Appendix to Reliable Visual Question Answering: Abstain Rather Than Answer Incorrectly

Spencer Whitehead[1⋆], Suzanne Petryk[1,2⋆], Vedaad Shakib[2], Joseph Gonzalez[2], Trevor Darrell[2], Anna Rohrbach[2], and Marcus Rohrbach[1]

[1] Meta AI
[2] UC Berkeley

Appendix A has more discussion on Selector ablations.
Appendix B shows an experiment with data augmentation for MaxProb.
Appendix C provides a manual evaluation of the label noise.
Appendix D gives further analysis comparing Selector versus MaxProb decisions.
Appendix E provides more qualitative results.
Appendix F presents results on threshold generalization.
Appendix G looks at the calibration metric ECE.
Appendix H has additional details on the dataset splits.
Appendix I has additional model details.
Appendix J provides standard deviations for results in Tab. 1 and Tab. 2.
Appendix K provides a proof of Lemma 1, providing a motivation for the definition of the Effective Reliability score Φ_c.
Appendix L discusses the relevance of related conformal prediction works.

A Selector Design Ablations

Extending the discussion in Sec. 5.4, we are isolating the effects of different features/modalities on the risk-coverage trade-off when using Selector. In this direction, we experiment with different input representation variants from CLIP-ViL [24] in Tab. 3 by ablating the question q, multimodal r, and answer $f'(x)$ representations as well as different image representations. For image representations, we ablate the usage of the visual representation $\tilde{v}$ directly from the CLIP visual encoder [21], as well as the visual representation v that is the concatenation of the respective pooled outputs from MCAN's self-guided attention module [31] and MoVie's modulated convolutional bottleneck [18], which are visual representations that also contain multimodal information from the question. Question representations are taken from the output of MCAN's self-attention module. The multimodal representation is the concatentation of the multimodal representations that are used as inputs to the softmax output (i.e., classification) layer of CLIP-ViL. For the answer representation, we use the logits just before the softmax in the output layer.

⋆ Equal contribution

Architecture	$\mathcal{C}$@1% ↑	AUC ↓	Φ_{100} ↑
1-layer Linear	10.95	10.68	**7.47**
2-layer MLP (ours)	13.32	**9.73**	7.32
4-layer Transformer	**13.48**	9.78	7.35

Table 4: Different Selector architectures with CLIP-ViL on our selection function validation split (Val in Tab. 9). All in %.

The results in Tab. 3 show the importance of using multimodal information for coverage at low risk levels. When comparing using each representation in isolation, we see that multimodal representations (r, v, and $f'(x)$) yield much stronger $\mathcal{C}$@1%, $\mathcal{C}$@5%, Φ_{10} and Φ_{100} than unimodal representations ($\tilde{v}$ and q). We also observe that the answer representation achieves the best performance for $\mathcal{C}$@10% and $\mathcal{C}$@20% when each input representation is used in isolation. Overall, we find that considering multimodal information (i.e., combinations of multimodal representations and unimodal representations from different modalities) to be most effective, with the top performers being the models that incorporate the answer representation alongside multimodal representations ($f'(x)$+r, $f'(x)$+v, and $f'(x)$+q+v+r).

Lastly, we also experiment with other architectures for the Selector using the same features as above. Our Selector is a 2-layer multi-layered perceptron (MLP) (Appendix I). In Tab. 4, we see that a simpler, 1-layer Selector has slightly higher Φ_{100}, yet lowers $\mathcal{C}$@1% by about 2.4%. A more complex Transformer yields comparable performance to our 2-layer Selector. Given these results as well as those in Tab. 3, we observe that the input representations and training objectives appear to be most important, and efforts for improving learned selection function performance can potentially focus on these.

B Comparing to Data Augmentation

In our experiments, we use a separate set to validate VQA models and train the selection functions (Dev in Tab. 9). However, one could use this data to augment the VQA training data, which could potentially improve performance for MaxProb as there is a relationship between accuracy and these reliability metrics (Sec. 5.2). Tab. 5 presents these results where we see that using this data to train the Selector is more effective for improving coverage at low risk levels and Φ_c with a high cost. Since the extra data helps improve accuracy, as the risk tolerance nears the error rate of the model and coverage approaches 100%, MaxProb surpasses Selector in coverage (i.e., $\mathcal{C}$@20%) and Effective Reliability (i.e., Φ_1). However, overall, these results suggest that using this data to train a Selector can be more beneficial to model reliability than using it for augmentation.

Model f	Selection function g	Acc. ↑	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R}=1\%$	$\mathcal{R}=5\%$	$\mathcal{R}=10\%$	$\mathcal{R}=20\%$	AUC ↓	Φ_c ↑ c=1	c=10	c=100
CLIP-ViL	MaxProb	71.48	3.33	31.92	53.93	84.36	10.59	55.10	20.22	1.93
	MaxProb-Aug	72.31	6.57	33.62	56.18	**86.20**	10.14	**56.64**	22.13	2.97
	Selector	71.48	**13.32**	**38.02**	**58.16**	85.03	**9.73**	56.09	**24.85**	**7.32**

Table 5: Comparison between augmenting the training data of CLIP-ViL with our dev set for MaxProb versus utilizing our dev split for training Selector. Results are on our selection function validation split (Val in Tab. 9). All in %.

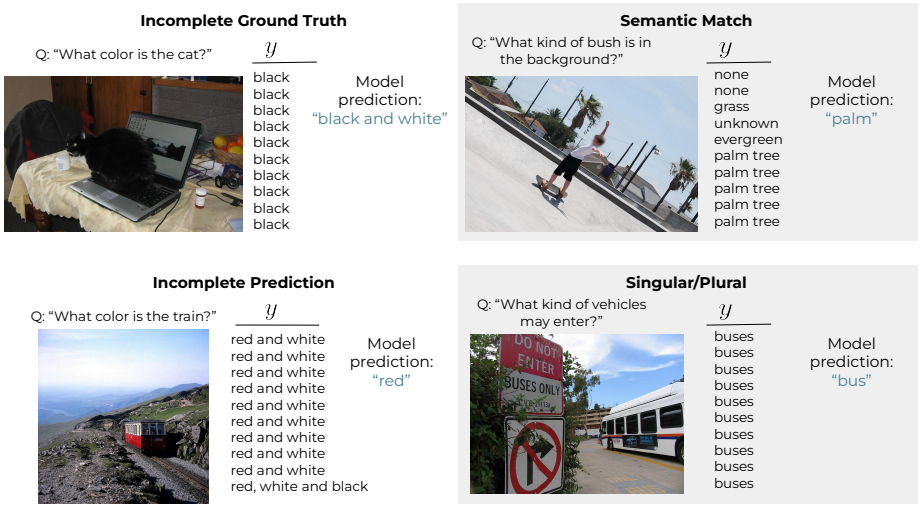

Fig. 4: Example questions, images, annotations, and model predictions for each category of label noise we discover.

C Manual Evaluation of Label Noise

As discussed in Sec. 5.3, we provide further details on our manual annotation for label noise as well as Φ_{100} when accounting for cases where the model may have been unfairly penalized. We specifically annotate image-question-answer triples, and discovered the following cases (Fig. 4 provides examples of each):

Incomplete Ground Truth: The ground truth is in some way incomplete and simply misses the predicted answer.

Semantic Match: The predicted answer is semantically correct but does not exactly match the ground truth.

Incomplete Prediction: The predicted answer is incomplete but has part of the correct answer.

Singular/Plural: The predicted answer is singular/plural while the ground truth is plural/singular (though only if providing the opposite singular/plural version is still correct).

We do these annotations for each considered VQA model and selection function trained to optimize Φ_{100} (i.e., the strongest penalty for wrong answers) and focus our efforts on questions with VQA accuracy of 0, meaning questions that contribute negatively to Φ_{100}. Once we have the annotations of unfairly penalized questions, we recompute the Effective Reliability score Φ'_{100} when counting those questions as either abstentions or as answered questions that achieved a VQA accuracy of 100%. Although the selection function decided to answer each of the unfairly penalized questions that we annotated, we compute Φ'_{100} under these two cases because it is unclear exactly how correct these non-matching answers should be considered. Counting them as abstentions serves as a lower bound for Φ'_{100}, whereas assigning a VQA accuracy of 100% is an upper bound.

We present the results before (Φ_{100}) and after (Φ'_{100}) controlling for noise in Tab. 6. We find that while this noise does contribute to some differences in performance, it does not affect the rankings between selection functions. For example, relative to each Φ_{100} with CLIP-ViL, Φ'_{100} yields an increase of 0.27% for MaxProb, 0.38% for Calibration, and 0.56% for Selector, yet the rankings remain the same. Qualitatively, we observe that there tends to be a very significant overlap in unfairly penalized examples between selection functions, which is likely part of why the rankings remain the same. Moreover, the amount of these label errors tends to be small, and the vast majority of questions contributing to the penalties in Φ_{100} across all models are properly marked as incorrect ($\sim$93%). Since the score for an incorrect sample (-100) is considerably lower than a sample marked as 100% correct (+1), there is also little difference in Φ'_{100} when considering these few unfairly penalized questions as abstentions versus as correct answers. These results imply that the comparisons between different selection functions at high cost (or low risk) for a given model are still meaningful despite the potential presence of noise.

D Analysis of Selector Decisions

We would like to understand any differences in the types of questions that the Selector chooses to abstain or answer as compared to MaxProb. We compare decisions on our test split for the two selective models, where thresholds were chosen to optimize Φ_{100} on validation. We use labels from [29] which assign one of the following categories to each question, in order of difficulty: unimodal (Level 1), where the question could be answered without looking at the image, "simple-multimodal" (Level 2), where the question is simple to answer when additionally considering the image, and "difficult-multimodal" (Level 3), where the question is difficult to answer even when considering both modalities. Fig. 5 compares the number of questions answered in each difficulty level by the MaxProb and Selector models. We find that the Selector not only answers 1.2× more unimodal questions than MaxProb, but also 1.9× more "simple-multimodal" and, impressively, 9.6× more "difficult-multimodal" questions.

Model f	Selection function g	% Correct GT	Φ_{100} ↑	Φ'_{100} ↑ *Abstain*	Φ'_{100} ↑ *Correct*
Pythia [11]	MaxProb	91.30	1.81	2.00	2.00
	Calibration	93.55	2.14	2.32	2.33
	Selector	87.50	**4.12**	**4.49**	**4.50**
ViLBERT [16]	MaxProb	97.75	1.67	1.86	1.86
	Calibration	94.94	2.92	3.30	3.30
	Selector	88.14	**5.41**	**6.07**	**6.08**
VisualBERT [14]	MaxProb	100.00	2.49	2.49	2.49
	Calibration	97.92	3.83	3.93	3.93
	Selector	85.29	**4.82**	**5.77**	**5.78**
CLIP-ViL [24]	MaxProb	94.74	1.82	2.09	2.09
	Calibration	93.44	5.78	6.16	6.16
	Selector	87.23	**8.76**	**9.32**	**9.32**

Table 6: Effect of label noise on Φ_{100}. % Correct GT indicates the percentage of answered samples with a VQA accuracy of 0, where the ground truth and resulting VQA accuracy was considered correct based on the question, image, annotations, and model prediction. Φ_{100} indicates the original score, whereas Φ'_{100} indicates the score when counting answered questions where label errors led to a VQA accuracy of 0 as abstentions (*Abstain*) or having a VQA accuracy of 100% (*Correct*) instead of being counted as incorrect. Although there is a small amount of label noise, it does not affect the ranking between selection functions with respect to Effective Reliability. All in %.

E More Qualitative Analysis

In Fig. 6, we show several more examples of cases from our test split that illustrate Selector and MaxProb decisions, where we use CLIP-ViL with selection functions optimized for Φ_{100} on the validation set (same as Fig. 3). In particular, we show cases where the decisions of Selector and MaxProb differed — where Selector chooses to answer while MaxProb abstains, and vice-versa. We see some cases where the MaxProb decision to abstain may have been influenced by variability in possible answers that may cause model confidence values to be split, yet the annotations themselves have underlying semantic agreement (e.g., Fig. 6 top left, where "*sunny*" weather conditions are also described as "*nice*" or "*clear*"). On the other hand, we also see cases where the model was incorrect on questions which may have been unclear or surprising, and Selector chose to abstain whereas MaxProb chose to answer (e.g., the second example on row (c) asks the unusual question "*Is the bear wearing a helmet?*"). In these cases, we would expect a selective VQA model to abstain from answering to avoid providing an incorrect answer. Additionally, we show several failure cases of Selector, which chose to answer on an incorrect question while MaxProb chose to abstain.

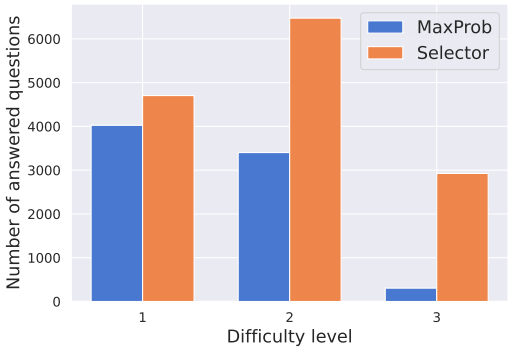

Fig. 5: Number of questions in our test split that the MaxProb and Selector selection functions chose to answer, grouped by difficulty level [29]. Level 1 corresponds to simple questions that could be answered without the image, Level 2 questions are simple to answer when considering both the question and image, and Level 3 questions are difficult to answer even when considering both modalities. Thresholds for the selection functions are chosen on the validation set to maximize Φ_{100}.

Selection function g	$\mathcal{R} =$	$\Delta\mathcal{R}$ 1%	5%	10%	20%	$\Delta\mathcal{C}$ 1%	5%	10%	20%
MaxProb		+0.12	−0.14	+0.17	−0.09	+0.92	−0.55	+0.81	−0.20
Selector		+0.14	+0.25	+0.17	−0.23	+2.00	+1.09	+0.59	−0.49

Table 7: Generalization of abstention thresholds γ from validation to test, with VisualBERT. $\Delta\mathcal{R}$ and $\Delta\mathcal{C}$ are the differences in risk and coverage percentages, respectively, when using γ selected for the target risk $\mathcal{R}$ on validation vs. γ with maximum $\mathcal{C}@\mathcal{R}$.

F Threshold Generalization

As discussed in Sec. 5.2, we evaluate how well a threshold selected for a target risk level on validation can achieve a similar level of risk on our test split. Experimenting with VisualBERT, comparing MaxProb and Selector, we see in Tab. 7 that the differences in risk for both selection functions tend to be at most 0.25%. Likewise, we observe corresponding differences in achieved coverage between the validation threshold and the maximum coverage ($\Delta\mathcal{C}$). This demonstrates that the thresholds can generalize reasonably well, although it does not allow for a direct comparison of coverage for the same risk. Effective Reliability, on the other hand, can use thresholds chosen from validation and still result in a clear comparison of models as it is a single metric.

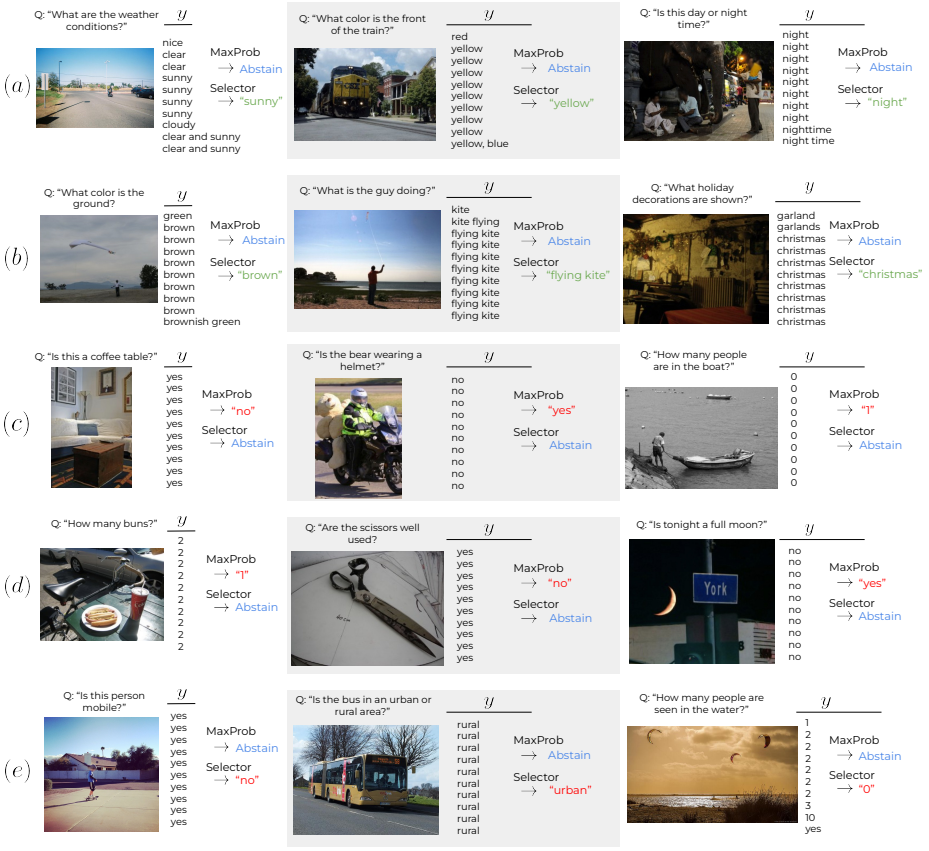

Fig. 6: More qualitative test set examples with CLIP-ViL selective model predictions, when optimized for Φ_{100} on validation. Rows (a) and (b) show cases where the model was correct, yet MaxProb chose to abstain and Selector chose to answer. Rows (c) and (d) show examples of the opposite case, where the model was wrong, yet MaxProb chose to answer (contributing to the risk) and Selector chose to abstain. Row (e) shows failure cases of Selector, which chose to answer on an incorrect sample when MaxProb chose to abstain.

G Effect of Model Calibration

We report the calibration performance of the vector scaling. Specifically, we measure the expected calibration error (ECE) [8, 17], which measures the expected difference between the model confidence and accuracy. The lower the ECE, the more that the model's confidence scores correspond to the actual accuracy of the predictions. Note that the ECE metric is designed for single label classification problems. To use the ECE metric for VQA, where there can be multiple possible answers for a question, we simply consider the most frequent human annotated answer as the ground truth for each question.

	Pythia		ViLBERT		VisualBERT		CLIP-ViL	
	MaxProb	Calib.	MaxProb	Calib.	MaxProb	Calib.	MaxProb	Calib.
ECE ↓	0.1702	**0.0938**	0.1457	**0.1120**	0.1458	**0.1169**	0.1978	**0.1521**

Table 8: ECE of different models with (Calibration, denoted Calib.) and without (MaxProb) the vector scaling calibration on our test split. Lower is better.

Source	Split Name	Usage	% src	#I	#Q	#A
VQA v2 train	Train	Train f	100%	82,783	443,757	4,437,570
VQA v2 val	Dev	Validate f / Train g	40%	16,202	86,138	861,380
	Val	Validate g	10%	4,050	21,878	218,780
	Test	Test h	50%	20,252	106,338	1,063,380

Table 9: Table of statistics for the dataset splits used for training as well as validating VQA models (f), training as well as validating selection functions (g), and testing full selective models ($h = (f, g)$). % src indicates the percentage of the source data (Source) that each split represents. #I, #Q, and #A indicate the number of images, questions, and answers, respectively.

We see in Tab. 8 that vector scaling does indeed improve calibration for all models. Taking this observation in combination with the improvements over MaxProb on $\mathcal{C}@\mathcal{R}$, AUC, and Effective Reliability seen in Tab. 1 and Tab. 2, it appears that improving model calibration can help improve the risk-coverage trade-off. However, as discussed in Sec. 4, it is necessary to use calibration techniques that can change the relative confidence rankings, such as vector scaling.

H Additional Dataset Split Details

We experiment on the VQA v2 dataset [7], which contains a large amount of human-annotated image-question-answer triplets. Tab. 9 lays out the data splits we use in our experiments. We create splits of the VQA v2 validation set since we require answer annotations to evaluate risk, coverage, and Effective Reliability. These splits are created such that no images (and therefore no question-answer annotations) are shared between them. Note that the data in the held out test set (Test in Tab. 9) is never seen during the training or validation of any component (f or g) and is only used for evaluations. All presented results are on our test set unless otherwise specified.

I Model Details

In this section, we present the details of the models used in our experiments.

Hyperparameters	Pythia	ViLBERT†	VisualBERT†	CLIP-ViL
Batch Size	512	896	896	32
Hidden Size	5,000	1,024	768	1,024
# Layers	L-1, V-1	L-12, V-6	12	6 / 4
Optimizer	Adamax[12]	AdamW[15]	AdamW[15]	AdamW[15]
Adam ϵ	1e-8	1e-8	1e-8	1e-9
Adam β_1	0.9	0.9	0.9	0.9
Adam β_2	0.999	0.98	0.98	0.98
Learning rate	0.01	5e-5	5e-5	5e-5
Dropout	–	0.1	0.1	0.1
# Steps	22,000	88,000	88,000	236,000
# Warmup Steps	1,000	2,000	2,000	54,000
Max Grad. L2-Norm	0.25	–	–	5

Table 10: Hyperparameters of each model used in our experiments. Max Grad. L2-Norm is used for gradient clipping. L and V indicate language and vision layers, respectively. The 6 / 4 for CLIP-ViL indicates that the model has 6 MCAN layers and 4 MoVie layers. † indicates that the hyperparameters are reported directly from [26].

I.1 VQA Models

We use the open-source MMF framework [25] for all our experiments, which contains implementations of each VQA model.[1] For training VQA models, we follow the hyperparameters from MMF, which we list in Tab. 10. All models treat VQA as a classification task and are trained with VQA accuracy as soft target scores via a binary cross-entropy loss [28]. We briefly discuss the models and settings used in our experiments, extending Sec. 5.1:

Pythia [11]: A previous state-of-the-art model that won the 2018 VQA challenge and is an optimization of the widely used bottom-up top-down (BUTD) VQA model [1]. This model uses BUTD object detection features [1] trained on Visual Genome [13], but the features are extracted from a ResNext-152 based FasterRCNN [22]. Pythia's implementation further uses grid features from a ResNet-152 [9] as additional inputs to improve performance [11]. GloVe embeddings [19] are used to initialize the word representations. We train this model from scratch on the VQA v2 training data.

ViLBERT [16]: A two-stream vision-and-language transformer model [4, 27] that also uses object detection features. The same object detection features from Pythia are used, but without the addition of grid features. We use the pretrained and fine-tuned model provided by MMF.[2] The MMF version of this model is from [26] is pretrained on the VQA v2 training data [7] using self-supervised objectives (masked language modeling and masked image modeling). The VQA

[1] `https://mmf.sh/`

[2] `https://github.com/facebookresearch/mmf/tree/main/projects/pretrain_vl_right`

model is initialized with the pretrained encoder weights, and then fine-tuned on the VQA v2 training data.

VisualBERT [14]: This model is a single-stream transformer architecture, like BERT [6]. Here, the setup is very similar to ViLBERT and we use the same visual features as ViLBERT. We again use the pretrained and fine-tuned model provided by MMF.[2] This MMF version of VisualBERT [26] is pretrained on MSCOCO captions [5] using a masked language modeling objective. Just like ViLBERT, the VQA model is also initialized with the pretrained encoder weights and fine-tuned on VQA v2.

CLIP-ViL [24]: This represents a state-of-the-art model that is trained from scratch on the VQA data whose visual encoder is from the CLIP model [21]. The visual representations are grid features that are obtained from the visual encoder of the CLIP model [21]. We use the implementation provided by the authors of [24] to extract the visual features.[3] The VQA architecture, MoVie+MCAN [18], is an ensemble of a transformer encoder-decoder [31] and modulated convolutional [18] model, which won the 2020 VQA challenge. GloVe embeddings [19] are also used to initialize the word representations. Like Pythia, we train this VQA model from scratch on VQA v2 training data.

I.2 Selection Functions

We detail the Calibration and Selector selection functions here. We do not cover MaxProb as no additional training is required. While training each selection function, we freeze the weights of the VQA model.

Calibration. The inputs to the calibration are the unnormalized answer logits (i.e., answer representation just before the softmax) of the VQA model, and the outputs are the calibrated logits. Since we use vector scaling [8, 20], we input the logits from the VQA model into a linear layer with a diagonal weight matrix and a bias term. During training, after the linear layer, we apply a sigmoid activation and, in contrast to [8], use these as input to a binary cross entropy loss with the soft VQA labels [28]. We train the linear layer using the AdamW optimizer [15] with a learning rate of 0.01 and a weight decay of 1e-4. At test time, we use the output of this linear layer as our calibrated logits, apply a softmax, and use the same abstention procedure as MaxProb (Sec. 4).

Selector. The inputs to Selector are the answer, question, image, and multimodal representations. For each input, we have a specific 1-layer MLP with a ReLU activation and hidden size of 512. We then concatenate the outputs of these layers and input them to a 2-layer MLP with ReLU activations and hidden size of 1,024, followed by a binary output layer to produce a confidence value. This architecture remains exactly the same for all models. However, if a model produces a set of representations for the image or question, then we max pool these features to collapse them to a single representation. For optimization, we employ the AdamW optimizer [15] with a learning rate of 1e-4, a batch size of 256, and gradient clipping with a max gradient L2 norm of 0.25.

[3] `https://github.com/clip-vil/CLIP-ViL/tree/master/CLIP-ViL-Direct/vqa`

Model f	Selection function g	Acc. ↑	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R}=1\%$	$\mathcal{R}=5\%$	$\mathcal{R}=10\%$	$\mathcal{R}=20\%$	AUC ↓
Pythia	MaxProb	66.17 ± 0.10	6.00 ± 0.37	24.71 ± 0.46	40.99 ± 0.39	71.45 ± 0.25	13.88 ± 0.08
	Calibration	66.45 ± 0.09	6.50 ± 0.43	25.07 ± 0.46	41.95 ± 0.38	73.44 ± 0.27	13.52 ± 0.08
	Selector	66.17 ± 0.10	8.79 ± 0.52	26.92 ± 0.31	43.24 ± 0.43	73.40 ± 0.25	13.30 ± 0.07
	Best Possible ($\mathcal{C}$)	66.17 ± 0.10	62.67 ± 0.11	68.41 ± 0.12	73.52 ± 0.11	82.71 ± 0.13	6.68 ± 0.04
CLIP-ViL	MaxProb	71.75 ± 0.13	6.78 ± 1.98	34.69 ± 1.30	55.72 ± 0.43	85.13 ± 0.23	10.23 ± 0.13
	Calibration	71.71 ± 0.11	13.12 ± 0.72	37.06 ± 0.35	56.06 ± 0.35	85.23 ± 0.20	9.91 ± 0.07
	Selector	71.75 ± 0.13	16.34 ± 0.73	39.48 ± 0.29	58.16 ± 0.40	85.37 ± 0.25	9.52 ± 0.07
	Best Possible ($\mathcal{C}$)	71.75 ± 0.13	68.49 ± 0.15	74.55 ± 0.15	79.72 ± 0.14	89.69 ± 0.16	4.58 ± 0.05

Table 11: Mean and standard deviations for risk-coverage metrics for different selection functions. All in %.

Model f	Selection function g	c=1 Φ_1 ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑	c=10 Φ_{10} ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑	c=100 Φ_{100} ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑
Pythia	—	38.49 ± 0.19	33.83 ± 0.10	100 ± 0.00	-210.62 ± 1.01	33.83 ± 0.10	100 ± 0.00	−2701.68 ± 9.19	33.83 ± 0.10	100 ± 0.00
	MaxProb	47.28 ± 0.12	21.62 ± 0.27	76.03 ± 0.57	15.15 ± 0.35	5.24 ± 0.40	25.62 ± 1.42	2.27 ± 0.18	0.85 ± 0.15	4.89 ± 1.04
	Calibration	48.06 ± 0.15	21.21 ± 0.34	76.18 ± 0.73	15.23 ± 0.36	5.85 ± 0.68	28.06 ± 2.31	2.19 ± 0.66	0.94 ± 0.28	5.88 ± 1.62
	Selector	48.16 ± 0.16	20.67 ± 0.65	74.84 ± 1.26	17.12 ± 0.24	5.99 ± 0.23	30.16 ± 0.75	3.84 ± 0.39	0.94 ± 0.18	8.23 ± 1.33
	Best Possible (Φ_c)	66.17 ± 0.10	8.51 ± 0.05	72.32 ± 0.09	66.17 ± 0.10	8.51 ± 0.05	72.32 ± 0.09	66.17 ± 0.10	8.51 ± 0.05	72.32 ± 0.09
CLIP-ViL	—	49.41 ± 0.25	28.25 ± 0.13	100 ± 0.00	-151.70 ± 1.32	28.25 ± 0.13	100 ± 0.00	-2162.80 ± 12.06	28.25 ± 0.13	100 ± 0.00
	MaxProb	55.82 ± 0.14	19.22 ± 0.30	83.45 ± 0.65	22.03 ± 0.59	5.59 ± 0.34	37.67 ± 1.31	2.85 ± 0.75	0.96 ± 0.23	6.97 ± 2.39
	Calibration	56.03 ± 0.17	18.30 ± 0.44	81.61 ± 0.98	23.24 ± 0.34	4.95 ± 0.46	36.82 ± 1.91	5.30 ± 0.71	0.73 ± 0.20	9.97 ± 2.35
	Selector	56.45 ± 0.16	17.44 ± 0.53	80.09 ± 1.13	26.06 ± 0.30	5.03 ± 0.48	39.59 ± 2.04	8.01 ± 0.68	0.55 ± 0.15	11.38 ± 2.10
	Best Possible (Φ_c)	71.75 ± 0.13	7.60 ± 0.07	77.66 ± 0.12	71.75 ± 0.13	7.60 ± 0.07	77.66 ± 0.12	71.75 ± 0.13	7.60 ± 0.07	77.66 ± 0.12

Table 12: Mean and standard deviation for Effective Reliability Φ_c over 10 trials. All in %.

J Extended Results

Tab. 11 and Tab. 12 provide the mean and standard deviation over the 10 random seeds for Pythia and CLIP-ViL results. Due to difficulties reproducing the pretrained and fine-tuned performance of ViLBERT and VisualBERT, we simply use existing checkpoints in MMF[2] and report single run metrics for these VQA models.

K Proof of Lemma 1

Lemma 1 states that if a model abstains "perfectly", the introduced Effective Reliability score is equal to the VQA Accuracy. In this section, we provide a proof of Lemma 1 in the main paper, which we repeat here for ease of understanding the proof:

Lemma 1. *The Effective Reliability score is equal to the VQA Accuracy ($\Phi_c(x) = Acc(x)$) if a model abstains ($g(x) = 0$)* iff *it is incorrect ($Acc(x) = 0$).*

Distilling this to the mathematical notation:

$$(g(x) = 0 \leftrightarrow Acc(x) = 0) \longrightarrow \Phi_c(x) = Acc(x) \tag{1}$$

Extending Eq. 6 to both cases, $Acc(x) = 0$ and $Acc(x) > 0$ (note, that Acc cannot be smaller than 0):

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ -c & \text{if } g(x) = 1 \text{ and } Acc(x) = 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) > 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{2}$$

To prove Lemma 1, we must show that the condition $(g(x) = 0 \leftrightarrow Acc(x) = 0)$ implies $\Phi_c(x) = Acc(x)$. The condition $(g(x) = 0 \leftrightarrow Acc(x) = 0)$ simplifies Eq. 2 as the second and third line contradict the condition:

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{3}$$

As the $Acc(x) = 0$, the second line can be re-written as:

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ Acc(x) & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{4}$$

Now, in both cases $\Phi_c(x) = Acc(x)$ □

L Relation to Conformal Prediction

Conformal prediction aims to predict a set of outputs, with a guarantee that the set contains the correct output with a specified probability [30, 23]. In VQA, the criterion of a set containing the "correct output" is harder to define. For example, two distinct answers might be both be true (*"yellow", "brown"*) for *"What color are the bananas?"*, but others sets might be contradictory (*"yes", "no"*). Further research might focus on how to best convey answer sets to users in VQA and how semantic similarity of answers should be modeled, or on the design of better criteria to determine a set-based risk. More generally, the field of risk control, which does not require variable-size output sets, provides theoretical guarantees that a given error measure is below a tolerance level with some specified probability [2, 10]. [3] describes how to choose a prediction threshold to satisfy a guarantee on error bound. [10] relates these guarantees to test sample accuracy based on training sample density. We view these probabilistic guarantees on error bounds as complementary to our framework, with opportunities for future work to incorporate them both.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6077–6086 (2018)

2. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511 (2021)
3. Angelopoulos, A.N., Bates, S., Candès, E.J., Jordan, M.I., Lei, L.: Learn then test: Calibrating predictive algorithms to achieve risk control. arXiv preprint arXiv:2110.01052 (2021)
4. Cao, J., Gan, Z., Cheng, Y., Yu, L., Chen, Y.C., Liu, J.: Behind the scene: Revealing the secrets of pre-trained vision-and-language models. In: European Conference on Computer Vision. pp. 565–580. Springer (2020)
5. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft COCO captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)
8. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. pp. 1321–1330. PMLR (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Ji, X., Pascanu, R., Hjelm, D., Lakshminarayanan, B., Vedaldi, A.: Test sample accuracy scales with training sample density in neural networks. arXiv preprint arXiv:2106.08365 (2021)
11. Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., Batra, D., Parikh, D.: Pythia v0. 1: the winning entry to the vqa challenge 2018. arXiv preprint arXiv:1807.09956 (2018)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (2015)
13. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**(1), 32–73 (2017)
14. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. In: Arxiv (2019)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
16. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Advances in neural information processing systems **32** (2019)
17. Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
18. Nguyen, D.K., Goswami, V., Chen, X.: Movie: Revisiting modulated convolutions for visual counting and beyond. In: Proceedings of the International Conference on Learning Representations (2021)

19. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
20. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers **10**(3), 61–74 (1999)
21. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
23. Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research **9**(3) (2008)
24. Shen, S., Li, L.H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.W., Yao, Z., Keutzer, K.: How much can clip benefit vision-and-language tasks? arXiv preprint arXiv:2107.06383 (2021)
25. Singh, A., Goswami, V., Natarajan, V., Jiang, Y., Chen, X., Shah, M., Rohrbach, M., Batra, D., Parikh, D.: Mmf: A multimodal framework for vision and language research. `https://github.com/facebookresearch/mmf` (2020)
26. Singh, A., Goswami, V., Parikh, D.: Are we pretraining it right? digging deeper into visio-linguistic pretraining. arXiv preprint arXiv:2004.08744 (2020)
27. Tan, H., Bansal, M.: LXMERT: Learning cross-modality encoder representations from transformers. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. pp. 5100–5111 (2019). https://doi.org/10.18653/v1/D19-1514, `https://www.aclweb.org/anthology/D19-1514`
28. Teney, D., Anderson, P., He, X., Van Den Hengel, A.: Tips and tricks for visual question answering: Learnings from the 2017 challenge. In: CVPR (2018)
29. Terao, K., Tamaki, T., Raytchev, B., Kaneda, K., Satoh, S.: Which visual questions are difficult to answer? analysis with entropy of answer distributions. arXiv preprint arXiv:2004.05595 (2020)
30. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Science & Business Media (2005)
31. Yu, Z., Yu, J., Cui, Y., Tao, D., Tian, Q.: Deep modular co-attention networks for visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6281–6290 (2019)