DYNAMIC PRETRAINING OF VISION-LANGUAGE MODELS

AJ Piergiovanni, Weicheng Kuo, Wei Li, Anelia Angelova
Google Research

ABSTRACT

Vision-Language pretraining aims to learn universal cross-modal representations and to create models with broad capabilities. While most models have taken the direction of scaling training to increasingly large models and datasets, in this paper, we propose a dynamic pretraining resampling approach which utilizes a variety of pretraining tasks, and which results in more sample-efficient models. We show that a set of diverse self- and weakly-supervised pretraining tasks dynamically sampled according to task difficulty provides strong performance. We show that a single 330M param pretrained model using only smaller and publicly accessible image-language datasets, achieves competitive or SOTA performance on three diverse groups of tasks: visual question answering, text-based image localization by referring expressions, and video question answering.

1 INTRODUCTION

We propose to improve the pretraining of vision-language models by leveraging existing datasets and tasks in more effective ways, showing improvements across a variety of vision-language tasks. Recent work suggests that incorporating more than one task or dataset can improve the downstream performance (Zhang et al., 2021; Yuan et al., 2022). Yet an unsolved question is how to leverage different pretraining mechanisms and mix all these tasks together. With the increasing number of tasks and datasets, the number of pretraining objectives can grow quickly. Standard hyperparameter methods, such as grid search, are computationally expensive when dealing with big data and big models. Naive solutions, such as uniform sampling, or sampling based on dataset size, we show, are not optimal. Instead, we address this by proposing *dynamic difficulty sampling*, which dynamically updates the sampling weights based on a task's current difficulty, which we find to highly correlate to downstream tasks performance and to outperform alternatives even with smaller task mixtures.

The key contributions of this paper are:

- Vision-language multi-task pretraining with a more label-efficient use of data. Our approach uses smaller but diverse datasets as opposed to large and inaccessible ones (Jia et al., 2021).
- A novel dynamic difficulty sampling method, based on curriculum learning, which dynamically updates the mix. This reduces the training steps to achieve similar performance.
- Results on a wide variety of tasks, showing strong performances on VQA, referring expressions comprehension and VideoQA.

2 RELATED WORK

Inspired by the success in language learning, image-language methods have offered powerful pretraining models (Tan & Bansal, 2019; Lu et al., 2019; Zhang et al., 2021; Kim et al., 2021; Li et al., 2022). Most common pretraining tasks directly inherit the tasks or losses from the language-learning counterparts, such as captioning or Masked Language Modeling (Lu et al., 2019; Chen et al., 2020).

In the context of text understanding, a mixture of pretraining tasks is more commonly used (Raffel et al., 2020), in some cases including supervised ones (Aribandi et al., 2021). Pretraining methods for image-language learning have mostly leveraged very large datasets or mixed a number of datasets (Yuan et al., 2022; Zhang et al., 2021; Chen et al., 2020; Cho et al., 2021). Furthermore, many use a fixed (typically) uniform sampling of the data or tasks or use 'round-robin' schedules (Lu et al., 2020), or alternatively sample according to data sizes.

Our approach is related to curriculum learning Bengio et al. (2009); Xu et al. (2020); Wang et al. (2019a); Li et al. (2020a), and is focused on trying to create better training batches. Prior works relied on measuring downstream validation loss Gottumukkala et al. (2020) or based on specific

heuristics for single modality tasks Xu et al. (2018). Others adjust the batches based on other metrics, e.g., Li et al. (2020b) measures difficulty based on item response theory, which requires variational Bayesian inference to determine the curriculum schedule. Liu et al. (2022); Xu et al. (2020) uses multiple metrics, computed over the entire training set, then re-ranks individual samples to create subsequent batches. Some use curriculum learning to adjust for class imbalance Wang et al. (2019b). Most of those works are focused on using samples from a single dataset and tasks, whereas here, we are focused on mixing different tasks which have different difficulties. Further, our approach is more generic and simple than the previous ones, applicable to a wide variety of tasks and modalities and requires no evaluation of the downstream tasks.

3 Pretraining Mixture Learning

The tasks are diverse in two dimensions: 1) the amount of supervision and accompanying noise, 2) the task the model is trying to solve, focusing on tasks that exercise different parts of the model.

Cross-modal weakly-supervised and self-supervised tasks. We use Image-Text Matching (ITM) and variations of Masked Language Modeling (MLM). Specifically, we use MLM (20% of random text masked out), Captioning (Cap) (all text masked out) and Caption Completion (CMP) (second half of text masked out).

Supervised tasks: Object-aware tasks. We further construct additional tasks, using labeled object-specific information (e..g, image-level or localization class labels). The tasks are designed to teach the model different aspects of image-text and object data, especially in the case of non-exhaustively annotated objects. The tasks are:

1. Input: 'List all objects' Output: '[obj1], [obj2], ...'
2. Input: 'Does [object] exist?' Output: Yes/No
3. Input: 'Does [obj1], [obj2] and/or [obj3] exist?' Output: Yes/No
4. Input: 'Which of [obj1], [obj2] and [obj3] exist?' Output: [obj1], [obj2].

These tasks exercise the model in different ways. For example, listing all the objects requires the model's decoder to generate the names for all the objects, however, since the image has some objects which are not annotated, this task can be quite difficult for the model.

We note that these tasks use the labeled data differently than traditional classification tasks. Experimentally, we find this to be better than training the vision model with classification data (see supp.).

3.1 Dynamic Difficulty Sampling

Consider a set of pretraining tasks $t \in T$. For each of these tasks, we have an associated loss, $\mathcal{L}_t$, which is computed for a batch. The final loss for a mixture of tasks, $M = t_1, t_2, \ldots t_n$ is computed as $\mathcal{L}_M = \sum_{t \in M} \mathcal{L}_t$. In this work, we equally weight the loss terms, instead focusing on weighting the number of samples in a batch.

Inspired by curriculum learning (Bengio et al., 2009), we argue that when a task has a higher loss, it is currently harder for the model to solve. Thus, the core of our approach is to sample tasks with higher losses more. Further, in the context of visual-language learning, we observed that for two mixtures of tasks, M_1 and M_2, if $\mathcal{L}_{M_1} > \mathcal{L}_{M_2}$, then the accuracy of the model finetuned on a VQA task from M_1 is greater than M_2. In Figure 1, we show a plot of the cross-entropy loss and downstream performance for a variety of tasks, confirming this observation. We believe this is because when the pretraining loss is higher (after sufficient training), the tasks are harder or more diverse and the model is able to learn more from those tasks. For example, a yes/no task is easy for the model to learn and does not provide as many learning signals as an open-vocab captioning task.

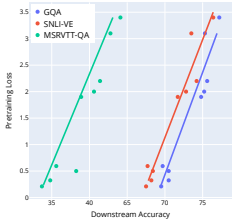

Figure 1: Difficulty (pretraining loss) vs downstream task accuracy for GQA, SNLI-VE and MSRVTT-QA datasets, illustrating the correlation between the pretraining loss and downstream accuracy.

Table 1: Performance on image VQA datasets. Our model gives strong results, even with smaller pre-training datasets (e.g. BLIP-L uses 129M, METER uses CLIP, pretrained on 400M image-text pairs, etc). Our model is only 0.17 points away on VQA2.0 from SimVLM-huge (1.5B params with 1.8B dataset) although we use open-vocabulary. Recent unpublished 80B Flamingo (Alayrac et al., 2022) outperforms us only by 2 points. We use very few GFLOPs (GF) (GF measured or obtained from authors). Note that our 14M data results use the same data as ALBEF.

	GF	Data	Params	GQA	SNLI-VE	VQA2.0
SimVLM-huge* (Wang et al., 2021)	900	1.8B	1.5B	-	**86.21**	**80.03**
UNITER (Chen et al., 2020)	-	-	-	-	79.39	72.5
12-in-1 (Lu et al., 2020)	-	-	-	60.5	-	71.3
VinVL (Zhang et al., 2021)	-	-	-	65.05	-	76.52
CFR (Nguyen et al., 2022)	-	-	-	73.6	-	69.8
FLAVA (Singh et al., 2022)	70	70M	240M	-	78.9	72.5
METER-CLIP-ViT-B (Dou et al., 2022)	130	400M	330M	-	80.86	77.68
ALBEF (Li et al., 2021)	160	4M	418M	-	80.30	74.54
ALBEF (Li et al., 2021)	160	14M	418M	-	80.80	75.84
BLIP (Li et al., 2022)	122	14M	475M	-	-	77.54
BLIP-L (Li et al., 2022)	250	129M	475M	-	-	78.25
Ours (14M data)	54	14M	330M	**80.7**	**81.0**	**78.43**
Ours (39M data)	54	39M	330M	**81.3**	**82.5**	**79.86**

By sampling in this way, we avoid training and evaluating exponential number of combinations of pretraining tasks.

Based on the above observations, we propose the **dynamic difficulty sampling** as follows. For each task and dataset in the training mixture we sample the next batch by re-weighting the sample size of each task as: $S_t = \frac{\mathcal{L}_t}{L}$. Here S_t is the percent of the batch used for task t. We enforce a minimum of 4 samples per-batch for each task, ensuring that we can always compute the difficulty of each task for the subsequent iterations. For stability, we accumulate the losses over K steps (here $K = 100$). Algorithm 1 summarizes the key steps of the dynamic difficulty sampling. We note that this method has very little computation overhead.

The pretraining tasks are selected so that they are likely helpful to the visual-language understanding at hand (by design), so we do not have tasks that are unnecessarily difficult, irrelevant or harmful and apply the same training objective, e.g., per-token cross-entropy loss, summed over all tasks. We note that while here we use the same loss for all tasks, the approach does not require this, any set of tasks and losses can be used. Though different losses (e.g., L_2) may have different ranges, requiring normalization before resampling.

Algorithm 1 Dynamic resampling pretraining algorithm

M - model
D - mixture of datasets, S_D - current sample/batch
$\mathcal{L}_t$ - loss for task t
for s = 1 to Iterations **do**
  for k = 1 to K (accumulation steps) **do**
    $M, \mathcal{L}_{k,t} \leftarrow train(M, S_D)$ ▷ Train with current mix S_D, accumulate per-task loss.
  end for
  $\mathcal{L}_t \leftarrow \sum_k \mathcal{L}_{k,t}$ ▷ Measure performance (losses) of pretraining tasks for last K steps.
  $S_D \leftarrow D \sim (D, \mathcal{L}_t)$ ▷ Resample the data according to task difficulty.
  ▷ The next iteration of training (i.e., the next K steps) will be trained with the new mixture weights.
end for

Table 2: Results on VideoQA (video+text) datasets MSRVTT-QA (Xu et al., 2016), MSVD-QA (Xu et al., 2017), IVQA (Yang et al., 2021). Accuracy (%). We note that VQA-T (Yang et al., 2021) uses much larger video pretraining dataset 100M, whereas we use 39M image-only dataset. *ClipBERT also uses image-only pretraining for the VideoQA tasks, similar to ours.

Model	MSRVTT-QA	MSVD-QA	IVQA
HCRN (Le et al., 2020)	36.1	35.6	-
ClipBERT* (Lei et al., 2021) 8x2, image-only pretraining	37.4	-	-
VQA-T (Yang et al., 2021), pretr. HowToVQA69M	41.5	46.3	35.4
MERLOT (Zellers et al., 2021), pretr. YT180M	43.1	-	-
AllInOne (Wang et al., 2022), pretr. YT180M,HowTo100M	44.3	**47.9**	-
Ours, image-only pretraining	**45.1**	47.1	**35.8**

Table 3: Experiments on localizing objects in the image by referring expressions. Without using any pretraining localization labels or localization components (e.g., box proposals or regression heads), we provide competitive performance to existing models. We compare to the best and most contemporary ones in the interest of space. With the exception of the top row and ours, all others use stronger backbone ResNet-101.*MDETR (Kamath et al., 2021) uses additional label supervision during training.

	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
TransVG (Deng et al., 2021) (R50)	80.32	82.67	78.12	63.50	68.15	55.63	50.6	-
TransVG (Deng et al., 2021) (R101)	81.02	82.72	78.35	64.82	70.70	56.94	67.02	-
UNITER (Large) (R101)	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA (Large) (R101)	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
MDETR* (R101)	86.75	89.58	81.41	**79.52**	84.09	70.62	**81.64**	80.89
Ours (R50)	**88.9**	**90.7**	**82.8**	78.4	**84.3**	**72.6**	80.9	**81.5**

Table 4: Ablating different sampling mechanisms for the tasks, including the popular round robin (Lu et al., 2020). Mixtures of all tasks are used in each experiment. Experiments are conducted with only 2/5 of the steps.

	GQA	SNLI-VE
Easiest Task	71.3	67.5
Most Difficult	74.2	76.4
Uniform Sampling	75.6	77.4
Sampling by dataset size	75.7	77.9
Loss Reweighting	75.9	77.5
DSG Round Robin, 12-in-1 (Lu et al., 2020)	76.4	77.8
Dynamic Sampling (ours)	**77.4** (+1.0)	**78.2** (+0.3)

4 EXPERIMENTS

We pretrain a single model, and then finetune it on three diverse sets of tasks (image-language, object localization and video-language), some of which are outside the initial domain of pretraining, without additional heads. Specifically we report results on VQA datasets **GQA** (Hudson & Manning, 2019), **VQA2.0** (Agrawal et al., 2015)), visual entailment (**SNLI-VE** (Suhr et al., 2017)) (Table 1), on VideoQA datasets **MSRVTT-QA** (Xu et al., 2016) and **MSVD-QA** (Xu et al., 2017) and **IVQA** (Yang et al., 2021) (Table 2) and on object localization tasks with refering expressions **RefCOCO** (Yu et al., 2016), **RefCOCO+** (Yu et al., 2016), **RefCOCOg** (Mao et al., 2016) (Table 3). For each of the datasets we follow the evaluation metrics and protocols established in prior work. Please see the supp. materials for training and evaluation details.

4.1 DATA SAMPLING METHODS AND PRETRAINING MIXTURES ABLATIONS

We compare different methods of sampling the data. We compare to common approaches: uniform sampling, e.g., the batch is composed of $BS/|T|$ samples from each task for $|T|$ tasks, and dataset-size based sampling. We also compare sampling proportional to difficulty and sampling only the most difficult/easiest task. We also compare to reweighting the loss, rather than changing the batch; and compare to the DSG algorithm (Lu et al., 2020). The results are shown in Table 4.

5 CONCLUSIONS

We present a visual-language model with a novel pretraining approach based on dynamic difficulty sampling utilizing self-, weak and supervised sources of pretraining. We demonstrate effectiveness on three classes of tasks including vision-language tasks, video ones and object localization ones. Our approach achieves competitive results despite using order of magnitude less data and smaller models.

REFERENCES

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Vamsi Aribandi, Yi Tay, Tal Schuster, Sanket Vaibhav Mehta Jinfeng Rao, Huaixiu Steven Zheng, and et al. Ext5: Towards extreme multi-task scaling for transfer learning. In *arXiv:2111.10952, 2021*, 2021.

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *Arxiv 2102.02779*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. *ICCV*, 2021.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18166–18176, 2022.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*, 2020.

Ananth Gottumukkala, Dheeru Dua, Sameer Singh, and Matt Gardner. Dynamic sampling strategies for multi-task reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 920–924, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over realworld images. In *CVPR*, 2019.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. In *https://arxiv.org/abs/2104.12763*, 2021.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL https://arxiv.org/abs/1602.07332.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *ICCV*, 128(7):1956–1981, 2020.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, 2020.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. A competence-aware curriculum for visual concepts learning via question answering. In *ECCV*, 2020a.

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*, pp. 141–157. Springer, 2020b.

Fenglin Liu, Shen Ge, and Xian Wu. Competence-based multimodal curriculum learning for medical report generation. *arXiv preprint arXiv:2206.14579*, 2022.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *CVPR*, 2019.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

Binh X. Nguyen, Tuong Do, Huy Tran, Erman Tjiputra, Quang D, and Anh Nguyen Tran. Coarse-to-fine reasoning for visual question answering. In *Multimodal Learning and Applications (MULA) Workshop, CVPR*, 2022.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*, 2020.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami abd Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *arxiv.org/pdf/2112.04482.pdf*, 2022.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *ICCV*, 2019a.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5017–5026, 2019b.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *https://arxiv.org/pdf/2108.10904.pdf*, 2021.

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. Dynamic curriculum learning for low-resource neural machine translation. *arXiv preprint arXiv:2011.14608*, 2020.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. Multi-task learning with sample re-weighting for machine reading comprehension. *arXiv preprint arXiv:1809.06963*, 2018.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. In *arxiv.org/pdf/2110.02095.pdf*, 2022.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021.

A VISUALIZATIONS

Figure 2 shows sampling weights over time. We can see the weights change over time, reflecting how hard the task is for the model at the current iteration. Some tasks that are easy consistently decrease, e.g., ITM. ITM is likely an easy task for two reasons: (1) The outputs are only 'yes' or 'no' and (2) since ITM is done by randomly picking other captions for the no case, often it is easy

Figure 2: The sampling weights for the 24 tasks over the 500k steps of training. The y-axis is the percent of the batch of each task. We can see that the tasks dynamically change in importance within a batch, for example, 'easy' tasks, such as ITM, get low weights quickly, while more challenging tasks, such as captioning increase in importance over time. We can also see that some tasks change in different ways over time, for example, 'List Objects' for ImageNet21k has a small decrease for the first 300k steps, then starts to increase.

to solve the task. We also see that other tasks increase, e.g., MLM. Others, such as listing objects, decrease then increase. This suggests that the model started to find those tasks a bit more difficult after lowering the sample weights. This figure helps show the effects of dynamic sampling, and shows the benefit of using it vs. manual tuning of the sampling weights. It also shows the ability of this method dynamically adjust the weights over time, which existing approaches do not do.

In Figure 1, we plot the loss of a training mixture M after 200k steps of pretraining. This loss is computed on the training split of the pretraining dataset mixtures. We then finetune each model on GQA, SNLI-VE and MSRVTT-QA and plot the resulting accuracy on their validation splits.

This observation may seem counter-intuitive. However, we argue here that because some of these tasks are inherently easier, e.g., 'yes' or 'no' outputs, they are easy for the model to learn, and their loss drops to nearly 0 quickly. Thus by having a mixture with a higher loss, here, indicates a more useful data mixture. Because all these models are trained for the same number of steps, with the same learning rate and other settings, they are directly comparable. This observation may not hold for other settings, e.g., learning rate.

B Implementation details

Model Details. We use a simple model (330M params), as our focus is on pretraining. It is a standard encoder-decoder – ResNet-50 (He et al., 2016) to extract images features, T5 (Raffel et al., 2020) encoder for text, and a T5 decoder to generate the answer. We concatenate the image and text features before decoding. Our model is trained from scratch, using public image+text datasets; the same pre-trained model is used for fine-tuning all tasks in the paper, including video and localization ones.

The model consists of a ResNet-50 for the image encoder. We take the feature map from conv_4, which has shape of 14×14 as the visual feature. We use an image size of 224x224 as input. For the text, we use the T5-base T5 model with the standard 32,000 token vocabulary. After the encoder layers, we concatenate the 196 vision features with the text features (we use an input sequence length of 32). These are then used as the input to the decoder layers. Specifically, we use 12 encoder and decoder layers with 768-dimension representations, following the T5-base settings. All models are trained from scratch.

Datasets. We use only 39M images, as opposed to other works, using hundreds of millions, or 1-2 billion images (Singh et al., 2022), (Yuan et al., 2022), (Wang et al., 2021). We use the Conceptual Captions 3M and 12M datasets (Sharma et al., 2018; Changpinyo et al., 2021), abbreviated as CC3+12M, ImageNet21k (IN) (Deng et al., 2009), Visual Genome (VG) (Krishna et al., 2016), Open Images (OI) (Kuznetsova et al., 2020) and its text annotations provided by the Localized Narratives (LN) (Pont-Tuset et al., 2020) dataset (we only use the annotations for OI). Note that we removed any images in the downstream validation/tests sets from these training sets.

Tasks. To create a pretraining mix, we define two groups of tasks over the datasets. For Localized Narratives, CC12m and CC3m, we use the 4 cross-modal image-text tasks: Captioning (Cap), Caption completion (CMP), Image-Text Matching (ITM) and Masked Language Modeling (MLM). For OpenImages, ImageNet21k and Visual Genome we use the 4 object aware tasks (Section 3). Note that while some of these datasets have bounding box labels, we omit those, using only class names. This results in a total of $3 \cdot 4 + 3 \cdot 4 = 24$ tasks, by mixing together all these datasets and tasks. Note that this approach is also easily extendable to other tasks and datasets.

Training Details. The final model is trained for 500k steps with a batch size of 4096. The loss is a per-token cross entropy loss over the 32,000 tokens from the vocabulary. We set the learning rate to 1e-4, use a linear warmup for the first 10,000 steps, followed by a cosine decay. We use the Adam optimizer with weight decay set to 0.1. We use label smoothing set to 0.1. We clip the gradient norm to 1. Due to computational constraints, these are the only hyperparameters we tried. Note that for most ablations, we only pretrain for 200k steps due to computational constraints, while the other settings are the same.

For finetuning, on the image datasets, we use the same learning settings as above. The batch size is 256 for 200k steps. For the VideoQA datasets, we use a smaller learning rate, 1e-6, batch size of 64, and 20,000 steps. For the RefCOCO tasks, we use 6e-4, with a batch size of 128 for 200k steps.

Dataset Splits. We train on the standard training sets splits. For the pretraining, we remove any images used in the validation and test sets of the downstream VQA datasets. We evaluate on the validation splits of the VQA datasets and the test splits for IVQA, MSRVTT-QA and MSVD-QA.

Model costs. Our model pretraining is of much lower cost compared to SOTA methods such as SimVLM (Wang et al., 2021), which takes about 244,000 TPU hours (2,048 for 5 days) for training. Our model takes only 384 TPU hours (16 for 1 day, **635x** cheaper), still obtaining competitive performance. This is an order of magnitude cheaper than other methods. Our pre-trained model is also used for downstream video-based tasks and object localization tasks, thus sharing the costs. Both are challenging tasks which other models did not evaluate on.

Box-As-Text Representation Details. For the RefCOCO tasks, we represent the boxes as text pix2seq. We first represent the box as $[y_{max}, x_{max}, y_{min}, x_{min}]$ where each x and y are floats between 0 and 1, which are the coordinates of a box in the image, normalized with respect to the image size. Following Pix2Seq pix2seq, we convert the normalized bounding boxes to integers by quantizing them into 100 intervals. We then convert these integers into tokens by treating them as strings in the T5 text tokenizer. An end of sequence token is appended at the end of the box.

To convert a prediction into a box for evaluation, we simply reverse the process. We convert the output tokens into integers, map the integer to the float by inversing the quantiziation step, then construct the box by un-normalizing the coordinates. We then compute the metrics using the standard RefCOOC evaluation code.

Compute Usage. Our model used 16 TPUs for 1 day to complete the pretraining. When finetuning the model, we used 4 TPUs for 8 hours. Both image and video tasks use the same pretrained model so this cost is shared. The VideoQA fine-tuning took 4 TPUs for about 8 hours too as it uses fewer steps. Note that these are the average times it took to train the model, machine downtime also affected the walltime used to train these models.

Pretraining MLM Details. For the pretraining tasks, we mask out 25% of the tokens for MLM. For the Caption Completion task, we mask out the last 20-60% of the tokens, where the exact portion is taken at random per example.

Evaluation details. We follow the standard evaluation settings from prior work. Since our model generates open-ended responses, we compare the generated string to the ground truth string. This is a more challenging setting since our model can generate close answers, e.g., 'oak trees' where a ground truth answer of 'tree' would be considered incorrect. For the VideoQA datasets, in order to compare directly with all prior work, we mask out the answers which are outside the standard vocabulary.

Given the generated answer for a model, we use the standard evaluations per dataset. For SNLI-VE, GQA, VQA2.0, MSRVTT-QA, MSVD-QA we use accuracy. For IVQA, we use their accuracy metric of $min(\frac{\text{\# ground truth answers == a}}{2}, 1)$, since there are multiple annotations per question. For RefCOCO, we use the standard AP50 metric which is the Average Precision with IoU threshold of 0.5.

B.1 Video Question and Answering

We further evaluate the performance of our pretraining on VideoQA tasks, here on the three datasets. The same 330M-parameter model, which provides pretraining for the VQA tasks above, is used for videos. We apply the ResNet per-frame, then concatenate the frames together to form the visual feature. As seen, our pretraining yields strong models for video, outperforming SOTA on all three datasets tested. This is also despite using image-only pretraining, which is of fewer images than e.g. the video pretraining of VQA-T (Yang et al., 2021), which uses 100M or 69M video snippets, or of MERLOT (Zellers et al., 2021) which uses 180M videos.

B.2 Object localization with Referring Expressions Comprehension

We also evaluate on three datasets of the Referring expressions comprehension task (Yu et al., 2016; Mao et al., 2016), in which a text describing an object in the image is given and the model needs to output the bounding box for it. This is an object localization task, for which the pretrained model is not explicitly trained. To accomplish that, in our generative text API, we use the pix2seq (Chen et al., 2022) approach where box coordinates are tokenized and treated as text (please see supp. for details). Our results on this challenging task with boxes represented as text during training, also achieves

Table 5: Comparison between uniform sampling vs. the proposed difficulty sampling, for an increasing number of pretraining task mixtures. Each entry shows the performance for a 'uniform sampling/difficulty sampling' pair on GQA and SNLI-VE. The improvement of difficulty sampling is consistent across all datasets and task mixtures. We note that using difficulty sampling even for 8 tasks is often outperforming the uniform sampling of 24 tasks (the latter also involving more datasets in the mix). A 330M model and only 2/5 of the steps (i.e. 200k steps) are used in this experiment.

	GQA	SNLI-VE
4 tasks	75.4 / 76.1	76.1 / 76.7
8 tasks	75.8 / 76.5	77.5 / 77.7
12 tasks	75.9 / 76.9	77.7 / 77.9
16 tasks	75.7 / 77.3	77.6 / 78.1
24 tasks	75.6 / 77.4 (+1.8)	77.4 / 78.2 (+0.8)

Table 6: Experiments on RefCOCO datasets which require localization of objects referred by text. We observe that combining Object Aware tasks are much more beneficial than only image-text pretraining tasks which are commonly used in large image-language models. Mixing many tasks with dynamic difficulty sampling performs the best, competitive with SOTA models on RefCOCO. SOTA results for RefCOCO/RefCOCO+ are from (Kamath et al., 2021), for RefCOCOg from (Deng et al., 2021).

	RefCOCO	RefCOCO+	RefCOCOg
SOTA	86.75	79.52	67.02
No pretraining	69.5	55.4	43.5
Cross-Modal (4 tasks)	73.7	59.7	48.7
Object Aware (4 tasks)	75.9	62.4	50.6
Difficulty Sampling (Ours, all 24 tasks)	**88.9**	**78.4**	**80.9**

SOTA or better than SOTA results (Table 3), compared to UNITER (Chen et al., 2020), VILLA (Gan et al., 2020), MDETR (Kamath et al., 2021), etc. We note that the RefCoco* benchmarks have been evaluated by more than 30 approaches, we only list the best here.

C ADDITIONAL ABLATIONS

We here include some additional experiments.

Using the dynamic difficulty sampling, we then explore the effects of adding more tasks. The results are shown in Table 5. We note that across datasets, for both small and large number of tasks the difficulty sampling is beneficial, and with the proposed difficulty sampling, the performance improves as more tasks are added. Interestingly, difficulty sampling for only 8 tasks tends to outperform uniform sampling for 24 tasks. We can also see that, as more tasks are added with uniform sampling, the performance can drop.

Table 6 shows the effect of our approach on localization tasks. As seen, the object-aware tasks are better than cross-modal tasks, confirming their ability to learn objects. However, they are not sufficient, as the proposed pretraining mixture tasks and dynamic difficulty sampling performs best.

We compare pretraining the vision model for classification to the object-aware tasks. For these experiments, we use the OpenImages dataset and the localized narrative annotations of the images for the cross modal tasks. This lets us keep the image data fixed and explore the different effects of the tasks. Table 7 shows the results. Note that these are pretrained for smaller number of steps (200k steps). These experiments use uniform sampling, to study the effects of adding them independently from difficulty sampling. We find that the object aware tasks are better than classification pretraining for these image-language tasks.

We observe that when only using the OpenImages data, the performance at 200k steps is roughly the same as at 500k steps, compared to Table 9. This suggests that when only using this data, 200k

Table 7: Comparison of pretraining with OpenImages as classification or object-aware tasks. We find that for vision-language tasks, the object-aware tasks provide a better model than the classification task. We also see benefit from combinations of the object-aware and cross-modal tasks.

	GQA	SNLI-VE
Classification PT	71.3	70.5
Object-Aware Tasks	75.5	74.6
Cross-Modal Tasks	77.4	73.2
Classification then Cross-Modal	77.6	73.8
Object-Aware + Cross-Modal	76.2	79.9

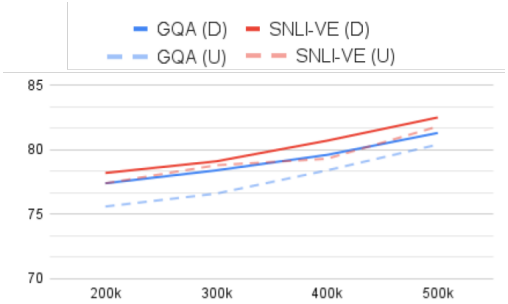

Figure 3: Performance of Uniform Sampling (U) and Difficulty Sampling (D) when pretrained for different numbers of steps then finetuned on VQA datasets. We observe that even to 500k steps, difficulty sampling is outperforming uniform sampling.

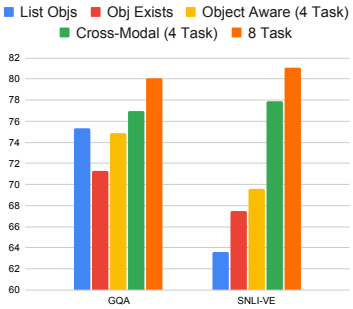

Figure 4: Comparing tasks. Individual tasks provide variable downstream performance, while mixtures of tasks with difficulty sampling are better.

steps is sufficient for pretraining, which is reasonable as 200k steps is roughly 91 epochs. The further benefits of additional training are observed when using the larger mixtures.

Table 5 Mixtures. In Table 5, we report results for 4, 8, 12, 16, and 24 task mixtures. For clarity, here we detail each of the mixtures.

- 4 Tasks: CC12M MLM, CC12M ITM, OI List Objs, OI Which Objs
- 8 Tasks: (above 4) + LN MLM, LN ITM, IN21k Which Objs, IN21k List Objs
- 12 Tasks: (above 8) + VG Which Objs, VG List Objs, CC3M MLM, CC3M ITM
- 16 Tasks: (above 12) + OI Exists, VG Exists, IN21k Exists, LN CMP
- 24 Tasks: (above 16) + (LN, CC3M, CC12M) Cap, (CC3M, CC12M) CMP, (VG, IN21k, OI) Multi-Exists

Here, OI is OpenImages, LN is Localized Narratives, CC is Conceptual Captions, VG is Visual Genome and IN21k is ImageNet 21k. ITM is Image Text Matching, CMP is Caption Completion, Cap is Captioning, and MLM is Masked Language Modeling. The 'List objects' task is Task 1 among the Object-Aware tasks, 'Exists' is task 2, Multi-Exists is task 3 and 'Which objs' is task 4.

Figure 3 further tracks the performance for a number of steps of uniform sampling vs. difficulty sampling. In Figure 4, we compare individual tasks to mixtures.

In Table 8, we find that using dynamic sampling also benefits other models, e.g., ViLT (Kim et al., 2021), showing it is not specific to the chosen architecture.

C.1 Small-sample mixtures ablations

In this section, we conduct experiments on even smaller mixture of pretraining tasks and a single dataset (here OpenImages, ∼9M samples). This illustrates the ability of the approach to leverage the data thanks to the mixture and dynamic difficulty sampling during pretraining very efficiently.

Table 8: Dynamic difficulty sampling applied to an alternative model architecture, ViLT (Kim et al., 2021) shows consistent improvements, as well.

	GQA	SNLI-VE
ViLT + Uniform Sampling	74.6	77.1
ViLT + Dynamic Diff. Sampling (Ours)	**77.2**	**77.9**

Table 9: Performance of VQA tasks on pretraining mixtures from OpenImages-only (with 9M images) and our model. The 4 tasks are the Cross-Modal tasks and the 8 are the Cross-Modal and Object Aware ones. Even in this setting our approach performs very competitive to SOTA.

	GQA	SNLI-VE
SOTA	65.5 (Zhang et al., 2021)	86.2 (Wang et al., 2021)
Uniform Sampling - 4 tasks	77.5	73.5
Difficulty Sampling - 4 tasks (ours)	76.2	79.2
Uniform Sampling - 8 tasks	76.4	80.0
Difficulty Sampling - 8 tasks (ours)	**78.7**	**81.5**

We train our model and consider the following tasks only: 4 cross-modal tasks (using the text annotations for OI) and 4 Object Aware tasks on OI. In the results, we see very competitive performance when dynamic difficulty sampling is used, even in this smaller model and smaller data scenario (Table 9). In summary we see that the pretraining mixture can bring in a lot of benefits and competitive results for even small-model, small-data scenarios.