Delving Deeper into Cross-lingual Visual Question Answering

Anonymous ACL submission

Abstract

Visual question answering (VQA) is one of the crucial vision-and-language tasks. Yet, existing VQA research has mostly focused on the English language, due to a lack of suitable evaluation resources. Previous work on cross-lingual VQA has reported poor zero-shot transfer performance of current multilingual multimodal Transformers with large gaps to monolingual performance, without any deeper analysis. In this work, we delve deeper into the different aspects of cross-lingual VQA, aiming to understand the impact of input data, fine-tuning and evaluation regimes, and interactions between the modalities in cross-lingual setups. The key results of our analysis are: **1)** We show that simple modifications to the standard training setup can substantially reduce the transfer gap to monolingual English performance, yielding +10 accuracy points over existing methods. **2)** We analyze cross-lingual VQA across different question types of varying complexity for different multilingual multimodal Transformers, and identify question types that are the most difficult to improve on. **3)** We provide an analysis of modality biases present in training data and models, revealing why zero-shot performance gaps remain for certain question types and languages. We will release our code at [URL-ANONYMOUS].

1 Introduction

The lack of multilingual resources has hindered the development and evaluation of Visual Question Answering (VQA) methods beyond the English language until recently. A rise in interest in creating multilingual Vision-and-Language (V&L) resources has inspired more research in this area (Srinivasan et al., 2021; Su et al., 2021; Liu et al., 2021a; Pfeiffer et al., 2022; Wang et al., 2021; Bugliarello et al., 2022, *inter alia*). Large Transformer-based models pretrained on images and text in *multiple* different languages have been proven as a viable vehicle for the development of multilingual V&L task architectures through transfer learning, but such models are still few and far between (M3P, UC2; Ni et al., 2021; Zhou et al., 2021). Large decreases in task performance between monolingual and (zero-shot) cross-lingual transfer setups have been measured and reported, among other multilingual V&L tasks, in VQA (Pfeiffer et al., 2022). Yet, the reasons for such low results in this pivotal V&L task have not been investigated in depth.

In this work, we aim to shed new light on the cross-lingual performance gap of cross-lingual VQA models from multiple angles. To the best of our knowledge, we are the first to provide a comprehensive analysis of multilingual VQA, with focus on cross-lingual transfer scenarios. We assess and discuss the impact of diverse prediction head architectures, extending input signals, as well as more sophisticated fine-tuning strategies on the final cross-lingual VQA performance, aiming to mitigate the present performance gap. We further conduct extensive analyses into cross-lingual VQA model configurations to better understand their current gaps and modes of failure, across different multilingual multimodal Transformers, and in zero-shot and few-shot scenarios. Finally, we investigate whether they suffer from the so-called unimodal biases: that is, we probe if the models truly reason over both images and questions to solve the VQA task, or if they take unimodal 'shortcuts' instead, exploiting spurious correlations and artifacts of data creation.

We find that standard approaches from text-only cross-lingual transfer scenarios (Pires et al., 2019; Hu et al., 2020) do not leverage the full multilingual capabilities of the pretrained models; we measure considerably worse performance of 'standard' fine-tuning compared to a simple modified fine-tuning regime. Interestingly, we report a discrepancy between monolingual and cross-lingual performance with the modified fine-tuning regime: while they

do not have any substantial impact on the model performance in the *source* language (English), they considerably improve *cross-lingual* VQA capabilities, achieving gains of more than 10 absolute accuracy points over the baselines.

2 Preliminaries

The VQA task is typically framed as a classification problem with a large number of classes. For instance, in the VQA task on the standard English GQA dataset (Hudson and Manning, 2019), given a pair of an image and a question, a model needs to predict a correct answer from 1,853 possible classes. GQA consists of diverse structural and semantic patterns, in which the questions are visually grounded in the image. In multilingual and cross-lingual VQA, the goal is to make similar predictions, but the questions can be posed in different *target* languages (Pfeiffer et al., 2022): e.g., the VQA task on the multilingual xGQA dataset (Pfeiffer et al., 2022) relies on the same set of 1,853 classes as English GQA.

We base all our analyses and experiments on the xGQA dataset, which is, due to its size and language coverage, arguably the most comprehensive evaluation resource for cross-lingual VQA to date. It has also been included in the multimodal multilingual evaluation benchmark IGLUE (Bugliarello et al., 2022). xGQA is the multilingual extension of the English GQA dataset (Hudson and Manning, 2019) to 7 typologically diverse languages.[1]

In this work, we utilize and empirically compare two state-of-the-art Transformer-based pretrained multimodal multilingual architectures: **M3P** (Ni et al., 2021) and **UC2** (Zhou et al., 2021).[2] The standard cross-lingual *zero-shot* transfer setup for VQA involves fine-tuning all the weights of the large pretrained model on the downstream task data in the source language only. In the *few-shot* setup, after the source-language fine-tuning, the model is additionally optimized on a handful of task-annotated examples in the target language (Pfeiffer et al., 2022).

3 Modeling Methods

Motivation. Recent work on VQA in cross-lingual settings (Pfeiffer et al., 2022; Bugliarello et al., 2022) benchmarked standard multimodal architectures in zero-shot and few-shot transfer scenarios on the xGQA dataset, without aiming to understand the particulars of the cross-lingual VQA task more profoundly. At the same time, they report large gaps of cross-lingual transfer performance when compared to monolingual English performance, suggesting that there is ample room for improvement. In this work, we aim to leverage novel insights into different aspects of the cross-lingual VQA task (e.g., analyses over different question types or classification architectures) to guide improved cross-lingual VQA methods, described in what follows. In particular, we assess the impact of three orthogonal directions: **1)** classification architectures (§3.1); **2)** (richer) input signals (§3.2); **3)** fine-tuning strategies (§3.3).

3.1 Classification Architecture Variants

The original work on xGQA (Pfeiffer et al., 2022) evaluated only a simple 'shallow' linear classification head, termed **Linear** here: the output [CLS] token of the pretrained Transformer-based model (which has cross-attended over all text and image features) is simply passed into such linear classification. However, we hypothesize that this choice might have a substantial impact on transfer performance. Therefore, in the so-called **Deep** variant, instead of a linear classification head, we add a 2-layer transformation network (f_{trans}) with the GELU activation function (Hendrycks and Gimpel, 2016), dropout and a layer-normalization layer, before feeding the representations into a linear layer for classification. The first layer of f_{trans} uses an orthogonal initializer (Saxe et al., 2014). Unless noted otherwise, all of our following experiments are based on this 'deeper' architecture.[3]

3.2 Input Signal

A large number of output classes (see §2) potentially amplifies the difficulty of zero-shot and few-shot cross-lingual transfer due to the need of aligning contextual representations in multiple languages for multi-class classifications. Standard VQA datasets such as GQA and xGQA contain questions of five different structural types (*Verify, Logical, Query, Choose, Compare*).[4] Pfeiffer et al. (2022) have demonstrated a considerable performance variation over different question types, e.g.,

[1] For further details regarding xGQA we refer the reader to the original work.

[2] For technical details of the two models, we refer the reader to their respective papers.

[3] We illustrate the architecture in Figure 3 in Appendix D.

[4] See Appendix B for example questions per each of the five question types.

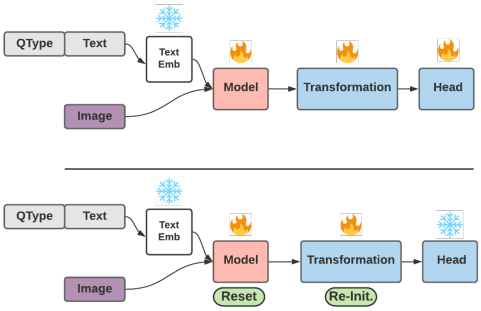


Figure 1: Self-Bootstrapping (§3.3). *Top:* Fine-tuning with frozen text embeddings (Stage 1). *Bottom:* Fine-tuning with text embeddings and classification head frozen. Other parameters are reset to their pretrained values or randomly initialized (Stage 2).

there is a large cross-lingual performance drop especially for *Choose*-type questions.

To help alleviate this issue, we propose to feed the model with designated question-type tokens (Prager, 2006; Murdock et al., 2012) which appear in GQA and xGQA. The idea is to influence the label distribution for the VQA classification task by additionally conditioning on a question-type token.[5] More concretely, we prepend a question-type token `QType` in English to the text input. We use structural question types as the question-type tokens; the text input then takes the following format: '`[QType] : [Question]`'.[6]

As the xGQA data contains questions with binary answers (i.e. *Yes/No* questions). We anticipate that for a large fixed number of output classes, these questions should benefit the most from using the question-type tokens. The models which rely on this question-type conditioning are denoted with the superscript ^{Q}, e.g., M3P^{Q}, see also later §3.4.

3.3 Fine-Tuning Strategy

Misalignment of multilingual text embeddings (Søgaard et al., 2018; Dubossarsky et al., 2020) has been indicated by Pfeiffer et al. (2022) as one of the principal causes for reduced zero-shot performance in the cross-lingual VQA tasks. Therefore, we propose two fine-tuning strategies, tailored exactly towards mitigating such undesired shifts in the multilingual embedding space.

Freezing Text Embeddings. In the first variant, we freeze text embeddings during fine-tuning and only optimize the Transformer weights and the classification head. This should prevent misalignment of the text embedding space during fine-tuning. This strategy, labelled **+FT**, is referred to as contrastive tuning by Zhai et al. (2021).

Self-Bootstrapping. Zero-shot cross-lingual transfer via standard fine-tuning is known to be sensitive to parameter initialization (Bugliarello et al., 2022), and a good initialization of the classification head improves generalization without degrading pretrained features (Kumar et al., 2022). Furthermore, warm-start training (Ash and Adams, 2020) periodically shrinks and perturbs weights to improve generalization. Motivated by these insights from prior research, we first train the network to learn the classification head, then reset and fine-tune the remaining model parameters. This leads to a two-stage fine-tuning process, termed *self-bootstrapping* (labeled **+SB**), illustrated in Figure 1 and outlined here:

Stage One: We fine-tune all parameters (with text embeddings frozen) on the task data.

Stage Two: We **1)** freeze the classification head (excluding the bias parameters) and text embeddings, **2)** reset the remaining parameters in the multimodal multilingual model to pretrained weights, and **3)** re-initialize the f_{trans} network (see §3.1). We then fine-tune the transformer weights on the task data.[7]

To make fair comparisons between +FT fine-tuning and self-bootstrapping, we define two extra +FT variants that match the fine-tuning budget of self-bootstrapping. In +FT$_{short}$ we fine-tune until the budget of self-bootstrapping's Stage 1 is matched. In +FT$_{long}$, we fine-tune until the total training budget of self-bootstrapping is matched.

3.4 Model Configurations and Notation

Different choices across the orthogonal axes of classification architecture, input, and fine-tuning strategy give rise to a wide spectrum of *model configurations*. In particular, we can independently choose **1)** between the Linear or Deep classification architecture; **2)** whether to include the information on

[5] See Appendix C for a probabilistic explanation. A similar idea for multi-task setups outside of the multilingual domain has been explored by, e.g., Cho et al. (2021).

[6] Recent work (Schick and Schütze, 2021; Li and Liang, 2021; Liu et al., 2021b; Shin et al., 2020) suggests that there exist more sophisticated prefixes/prompts and prompt-tuning methods. As our focus is not on conducting a large-scale analysis over different prompt-based conditioning, we leave this topic for future work.

[7] In our preliminary experiments, we found that self-bootstrapping-based fine-tuning still achieves better performance even if we perform Stage 1 with tunable text embeddings (i.e., standard fine-tuning). Freezing text embeddings in Stage 1 is an empirical decision, freezing them in Stage 2 is essential for self-bootstrapping to work.

the question type at input (^{Q}) or not; **3)** whether to apply standard fine-tuning from prior work (Pfeiffer et al., 2022), or rely on +FT or +SB fine-tuning strategies. On top of this, we can also vary **4)** the underlying model (M3P or UC2), and **5)** the transfer scenario (zero-shot versus few-shot). For clarity of presentation, unless noted otherwise, we always assume zero-shot scenarios and Deep classification architecture. Moreover, different variants are also labelled in a systematic manner using abbreviations introduced in §3.1-§3.3: e.g., *M3P+SB* means that we apply self-bootstrapping on the underlying M3P model (with Deep architecture assumed). In another example, *UC2^{Q}+FT$_{long}$* means that we apply the *long* variant of +FT fine-tuning (see §3.3) with UC2 as the underlying model, and we condition the model on the information about question types.

4 Analysis Methods

The VQA task is inherently multimodal—a model is required to reason over both images and questions in textual form to solve the task. However, as with some unimodal text-only tasks (Gururangan et al., 2018; Poliak et al., 2018) VQA models might also be prone to 'taking shortcuts', that is, exploiting spurious correlations and artifacts of data creation. In other words, the VQA model might circumvent the multimodal aspect and only focus on a single modality to solve the task (Agrawal et al., 2016, 2018). Therefore, to better understand the multimodal reasoning abilities of VQA models in cross-lingual transfer, we propose several diagnostic approaches and methods that ablate the input features of the models, inspired by the diagnostic methods of Frank et al. (2021) and Shrestha et al. (2020) in monolingual setups. They should provide us with deeper insights into the inner workings of cross-lingual VQA models.

4.1 Unimodal *Evaluation*

The first set of analyses involves a combination of standard multimodal (**MM**) training with unimodal inference/evaluation. During training, we pass both visual features and text tokens into the model. However, at inference, we provide the model with features of only one modality (Visual modality: **V** or Text: **T**). This naturally gives rise to the following two experimental setups:

MM-V: When evaluating on xGQA's test set, we pass only a single '?' as textual input to the model, while the standard visual features are used.

MM-T: At inference, we zero out all visual features (e.g., object features, spatial features), only providing the model with the total number of objects detected; the unchanged questions in the textual form are provided to the model.

4.2 Unimodal *Training* and *Evaluation*

Next, we probe purely unimodal models *trained* on a single modality (**V** or **T**): during training, the model is provided only with visual features or text tokens; at inference, we again only provide the model with unimodal features from the same modality. This creates three experimental setups:

V-V: We pass only '?' as a (placeholder) textual input to the model, while the standard visual features (from the full multimodal model) are used.

T-T: All visual features are zeroed out; we only provide the number of objects detected; the unchanged questions in the textual form are provided.

T^{G}-T^{G}: We randomly sample object features from a Gaussian distribution with a mean and a standard deviation that match the actual object feature distribution for that image. Spatial features and the number of objects detected are kept as in the full MM model. The standard unchanged questions in the textual form are provided to the model.

5 Experimental Setup

Pretrained Models and Data. As introduced in §2, we 1) rely on two standard state-of-the-art multimodal multilingual transformers (M3P, UC2; Ni et al., 2021; Zhou et al., 2021) as the underlying pretrained models, and 2) conduct all evaluations on the standard monolingual English GQA dataset, and its multilingual extension: xGQA.

The GQA dataset consists of two training sets: **full** and **balanced**. The full dataset contains 113K images and 22M questions, whereas the balanced dataset consists of 1.7M data samples. The dataset also contains a balanced test-dev set with 12,578 questions and 398 images for evaluation. In xGQA, the questions are manually translated from the GQA test-dev set into 7 different languages: Bengali, Chinese (simplified), German, Indonesian, Korean, Portuguese, and Russian. xGQA provides a zero-shot evaluation set and a different training/evaluation set for the few-shot setting. Please see the original paper for details.

Training Details and Hyperparameters. Following the recommendations from Bugliarello et al.

(2022), and due to a large number of experiments, we predominantly run training on the more lightweight *balanced* subset of GQA.[8] We also define a total training budget of 6 epochs (less than 24 hours of training). For the self-bootstrapping procedure, this means the total training time (Stage 1 + Stage 2) is equal to 6 epochs.[9]

6 Results and Discussion

In §6.1, we discuss the results of the different modeling approaches across the three dimensions (see §3): classification architectures, input signals and fine-tuning strategies. A finer-grained analysis pertaining to different structural question types is provided in §6.2. Finally, in §6.3 we delve deeper into the VQA models' susceptibility towards exploiting unimodal biases and artifacts of the VQA datasets, relying on model variants discussed in §4.

6.1 Model Configurations

A summary of the results with a wide spectrum of possible model configurations (see §3.4) is provided in Table 1, with accuracy as the main metric.

First, an interesting trend emerges: different model configurations have *no* significant effect on performance in the source language (English), especially so for the better-performing pretrained model UC2. However, variations in different modelling choices from §3 do show *considerable* impact on cross-lingual transfer performance: we report gains by more than 16 and 13 absolute accuracy points for M3P and UC2, respectively.

Classification Architectures. Surprisingly, simply adding additional non-linear layers to the prediction head has a considerable impact on the cross-lingual transfer performance of the baseline models (especially for the M3P model) while performance in the source language stays nearly the same (Table 1, Group G1). Put simply, a deeper classification architecture seems to benefit cross-lingual transfer performance, and the extent of its impact cannot be captured by monolingual English-only evaluation.[10] Another key observation is that the impact of depth is model-dependent with stronger configurations. While it yields large gains when we start from the baseline transfer models (G1), the gains from the classification architecture are less pronounced or even non-existent, e.g., for the best-performing UC2^{Q} + SB model variant (see Group G4): 39.87 (Deep) versus 40.89 (Linear).[11]

Input Signal. The large number of output classes of GQA potentially results in a noisy distribution over the predicted labels when sentences in a different language are passed into the model. We find that including the question-type token (Q) improves the average cross-lingual zero-shot transfer accuracy by more than 10% relatively for both M3P^{Q} and UC2^{Q} (Table 1, Group G2). This modelling decision again has an inconsequential impact on the source language but suggests that the question-type token can partially mitigate the poor performance of cross-lingual transfer. A comparison of G3 versus G4 models in Table 1 demonstrates that including the question type at input yields gains of almost 6 accuracy points with M3P, and more than 3 points with UC2, with especially large gains for Bengali as the lowest-performing language.

Fine-tuning Strategy. Freezing the embeddings to mitigate a shift in the multilingual embedding space results in positive gains for cross-lingual scenarios (Table 1, Group G4).[12] The self-bootstrapping strategy (+SB with and without Q) achieves further gains over both +FT embedding-freezing experimental setups. At the same time, it also yields much lower variance across languages (with Q). This validates that resetting parameters with self-bootstrapping positively impacts model performance, and supports our hypothesis that first fixing the classifier weights to a good value leads to better performance and lower variance.[13]

[8] Another established yet less efficient training procedure is to train on the full GQA dataset first, then further train on the balanced dataset (Li et al., 2020). This procedure can produce good results on the English evaluation dataset at the cost of a substantial increase in computation demands (∼4 days on one NVIDIA V100 for one model). Furthermore, our initial experiments have indicated that training with the balanced set performs similarly to the previously reported baselines in the xGQA paper while using substantially less computing. We stress that we also further run experiments under the more demanding training regime (Li et al., 2020) with the best-performing model configuration from our experiments. For more details, we refer the reader to §7.

[9] For detailed hyperparameters and breakdown of training times, please see Appendix A.

[10] Further, to isolate the source of these improvements, we conducted additional experiments by removing the LayerNorm from the deeper architecture. The results are in Table 8 in the appendix. Removing the LayerNorm reduces zero-shot accuracy of M3P, and increases the average variance of UC2.

[11] The gains from classification architecture remained for the M3P model variant: 36.19 (Deep) versus 18.24 (Linear).

[12] However, we do witness a slight decrease for UC2 when training for longer.

[13] The average +SB results of UC2 are statistically significant against UC2^{Q} and UC2^{Q} +FT$_{short}$ ($p < 0.05$).

	Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
G1	M3P * (Linear)	51.88±0.7	27.45±5.8	16.33±8.3	13.70±5.4	25.25±11.4	10.59±3.4	21.10±3.4	20.95±3.3	19.34
	M3P*	51.66±0.6	35.33±5.4	27.80±10.9	25.55±11.4	30.54±9.8	17.94±8.6	30.61±7.2	29.74±6.6	28.22
G2	M3P^{Q}	50.90±0.5	37.95±1.5	35.06±2.6	32.31±3.4	36.56±2.0	27.69±1.8	36.64±2.4	37.30±4.6	34.79
G3	M3P + SB	47.26±1.0	35.71±6.1	29.70±8.2	30.33±8.3	28.16±2.7	20.70±3.9	34.65±6.5	34.63±6.9	30.56
G4	M3P^{Q} + FT$_{short}$	49.48±0.3	38.68±2.6	34.94±2.2	34.17±2.6	**37.18**±2.4	**30.00**±2.2	37.35±1.9	37.57±2.4	35.56
	M3P^{Q} + FT$_{long}$	51.00±0.9	38.42±2.1	35.05±2.1	33.38±2.5	36.24±2.3	27.77±1.7	36.78±2.3	37.42±2.0	35.01
	M3P^{Q} + SB	46.70±0.7	**39.52**±1.3	**36.15**±0.9	**35.67**±1.1	36.73±1.6	29.75±1.4	**37.59**±0.8	**37.93**±0.9	**36.19**
G1	UC2 * (Linear)	57.83±0.3	40.57±1.7	35.54±3.4	16.95±6.1	34.18±0.8	8.53±1.9	24.90±3.7	24.05±4.6	26.39
	UC2*	58.31±0.2	41.33±1.6	34.77±2.2	23.87±1.5	34.79±1.3	11.82±1.9	29.30±4.5	29.41±3.7	29.33
G2	UC2^{Q}	58.35±0.4	45.13±0.8	42.85±0.9	31.33±1.0	35.64±0.9	24.86±0.6	37.19±0.6	38.61±0.9	36.52
G3	UC2 + SB	58.52±0.4	48.51±1.3	43.97±0.3	35.08±2.0	37.33±3.2	19.09±4.5	35.29±2.9	35.99±3.5	36.46
G4	UC2^{Q} + FT$_{short}$	57.83±0.5	47.17±1.6	45.59±0.9	34.19±0.7	37.04±1.1	24.94±0.5	38.32±1.2	39.96±1.4	38.17
	UC2^{Q} + FT$_{long}$	58.15±0.6	44.27±0.5	42.49±0.4	29.75±0.3	36.81±0.4	24.48±0.2	35.39±0.4	37.32±0.4	35.79
	UC2^{Q} + SB	58.57±0.2	**49.51**±1.1	**46.52**±0.9	**36.48**±1.3	**38.92**±1.3	**26.23**±1.5	**39.76**±0.6	**41.72**±0.3	**39.87**

Table 1: Zero-shot transfer results on xGQA. Avg. refers to the average accuracy across languages excluding English. Group G1: baselines. $*$: our runs of baselines trained on balanced GQA. Group G2: results using a question-type token. Group G3: results using self-bootstrapping (+SB). Group G4: combining different fine-tuning strategy with the use of question-type tokens. Best results in each column and per each pretrained model across Groups G1-G4 are shown in **bold**. Results are averaged across 4 random seeds.

6.2 Performance across Question Types

Finer-grained results per individual question type are summarized in Figure 2, where we compare the baseline models with the best-performing variant, which utilizes the question-type at the input and the self-bootstrapping strategy. In sum, we observe gains across all structural question-types for such ^{Q}+SB model configurations, both for M3P and UC2. Performance on *Query* and *Choose* questions meets substantial gains, suggesting that improving the alignment between multilingual text embeddings has a positive effect on performance, especially for non-binary, free-form question-types.[14]

6.3 Multi-Modal versus Unimodal VQA?

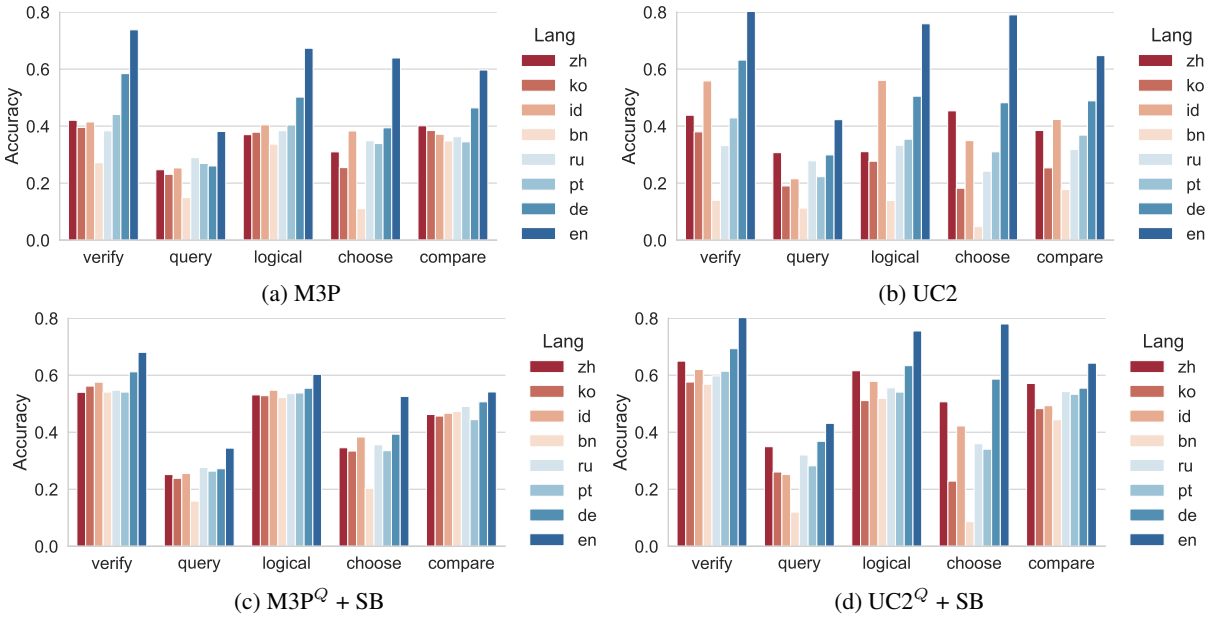
We further aim to understand whether or not the underlying models learn to rely on a single modality to make predictions, either due to spurious correlations in the data or the model's inability to effectively combine multi-modal features. The main results are provided in Table 2.

Unimodal *Evaluation*. The scores of MM-T/MM-V ablations reveal the sensitivity to missing features in each input modality at test time. We observe a drop in accuracy of more than 50% across all question types in the MM-T/MM-V experiments compared to their counterparts that assume 'full-feature' multi-modal input at inference. Moreover, *Verify*, *Logical* and *Compare* questions seem more dependent on text features. The results confirm that the trained model needs both modalities to achieve good cross-lingual performance, although not at equal proportions. In other words, high zero-shot transfer performance observed in our experiments are obtained by leveraging both modalities in synergy, and not by 'taking unimodal shortcuts' (§4).

Unimodal *Training* and *Evaluation*. V-V/T-T/T^{G}-T^{G} experiments reveal the worst-case exploitation of the data biases in modalities by the models. The results suggest that a majority of the final performance can be attained with text features in fine-tuned models for the *Logical*, *Verify*, and *Compare* question types. Therefore, the results indicate that these question types contain modality biases that can be exploited by unimodal VQA architectures. The exploitable data biases could also explain the observations from prior experiments, where we noted that the VQA models assign different attention to the text and vision features. We suspect this could also explain the asymmetrical attention over modalities, observed by Frank et al. (2021) in monolingual multi-modal models.

Biases across Question Types. Unimodally trained models can only attain ∼20% (M3P) and ∼26% (UC2) accuracy at best for the *Query* ques-

[14] See exact numerical values in Appendix B.



(a) M3P (b) UC2

(c) M3P^{Q} + SB (d) UC2^{Q} + SB

Figure 2: Zero-shot cross-lingual transfer performance across individual question types in GQA and xGQA.

M3P	V-V	T-T	T^{G}-T^{G}	M3P^{Q}	M3P^{Q}+SB	MM-V	MM-T
Verify	45.19	53.98	54.88	58.35	55.98	0.1	18.59
Logical	43.18	51.66	53.06	53.89	53.65	0.0	19.87
Compare	27.76	46.22	39.64	45.82	47.14	0.1	17.85
Query	2.63	4.39	11.42	21.86	**24.50**	6.81	4.46
Choose	1.21	8.52	22.26	29.43	**33.57**	2.08	12.22
UC2	V-V	T-T	T^{G}-T^{G}	UC2^{Q}	UC2^{Q}+SB	MM-V	MM-T
Verify	44.60	51.87	57.00	59.94	61.70	4.21	24.91
Logical	44.26	50.78	52.57	54.87	56.49	6.27	21.12
Compare	33.45	40.55	46.91	49.15	51.73	2.85	21.08
Query	3.39	6.23	12.11	23.94	**27.88**	7.30	0.02
Choose	1.39	17.24	23.76	29.66	**36.14**	2.27	0.14

Table 2: Zero-shot transfer results of M3P^{Q}/UC2^{Q} trained and tested with visual features only (V-V), text features only (T-T), text features with partial visual features (T^{G}-T^{G}), as well as of M3P^{Q}+SB/UC2^{Q}+SB trained using all features, but exposed only to visual features (MM-V) or text features (MM-T) at inference (§4). The scores are averaged over all target languages in xGQA, excluding English.

tion type, with similar trends observed for *Choose*. Exposing the models to increasingly more visual features (from T-T over T^{G}-T^{G} to the full multimodel) yields large performance gains. It thus indicates that *Query* and *Choose* questions contain fewer exploitable data biases, and additional image-text grounding could help improve predictions. Further, Table 2 also reveals that more sophisticated fine-tuning strategies such as self-bootstrapping, which prevent multilingual text embedding shifts, are an effective way to improve performance on these two (most challenging) question types.

In sum, it is crucial to conduct such finer-grained analyses across different question types in the multilingual VQA tasks, and not treat them equally with only a global accuracy metric. In particular, our results render *Query* and *Choose* question types as by far the most challenging question types for cross-lingual transfer and the types that do not suffer from exploitable data biases. Future research in multilingual VQA should put more emphasis on such questions, and approaches that prevent the exploitation of unimodal data biases. Future research should also look beyond the question types currently covered by xGQA, and introduce even more challenging types.

7 Further Analyses

Training with Full English GQA. To validate the effectiveness of our approach in setups where more data in the source language is available, we additionally run experiments in another VQA setup: we train the best-performing method UC2^{Q}+SB for 5 epochs on the unbalanced English GQA dataset, followed by 2 epochs on the balanced dataset. Despite the fact that this variant leverages more source-language training data and consumes considerably more compute, we do not observe any gain on monolingual English performance, and observe only a small gain in the cross-lingual zero-shot setup: the accuracy score, averaged across all the target languages, increases from 39.87 to 40.51.[15]

[15]Table 9 in Appendix F provides per-language accuracy.

Method	0	1	5	48
M3P	35.58	37.62	39.29	42.28
M3P + SB	33.73	35.89	39.27	42.46
M3P^{Q}	33.81	35.40	37.80	41.87
M3P^{Q} + SB	37.14	37.50	38.16	40.00
UC2	30.15	36.09	38.67	44.37
UC2 + SB	38.09	40.51	42.14	46.68
UC2^{Q}	37.28	39.24	40.88	45.11
UC2^{Q} + SB	39.83	42.35	43.68	46.62

Table 3: Averaged few-shot (0/1/5/48-shot) accuracy scores on xGQA (excl. English) for selected models.

Few-shot Experiments. Besides the zero-shot transfer scenario—which is the primary focus of this work—we also evaluate whether similar findings extend to few-shot scenarios, where a handful of annotated examples in the target language is assumed. Following the standard setup of Lauscher et al. (2020) we start from the weights of the best-performing model, already fine-tuned on English VQA data. We then further fine-tune it on the few examples in the target language. In particular, we conduct few-shot experiments with 1, 5, and 48 images.[16] Following Pfeiffer et al. (2022) we fine-tune for 10 epochs, with a learning rate of 5e-5.[17]

The results are summarized in Table 3, and indicate two key findings. First, we corroborate findings from prior work, where it was shown that fine-tuning on an increasing number of shots/examples in the target language generally improves model performance.[18] Second, although baseline models are able to recover more performance from zero-shot to few-shot setups, our best-performing configuration with UC2 still significantly outperforms the baseline.[19] These results indicate that few-shot fine-tuning is an *additional* cost-efficient approach, orthogonal to our modelling enhancements from §3, to further improve VQA model performance in the target language.

8 Related Work

Transformer-based models trained on multimodal data (Tan and Bansal, 2019; Li et al., 2020; Cho et al., 2021; Shen et al., 2021; Kamath et al., 2021, *inter alia*) have demonstrated impressive results on English-only VQA tasks. However, as training and evaluation data has previously only been available in high resource languages (Elliott et al., 2016, 2017; Barrault et al., 2018; Gao et al., 2015), progress in multilingual vision-and-language learning has not kept pace.

More comprehensive multilingual multimodal benchmarks have been developed only recently (Srinivasan et al., 2021; Su et al., 2021; Liu et al., 2021a; Pfeiffer et al., 2022; Wang et al., 2021; Bugliarello et al., 2022, *inter alia*) making it possible to evaluate multimodal models which have either been pretrained on multilingual data (Ni et al., 2021; Zhou et al., 2021) or extended to unseen languages (Liu et al., 2021a; Pfeiffer et al., 2022).

Our work complements this recent line of work by delving deeper into cross-lingual visual question answering, again highlighting the inherent difficulty of multilingual multimodal learning.

9 Conclusion

In this work, we provide an extensive analysis of the issues present in VQA-related multilingual vision-and-language learning, aiming to inspire new solutions that can improve cross-lingual VQA performance. To this end, we studied simple yet effective methods that increase previously low transfer performance and thus substantially reduce the gap to monolingual English performance. This has been achieved through more sophisticated classification architectures, fine-tuning strategies, and modifications of the model input via question-type conditioning. We also conducted further analyses and empirical comparisons, including detection of unimodal biases in training and evaluation data, fine-grained analyses across different question types, and comparisons across different multilingual Transformer models and transfer scenarios. We hope that this work will spark more interest and inspire future research on cross-lingual VQA tasks in particular, as well as on multilingual multimodal learning in general.

[16] We choose 1 and 5 shots because these are typical in few-shot training setups (Zhao et al., 2021). 48 shots are the maximum available training data for the few-shot evaluation.

[17] For reproducibility, see again Appendix A for a detailed list of hyperparameters.

[18] See Table 10 in Appendix G for full 'uncompressed' scores across models and languages.

[19] We attribute the on-par performance across M3P variants to M3P's sensitivity to initialization and high variance.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look

and answer: Overcoming priors for visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4971–4980. Computer Vision Foundation / IEEE Computer Society.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society.

Jordan T. Ash and Ryan P. Adams. 2020. On warm-starting neural network training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulic. 2022. IGLUE: A benchmark for transfer learning across modalities, tasks, and languages. *arXiv preprint*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.

Haim Dubossarsky, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2020. The secret is in the spectra: Predicting cross-lingual task performance with spectral similarity measures. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2377–2390, Online. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9847–9857, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question answering. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, page 2296–2304, Cambridge, MA, USA. MIT Press.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (GELUs). *arXiv preprint*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 4411–4421.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6700–6709. Computer Vision Foundation / IEEE.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. MDETR - modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1780–1790.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing, pages 4483–4499, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision (ECCV) 2020 - 16th European Conference, Glasgow, UK*, volume 12375, pages 121–137. Springer.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021a. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *arXiv preprint*.

Francisco Massa and Ross Girshick. 2018. maskrnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`. Accessed: [Insert date here].

J. William Murdock, Aditya Kalyanpur, Chris Welty, James Fan, David A. Ferrucci, David Gondek, Lei Zhang, and Hiroshi Kanayama. 2012. Typing candidate answers using type coercion. *IBM J. Res. Dev.*, 56(3):7.

Minheng Ni, Haoyang Huang, Lin Su, Edward Cui, Taroon Bharti, Lijuan Wang, Dongdong Zhang, and Nan Duan. 2021. M3P: learning universal representations via multitask multilingual multimodal pretraining. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3977–3986. Computer Vision Foundation / IEEE.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O. Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. 2022. xGQA: Cross-lingual visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 180–191. Association for Computational Linguistics.

John M. Prager. 2006. Open-domain question-answering. *Found. Trends Inf. Retr.*, 1(2):91–231.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can CLIP benefit vision-and-language tasks? *arXiv preprint*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4222–4235, Online. Association for Computational Linguistics.

Robik Shrestha, Kushal Kafle, and Christopher Kanan. 2020. A negative case analysis of visual grounding methods for VQA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8172–8181, Online. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. *WIT: Wikipedia-Based Image Text Dataset for Multimodal*

Multilingual Machine Learning, page 2443–2449. Association for Computing Machinery, New York, NY, USA.

Lin Su, Nan Duan, Edward Cui, Lei Ji, Chenfei Wu, Huaishao Luo, Yongfei Liu, Ming Zhong, Taroon Bharti, and Arun Sacheti. 2021. GEM: A general evaluation benchmark for multimodal tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2594–2603, Online. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. 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 (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Josiah Wang, Pranava Madhyastha, Josiel Figueiredo, Chiraag Lala, and Lucia Specia. 2021. MultiSubs: A large-scale multimodal and multilingual dataset. *arXiv preprint*.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. 2021. LiT: Zero-shot transfer with locked-image text tuning. *arXiv preprint*.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics.

Mingyang Zhou, Luowei Zhou, Shuohang Wang, Yu Cheng, Linjie Li, Zhou Yu, and Jingjing Liu. 2021. UC2: Universal cross-lingual cross-modal vision-and-language pre-training. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4155–4165. Computer Vision Foundation / IEEE.

A Details of Training Setup and Hyperparameters

The hyperparameters used to train M3P and UC2 models are summarized in Table 4. We conducted all experiments with either an NVIDIA V100 or A100 GPU. The numbers of training epochs across different model configurations are summarized in Table 5. Training time for the rest of our zero-shot experiments ranges from 8 to 24 hours.

We searched over the following learning rates: 2e-5, 5e-5, and 1e-4.

We note that experiments which rely on Full GQA data ($-full$) have a significantly different training budget. This setup followed the previously recommended training setup of Li et al. (2020).

We use pretrained, state-of-the-art Transformer-based M3P and UC2 models (open-sourced), which build on pre-extracted image features from pre-trained object detectors. M3P was pretrained via masked language modeling, cross-lingual masked language modeling and cross-modal text-image region alignments objectives. UC2 was trained similar to M3P with an additional auxiliary task (i.e. translation).

We extracted image features for M3P using the ResNet-101 backbone using the `vqa-maskrcnn-benchmark` model (Massa and Girshick, 2018) (100 bounding boxes), and we extracted image features for UC2 using the bottom-up-attention (Anderson et al., 2018) (100 bounding boxes). The feature extraction procedures are different because the pretrained M3P and UC2 use different features.

Name	Value
learning rate (M3P)	0.00002
learning rate (UC2)	0.0001
train batch size	192
warmup steps	0
weight decay	0.05
max grad norm	1
dropout rate	0.5
max seq length	70
max img seq length	50
f_{trans} hidden dim	768
optimizer	AdamW

Table 4: Hyperparameters.

B Structural Question Types in GQA and xGQA

There are 5 different structural questions types in GQA and, consequently, in xGQA. We used the exact lowercased name of each question type as the `QType` token in our experiments, namely: *verify, logical, compare, query, and choose*. The text input follows the format of: `'[QType] : [Question]'` (see again §3.2). Some example questions for each question type are as follows:

Verify: Yes/No questions. E.g. *Do you see books near the device that looks gray? Is the bus blue?*

Logical: Questions that require logical inference. E.g. *Is there any motorcycle or ball in the scene? Does the dirt look brown and fine?*

Compare: Comparison questions between two or more objects. E.g. *Who seems to be younger, the boy or the woman?*

Query: Open questions. E.g. *What color are the pants? What is the animal that is standing on the grass called?*

Choose: Choose from two presented alternatives. E.g. *Is it red or blue? What size is the jacket, small or large?*

Verify and Logical question types are binary question types (Yes/No). The question type distribution in the test-dev set of GQA is given in Table 6, while we provide average accuracy scores

Exp.	Balanced Stage 1	Balanced Stage 2	Total Ep.	Time
M3P^{Q}	6	-	6	<24hrs
M3P^{Q} + FT $_{short}$	4	-	4	<24hrs
M3P^{Q} + FT $_{long}$	6	-	6	<24hrs
M3P^{Q} + SB	4	2	6	<24hrs
UC2^{Q}	6	-	6	<24hrs
UC2^{Q} + FT $_{short}$	3	-	3	<24hrs
UC2^{Q} + FT $_{long}$	6	-	6	<24hrs
UC2^{Q} + SB	3	3	6	<24hrs
Exp.	**Full Stage 1**	**Balanced Stage 2**	**Total Ep.**	**Time**
$-full$	5	2	7	4 days

Table 5: Training epochs and times. Full and Balanced indicate the GQA subset used for training. The self-bootstrapping experiments are initialized from the weights of *short* experiments.

Question Type	Count
Verify	2,251
Logical	1,803
Compare	5,89
Query	6,804
Choose	1,129

Table 6: GQA test-dev set: distribution of questions over question types.

Question Type	M3P	M3P + SB	M3P^{Q}	M3P^{Q} + SB
Verify	40.15	44.45	**58.35**	55.98
Logical	39.15	45.29	**53.89**	53.65
Compare	35.95	40.75	45.82	**47.14**
Query	**24.57**	21.42	21.86	24.50
Choose	30.63	29.07	29.43	**33.57**
Question Type	**UC2**	**UC2 + SB**	**UC2^{Q}**	**UC2^{Q} + SB**
Verify	41.55	51.27	59.94	**61.70**
Logical	35.40	48.32	54.87	**56.49**
Compare	34.48	44.71	49.15	**51.73**
Query	23.18	27.68	23.94	**27.88**
Choose	29.51	36.62	29.66	**36.14**

Table 7: Average accuracy on different structural question types from xGQA (excluding English). M3P and UC2 are using Deep architecture.

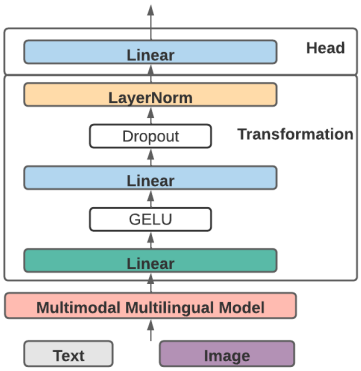


Figure 3: Deep(er) classification architecture (see §3.1). The first linear layer in the transformation uses an orthogonal initializer.

over all target languages in xGQA (excluding English), with a representative set of models, in Table 7.

C Conditioning on Question-Type Tokens: A Probabilistic Perspective

For simplicity, let x and y represent inputs and labels of the model. Let ϕ represent task-specific information in the form of the question-type token, which follows a categorical distribution. In a standard classification task, our goal is to learn a discriminative model $P(y|x)$. Decomposing this using ϕ, we have:

$$P(y|x) = \sum_{\phi} P(y|x,\phi)P(\phi|x),$$

which is a mixture model with mixture components $P(y|x,\phi)$ and mixture weights $P(\phi|x)$.

Without knowing ϕ, the model has to learn the mixture structure or mixture weights.

By choosing a ϕ that represents the question-type, we essentially consider $P(\phi|x)$ to be deterministic (i.e. a delta function centered at the correct question-type for each x), as a single x can only belong to one question type.

Hence, learning $P(y|x)$ is simplified to learning the mixture components $P(y|x,\phi)$ without having to learn the mixture structure or mixture weights. We further take advantage of the ability of neural networks to learn distinct distributions over the label space based on an additional input (conditioning) variable.

D Classification Architecture with and without Layer Normalization

Deeper variant of the classification architecture from §3.1 is illustrated in Figure 3. The *Multimodal Multilingual Model* block in Figure 3 denotes one of the two pretrained multimodal multilingual models used throughout the (main) paper: UC2 and M3P.

We further experimented with another variant of the architecture, where we removed the layer normalization (LayerNorm) layer. The results of this variant are available in Table 8.

In a nutshell, LayerNorm has more impact on M3P's zero-shot transfer accuracy scores than on UC2. However, the variance of UC2 results increases with the removal of LayerNorm.

E Accuracy vs. Total Training Epochs

We conducted experiments with different total numbers of training epochs with M3P in order to understand the effect of the self-bootstrapping fine-tuning strategy. We experimented with the following three model configurations across different setups:

1. M3P^{Q} + FT: We train the M3P^{Q} model with text embeddings frozen for 4, 6 and 10 epochs.

2. M3P^{Q*} + FT: We initialize the M3P^{Q} model with fine-tuned weights (including transformation, classification head) from 1 (i.e., the variant above), and train for 4 epochs. We continue to fine-tune the model for 2 or 5 more epochs after resetting the learning rate and the optimizer.

3. M3P^{Q} + SB: We train the M3P^{Q} model with self-bootstrapping and the classification head weights from variant 1 above, and do it for 4 epochs. We continue to fine-tune the model for 2 or 5 epochs.

Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
M3P (Linear)	51.88$_{\pm 0.7}$	27.45$_{\pm 5.8}$	16.33$_{\pm 8.3}$	13.70$_{\pm 5.4}$	25.25$_{\pm 11.4}$	10.59$_{\pm 3.4}$	21.10$_{\pm 3.4}$	20.95$_{\pm 3.3}$	19.34
M3P w/ LN	51.66$_{\pm 0.6}$	35.33$_{\pm 5.4}$	27.80$_{\pm 10.9}$	25.55$_{\pm 11.4}$	30.54$_{\pm 9.8}$	17.94$_{\pm 8.6}$	30.61$_{\pm 7.2}$	29.74$_{\pm 6.6}$	28.22
M3P w/o LN	50.89$_{\pm 1.0}$	32.92$_{\pm 5.6}$	22.14$_{\pm 8.0}$	20.33$_{\pm 9.1}$	25.44$_{\pm 6.5}$	16.88$_{\pm 8.0}$	29.40$_{\pm 7.8}$	29.31$_{\pm 7.9}$	25.20
UC2 (Linear)	57.83$_{\pm 0.3}$	40.57$_{\pm 1.7}$	35.54$_{\pm 3.4}$	16.95$_{\pm 6.1}$	34.18$_{\pm 0.8}$	8.53$_{\pm 1.9}$	24.90$_{\pm 3.7}$	24.05$_{\pm 4.6}$	26.39
UC2 w/ LN	58.31$_{\pm 0.2}$	41.33$_{\pm 1.6}$	34.77$_{\pm 2.2}$	23.87$_{\pm 1.5}$	34.79$_{\pm 1.3}$	11.82$_{\pm 1.9}$	29.30$_{\pm 4.5}$	29.41$_{\pm 3.7}$	29.33
UC2 w/o LN	58.03$_{\pm 0.5}$	42.74$_{\pm 1.4}$	37.84$_{\pm 3.0}$	24.91$_{\pm 5.2}$	33.56$_{\pm 1.6}$	13.21$_{\pm 4.5}$	29.99$_{\pm 4.5}$	29.47$_{\pm 6.3}$	30.25

Table 8: Zero-shot cross-lingual transfer results with and without LayerNorm.

Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
UC2^{Q} + SB - full	57.88$_{\pm 0.2}$	50.52$_{\pm 0.5}$	47.63$_{\pm 0.2}$	37.56$_{\pm 1.7}$	40.37$_{\pm 1.6}$	25.25$_{\pm 1.4}$	40.56$_{\pm 0.2}$	41.67$_{\pm 0.8}$	40.51

Table 9: Zero-shot results when the models are trained with Full GQA data.

We also run similar variants with UC2 as the underlying model with shorter training epochs. These variants are: UC2^{Q} + FT / UC2^{Q*} + FT / UC2^{Q} + SB where superscripts and acronyms remain the same as the M3P variants. Results of these experiments are provided in Figure 4a (M3P) and Figure 4b (UC2).

We observe that the gains in cross-lingual transfer with +FT variants diminish or even start decreasing with the increase of training time. Similar results are observed when we reset the learning rate, weight decay and optimizer after training for 4 epochs. We also find that self-bootstrapping training continually improves the results, even with less additional total training epochs.

Moreover, the performance of self-bootstrapping is considerably more stable (lower variance) across random seeds, even though its classification heads are initialized from the corresponding trained weights from the M3P^{Q} + FT experiments.

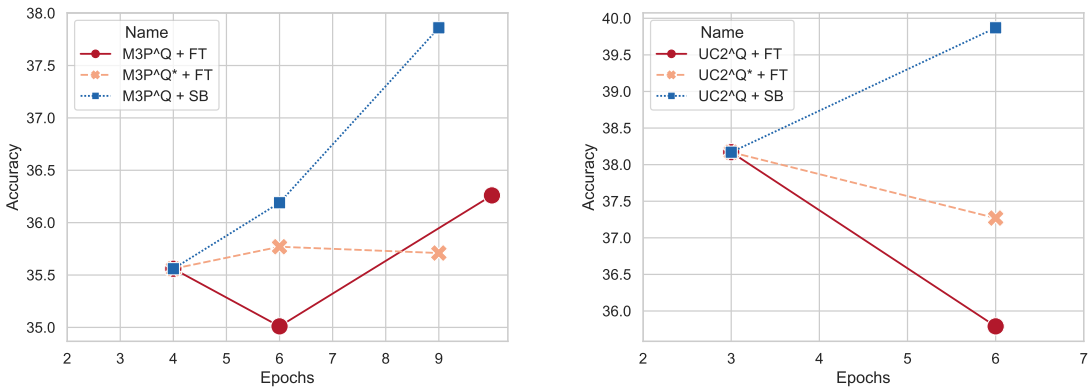
We also observe an increase in zero-shot transfer accuracy scores with more epochs of training in Stage 2 of self-bootstrapping. However this results in much longer training times, which may not be realistic for academic and even some industry settings.

F Results with Full GQA Data

It is worth to note that the experiments trained with full GQA data ($-full$) have a significantly different (and larger) training budget (see §7). We follow the previously recommended total training budget of Li et al. (2020), and combine with our self-bootstrapping fine-tuning strategy. Table 9 shows the detailed results.

G Few-shot Experiments: Full Results

Table 10 shows the detailed results of our few-shot experiments, where the summary table is provided in the main paper: Table 3 in §7.



(a) M3P (b) UC2

Figure 4: Average accuracy versus total training epochs.

Lang	**Method**	0	1	5	48
de	M3P	39.45	40.76	41.88	44.20
	M3P + SB	39.25	39.72	40.98	43.08
	M3P^{Q}	36.84	38.28	40.11	43.18
	M3P^{Q} + SB	40.99	40.74	40.39	41.71
zh	M3P	35.76	37.65	40.28	42.18
	M3P + SB	32.96	35.55	38.24	41.15
	M3P^{Q}	33.74	35.97	37.95	41.28
	M3P^{Q} + SB	36.95	36.88	37.60	39.38
ko	M3P	34.53	36.58	36.79	39.61
	M3P + SB	36.04	36.92	37.31	39.41
	M3P^{Q}	31.96	32.77	35.39	40.45
	M3P^{Q} + SB	35.78	35.38	37.46	38.99
id	M3P	38.38	39.39	40.63	42.57
	M3P + SB	29.17	36.94	39.49	41.16
	M3P^{Q}	34.69	35.37	38.50	42.12
	M3P^{Q} + SB	37.75	36.25	38.57	39.92
bn	M3P	24.27	30.53	34.72	40.73
	M3P + SB	22.71	25.94	33.96	40.46
	M3P^{Q}	27.67	29.95	33.15	40.36
	M3P^{Q} + SB	30.50	31.77	34.08	39.24
pt	M3P	38.19	38.35	40.54	44.27
	M3P + SB	38.17	37.98	39.35	43.01
	M3P^{Q}	36.87	37.93	39.72	43.08
	M3P^{Q} + SB	38.56	39.24	39.71	40.56
ru	M3P	38.46	40.06	40.22	42.38
	M3P + SB	37.84	38.20	38.54	41.95
	M3P^{Q}	34.86	37.51	39.82	42.64
	M3P^{Q} + SB	38.76	39.74	39.39	40.19
de	UC2	40.39	44.23	46.03	49.51
	UC2 + SB	49.52	50.10	50.30	51.42
	UC2^{Q}	46.26	46.95	46.94	49.42
	UC2^{Q} + SB	50.23	50.70	50.53	51.39
zh	UC2	37.26	41.70	42.68	46.32
	UC2 + SB	43.54	46.30	47.17	48.80
	UC2^{Q}	43.89	44.90	45.56	47.24
	UC2^{Q} + SB	46.37	47.82	48.32	48.47
ko	UC2	25.93	32.63	36.11	41.11
	UC2 + SB	36.48	36.73	37.84	43.90
	UC2^{Q}	32.45	35.79	37.37	42.04
	UC2^{Q} + SB	37.80	39.05	40.68	43.38
id	UC2	35.76	39.35	40.12	44.24
	UC2 + SB	32.70	38.18	42.88	47.06
	UC2^{Q}	36.70	39.54	41.40	45.78
	UC2^{Q} + SB	38.34	42.16	42.33	47.01
bn	UC2	12.00	21.91	25.95	39.75
	UC2 + SB	24.66	29.76	32.31	42.08
	UC2^{Q}	25.29	27.68	32.75	39.82
	UC2^{Q} + SB	24.07	31.67	35.77	42.83
pt	UC2	29.79	33.86	40.18	45.23
	UC2 + SB	38.79	40.49	41.95	47.34
	UC2^{Q}	36.60	39.56	40.67	46.45
	UC2^{Q} + SB	40.36	42.65	43.79	47.63
ru	UC2	29.94	38.97	39.66	44.41
	UC2 + SB	40.93	42.02	42.54	46.15
	UC2^{Q}	39.76	40.26	41.46	45.04
	UC2^{Q} + SB	41.62	42.42	44.32	45.63

Table 10: Few-shot transfer average accuracy with different amounts of training data. M3P and UC2 are using the deeper classification architecture.