Supplementary Material: 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

We provide contents omitted in the main text.

- Section 1: details on *Name that dataset!* (Sect. 3.2 of the main text).
- Section 2: details on the proposed domain adaptation algorithm (Sect. 4.2 and 4.3 of the main text).
- Section 3: details on the experimental setup (Sect. 5.2 of the main text).
- Section 4: additional experimental results (Sect. 5.3 and 5.4 of the main text).

1. *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 [5] as visual features to represent I and the average WORD2VEC embeddings [10] as text features to represent Q and each C $\in$ A, as in [6]. We represent the whole set of decoys (denoted as D) in A by the average of those decoys' features[1]. 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 [1] or Visual7W [16]).

2. Details on the Proposed Domain Adaptation Algorithm

2.1. 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 $\mathsf{WhichDomain}(\cdot)$ to detect the domain of a question Q, a correct answer T, or a QT pair, following the idea of Generative Adversarial Network [3]. The architecture of $\mathsf{WhichDomain}(\cdot)$ 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 $g_a(\cdot)$.

2.2. 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.

3. Details on the Experimental Setup

3.1. Implementation details

For all our experiments on training $g_q(\cdot)$, $g_a(\cdot)$, and $\mathsf{WhichDomain}(\cdot)$, we use Adam [7] 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.

3.2. Domain adaptation settings

Note that the "Yes" or "NO" issue we consider between VQA [1], VQA2 [4] and Visaul7W [16], Visual Genome

* Equal contributions

[1] Visual7W [16] has 3 decoys per triplet and VQA [1] 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.

```
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 $\mathsf{WhichDomain}(\cdot)$, where $\boldsymbol{\phi}_q$, $\boldsymbol{\phi}_a$, and $\boldsymbol{\theta}$ are the
  parameters to learn, respectively. $\mathsf{WhichDomain}(\cdot)$ gives the conditional probability of being from the source domain;

for number of training iterations do
    Initialize the parameters $\boldsymbol{\theta}$ of $\mathsf{WhichDomain}(\cdot)$;
    for k steps do
        Sample a mini-batch of $m$ pairs $\{Q^{(j)}_{\mathsf{SD}}, T^{(j)}_{\mathsf{SD}}\}^m_{j=1} \sim \mathsf{SD}$;
        Sample a mini-batch of $m$ pairs $\{Q^{(j)}_{\mathsf{TD}}, T^{(j)}_{\mathsf{TD}}\}^m_{j=1} \sim \mathsf{TD}$;
        Update $\mathsf{WhichDomain}(\cdot)$ 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 $\{Q^{(j)}_{\mathsf{TD}}, T^{(j)}_{\mathsf{TD}}, D^{(j)}_{\mathsf{TD}}\}^m_{j=1} \sim \mathsf{TD}$;
        Update the transformations by descending their stochastic gradients;
        $\nabla_{\boldsymbol{\phi}_q, \boldsymbol{\phi}_a} \left\{ \frac{1}{m} \sum_{i=1}^m \log(1 - \mathsf{WhichDomain}(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{t\mathsf{TD}}^{(j)})\})) +$
          $\lambda \left( \ell(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{t\mathsf{TD}}^{(j)})\}) + \ell(\{g_q(f_{q\mathsf{TD}}^{(j)}), g_a(f_{d\mathsf{TD}}^{(j)})\}) \right) \right\}$
    end
end
```

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

(VG) [8], COCOQA [11] is orthogonal to the one addressed in [4, 15], which deal with the prior of answers within a single dataset.

3.3. Sophisticated Visual QA models

In the main text we experiment with a variant of the spatial memory network (SMem) [14]. 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 [9], 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. 4.4. *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.*

4. Additional Experimental Results

4.1. The effect of the discriminative loss surrogate

We provide in Table 1 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 $\hat{\ell}_{\mathsf{TD}}$. 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 $\hat{\ell}_{\mathsf{TD}}$ 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

Table 1. Domain adaptation (DA) results (in %) with or without the discriminative loss surrogate term

original

	VQA$^-$ $\rightarrow$ Visual7W		Visual7W $\rightarrow$ 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$^-$ $\rightarrow$ Visual7W		Visual7W $\rightarrow$ 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

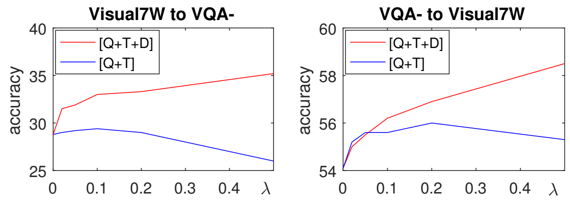

Figure 1. Results by varying λ on the original VQA and Visual7W datasets, for both the [Q+T] and [Q+T+D] settings.

Table 2. DA results (in %) on *original* datasets, with target data sub-sampling by 1/16. FT: fine-tuning. (best DA result in bold)

VQA$^-$ $\rightarrow$ Visual7W									
Direct	[12]	[13]	[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 $\rightarrow$ VQA$^-$									
Direct	[12]	[13]	[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

in Fig. 1. 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. $\lambda = 0.5$).

4.2. Domain adaptation using a subset of data

Following Table 5 of the main text, we include in Table 2 the results on the original VQA and Visual7W datasets, with target data sub-sampling by 1/16.

We further consider domain adaptation (under Setting[Q+T+D] with $\lambda = 0.1$) between Visual7W [16] and VQA$^-$ [1] 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. 2. 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 $\rightarrow$ VQA$^-$ (original), domain adaptation (DA) using our algorithm outperforms the **Within** results after a certain sub-sampling rate. For example, on the case VQA$^-$ $\rightarrow$ Visual7W (revised), DA already outperforms **Within** under $\frac{1}{4}$ of the target data.

Table 3. DA results (in %) on on *original* datasets using a variant of the SMem [14] model.

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

Table 4. DA results (in %) on VQA and Visual7W (both original and revised) using a variant of the HieCoAtt model [9].

original

VQA$^-$ $\rightarrow$ Visual7W			Visual7W $\rightarrow$ 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$^-$ $\rightarrow$ Visual7W			Visual7W $\rightarrow$ VQA$^-$		
Direct	[Q+T+D]	Within	Direct	[Q+T+D]	Within
46.4	48.2	51.5	44.5	46.3	55.6

Table 5. OE results (VQA$^-$ $\rightarrow$ COCOQA, sub-sampled by 1/16).

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

4.3. 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.

4.4. Results on sophisticated Visual QA models

Following Table 6 of the main text, we include in Table 3 the results of SMem [14] on the original datasets.

We further experiment with a variant of the HieCoAtt model [9] for Visual QA across datasets. See Sect. 3.3 for more details. The results are shown in Table 4, 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.

4.5. 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 ones, or from the set of all possible answers in the training data). The result on transferring from VQA$^-$ to COCOQA is in Table 5. Our adaptation algorithm still helps transferring.

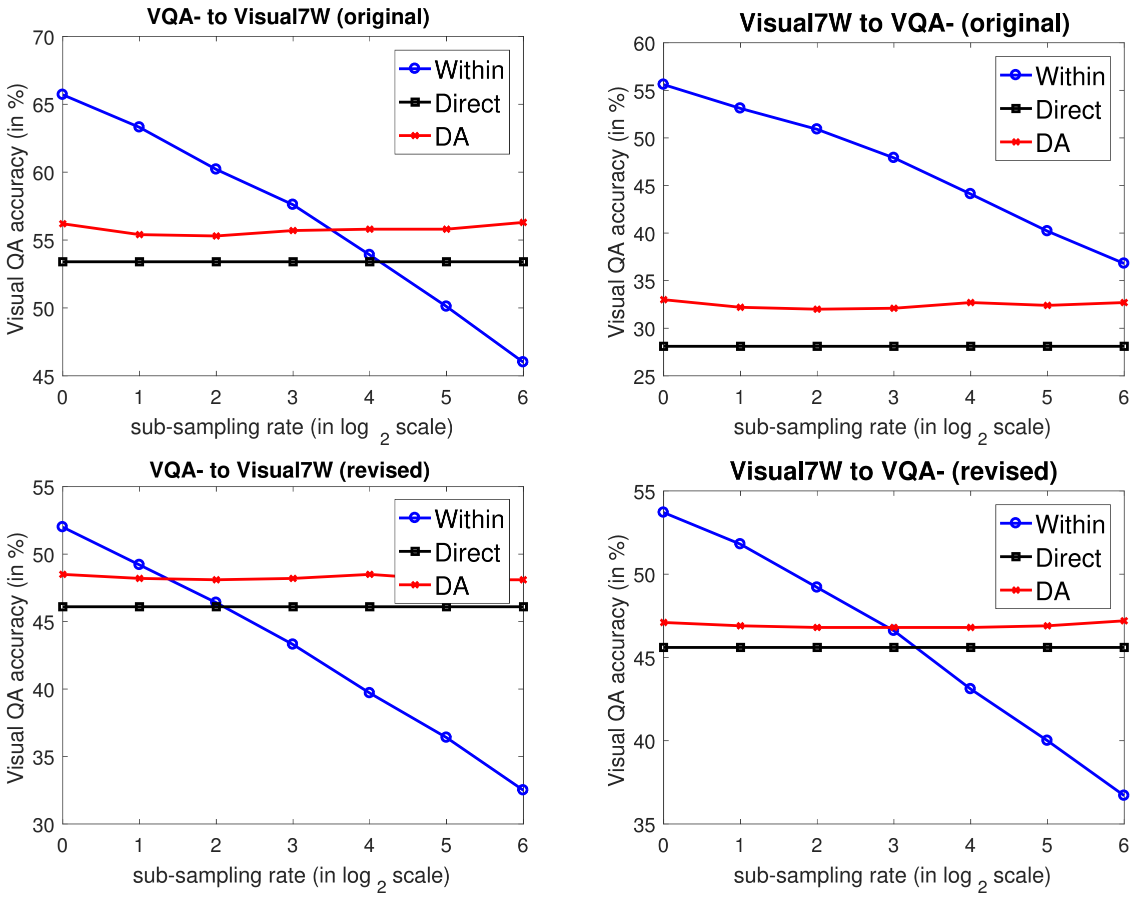

Figure 2. Domain adaptation (DA) results (in $\%$) with limited target data, under Setting[Q+T+D] with $\lambda = 0.1$. A sub-sampling rate a means using $\frac{1}{2^a}$ of the target data.

4.6. Cross-dataset results across five datasets

Table 6 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 6, **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 Visual QA model learned on the source domain can be directly applied to the target domain and leads to better results than **Within**.

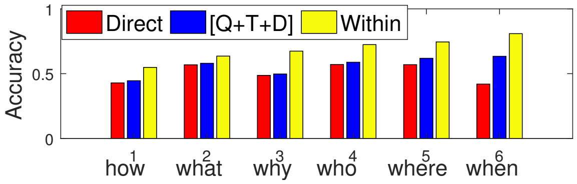

Figure 3. Qualitative comparison on different type of questions, following the analysis of Sect. 5.3 of the main text when transferring from VQA— to Visual7W (on the original datasets).

4.7. Qualitative results

Following the analysis of Sect. 5.3 of the main text, we shown in Fig 3 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.

Table 6. Transfer results (in %) across datasets. The decoys are generated according to [2], 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.

	Visaul7W [16]			VQA$^-$ [1]			VG [8]			COCOQA [11]			VQA2$^-$ [4]		
Training/Testing	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within	Direct	DA	Within
Visual7W [16]	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$^-$ [1]	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 [8]	58.1	58.2	52.0	52.6	53.7	53.7	58.5	-	-	65.5	67.0	75.8	50.1	51.5	53.8
COCOQA [11]	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$^-$ [4]	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	-	-

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1, 3, 5

[2] W.-L. Chao, H. Hu, and F. Sha. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL*, 2018. 5

[3] 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. 1

[4] 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. 1, 2, 5

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[6] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *ECCV*, 2016. 1

[7] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[8] 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. 2, 5

[9] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 2, 3

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 1

[11] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 2, 5

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

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

[14] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 2, 3

[15] 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

[16] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 3, 5