth answer has low prior probability. Additional examples are provided in Appendix A.3.
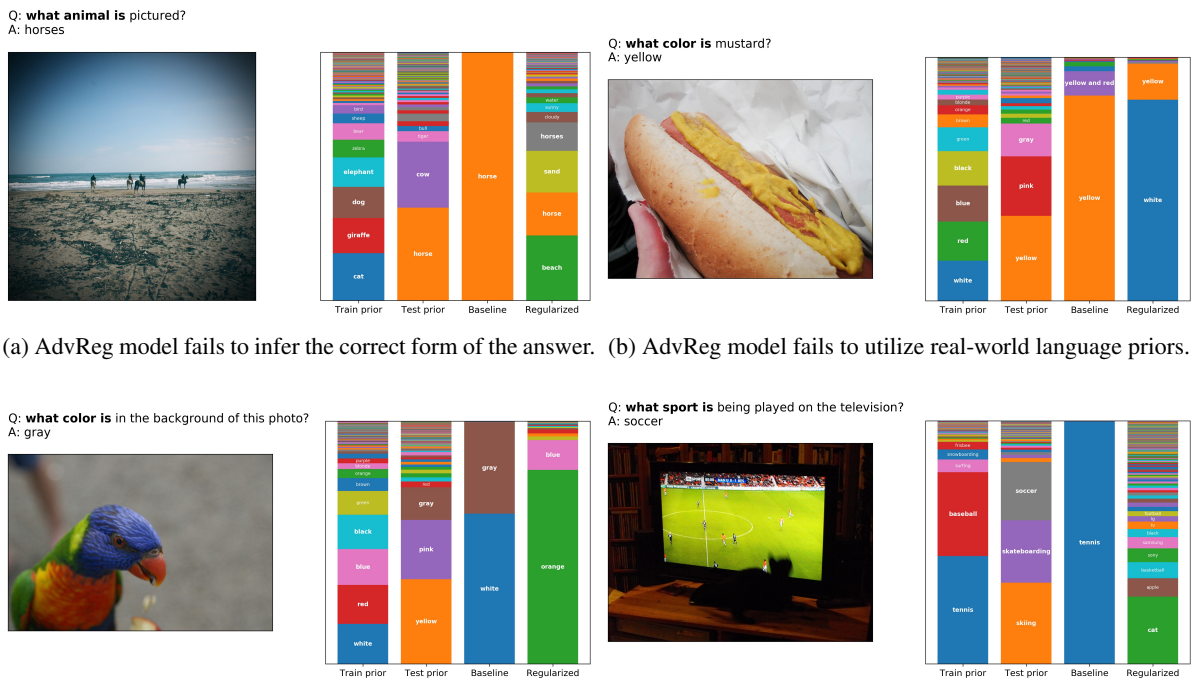
(a) AdvReg model fails to infer the correct form of the answer.

(b) AdvReg model fails to utilize real-world language priors.

(c) AdvReg model distracted by visually-salient image features.

(d) AdvReg model relies on image features, while baseline model relies on language priors.

Figure 6: Common failure modes of adversarial regularization. Additional examples are provided in Appendix A.3.

relies on real-world language priors (i.e., mustard is yellow). In Fig. 6c, the parrot's salient orange color distracts the regularized model from attending to the correct image region. Fig. 6d shows an example where the regularized model relies on visual features (the cat), while the baseline relies on language priors (tennis is a common answer to sport questions). These findings suggest that AdvReg may encourage models to rely on visual features at the expense of learning to interpret task-relevant linguistic information.

7 Conclusion

In this work, we investigated several strengths and limitations of adversarial regularization, a recently introduced technique for reducing language biases in VQA models. Though we find AdvReg improves performance on out-of-domain examples in VQA-CP, one concern is that the pendulum has swung too far: there are both quantitative and qualitative signs that our models are over-regularized. Quantitatively, the performance of our AdvReg models suffers on in-domain examples in VQA-CP and the original VQA datasets. Additionally, while AdvReg boosts performance on binary questions, it impairs performance on other question types. Qualitatively, we observe that AdvReg models draw on salient image features while ignoring important linguistic cues in questions. These results demonstrate that AdvReg interferes with certain key aspects of reasoning.

Our findings highlight the need for further research in two areas: datasets and modeling. The lack of a validation set in VQA-CP makes it difficult to perform hyperparameter tuning in a principled way. Moreover, the exaggerated biases in the existing VQA-CP splits may encourage over-regularization, as evidenced by the sharp discrepancy between AdvReg performance on binary and non-binary question types. To address these issues, future iterations of VQA-CP could contain three or more splits with moderate but distinct ratios of Yes/No answers. Restructuring VQA-CP in this way would help balance the importance of binary and non-binary questions, while providing researchers with more sound evaluation metrics.

On the modeling side, our findings suggest that AdvReg requires further refinement to avoid impairing learning of task-relevant linguistic information. One possible approach would be to use attention to apply different amounts of regularization to different words in the question. In this way, regularization could be focused on the first few words of the question (e.g., "Is there a...?") that encode answer distribution biases, while preserving other useful linguistic information. Such enhancements could lead to more targeted regularization techniques that preserve the benefits of AdvReg while reducing the drawbacks discussed in this work.

Acknowledgements

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the Behavior of Visual Question Answering Models. In *EMNLP*, pages 1955–1960.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *CVPR*, pages 4971–4980.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR*, volume 3, page 6.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *ICCV*, pages 2425–2433.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander Rush. 2019. On Adversarial Removal of Hypothesis-only Bias in Natural Language Inference. In *The Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*.

Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018. Being Negative but Constructively: Lessons Learnt from Creating Better Visual Question Answering Datasets. In *NAACL-HLT*, volume 1, pages 431–441.

Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. 2017. Counting Everyday Objects in Everyday Scenes. In *CVPR*, pages 1135–1144.

Carsten Eickhoff. 2018. Cognitive Biases in Crowdsourcing. In *WSDM*, pages 162–170. ACM.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *JMLR*, 17(1):2096–2030.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the Role of Image Understanding in Visual Question Answering. In *CVPR*, pages 6904–6913.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation Artifacts in Natural Language Inference Data. In *NAACL-HLT*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. 2018. Women also Snowboard: Overcoming Bias in Captioning Models. In *ECCV*, pages 771–787.

Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. Revisiting Visual Question Answering Baselines. In *ECCV*, pages 727–739. Springer.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018a. Pythia. https://github.com/facebookresearch/pythia.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018b. Pythia v0.1: The Winning Entry to the VQA Challenge 2018. *arXiv preprint arXiv:1807.09956*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *CVPR*, pages 1988–1997. IEEE.

Divyansh Kaushik and Zachary C. Lipton. 2018. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. In *EMNLP*, pages 5010–5015, Brussels, Belgium. ACL.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755. Springer.

Emiel van Miltenburg. 2016. Stereotyping and Bias in the Flickr30k Dataset. *arXiv preprint arXiv:1605.06083*.

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. 2016. Seeing Through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels. In *CVPR*, pages 2930–2939.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis Only Baselines in Natural Language Inference. In **SEM*, pages 180–191, New Orleans, Louisiana. ACL.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming Language Priors in Visual Question Answering with Adversarial Regularization. In *NIPS*, pages 1548–1558.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. Story Cloze Task: UW NLP System. In *LSDSem*.

Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2018. Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge. In *CVPR*, pages 4223–4232.

Jesse Thomason, Daniel Gordan, and Yonatan Bisk. 2018. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. *arXiv preprint arXiv:1811.00613*.

Alexander Trott, Caiming Xiong, and Richard Socher. 2018. Interpretable Counting for Visual Question Answering. In *ICLR*.

Masatoshi Tsuchiya. 2018. Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment. In *LREC*.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and Answering Binary Visual Questions. In *CVPR*, pages 5014–5022. IEEE.

Yan Zhang, Jonathon Hare, and Adam Prgel-Bennett. 2018. Learning to Count Objects in Natural Images for Visual Question Answering. In *ICLR*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *EMNLP*.

A Appendix

A.1 Implementation Details

Here, we provide additional details of our implementation. We experimented with different numbers of hidden layers $N = 1, 2, 3$ and hidden units $h = 256, 512, 1024, 2048$ in the adversarial classifier. We found the details of the adversary architecture to have little impact on performance, with the exception that adversaries with $N > 1$ hidden layers were more effective than one-layer adversaries. Both the adversary and the base VQA model are randomly initialized with a fixed seed at the start of training. We co-train the networks for 16k iterations with two separate PyTorch Adamax optimizers with batch size 512 and learning rate 0.001. Unlike Jiang et al. (2018b), we keep the learning rate fixed throughout training to minimize the possibility of gradient scaling mismatch between the base model and the adversary. While this modification causes the performance of the baseline VQA model to drop 1.1%, it greatly improves stability and convergence during adversarial training.

A.2 GRL Scheduling Details

For both VQA-CP v1 and v2, we performed a grid search to determine the optimal hyperparameters μ and w for the GRL schedule. We tested all combinations of delay $\mu = 0, 1000, 2000, 3000, 4000, 5000, 6000$ and warmup duration $w = 1000, 2000, 3000, 4000$. Given that the baseline model demonstrates signs of overfitting on VQA-CP v1 as early as 2000 iterations into training, we tested an additional set of accelerated GRL schedules for VQA-CP v1 that consisted of all combinations of $\mu = 500, 1000, 1500, 2000, 2500, 3000, 3500$ and $w = 500, 1000, 2000, 4000$.

Sometimes when AdvReg is introduced on a delayed schedule (especially if the value of μ is large), overfitting occurs before AdvReg takes effect. To avoid ending training prematurely, we always train for at least μ iterations before early stopping can be triggered. For instance, if $\mu = 3000$, then the earliest that we will stop training is $t = 4000$. For the purposes of evaluation, we also consider only scores from $t > \mu$ when scoring models under GRL scheduling.
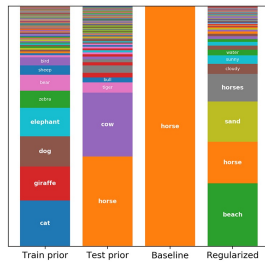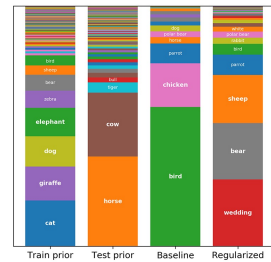
A.3 Additional Examples
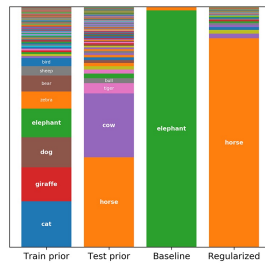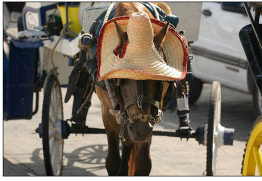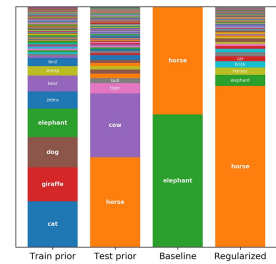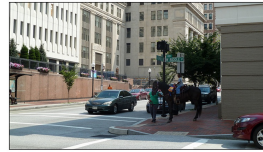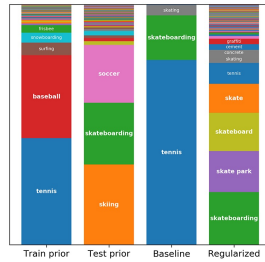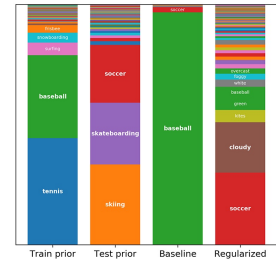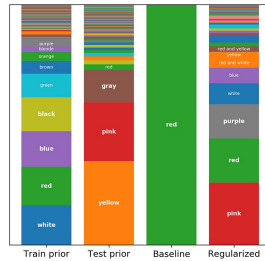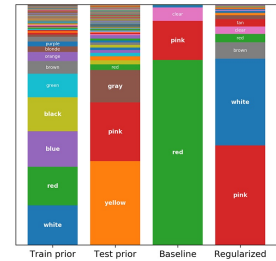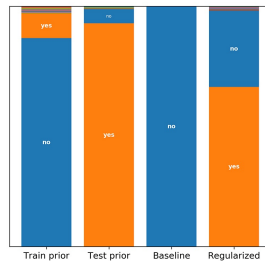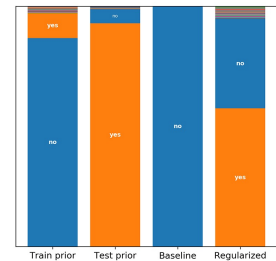
Figure 7: Visualization of AdvReg success cases. In each example, the leftmost two bars show the prior distribution over answers for the given question type (in bold). The rightmost two bars show the scores assigned to different answers by the baseline and AdvReg models for a particular example of the given type. The baseline model frequently assigns high probability to incorrect answers that are prominent in the training distribution. In contrast, the regularized model is able to make correct inferences in cases where the ground truth answer has low prior probability.
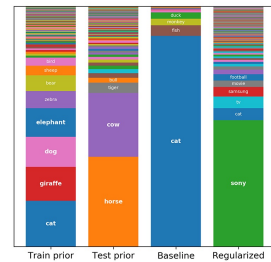
Figure 8: Common failure modes of adversarial regularization. First row: the regularized model fails to infer the correct form of the answer from the question, answering "beach" and "wedding" to questions that entail animal answers. Second row: the regularized model struggles with questions that rely on real-world language priors; i.e., mustard is yellow, sunset is orange. Third row: salient colors in the image distract the regularized model from attending to the correct image regions. Fourth row: both the baseline and regularized models perform poorly on questions where the answer relates to a localized image region (i.e., inside a TV) as opposed to the global image. In these cases, the regularized model relies on generic visual features in the image in its inferences, while the baseline model relies on language priors.