Cross-Dataset Adaptation for Visual Question Answering

Wei-Lun Chao*
U. of Southern California
Los Angeles, CA
weilunchao760414@gmail.com

Hexiang Hu*
U. of Southern California
Los Angeles, CA
hexiang.frank.hu@gmail.com

Fei Sha
U. of Southern California
Los Angeles, CA
feisha@usc.edu

Abstract

We investigate the problem of cross-dataset adaptation for visual question answering (Visual QA). Our goal is to train a Visual QA model on a source dataset but apply it to another target one. Analogous to domain adaptation for visual recognition, this setting is appealing when the target dataset does not have a sufficient amount of labeled data to learn an "in-domain" model. The key challenge is that the two datasets are constructed differently, resulting in the cross-dataset mismatch on images, questions, or answers.

We overcome this difficulty by proposing a novel domain adaptation algorithm. Our method reduces the difference in statistical distributions by transforming the feature representation of the data in the target dataset. Moreover, it maximizes the likelihood of answering questions (in the target dataset) correctly using the Visual QA model trained on the source dataset. We empirically studied the effectiveness of the proposed approach on adapting among several popular Visual QA datasets. We show that the proposed method improves over baselines where there is no adaptation and several other adaptation methods. We both quantitatively and qualitatively analyze when the adaptation can be mostly effective.

1. Introduction

Visual Question Answering (Visual QA) has emerged as a very useful task to pry into how well learning machines can comprehend and reason with both visual and textual information, which is an important functionality for general artificial intelligence. In this task, the machine is presented with an image and a relevant question. The machine can generate a free-form answer or select from a pool of candidates. In the last few years, more than a dozen datasets for the task have been developed [17, 23, 43]. Despite a steady and significant improvement in modeling, the gap in performance between humans and machines on those datasets is

Figure 1. An illustration of the dataset bias in visual question answering. Given the same image, Visual QA datasets like VQA [4] (right) and Visual7W [50] (left) provide different styles of questions, correct answers (red), and candidate answer sets, each can contributes to the bias to prevent cross-dataset generalization.

still substantial. For instance, on the VQA dataset [4] where human attains accuracy of 88.5%, the state-of-the-art model on the Visual QA multiple-choice task achieves 71.4% [48].

In this paper, we study another form of performance gap. Specifically, *can the machine learn knowledge well enough on one dataset so as to answer adeptly questions from another dataset*? Such study will highlight the similarity and difference among different datasets and guides the development of future ones. It also sheds lights on how well learning machines can understand visual and textual information in their generality, instead of learning and reasoning with dataset-specific knowledge.

Studying the performance gap across datasets is reminiscent of the seminal work by Torralba and Efros [39]. There, the authors study the bias in image datasets for object recognition. They have showed that the idiosyncrasies in the data collection process cause domain mismatch such that classifiers learnt on one dataset degrade significantly on another dataset [13, 11, 12, 25, 30, 38, 19, 37].

The language data in the Visual QA datasets introduces an addition layer of difficulty to bias in the visual data (see Fig. 1). For instance, [8] analyzes several datasets and illustrates their difference in syntactic complexity as well as within- and cross-dataset perplexity. As such, data in Visual QA datasets are likely more taletelling the origins from which datasets they come.

To validate this hypothesis, we had designed a *Name*

---

* Equal contributions

That Dataset! experiment, similar to the one in [39] for comparing visual object images. We show that the two popular Visual QA datasets VQA [4] and Visual7W [50] are almost complete distinguishable using either the question or answer data. See Sect. 3 for the details of this experiment.

Thus, Visual QA systems that are optimized on one of those datasets can focus on dataset-specific knowledge such as the type of questions as well as how the questions and answers are phrased. This type of bias exploitation hinders cross-dataset generalization and does not result in AI systems that can reason well over vision and text information in different or new characteristics.

In this paper, we investigate the issue of cross-dataset generalization in Visual QA. We assume that there is a source domain with a sufficiently large amount of annotated data such that a strong Visual QA model can be built, albeit adapted to the characteristics of the source domain well. However, we are interested in using the learned system to answer questions from another (target) domain. The target domain does not provide enough data to train a Visual QA system from scratch. We show that in this domain-mismatch setting, applying directly the learned system from the source to the target domain results in poor performance.

We thus propose a novel adaptation algorithm for Visual QA. Our method has two components. The first is to reduce the difference in statistical distributions by transforming the feature representation of the data in the target dataset. We use an adversarial type of loss to measure the degree of differences—the transformation is optimized such that it is difficult to detect the origins of the transformed features.

The second component is to maximize the likelihood of answering questions (in the target dataset) correctly using the Visual QA model trained on the source dataset. This ensures the learned transformation from optimizing domain matches retaining the semantic understanding encoded in the Visual QA model learned on the source domain.

The rest of this paper is organized as follows. In Sect. 2, we review related work. In Sect. 3, we analyze the dataset bias via the game *Name That Dataset!* In Sect. 4, we define tasks of domain adaptation for Visual QA. In Sect. 4.2, we describe the proposed domain adaptation algorithm. In Sect. 5, we conduct extensive experimental studies and further analysis. Sect. 6 concludes the paper.

2. Related Work

Datasets for Visual QA About a dozen of Visual QA datasets have been created [23, 43, 17, 16, 22, 1]. In all the datasets, there are a collection of **images (I)**. Most of existing datasets use natural images from large-scale common image databases (e.g. MSCOCO [28]). For each image, human annotators are asked to generate multiple **questions (Q)** and to provide the corresponding **"correct" answers (T)**. This gives rise to image-question-correct answer (IQT)

triplets. Visual7W [50] and VQA [4] further include artificially generated **"negative" candidate answers (D)**, referred as decoys, for the multiple-choice setting. Dataset biases can occur in any of these steps in creating the datasets.

Tasks While the machine can generate free-form answers, evaluating the answers is challenging and not amenable to automatic evaluation. Thus, so far a convenient paradigm is to evaluate machine systems using multiple-choice based Visual QA [4, 6, 50, 20]. The machine is presented the correct answer, along with several decoys (incorrect ones) and the aim is to select the right one. The evaluation is then automatic: one just needs to record the accuracy of selecting the right answer. Alternatively, the other setting is to select one from the top frequent answers and compare it to multiple human-annotated ones [3, 4, 5, 9, 16, 22, 29, 44, 45, 47, 48], avoiding constructing decoys that are too easy such that the performance is artificially boosted [4, 16].

Methods for Visual QA As summarized in [23, 43, 17], one popular framework of Visual QA algorithms is to learn a joint image-question embedding, e.g., by the attention mechanism, followed by a multi-way classifier (among the top frequent answers) [48, 3, 5, 9, 45, 29]. Though lacking the ability to generate novel answers beyond the training set, this framework has been shown to outperform those who can truly generate free-form answers. For the multiple-choice setting, one line of algorithms is to learn a scoring function with image, question, and a candidate answer as the input. Even a simple multi-layer perceptron (MLP) model achieves the state of the art [20, 9, 35].

Bias in vision and language datasets In [8], Ferraro et al. surveyed several exiting image captioning and Visual QA datasets in terms of their linguistic patterns. They proposed several metrics including perplexity, part of speech distribution, and syntactic complexity to characterize those datasets, demonstrating the existence of the reporting bias—the frequency that annotators write about actions, events, or states does not reflect the real-world frequencies. However, they do not explicitly show how such a bias affects the downstream tasks (i.e., Visual QA and captioning).

Specifically for Visual QA, there have been several work discussing the bias *within a single dataset* [16, 49, 20, 21, 6]. For example, [16, 49] argue the existence of priors on answers given the question types and the correlation between the questions and answers (without images) in VQA [4], while [6] points out the existence of bias in creating decoys. They propose to augment the original datasets with additional IQT triplets or decoys to resolve such issues. [20, 1] studies biases across datasets, and show the difficulties in transferring learned knowledge across datasets.

Our work investigates the causes of the poor cross-dataset generalization and proposes to resolve them via

domain adaptation. Those causes are orthogonal to biases in a single dataset that a learning model can exploit. Thus, merely improving the Visual QA model's performance on in-domain datasets does not imply reducing the cross-dataset generalization gap, as shown in Sect. 3.

Domain adaptation (DA) Extensive prior work has been done to adapt the domain mismatch between datasets [40, 10, 41, 7, 14, 12], mostly for visual recognition while we study a new task of Visual QA. One popular method is to learn a transformation that aligns source and target domains according to a certain criterion. Inspired by the recent flourish of Generative Adversarial Network [15], many algorithms [10, 41, 7, 46] train a domain discriminator as a new criterion for learning such a transformation. Our method applies a similar approach, but aims to perform adaptation simultaneously on data with multiple modalities (i.e., images, questions, and answers). To this end, we leverage the Visual QA knowledge learned from the source domain to ensure that the transformed features are semantically aligned. Moreover, in contrast to most existing methods, we learn the transformation from the target domain to the source one, similar to [36, 41][1], enabling applying the learned Visual QA model from the source domain without re-training.

3. Visual QA and Bias in the Datasets

In what follows, we describe a simple experiment *Name That Dataset!* to illustrate the biases in Visual QA datasets —questions and answers are idiosyncratically constructed such that a classifier can easily tell one apart from the other by using them as inputs. We then discuss how those biases give rise to poor cross-dataset generalization errors.

3.1. Visual QA

In Visual QA datasets, a training or test example is a IQT triplet that consists of an image I, a question Q, and a (ground-truth) correct answer T[2]. During evaluation or testing, given a pair of I and Q, a machine needs to *generate* an answer that matches exactly or is semantically similar to T.

In this work, we focus on multiple-choice based Visual QA, *since the two most-widely studied datasets—VQA [4] and Visual7W [50]—both consider such a setting.* In this setting, the correct answer T is accompanied by a set of K "negative" candidate answers, resulting in a candidate answer set A consist of a single T and K decoys denoted by D. An IQA triplet is thus $\{I, Q, A = \{T, D_1, \cdots, D_K\}\}$. We use C to denote an element in A. During testing, given I, Q, and A, a machine needs to select T from A. Multiple-choice
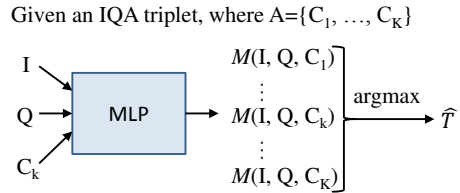

Figure 2. An illustration of the MLP-based model for multiple-choice Visual QA. Given an IQA triplet, we compute the $M(I, Q, C_k)$ score for each candidate answer C_k. The candidate answer that has the highest score is selected as the model's answer.

based Visual QA has the benefit of simplified evaluation procedure and has been popularly studied [20, 47, 9, 35, 24]. Note that in the recent datasets like VQA2 [16], the candidate set A is expanded to include the most frequent answers from the whole training set, instead of a smaller subset typically used in earlier datasets. Despite this subtle difference, we do not lose in generality by studying cross-dataset generalization with multiple-choice based Visual QA datasets.

We follow [20] to train one-hidden-layer MLP models for multiple-choice based Visual QA. The MLP M takes the concatenated features of an IQC triplet as input and outputs a compatible score $M(I, Q, C) \in [0, 1]$, measuring how likely C is the correct answer to the IQ pair. During training, M is learned to maximize the binary cross-entropy, where each IQC triplet is labeled with 1 if C is the correct answer; 0, otherwise. During testing, given an IQA triplet, the $C \in$ A that leads to the highest score is selected as the model's answer. We use the penultimate layer of ResNet-200 [18] as visual features to represent I and the average WORD2VEC embeddings [31] as text features to represent Q and C, as in [20]. See Fig. 2 for an illustration.

3.2. Bias in the Datasets

We refer the term "bias" to any idiosyncrasies in the datasets that learning algorithms can overfit to and cause poor cross-dataset generalization.

Name That Dataset! To investigate the degree and the cause of the bias, we construct a game *Name That Dataset!*, similar to the one described in [39] for object recognition datasets. In this game, the machine has access to the examples (ie, either IQT or IQA triplet) and needs to decide which dataset those examples belong to. We experiment on two popular datasets Visual7W [50] and VQA [4]. We use the same visual and text features described in Sect. 3.1 to represent I, Q, T, and D[3]. We then concatenate these features to form the joint feature. We examine different combination of I, Q, T, D as the input to a one-hidden-layer MLP for predicting the dataset from which the sample comes. We sample 40,000, 5,000 and 20,000 triplets from each dataset

---

[1]Most DA algorithms, when given a target domain, adjust the features for both domains and retrain the source model on the adjusted features— they need to retrain the model when facing a new target domain. Note that [36, 41] do not incorporate the learned source-domain knowledge as ours.

[2]Some datasets provide multiple correct answers to accommodate the ambiguity in the answers.

[3]Visual7W [50] has 3 decoys per triplet and VQA [4] has 17 decoys. For fair comparison, we subsample 3 decoys for VQA. We then average the WORD2VEC embedding of each decoy to be the feature of decoys.

Table 1. Results of *Name That Dataset!*

Information	I	Q	T	D	Q + T	Q + D	T + D	Q + T + D	Random
Accuracy	52.3%	76.3%	74.7%	95.8%	79.8%	97.5%	97.4%	97.5%	50.00%

and merge them to be the training, validation and test sets. Details are in the Suppl.

As shown in Table 1, all components but images lead to strong detection of the data origin, with the decoys contributing the most (i.e., 95.8% alone). Combining multiple components further improve the detection accuracy, suggesting that datasets contain different correlations or relationships among components. Concatenating all the components together results in nearly 100% classification accuracy. In other words, the image, question, and answers in each dataset are constructed characteristically. Their distributions (in the joint space) are sufficiently distant from each other. Thus, one would not expect a Visual QA system trained on one dataset to work well on the other datasets. See below for results validating this observation.

Question Type is just one biasing factor Question type is an obvious culprit of the bias. In Visual7W, questions are mostly in the $6W$ categories (ie, what, where, how, when, why, who). On the other hand, the VQA dataset contains additional questions whose correct answers are either Yes or No. Those questions barely start with the $6W$ words. We create a new dataset called VQA$^-$ by removing the Yes or No questions from the original VQA dataset.

We reran the *Name That Dataset!* (after retraining on the new dataset). The accuracies of using Q or Q+T have dropped from 76.3% and 79.8% to 69.7% and 73.8%, respectively, which are still noticeably higher than 50% by chance. This indicates that the questions or correct answers may phrased differently between the two datasets (e.g., the length or the use of vocabularies). Combining them with the decoys (i.e., Q+T+D) raises the accuracy to 96.9%, again nearly distinguishing the two datasets completely. This reflects that the incorrect answers must be created very differently across the two datasets (In most cases, decoys are freely selected by the data collectors—being incorrect answers to the questions affords the data collectors to sample from unconstrained spaces of possible words and phrases.)

Poor cross-dataset generalization Using the model described in Sect. 3.1, we obtain the Visual QA accuracies of 65.7% and 55.6% on Visual7W and VQA$^-$ when training and testing using the same dataset. However, when the learned models are applied to the other dataset, the performance drops significantly to 53.4% (trained on VQA$^-$ but applied to Visual7W) and 28.1% (trained on Visual7W but applied to VQA$^-$). See Table 3 for the details.

We further evaluate a variant of the spatial memory network [44], a more sophisticated Visual QA model. A similar performance drop is observed. See Table 6 for details.

Table 2. Various Settings for cross-dataset Adaptation. Source domain always provide I, Q and A (T+D) while the target domain provides the same *only* during testing.

Shorthand	Data from Target at Training
Setting[Q]	Q
Setting[Q+T] (or [Q+T+D])	Q, T (or Q, T+D)
Setting[T] (or [T+D])	T (or T+D)

4. Cross-Dataset Adaptation

We propose to overcome the cross-dataset bias (and the poor cross-dataset generalization) with the idea of *domain adaptation*. Similar ideas have been developed in the past to overcome the dataset bias for object recognition [34, 13].

4.1. Main Idea

We assume that we have a source domain (or dataset) with plenty of annotated data in the form of Image-Question-Candidate Answers (IQA) triplets such that we can build a strong Visual QA system. We are then interested in applying this system to the target domain. However, we do not assume there is any annotated data (i.e., IQA/IQT triplets) from the target domain such that re-training (either using the target domain alone or jointly with the source domain) or fine-tuning [32, 42] the system is feasible[4].

Instead, the target domain provides *unsupervised data*. The target domain could provide images, images and questions (without either correct or incorrect answers), questions, questions with either correct or incorrect answers or both, or simply a set of candidate answers (either correct or incorrect or both). This last two scenarios are particularly interesting[5]. From the results in Table 1, the discrepancy in textual information is a major contributor to domain mismatch, cf. the columns starting Q.

Given the target domain data, it is not feasible to train an "in-domain" model with the data (as it is incomplete and unsupervised). We thus need to model jointly the source domain supervised data and the target domain data that reflect distribution mismatch. Table 2 lists the settings we work on.

4.2. Approach

Our approach has two components. In the first part, we match features encoding questions and/or answers across

---

[4]Annotated data from the target data, if any, can be easily incorporated into our method as a supervised learning discriminative loss. We leave this for a full version of the current paper.

[5]Most existing datasets are derived from MSCOCO. Thus there are limited discrepancies between images, as shown in the column I in Table 1. Our method can also be extended to handle large discrepancy in images. Alternatively, existing methods of domain adaptation for visual recognition could be applied to images first to reduce the discrepancy.

two domains. In the second part, we ensure the correct answers from the target domain have higher likelihood in the Visual QA model trained on the source domain. Note that we do not re-train the Visual QA model as we do not have access to complete data on the target domain.

Matching domain   The main idea is to transform features computed on the target domain (TD) to match those features computed on the source domain (SD). To this end, let $g_q(\cdot)$ and $g_a(\cdot)$ denote the transformation for the features on the questions and on the answers respectively. We also use f_q, f_t, f_d, and f_c to denote feature representations of a question, a correct answer, an incorrect decoy, or a candidate answer. In the Visual QA model, all these features are computed by the average WORD2VEC embeddings of words.

The matching is computed as the Jensen-Shannon Divergence (JSD) between the two empirical distributions across the datasets. For the Setting[Q], the matching is

$$m(\mathsf{TD} \rightarrow \mathsf{SD}) = JSD(\hat{p}_{\mathsf{SD}}(f_q), \hat{p}_{\mathsf{TD}}(g_q(f_q))) \quad (1)$$

where $\hat{p}_{\mathsf{SD}}(f_q)$ is the empirical distribution of the questions in the source domain and $\hat{p}_{\mathsf{TD}}(g_q(f_q))$ is the empirical distribution of the questions in the target domain, after being transformed with $g_q(\cdot)$,

The JSD divergence between two distributions P and P' is computed as

$$JSD(P, P') = \frac{1}{2}\left\{ KL\left(P; \frac{P+P'}{2}\right) + KL\left(P'; \frac{P+P'}{2}\right) \right\}, \quad (2)$$

while KL is the KL divergence between two distributions. The JSD divergence is closely related to discriminating two distributions with a binary classifier [15] but difficult to compute. We thus use an adversarial lose to approximate it. See the Suppl. for details.

For both the Setting[Q+T] and the Setting[Q+T+D], the matching is

$$m(\mathsf{TD} \rightarrow \mathsf{SD}) = JSD(\hat{p}_{\mathsf{SD}}(f_q, f_t), \hat{p}_{\mathsf{TD}}(g_q(f_q), g_a(f_t))) \quad (3)$$

with the empirical distributions computed over both the questions and the correct answers. Note that even when the decoy information is available, we deliberately ignore them in computing domain mismatch. This is because the decoys can be designed very differently even for the same IQT triplet. Matching the distributions of D thus can cause undesired mismatch of T since they share the same transform during testing[6].

[6]Consider the following highly contrived example. To answer the question "what is in the cup?", the annotators in the source domain could answer with "water" as the correct answer, and "coffee", "juice" as decoys, while the annotators in the target domain could answer with "sparkling water" (as that is the correct answer), then "cat" (as in cupcats), and "cake" (as in cupcakes) as decoys. While it is intuitive to match the distribution of correct answers, it makes less sense to match the distributions of the decoys as they are much more dispersed.

For the Setting[T] and Setting[T+D], the matching is

$$m(\mathsf{TD} \rightarrow \mathsf{SD}) = JSD(\hat{p}_{\mathsf{SD}}(f_t), \hat{p}_{\mathsf{TD}}(g_a(f_t))) \quad (4)$$

while the empirical distributions are computed over the correct answers only.

Leverage Source Domain for Discriminative Learning In the Setting[Q+T], Setting[Q+T+D], Setting[T] and Setting[T+D], the learner has access to the correct answers T (and the incorrect answers D) from the target domain. As we intend to use the transformed feature $g_q(f_q)$ and $g_a(f_c)$ with the Visual QA model trained on the source domain, we would like those transformed features to have high likelihood of being correct (or incorrect).

To this end, we can leverage the source domain's data which always contain both T and D. The main idea is to construct a Visual QA model on the source domain using the same partial information as in the target domain, then to assess how likely the transformed features remain to be correct (or incorrect).

In the following, we use the Setting[Q+T+D] as an example (other settings can be formulated similarly). Let $h_{\mathsf{SD}}(q, c)$ be a model trained on the source domain such that it tells us the likelihood an answer c can be correct with respect to question q. Without loss of generality, we assume $h_{\mathsf{SD}}(q, c)$ is the output of a binary logistic regression.

To use this model on the target data, we compute the following loss for every pair of question and candidate answer:

$$\ell(q, c) = \begin{cases} -\log h_{\mathsf{SD}}(g_q(f_q), g_a(f_c)) & \text{if } c \text{ is correct,} \\ -\log(1 - h_{\mathsf{SD}}(g_q(f_q), g_a(f_c))) & \text{otherwise.} \end{cases}$$

The intuition is to raise the likelihood of any correct answers and lowering the likelihood of any incorrect ones. Thus, even we do not have a complete data for training models on the target domain discriminatively, we have found a surrogate to minimize,

$$\hat{\ell}_{\mathsf{TD}} = \sum_{(q,c) \in \mathsf{TD}} \ell(q, c), \quad (5)$$

measuring all the data provided in the target data and how they are likely to be correct or incorrect.

4.3. Joint optimization

We learn the feature transformation by jointly balancing the domain matching and the discriminative loss surrogate

$$\arg\min_{g_q, g_a} m(\mathsf{TD} \rightarrow \mathsf{SD}) + \lambda \hat{\ell}_{\mathsf{TD}}. \quad (6)$$

We select λ to be large while still allowing $m(\mathsf{TD} \rightarrow \mathsf{SD})$ to decrease in optimization: λ is 0.5 for Setting[Q+T+D] and Setting[T+D], and 0.1 for the other experiments. The learning objective can be similarly constructed when the target domain provides Q and T, T, or T+D, as explained above. If the target domain only provides Q, we omit the term $\hat{\ell}_{\mathsf{TD}}$.

Once the feature transformations are learnt, we use the Visual QA model on the source domain M_{SD}, trained using image, question, and answers all together to make an inference on an IQA triplet (i, q, A) from the target

$$\hat{t} = \arg\max_{c \in A} M_{\mathsf{SD}}(f_i, g_q(f_q), g_a(f_c)),$$

where we identify the best candidate answer from the pool of the correct answers and their decoys A using the source domain's model. See Sect. 5.2 and the Suppl. for the parameterization of $g_q(\cdot)$ and $g_a(\cdot)$, and details of the algorithm.

5. Experiments

5.1. Dataset

We first evaluate our algorithms on the domain adaptation settings defined in Sect. 4 between Visual7W [50] and VQA [4]. Experiments are conducted on both the original datasets and a revised version [6] of them. We then include Visual Genome [27] with the decoys created by [6] and apply the same procedure to create decoys for the CO-COQA [33] and VQA2 [16] datasets, leading to a comprehensive study of cross-dataset generalization.

VQA Multiple Choice [4] The dataset uses images from the MSCOCO [28] dataset, with the same split setting. It contains 248,349/121,512/244,302 IQA triplets for training/validation/test. Each triplet has 17 decoys, where in general 3 decoys are human-generated, 4 are randomly sampled, and 10 are from fixed set of high frequency answers.

Visual7W Telling [50] The dataset contains 47,300 images from MSCOCO [28] and in total 139,868 IQA triplets (69,817/28,020/42,031 for training/validation/test). Each triplet has 3 decoys: all of them are human-generated.

Curated Visual7W & VQA [6] These datasets are revised versions of Visual7w and VQA, in which the decoys are carefully designed to prevent machines from ignoring the visual information, the question, or both while still doing well on the task. Each IQT triplet in Visual7W and VQA are augmented with 6 auto-generated decoys as candidate answers. Since [6] only provide revised decoys for the training and validation splits, for all the studies on VQA we report results on the validation set.

Visual Genome [27] & COCOQA [33] & VQA2 [16] These three datasets only provide IQT triplets, while [6] creates decoys for Visual Genome (VG). We required the codes from the authors of [6], and apply the same procedure to create decoys for COCOQA and VQA2—each IQT triplet is augmented with 6 auto-generated decoys. The resulting datasets have 727,751/283,666/433,905 IQA triplets for training/validation/test on VG, 78,736/38,948 for training/test on COCOQA, and 443,757/214,354 for training/validation on VQA2. All datasets use images from MSCOCO [28].

Evaluation metric For Visual7W, VG, and COCOQA, we compute the accuracy of picking the correct answer from multiple choices. For VQA and VQA2, we follow its protocol to compute accuracy, comparing the picked answer to the 10 human-annotated correct answers. The accuracy is computed based on the number of exact matches among the 10 answers (divided by 3 and clipped at 1).

5.2. Experimental setup

Visual QA model In all our experiments, we use a one hidden-layer MLP model (with 8,192 hidden nodes and ReLU) to perform binary classification on each input IQC (image, question, candidate answer) triplet, following the setup as in [20, 6]. Please see Fig. 2 and Sect. 3.1 for explanation. The candidate $C \in A$ that has the largest score is then selected as the answer of the model. Such a simple model has achieved the state-of-the-art results on Visual7W and comparable results on VQA.

For images, we extract convolutional activation from the last layer of a 200-layer Residual Network [18]; for questions and answers, we extract the 300-dimensional WORD2VEC [31] embedding for each words in a question/answer and compute their average as the feature. We then concatenate these features to be the input to the MLP model. Besides the Visual QA model that takes I, Q, and C as input, we also train two models that use only Q + C and C alone as the input. These two models can serve as h_{SD} described in Sect 4.2.

Using simple models like MLP and average WORD2VEC embeddings adds credibility to our studies—if models with limited capacity can latch on to the bias, models with higher capacity can only do better in memorizing the bias.

Domain adaptation model We parameterize the transformation $g_q(\cdot)$, $g_a(\cdot)$ as a one hidden-layer MLP model (with 128 hidden nodes and ReLU) with residual connections directly from input to output. Such a design choice is due to the fact that the target embedding can already serve as a good starting point of the transforms. We approximate the $m(\mathsf{TD} \rightarrow \mathsf{SD})$ measure by adversarially learning a one hidden-layer MLP model (with 8,192 hidden nodes and ReLU) for binary classification between the source and the transformed target domain data, following the same architecture as the classifier in *Name That Dataset!* game.

For all our experiments on training $g_q(\cdot)$, $g_a(\cdot)$ and approximating $m(\mathsf{TD} \rightarrow \mathsf{SD})$, we use Adam [26] for stochastic gradient-based optimization. See the Suppl. for details.

Domain adaptation settings As mentioned in Sect. 3, VQA (as well as VQA2) has around 30% of the IQA triplets with the correct answers to be either "Yes" or "NO". On the other hand, Visual7W, COCOQA, and VG barely have triplets with such correct answers. Therefore, we remove those triplets from VQA and VQA2, leading to a reduced

Table 3. Domain adaptation (DA) results (in %) on *original* VQA [2] and Visual7W [50]. **Direct**: direct transfer without DA. [36]: CORAL. [41]: ADDA. **Within**: apply models trained on the target domain if supervised data is provided. (best DA result in bold)

VQA⁻ → Visual7W								
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within
53.4	53.4	54.1	53.6	54.5	55.7	55.2	**58.5**	65.7
Visual7W → VQA⁻								
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within
28.1	26.9	29.2	28.1	29.7	33.6	29.4	**35.2**	55.6

Table 4. Domain adaptation (DA) results (in %) on *revised* VQA and Visual7W [6]. (best DA result in bold)

VQA⁻ → Visual7W								
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within
46.1	47.2	47.8	46.2	47.6	47.6	48.4	**49.3**	52.0
Visual7W → VQA⁻								
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within
45.6	45.3	45.9	45.9	45.9	47.8	45.8	**48.1**	53.7

dataset VQA⁻ and VQA2⁻ that has 153,047/76,034 and 276,875/133,813 training/validation triplets, respectively.

We learn the Visual QA model using the training split of the source dataset and learn the domain adaptation transform using the training split of both datasets.

Other implementation details Questions in Visual7W, COCOQA, VG, VQA⁻, and VQA2⁻ are mostly started with the 6W words. The frequencies, however, vary among datasets. To encourage g_q to focus on matching the phrasing style rather than transforming one question type to the others, when training the binary classifier for $m(\mathsf{TD} \rightarrow \mathsf{SD})$ with Adams, we perform weighted sampling instead of uniform sampling from the source domain—the weights are determined by the ratio of frequency of each of the 6W question types between the target and source domain. This trick makes our algorithm more stable.

5.3. Experimental Results on Visual7W and VQA−

We experiment on the five domain adaptation (DA) settings introduced in Sect. 4 using the proposed algorithm. We also compare with ADDA [41] and CORAL [36], two DA algorithms that can also learn transformations from the target to the source domain and achieves comparable results on many benchmark datasets. Specifically, we learn two transformations to match the (joint) distribution of the questions and target answers. *We only report the best performance among the five settings for ADDA and CORAL.* Table 3 and Table 4 summarize the results on the original and revised datasets, together with **Direct** transfer without any domain adaptation and **Within** domain performance where the Visual QA model is learned using the *supervised data (i.e., IQA triplets)* of the target domain. Such supervised data is inaccessible in the adaptation settings we considered.

Domain mismatch hurts cross-dataset generalization The significant performance drop in comparing **Within** do-

Table 5. DA results (in %) on *revised* datasets, with target data sub-sampling by 1/16. FT: fine-tuning. (best DA result in bold)

VQA⁻ → Visual7W									
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within	FT
46.1	45.6	47.8	46.1	47.5	47.6	48.3	**49.1**	39.7	48.3
Visual7W → VQA⁻									
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within	FT
45.6	44.8	45.6	46.0	45.9	47.8	45.8	**48.0**	43.1	48.2

Table 6. DA results (in %) on on *revised* datasets using a variant of the SMem [44] model.

VQA⁻ → Visual7W			Visual7W → VQA⁻		
Direct	[Q+T+D]	Within	Direct	[Q+T+D]	Within
48.6	51.2	52.8	46.6	48.4	58.6

main and **Direct** transfer performance suggests that the learned Visual QA models indeed exploit certain domain-specific bias that may not exist in the other datasets. Such a drop is much severe between the original datasets than the revised datasets. Note that the two versions of datasets are different only in the decoys, and the revised datasets create decoys for both datasets by the same automatic procedure. Such an observation, together with the finding from *Name That Dataset!* game, indicate that decoys contribute the most to the domain mismatch in Visual QA.

Comparison on domain adaptation algorithms Our domain adaptation algorithm outperforms **Direct** transfer in all the cases. On contrary, CORAL [36], which aims to match the first and second order statistics between domains, fails in several cases, indicating that for domain adaptation in Visual QA, it is crucial to consider higher order statistics.

We also examine setting λ in Eq. (6) to 0 for the [T] and [Q+T] settings[7] (essentially ADDA [41] extended to multiple modalities), which leads to a drop of $\sim 1\%$, demonstrating the effectiveness of leveraging the source domain for discriminative learning. See the Suppl. for more details.

Different domain adaptation settings Among the five settings, we see that [T] generally gives larger improvement over **Direct** than [Q], suggesting that the domain mismatch in answers hinder more in cross-dataset generalization.

Extra information on top of [T] or [Q] generally benefits the domain adaptation performance, with [Q+T+D] giving the best performance. Note that different setting corresponds to different objectives in Eq. (6) for learning the transformations g_q and g_a. Comparing [T] to [T+D], we see that adding D helps take more advantage of exploiting the source domain's Visual QA knowledge, leading to a g_a that better differentiates the correct answers from the decoys. On the other hand, adding T to [Q], or vice versa, helps constructing a better measure to match the feature distribution between domains.

Domain adaptation using a subset of data The domain adaptation results presented in Table 3 and 4 are based on

---

[7]When $\lambda = 0$, D has no effect (i.e., [Q+T+D] is equivalent to [Q+T]).

Table 7. Transfer results (in %) across different datasets (the decoys are generated according to [6]). The setting for domain adaptation (DA) is on [Q+T+D] using **1/16** of the training examples of the target domain.

Training/Testing	Visual7W			VQA$^-$			VG			COCOQA			VQA2$^-$		
	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within
Visual7W	52.0	-	-	45.6	48.0	43.1	49.1	49.4	48.0	58.0	63.1	65.2	43.9	45.5	43.6
VQA$^-$	46.1	49.1	39.7	53.7	-	-	44.8	47.4	48.0	59.0	63.4	65.2	50.7	50.6	43.6
VG	58.1	58.3	39.7	52.6	54.6	43.1	58.5	-	-	65.5	68.8	65.2	50.1	51.3	43.6
COCOQA	30.1	35.5	39.7	35.1	40.4	43.1	29.1	33.1	48.0	75.8	-	-	33.3	37.5	43.6
VQA2$^-$	48.8	50.8	39.7	55.2	55.3	43.1	47.3	49.1	48.0	60.3	64.9	65.2	53.8	-	-

learning the transformations using all the training examples of the source and target domain. We further investigate the robustness of the proposed algorithm under a limited number of target examples. We present the results using only 1/16 of the them in Table 5. The proposed algorithm can still learn the transformations well under such a scenario, with a slight drop in performance (i.e., $< 0.5\%$). In contrast, learning Visual QA models with the same amount of limited target data (assuming the IQA triplets are accessible) from scratch leads to significant performance drop. We also include the results by fine-tuning, which is infeasible in any setting of Table 2 but can serve as an upper bound.

Results on sophisticated Visual QA model We further investigate a variant of the spatial memory network (SMem) [44] for Visual QA, which utilizes the question to guide the visual attention on certain parts of the image for extracting better visual features. The results are shown in Table 6, where a similar trend of improvement is observed.

Qualitative analysis The question type (out of the $6W$ words) that improves the most from Direct to DA, when transferring from VQA− to Visual7W in Table 3 using [Q+T+D], is "When" (from 41.8 to 63.4, while Within is 80.3). Other types improve $1.0 \sim 5.0$. This is because that the "When"-type question is scarcely seen in VQA−, and our DA algorithm, together with the weighted sampling trick, significantly reduces the mismatch of question/answer phrasing of such a type. See the Suppl. for other results.

5.4. Experimental Results across five datasets

We perform a more comprehensive study on transferring the learned Visual QA models across five different datasets. We use the *revised* candidate answers for all of them to reduce the mismatch on how the decoys are constructed. We consider the [Q+T+D] setting, and limit the disclosed target data to **1/16** of its training split size. The models for **Within** are also trained on such a size, using the supervised IQA triplets. Table 7 summarizes the results, where rows/columns correspond to the source/target domains.

On almost all (source, target) pairs, domain adaptation (DA) outperforms **Direct**, demonstrating the wide applicability and robustness of our algorithm. The exception is on (VQA$^-$, VQA2$^-$), where DA degrades by 0.1%. This is likely due to the fact that these two datasets are constructed similarly and thus no performance gain can be achieved.

Such a case can also be seen between Visual7W and VG. Specifically, domain adaptation is only capable in better transferring the knowledge learned in the source domain, but cannot acquire novel knowledge in the target domain.

The reduced training size significantly limits the performance of training from scratch (**Within**). In many cases **Within** is downplayed by DA, or even by **Direct**, showing the essential demand to leverage source domain knowledge. Among the five datasets, Visual QA models trained on VG seems to generalize the best—the DA results to any target domain outperforms the corresponding **Within**—indicating the good quality of VG.

In contrast, Visual QA models trained on COCOQA can hardly transfer to other datasets—none of its DA results to other datasets is higher than **Within**. It is also interesting to see that none of the DA results from other source domain (except VG) to COCOQA outperforms COCOQA's **Within**. This is, however, not surprising given how differently in the way COCOQA is constructed; i.e., the questions and answers are automatically generated from the captions in MSCOCO. Such a significant domain mismatch can also be witnessed from the gap between **Direct** and DA on any pair that involves COCOQA. The performance gain by DA over **Direct** is on average over 4.5%, larger than the gain of any other pair, further demonstrating the effectiveness of our algorithms in reducing the mismatch between domains.

6. Conclusion

We study cross-dataset adaptation for visual question answering. We first analyze the causes of bias in existing datasets. We then propose to reduce the bias via domain adaptation so as to improve cross-dataset knowledge transfer. To this end we propose a novel domain adaptation algorithm that minimizes the domain mismatch while leveraging the source domain's Visual QA knowledge. Through experiments on knowledge transfer among five popular datasets, we demonstrate the effectiveness of our algorithm, even under limited and fragment target domain information.

References

[1] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018. 2

[2] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. Lawrence Zitnick, D. Parikh, and D. Batra. Vqa: Visual question answering. *IJCV*, 2016. 7

[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. In *CVPR*, 2018. 2

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1, 2, 3, 6, 11, 12, 14

[5] H. Ben-younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017. 2

[6] W.-L. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 2, 6, 7, 8, 14

[7] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *ICCV*, 2017. 3

[8] F. Ferraro, N. Mostafazadeh, L. Vanderwende, J. Devlin, M. Galley, M. Mitchell, et al. A survey of current datasets for vision and language research. In *EMNLP*, 2015. 1, 2

[9] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 2, 3

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(59):1–35, 2016. 3

[11] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, 2013. 1

[12] B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, 2013. 1, 3

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 1, 4

[14] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In *ICML*, 2016. 3

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 3, 5, 11

[16] 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 *CVPR*, 2017. 2, 3, 6, 11, 14

[17] A. K. Gupta. Survey of visual question answering: Datasets and techniques. *arXiv preprint arXiv:1705.03865*, 2017. 1, 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6, 11

[19] L. Herranz, S. Jiang, and X. Li. Scene recognition with cnns: Objects, scales and dataset bias. In *CVPR*, 2016. 1

[20] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016. 2, 3, 6, 11

[21] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2

[22] K. Kafle and C. Kanan. An analysis of visual question answering algorithms. In *ICCV*, 2017. 2

[23] K. Kafle and C. Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017. 1, 2

[24] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 3

[25] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 1

[26] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6, 11

[27] 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. *IJCV*, 2017. 6, 11, 14

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6

[29] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 2, 11, 12, 13

[30] N. McLaughlin, J. M. Del Rincon, and P. Miller. Data-augmentation for reducing dataset bias in person re-identification. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, 2015. 1

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 3, 6, 11

[32] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 4

[33] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 6, 11, 14

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, 2010. 4

[35] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 2, 3

[36] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016. 3, 7, 12

[37] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. In *German Conference on Pattern Recognition*, 2015. 1

[38] T. Tommasi and T. Tuytelaars. A testbed for cross-dataset analysis. In *ECCV Workshop*, 2014. 1

[39] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1, 2, 3

[40] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 3

[41] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 3, 7, 12

[42] G. Wiese, D. Weissenborn, and M. L. Neves. Neural domain adaptation for biomedical question answering. In *CoNLL*, 2017. 4

[43] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017. 1, 2

[44] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 2, 4, 7, 8, 11, 12, 13

[45] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 2

[46] Z. Yang, J. Hu, R. Salakhutdinov, and W. W. Cohen. Semi-supervised qa with generative domain-adaptive nets. In *ACL*, 2017. 3

[47] D. Yu, J. Fu, T. Mei, and Y. Rui. Multi-level attention networks for visual question answering. In *CVPR*, 2017. 2, 3

[48] Z. Yu, J. Yu, J. Fan, and D. Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 2017. 1, 2

[49] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and yang: Balancing and answering binary visual questions. In *CVPR*, 2016. 2, 11

[50] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 2, 3, 6, 7, 11, 12, 14

Supplementary Material

We provide contents omitted in the main text.

- Section A: details on *Name that dataset!* (Sect. 3.2 of the main text).
- Section B: details on the proposed domain adaptation algorithm (Sect. 4.2 and 4.3 of the main text).
- Section C: details on the experimental setup (Sect. 5.2 of the main text).
- Section D: additional experimental results (Sect. 5.3 and 5.4 of the main text).

A. *Details on Name that Dataset!*

As mentioned in Sect. 3.2 of the main text, we train a one-hidden-layer MLP to perform binary classification for detecting the origin of an input IQA triplet. The hidden layer is of 8,192 nodes and with the ReLU activation. The output of the MLP is normalized into $[0, 1]$ via the sigmoid function, and we train the MLP with the logistic (cross entropy) loss. We use the penultimate layer of ResNet-200 [18] as visual features to represent I and the average WORD2VEC embeddings [31] as text features to represent Q and each $C \in A$, as in [20]. We represent the whole set of decoys (denoted as D) in A by the average of those decoys' features[8]. The input to the MLP is the concatenation of features from I, Q, T, and D (or a subset of them). The size of the training/validation/test triplets is 80,000/10,000/40,000, half from each dataset (i.e., either VQA [4] or Visual7W [50]).

B. Details on the Proposed Domain Adaptation Algorithm

B.1. Approximating the JSD divergence

As mentioned in Sect. 4.2 of the main text, we use the Jensen-Shannon Divergence (JSD) to measure the domain mismatch between two domains according to their empirical distributions. Dependent on the domain adaptation (DA) setting, the empirical distribution is computed on the (transformed) questions, (transformed) correct answers, or both.

Since JSD is hard to compute, we approximate it by training a binary classifier WhichDomain$(\cdot)$ to detect the domain of a question Q, a correct answer T, or a QT pair, following the idea of Generative Adversarial Network [15]. The architecture of WhichDomain$(\cdot)$ is exactly the same as that used for *Name that dataset!*, except that the input features of examples from the target domain are after the transformations $g_q(\cdot)$ and $g_a(\cdot)$.

B.2. Details on the proposed algorithm

We summarize the proposed domain adaptation algorithm for Visual QA under Setting[Q+T+D] in Algorithm 1. Algorithms of the other settings can be derived by removing the parts corresponding to the missing information.

C. Details on the Experimental Setup

C.1. Implementation details

For all our experiments on training $g_q(\cdot)$, $g_a(\cdot)$, and WhichDomain$(\cdot)$, we use Adam [26] for stochastic gradient-based optimization, with learning rate $= 10^{-4}$ and mini-batch size = 100. We set $\lambda = 0.5$ for Setting[Q+T+D] and Setting[T+D], and 0.1 for the others. We set $k = 500$, and $l = 5$, and train for 1,000 iterations.

---

[8]Visual7W [50] has 3 decoys per triplet and VQA [4] has 17 decoys. For fair comparison, we subsample 3 decoys for VQA. We then average the WORD2VEC embedding of each decoy to be the feature of decoys.
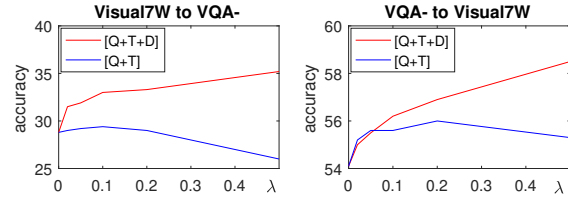

Figure 3. Results by varying λ on the original VQA and Visual7W datasets, for both the [Q+T] and [Q+T+D] settings.

C.2. Domain adaptation settings

Note that the "Yes" or "NO" issue we consider between VQA [4], VQA2 [16] and Visaul7W [50], Visual Genome (VG) [27], COCOQA [33] is orthogonal to the one addressed in [16, 49], which deal with the prior of answers within a single dataset.

C.3. Sophisticated Visual QA models

In the main text we experiment with a variant of the spatial memory network (SMem) [44]. Instead of computing the visual attention for each word of the question, we directly compute the visual attention for the question using the average WORD2VEC embeddings. We then concatenate the resulting visual features with the features of the question and a candidate answer (in the same way as the Visual QA model in Sect. 5.2 of the main text) as the input to train a one-hidden-layer MLP for binary classification.

We choose to train an MLP with candidate answers as a part of the input rather than training a multi-way classifier for the top frequent answers because the answer distributions can vary drastically across different domains or datasets. In such a case, an IQA triplet of the target domain can never be answered correctly by the learnt multi-way classifier on the source domain if the correct answer is not in the top frequent answers of the source domain.

In the Supplementary Material, we further experiment with a variant of the HieCoAtt model [29], which applies the attention mechanism not only to images but also to questions (e.g., which word or phrase is more important). We extract the HieCoAtt features by removing the last layer (i.e., the multi-way classifier) of the HieCoAtt model, and concatenate the features again with the average WORD2VEC embeddings of the question and a candidate answer to train an MLP for binary classification. The cross-dataset results are presented in Sect. D.4. *Note that we conduct this experiment not to achieve better performance, but to show that the dataset bias will also hinder cross-dataset generalization for more sophisticated models.*

D. Additional Experimental Results

D.1. The effect of the discriminative loss surrogate

We provide in Table 8 the domain adaptation results on the [T] and [Q+T] settings when λ is set to 0 (cf. Eq. (6) of the main text), which corresponds to omitting the discriminative loss surrogate $\hat{\ell}_{TD}$. In most of the cases, the results with $\lambda = 0.1$ outperforms $\lambda = 0$, showing the effectiveness of leveraging the source domain for discriminative learning. Also note that when D is provided for the target domain (i.e., [T+D] or [Q+T+D]), it is the $\hat{\ell}_{TD}$ term that utilizes the information of D, leading to better results than [T] or [Q+T], respectively.

We further experiment on different values of λ, as shown in Fig. 3. For [Q+T], we achieve consistent improvement for $\lambda \leq 0.1$. For [Q+T+D], we can get even better results by choosing a larger λ (e.g. $\lambda = 0.5$).

D.2. Domain adaptation using a subset of data

Following Table 5 of the main text, we include in Table 9 the results on the original VQA and Visual7W datasets, with target data sub-sampling by 1/16.

Notations Denote the features of Q, T, D by f_q, f_t, and f_d. *The D here stands for one decoy.*

Goal Learn transformations $g_q(\cdot)$, $g_a(\cdot)$ and a binary domain classifier WhichDomain$(\cdot)$, where ϕ_q, ϕ_a, and $\boldsymbol{\theta}$ are the parameters to learn, respectively. WhichDomain$(\cdot)$ gives the conditional probability of being from the source domain;

for *number of training iterations* **do**
    Initialize the parameters $\boldsymbol{\theta}$ of WhichDomain$(\cdot)$;
    for *k steps* **do**
        Sample a mini-batch of m pairs $\{Q_{\mathsf{SD}}^{(j)}, T_{\mathsf{SD}}^{(j)}\}_{j=1}^{m} \sim \mathsf{SD}$;
        Sample a mini-batch of m pairs $\{Q_{\mathsf{TD}}^{(j)}, T_{\mathsf{TD}}^{(j)}\}_{j=1}^{m} \sim \mathsf{TD}$;
        Update WhichDomain$(\cdot)$ by ascending its stochastic gradient;
$$\nabla_{\boldsymbol{\theta}} \left\{ \frac{1}{m} \sum_{j=1}^{m} \left[\log \mathsf{WhichDomain}(\{f_{q\mathsf{SD}}^{(j)}, f_{t\mathsf{SD}}^{(j)}\}) + \log(1 - \mathsf{WhichDomain}(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{t\mathsf{TD}}^{(j)})\})) \right] \right\}$$
    end
    for *l steps* **do**
        Sample a mini-batch of m triplet $\{Q_{\mathsf{TD}}^{(j)}, T_{\mathsf{TD}}^{(j)}, D_{\mathsf{TD}}^{(j)}\}_{j=1}^{m} \sim \mathsf{TD}$;
        Update the **transformations** by descending their stochastic gradients;
$$\nabla_{\phi_q, \phi_a} \left\{ \frac{1}{m} \sum_{i=1}^{m} \log(1 - \mathsf{WhichDomain}(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{t\mathsf{TD}}^{(j)})\})) + \right.$$
$$\left. \lambda \left(\ell(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{t\mathsf{TD}}^{(j)})\}) + \ell(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{d\mathsf{TD}}^{(j)})\}) \right) \right\}$$
    end
end

Algorithm 1: The proposed domain adaptation algorithm for Setting[Q+T+D]. $D_{\mathsf{TD}}^{(j)}$ denotes a single decoy. When the decoys of the target domain are not provided (i.e., Setting[Q+T]), the ℓ term related to $D_{\mathsf{TD}}^{(j)}$ is ignored.

Table 8. Domain adaptation (DA) results (in %) with or without the discriminative loss surrogate term

	original			
	VQA$^-$ → Visual7W		Visual7W → VQA$^-$	
Setting	[T]	[Q+T]	[T]	[Q+T]
$\lambda = 0$	54.1	54.1	29.2	28.8
$\lambda = 0.1$	54.5	55.2	29.7	29.4
	revised			
	VQA$^-$ → Visual7W		Visual7W → VQA$^-$	
Setting	[T]	[Q+T]	[T]	[Q+T]
$\lambda = 0$	47.8	47.8	45.9	45.7
$\lambda = 0.1$	47.6	48.4	45.9	45.8

Table 9. DA results (in %) on *original* datasets, with target data sub-sampling by 1/16. FT: fine-tuning. (best DA result in bold)

VQA$^-$ → Visual7W									
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within	FT
53.4	52.6	54.0	53.6	54.4	56.3	55.1	**58.2**	53.9	60.1

Visual7W → VQA$^-$									
Direct	[36]	[41]	[Q]	[T]	[T+D]	[Q+T]	[Q+T+D]	Within	FT
28.1	26.5	28.8	28.1	29.3	33.4	29.2	**35.2**	44.1	47.9

We further consider domain adaptation (under Setting[Q+T+D] with $\lambda = 0.1$) between Visual7W [50] and VQA$^-$ [4] for both the original and revised decoys using $\frac{1}{2^a}$ of training data of the target domain, where $a \in [0, 1, \cdots, 6]$. The results are shown in Fig. 4. Note that the **Within**

results are from models trained on the same sub-sampled size using the supervised IQA triplets from the target domain.

As shown, our domain adaptation (DA) algorithm is highly robust to the accessible data size from the target domain. On the other hand, the **Within** results from models training from scratch significantly degrade when the data size decreases. Except the case Visual7W → VQA$^-$ (original), domain adaptation (DA) using our algorithm outperforms the **Within** results after a certain sub-sampling rate. For example, on the case VQA$^-$ → Visual7W (revised), DA already outperforms **Within** under $\frac{1}{4}$ of the target data.

D.3. Sharing transformations degrades the performance

Although both questions and answers are text-based, they may have different degrees of domain mismatch (as shown in Table 1 of the main text). Ignoring such a fact and learning a single shared transformation degrades the performance. In Table 3 of the main text, the result on [Q+T+D] degrades from 56.2 to 55.6.

D.4. Results on sophisticated Visual QA models

Following Table 6 of the main text, we include in Table 10 the results of SMem [44] on the original datasets.

We further experiment with a variant of the HieCoAtt model [29] for Visual QA across datasets. See Sect. C.3 for more details. The results are shown in Table 11, where a similar trend of performance drop by **Direct** transfer and improvement by domain adaptation (in the [Q+T+D] setting) to those shown in the main text is observed.
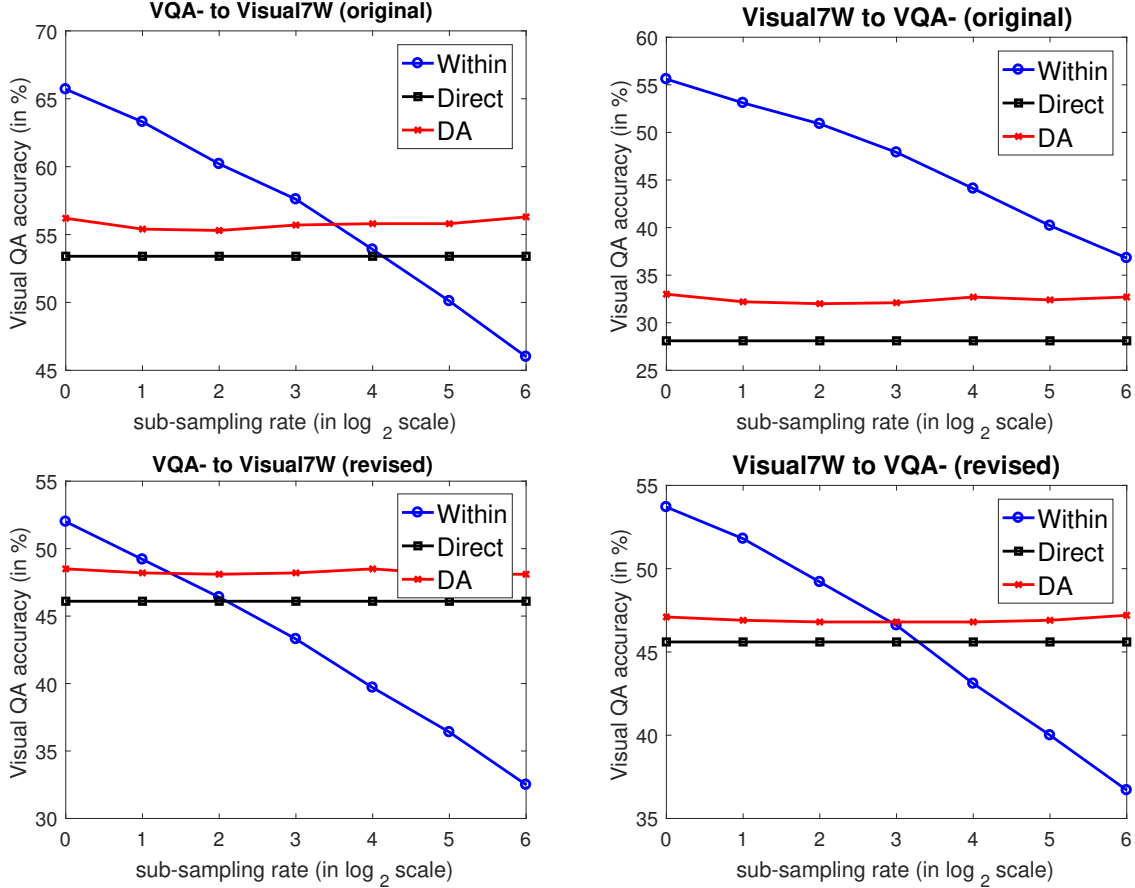
Figure 4. Domain adaptation (DA) results (in %) with limited target data, under Setting[Q+T+D] with $\lambda = 0.1$. A sub-sampling rate a means using $\dfrac{1}{2^a}$ of the target data.

Table 10. DA results (in %) on on *original* datasets using a variant of the SMem [44] model.

VQA$^-$ → Visual7W			Visual7W → VQA$^-$		
Direct	[Q+T+D]	Within	Direct	[Q+T+D]	Within
56.3	61.0	65.9	27.5	34.1	58.5

Table 11. DA results (in %) on VQA and Visual7W (both original and revised) using a variant of the HieCoAtt model [29].

original					
VQA$^-$ → Visual7W			Visual7W → VQA$^-$		
Direct	[Q+T+D]	Within	Direct	[Q+T+D]	Within
51.5	56.2	63.9	27.2	33.1	54.8
revised					
VQA$^-$ → Visual7W			Visual7W → VQA$^-$		
Direct	[Q+T+D]	Within	Direct	[Q+T+D]	Within
46.4	48.2	51.5	44.5	46.3	55.6

D.5. Open-ended (OE) results

We apply Visual QA models learned with the multiple-choice setting to evaluate on the open-ended one (i.e., select an answer from the top frequent

Table 12. OE results (VQA$^-$ → COCOQA, sub-sampled by 1/16).

Direct	[Q+T+D]	Within
16.7	24.0	26.9

ones, or from the set of all possible answers in the training data). The result on transferring from VQA$^-$ to COCOQA is in Table 12. Our adaptation algorithm still helps transferring.

D.6. Cross-dataset results across five datasets

Table 13 summarizes the results of the same study as in Sect. 5.4 of the main text, except that now **all** the training examples of the target domain are used. The models for **Within** are also trained on such a size, using the supervised IQA triplets.

Compared to Table 7 of the main text, we see that the performance drop of DA from using all the training examples of the target domain to 1/16 of them is very small (mostly smaller than 0.3%), demonstrating the robustness of our algorithm under limited training data. On the other hand, the drop of **Within** is much more significant—for most of the (source, target) pairs, the drop is at least 10%.

For most of the (source, target) pairs shown in Table 13, **Within** outperforms **Direct** and DA. The notable exceptions are (VG, Visual7W) and (VQA2$^-$, VQA$^-$). This is likely due to the fact that VG and Visual7W are constructed similarly while VG has more training examples than Visual7W. The same fact applies to VQA2$^-$ and VQA$^-$. Therefore, the

Table 13. Transfer results (in %) across datasets. The decoys are generated according to [6], where each IQT triplet is accompanied by 6 decoys (the accuracy of random guess is 14.3%). The setting for domain adaptation (DA) is on [Q+T+D] using **all** the training examples of the target domain.

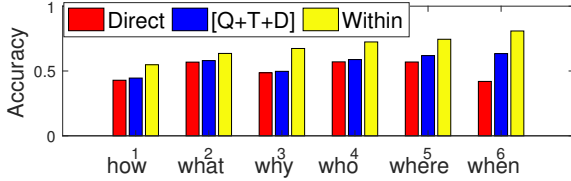| Training/Testing	Visaul7W [50]			VQA$^-$ [4]			VG [27]			COCOQA [33]			VQA2$^-$ [16]		
	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within
Visual7W [50]	52.0	-	-	45.6	48.1	53.7	49.1	49.6	58.5	58.0	63.0	75.8	43.9	45.6	53.8
VQA$^-$ [4]	46.1	49.3	52.0	53.7	-	-	44.8	47.9	58.5	59.0	64.7	75.8	50.7	50.6	53.8
VG [27]	58.1	58.4	52.0	52.6	54.4	53.7	58.5	-	-	65.5	68.8	75.8	50.1	51.5	53.8
COCOQA [33]	30.1	34.4	52.0	35.1	40.2	53.7	29.1	33.4	58.5	75.8	-	-	33.3	37.9	53.8
VQA2$^-$ [16]	48.8	51.0	52.0	55.2	55.3	53.7	47.3	49.6	58.5	60.3	65.2	75.8	53.8	-	-

Figure 5. Qualitative comparison on different type of questions, following the analysis of Sect. 5.3 of the main text when transferring from VQA− to Visual7W (on the original datasets).

Visual QA model learned on the source domain can be directly applied to the target domain and leads to better results than **Within**.

D.7. Qualitative results

Following the analysis of Sect. 5.3 of the main text, we shown in Fig 5 the results on each question type when transferring from VQA− to Visual7W (on the original datasets). [Q+T+D] outperforms **Direct** at all the question types.