Task Progressive Curriculum Learning for Robust Visual Question Answering

Ahmed Akl[1,2]
ahmed.akl@(griffithuni.edu.au,data61.csiro.au)
Abdelwahed Khamis[2]
abdelwahed.khamis@data61.csiro.au
Zhe Wang[1]
zhe.wang@griffith.edu.au
Ali Cheraghian[2]
Ali.Cheraghian@data61.csiro.au
Sara Khalifa[3]
sara.khalifa@qut.edu.au
Kewen Wang[1]
k.wang@griffith.edu.au

[1] School of Information and Communication Technology
Griffith University
Australia
[2] Data61, CSIRO
Australia
[3] School of Information Systems
Queensland University of Technology
Australia

Abstract

Visual Question Answering (VQA) systems are notoriously brittle under distribution shifts and data scarcity. While previous solutions—such as ensemble methods and data augmentation—can improve performance in isolation, they fail to generalise well across *in-distribution (IID)*, *out-of-distribution (OOD)*, and *low-data* settings simultaneously. We argue that this limitation stems from the suboptimal training strategies employed. Specifically, treating all training samples uniformly—without accounting for question difficulty or semantic structure— leaves the models vulnerable to dataset biases. Thus, they struggle to generalise beyond the training distribution.

To address this issue, we introduce **Task-Progressive Curriculum Learning (TPCL)**—a simple, model-agnostic framework that progressively trains VQA models using a curriculum built by jointly considering *question type* and *difficulty*. Specifically, TPCL first groups questions based on their semantic type (e.g., yes/no, counting) and then orders them using a novel Optimal Transport-based difficulty measure. Without relying on data augmentation or explicit debiasing, TPCL improves generalisation across *IID*, *OOD*, and *low-data* regimes and achieves state-of-the-art performance on VQA-CP v2, VQA-CP v1, and VQA v2. It outperforms the most competitive robust VQA baselines by over **5%** and **7%** on VQA-CP v2 and v1, respectively, and boosts backbone performance by up to **28.5%**. *Our source code is available at* `https://github.com/AhmedAAkl/tpcl`.

1 Introduction

Visual Question Answering (VQA) is a challenging multi-modal task that requires the model to generate a correct answer given pair of image and question [4]. Numerous studies [1, 13, 41] have pointed out that VQA models are prone to biases within the dataset and rely on language bias within the dataset, making predictions based on superficial correlations between the question and answer rather than understanding the image. Consequently, these methods tend to perform well in the In-Distribution (ID) test scenario, where the answer distribution aligns closely with the training split, but they struggle in the Out-Of-Distribution (OOD) test scenario, where the answer distribution differs significantly or is even reversed.

To address this issue, many methods [9, 11, 13, 30, 32, 38], such as data augmentation and ensemble learning, were developed to enhance the VQA models' performance in the Out-Of-Distribution (OOD) dataset. *Data augmentation*(CSS [9], DGG [38], MMBS [32]) generates additional question-answer pairs for each sample in the original dataset to balance the distribution of training data. Such strategies may assign wrong answers to the produced samples [38] or destroy the semantics of generated questions [38]. *Ensemble learning* methods augment the VQA model with additional branches to identify the visual and/or linguistic biases and suppress them during the training (GenB [11], RUBi [8] and Q-Adv+DoE [29]). Such methods are sensitive to the underlying model architecture [38] [22].

We observe that many existing works ignore the linguistic difficulty associated with different question types. Most current debiasing approaches focus on identifying biased samples or augmenting the dataset, without considering the varying importance or complexity levels of training questions. For instance, in child language acquisition, Wh- questions are generally easier to comprehend and process compared to binary (yes/no) questions — an insight that remains largely unaddressed in VQA training strategies [23, 24]. To address this issue, we render the VQA problem as a multi-task learning (MTL) problem in which each task corresponds to a single question type. For example, all questions beginning with "How many...?" bear some semantic relatedness and can be grouped into a single smaller task. In light of this vision, we explore MTL solutions in VQA. One category demonstrated that **learning the tasks sequentially in an order determined by a curriculum** [27] is superior to **learning all the tasks simultaneously**. This builds on the established principle that models are more transferable between closely related tasks [27, 33]. Such task-based curriculum learning was employed in a number of applications [14, 27].

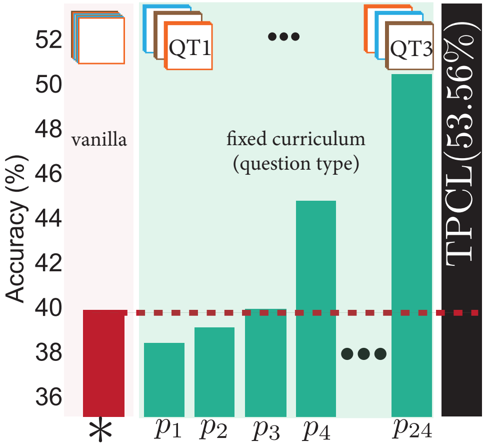

Figure 1: Encouraged by the unexpected advantage of fixed curricula over vanilla VQA training, we introduced TPCL, which achieves the highest performance. $p_1, \cdots p_{24}$ denote all possible permutations for four question-type (QT) tasks; Wh-, Binary, Number, Others.

Moreover, we conducted a pilot study to investigate the impact of different linguistic tasks ordering on the model performance compared to conventional training, Figure 1. For example, Order 1 is (binary, other, number, Wh-) questions, see appendix for other orders. This analysis suggests that instead of randomly sampling the training data, grouping the semantically related samples and processing them in a structured order improves the model's generalisation ability. Motivated by these findings, we introduce Task Progressive Curriculum Learning (TPCL), a novel training strategy that rendered the VQA task into a multi-task learning problem to improve the model generalisation. Surprisingly, this was not investigated in the VQA domain, and we took the first attempt. Specifically, TPCL splits the challenging VQA learning problem into smaller sub-problems (each constrained to semantically related samples). Then, it trains the model sequentially on sequences of tasks in each iteration. The sequences are judiciously sampled in each iteration such that they are progressively less challenging. TPCL leverages sequential multi-task learning that established the principle that models are more transferable between closely related tasks and superior to learning all the tasks simultaneously [27, 33].

The main challenge here is the curriculum design. Numerous methods have been proposed in multi-task learning problems like Curriculum Learning (CL) [6] or dynamic task prioritisation [14]. Curriculum learning, originally proposed by Bengio et al. [6], is a learning strategy inspired by human learning that trains a model in a way that starts with simpler, easier examples and gradually increases the complexity of the data as the training process progresses, and the model's performance improves. While Dynamic Task Prioritisation or anti-curriculum learning investigated the importance of training with difficult tasks first. Very few works [19] explored CL in VQA. LBCL [19] demonstrated CL potential as part of a bigger training pipeline supported by additional mechanisms such as knowledge distillation and ensemble learning. A key distinction between our work and the previous CL works [5, 19] is that the atomic component of our curriculum is not the individual sample but the task (i.e., a group of semantically related samples).

Indeed, as shown repeatedly in the literature, the curriculum can make [22] or break [31] the model. The task-based CL scheme introduced can be very open-ended. Making it unclear how to assess task difficulty to control the learning progression. To tackle this, we opt for a self-taught difficulty metric that uses the model loss during training to estimate the difficulty of each sample. Unlike instance-based CL works [19], TPCL is task-oriented and can not directly utilise the sample loss. Consequently, we propose a novel difficulty measure. Specifically, each task score is represented by a distribution of its samples losses. Then, the difficulty is estimated are the divergence (vs stability) of the task distribution across training iterations. *Tasks with less divergence are more memorable (easier), while tasks with higher divergence are harder to learn* [44]. Based on our observations of the distributions shifts during the training, we base our divergence on Optimal Transport [17]; a mathematically principled framework that leverages the underlying geometry of distributions and can estimate the divergence even when the distributions do not exactly overlap. In summary, the contributions of this work are as follows:

- We introduce, for the first time, the idea of Task-based Curriculum Learning in the robust Visual Question-answering problem. Effectively, we reformulate the VQA problem as a multi-task problem based on the question types and utilise CL to boost the VQA model and enable OOD generalisation.

- We design and implement a novel training strategy called Task Progressive Curriculum Learning (TPCL) and integrates a novel distributional difficulty measure. Unlike instance-based CL techniques, our technique considers the difficulty of all samples within a task and achieves superior performance.

- Based on a comprehensive evaluation, we demonstrate that TCPL single-handedly realises out-of-distribution generalisation in VQA and achieves state-of-the-art on multiple datasets. Furthermore, the performance gains by TPCL are demonstrated to be consistent in in-distribution VQA and low data regimes.

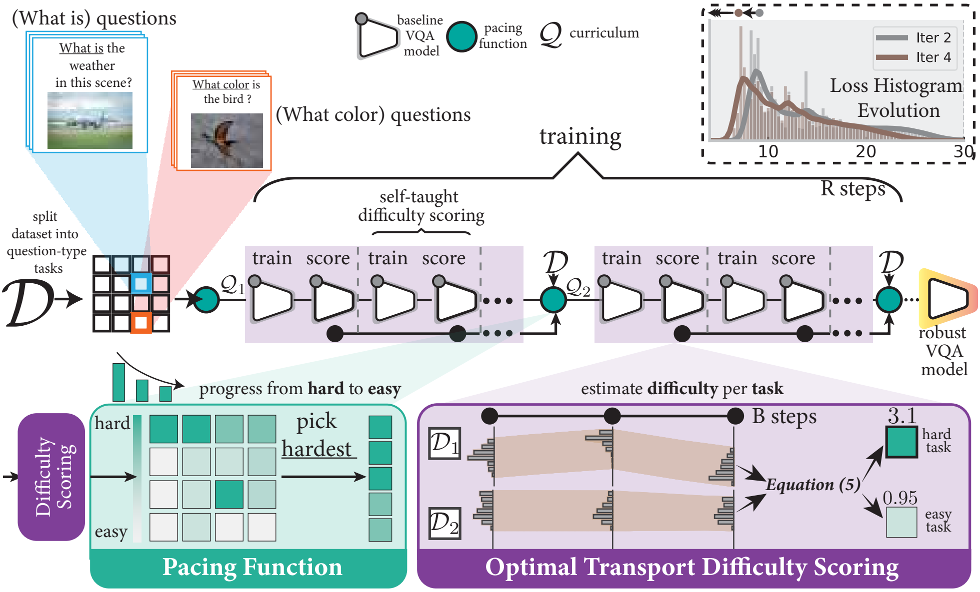

Figure 2: **Dynamic Curriculum Training.** TPCL training progresses from hard to easy to make the model focus on the challenging tasks first and enable out-of-distribution generalisation. The VQA model is exposed to a sequence of curricula $\mathcal{Q}_1, \cdots, \mathcal{Q}_R$ that are determined using a pacing function (●) and the (VQA) self-reported difficulty scores (●). TPCL innovates a task-specific difficulty measurer that 1) considers the distribution of all samples within the task (histogram) and 2) stabilises the scores by Optimal Transport-based consolidation over a B-length scores history window.

2 Related Work

VQA: is a challenging multi-modal task that has been actively explored in recent years, achieving performance approaching the human levels [3, 4, 34, 40] in In-Distribution (ID) datasets (VQA and VQA v2 [13]). However, they suffer from accuracy degradation in OOD due to the reliance on the biases presented in the dataset as explored by [1]. To evaluate the robustness of the VQA models [2] proposed the Visual Question Answering under Changing Prior (VQA-CP v2) and (VQA-CP v1) datasets as new settings for the original VQA v1 and VQA v2.

Many methods have been proposed to overcome the OOD problem in VQA models [11, 25, 32, 38]. The straightforward solution is balancing the dataset by acquiring new training samples [13], or synthetic data augmentation CSS [9]. Although these methods improved the performance, the dataset has statistical co-occurrences [2]. Besides, these methods require additional annotations that may have wrong answer assignments [38]. Ensemble learning approaches were used to tackle the OOD problem directly by training an auxiliary branch concurrently with the VQA model GenB [8, 11]. These methods introduce additional neural components for debiasing and potentially are backbone-sensitive [22, 38]. *TPCL outperforms the previous approaches while being entirely based on a novel training strategy without requiring additional data or debiasing neural components.*

Curriculum Learning: CL has been applied to different domains like computer vision and natural language processing [10, 20, 28, 42]. Curriculum learning is under-explored in VQA.

Pan *et al.* [25] combines casual inference, knowledge distillation and curriculum learning in a two-stage approach for debiased VQA. LBCL [19] utilised curriculum learning and knowledge distillation to mitigate OOD by employing a visually sensitive coefficient metric. Previous techniques integrate additional supporting debiasing mechanisms such as knowledge distillation. At the technical level, TPCL's task-based nature calls for a novel CL design (e.g. distributional difficulty), while the previous approaches are instance-based. Very recently, CurBench[45], showed the performance gains of CL on non-standard data (e.g. noisy) through systematic evaluation of 15 methods on data from various domains. Specifically, CL boosts the models' performance considerably in class-imbalanced and noisy data setups. *TPCL complements these findings by demonstrating that CL can enable out-of-distribution generalisation in VQA.*

3 Task Progressive Curriculum Learning

We propose the TPCL pipeline to enhance robustness in VQA. Given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ with N samples $\mathbf{x}_i = (\mathbf{q}_i, \mathbf{v}_i, \mathbf{a}_i, \tau_i)$, each question $\mathbf{q}_i \in \mathbb{R}^{d_q}$ relates to an image $\mathbf{v}_i \in \mathbb{R}^{d_v}$, with ground truth $\mathbf{a}_i \in [0,1]^{|\mathcal{A}|}$ and $\tau_i \in [T]$ denoting the question type. Though τ_i is readily available and derived from $\mathbf{q}_i$, it is often underutilised in VQA training. We follow the categorization in [2], where $T = 65$. Without modifying model architecture, we leverage τ_i in curriculum construction, excluding it from inference to retain compatibility. Our goal is to learn a model $f : \mathbb{R}^{d_q} \times \mathbb{R}^{d_v} \mapsto [0,1]^{|\mathcal{A}|}$ that predicts $\mathbf{a}_i$ from $(\mathbf{v}_i, \mathbf{q}_i)$, framed as a multi-class classification task [22].

Task Progressive Curriculum Learning. In our approach to build a robust VQA, we design a task-based curriculum that can be used to train a baseline backbone (e.g., SAN [40], UpDn [3], etc) and enable out-of-distribution generalisation. The task-based curriculum framework we adopt here is generic. Thus, it can be instantiated in multiple ways depending on the design choices of the main CL components discussed below. Figure 2 is pictorial summary of the proposed training strategy. Prior to applying the curriculum strategy, we decompose the dataset based on the question type. More formally, with slight abuse of notation, for a set of question types $\tau \in [T]$, we reorganise the dataset into a group of T VQA sub-tasks $\{\mathcal{D}_\tau\}_{\tau=1}^T$ where task $\mathcal{D}_\tau \subset \mathcal{D}$ is the data subset whose questions belong to type τ. We note that the tasks are not uniform in number of samples as some question types can have considerably more samples than others.

Our approach follows the general Curriculum Learning pipeline. Curriculum Learning can be abstracted into two integrated components: *difficulty measurer* and *pacing function*. The first determines the relative difficulty of the tasks. The latter, based on the feedback from the first, decides (selects) the group of tasks to be exposed to the model in each training iteration. Combined together, they define a sequence of training stages $\mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_R$ where $\mathcal{Q}_r \subseteq \mathcal{D}$ is a collection of tasks and the training stages are ordered by difficulty (e.g. $\mathcal{Q}_1 > \mathcal{Q}_2 > \cdots > \mathcal{Q}_R$). The two components, although discussed separately below, work in tandem. We explore two variants for each component, including a novel dynamic difficulty measurer.

3.1 Difficulty Measurer

A) Dynamic Difficulty. The idea here is to sort the tasks based on the baseline backbone performance (*dynamically*) in each iteration before passing the data to the pacing function.

Algorithm 1 *Dynamic TPCL*: Dynamic Task Progressive Curriculum Learning.

Input: $\mathcal{D} = \{\mathcal{D}_\tau\}_{\tau=1}^{T}$: training dataset; θ : baseline VQA backbone; p : pacing function; R: max training iterations; B: score consolidation iterations.
Output: θ_R: the target model.

```
Q_1 ← D  ▶ Warm-up on the whole dataset
for r = 1,...,R do
    for b = 1,...,B do
        θ_r ← train model on Q_r  ▶ Train
        Compute S_{r,b} using Equation (1)  ▶ Score computation
    end for
    Compute Φ̈_r using Equation (5)  ▶ Score consolidation
    D' ← sort(D, Φ̈_r)
    size ← p(r) using Equation (6)
    Q_r ← {D'_i}_{i=1}^{size}
    θ_{r+1} ← θ_r
end for
return θ_R
```

This self-taught difficulty was proven to be effective in various applications [15, 44]. The difficulty scores are initially estimated as the loss of the warm-up phase for the backbone. Then, the model is trained, and updated weights are used to re-calculate the scores for the next iteration.

Specifically, a VQA backbone f parameterised by θ_r at training iteration r calculates the samples scores as following:

$$\mathcal{S}_r = \Big\{ \ell(f(\mathbf{x}_i;\theta_r)) \Big\}_{\mathbf{x}_i \in \mathcal{D}} \tag{1}$$

where ℓ is the binary cross entropy loss. Note that the scores are calculated for all samples in $\mathcal{D}$ in each iteration r. Unlike previous works [19] that estimate difficulty for each sample, we need to assess the difficulty per-task. Since the loss in Eq.(1) is estimated for each sample, we need an aggregate metric that represents the whole task. One option here is averaging the sample losses in each task. However, we noticed that the mean can be misleading as some tasks coincide on means despite the big discrepancy in their loss ranges (check the experiments in Sec. 4). To tackle this issue, we propose a distributional score of losses that captures the difficulty of all samples belonging to the task. Thus, we create a distribution of scores for each question type. Then, we track the distributional divergence across iterations. Question types whose loss scores distributions change significantly across iterations are considered harder [44]. This is analogous to the way the instance-based CL methods [12, 44] track the loss fluctuations across iterations as a reliable difficulty scoring mechanism (i.e. better than instantaneous hardness). Unlike them, we track task loss distributions rather than individual samples.

Formally, we first map $\mathcal{S}_r$ into $[s_r^1, \cdots, s_r^T]$, where $s_r^\tau \in \mathcal{R}^M$ denotes the scores histogram for question type τ where M is the histogram bins (details in the supplementary). Then, we estimate the tasks scores as the distributional divergence between the scores of the last two iterations. Specifically, for the histograms s_r^τ and s_{r-1}^τ supported on μ and ν respectively, we calculate:

$$\mathrm{OT}(s_r^\tau, s_{r-1}^\tau) = \inf_{\gamma \in \Pi(s_r^\tau, s_{r-1}^\tau)} \mathbb{E}_{(\mu,\nu)\sim\gamma}[d(\mu,\nu)] \tag{2}$$

where OT denotes the Wasserstein Optimal Transport distance [17] , $\Pi(s_r^\tau, s_{r-1}^\tau)$ is the set of all joint distributions whose marginals are s_r^τ, s_{r-1}^τ and $d(\mu,\nu)$ is the ground cost defined as the distance between μ in the histogram s_r^τ and bin ν in the histogram s_{r-1}^τ. Intuitively, OT represents the minimum "cost" to move the probability mass of one task distribution to match the other. We use OT here as the histograms s^τ tend to shift horizontally towards zero as the training progresses (check visual examples in the appendix); a situation where OT is a good fit as a metric. Alternative metrics, such as the Kullback–Leibler (KL) divergence, result in undefined values for the same situation as the distributions do not exactly overlap. OT, on the other hand, is resilient to this issue as it takes the underlying geometry into account [17]. Accounting for d while computing the divergence makes OT aware of the distribution geometry. We set d to be the squared Euclidean distance. These benefits come with a negligible computational overhead during training. In our experiments, Equation (2) takes, on average, 0.9 milliseconds for $M = 100$ and 1.2 milliseconds for $M = 200$. Thus totalling about 50.4-78 milliseconds (0.9/1.2 $\times$ 65 tasks) per iteration.

DIH [44] observed that instantaneous "hardness" (i.e. difficulty score from the last iteration) in CL can be misleading. The hardness of the sample can change dramatically from one iteration to another. Inspired by this, we calculated a *consolidated difficulty score* $\ddot{\Phi}$. Specifically, in each training stage r, we repeat the training on the same curriculum for additional B consolidation iterations (instead of one).

$$\phi_b^\tau = \mathrm{OT}(s_{r,b}^\tau || s_{r,b-1}^\tau) \tag{3}$$

$$\Phi_{r,b} = [\phi_b^1, \cdots, \phi_b^T] \tag{4}$$

where $s_{r,b}^\tau$ denotes the task t score in the r-th iteration and b-th consolidation cycle. The final distributional difficulty is calculated as the weighted sum:

$$\ddot{\Phi}_r = \sum_{b=2}^{B} \alpha_b \Phi_{r,b} \tag{5}$$

where α is a coefficient controlling the contribution of past consolidation iterations, and B is the back window length. The α values can be chosen to balance between historical information (difficulty from earlier iterations) and the current model state (later iterations). In our implementation, we prioritise later iterations by giving them higher weights. By default, we set the values of B and α to 5 and $[0.1, 0.1, 0.3, 0.5]$; respectively. We note that we did not perform hyper-parameter optimisation. In the supplementary, we include ablations regarding this. Additionally, we follow [44] and conduct a warm up in $\mathrm{TPCL}_{\mathrm{Dyn}\uparrow}$. Specifically, we train the backbone for 5 iterations on the whole dataset $\mathcal{D}$. Algorithm 1 shows the full dynamic TPCL pipeline. The colours purple and teal in Figure. 2 denote the difficulty measure and pacing component, respectively.

B) Fixed Difficulty. An alternative option for designing the curriculum is fixing the tasks order offline (before the training) by estimating the difficulty based on heuristics, check the appendix for more details.

Methods		backbone	VQA-CP v2				VQA-CP v1				VQA v2
			Overall	Y/N	Num	Others	Overall	Y/N	Num	Others	Overall
UpDn [3]	CVPR'18	-	39.74	42.27	11.93	46.05	37.96	42.79	12.41	42.53	63.48
LXMERT [34]	EMNLP'19	-	48.66	47.49	22.24	56.52	52.82	54.08	25.05	62.72	73.06
LBCL[19]	TMM'21	UpDn	60.74	88.28	45.77	50.14	61.57	84.48	42.84	46.32	-
D-VQA [37]	NeurIPS'21	LXMERT	69.75	80.43	58.57	67.23	-	-	-	-	-
SIMPLEAUG [18]	EMNLP'21	LXMERT	62.24	69.72	53.63	60.69	-	-	-	-	74.98
GGD [16]	TPAMI'23	UpDn	59.37	88.23	38.11	49.82	-	-	-	-	62.15
DGG [38]	ACL'23	UpDn	61.14	88.77	49.33	49.9	-	-	-	-	65.54
GENB [11]	CVPR'23	UpDn	59.15	88.03	40.05	49.25	62.74	86.18	43.85	47.03	-
PWVQA [35]	TMM'24	UpDn	59.06	88.26	52.89	45.45	-	-	-	-	62.63
BILI [43]	KNOSYS'24	LXMERT	71.18	92.18	64.90	61.90	-	-	-	-	-
CVIV [26]	TMM'24	UpDn	60.08	88.85	40.77	50.30	-	-	-	-	61.93
FAN-VQA [7]	TCSVT'24	LXMERT	72.18	84.76	65.98	67.29	-	-	-	-	-
SCLSM [39]	CVIU'24	LXMER	70.27	82.35	58.97	67.03	-	-	-	-	-
PDGH [21]	AAAI'25	-	61.68	89.29	53.13	50.32	64.56	89.56	47.35	46.01	-
TPCL$_{\mathbf{Fix}\uparrow}$	ours	LXMERT	**75.83**	91.55	**68.49**	**69.61**	**76.78**	**90.74**	**72.22**	**64.72**	**78.42**
TPCL$_{\mathbf{Dyn}\uparrow}$	ours	LXMERT	**77.23**	**93.10**	**72.00**	**70.34**	**76.15**	**93.93**	**62.62**	**63.91**	**78.03**

Table 1: Comparisons with SOTA on two OOD VQA-CP v2 and VQA-CP v1 datasets and IID VQA v2 dataset.

3.2 Pacing Function

The pacing function determines the rate at which new training tasks are introduced to a model during learning. This function essentially manages the "curriculum" of data, allowing a model to start with harder tasks or samples and gradually move to less challenging ones as learning progresses. We use a standard step pacing function [36] that adds a fraction of the training data every d iterations as:

$$p_{\uparrow}(r) = \min\left(1, \lambda_0 + \frac{1-\lambda_0}{\lambda_{\text{grow}}}.r\right) \tag{6}$$

where λ_0, λ_{grow} and r denote the initial data rate, the data growth rate, and the current training epoch; respectively. The subscript $\uparrow$ denotes incremental pacing that gradually increases the size of the data presented to the model. Alternatively, one can adopt the decremental pacing by $p_{\downarrow}(r) = \max\left(0, 1 - \frac{1-\lambda_0}{\lambda_{\text{grow}}}.r\right)$. This stepwise uniformly spaced function is applied in the dynamic curriculum. In the fixed curriculum, we use a discrete pacing proportional to the questions in each task (i.e. [0.49, 0.94, 0.95, 1.0]).

4 Evaluation

We start by evaluating TPCL performance in out-of-distribution and in-distribution datasets. We report the performance compared to SOTA approaches. Then, we evaluate TPCL backbone sensitivity by testing on three standard VQA backbones. An ablation of the distributional difficulty is conducted. Finally, we show TPCL performance in a low data regime. Due to the page limit, we include a qualitative evaluation and additional ablations in the appendix.

VQA Evaluation in OOD: We compare the performance of TPCL on the VQA-CP v2 and VQA-CP v1 datasets against recent and state-of-the-art approaches (Table.1). We implemented TPCL on the most used baseline models: LXMERT [34], UpDn [3], and SAN [40]. However,

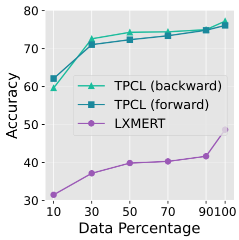

Figure 3: Low data performance.

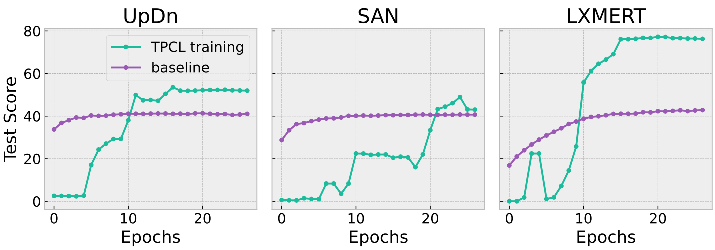

Figure 4: TPCL learning dynamics.

our approach is not restricted to these specific backbones and is adaptable to other architectures as well.

VQA Evaluation in ID: As revealed in a number of works [22, 32] in the literature, a pitfall of many robust VQA systems is that they tend to perform well in the out-of-distribution setting at the expense of in-distribution performance. To test this aspect, we evaluate TPCL on the VQAv2 dataset. As shown in Table 1, TPCL (LXMERT) outperforms the previous approaches and outperforms the second best approach, SIMPLEAUG (LXMERT) [18], by 3.44%. Additionally, $\text{TPCL}_{\text{Fix}\uparrow}$ outperforms $\text{TPCL}_{\text{Dyn}\uparrow}$ in this setup. This suggests that the dynamic difficulty measure is more suited for situations where the distribution of the answers is unknown (i.e. out of distribution).

Backbone Agnostic Approach: We showed that TPCL achieves superior results using LXMERT. As shown in Figure 5, we consistently achieve high gains compared to the baseline backbones using both the fixed and dynamic curriculum variants. Specifically, we improve the performance by a minimum of 7.11% by the fixed curriculum in SAN on VQA-CP v2. The improvement goes up to 28.57% in LXMERT on VQA-CP v2 using the dynamic curriculum. We again observe that dynamic TPCL is consistently better in out-of-distribution compared to the fixed TPCL.

TPCL Training Dynamics: Figure 4 illustrates the test performance of baseline models using the conventional training approach alongside the TPCL training strategy. All the baseline models began their training with higher evaluation scores compared to our TPCL approach. This discrepancy in training behaviour can be attributed to the TPCL method's strategy of initiating training with the most challenging tasks, unlike the baseline models, which seem to quickly memorise and overfit to the dataset. This becomes apparent in regions where the performance of the baseline models stagnates. TPCL, on the other hand, starts slow cause it trains mostly on the hard tasks. Once it masters hard tasks, it quickly picks up and surpasses the vanilla baseline by a margin. Additionally, TPCL training strategy is more rewarding with complex models (e.g. LXMERT), achieving significant gains in performance.

Distributional Difficulty Ablation: We ablate the effectiveness of distributional difficulty by considering a simple (non-distributional) approach.This is an alternative to the Optimal Transport-based distributional difficulty measurer that relaxes the distribution and consolidation requirements. Specifically, it uses the mean difficulty of the samples instead of the whole distribution. The sample difficulty is estimated from the last iteration instead of the B-length consolidation window.

Figure 6 summarises the findings. The results clearly show that utilising the loss distribution metric offers superior performance compared to a mean-based metric across all

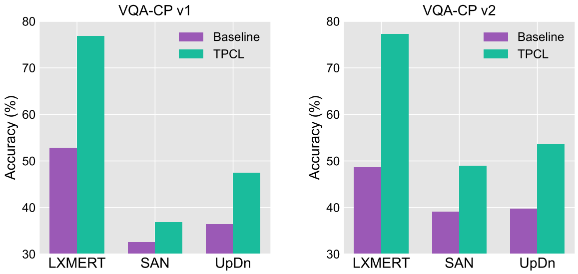

Figure 5: TPCL with different backbones on OOD datasets.

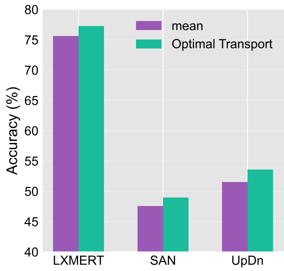

Figure 6: OT vs mean difficulty on VQA-CP v2.

baseline models. Specifically, OT improved the performance of the SAN model over the mean difficulty by approximately 1.37%. For the UpDn model, the mean difficulty achieved a performance of 51.56%, which was enhanced to 53.56% under OT, marking a 2% improvement. In the case of the LXMERT backbone, OT demonstrated a 1.6% improvement. Therefore, using distributional loss change, which leverages the model's performance history, is more effective than relying solely on the score metric.

TPCL in Low Data Regime: To demonstrate the effectiveness of our curriculum learning strategy in a limited regime, we trained the LXMERT backbone with varying percentages of the VQA-CP v2 dataset. We explored two different curriculum learning approaches: forward (training from easy to hard) and backward (training from hard to easy) in a dynamic manner. The results, as shown in Figure 3, reveal the following insights: 1) Using only 30% of the dataset, our LXMERT backbone achieved the state-of-the-art performance of 72.58%, 2) The backward curriculum learning approach outperforms the forward approach. Specifically, training the VQA model by first presenting harder question types and subsequently introducing easier samples enhances the model's generalisability more effectively than starting with the easier samples and then progressing to harder ones.

5 Conclusion

In this paper, we propose a simple and novel Curriculum Learning (CL) strategy for Robust VQA. TPCL breaks the main VQA problem into smaller, easier tasks based on the question type, and progressively trains the model on a carefully crafted sequence of tasks. We demonstrate the effectiveness of TPCL through comprehensive evaluations on standard datasets. Without requiring data augmentation or explicit debiasing mechanisms, our method achieves state-of-the-art on multiple datasets.

References

[1] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016.

[2] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4971–4980, 2018.

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 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*, pages 6077–6086, 2018.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[5] Narjes Askarian, Ehsan Abbasnejad, Ingrid Zukerman, Wray Buntine, and Gholamreza Haffari. Curriculum learning effectively improves low data vqa. In *Australasian Language Technology Association Workshop 2021*, pages 22–33. Association for Computational Linguistics (ACL), 2021.

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[7] Yandong Bi, Huajie Jiang, Yongli Hu, Yanfeng Sun, and Baocai Yin. Fair attention network for robust visual question answering. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[8] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in neural information processing systems*, 32, 2019.

[9] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809, 2020.

[10] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439, 2015.

[11] Jae Won Cho, Dong-Jin Kim, Hyeonggon Ryu, and In So Kweon. Generative bias for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11681–11690, 2023.

[12] Yan Dai, Beitao Chen, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Dmh-cl: Dynamic model hardness based curriculum learning for complex pose estimation. *IEEE Transactions on Multimedia*, 2023.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 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*, pages 6904–6913, 2017.

[14] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 270–287, 2018.

[15] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR, 2019.

[16] Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. General greedy de-bias learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (8):9789–9805, 2023.

[17] Abdelwahed Khamis, Russell Tsuchida, Mohamed Tarek, Vivien Rolland, and Lars Petersson. Scalable optimal transport methods in machine learning: A contemporary survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[18] Jihyung Kil, Cheng Zhang, Dong Xuan, and Wei-Lun Chao. Discovering the unknown knowns: Turning implicit knowledge in the dataset into explicit training examples for visual question answering. *arXiv preprint arXiv:2109.06122*, 2021.

[19] Mingrui Lao, Yanming Guo, Yu Liu, Wei Chen, Nan Pu, and Michael S Lew. From superficial to deep: Language bias driven curriculum learning for visual question answering. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3370–3379, 2021.

[20] Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. A competence-aware curriculum for visual concepts learning via question answering. In *European Conference on Computer Vision*, pages 141–157. Springer, 2020.

[21] Yishu Liu, Jiawei Zhu, Congcong Wen, Guangming Lu, Hui Lin, and Bingzhi Chen. Towards robust visual question answering via prompt-driven geometric harmonization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5721–5729, 2025.

[22] Jie Ma, Pinghui Wang, Dechen Kong, Zewei Wang, Jun Liu, Hongbin Pei, and Junzhou Zhao. Robust visual question answering: Datasets, methods, and future challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[23] Sara Moradlou and Jonathan Ginzburg. Young children's answers to questions. In *Workshop on the Role of Pragmatic Factors on Child Language Processing*, 2016.

[24] Sara Moradlou, Xiaobei Zheng, Ye Tian, and Jonathan Ginzburg. Wh-questions are understood before polars. *Proceedings of Architectures and Mechanisms for Language Processing (AMLaP)*, 2018.

[25] Yonghua Pan, Zechao Li, Liyan Zhang, and Jinhui Tang. Causal inference with knowledge distilling and curriculum learning for unbiased vqa. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(3):1–23, 2022.

[26] Yonghua Pan, Jing Liu, Lu Jin, and Zechao Li. Unbiased visual question answering by leveraging instrumental variable. *IEEE Transactions on Multimedia*, 26:6648–6662, 2024.

[27] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. Curriculum learning of multiple tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5492–5500, 2015.

[28] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019.

[29] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. *Advances in Neural Information Processing Systems*, 31, 2018.

[30] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2591–2600, 2019.

[31] Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A Erdogdu, and Ross J Anderson. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems*, 34:18021–18032, 2021.

[32] Qingyi Si, Yuanxin Liu, Fandong Meng, Zheng Lin, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. Towards robust visual question answering: Making the most of biased samples via contrastive learning. *arXiv preprint arXiv:2210.04563*, 2022.

[33] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, pages 9120–9132. PMLR, 2020.

[34] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[35] Ali Vosoughi, Shijian Deng, Songyang Zhang, Yapeng Tian, Chenliang Xu, and Jiebo Luo. Cross modality bias in visual question answering: A causal view with possible worlds vqa. *IEEE Transactions on Multimedia*, 2024.

[36] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[37] Zhiquan Wen, Guanghui Xu, Mingkui Tan, Qingyao Wu, and Qi Wu. Debiased visual question answering from feature and sample perspectives. *Advances in Neural Information Processing Systems*, 34:3784–3796, 2021.

[38] Zhiquan Wen, Yaowei Wang, Mingkui Tan, Qingyao Wu, and Qi Wu. Digging out discrimination information from generated samples for robust visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6910–6928, 2023.

[39] Shuwen Yang, Luwei Xiao, Xingjiao Wu, Junjie Xu, Linlin Wang, and Liang He. Simple contrastive learning in a self-supervised manner for robust visual question answering. *Computer Vision and Image Understanding*, 241:103976, 2024.

[40] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[41] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5014–5022, 2016.

[42] Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*, 2019.

[43] Liang Zhao, Kefeng Li, Jiangtao Qi, Yanhan Sun, and Zhenfang Zhu. Robust visual question answering utilizing bias instances and label imbalance. *Knowledge-Based Systems*, page 112629, 2024.

[44] Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613, 2020.

[45] Yuwei Zhou, Zirui Pan, Xin Wang, Hong Chen, Haoyang Li, Yanwen Huang, Zhixiao Xiong, Fangzhou Xiong, Peiyang Xu, Wenwu Zhu, et al. Curbench: Curriculum learning benchmark. In *Forty-first International Conference on Machine Learning*.