Appendix to Improving Selective Visual Question Answering by Learning from Your Peers

Corentin Dancette[1,3†*] Spencer Whitehead[1*] Rishabh Maheshwary[1] Ramakrishna Vedantam[1]
Stefan Scherer[2] Xinlei Chen[1] Matthieu Cord[3,4] Marcus Rohrbach[1]

[1]FAIR, Meta AI [2]Reality Labs Research, Meta [3]Sorbonne Université [4]Valeo.ai

Acknowledgements

From the Sorbonne Université side, this effort was partly supported by ANR grant VISADEEP (ANR-20-CHIA-0022). This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011011588R2 made by GENCI.

Index

Appendix A shows additional ablations for adding known OOD data, confirming the findings in the main paper.

Appendix B provides the result tables for all the ID/OOD mixtures, including the larger percentages of OOD examples.

Appendix C compares our staged training setup to jointly training VQA model and selector.

Appendix D has the details of the experimental setup, including model details and dataset splits.

Appendix E has the details of the OOD Detection features, which we use in the experiments in Table 3 of the main paper.

Appendix F offers a closer look at the difficulty in estimating the abstention threshold on in-domain data, when OOD data is present at test time.

Appendix G presents experiments using VizWiz [3] as the source of OOD data.

Appendix H presents selective prediction experiments on another multimodal task: SNLI-VE [16].

Appendix I illustrates qualitative results in Figs. 9 to 11.

*Equal contribution.
†Work primarily done during internship at FAIR.
Code: https://github.com/facebookresearch/selective-vqa_ood

A. Training with Known OOD Data

To further understand the usefulness of additional OOD data for Selector training in our multimodal setting as suggested in [7] for text-only NLP tasks, we provide an additional ablation: In Tab. 8 for the ID/OOD training setup, we train only on the B set + the OK-VQA training set, i.e. without VizWiz data (third line in each section of the table). The OK-VQA is more similar to AdVQA compared to VizWiz. However, we observe similar results compared to using both OOD datasets: Additional Known OOD does not consistently improve the results over the baseline (i.e. most identical to the selector setup in [15]), especially for low risk, the model with OK-VQA does not perform well. Alternative use of such known OOD data in the multimodal setting is out of scope for this work, but it is an interesting avenue for future work to study how to potentially better exploit such data.

Train Set f	Selector g	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
90% VQA v2, 10% AdVQA					
A	B	**19.00**	<u>41.64</u>	58.97	<u>9.34</u>
A	B + OOD	<u>18.48</u>	41.08	<u>59.40</u>	9.36
A	B + OK-VQA	18.38	**42.33**	**59.80**	**9.17**
50% VQA v2, 50% AdVQA					
A	B	**2.68**	**15.98**	<u>26.72</u>	<u>18.97</u>
A	B + OOD	<u>2.56</u>	14.93	**26.82**	19.08
A	B + OK-VQA	1.73	<u>15.37</u>	26.33	**18.86**

Table 8. Results with exposure to known OOD examples for OFA-Base. OOD = OK-VQA + VizWiz. **Bold** denotes best and <u>underline</u> is second best per table section.

B. Additional OOD Results

We show the AUC for our models on various mixtures of ID/OOD data in Fig. 5. Overall, our method consistently improves AUC over the baseline, for the three models (note lower is better for AUC).

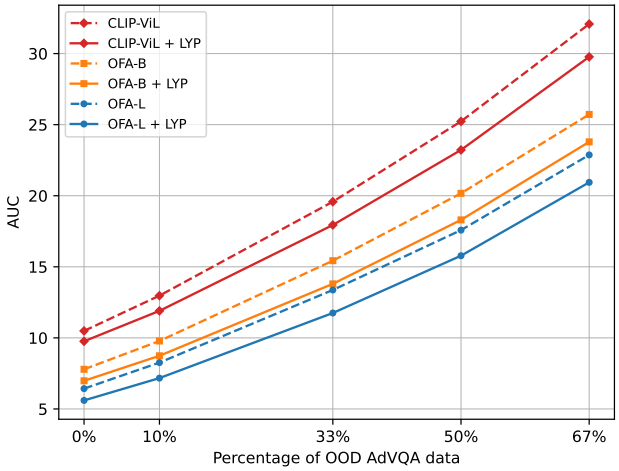

Figure 5. AUC for various mixtures of VQA v2 + AdVQA. Note: Lower is better for AUC. The baselines for each model is MaxProb.

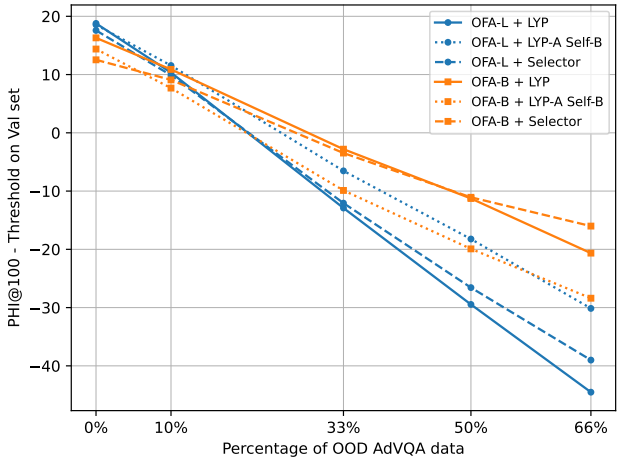

Figure 6. Φ_{100} scores for OFA-L and OFA-B models. The Selector is trained on B, based on the VQA model trained on A. We also show the alternative selector training strategy "LYP-A Self-B".

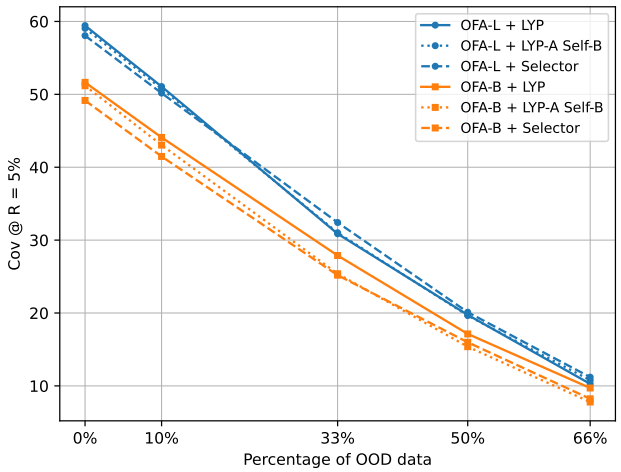

Figure 7. $\mathcal{C}@5\%$ scores for OFA-L and OFA-B models. The Selector is trained on B, based on the VQA model trained on A. We also show the alternative selector training strategy "LYP-A Self-B".

In Tabs. 15 to 19, we present the results for our experiments on the all OOD mixtures of VQA v2 and AdVQA [11]. While we on average see that the Selector models and Selector + LYP perform better than the corresponding baselines models out-of-the-box (MaxProb), all models degrade dramatically if there is a high percentage of OOD data in the test mixture, especially for low risk ($\mathcal{C}@1\%$) or high cost of error (Φ_{100}). Especially if we look at the realistic scenario where the threshold is chosen on the validation set and used at test time (as for Φ_{100}), we notice that the scores of all methods drop below 0 with 33.3% or more OOD data. This can be seen in the last column of Tabs. 17 to 19. These results demonstrate that these thresholds can be overconfident on OOD examples, which leads to poor abstention decisions such that the cost of the models' incorrect outputs outweighs the gains of the correct ones. Future work is needed to improve such OOD generalization and recognizing samples that cannot reliably be answered in this challenging setup, which this work provides a new and interesting test setup for.

B.1. Alternative LYP Strategy

As mentioned in the main paper, there are cases where LYP does not perform quite as well as the baseline Selector that is trained on held-out data. This happens with OFA-Large on high OOD levels, particularly with the Φ_{100} metric, which involves generalizing a confidence threshold chosen on ID data to test time where both ID and OOD data are present. This is shown in Figs. 6 and 7, where at higher percentages of OOD, OFA-Large with LYP (+LYP) can have lower $\mathcal{C}@5\%$ and Φ_{100} than the baseline Selector trained on held-out data (+Selector).

We propose a potential mitigation strategy for such cases. First, we use the VQA model trained only on the A subset, instead of the one trained on the full A+B like in LYP. Then, we train a Selector on the full A+B data, using the following strategy: we use LYP only on the A subset and use the model's own labels on the B subset. This allows the Selector to be trained partly on some data that was unseen during the VQA model's training, with real confidence labels. This potentially helps the selector capture the model's real uncertainty. We call this strategy **LYP-A Self-B** and report it in all Tabs. 15 to 19.

Fig. 6 illustrates the effect of this method on OFA-Base and OFA-Large for the Φ_{100} metric. We show that us-

ing this LYP-A Self-B strategy improves the Φ_{100} scores significantly for OFA-Large, surpassing both the baselines and the standard LYP. On OFA-Base, however, the base Selector and LYP perform similarly while LYP-A Self-B under-performs them. Therefore, it appears that this method should not be applied in all cases but might help to improve the results when LYP is less effective (e.g., because the VQA model overfits too much on the training set while not benefiting sufficiently from the additional training data in B).

C. Jointly Training OFA and Selector

Discussed in Appendix D, for training Selector, we follow a staged procedure [15]: The VQA model is first trained until convergence on the VQA task. Then, the weights are frozen, Selector is added to the model, and Selector is learned on top of the frozen model.

Since we are able to train OFA and Selector on the same data, a natural comparison to make is between the staged training procedure we use and joint training (i.e., simultaneously optimizing the VQA model and Selector), similar to [2]. We experiment with joint training by summing their losses. We perform this on OFA-Base, training both OFA-Base and Selector with the full A+B data. We also experiment with first joint training OFA-Base and Selector until OFA-Base has converged for the VQA task, freezing OFA-Base, and continuing to fine-tune Selector on A+B.

The results in Tab. 9 illustrate that joint training decreases the overall performance of the Selector. All metrics yield worse performance with joint training alone, though the gap shrinks when freezing the VQA model and continuing to fine-tune Selector. This is despite the fact that the overall VQA accuracy remains roughly the same with or without joint training. We conjecture that the reason for this may be that joint training creates a somewhat non-stationary optimization problem for Selector. Specifically, the VQA model's representations and VQA accuracy are changing throughout training. This means that the statistics of the inputs and training targets for Selector (see Appendix D) are changing, which may make optimizing Selector more difficult. Other techniques may be needed in order to properly optimize the VQA model and Selector together.

Training	Acc	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	AUC
ID (100% VQA v2)					
joint	75.08	16.04	42.78	65.91	8.11
joint+FT	75.08	24.42	50.01	69.20	7.21
staged	75.18	**26.64**	**50.80**	**69.56**	**7.10**
90% VQA v2, 10% AdVQA					
joint	71.97	10.74	34.61	53.81	10.12
joint+FT	71.97	18.17	42.44	60.50	8.98
staged	72.00	**19.72**	**42.70**	**60.84**	**8.90**

Table 9. Comparison of joint and staged training of OFA-Base and Selector. FT indicates that Selector is further fine-tuned after OFA-Base converges on the VQA training objective. All models are trained on A+B.

D. Experimental Setups

D.1. Models

D.1.1 LYP Peer Models

Our LYP approach requires training *peer* models to label the training data for the full Selector. For all LYP peer models, we simply follow the corresponding VQA model training settings. Once trained, we run inference on the respective held-out sets for each peer to obtain labels.

D.1.2 CLIP-ViL

We use the implementations for MaxProb and Selector provided by [15].[2] For the CLIP-ViL MaxProb and Selector models trained on held-out data (i.e., Train B), which exactly match the setup of [15], we use the model weights given by the authors as well. Note that the available model weights are for a single run, whereas the results in [15] are averaged over ten runs, so there are some variations in numbers between those reported in this work and [15]. Additionally, we compare to the results in the arXiv version of [15] as this has updated results (see appendix of [15]). For the remaining CLIP-ViL models, we train them following the provided hyperparameters and settings. We refer readers to [15] for details.

D.1.3 OFA

OFA first processes the image using a convolutional network [4] to obtain a set of visual representations $\tilde{V}$. Likewise, the question is tokenized and converted to a sequence of question token embeddings $\tilde{Q}$. Then, the visual features are flattened into a sequence and concatenated with the question token embedding sequence. This entire sequence is given as input to an encoder-decoder transformer model [13] to predict the answers. The encoder produces multimodal representations of the image tokens $\{v_i\}_{i=1}^{|\tilde{V}|}$ and question tokens $\{q_j\}_{j=1}^{|\tilde{Q}|}$. The encoded tokens are used as input to cross-attention layers in the transformer decoder at each decoding step. The decoder generates output token representations $\{o_l\}_{l=1}^{L}$ for an answer of L tokens. These token representations can be fed to a linear layer to give the output logits over the token vocabulary. We use beam search to decode the answers.

We fine-tune OFA from the pre-trained checkpoints pro-

[2] https://github.com/facebookresearch/reliable_vqa

	OFA-Base	OFA-Large
Batch Size	256	512
Learning Rate	1e-4	4e-4
LR warmup	no	no
LR-decay (linear)	-1e-10/step	-1e-10/step
Optimizer	Adam	Adam
Optimizer Beta	(0.9,0.999)	(0.9,0.999)
Gradient clipping	1.0	1.0
Selector Dropout	0.1	0.1
Main model dropout	0.1	0.1
Image size	480	480

Table 10. Hyperparameters for Selector Training on top of OFA

vided by the authors of [14].[3] We follow the hyperparameters from the original paper for fine-tuning. In the following, we detail the setup for the selection functions:

MaxProb. Since OFA is a sequence-to-sequence model that generates answers token-by-token, for the MaxProb baseline, we use the joint probability of each answer token as the confidence value, similar to common decoding algorithms like beam search.

Selector. We largely replicate the same Selector architecture and training as [15] (i.e., two-layer MLP), but with some slight differences. We remove the non-linear projection (or one-layer MLP) for each input representation. We also use slightly different input representations: First, we max-pool the encoder image (v_i) and question (q_i) token representations to obtain a single representation for each set of representations. Then, we extract the probability of the predicted answer p, which is the joint probability of each answer token. Finally, we extract the first output token embedding o_1 that is used to predict the first answer token. We concatenate these representations and feed this as input to the Selector.

Training Selector with OFA. We report the training parameters in Tab. 10. We first train the VQA model as discussed above, freeze the VQA model, and then train Selector on top of this frozen model, following [15]. We train for a maximum number of 32 epochs and perform early-stopping on the Val split (Tab. 11) using the AUC metric. We keep the dropout in the main model during the selector training, as we found this improved performance of the selector.

D.2. Dataset Splits

D.2.1 In-Distribution Splits

We follow [15] and use the splits provided in the official implementation. We detail the splits again in Tab. 11. Note, in our work we repurpose the "Dev" set from [15] for our Train B split. No images (or question-answer annotations) are shared between splits.

[3] https://github.com/OFA-Sys/OFA

Split	Usage	Source	%src	#I	#Q
Train A	Train f,g	VQA v2 train	100%	82,783	443,757
Train B	Train f, g	VQA v2 val	40%	16,202	86,138
Val	Validate f, g	VQA v2 val	10%	4,050	21,878
Test	Test $h = (f, g)$	VQA v2 val	50%	20,252	106,338

Table 11. Size of the splits of VQA v2 from [15]. Note, the "Usage" is the setting for the full model (A+B). Some models are trained on subsets (e.g., just A) as specified in the corresponding tables.

D.2.2 ID/OOD Mixtures

We use AdVQA as our source of OOD data. As discussed, AdVQA is an adversarial dataset where human annotators intentionally ask questions that state-of-the-art models trained on VQA v2 answer incorrectly. The images in AdVQA come from [9], as do VQA v2. However, we consider this as OOD since the questions are adversarial in nature and contain distribution shifts meant to induce errors for models trained on VQA v2.

In our work, we create mixtures of ID/OOD examples for our evaluations. To form our mixtures, we first discard all AdVQA images that overlap with the A+B train set. This leaves 5,008 AdVQA examples. For each setting, we randomly sample examples from the ID Test split (Tab. 11) to create the desired OOD proportion: 45K for 10% OOD, 10K for 33% OOD, 5K for 50% OOD and 2.5K for 66% OOD.

E. OOD Detection Features

In Table 3 of the main paper, we experiment with OOD detection features as additional input to the selector, inspired by [1]. To compute those metrics, we use the representations from the encoder of OFA. We average the output question tokens q_i and the image tokens v_i, which respectively yield $\bar{q}$ and $\bar{v}$. We compute OOD detection features for each representation with respect to the training data. The computed features are as follows:

kNN [12]. Given an input example, we compute the cosine distance to the k nearest neighbors in the training data. This distance is used as an OOD score: higher scores signify more "in-distribution" examples, while lower scores signify "out-of-distribution". We use the efficient vector-search library FAISS [5] to compute the distances and identify the k closest points. We experimented with various numbers of neighbors from 1 to 1000 and found no significant improvements for any value. We also experimented with using the distance to *correct* and *incorrect* neighbors, to align the distances to our task of selective prediction.

SSD [10]. SSD [10] is a parametric OOD-detection method that first builds k clusters in feature space and then fits a multivariate normal distribution for each of the k ensembles of features. For a new example, the Mahalanobis distance [8] to this normal distribution is used as an OOD score. Note that for a classification task, the labels might be used as clusters, but we prefer to use a cluster-based algorithm, as the VQA answers do not represent a coherent ensemble of image or question concepts. We experimented with various numbers of clusters in the range of $[1, 1000]$, and saw no improvements.

For these OOD detection features, we give them as additional inputs to the Selector to provide a signal for whether a given example is ID or OOD.

F. Threshold Generalization

In this section, we investigate threshold generalization. All previous tables reported numbers on "maximum coverage" at risk $\mathcal{R}$. This metric is irrespective of the threshold chosen as it solves for the coverage that satisfies a given risk level. In a real-world setting, the threshold would need to be fixed once using a validation set and then used at test time. We already evaluate this setting of evaluating the optimal threshold on the validation set for the cost-based metric Φ_c in the main paper. In contrast to Φ_c, which allows comparing a single number, for risk and coverage, choosing a threshold on a validation set leads to changes in coverage *and* risk, making it difficult to compare two methods. Still, in this section, we evaluate how the threshold generalizes to ID and OOD settings.

Our method improves risk generalization over out-of-the-box MaxProb. In Fig. 8, we show the test risk on various ID/OOD mixtures with a threshold set on the ID validation split of VQA v2 for a target risk of 1%. We see that LYP (solid line) consistently improves the generalization of risk over the MaxProb baseline: The curves corresponding to LYP are closer to the 1% target risk level compared to MaxProb.

Risk generalization is limited for OOD data. While we observe reasonable good risk generalization for ID, the generalization is really limited for larger percentages of OOD data.

CLIP-ViL is the best model for risk generalization. We see that all variants of CLIP-ViL outperform their corresponding methods on OFA-B and OFA-L. Note that the associated coverages are lower for the same risk level, thus CLIP-ViL is not the best method overall. This is somewhat surprising, as [6] found that larger language models were better calibrated on NLP tasks.

Full results are available in Tab. 20 and Tab. 21 for our in-distribution testing set and our mixed setting with 90% of VQA v2 and 10% of AdVQA examples.

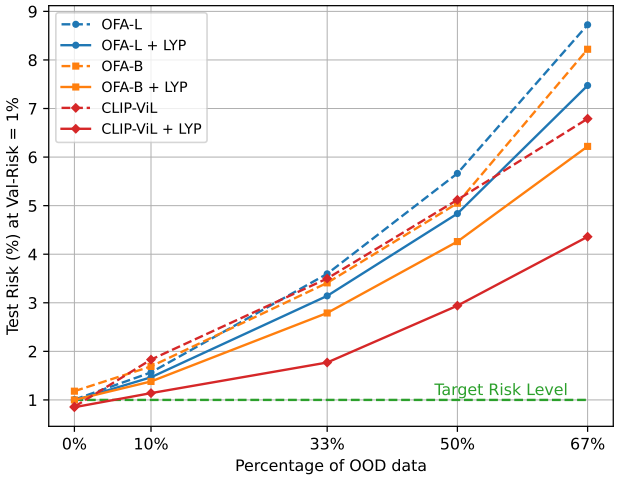

Figure 8. Risk at various percentages of OOD when the threshold is optimized on the validation set for maximum coverage, with a target risk level of 1%. The baseline for each model is MaxProb.

			10% OOD				
Model	f train	g train	AUC	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	Acc.
MaxProb	A	–	10.61	9.44	37.4	54.30	69.02
Selector	A	B	9.14	25.76	46.57	62.07	69.02
Selector + LYP	AB	AB	**5.79**	**36.50**	**59.61**	**76.20**	76.12

Table 12. OFA-Base results with 90% **VQAv2** and 10% **VizWiz** data. For LYP, the VQA model is trained on A+B and selector on A+B with annotations from 10 models.

			50% OOD				
Model	f train	g train	AUC	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	Acc.
MaxProb	A	–	25.53	0.00	14.55	23.49	48.07
Selector	A	B	22.83	9.31	21.87	32.02	48.07
Selector + LYP	AB	AB	**22.71**	**11.81**	**22.51**	**32.18**	48.14

Table 13. OFA-Base results with 50% **VQAv2** and 50% **VizWiz** data. For LYP, the VQA model is trained on A+B and selector on A+B with annotations from 10 models.

G. VizWiz as OOD Data Source

We show in Tabs. 12 and 13 results for our LYP method on another OOD dataset: VizWiz [3]. This dataset is much more different from the original VQA v2 than AdVQA: the image and questions were collected by visually impaired users using a smartphone. Therefore, this makes it easier for models to discriminate between VQA and VizWiz examples. We see that overall, the coverages are much higher for these setups than with AdVQA. We also see that our LYP method is very efficient to improve the results over the regular Selector model setup from [15].

f train	Model	g train	AUC	$\mathcal{C}$@1%	$\mathcal{C}$@5%	$\mathcal{C}$@10%	Acc
A	MaxProb	–	10.10	6.69	28.08	49.75	77.31
	Selector	B	8.86	13.19	38.51	59.29	77.24
AB	MaxProb	–	8.78	12.04	35.44	59.56	77.88
	Selector	AB (Self)	8.49	12.62	37.69	61.64	77.91
	Selector	AB (LYP)	**8.22**	**16.57**	**40.02**	**62.90**	77.91

Table 14. OFA-Base results on SNLI-VE (without textual premise).

H. Selective Prediciton for Visual Entailment

We show experiments for our models and LYP on the visual entailment dataset SNLI-VE [16] in Tab. 14. Given an image premise, and a text hypothesis, the model has to return one of the three possible outputs: *entailment*, *neutral*, or *contradiction*. We run the experiments on OFA-Base, and use the same setup as the original OFA paper [14], except that we do not use the textual premise to make it comparable to previous works. We divide the SNLI-VE training set into 80% for the A split, and the remaining 20% for the B split. We use the original validation and test splits for model selection and test.

We see that LYP is very effective on this task: it improves the coverage across all risk levels compared to MaxProb and Selector baselines.

I. Qualitative Examples

Figs. 9 to 11 show qualitative results comparing the OFA-Large + LYP and OFA-Large + MaxProb, on the AdVQA dataset. In both cases, the OFA-Large model f is trained on A+B. For all examples, the abstention threshold is set on the in-distribution validation set to get maximum coverage at 5% risk.

Fig. 9 shows examples where the VQA model (OFA-Large) is incorrect. Thus, the correct behavior is to abstain. But the MaxProb model does not abstain using the provided threshold, instead, it answers incorrectly. On the contrary, our model OFA-L + LYP abstains.

Fig. 10 shows examples where the OFA-L model is correct: the best behavior is to answer. The MaxProb model abstains, while our method answers correctly.

Fig. 11 shows two kinds of failure cases of our models: In the first line, OFA-L + LYP incorrectly abstains, as the VQA model was correct. In the second line, our model incorrectly answer instead of abstaining, as the answer provided by the model was incorrect.

VQA Model f		Selection func. g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Train Set	Name	Train Set	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	- [15]	69.98	4.97	33.79	53.62	10.92	54.67	21.40	1.32
		Selector	B	Self [15]	69.98	15.79	37.79	55.65	10.21	55.44	25.82	8.74
	A+B	MaxProb	-	-	70.72	5.54	34.84	55.04	10.49	55.93	22.81	2.59
		Selector	A+B	Self	70.72	6.45	34.26	56.07	10.48	56.07	22.99	2.39
		Selector	A+B	LYP	70.72	**18.40**	**38.65**	**57.40**	**9.76**	**56.53**	**26.45**	**9.74**
OFA-Base	A	MaxProb	-	-	74.87	3.45	45.60	66.61	7.99	62.52	30.57	6.81
		Selector	B	Self	74.87	23.78	49.16	69.00	7.32	63.03	34.39	12.53
		Selector	A+B	LYP-A Self-B	74.87	26.03	51.18	69.97	7.13	63.48	35.91	14.38
	A+B	MaxProb	-	-	75.18	14.88	46.15	67.51	7.79	63.04	30.13	7.29
		Selector	A+B	Self	75.18	26.64	50.80	69.56	7.10	63.66	34.92	12.92
		Selector	A+B	LYP	75.18	**27.71**	**51.64**	**70.20**	**6.98**	**63.88**	**36.29**	**16.30**
OFA-Large	A	MaxProb	-	-	77.53	20.57	53.99	75.18	6.42	66.68	36.12	8.21
		Selector	B	Self	77.53	30.86	58.05	76.65	5.81	67.34	41.43	17.58
		Selector	A+B	LYP-A Self-B	77.53	32.05	59.05	77.10	5.69	67.61	42.10	18.55
	A+B	MaxProb	-	-	77.79	16.31	53.83	75.27	6.43	66.96	36.06	6.29
		Selector	A+B	Self	77.79	31.47	58.80	77.14	5.69	67.82	41.43	16.08
		Selector	A+B	LYP	77.79	**32.92**	**59.43**	**77.52**	**5.60**	**68.02**	**42.83**	**18.78**

Table 15. Risk-coverage metrics and effective reliability on ID data (i.e., VQA v2 test split [15]). Scores for OFA-Large are averaged over 5 trials. This table is a copy from the main paper with the additional lines "LYP-A Self-B", discussed in Appendix B.1.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Train Set	Name	Train Set	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	- [15]	66.35	0.00	24.16	43.53	13.55	49.12	14.39	-4.64
		Selector	B	Self [15]	66.35	12.69	31.12	46.96	12.47	50.36	20.15	5.22
	A+B	MaxProb	-	-	67.12	2.60	26.13	45.25	12.97	50.49	16.59	-0.93
		Selector	A+B	Self	67.12	2.97	26.70	46.19	12.80	50.89	18.19	-0.65
		Selector	A+B	LYP	67.12	**15.22**	**32.58**	**49.18**	**11.90**	**51.43**	**22.09**	**7.12**
OFA-Base	A	MaxProb	-	-	71.59	0.01	36.07	56.49	10.10	57.49	23.15	-0.34
		Selector	B	Self	71.59	18.32	41.48	59.74	9.19	57.97	27.22	9.09
		Selector	A+B	LYP-A Self-B	71.61	19.49	43.04	61.04	9.00	58.43	29.23	7.68
	A+B	MaxProb	-	-	72.00	1.74	37.02	57.57	9.78	58.11	22.09	0.53
		Selector	A+B	Self	72.00	19.72	42.70	60.84	8.90	58.90	28.05	2.88
		Selector	A+B	LYP	72.00	**21.58**	**44.09**	**61.69**	**8.74**	**59.11**	**28.79**	**10.88**
OFA-Large	A	MaxProb	-	-	74.56	4.76	44.53	66.06	8.21	61.90	28.20	0.21
		Selector	B	Self	74.56	23.53	50.17	68.76	7.33	62.96	34.43	9.88
		Selector	A+B	LYP-A Self-B	74.56	24.34	50.78	69.46	7.25	63.15	34.79	**11.55**
	A+B	MaxProb	-	-	74.79	1.30	43.70	65.95	8.26	62.24	27.09	-2.46
		Selector	A+B	Self	74.79	22.68	50.27	69.27	7.32	63.03	33.50	4.92
		Selector	A+B	LYP	74.79	**25.38**	**51.07**	**69.74**	**7.17**	**63.41**	**34.85**	10.34

Table 16. Mixed ID/OOD scenario, composed of 90% VQA v2 and 10% AdVQA examples. This table is a copy from the main paper with the additional lines "LYP-A Self-B", discussed in Appendix B.1.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Train Set	Name	Train Set	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	- [15]	58.36	0.00	7.08	21.97	20.62	36.59	-1.47	-14.38
		Selector	B	Self [15]	58.36	5.87	17.41	29.21	18.90	38.76	7.11	-2.20
	A+B	MaxProb	-	-	59.29	1.11	10.17	24.99	19.58	38.42	2.99	-9.79
		Selector	A+B	Self	59.29	0.07	11.21	25.86	19.28	39.17	5.90	-7.37
		Selector	A+B	LYP	59.29	**7.07**	**19.13**	**31.53**	**17.94**	**39.85**	**12.67**	**3.40**
OFA-Base	A	MaxProb	-	-	64.17	0.01	18.83	34.15	15.71	46.05	5.33	-28.66
		Selector	B	Self	64.17	6.97	25.19	39.53	14.44	46.41	**11.98**	-3.47
		Selector	A+B	LYP-A Self-B	64.16	6.76	25.39	41.49	14.26	46.99	14.42	-9.89
	A+B	MaxProb	-	-	64.63	0.03	17.57	33.94	15.43	46.32	2.11	-19.21
		Selector	A+B	Self	64.63	5.11	25.83	40.13	14.09	47.58	10.75	-21.18
		Selector	A+B	LYP	64.63	**9.41**	**27.89**	**42.00**	**13.80**	**48.03**	11.89	**-2.81**
OFA-Large	A	MaxProb	-	-	67.78	0.39	22.67	43.82	13.05	50.88	10.05	-20.68
		Selector	B	Self	67.79	9.18	32.42	50.06	**11.60**	52.74	18.31	-12.07
		Selector	A+B	LYP-A Self-B	67.79	**11.24**	**31.01**	**50.37**	11.65	**52.76**	**18.47**	**-6.53**
	A+B	MaxProb	-	-	67.78	0.13	21.01	42.31	13.37	51.02	7.58	-26.60
		Selector	A+B	Self	67.77	6.58	30.15	48.83	11.98	51.79	15.37	-23.92
		Selector	A+B	LYP	67.77	9.44	30.87	49.69	11.75	52.51	16.95	-12.91

Table 17. Results on a mixed ID/OOD setting, composed of 66.7% VQA v2 data (Test split in Tab. 11) and 33.3% AdVQA examples. Discussion in Appendix B.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Train Set	Name	Train Set	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	- [15]	52.66	0.00	3.08	9.77	26.57	27.65	-13.20	-20.60
		Selector	B	Self [15]	52.66	**4.19**	10.29	18.17	24.49	30.62	-2.24	-7.94
	A+B	MaxProb	-	-	53.83	0.97	3.66	12.27	25.23	29.82	-6.40	-15.61
		Selector	A+B	Self	53.83	0.04	5.52	13.38	24.96	30.82	-2.96	-11.50
		Selector	A+B	LYP	53.83	3.41	**11.19**	**20.42**	**23.22**	**31.85**	**5.49**	**-0.04**
OFA-Base	A	MaxProb	-	-	59.17	0.01	5.78	18.48	20.80	38.14	-6.11	-29.98
		Selector	B	Self	59.18	3.21	15.98	26.27	19.00	38.49	0.10	-11.07
		Selector	A+B	LYP-A Self-B	59.18	2.28	15.38	26.72	18.80	39.24	**3.46**	-19.92
	A+B	MaxProb	-	-	59.61	0.06	6.91	20.86	20.17	38.45	-12.19	-31.48
		Selector	A+B	Self	59.62	2.29	15.78	27.38	18.70	39.88	-0.74	-36.10
		Selector	A+B	LYP	59.62	**3.98**	**17.13**	**28.53**	**18.30**	**40.35**	-0.49	**-11.27**
OFA-Large	A	MaxProb	-	-	63.02	0.31	11.53	27.85	17.18	43.42	-3.11	-34.01
		Selector	B	Self	63.01	**5.56**	**20.11**	**35.51**	**15.52**	45.49	**6.48**	-26.57
		Selector	A+B	LYP-A Self-B	63.01	5.07	19.65	34.80	15.61	**45.55**	6.11	**-18.23**
	A+B	MaxProb	-	-	62.93	0.12	6.22	26.58	17.58	43.40	-6.02	-40.93
		Selector	A+B	Self	62.93	1.00	18.55	33.48	16.03	44.14	2.57	-43.03
		Selector	A+B	LYP	62.93	3.51	19.74	34.18	15.78	45.03	4.19	-29.46

Table 18. Results on a mixed ID/OOD setting, composed of 50% VQA v2 data (Test split in Tab. 11) and 50% AdVQA examples. Discussion in Appendix B.

VQA Model f		Selection function g			Acc ↑	$\mathcal{C}@\mathcal{R}$ in % ↑			AUC ↓	Φ_1	Φ_{10}	Φ_{100}
Name	Train Set	Name	Train Set	Targets		$\mathcal{C}@1\%$	$\mathcal{C}@5\%$	$\mathcal{C}@10\%$				
CLIP-ViL	A	MaxProb	-	- [15]	46.66	0.00	0.00	3.04	33.67	18.32	-24.68	-28.56
		Selector	B	Self [15]	46.66	1.91	5.65	10.09	31.43	22.00	-11.50	-12.05
	A+B	MaxProb	-	-	47.94	0.67	1.28	5.59	32.08	20.87	-16.85	-21.99
		Selector	A+B	Self	47.94	0.05	1.44	5.49	31.79	22.20	-11.69	-15.28
		Selector	A+B	LYP	47.94	**2.13**	**6.60**	**10.44**	**29.77**	**23.60**	**-0.77**	**-0.89**
OFA-Base	A	MaxProb	-	-	53.71	0.00	0.45	8.44	26.47	29.64	-17.60	-43.15
		Selector	B	Self	53.77	2.00	8.23	15.97	24.45	30.21	-10.46	-16.01
		Selector	A+B	LYP-A Self B	53.77	1.73	7.79	15.51	24.35	30.86	**-7.20**	-28.39
	A+B	MaxProb	-	-	54.28	0.03	0.53	10.16	25.72	29.96	-25.56	-44.75
		Selector	A+B	Self	54.26	1.52	8.79	16.23	24.15	31.88	-12.68	-52.56
		Selector	A+B	LYP	54.26	**1.95**	**9.71**	**17.11**	**23.79**	**32.38**	-12.12	**-20.65**
OFA-Large	A	MaxProb	-	-	57.69	0.13	3.65	14.24	22.36	34.91	-16.36	-49.70
		Selector	B	Self	57.71	**3.03**	**11.20**	**22.04**	**20.44**	37.45	**-5.27**	-39.01
		Selector	A+B	LYP-A Self-B	57.71	1.89	10.78	20.09	20.63	**37.51**	-6.12	**-30.14**
	A+B	MaxProb	-	-	57.52	0.08	0.54	13.41	22.87	34.70	-20.37	-56.21
		Selector	A+B	Self	57.50	0.46	9.02	20.14	21.10	35.39	-10.72	-61.53
		Selector	A+B	LYP	57.50	0.08	10.28	19.93	20.94	36.60	-8.58	-44.52

Table 19. Results on a mixed ID/OOD setting, composed of 33.3% VQA v2 data (Test split in Tab. 11) and 66.7% AdVQA examples. Discussion in Appendix B.

VQA Model f		Selection function g			Acc ↑	$\mathcal{R} = 1\%$		$\mathcal{R} = 5\%$		$\mathcal{R} = 10\%$	
Name	Train Set	Name	Train Set	Targets		$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$
CLIP-ViL	A	MaxProb	-	- [15]	69.98	0.86	3.49	4.55	31.59	9.60	52.35
		Selector	B	Self [15]	69.98	0.72	13.26	4.74	36.66	9.97	55.58
	A+B	MaxProb	-	-	70.72	1.08	6.67	4.59	32.85	9.83	54.47
		Selector	A+B	Self	70.72	1.10	7.60	4.78	34.16	9.73	54.63
		Selector	A+B	LYP	70.72	0.85	16.78	4.96	38.30	10.08	57.34
OFA-Base	A	MaxProb	-	-	74.87	1.18	5.32	4.96	45.45	9.96	66.44
		Selector	B	Self	74.87	1.05	24.54	5.07	49.53	10.18	69.67
	A+B	MaxProb	-	-	75.18	0.82	4.32	4.98	46.03	10.08	67.88
		Selector	A+B	Self	75.18	1.14	27.88	5.23	51.76	10.09	69.87
		Selector	A+B	LYP	75.18	1.00	27.84	5.17	52.44	10.35	71.31
OFA-Large	A	MaxProb	-	-	77.53	0.99	20.46	4.95	53.73	9.80	74.51
		Selector	B	Self	77.53	1.10	32.01	5.04	58.23	9.98	76.63
	A+B	MaxProb	-	-	77.80	1.01	15.06	4.85	53.11	9.83	74.67
		Selector	A+B	Self	77.79	1.00	31.45	4.94	58.57	9.97	77.08
		Selector	A+B	LYP	77.79	0.99	32.79	4.99	59.39	10.05	77.67

Table 20. Results on the ID VQA v2 evaluation set (Test split in Tab. 11). Thresholds for desired risk level are selected on the in-distribution Val split. Discussion in Appendix F.

VQA Model f		Selection function g			Acc ↑	$\mathcal{R} = 1\%$		$\mathcal{R} = 5\%$		$\mathcal{R} = 10\%$	
Name	Train Set	Name	Train Set	Targets		$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$	$\mathcal{R}$	$\mathcal{C}$
CLIP-ViL	A	MaxProb	-	- [15]	66.35	1.83	3.21	6.25	29.53	12.05	50.06
		Selector	B	Self [15]	66.35	0.95	12.14	5.75	33.92	11.78	52.33
	A+B	MaxProb	-	-	67.12	1.59	6.11	5.97	30.70	12.02	52.14
		Selector	A+B	Self	67.12	1.52	6.97	6.04	31.95	11.63	51.83
		Selector	A+B	LYP	67.12	1.14	15.26	5.81	35.46	11.72	54.08
OFA-Base	A	MaxProb	-	-	71.59	1.69	4.88	6.54	43.00	12.11	64.13
		Selector	B	Self	71.60	1.43	22.60	6.19	46.18	12.23	67.04
	A+B	MaxProb	-	-	72.00	1.30	3.95	6.56	43.59	12.05	65.67
		Selector	A+B	Self	72.02	1.60	25.72	6.49	48.75	11.82	67.13
		Selector	A+B	LYP	72.01	1.38	25.61	6.27	48.97	12.07	68.25
OFA-Large	A	MaxProb	-	-	74.56	1.58	18.93	6.50	51.43	11.96	73.01
		Selector	B	Self	74.56	1.56	29.66	6.23	55.37	11.84	74.44
	A+B	MaxProb	-	-	74.79	1.57	13.90	6.50	50.93	12.00	73.16
		Selector	A+B	Self	74.79	1.52	29.05	6.34	55.91	11.90	75.17
		Selector	A+B	LYP	74.79	1.47	30.17	6.29	56.31	11.99	75.66

Table 21. Results on the mixed 90% VQA v2 + 10% AdVQA evaluation set (VQA v2 data is from the Test split in Tab. 11). Thresholds for desired risk level are selected on our in-distribution Val set. Discussion in Appendix F.

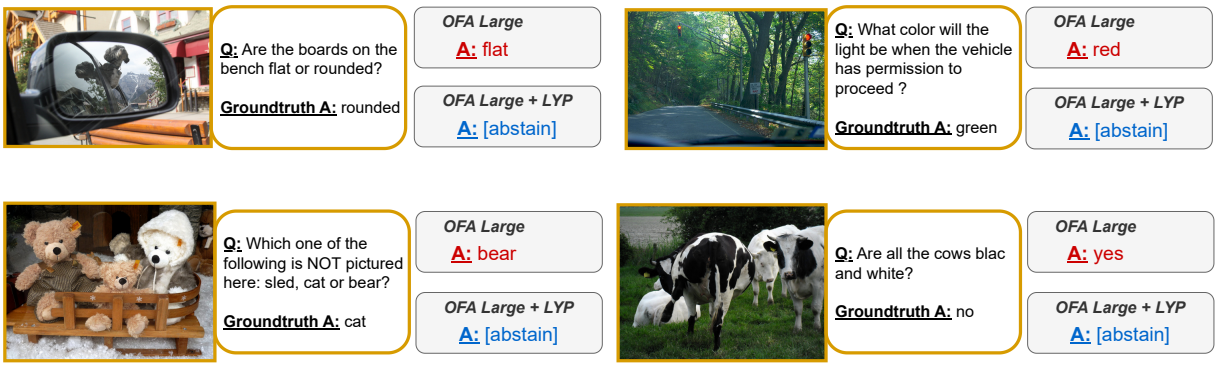

Figure 9. Qualitative examples for OFA-Large on AdVQA: On those examples, the baseline (MaxProb) answers incorrectly the answer, and our model with LYP abstains. For both models, the threshold is selected on in-distribution data for maximum coverage at 5% risk.

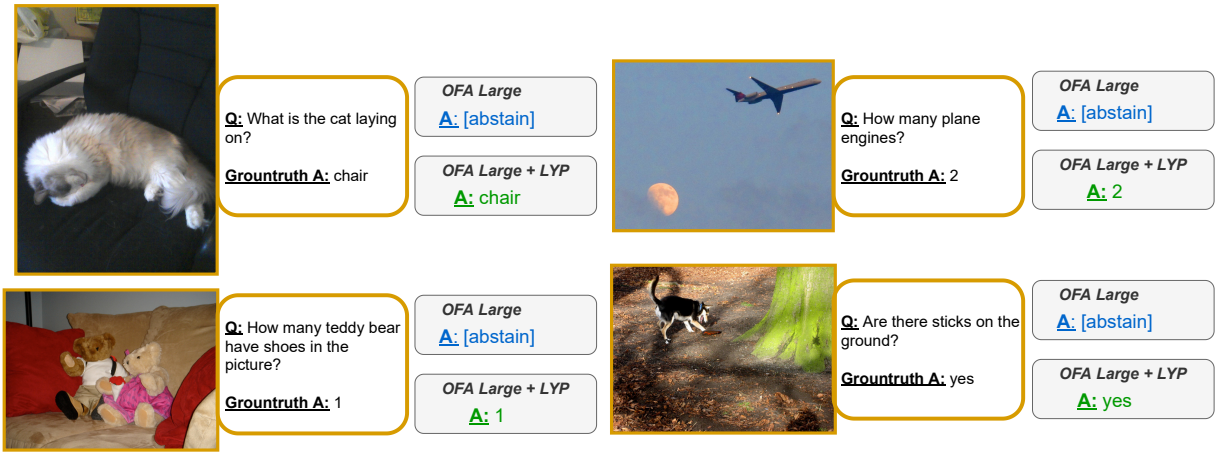

Figure 10. Qualitative examples for OFA-Large on AdVQA: On those examples, the baseline model abstains but had predicted the correct answer. OFA-Large + LYP does not abstain. The threshold is selected on in-distribution data for maximum coverage at 5% risk.

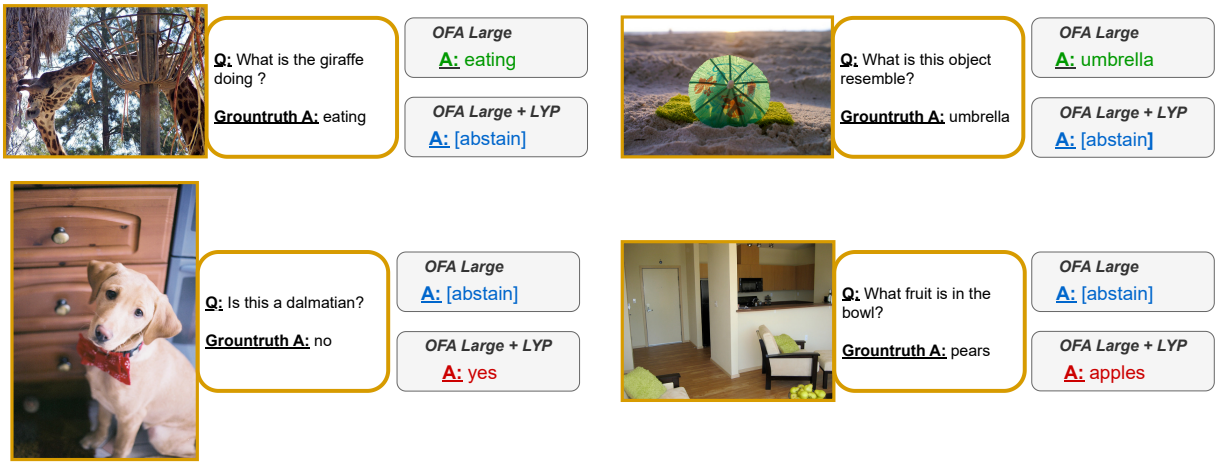

Figure 11. Failure cases for OFA-Large + LYP on AdVQA: On the first two examples, the baseline predicts the correct answer, and OFA-Large + LYP abstains. On the second line, the baseline abstains from answering an incorrect answer, while OFA-Large + LYP still answers. For both models, the threshold is selected on in-distribution data for maximum coverage at 5% risk.

References

[1] Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Calibrated selective classification. *arXiv preprint arXiv:2208.12084*, 2022. 4

[2] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pages 2151–2159. PMLR, 2019. 3

[3] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018. 1, 5

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 4

[6] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 5

[7] Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online, July 2020. Association for Computational Linguistics. 1

[8] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 5

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[10] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*, 2021. 5

[11] Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Magana, Tristan Thrush, Wojciech Galuba, Devi Parikh, and Douwe Kiela. Human-adversarial visual question answering. *Advances in Neural Information Processing Systems*, 34:20346–20359, 2021. 2

[12] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*, 2022. 4

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[14] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 4, 6

[15] Spencer Whitehead, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. Reliable visual question answering: Abstain rather than answer incorrectly. *arXiv preprint arXiv:2204.13631*, 2022. 1, 3, 4, 5, 7, 8, 9, 10

[16] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 1, 6