GIT: A Generative Image-to-text Transformer for Vision and Language

Jianfeng Wang *jianfw@microsoft.com*
Zhengyuan Yang *zhengyang@microsoft.com*
Xiaowei Hu *xiaowei.hu@microsoft.com*
Linjie Li *lindsey.li@microsoft.com*
Kevin Lin *keli@microsoft.com*
Zhe Gan *zhe.gan@microsoft.com*
Zicheng Liu *zliu@microsoft.com*
Ce Liu *ce.liu@microsoft.com*
Lijuan Wang *lijuanw@microsoft.com*
Microsoft Cloud and AI

Abstract

In this paper, we design and train a **G**enerative **I**mage-to-text **T**ransformer, GIT, to unify vision-language tasks such as image/video captioning and question answering. While generative models provide a consistent network architecture between pre-training and fine-tuning, existing work typically contains complex structures (uni/multi-modal encoder/decoder) and depends on external modules such as object detectors/taggers and optical character recognition (OCR). In GIT, we simplify the architecture as one image encoder and one text decoder under a single language modeling task. We also scale up the pre-training data and the model size to boost the model performance. Without bells and whistles, our GIT establishes new state of the arts on numerous challenging benchmarks with a large margin. For instance, our model surpasses the human performance for the first time on TextCaps (138.2 vs. 125.5 in CIDEr). Furthermore, we present a new scheme of generation-based image classification and scene text recognition, achieving decent performance on standard benchmarks.

1 Introduction

Table 1: Comparison with prior SOTA on image/video captioning and question answering (QA) tasks. *: evaluated on the public server. CIDEr scores are reported for Captioning tasks. Prior SOTA: COCO(Zhang et al., 2021a), nocaps (Yu et al., 2022), VizWiz-Caption (Gong et al., 2021), TextCaps (Yang et al., 2021c),ST-VQA (Biten et al., 2022),VizWiz-VQA (Alayrac et al., 2022),OCR-VQA (Biten et al., 2022),MSVD (Lin et al., 2021a),MSRVTT (Seo et al., 2022),VATEX (Tang et al., 2021),TVC (Tang et al., 2021),MSVD-QA (Wang et al., 2022a),TGIF-Frame (Zellers et al., 2021),Text Recog. (Lyu et al., 2022). Details of GIT2 are presented in supplementary materials.

	Image captioning				Image QA			Video captioning				Video QA		Text Rec.
	COCO*	nocaps*	VizWiz*	TextCaps*	ST-VQA*	VizWiz*	OCR-VQA	MSVD	MSRVTT	VATEX*	TVC*	MSVD-QA	TGIF-Frame	Avg on 6
Prior SOTA[1]	138.7	120.6	94.1	109.7	69.6	65.4	67.9	120.6	60	86.5	64.5	48.3	69.5	93.8
GIT (ours)	148.8	123.4	114.4	138.2	69.6	67.5	68.1	180.2	73.9	93.8	61.2	56.8	72.8	92.9
Δ	+10.1	+2.8	+20.3	+28.5	+0.0	+2.1	+0.2	+59.6	+13.9	+7.3	-3.3	+8.5	+3.3	-0.9
GIT2 (ours)	149.8	124.8	120.8	145.0	75.8	70.1	70.3	185.4	75.9	96.6	65.0	58.2	74.9	94.5
Δ	+11.1	+ 4.2	+26.7	+35.3	+6.2	+4.7	+2.4	+64.8	+15.9	+10.1	+0.5	+9.9	+5.4	+0.7

[1]Prior SOTA: among all the numbers reported in publications before 8/2022, as far as we know.

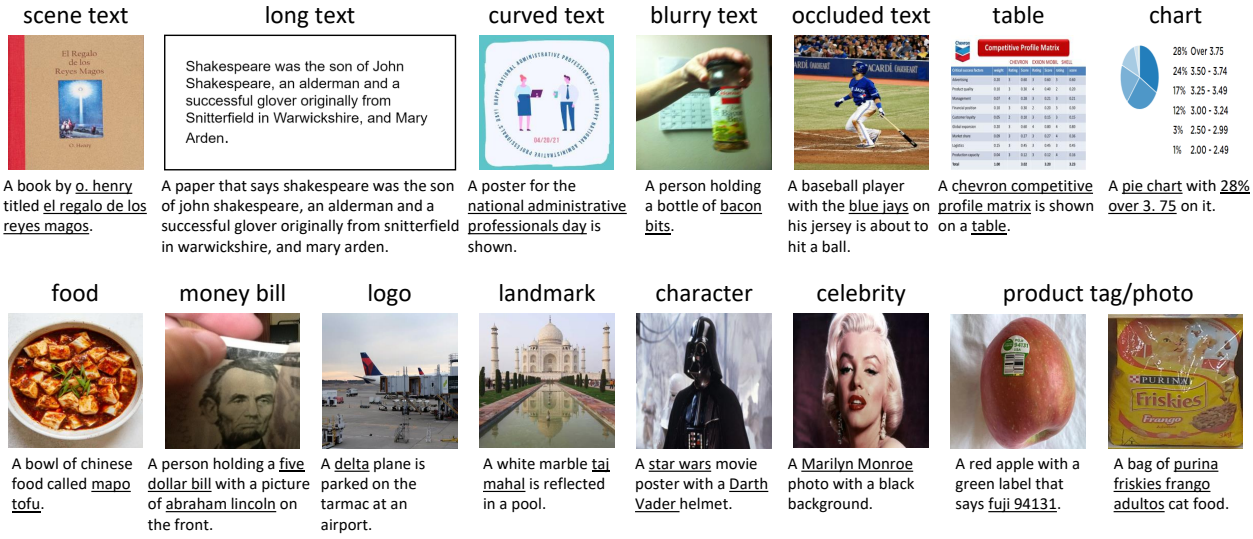


Figure 1: Example captions generated by GIT. The model demonstrates strong capability of recognizing scene text, tables/charts, food, banknote, logos, landmarks, characters, products, *etc.*

Tremendous advances have been made in recent years on vision-language (VL) pre-training, especially based on the large-scale data of image-text pairs, *e.g.*, CLIP (Radford et al., 2021), Florence (Yuan et al., 2021), and SimVLM (Wang et al., 2021b). The learned representation greatly boosts the performance on various downstream tasks, such as image captioning (Lin et al., 2014), visual question answering (VQA) (Goyal et al., 2017), and image-text retrieval.

During pre-training, Masked Language Modeling (MLM) and Image-Text Matching (ITM) tasks have been widely used (Wang et al., 2020; Fang et al., 2021c; Li et al., 2020b; Zhang et al., 2021a; Chen et al., 2020b; Dou et al., 2021; Wang et al., 2021a; Kim et al., 2021). However, these losses are different from the downstream tasks, and task-specific adaptation has to be made. For example, ITM is removed for image captioning (Wang et al., 2021a; Li et al., 2020b), and an extra randomly initialized multi-layer perceptron is added for VQA (Wang et al., 2021b; Li et al., 2020b). To reduce this discrepancy, recent approaches (Cho et al., 2021; Wang et al., 2021b; Yang et al., 2021b; Wang et al., 2022b) have attempted to design unified generative models for pre-training, as most VL tasks can be cast as generation problems. These approaches typically leverage a multi-modal encoder and a text decoder with careful design on the text input and the text target. To further push the frontier of this direction, we present a simple Generative Image-to-text Transformer, named GIT, which consists only of one image encoder and one text decoder. The pre-training task is just to map the input image to the entire associated text description with the language modeling objective. Despite its simplicity, GIT achieves new state of the arts across numerous challenging benchmarks with a large margin, as summarized in Table 1.

The image encoder is a Swin-like vision transformer (Dosovitskiy et al., 2021; Yuan et al., 2021) pre-trained on massive image-text pairs based on the contrastive task (Jia et al., 2021; Radford et al., 2021; Yuan et al., 2021). This eliminates the dependency on the object detector, which is used in many existing approaches (Anderson et al., 2018; Li et al., 2020b; Wang et al., 2020; Zhang et al., 2021a; Chen et al., 2020b; Fang et al., 2021c). To extend it to the video domain, we simply extract the features of multiple sampled frames and concatenate them as the video representation. The text decoder is a transformer network to predict the associated text. The entire network is trained with the language modeling task. For VQA, the input question is treated as a text prefix, and the answer is generated in an auto-regressive way. Furthermore, we present a new generation-based scheme for ImageNet classification, where the predicted labels come directly from our generative model without pre-defining the vocabulary.

The approach is simple, but the performance is surprisingly impressive after we scale up the pre-training data and the model size. Fig. 1 shows captions generated by the GIT fine-tuned with TextCaps. The samples

demonstrate the model's strong capability of recognizing and describing scene text, tables, charts, food, banknote, logos, landmarks, characters, celebrities, products, *etc.*, indicating that our GIT model has encoded rich multi-modal knowledge about the visual world.

Our main contributions are as follows.

- We present GIT, which consists of only one image encoder and one text decoder, pre-trained on 0.8 billion image-text pairs with the language modeling task.
- We demonstrate new state-of-the-art performance over numerous tasks on image/video captioning and QA (Table 1), without the dependency on object detectors, object tags, and OCR. On TextCaps, we surpass the human performance for the first time. This implies that a simple network architecture can also achieve strong performance with scaling.
- We demonstrate that GIT pre-trained on the image-text pairs is capable of achieving new state-of-the-art performance even on video tasks without video-dedicated encoders.
- We present a new scheme of generation-based image classification. On ImageNet-1K, we show a decent performance (88.79% top-1 accuracy) with our GIT.

2 Related Work

In VL pre-training, multi-task pre-training has been widely used to empower the network with multiple or enhanced capabilities. For example, MLM and ITM are widely adopted pre-training tasks (Li et al., 2020b; Kim et al., 2021; Zhang et al., 2021a; Wang et al., 2020; Xue et al., 2021b; Lu et al., 2019; Tan & Bansal, 2019). Recently, the image-text contrastive loss has also been added in Yu et al. (2022); Li et al. (2021a); Wang et al. (2021a). Since most VL tasks can be formulated as the text generation task (Cho et al., 2021), a single generation model can be pre-trained to support various downstream tasks. The input and output texts are usually carefully designed to pre-train such a generation model. For example in Cho et al. (2021), the text is properly masked as the network input and the goal is to recover the masked text span. SimVLM (Wang et al., 2021b) randomly splits a text sentence into the input and the target output. In these methods, a multi-modal transformer encoder is utilized to incorporate the text inputs before decoding the output.

For image representation, Faster RCNN has been used in most existing approaches (Anderson et al., 2018; Li et al., 2020b; Wang et al., 2020; Zhang et al., 2021a; Chen et al., 2020b; Fang et al., 2021c) to extract the region features. Recently, a growing interest is in dense representation (Huang et al., 2020; Wang et al., 2021b;a; Kim et al., 2021; Fang et al., 2021b; Dou et al., 2021; Li et al., 2021a) from the feature map, which requires no bounding box annotations. Meanwhile, it is easy to train the entire network in an end-to-end way. In addition to the representation from the feature map, object tags (Li et al., 2020b; Wang et al., 2020; Zhang et al., 2021a; Cornia et al., 2021; Fang et al., 2021b) are leveraged to facilitate the transformer to understand the context, especially the novel objects. For scene-text-related tasks, OCR is invoked to generate the scene text as additional network input, *e.g.*, in Hu et al. (2020); Yang et al. (2021c). For the text prediction, A transformer network is typically used, which can incorporate the cross-attention module to fuse the image tokens, *e.g.*, Cho et al. (2021); Alayrac et al. (2022); Yang et al. (2021b); Yu et al. (2022), or only the self-attention modules where the image tokens are concatenated with the text tokens, *e.g.*, Li et al. (2020b); Chen et al. (2020b); Zhang et al. (2021a); Wang et al. (2020); Fang et al. (2021b).

Along the direction of scaling on VL tasks, LEMON (Hu et al., 2021a) studies the behavior of the detector-based captioning model with MLM. CoCa (Yu et al., 2022) studies different model sizes, but on the same pre-training data. In this paper, we present a comprehensive study on 9 various benchmarks (3 in main paper and 6 in supplementary materials, image/video captioning & QA tasks) with 3 different model sizes and 3 different pre-training data scales (9 data points for each benchmark).

3 Generative Image-to-text Transformer

With large-scale image-text pairs, our goal is to pre-train a VL model which is simple yet effective to benefit image/video captioning and QA tasks. As the input is the image and the output is the text, the minimal set

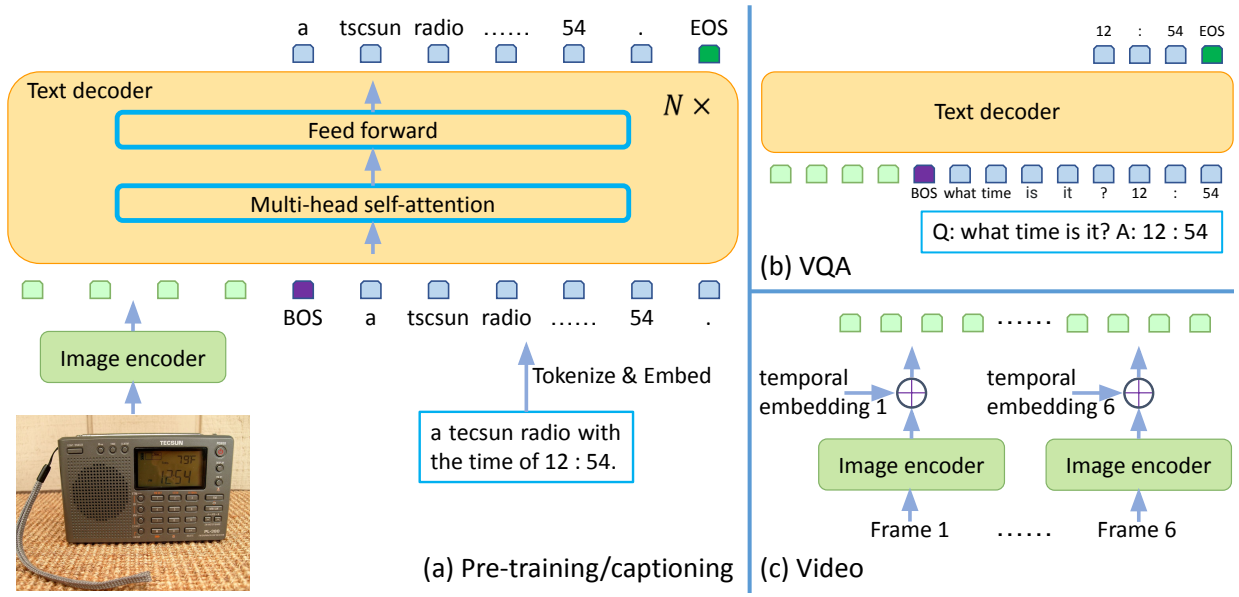


Figure 2: Network architecture of our GIT, composed of one image encoder and one text decoder. (a): The training task in both pre-training and captioning is the language modeling task to predict the associated description. (b): In VQA, the question is placed as the text prefix. (c): For video, multiple frames are sampled and encoded independently. The features are added with an extra learnable temporal embedding (initialized as 0) before concatenation.

of components could be one image encoder and one text decoder, which are the only components of our GIT as illustrated in Fig. 2.

3.1 Network Architecture

The image encoder is based on the contrastive pre-trained model (Yuan et al., 2021). The input is the raw image and the output is a compact 2D feature map, which is flattened into a list of features. With an extra linear layer and a layernorm layer, the image features are projected into D dimensions, which are the input to the text decoder. We use the image encoder pre-trained with contrastive tasks because recent studies show superior performance with such image encoder, e.g. Yuan et al. (2021); Dou et al. (2021); Alayrac et al. (2022). In Sec 4.6 and supplementary materials, we also observe the VL performance boosts significantly with a stronger image encoder. This is consistent with the observation in object detection-based approaches, e.g. in Wang et al. (2020); Zhang et al. (2021a). The concurrent work of CoCa (Yu et al., 2022) unifies the contrastive task and the generation task. as one pre-training phase. Our approach is equivalent to separating the two tasks sequentially: (i) using the contrastive task to pre-train the image encoder followed by (ii) using the generation task to pre-train both the image encoder and text decoder.

The text decoder is a transformer module to predict the text description. The transformer module consists of multiple transformer blocks, each of which is composed of one self-attention layer and one feed-forward layer. The text is tokenized and embedded into D dimensions, followed by an addition of the positional encoding and a layernorm layer. The image features are concatenated with the text embeddings as the input to the transformer module. The text begins with the `[BOS]` token, and is decoded in an auto-regressive way until the `[EOS]` token or reaching the maximum steps. The `seq2seq` attention mask as in Fig. 3 is applied such that the text token only depends on the preceding tokens and all image tokens, and image tokens can attend to each other. This is different from a unidirectional attention mask, where not every image token can rely on all other image tokens.

Instead of well initializing the image encoder, we randomly initialize the text decoder. This design choice is highly motivated from the experiment studies of Wang et al. (2020), in which the random initialization shows

similar performance, compared with the BERT initialization. This could be because the BERT initialization cannot understand the image signal, which is critical for VL tasks. Without dependency of the initialization, we can easily explore different design choices. The concurrent work of Flamingo (Alayrac et al., 2022) employs a similar architecture of image encoder + text decoder, but their decoder is pre-trained and frozen to preserve the generalization capability of the large language model. In our GIT, all parameters are updated to better fit the VL tasks.

An alternative architecture is the cross-attention-based decoder to incorporate the image signals instead of concatenation with self-attention. Empirically as shown in supplementary material (Appendix G.2), with large-scale pre-training, we find the self-attention-based decoder achieves better performance overall, while in small-scale setting, the cross-attention-based approach wins. A plausible explanation is that with sufficient training, the decoder parameters can well process both the image and the text, and the image tokens can be better updated with the self-attention for text generation. With cross-attention, the image tokens cannot attend to each other.

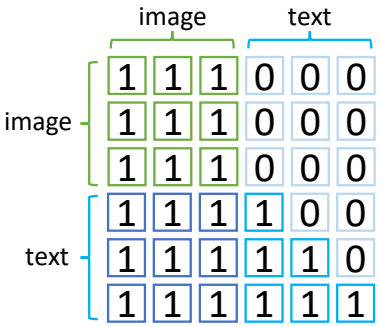


Figure 3: `seq2seq` attention mask is applied to the transformer. If (i, j) is 1, the i-th output can depend on the j-th input; otherwise, not.

3.2 Pre-training

For each image-text pair, let I be the image, $y_i, i \in \{1, \cdots, N\}$ be the text tokens, y_0 be the `[BOS]` token and y_{N+1} be the `[EOS]` token. We apply the language modeling (LM) loss to train the model. That is,

$$l = \frac{1}{N+1} \sum_{i=1}^{N+1} \text{CE}(y_i, p(y_i | I, \{y_j, j = 0, \cdots, i-1\})), \quad (1)$$

where CE is the cross-entropy loss with label smoothing of 0.1.

An alternative choice is MLM, which predicts typically 15% of input tokens in each iteration. To predict all tokens, we have to run at least $1/0.15 = 6.7$ epochs. For LM, each iteration can predict all tokens, which is more efficient for large-scale pre-training data. In Hu et al. (2021a), the ablation studies also show that LM can achieve better performance with limited epochs. In our large-scale training, the number of epoch is only 2 due to computational resource limitation, and thus we choose LM. Meanwhile, most of the recent large-scale language models are also based on LM, e.g. Brown et al. (2020); Chowdhery et al. (2022).

Without the image input, the model is reduced to a decoder-only language model, similar to GPT3 (Brown et al., 2020) in the architecture wise. Thus, this design also enables the possibility to leverage the text-only data to enrich the decoding capability with a scaled-up decoder. We leave this as future work.

3.3 Fine-tuning

For the image captioning task, as the training data format is the same as that in pre-training, we apply the same LM task to fine-tune our GIT.

For visual question answering, the question and the ground-truth answer are concatenated as a new special caption during the fine-tuning, but the LM loss is only applied on the answer and the `[EOS]` tokens. During inference, the question is interpreted as the caption prefix and the completed part is the prediction. Compared with the existing approaches (Wang et al., 2021a;b; Zhang et al., 2021a; Li et al., 2022b) for VQAv2 (Goyal et al., 2017), our model is generative without pre-defining the candidate answers, even in inference. This imposes more challenges as the model has to predict at least two correct tokens: one for the answer and another for `[EOS]`. In contrast, the existing work pre-collects the answer candidate, recasts the problem as a classification problem, and only needs to predict once. However, considering the benefit of the free-form answer, we choose the generative approach. Due to difficulty of the generative model, we observe slightly worse performance on VQAv2 than the discriminative existing work. For the scene-text related VQA tasks, existing approaches (Yang et al., 2021c; Hu et al., 2020) typically leverages the OCR engine to generate the

scene text and use dynamic pointer network to decide the current output token should be OCR or the general text. Here, our approach depends on no OCR engine, and thus no dynamic pointer network. Empirically, we find the model gradually learns how to read the scene text with large-scale pre-training, and our model achieves new SoTA performance on these tasks.

Our model is not specifically designed for the video domain, but we find our model can also achieve competitive or even new SOTA performance with a simple architecture change. That is, we sample multiple frames from each video clip, and encode each frame via the image encoder independently. Afterwards, we add a learnable temporal embedding (initialized as zeros), and concatenate the features from sampled frames. The final representation is used in a similar way as the image representation for captioning and question answering.

We also apply our generation model to the image classification task, where the class names are interpreted as image captions, and our GIT is fine-tuned to predict the result in an auto-regressive way. This is different from existing work which normally pre-defines the vocabulary and uses a linear layer (with softmax) to predict the likelihood of each category. This new generation-based scheme is beneficial when new data and new categories are added to the existing dataset. In this case, the network can continuously train on the new data without introducing new parameters.

4 Experiments

4.1 Setting

We collect 0.8B image-text pairs for pre-training, which include COCO (Lin et al., 2014), Conceptual Captions (CC3M) (Sharma et al., 2018), SBU (Ordonez et al., 2011), Visual Genome (VG) (Krishna et al., 2016), Conceptual Captions (CC12M) (Changpinyo et al., 2021), ALT200M (Hu et al., 2021a), and an extra 0.6B data following a similar collection procedure in Hu et al. (2021a). The image encoder is initialized from the pre-trained contrastive model (Yuan et al., 2021). The hidden dimension (D) is 768. The text decoder consists of 6 randomly-initialized transformer blocks. The total number of model parameters is 0.7 billion. The learning rates of the image encoder and the decoder are $1e^{-5}$ and $5e^{-5}$, respectively, and follow the cosine decay to 0. The total number of epochs is 2. During inference, the beam size is 4 and the length penalty (Wu et al., 2016) is 0.6 by default.

Supplementary materials show results on two smaller model variants (GIT_B and GIT_L) and one even larger model (GIT2) with full details. When comparing with existing approaches, the reference numbers are the best one reported in the corresponding paper unless explicitly specified.

4.2 Results on Image Captioning and Question Answering

We comprehensively evaluate the captioning performance on the widely-used Karpathy split (Karpathy & Li, 2015) of COCO (Lin et al., 2014) and Flickr30K (Young et al., 2014), the COCO test set, nocaps (Agrawal et al., 2019)[2] which focuses on novel objects, TextCaps (Sidorov et al., 2020) which focuses on scene-text understanding, and VizWiz-Captions (Gurari et al., 2020) which focuses on the real use case by the vision-impaired people. The results in CIDEr (Vedantam et al., 2015) are shown in Table 2 and 3. From the results, we can see our model achieves the new SOTA performance on all these metrics except on COCO Karpathy test. On nocaps, compared with CoCa (Yu et al., 2022), our model is much smaller in the model size (0.7B vs 2.1B), but achieves higher performance (123.0 vs 120.6 in CIDEr). On Textcaps, our solution outperforms the previous SOTA (TAP Yang et al. (2021c)) by a breakthrough margin (28.5 points in CIDEr), and also surpasses the human performance for the first time. For zero/few-shot evaluation as shown in Table 3, our model can significantly benefit from more shots. With 32-shots, our approach is also better than Flamingo.

On VQA, the evaluation benchmarks include VQAv2 (Goyal et al., 2017), TextVQA (Singh et al., 2019), VizWiz-VQA (Gurari et al., 2018). ST-VQA (Biten et al., 2019), and OCR-VQA (Mishra et al., 2019). Before fine-tuning the model, we run an intermediate fine-tuning on the combination of the training data of VQAv2, TextVQA, ST-VQA, OCR-VQA, VizWiz-VQA, Visual Genome QA (Krishna et al., 2016), GQA (Hudson &

[2]We compare all approaches including using external image-text datasets.

Table 2: Results on image captioning. *: the nubmers are from Sidorov et al. (2020); CE: cross-entropy optimization. All numbers are CIDEr scores, and other metrics are shown in supplementary materials. #: winner entry of the CVPR 2021 workshop challenge Anc.-Cap.: Xu et al. (2021) AoANet: Huang et al. (2019) BUTD: Anderson et al. (2018), CoCa: Yu et al. (2022), DistillVLM: Fang et al. (2021c), Flamingo: Alayrac et al. (2022), Human: Agrawal et al. (2019), LEMON: Hu et al. (2021a), M4C-Cap.: Hu et al. (2020) MiniVLM: Wang et al. (2020), MTMA: Gong et al. (2021), OFA: Wang et al. (2022b), OSCAR: Li et al. (2020b), UFO: Wang et al. (2021a), UniversalCap: (Cornia et al., 2021) ViTCap: Fang et al. (2021b), VinVL: Zhang et al. (2021a), VIVO: Hu et al. (2021b) SimVLM: Wang et al. (2021b), TAP: Yang et al. (2021c).

Method	CE
MiniVLM	119.8
DistillVLM	120.8
ViTCap	125.2
OSCAR	127.8
VinVL	130.8
UFO	131.2
Flamingo	138.1
LEMON	139.1
SimVLM	143.3
CoCa	143.6
OFA	**145.3**
GIT	144.8

(a) COCO Karp.

Method	C
BUTD	120.5
VinVL	138.7
GIT	**148.8**

(b) COCO test (c40)

Method	test-std
MTMA	94.1
GIT	**114.4**

(c) VizWiz-Captions

Method	Test
OSCAR	80.9
Human	85.3
VIVO	86.6
VinVL	92.5
UFO	92.3
SimVLM	115.2
LEMON	114.3
UniversalCap	119.3
CoCa	120.6
GIT	**123.4**

(d) nocaps

Method	Test
BUTD*	33.8
AoANet*	34.6
M4C-Cap.*	81.0
Anc.-Cap.	87.4
TAP	103.2
TAP#	109.7
Human	125.5
GIT	**138.2**

(e) TextCaps

Table 3: Zero/Few/Full-shot evaluation on Flickr30K with Karpathy split.

Shot	0	16	32	290 (1%)	full
Zhou et al. (2020)	-	-	-	-	68.5
Flamingo	67.2	78.9	75.4	-	-
GIT	49.6	78.0	80.5	86.6	98.5

Manning, 2019), and OK-VQA (Marino et al., 2019). To avoid data contamination, we remove the duplicate images of the test and validation set of the target benchmarks. As illustrated in Table 4, we achieve new SOTA on VizWiz-VQA and OCR-VQA, and same performance with prior SOTA of LaTr (Biten et al., 2022) on ST-VQA. Compared with the concurrent work of Flamingo (Alayrac et al., 2022), we achieve higher accuracy (+5.4) on TextVQA and lower (-3.29) on VQAv2. Note that Flamingo's model size is 80B, which is 114 times of ours (0.7B). On VQAv2, we observe that our model performs worse in 1.5 points than the discriminative model of Florence (Yuan et al., 2021), which shares the same image encoder. The reason might be the increased difficulty of the generative model. That is, each correct answer requires at least two correct predictions (answer and `[EOS]`; 2.2 on average), while the discriminative model requires only one correct prediction. In (Wang et al., 2021b), the ablation study also shows the better performance by around 1 point than the discriminative counterpart. Another reason could be that the model of Florence for VQA leverages RoBerta (Liu et al., 2019) as the text encoder, which implicitly uses the text-only data to improve the performance.

4.3 Results on Video Captioning and Question Answering

On the video captioning task, the performance is evaluated on MSVD (Chen & Dolan, 2011) with the widely-used splits from Venugopalan et al. (2014), MSRVTT (Xu et al., 2016), YouCook2 (Zhou et al., 2018) (results in supplementary materials.) VATEX (Wang et al., 2019b), and TVC (Lei et al., 2020) (results in supplementary materials.). On VATEX, the performance is evaluated on both the public test and private test (evaluated on the server). Video QA is evaluated on MSVD-QA (Xu et al., 2017; Chen & Dolan, 2011), MSRVTT-QA (Xu et al., 2017; 2016), and TGIF-Frame (Jang et al., 2017), which are all open-ended tasks. The results are shown in Table 5 and Table 6 for captioning and QA, respectively. Although our model is not

Table 4: Results on visual question answering. (a): for VQAv2, approaches are divided according to whether the answer vocabulary is pre-defined (Closed) or not (Open) during inference. The model with closed vocabulary can be a classification model or generation model with constrained outputs, *e.g.*, Wang et al. (2022b); Li et al. (2022b). The two numbers in parenthesis are the number of parameters and the number of images (the images for pre-trained modules are not counted) in VL pretraining. (b): for TextVQA, Mia (Qiao et al., 2021)[#] is the winner entry of TextVQA Challenge 2021 with a fine-tuned T5-3B (Raffel et al., 2020) model. (c): [##]: winner entry of 2021 VizWiz Grand Challenge Workshop. ALBEF: Li et al. (2021a), BLIP: Li et al. (2022b), BLOCK+CNN+W2V: Mishra et al. (2019), CLIP-ViL: Shen et al. (2021), CoCa: Yu et al. (2022), CRN: Liu et al. (2020a), Flamingo: Alayrac et al. (2022), Florence: Yuan et al. (2021), LaAP-Net: Han et al. (2020), LaTr: Biten et al. (2022), M4C: Hu et al. (2020), M4C: Hu et al. (2020), METER: Dou et al. (2021), Mia: Qiao et al. (2021), mPlug: Li et al. (2022a), OSCAR: (Li et al., 2020b), OFA: Wang et al. (2022b), UFO: Wang et al. (2021a), UNITER: (Chen et al., 2020b), UNIMO: Li et al. (2021c), SA-M4C: Kant et al. (2020), SimVLM: Wang et al. (2021b), SMA Gao et al. (2020), SMA: Gao et al. (2020), TAP: Yang et al. (2021c), VinVL: Zhang et al. (2021a), VILLA: Gan et al. (2020).

Vocabulary	Method	test-std
Closed	OSCAR	73.82
	UNITER	74.02
	VILLA	74.87
	UNIMO	75.27
	ALBEF	76.04
	VinVL	76.60
	UFO	76.76
	CLIP-ViL	76.70
	METER	77.64
	BLIP	78.32
	SimVLM (-, 1.8B)	80.34
	Florence (0.9B, 14M)	80.36
	mPlug (0.6B, 14M)	81.26
	OFA (0.9B, 54M)	82.0
	CoCa (2.1B, 4.8B)	**82.3**
Open	Flamingo (80B, 2.3B)	**82.1**
	GIT (0.7B, 0.8B)	78.81

(a) VQAv2

Method	test
M4C	40.46
LaAP-Net	41.41
SA-M4C	44.6
SMA	45.51
TAP	53.97
Flamingo	54.1
Mia	**73.67**
GIT	59.75

(b) TextVQA

Method	test
(Liu et al., 2021)[##]	60.6
Flamingo	65.4
GIT	**67.5**

(c) VizWiz-QA

Method	Test ANLS
M4C	46.2
SMA	46.6
CRN	48.3
LaAP-Net	48.5
SA-M4C	50.4
TAP	59.7
LaTr	**69.6**
GIT	**69.6**

(d) ST-VQA

Method	test
BLOCK+CNN+W2V	48.3
M4C	63.9
LaAP-Net	64.1
LaTr	67.9
GIT	**68.1**

(e) OCR-VQA

dedicated for video tasks, our model achieve new SOTA on MSRVD, MSRVTT, and VATEX for captioning and on MSVD-QA and TGIF-Frame for QA. For example on VATEX private test, our results are even better (93.8 vs 86.5) than CLIP4Caption++ (Tang et al., 2021), which relies on model ensemble and additional subtitle input. This is also better than Flamingo (Alayrac et al., 2022) (84.2) with 80B parameters.

4.4 Results on Image Classification

We fine-tune GIT on ImageNet-1k. Each category is mapped to a unique class name, and the prediction is correct only if it is exactly matched with the ground-truth label subject to more or fewer whitespaces[3]. As shown in Table 7, our approach can achieve descent accuracy without pre-defining the vocabulary. Compared with Florence (Yuan et al., 2021) (same image encoder), our approach is worse in about 1.2 points. The reason might be similar to the case on VQAv2. That is, the generative approach needs to predict more tokens correctly to make one correct prediction, which increases the difficulty.

Zero-shot/Few-shot. The result is shown in Table 9. With no knowledge of the vocabulary, the pretrained GIT cannot infer the expected vocabulary, and thus the exactly-match accuracy is only 1.93% (in the column of *equal*). However, if we relax the requirement and take it correct if the prediction contains the ground-truth, the accuracy is 40.88% (in the column of *in*), which shows the predicted caption can well identify the image content. If we have the vocabulary as a prior and limit the output tokens to be within the vocabulary, the accuracy drops to 33.48% (in the column of *voc-prior*). This may suggest the network is less natural to

[3]pred.replace(‘ ’, ‘’) == gt.replace(‘ ’, ‘’)

Table 5: Results on video captioning. ^{E}: model ensemble; ^{T}: with the subtitle as additional input. C.4Cap.: Tang et al. (2021) GRU-EVE: Aafaq et al. (2019) MGSA: Chen & Jiang (2019) MGSA: Chen & Jiang (2019) MV-GPT: Seo et al. (2022) PickNet: Chen et al. (2018) PMI-CAP: Chen et al. (2020a) SibNet: Liu et al. (2020b) OA-BTG: Zhang & Peng (2019) ORG-TRL: Zhang et al. (2020) OpenBook: Zhang et al. (2021b) POS+VCT: Hou et al. (2019) POS+CG: Wang et al. (2019a) SAAT: Zheng et al. (2020), STG-KD: Pan et al. (2020) SwinBERT: Lin et al. (2021a) Support-set: Patrick et al. (2021) VaTeX: Wang et al. (2019b) VALUE: Li et al. (2021b)

Method	B@4	C
PickNet	52.3	76.5
GRU-EVE	47.9	78.1
SAAT	46.5	81.0
MGSA	53.4	86.7
POS+VCT	52.8	87.8
SibNet	54.2	88.2
POS+CG	52.5	88.7
OA-BTG	56.9	90.6
STG-KD	52.2	93.0
PMI-CAP	54.6	95.1
ORG-TRL	54.3	95.2
SwinBERT	58.2	120.6
GIT	**79.5**	**180.2**

(a) MSVD

Method	B@4	C
SAAT	39.9	51.0
MGSA	42.4	47.5
POS+VCT	42.3	49.1
SibNet	40.9	47.5
POS+CG	42.0	48.7
OA-BTG	41.4	46.9
STG-KD	40.5	47.1
Support-set	38.9	48.6
PMI-CAP	42.1	49.4
ORG-TRL	43.6	50.9
OpenBook	33.9	52.9
SwinBERT	41.9	53.8
MV-GPT^{T}	48.9	60
GIT	**53.8**	**73.9**

(b) MSRVTT

Method	C
VaTeX	45.1
OpenBook	57.5
VALUE^{T}	58.1
SwinBERT	73.0
C.4Cap.^{ET}	85.7
GIT	**91.5**

(a) VATEX public test

Method	C
X-L.+T.^{E}	81.4
Flamingo	84.2
C.4Cap.^{ET}	86.5
GIT	**93.8**

(e) VATEX private test

Table 6: Results on video question answering. All are open-ended question answering tasks. All-in-one: Wang et al. (2022a), ClipBERT: Lei et al. (2021), CoMVT: Seo et al. (2021), Flamingo: Alayrac et al. (2022), JustAsk: Yang et al. (2021a), MERLOT: Zellers et al. (2021), MV-GPT: Seo et al. (2022), QueST: Jiang et al. (2020), HCRN: Le et al. (2021), VIOLET: Fu et al. (2021).

Method	Accuracy
QueST	34.6
HCRN	36.1
CoMVT	42.6
JustAsk	46.3
VIOLET	47.9
All-in-one	48.3
GIT	**56.8**

(a) MSVD-QA

Method	Accuracy
JustAsk	41.5
MV-GPT	41.7
MERLOT	43.1
VIOLET	43.9
All-in-one	46.8
Flamingo	**47.4**
GIT	43.2

(b) MSRVTT-QA

Method	Accuracy
HCRN	55.9
QueST	59.7
ClipBERT	60.3
All-in-one	66.3
VIOLET	68.9
MERLOT	69.5
GIT	**72.8**

(c) TGIF-Frame

directly predict the category name. By fine-tuning the model with only 1 shot or 5 shots per category, we observe that the accuracy is significantly improved. This demonstrates our model can be easily adapted to downstream tasks even with a few training samples. With the shot increased from 1 to 5, the gap between *voc-prior* and the other two columns (*equal* and *in*) becomes smaller. This is expected as more shots can be better to guide the network to predict in-vocabulary output.

Compared with Flamingo, our GIT achieves higher accuracy. Flamingo conducts the few-shot learning without parameter update, but each test image is combined with the support training examples as extra network inputs. Meanwhile, different test image requires different support shots based on Yang et al. (2022b). These may increase the inference cost. In contrast, our model updates the parameters by a lightweight fine-tuning once, and then all these training shots are not required during inference.

4.5 Results on Scene Text Recognition

The task (Graves et al., 2006) aims to read scene text directly from the image. We evaluate our model in two settings. One is the GIT fine-tuned on TextCaps. The prediction is considered correct if the caption

Table 7: Results on ImageNet-1k classification task. Our approach takes the class name as the caption and predict the label in an auto-regressive way without pre-defining the vocabulary.

Vocabulary	Method	Top-1
Closed	ALIGN (Jia et al., 2021)	88.64
	Florence (Yuan et al., 2021)	90.05
	CoCa (Yu et al., 2022)	**91.0**
Open	GIT	88.79

Table 8: Results on scene text recognition. MJ and ST indicate the MJSynth (MJ) (Jaderberg et al., 2014; 2016) and SynthText (ST) (Gupta et al., 2016) datasets used for training scene text recognition models.

Method	FT data	Average
SAM (Liao et al., 2019)	MJ+ST	87.8
Ro.Scanner (Yue et al., 2020)	MJ+ST	87.5
SRN (Yu et al., 2020)	MJ+ST	89.6
ABINet (Fang et al., 2021a)	MJ+ST	91.9
S-GTR (He et al., 2022b)	MJ+ST	91.9
MaskOCR (Lyu et al., 2022)	MJ+ST	**93.8**
GIT	TextCaps	89.9
	MJ+ST	92.9

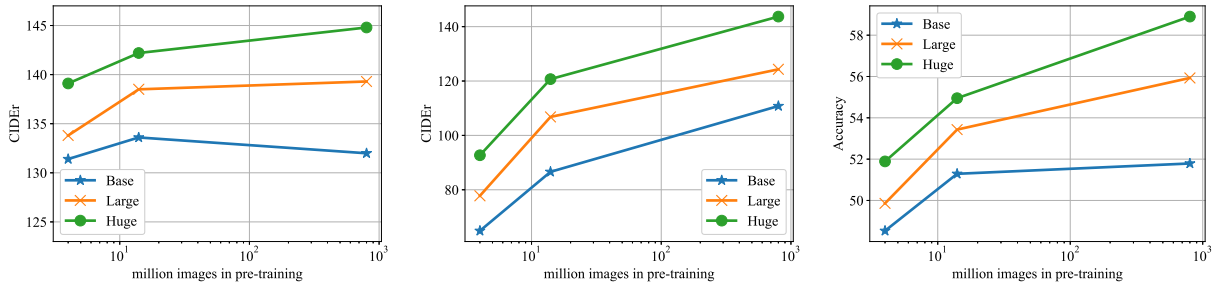
Table 9: Zero/Few-shot evaluation on ImageNet with 3 metrics. *equal*: the unrestricted prediction should be exactly matched to the ground-truth. *in*: the unrestricted prediction should contain the ground-truth label name. *voc-prior*: the vocabulary is pre-defined as a prior. For our GIT, a trie structure is constructed motivated from Wang et al. (2022b) to limit the candidate tokens during each token prediction, such that the predicted result is guaranteed to be within the vocabulary.

	Zero-shot			1-shot per class			5-shot per class		
Accuracy type	equal	in	voc-prior	equal	in	voc-prior	equal	in	voc-prior
Flamingo	-	-	-	-	-	71.7	-	-	77.3
GIT	1.93	40.88	33.48	64.54	66.76	72.45	79.79	80.15	80.95

contains the ground-truth scene text word. The other is to fine-tune the model on two large scene text datasets: MJSynth (MJ) (Jaderberg et al., 2014; 2016) and SynthText (ST) (Gupta et al., 2016), where the ground-truth scene text is used as the *caption*. The prediction is correct if the output is the exact match to the ground-truth. Following the established setup, we evaluate on six standard benchmarks, including ICDAR 2013 (IC13) (Karatzas et al., 2013), ICDAR 2015 (IC15) (Karatzas et al., 2015), IIIT 5K-Words (IIIT) (Mishra et al., 2012), Street View Text (SVT) (Wang et al., 2011), Street View Text-Perspective (SVTP) (Phan et al., 2013), and CUTE80 (CUTE) (Risnumawan et al., 2014). The average accuracy is reported in Table 8. The accuracy on individual test sets is in supplementary materials. Our TextCaps-fine-tuned captioning model achieves an 89.9 accuracy, which demonstrates the strong scene text comprehension capability of our captioning model. After fine-tuning the model on the standard MJ+ST datasets, GIT achieves 92.9 that surpasses the prior arts (Fang et al., 2021a; He et al., 2022b) of 91.9.

4.6 Analysis

Model and data scaling. To study the trending with data scales, we construct two smaller pre-training datasets: one is the combination of COCO, SBU, CC3M and VG, leading to 4M images or 10M image-text pairs; the other is to further combine CC12M, leading to about 14M images or 20M image-text pairs. When pre-training on small-scale datasets, we use 30 epochs rather than 2 epochs as on the 0.8B data. For the network structure, we name our model as *Huge* and replace the image encoder with ViT-B/16 and ViT-L/14 from CLIP Radford et al. (2021) as *Base* and *Large*, respectively. Fig. 4 shows the results on COCO, TextCaps, and VizWiz-QA. On COCO, the base model benefits from 4M to 14M, but the performance drops with 0.8B data. The 14M data are more similar to COCO than the majority of the noisy 0.8B data. Meanwhile, the Base model with limited capacity may not be able to benefit effectively from large-scale data. Similar observations are also reported in Kolesnikov et al. (2020) for ImageNet-1k classification. On TextCaps and VizWiz-QA, all model variants benefit significantly from more pre-training data. Also, a larger backbone improves more especially with 0.8B data.



(a) COCO (b) TextCaps (c) VizWiz-QA

Figure 4: Performance with different pre-training data scales and different model sizes.

Table 10: Ablation study of larger text decoders. The models are pre-trained on a subset of 0.4B image-text pairs. No beam search and no SCST are performed.

Layers	COCO				nocaps	
	B@4	M	C	S	C	S
6	38.9	30.7	136.4	24.6	119.3	15.9
12	38.9	30.6	136.0	24.2	118.1	15.5
24	39.1	30.2	134.6	23.8	115.4	15.1

Here, we scale the image encoder. Empirically, we find it is difficult to effectively scale up the text decoder. Preliminary results are shown in Table 10, which shows a larger decoder shows no improvement. The reason might be that it is difficult to effectively train with limited amount of text by LM. Another plausible reason is that the image encoder is responsible for object recognition, and the decoder is responsible for organizing the object terms in a natural language way. The latter task might be easy since most of the descriptions follow similar patterns, e.g. object + verb + subject, and thus a small decoder is enough during end-to-end training. Larger decoders increase the learning difficulty, which might degrade the performance.

Flamingo (Alayrac et al., 2022) shows a larger decoder improves the performance. However, their decoder is pre-trained and frozen during the VL pre-training, which avoids the problem of how to effectively train the decoder. In LEMON (Hu et al., 2021a), the transformer can be scaled up to 32 layers. The reason could be that LEMON uses MLM, instead of LM, which might be more difficult to train.

Scene text in pre-training data. To understand the capability of scene text comprehension, we examine the pre-training dataset and study how many image-text pairs contain the scene text. We first run the Microsoft Azure OCR API[4] against all images in CC12M and 500K images in the web crawled images. The OCR result is compared with the associated text. It is considered *matched* only if the text contains an OCR result that is longer than 5 characters. It is estimated that 15% of CC12M and 31% of the downloaded images contain scene text descriptions. As the training task is to predict the texts, the network gradually learns to read the scene text.

5 Conclusion

In the paper, we design and train a simple generative model, named GIT, to map the input image to the associated text description on large-scale image-text pairs. On image/video captioning and question answering tasks, our model achieves new state-of-the-art performance across numerous benchmarks and surpasses the human performance on TextCaps for the first time. For the image classification, we apply the generation task to predict the label name directly. The strategy is different from the existing work with a pre-defined and fixed vocabulary, and is beneficial especially when new category data are added.

[4]https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-recognizing-text

Limitations. We focus on the pretraining-and-finetuning strategy to improve the absolute performance. Empirically, we find it is unclear on how to control the generated caption and how to perform in-context learning without parameter update, which we leave as future work.

Societal impact. Compared with the existing work, our model clearly improves the performance and be more appropriate to help visually-impaired people. The model is pre-trained on large-scale data, and the data are not guaranteed to contain no toxic language, which may poison the output. Although we observe few such instances qualitatively, special care should be taken to deploy the model in practice and more research exploration is required to control the output.

Appendix

The supplementary materials provide more details on the experiments, including results with different model variants, more visualizations, ablation analysis on decoder architectures, more results on data and model scaling, *etc.*

A Setting

A.1 Data Preprocessing

We follow Wang et al. (2021a) to preprocess the pre-training data. That is, make sure the shorter length of the image no larger than 384 and the longer side no larger than 640 while maintaining the aspect ratio. Meanwhile, all images are re-saved with quality being 90 in the JPEG format. This results in 39 terabytes. No such preprocessing is applied on the fine-tuning dataset.

A.2 Platform

The data are stored in Azure Blob Storage[5], and the training is conducted on A100 provisioned by Azure Machine Learning[6]. The code is in python with packages including Pytorch[7] DeepSpeed[8], Transformers[9], maskrcnn-benchmark[10], CLIP[11], OSCAR[12], and VirTex (Desai & Johnson, 2021)[13].

A.3 Network

In the main paper, we present the results of our GIT. Here, we construct two smaller model variants, named GIT_B and GIT_L on smaller pre-training dataset. As shown in Table 11, GIT_B uses CLIP/ViT-B/16 (Radford et al., 2021) as the image encoder and is pre-trained on 10M image-text pairs or 4M images, which is a combination of COCO, SBU, CC3M and VG. GIT_L uses CLIP/ViT-L/14 (Radford et al., 2021) as the image encoder and is pre-trained on 20M image-text pairs or 14M images, which is a combination of the 10M image-text pairs with CC12M.

The three model variants share the same pre-training hyperparameters. The learning rate is warmed up in the first 500 iterations, and then follows cosine decay to 0. The learning rate is $1e^{-5}$ for the image encoder and is multiplied by 5 for the randomly initialized text decoder. The batch size is 4096. Parameters are updated by AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The number of epochs is 2.

As the performance exhibits no signs of plateau, we further scale up the model size to 5.1B and the number of pretraining images to 10.5B (12.9B image-text pairs). The image encoder is scaled to 4.8B based on

[5] `https://azure.microsoft.com/en-us/services/storage/blobs`
[6] `https://docs.microsoft.com/en-us/azure/machine-learning/`
[7] `https://pytorch.org/`, license: `https://github.com/pytorch/pytorch/blob/master/LICENSE`
[8] `https://github.com/microsoft/DeepSpeed`, MIT license)
[9] `https://github.com/huggingface/transformers`, Apache License 2.0
[10] `https://github.com/facebookresearch/maskrcnn-benchmark`, MIT license
[11] `https://github.com/openai/CLIP`, MIT license
[12] `https://github.com/microsoft/Oscar`, MIT license
[13] `https://github.com/kdexd/virtex`, MIT license

Table 11: Model configurations in pre-training. The decoder is a 6-layer transformer network. The hidden size is 768 with 12 attention heads except GIT2. Parameters of text token embeddings and the last projection weight before the softmax layer are shared and not counted in the model size.

Name	images	image-text pairs	image encoder	epochs	model size	image size
GIT_B	4M	10M	CLIP/ViT-B/16 (Radford et al., 2021)	30	129M	224
GIT_L	14M	20M	CLIP/ViT-L/14 (Radford et al., 2021)	30	347M	224
GIT	0.8B	0.8B	Florence/CoSwin (Yuan et al., 2021)	2	681M	384
GIT2	10.5B	12.9B	DaViT (Ding et al., 2022) (4.8B)	2	5.1B	384

DaViT (Ding et al., 2022) and is pre-trained with the UniCL (Yang et al., 2022a; Yuan et al., 2021) task. The text decoder is enlarged to 0.3B, the hyperparameters (number of transformer layers, hidden dimension, etc) of which follow BERT-Large (Devlin et al., 2018). The model is named as GIT2.

A.4 Implementation of the Data Loader

A challenging problem is to implement the data loader efficiently as the total data size (39TB for the 0.8B images) is much larger than the local disk size (around 7TB). As the data are stored in Azure Storage, we download the data to the local disk before reading it rather than directly from the cloud. Considering the data scale may increase even larger in the future, we should make sure each operation is independent to the dataset size. In the meanwhile, the data downloading should be overlapped with the GPU computing, such that the data are always locally available when needed. The solution is outlined as follows.

1. The image-text pairs are evenly split among C compute nodes. Each node only accesses the corresponding part.
2. Each node consumes the data trunk by trunk. Each trunk is 2^{20} image-text pairs except the last which may have fewer than 2^{20} data.
3. The data in each trunk is randomly shuffled. We shuffle the data in the trunk level such that the cost is not related with the dataset size, and hence it can be applied to even larger dataset.
4. The shuffled trunk data are split evenly among the GPUs within the node.
5. One extra process on each node (launched by local rank = 0) is created to pre-fetch at most 7 future trunks. As each trunk is designed for all ranks in one node, it is not required for other ranks to launch the pre-fetching process, which avoids the race condition.
6. Local storage contains at most 12 trunk data, and the oldest will be removed.

Empirically, we observe almost no[14] time cost on the data loading during model training and the speed is also stable.

B Results on Image Captioning

On each task, the model is fine-tuned with 10 epochs. The batch size is 512 and the learning rate is $2.5e^{-6}$. SCST (Rennie et al., 2017) follows the same hyperparameters if performed.

COCO Fig. 12 shows the complete results including GIT_B and GIT_L on COCO Karpathy split (Karpathy & Li, 2015). For the base-sized and large-sized models, our model achieves competitive performance with existing approaches but with a simplified architecture. We observe that UniversalCaptioner (Cornia et al., 2021) achieves much better performance. As a strong image encoder of CLIP/ViT-L with 0.3B parameters is used in UniversalCaptioner for both the base and large model, effectively, the model size is much larger

[14]That is, the data preprocessing is faster than the training and is overlapped with the GPU training.

Table 12: Results on COCO captioning with Karpathy (Karpathy & Li, 2015) split. SimVLM: C4 (800GB) dataset are used and not included in the table; Flamingo: 27M video-text pairs are not counted in the table. UniversalCaptioner: the extra 0.3B in parameters is CLIP/ViT-L, which is used as feature and keyword extractor. the data for pre-training CLIP/ViT-L are not counted . VinVL/LEMON/OSCAR/MiniVLM/DistillVLM: the extra parameters are for object detector; data for the object detectors are not counted. CoCa: Yu et al. (2022), BLIP: Li et al. (2022b), mPLUG: Li et al. (2022a), MiniVLM: Wang et al. (2020), DistillVLM: Fang et al. (2021c), Flamingo: Alayrac et al. (2022), LEMON: Hu et al. (2021a), OSCAR: Li et al. (2020b), OFA: Wang et al. (2022b), UFO: Wang et al. (2021a), UniversalCap: Cornia et al. (2021), VinVL: Zhang et al. (2021a), ViTCap: Fang et al. (2021b), SimVLM: Wang et al. (2021b).

Method	#Param.	#Images	Cross-Entropy				SCST			
			B@4	M	C	S	B	M	C	S
Tiny-sized models										
MiniVLM	46M+8M	11M	35.6	28.6	119.8	21.6	39.2	29.7	131.7	23.5
DistillVLM	46M+8M	4M	35.6	28.7	120.8	22.1	-	-	-	-
Base-sized models										
ViTCap	0.2B	4M	36.3	29.3	125.2	22.6	41.2	30.1	138.1	24.1
OSCAR_{B}	0.1B+64M	4M	36.5	30.3	123.7	23.1	40.5	29.7	137.6	22.8
VinVL_{B}	0.1B+0.2B	6M	38.2	30.3	129.3	23.6	40.9	30.9	140.4	25.1
UFO_{B}	0.1B	4M	36.0	28.9	122.8	22.2	-	-	-	-
$\text{UniversalCap}_{\text{B}}$	0.2B+0.3B	36M	-	-	-	-	42.9	31.4	149.7	25.0
GIT_{B}	0.1B	4M	40.4	30.0	131.4	23.0	41.3	30.4	139.1	24.3
Large-sized models										
OSCAR_{L}	0.3B+64M	4M	37.4	30.7	127.8	23.5	41.7	30.6	140.0	24.5
VinVL_{L}	0.3B+0.2B	6M	38.5	30.4	130.8	23.4	41.0	31.1	140.9	25.2
UFO_{L}	0.3B	4M	38.7	30.0	131.2	23.3	-	-	-	-
$\text{BLIP}_{\text{ViT-L}}$	-	129M	40.4	-	136.7	-	-	-	-	-
$\text{UniversalCap}_{\text{L}}$	0.5B+0.3B	36M	-	-	-	-	42.9	31.5	150.2	25.2
mPLUG	0.6B	14M	43.1	31.4	141.0	24.2	**46.5**	32.0	**155.1**	26.0
GIT_{L}	0.3B	14M	42.0	30.8	138.5	23.8	42.3	31.2	144.6	25.4
Huge/Giant-sized models										
Flamingo	80B	2.3B	-	-	138.1	-	-	-	-	-
$\text{LEMON}_{\text{huge}}$	0.7B+0.2B	0.2B	41.5	30.8	139.1	24.1	42.6	31.4	145.5	25.5
$\text{SimVLM}_{\text{Huge}}$	-	1.8B	40.6	33.7	143.3	**25.4**	-	-	-	-
OFA	0.9B	54M	43.9	31.8	**145.3**	24.8	**44.9**	**32.5**	154.9	**26.6**
CoCa	2.1B	4.8B	40.9	**33.9**	143.6	24.7	-	-	-	-
GIT	0.7B	0.8B	**44.1**	31.5	144.8	24.7	44.1	32.2	151.1	26.3
GIT2	5.1B	10.5B	**44.1**	31.4	145.0	24.8	44.0	32.2	152.7	26.4

Table 13: Results on COCO test set evaluated on the public server. c5/c40: Each image is paired with 5 or 40 reference ground-truth captions. B: BLEU (Papineni et al., 2002); M: METEOR (Denkowski & Lavie, 2014); R: ROUGE-L (Lin & Och, 2004); C: CIDEr-D (Vedantam et al., 2015). BUTD: Anderson et al. (2018), VinVL: Zhang et al. (2021a).

Method	B@1		B@2		B@3		B@4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
BUTD	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
VinVL	81.9	96.9	66.9	92.4	52.6	84.7	40.4	74.9	30.6	40.8	60.4	76.8	134.7	138.7
GIT	84.0	97.9	69.8	94.4	55.6	87.6	43.2	78.3	31.9	42.0	62.0	78.4	145.5	148.8
GIT2	84.5	98.1	70.0	94.4	55.7	87.6	43.2	78.3	31.9	42.1	62.0	78.4	146.4	**149.8**

Table 14: Results on nocaps. in.: in-domain; near.: near domain; out.: out-of-domain; C: CIDEr. S: SPICE. OSCAR: Li et al. (2020b), Human: Agrawal et al. (2019), VIVO: Hu et al. (2021b), VinVL: Zhang et al. (2021a), UFO: Wang et al. (2021a), mPLUG: Li et al. (2022a), SimVLM: Wang et al. (2021b), LEMON: Hu et al. (2021a), UniversalCap: Cornia et al. (2021), CoCa: Yu et al. (2022).

Method	Validataion set								Test set							
	in.		near.		out.		overall		in.		near.		out.		overall	
	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
OSCAR	85.4	11.9	84.0	11.7	80.3	10.0	83.4	11.4	84.8	12.1	82.1	11.5	73.8	9.7	80.9	11.3
Human	84.4	14.3	85.0	14.3	95.7	14.0	87.1	14.2	80.6	15.0	84.6	14.7	91.6	14.2	85.3	14.6
VIVO	92.2	12.9	87.8	12.6	87.5	11.5	88.3	12.4	89.0	12.9	87.8	12.6	80.1	11.1	86.6	12.4
VinVL	103.7	13.7	95.6	13.4	83.8	11.9	94.3	13.1	98.0	13.6	95.2	13.4	78.0	11.5	92.5	13.1
UFO	103.9	14.5	95.5	13.8	83.5	12.3	94.3	13.6	98.9	14.3	94.7	13.9	77.9	12.1	92.3	13.6
mPLUG	-	-	-	-	-	-	114.8	14.8	-	-	-	-	-	-	-	-
SimVLM	113.7	-	110.9	-	115.2	-	115.2	-	113.7	-	110.9	-	115.2	-	115.2	-
LEMON	118.0	15.4	116.3	15.1	120.2	14.5	117.3	15.0	112.8	15.2	115.5	15.1	110.1	13.7	114.3	14.9
UniversalCap	123.2	15.0	121.5	15.3	123.4	14.4	122.1	15.0	118.9	15.4	120.6	15.3	114.3	14.1	119.3	15.1
CoCa	-	-	-	-	-	-	122.4	15.5	-	-	-	-	-	-	120.6	15.5
GIT_B	100.7	13.8	97.7	13.5	89.6	12.5	96.6	13.4	-	-	-	-	-	-	-	-
GIT_L	107.7	14.9	107.8	14.5	102.5	13.7	106.9	14.4	-	-	-	-	-	-	-	-
GIT	**129.8**	**16.3**	124.1	16.0	127.1	15.7	125.5	16.0	122.4	16.2	123.9	16.0	122.0	**15.7**	123.4	15.9
GIT2	126.9	16.1	**125.8**	**16.2**	**130.6**	**15.8**	**126.9**	**16.1**	**124.2**	**16.4**	**125.5**	**16.1**	**122.3**	15.6	**124.8**	**16.1**

than those in respective categories. In the meanwhile, both UniversalCaptioner (Cornia et al., 2021) and OFA (Wang et al., 2022b) use more data than our approach within base/large-sized model sizes. Fig. 13 shows the full results on the COCO test set.

nocaps. The main paper presents the overall performance on nocaps. Table 14 contains the complete results for each sub domain and other model variants.

Fig. 5 shows random[15] prediction examples on the nocaps validation set. To visualize the novel concept recognition capability, we also collect sample images whose prediction contains at least one word not in the COCO training set, as illustrated in Fig. 6. As we can see, the model can well identify the novel object without the object tags as the network input.

TextCaps. No SCST (Rennie et al., 2017) is performed. Table 15 shows full results. Fig. 7 shows predictions on random validation images. We also manually group the predictions according to different scenarios, as illustrated in Fig. 8 and 9. In Fig. 8, (1-5) show examples on which the model describes the digital time displayed on screens, which is correct most of the time. (6-10) provide examples of reading scene text in Latin (Romance) languages such as French and Spanish. (11-15) show GIT's ability in recognizing scene text in languages such as Arabic, Japanese, Korean, and Chinese. (16-20) provide examples of recognizing scene text in stylized fonts. As shown in (21-25), GIT also performs well in reading curved scene text, which is generally considered a challenging case in scene text recognition studies. In Fig. 9, samples (1-5) show examples of reading numbers on jerseys. As shown in (6-10), we observe that GIT has a strong ability in inferring occluded scene text, based on both visual and text context information. For example, "blue jays" is a baseball team name in sample (6), "asahi" is a beer brand in sample (9), and the occluded letter could be letter "t" in sample (8). (11-15) provide examples of reading hand-written scene text. (16-20) demonstrate GIT's ability in reading long pieces of scene texts. GIT works well in organizing scene text words into a fluent and informative sentence. (21-25) show the challenging case of describing a book page, where the model needs to recognize and select the key information to describe. For example in sample (24), GIT covers the name and author of the book in the image.

In addition to the scene text captioning ability, we observe that the TextCaps-fine-tuned GIT is knowledgeable and can produce diverse and informative captions. We group the representative captions in Fig. 10. Samples

[15]Disgusting images and images containing clear people identification information are excluded.

Figure 5: Captioning results of our COCO-fine-tuned GIT on random samples from the nocaps validation set. Words not in COCO training captions are underlined.

Pred: a white stingray swimming in an aquarium.

Pred: a bumble bee sitting on a white flower.

Pred: a large starfish and fish swimming in the water.

Pred: a blue starfish sitting on top of a coral.

Pred: a small dragonfly sitting on top of a blade of grass.

Pred: a black and white lemur sitting in a tree.

Pred: a green roulette table in a room with chairs.

Pred: a couple of barbies on a cake.

Pred: a lion head door knocker on a wooden door.

Pred: a group of lipsticks sitting next to each other.

Pred: a bug sitting on top of a yellow dandelion.

Pred: a banjo sitting on top of a table with a giveaway sign.

Pred: a group of violins hanging on a wall.

Pred: a photocopier machine sitting on top of a white background.

Pred: a closet with a white kallax shelves and clothes.

Pred: a row of jeans stacked up with the date of september.

Pred: a small chipmunk eating nuts on the floor.

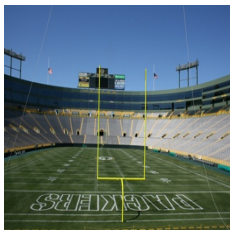
Pred: a close up of a wasp nest on a green leaf.

Pred: a sonicare electric toothbrush in a package.

Pred: a blue and white spotted stingray laying on the sand.

Figure 6: Captioning results of our COCO-fine-tuned GIT on random samples whose prediction contains novel terms from the nocaps validation set. Novel terms, which are not in COCO training captions, are underlined.

Pred: a bottle of dog house wine sits on a table.



Pred: a green bay packers football field with the goal posts.

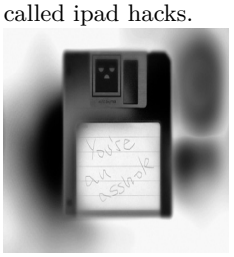
Pred: two computer monitors are on a desk, one of which is called ipad hacks.

Pred: a black and red poster that says end police impunity.

Pred: a collection of tin boxes on a shelf with one that says star trek.

Pred: a baseball player with the number 37 stands on the mound.

Pred: a baseball player with the number 35 on his jersey



Pred: a black and white photo of a floppy disk that says you're an asshole.

Pred: an orange train with the number 16433r on the side.

Pred: a billboard for the fifth elephant machine learning and analytics conference.

Pred: a clock with the name charvet on it

Pred: soccer players with the number 13 on their jersey

Pred: an airasia zest plane is parked on the tarmac.

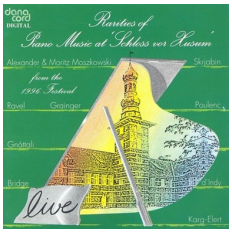
Pred: a can of sanpellegrino chino sits on a table.

Pred: four iphones are lined up on a carpet.

Pred: a grocery store aisle full of canned peas.

Pred: a white lenovo laptop with a black screen on it.

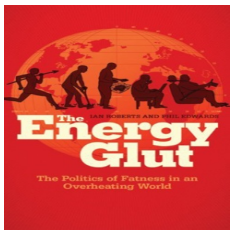
Pred: a green street sign for troublesome valley road.



Pred: a cd cover that says " rarities of piano music at schloss vor kusum ".

Pred: a white shelf with a picture of a dog on it.

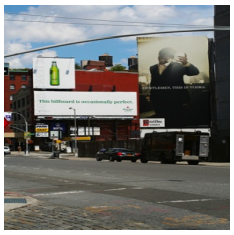
Pred: a poster for the game dynasty warriors gundam reborn.



Pred: the cover of the book the energy glut by ian roberts and phil edwards.

Pred: several cars are lined up in front of a casino.

Pred: a bottle of il bruciato wine from 2009.



Pred: a billboard that says this billboard is occasionally perfect.

Figure 7: Visualization of our model on random validation images of TextCaps.

(1)

A tecsun radio with the time of 12 : 54.

(2)

A cell phone screen shows the time of 1 : 44 wednesday, november 4.

(3)

A lenovo phone with the time of 03 : 03 on the screen.

(4)

A phone that has the time 13 : 05 on it.

(5)

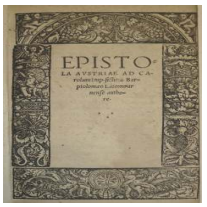
A blue lg phone with a colorful background and the time 10 : 30.

(6)

A poster for the movie la nuit du loup garou.

(7)

A trash can with a red sign that says la poubelle, l'ebouseur et le citoyen.

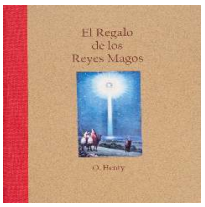


(8)

An old book with a page that says episto la avstriae ad ca.

(9)

A bottle of la fin du monde is next to a glass.



(10)

A book by o. henry titled el regalo de los reyes magos.

(11)

A red and white stop sign with arabic writing on it.

(12)

A plastic bag with a blue monster figure and a japanese poster.

(13)

A phone screen with korean text and usb on it.

(14)

Two books on a wooden table with chinese characters on the cover.

(15)

A bottle of whisbih liq is shown with chinese writing.

(16)

A bottle of aecht schlenferla rauchbier next to a glass of beer.

(17)

A poster that says in god i trust on it.

(18)

A ruler measures a banknote that says " hai dong " on it.

(19)

A blue poster that says build on it.

(20)

A bottle of rubino del casale vino da tavola rosso.

(21)

A united states of america half dollar coin is in a plastic container.

(22)

A gold and black sign that says university of colorado 1876.

(23)

A coin from 1969 with the word in god we trust on it.

(24)

Two gold coins with the words city of chicago on them.

(25)

A silver coin with a eagle on it that says pluribus unum.

Figure 8: Grouped caption predictions from TextCaps. The scene text is underlined in descriptions. (1-5) Screen time. (6-10) Language-French/Spanish. (11-15) Language-Arabic/Japanese/Korean/Chinese. (16-20) Scene text in stylized fonts. (21-25) Coin/Curved text.

(1)

A baseball game with a player wearing number 11 on his back.

(2)

A person wearing a number 19 jersey runs across a field.

(3)

A player with the number 7 on her jersey is running to first base.

(4)

A baseball player with the number 34 on his jersey.

(5)

A soccer player with the number 6 on his jersey kicks the ball.

(6)

A baseball player with the blue jays on his jersey is about to hit a ball.

(7)

A small train ride with a sign that says " old timer ".

(8)

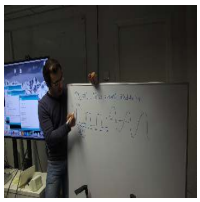
A boat that says " desert belle " on it.

(9)

A bottle of asahi dry black beer next to a glass.

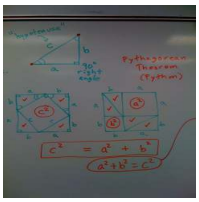
(10)

A display of legos in a store with a neighborhood health sign in the background.



(11)

A man is writing on a white board that says pwm = pulse width modulation.



(12)

A white board with a drawing of pythagorean theorem and a right angle.

(13)

A white board with a drawing of a dinosaur and the words do not erase.

(14)

A white board with welcome minis 2013 written on it.

(15)

A white board with a drawing of a cat and the words i'm smart.

(16)

A road sign that says viaduct lookout (deaths corner) 300 m.

(17)

A drawing of a whale with the words " the largest in the world, even this creature you can see ".

(18)

An advertisement for telbru call rates yet for as low as $ 0. 25 per minute.

(19)

A colorful poster that says to be happy make other people happy.

(20)

A red sign that says ensure you always wear your emergency escape set.

(21)

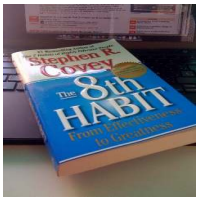
A book is open to the page and the title says " moderni politici sopa i delitti e le pene ".

(22)

A person reading a book that says house on the prairie.

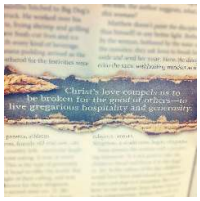
(23)

A book is open to a page titled brisbane in motion moving pictures.



(24)

A book by stephen r. covey titled the 8th habit.



(25)

A page of a book with a quote about christ's love compels us to be broken for the good of others - to live gracious hospitality and generosity.

Figure 9: Grouped caption predictions from TextCaps. (1-5) Numbers on jerseys. (6-10) Occluded scene text. (11-15) Hand-written scene text. (16-20) Long pieces of scene texts. (21-25) Bookpages.

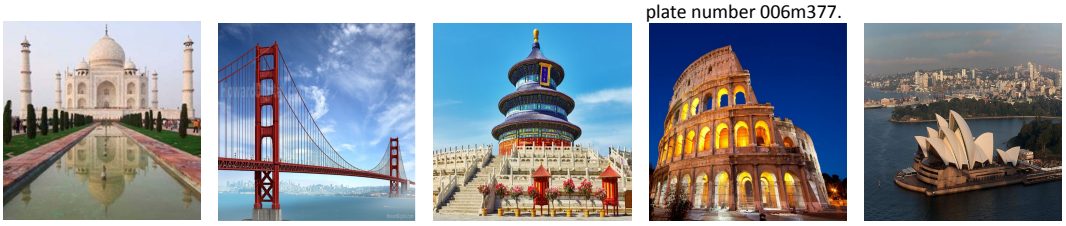
(1) A delta plane is parked on the tarmac at an airport.

(2) A tesla model s car is parked in a dirt road.

(3) A microsoft store in the mall.

(4) A white bentley convertible drives down a road with a license plate number 006m377.

(5) A group of people are outside of a store called xiaomi.



(6) A white marble taj mahal is reflected in a pool.

(7) A golden gate bridge with a city in the background.

(8) A temple of heaven with a blue sign on it.

(9) The colosseum is lit up at night with a fence in the background.

(10) A sydney opera house is on the water with a city in the background.

(11) A bowl of chinese food called mapo tofu.

(12) A bowl of pad thai with a chicken and some sprouts.

(13) A paella is cooked in a pan with a lemon and shrimp.

(14) A beef wellington on a cutting board with a knife.

(15) A plate of caprese salad with tomatoes and basil.

(16) A star wars movie poster with a Darth Vader helmet.

(17) A poster for the movie the matrix.

(18) Bart simpson is shown in a scene from the simpsons.

(19) A marilyn monroe photo with a black background.

(20) Elon Musk in a black jacket.

(21) A red apple with a green label that says fuji 94131.

(22) A yellow and red apple with a honeycrisp sticker on it.

(23) A bunch of orange peppers with a logo that says whole foods market.

(24) A package of whole baby bella mushrooms from food lion.

(25) A bag of mayan sweets premium sweet onions.

Figure 10: Grouped caption predictions on web images generated by TextCaps-fine-tuned GIT. (1-5) Logos. (6-10) Landmarks. (11-15) Foods. (16-20) Characters and celebrities. (21-25) Products.

Table 15: Results on TextCaps (Sidorov et al., 2020). Test set is evaluated by the server. *: the numbers are from Sidorov et al. (2020). B: BLEU@4; M: METEOR; R: ROUGE-L; S: SPICE; C: CIDEr. #: winner entry of the CVPR 2021 workshop challenge. BUTD: Anderson et al. (2018), AoANet: Huang et al. (2019), M4C-Cap.: Hu et al. (2020), Anc.-Cap.: Xu et al. (2021), TAP: Yang et al. (2021c), Human: Sidorov et al. (2020).

Method	Validation set					Test set				
	B	M	R	S	C	B	M	R	S	C
BUTD*	20.1	17.8	42.9	11.7	41.9	14.9	15.2	39.9	8.8	33.8
AoANet*	20.4	18.9	42.9	13.2	42.7	15.9	16.6	40.4	10.5	34.6
M4C-Cap.*	23.3	22.0	46.2	15.6	89.6	18.9	19.8	43.2	12.8	81.0
Anc.-Cap.	24.7	22.5	47.1	15.9	95.5	20.7	20.7	44.6	13.4	87.4
TAP	25.8	23.8	47.9	17.1	109.2	21.9	21.8	45.6	14.6	103.2
TAP#	28.1	24.4	49.3	17.7	119.0	22.9	22.0	46.5	14.6	109.7
Human	-	-	-	-	-	24.4	26.1	47.0	18.8	125.5
GIT_B	24.1	21.1	45.2	15.7	64.9	-	-	-	-	-
GIT_L	30.6	24.6	50.3	18.6	106.3	-	-	-	-	-
GIT	37.0	27.6	54.1	21.1	143.7	33.1	26.2	52.2	19.6	138.2
GIT2	**38.4**	**28.3**	**54.6**	**21.9**	**148.6**	**33.8**	**27.0**	**53.0**	**20.2**	**145.0**

Table 16: Results on VizWiz-Captions. Both test-dev and test-std are evaluated on the server. #: winner entry of 2021 VizWiz Grand Challenge[16]. B@4: BLEU@4; M: METEOR; R: ROUGE-L; C: CIDEr-D; S: SPICE. MTMA: Gong et al. (2021).

Method	test-dev					test-std				
	B@4	M	R	C	S	B@4	M	R	C	S
MTMA#	30.8	23.7	51.9	94.9	19.9	30.7	23.6	51.6	94.1	19.9
GIT_B	25.1	21.7	49.4	71.5	17.8	-	-	-	-	-
GIT_L	29.4	23.5	50.0	96.1	20.1	-	-	-	-	-
GIT	33.1	25.5	53.1	113.1	22.2	33.4	25.6	53.2	114.4	22.3
GIT2	**36.7**	**26.0**	**54.6**	**119.4**	**22.7**	**37.1**	**26.2**	**54.9**	**120.8**	**22.8**

(1-5) contain the descriptions of logos, such as "delta," "tesla," "oneplus," *etc.* GIT also shows the capability of describing landmarks, *e.g.*, "taj mahal," "golden gate bridge," "temple of heaven," "Colosseum," and "Sydney opera house" in (6-10). Samples (11-15) show examples on food images, such as "mapo tofu," "pad thai," "paella," "beef wellington," and "caprese salad." (16-20) provide more examples of recognizing movie/cartoon characters and celebrities. Samples (21-25) describe products based on the tag or packaging information.

VizWiz-Captions. SCST is performed except GIT2, and the full results are shown in Table 16. Fig. 11 visualizes the predictions on random test images. Fig. 12 groups the results by different scenarios. The model can well recognize the banknotes, scene text on bottles/cans, menus, screens, *etc.*, and can better help vision-impaired people in real use cases. The first row (1-5) of Fig. 12 shows the generated captions on blurry images. The second row (6-10) shows images with low image quality or key information partially occluded. For example, GIT reads the scene text "metro," "diet coke," and "mortrin" in samples (6,9,10), and infers the object "toothpaste" and "hard drive" in samples (7,8). Samples (11-15) recognize banknotes in different currencies and denominations. (16-20) describe scene text on bottles and cans, thus providing more informative captions such as the "bacon bits" in (16) and the "nestle water" in (20). GIT also works well in summarizing menus, pages, and screens, as shown in the bottom row (21-25).

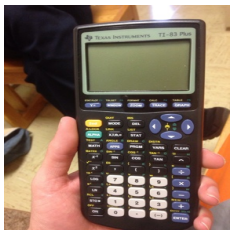
[16]https://vizwiz.org/workshops/2021-workshop/

Pred: a close up of a grey piece of fabric with a seam.

Pred: a close up of a yellow object on a white background.

Pred: the back of a package of food with the cooking instructions.

Pred: the front of a jar of chicken light salad dressing on a kitchen counter.



Pred: a hand holding a black calculator with a screen.

Pred: a container of old fashion hard candies on a table.

Pred: the top of a microwave with buttons on it.

Pred: a black bottle of moisture rich shampoo on a white blanket.

Pred: a grey and black cat with a pink collar laying on a couch.

Pred: a black television screen on a wooden table with a grey object.

Pred: the top of a box of frozen dinner on a wooden table.

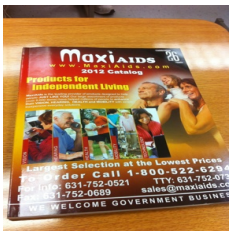
Pred: the top of a box of pretzel bread on a counter.

Pred: the top of a box of healthy choice mediterranean balsamic garlic chicken frozen dinner.

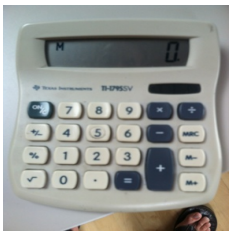
Pred: a blank white piece of paper on a couch.

Pred: the top of a package of canadian bacon.

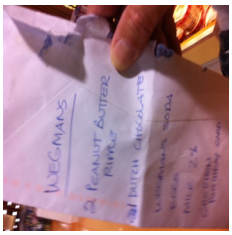
Pred: the top of a green bottle of liquor with a label.



Pred: the front cover of a catalog for 2012 catalog.



Pred: the top of a calculator with white buttons on a table.



Pred: a hand holding a piece of paper with a grocery list.

Pred: the front of a white box for a cell phone.

Pred: a blue sweater with a blue scarf hanging on a hanger.

Pred: the top of a box of fettuccine alfredo.

Pred: the top of a christmas tree with lights on it.

Pred: a bottle of organic apple cider tea sitting on top of a stove.

Pred: a bottle of 14 hands red wine on a table.

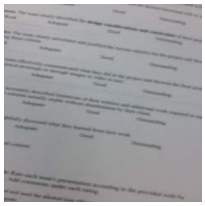
Figure 11: Visualization of our model on random test images of VizWiz-Captions.

(1)

A black olympus device with a green screen and white text.

(2)

A red and yellow box of food with a recipe on the back.

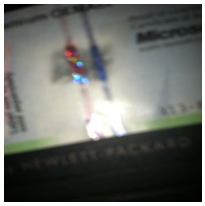


(3)

A piece of paper with a questionnaire on it.

(4)

A white refrigerator with magnets on it.



(5)

A package of a microsoft computer with a red and blue cord.

(6)

A metro card is on a wooden table.

(7)

A box of toothpaste is on a red surface.

(8)

A hard drive with a white label and barcode on it.

(9)

A person is holding a can of diet coke.

(10)

A bottle of motrin is on a counter.

(11)

A close up of a one dollar bill with george washington on the front.

(12)

A person holding a five dollar bill with a picture of abraham lincoln on the front.

(13)

A one dollar bill with the eye of providence on it.

(14)

A person is holding a one dollar bill in their hand.

(15)

A ten pound bill with a picture of queen elizabeth on it.

(16)

A person holding a bottle of bacon bits.

(17)

A person is holding a container of cheese.

(18)

A bottle of night time medicine being held by a person.

(19)

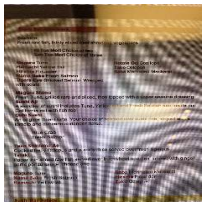
A can of crest top brand diced carrots on a stove top.

(20)

Two bottles of nestle pure life water on a desk.

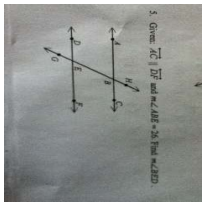
(21)

A menu for a house favorite bbq on a table.



(22)

A menu for sushi appetizers on a computer screen.

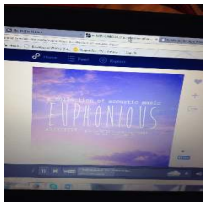


(23)

A math problem with a parallel lines and a line of angles.

(24)

A page from a book about prokaryotic and eukaryotic cells.



(25)

A computer screen with a website for euphonious.

Figure 12: Grouped caption predictions from Vizwiz-Captions. (1-5) Blurry images. (6-10) Low-quality or occluded images. (11-15) Banknotes. (16-20) Bottles and cans. (21-25) Menus, pages, and screens.

Flickr30K. Table 17 shows the full results. SCST is not applied. For the 16/32-shot setting, the batch size is reduced to 16, and the number of iterations is 100.

Table 17: Zero/Few/Full-shot evaluation on Flickr30K with Karpathy split.

Shot	0	16	32	290 (1%)	full
Zhou et al. (2020)	-	-	-	-	68.5
Flamingo	67.2	78.9	75.4	-	-
GIT_B	35.2	65.8	66.4	71.8	81.8
GIT_L	39.2	64.4	68.5	75.4	92.4
GIT	49.6	78.0	80.5	86.6	98.5
GIT2	50.7	79.6	82.0	88.2	98.5

C Results on Visual Question Answering

Except on VizWiz-QA, the number of fine-tuning epochs is 20 and the learning rate is $1e^{-5}$. On VizWiz-QA, the number of epochs is 40 and the learning rate is $2e^{-5}$. The input size is 384 and 576 for intermediate fine-tuning and the final fine-tuning, respectively. No intermediate fine-tuning is conducted for GIT_B and GIT_L. Full results are shown in Table 18. Fig. 14 and Fig. 13 show correct prediction on randomly selected images of VizWiz-VQA and ST-VQA, respectively. Fig. 16 and Fig. 15 show the randomly selected incorrect predictions.

D Results on Video Captioning and Question Answering

Table 21 shows the fine-tuning hyperparameters on video tasks for GIT. Table 19 and Table 20 show the complete results on video captioning and video question answering, respectively. During training, we randomly sample 6 frames with equal interval, and apply the same random crop on these frames. During inference, we uniformly sample 6 frames with center crop.

E Results on Image Classification

On ImageNet-1K (Deng et al., 2009), we map each label to a unique name. Each label belongs to an entry of WordNet hierarchy and is represented with a unique offset, *e.g.*, 2012849. Fig. 17 illustrates the python script to generate a readable unique name given the offset. The model is fine-tuned with 10 epochs and the learning rate is $1e^{-5}$. The batch size is 4096 for the full fine-tuning and 16 for the few-shot setting. No beam search is performed during inference.

Table 22 and Table 23 shows the full results with other model variants.

In the main paper, we demonstrated a decent accuracy of 88.79% top-1 on ImageNet-1k with our generative model in the full fine-tuning setting. As no constraint is on the output, we find that only 13 or (or 0.026%) predictions are outside of the 1K category. Fig. 18 illustrates 10 samples. Although deemed as incorrect, some predictions are reasonable. For example, the prediction of Fig. 18 (e) is *ipad* and is reasonable, although the ground-truth label is *hand-held computer*. These observations also imply that the generation model can quickly adapt to the classification task without pre-defining the vocabulary. Fig. 19 and Fig. 20 show the correct and incorrect predictions, respectively.

Table 18: Results on visual question answering. (a): for VQAv2, approaches are divided according to whether the answer vocabulary is pre-defined (Closed) or not (Open) during inference. The model with closed vocabulary can be a classification model or generation model with constrained outputs, *e.g.*, Wang et al. (2022b); Li et al. (2022b). (b): for TextVQA, Mia$^{\#}$ is the winner entry of TextVQA Challenge 2021 with a fine-tuned T5-3B (Raffel et al., 2020) model. (c): $^{\#\#}$: winner entry of 2021 VizWiz Grand Challenge Workshop. ALBEF: Li et al. (2021a), BLOCK+CNN+W2V: Mishra et al. (2019), BLIP: Li et al. (2022b), CLIP-ViL: Shen et al. (2021), CoCa: Yu et al. (2022), Florence: Yuan et al. (2021), Flamingo: Alayrac et al. (2022), LaAP-Net: Han et al. (2020), LaTr: Biten et al. (2022), mPlug: Li et al. (2022a), M4C: Hu et al. (2020), METER: Dou et al. (2021), Mia: Qiao et al. (2021), OSCAR: Li et al. (2020b), OFA: Wang et al. (2022b), PixelBERT: Huang et al. (2020), UFO: Wang et al. (2021a), UNITER: Chen et al. (2020b), UNIMO: Li et al. (2021c), Visual Parsing: Xue et al. (2021a), VILLA: Gan et al. (2020), VinVL: Zhang et al. (2021a), SA-M4C: Kant et al. (2020), SMA: Gao et al. (2020), SimVLM: Wang et al. (2021b), TAP: Yang et al. (2021c).

Vocabulary	Model	test-dev	test-std
Closed	OSCAR	73.61	73.82
	UNITER	73.82	74.02
	Visual Parsing	74.00	74.17
	PixelBERT	74.45	74.55
	VILLA	74.69	74.87
	UNIMO	75.06	75.27
	ALBEF	75.84	76.04
	VinVL	76.52	76.60
	UFO	76.64	76.76
	CLIP-ViL	76.48	76.70
	METER	77.68	77.64
	BLIP	78.25	78.32
	OFA	79.87	80.02
	SimVLM	80.03	80.34
	Florence	80.16	80.36
	mPlug	81.27	81.26
	CoCa	**82.3**	**82.3**
Open	Flamingo (80B)	**82.0**	**82.1**
	GIT_B (0.1B)	72.72	-
	GIT_L (0.3B)	75.51	-
	GIT (0.7B)	78.56	78.81
	GIT2 (5.1B)	81.74	81.92

(a) VQAv2

Model	validation	test
M4C	40.55	40.46
LaAP-Net	41.02	41.41
SA-M4C	45.4	44.6
SMA	44.58	45.51
TAP	54.71	53.97
Flamingo	57.1	54.1
LaTr	61.05	61.60
Mia$^{\#}$	-	**73.67**
GIT_B	18.81	-
GIT_L	37.47	-
GIT	59.93	59.75
GIT2	68.38	67.27

(b) TextVQA

Model	test-dev	test
Liu et al. (2021)$^{\#\#}$	61.8	60.6
Flamingo	65.7	65.4
GIT_B	54.6	-
GIT_L	62.5	-
GIT	68.0	67.5
GIT2	**70.97**	**70.1**

(c) VizWiz-QA

Model	Val Acc.	Val ANLS	Test ANLS
M4C	38.1	47.2	46.2
LaAP-Net	39.7	49.7	48.5
SA-M4C	42.2	51.2	50.4
TAP	50.8	59.8	59.7
LaTr	61.64	70.2	69.6
GIT_B	14.7	20.7	-
GIT_L	32.3	44.6	-
GIT	59.2	69.1	69.6
GIT2	**66.6**	**75.1**	**75.8**

(d) ST-VQA

model	val	test
BLOCK+CNN+W2V	-	48.3
M4C	63.5	63.9
LaAP-Net	63.8	64.1
LaTr	67.5	67.9
GIT_B	57.3	57.5
GIT_L	62.4	62.9
GIT	67.8	68.1
GIT2	**69.9**	**70.3**

(e) OCR-VQA

Q: What is the number on the bus?

Pred: 12

Q: What is this phone manufacturer's name?

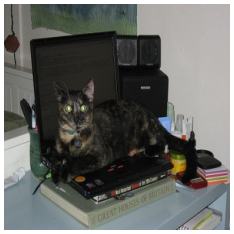
Pred: htc

Q: What is the number on the train?

Pred: 126

Q: What letters are printed on the white sign?

Pred: jef

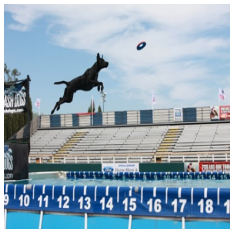


Q: How many women are featured in the black book

Pred: 100

Q: What animal is pictured?

Pred: dog



Q: What car company is an official sponsor?

Pred: ford

Q: What type of keyboard is pictured?

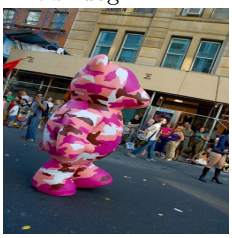
Pred: nord

Q: What brand name is on the bag?

Pred: knorr

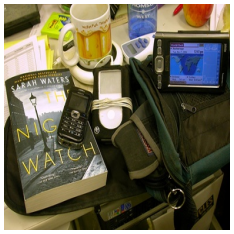
Q: What is the brand name of the mobile phone?

Pred: nokia



Q: What type of booth is behind the people?

Pred: phone



Q: What is the person's first name at the top of the book?

Pred: sarah

Q: WHICH BRAND IS IT

Pred: kroger

Q: what is written in blue color?

Pred: inreach

Q: what are the numbers on train

Pred: 378 136

Q: What is the brand of train?

Pred: dolgoch

Q: What does the bottom sign say?

Pred: cherry st

Q: What is written in white on the red-orange line on the side of this vehicle?

Pred: ambulanza

Q: What is the destination of the white bus?

Pred: crosstown

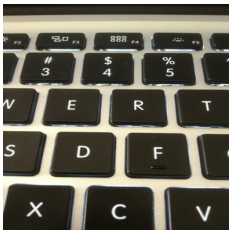
Q: what food is shown in this picture?

Pred: cottage pie

Figure 13: Correct predictions on random validation images of ST-VQA.

Q: What is this?
Pred: unanswerable

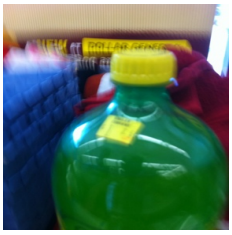
Q: What is this?
Pred: calculator



Q: What is this?
Pred: keyboard

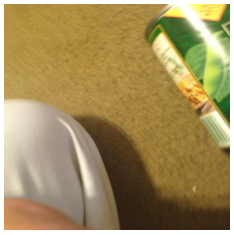
Q: Is this valium?
Pred: unanswerable

Q: What are these?
Pred: traffic cones



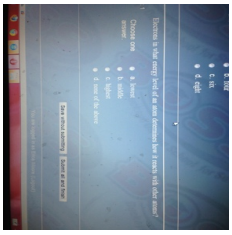
Q: In this bottle.

Pred: unanswerable



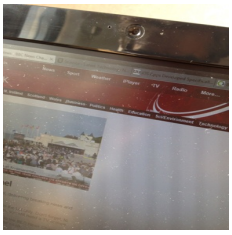
Q: What is this Kane?
Pred: unanswerable

Q: What color is this?
Pred: grey

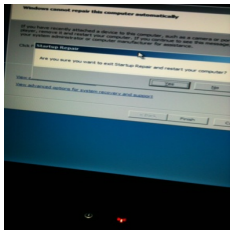


Q: Help with question.
Pred: unanswerable

Q: What is in this tin?
Pred: corn



Q: Who is this picture?
Pred: unanswerable



Q: What's on the screen?
Pred: startup repair

Q: What about the scenery?
Pred: unanswerable

Q: What does my screen say?
Pred: unsuitable

Q: What color is the castle?
Pred: unanswerable

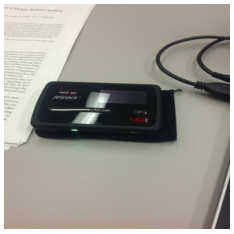
Q: What color is this shirt?

Pred: unsuitable

Q: What is the expiration date?
Pred: unanswerable

Q: Which country does this belong to?
Pred: unanswerable

Q: Can you tell me what flavor this is?
Pred: diet dr pepper



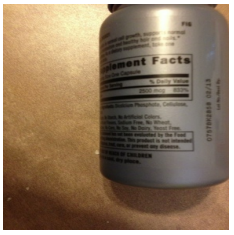
Q: what company is this my hotspot from?
Pred: verizon

Q: What flavor of take five gum is this?

Pred: unanswerable

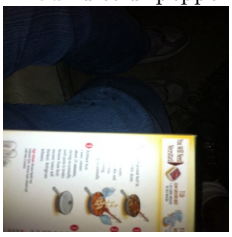
Q: Could you tell me what brand this is? Thank you.

Pred: grace



Q: Is the label showing? And what is in this bottle?

Pred: unanswerable



Q: How much milk and water do I add to my hamburger helper?
Pred: unsuitable

Q: Who is the fifth president and who is on the six dollar bill?

Pred: abraham lincoln

Figure 14: Visualization of correct predictions for the validation set on VizWiz-VQA.

Q: what are the numbers on the track on the cake?
Pred: 13
GT: 123

Q: What sale is advertised for the store in the image?
Pred: sale
GT: 40%

Q: How does one get a dial tone?
Pred: 6
GT: wait

Q: What numbers are on the plane?
Pred: 25
GT: n334sw

Q: What number is the plane?
Pred: n2889sa
GT: n288sa

Q: What can you get 6 of for $5?
Pred: $ 5
GT: donuts

Q: What word is on the third line of the sign?
Pred: bicycle
GT: parking

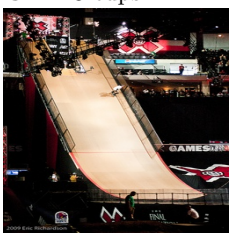
Q: How many cups in 3 gallons?
Pred: 48
GT: 48 cups

Q: What flavor is the ketchup?
Pred: texas
GT: original

Q: What is the weight in ounces?
Pred: 425g
GT: 15, 15 oz.

Q: What airline and gate number?
Pred: 7
GT: delta c7

Q: Which word is shown above the man with the white hat?
Pred: lancia
GT: bordeaux



Q: What restaurant is advertised at the bottom of this picture?
Pred: games
GT: taco bell

Q: WHAT IS WRITTEN ON THE WALL?
Pred: allews
GT: dallus, allus

Q: WHAT IS THAT
Pred: ancona computer monitor
GT: computer monitor

Q: What is written in this picture?
Pred: windows xp
GT: microsoft windows

Q: What is the Brand name?
Pred: celestial
GT: celestial seasonings

Q: What is the name displayed on the board?
Pred: loews theatre
GT: loew's paradise theatre

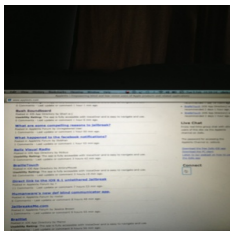
Q: What are the license numbers on white motor bike to the left?
Pred: 61 - 12
GT: 67-n9 67 1024 1024, 67-n9 1024

Q: Where could this product be purchased online?
Pred: locus
GT: www.shoplocus.com, shoplocus.com

Figure 15: Incorrect predictions on random validation images of ST-VQA.

Q: What scent is this?
Pred: lavender

GT: french lavendar, french lavender

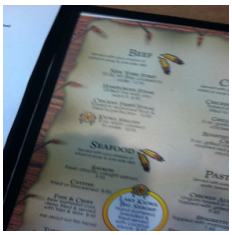


Q: What's on the screen?
Pred: unsuitable

GT: web links, web search results, unsuitable image, search results, words, search

Q: What is this question?
Pred: cleaning brush

GT: pipe cleaner, i dont know, dish cleaner, lint brush, unanswerable, brush, bottle cleaner



Q: Specials on this menu.
Pred: beef

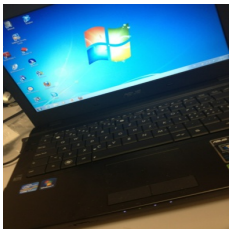
GT: shrimp, unsuitable image, dont see any, unanswerable, beef seafood

Q: How much fat does it contain?
Pred: 180

GT: 12%, g, 8 grams, 8 g, g per serving 3 servings per container

Q: What kind of can food is this?
Pred: unsuitable
GT: baby corn, vegetables, bamboo shoots, salmon, soup, noodles, unanswerable

Q: What is the title of the book?
Pred: unanswerable
GT: auf die lange, auf die lange kommt es, auf die länge kommt es



Q: what can i hear on the screen.
Pred: windows logo
GT: nothing, desktop screen, computer screen to login, just shows normal desk top home screen, unanswerable, windows desktop

Q: What kind of dinner bowl is this?
Pred: unanswerable
GT: meatballs adn sausage marinara, meatballs sausage marinara, food item, cup, ceramic

Q: Please just describe what this is.
Pred: candy
GT: candy bag, unsuitable, package, candy in clear plastic bag, bag candy, bag candy setting on kitchen table

Figure 16: Visualization of incorrect predictions for the validation set on VizWiz-VQA.

```python
from nltk.corpus import wordnet as wn
def get_name(offset):
    white_list = {
      2012849: 'crane bird',
      3126707: 'crane machine',
      2113186: 'cardigan dog',
      2963159: 'cardigan jacket',
      3710637: 'maillot tights',
      3710721: 'maillot bathing suit',
    }
    if offset in white_list:
        return white_list[offset]
    name = wn.synset_from_pos_and_offset('n', offset).name()
    return name[:-5].replace('_', ' ')
```

Figure 17: Python script to generate a unique name for each offset in ImageNet-1K categories.

Table 19: Results on video captioning. ^{E}: model ensemble; ^{T}: with the subtitle as additional input. YouCook2 is on the validation set. ActBERT: Zhu & Yang (2020), CLIP4Caption++: Tang et al. (2021), Flamingo: Alayrac et al. (2022), HERO: Li et al. (2020a), MMT: Lei et al. (2020), MV-GPT: Seo et al. (2022), SibNet: Liu et al. (2020b), OA-BTG: Zhang & Peng (2019), OpenBook: Zhang et al. (2021b), ORG-TRL: Zhang et al. (2020), PMI-CAP: Chen et al. (2020a), POS+CG: Wang et al. (2019a), UniVL: Luo et al. (2020), VaTeX: Wang et al. (2019b), VALUE: Li et al. (2021b), VideoBERT: Sun et al. (2019), Support-set: Patrick et al. (2021), STG-KD: Pan et al. (2020), SwinBERT: Lin et al. (2021a), X-L.+T.: Zhu et al. (2019).

Method	B@4	M	R	C
SibNet	54.2	34.8	71.7	88.2
POS+CG	52.5	34.1	71.3	88.7
OA-BTG	56.9	36.2	-	90.6
STG-KD	52.2	36.9	73.9	93.0
PMI-CAP	54.6	36.4	-	95.1
ORG-TRL	54.3	36.4	73.9	95.2
SwinBERT	58.2	41.3	77.5	120.6
GIT_B	69.3	44.5	81.4	142.6
GIT_L	75.8	48.7	85.5	162.9
GIT	79.5	51.1	87.3	180.2
GIT2	**82.2**	**52.3**	**88.7**	**185.4**

(a) MSVD

Method	B@4	M	R	C
STG-KD	40.5	28.3	60.9	47.1
Support-set	38.9	28.2	59.8	48.6
PMI-CAP	42.1	28.7	-	49.4
ORG-TRL	43.6	28.8	62.1	50.9
OpenBook	33.9	23.7	50.2	52.9
SwinBERT	41.9	29.9	62.1	53.8
MV-GPT^{T}	48.9	38.7	64.0	60
GIT_B	46.6	29.6	63.2	57.8
GIT_L	48.7	30.9	64.9	64.1
GIT	53.8	32.9	67.7	73.9
GIT2	**54.8**	**33.1**	**68.2**	**75.9**

(b) MSRVTT

Method	B@4	M	R	C
VideoBERT	4.3	11.9	-	55.0
ActBERT	5.4	13.3	-	65.0
SwinBERT	9.0	15.6	37.3	109.0
Flamingo	-	-	-	118.6
VALUE^{T}	12.4	18.8	40.4	130.3
UniVL^{T}	17.4	22.4	46.5	181
MV-GPT^{T}	**21.9**	**27.1**	**49.4**	**221**
GIT_B	5.8	12.2	31.5	80.3
GIT_L	7.5	14.4	34.9	98.3
GIT	10.3	17.3	39.8	129.8
GIT2	9.4	15.6	37.5	131.2

(c) YouCook2

Method	B@4	R	M	C
VaTeX	28.4	47.0	21.7	45.1
OpenBook	33.9	50.2	23.7	57.5
VALUE^{T}	-	-	-	58.1
SwinBERT	38.7	53.2	26.2	73.0
CLIP4Caption++^{ET}	40.6	54.5	-	85.7
GIT_B	37.9	51.9	24.4	60.0
GIT_L	41.6	54.3	26.2	72.5
GIT	41.6	55.4	28.1	91.5
GIT2	**42.7**	**56.5**	**28.8**	**94.5**

(d) VATEX public test

Method	C
X-L.+T.^{E}	81.4
Flamingo	84.2
CLIP4Caption++^{ET}	86.5
GIT	93.8
GIT2	**96.6**

(e) VATEX private test

Method	B@4	R	M	C
MMT^{T}	10.8	32.8	16.9	45.3
HERO^{T}	12.3	34.1	17.6	49.9
VALUE^{T}	11.6	33.9	17.6	50.5
SwinBERT	14.5	36.1	18.5	55.4
CLIP4Caption++^{ET}	15.0	36.9	-	66.0
GIT_B	13.0	33.2	16.6	47.3
GIT_L	14.9	35.4	18.0	55.7
GIT	16.2	36.7	18.9	63.0
GIT2	**16.9**	**37.2**	**19.4**	**66.1**

(f) TVC Lei et al. (2020) validation

Method	C
CLIP4Caption++^{T}	59.41
CLIP4Caption++^{ET}	64.49
GIT	61.19
GIT2	**65.02**

(g) TVC private test

Table 20: Results on video question answering. All are open-ended question answering tasks. All-in-one: Wang et al. (2022a), ClipBERT: Lei et al. (2021), CoMVT: Seo et al. (2021), Flamingo: Alayrac et al. (2022), JustAsk: Yang et al. (2021a), MERLOT: Zellers et al. (2021), MV-GPT: Seo et al. (2022), QueST: Jiang et al. (2020), HCRN: Le et al. (2021), VIOLET: Fu et al. (2021).

Method	Accuracy
QueST	34.6
HCRN	36.1
CoMVT	42.6
JustAsk	46.3
VIOLET	47.9
All-in-one	48.3
GIT_B	51.2
GIT_L	55.1
GIT	56.8
GIT2	**58.2**

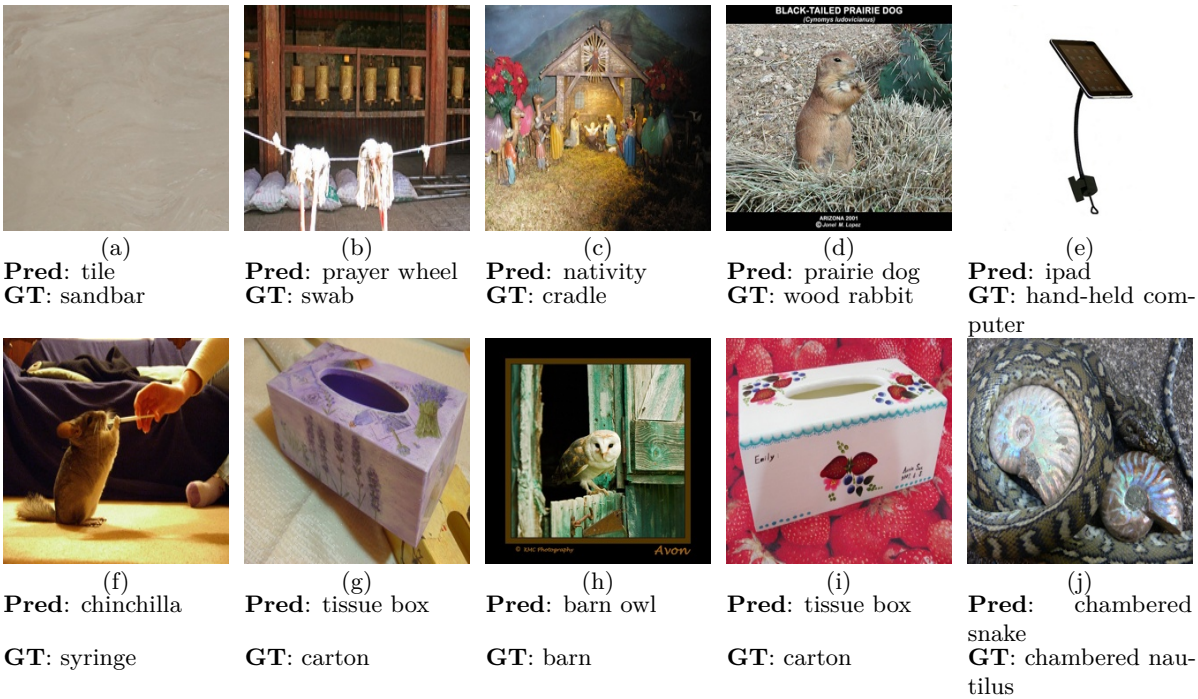
(a) MSVD-QA

Method	Accuracy
JustAsk	41.5
MV-GPT	41.7
MERLOT	43.1
VIOLET	43.9
All-in-one	46.8
Flamingo	**47.4**
GIT_B	41.0
GIT_L	42.7
GIT	43.2
GIT2	45.6

(b) MSRVTT-QA

Method	Accuracy
HCRN	55.9
QueST	59.7
ClipBERT	60.3
All-in-one	66.3
VIOLET	68.9
MERLOT	69.5
GIT_B	69.1
GIT_L	71.9
GIT	72.8
GIT2	**74.9**

(c) TGIF-Frame



(a) **Pred**: tile **GT**: sandbar

(b) **Pred**: prayer wheel **GT**: swab

(c) **Pred**: nativity **GT**: cradle

(d) **Pred**: prairie dog **GT**: wood rabbit

(e) **Pred**: ipad **GT**: hand-held computer

(f) **Pred**: chinchilla **GT**: syringe

(g) **Pred**: tissue box **GT**: carton

(h) **Pred**: barn owl **GT**: barn

(i) **Pred**: tissue box **GT**: carton

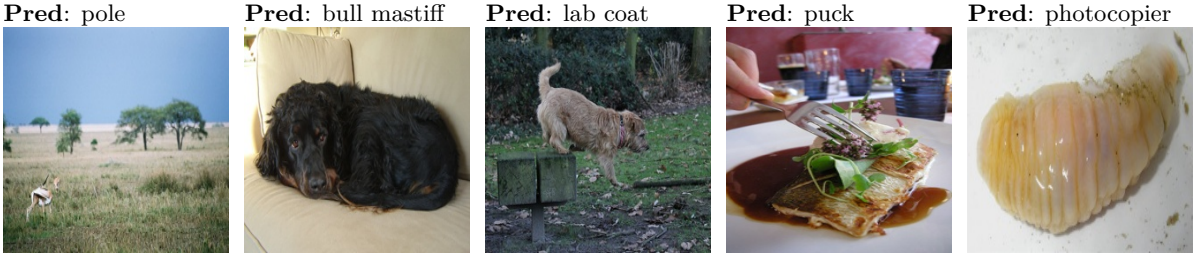
(j) **Pred**: chambered snake **GT**: chambered nautilus

Figure 18: Image samples on which the prediction of our GIT is out of the 1K categories with the whitespace removed on ImageNet-1K.

Pred: wooden spoon | **Pred**: wall clock | **Pred**: puck | **Pred**: piggy bank | **Pred**: steam locomotive

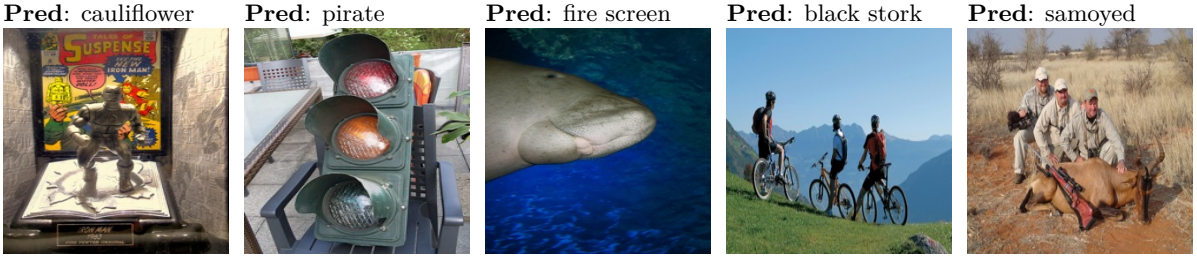
Pred: minibus | **Pred**: ice cream | **Pred**: king penguin | **Pred**: grey fox | **Pred**: grasshopper

Pred: pole | **Pred**: bull mastiff | **Pred**: lab coat | **Pred**: puck | **Pred**: photocopier



Pred: gazelle | **Pred**: gordon setter | **Pred**: norfolk terrier | **Pred**: plate | **Pred**: sea cucumber

Pred: cauliflower | **Pred**: pirate | **Pred**: fire screen | **Pred**: black stork | **Pred**: samoyed



Pred: comic book | **Pred**: traffic light | **Pred**: dugong | **Pred**: mountain bike | **Pred**: hartebeest

Figure 19: Visualization of correct predictions of our generative model on ImageNet-1K.

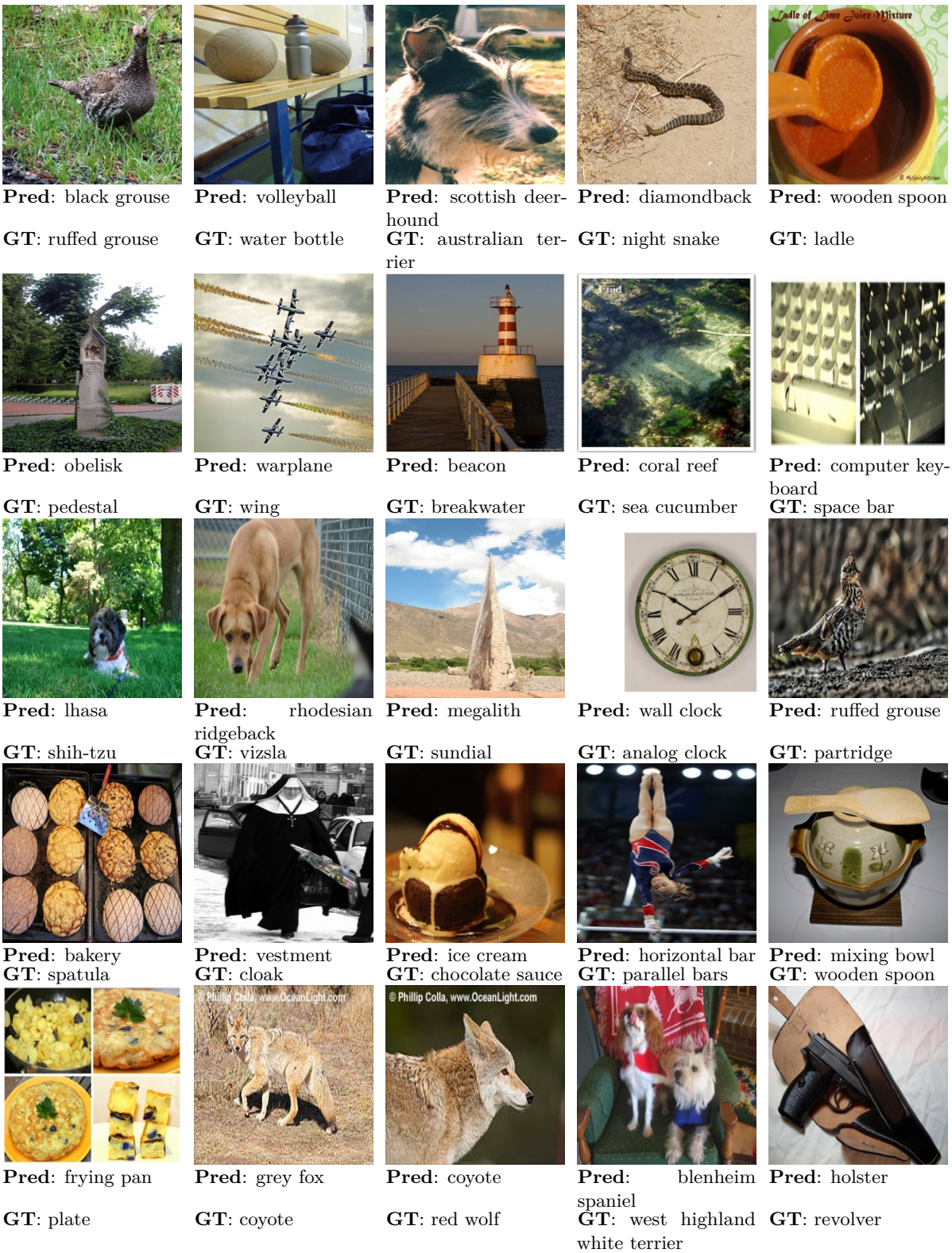


Figure 20: Visualization of incorrect predictions of our generative model on ImageNet-1K.

Table 21: Fine-tuning epochs and learning rate of GIT on video tasks. SCST (Rennie et al., 2017) is performed for VATEX with the same hyperprameters.

	Video Captioning					Video Question Answering		
	MSVD	MSRVTT	YouCook2	VATEX	TVC	MSVD	MSRVTT	TGIF-Frame
epochs	10	20	20	10	20	40	20	20
learning rate	$1e^{-6}$	$2.5e^{-6}$	$1e^{-5}$	$2.5e^{-6}$	$5e^{-6}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$

Table 22: Results on ImageNet-1k classification task. Our approach takes the class name as the caption and predict the label in an auto-regressive way without pre-defining the vocabulary.

Vocabulary	Method	Top-1
Closed	ALIGN (Jia et al., 2021)	88.64
	Florence (Yuan et al., 2021)	90.05
	CoCa (Yu et al., 2022)	**91.0**
Open	GIT_B	78.86
	GIT_L	84.05
	GIT	88.79
	GIT2	89.22

F Results on Scene Text Recognition

Table 24 shows the performance on six individual evaluation sets. Fig. 22 shows the visualization samples with our TextCaps-fine-tuned GIT (denoted as $\text{GIT}_\text{TextCaps}$) and with the MJ+ST-fine-tuned GIT (denoted as GIT_MJSJ). For scene text recognition, we resize the image with the longer edge to 384 and pad the image to a square. Visually, $\text{GIT}_\text{TextCaps}$ can well recognize the scene text, almost as good as GIT_MJSJ, but in the natural language form. GIT_MJSJ can adapt well to the task and predict a clean result of the scene text. Fig. 22 shows the visualizations on all six experimented benchmarks, *i.e.*, IC13 (Karatzas et al., 2013), SVT (Wang et al., 2011), IIIT (Mishra et al., 2012), IC15 (Karatzas et al., 2015), SVTP (Phan et al., 2013), CUTE (Risnumawan et al., 2014) from the top to the bottom row, respectively. GIT performs especially well on testing images visually similar to natural images, such as the CUTE dataset shown in the bottom row. Quantitatively, GIT achieves an even larger performance improvement of 3.9% absolute accuracy on Irregular-Text CUTE80.

We also finetune the pretrained GIT on the TextOCR (Singh et al., 2021) benchmarks. As the test annotations are not publicly available, we evaluate the performance on the validation set and achieve 81.27% accuracy.

G Analysis

G.1 Model and data scaling

In the main paper, we present the impact of scaling on COCO, TextCaps and VizWiz-QA. Fig. 21 shows results on other tasks. On scene-text-related QA tasks (a) and video captioning (d)/(e)/(f), both larger model sizes and more pre-training data boost the performance significantly. For VQAv2 (b), the 0.8B data help little or even worsen the performance slightly. The task data (Goyal et al., 2017) are from COCO, and the first 20M image-text pairs are more similar to COCO images than the majority of the web crawled 0.8B data. This may indicate the first 20M image-text pairs are enough for VQAv2. For video QA (c), the improvement on more pre-training data is mild. The reason might be the domain gap between the image-text pairs and the video-question-answer triplets, which reduces the benefit of more image-text data.

Table 23: Zero/Few-shot evaluation on ImageNet with 3 metrics. *equal*: the unrestricted prediction should be exactly matched to the ground-truth. *in*: the unrestricted prediction should contain the ground-truth label name. *voc-prior*: the vocabulary is pre-defined as a prior. For our GIT, a trie structure is constructed motivated from Wang et al. (2022b) to limit the candidate tokens during each token prediction, such that the predicted result is guaranteed to be within the vocabulary.

	Zero-shot			1-shot per class			5-shot per class		
Accuracy type	equal	in	voc-prior	equal	in	voc-prior	equal	in	voc-prior
Flamingo	-	-	-	-	-	71.7	-	-	77.3
GIT_B	0	11.49	9.34	29.8	30.97	35.41	51.99	52.39	53.4
GIT_L	0	15.28	10.35	43.24	45.75	53.06	68.35	68.85	69.91
GIT	1.93	40.88	33.48	64.54	66.76	72.45	79.79	80.15	80.95
GIT2	1.91	41.92	40.57	67.01	69.07	74.93	80.06	80.57	82.29

Table 24: Results on scene text recognition. MJ and ST indicate the MJSynth (MJ) (Jaderberg et al., 2014; 2016) and SynthText (ST) (Gupta et al., 2016) datasets used for training scene text recognition models. SAM: Liao et al. (2019), Ro.Scanner: Yue et al. (2020) SRN: Yu et al. (2020) ABINet: Fang et al. (2021a) S-GTR: He et al. (2022b) MaskOCR: Lyu et al. (2022)

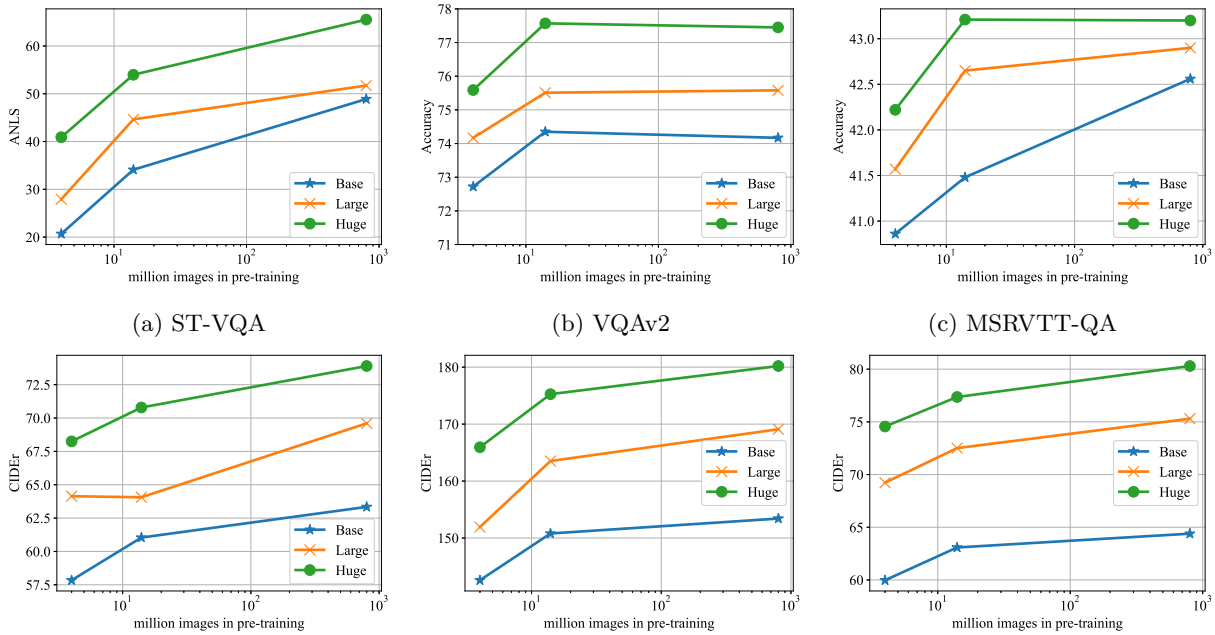
Method	Fine-tuning Data	Regular Text			Irregular Text			Average
		IC13	SVT	IIIT	IC15	SVTP	CUTE	
SAM (Liao et al., 2019)	MJ+ST	95.3	90.6	93.9	77.3	82.2	87.8	87.8
Ro.Scanner (Yue et al., 2020)	MJ+ST	94.8	88.1	95.3	77.1	79.5	90.3	87.5
SRN (Yu et al., 2020)	MJ+ST	95.5	91.5	94.8	82.7	85.1	87.8	89.6
ABINet (Fang et al., 2021a)	MJ+ST	97.4	93.5	96.2	86.0	89.3	89.2	91.9
S-GTR (He et al., 2022b)	MJ+ST	96.8	94.1	95.8	84.6	87.9	92.3	91.9
MaskOCR (Lyu et al., 2022)	MJ+ST	**97.8**	94.1	96.5	**88.7**	90.2	92.7	93.8
GIT	TextCaps	94.2	91.5	92.9	78.2	87.1	95.5	89.9
	MJ+ST	97.3	95.2	95.3	83.7	89.9	96.2	92.9
GIT2	MJ+ST	**97.8**	**95.5**	**97.6**	85.6	**91.3**	**99.0**	**94.5**

G.2 Cross-attention-based decoder

We concatenate the representations of the image and the text as the input to the transformer. An alternative way is to use a cross-attention module to incorporate the image representations, as in Alayrac et al. (2022); Li et al. (2022b). The former allows the image tokens to attend each other, which may refine the representation for better performance; while the latter isolates each image token. However, the former uses a shared set of projections for both the image tokens and the text tokens, while the latter uses separate projection matrices. A shared projection may be hard to learn effectively. Table 25 shows the comparison with different sets of pre-training image-text pairs. With smaller dataset, the latter with cross-attention outperforms, while with large-scale data, the former wins. A plausible explanation is that with more pre-training data, the parameters are well optimized such that the shared projection can adapt to both the image and the text

Table 25: Comparison between pure self-attention-based decoder and the cross-attention-based decoder under different amounts of pre-training data. No SCST is applied on captioning. No intermediate fine-tuning is applied for VQA.

		Captioning				Visual Question Answering		
data	Cross-Att.	COCO	nocaps	TextCaps	VizWiz	ST-VQA	TextVQA	VizWiz
0.8B	w/o	**144.2**	**120.3**	**143.7**	**107.2**	**65.3**	**58.5**	**59.1**
	w/	143.2	118.2	139.3	103.0	63.1	55.6	58.9
10M	w/o	**139.1**	75.4	92.7	**89.3**	40.9	33.0	51.8
	w/	138.1	**86.2**	**93.9**	88.5	**42.7**	**34.7**	**54.8**



(a) ST-VQA (b) VQAv2 (c) MSRVTT-QA

(d) MSRVTT (e) MSVD (f) VATEX

Figure 21: Performance with different pre-training data scales and different model sizes.

Table 26: Ablation study of different initializations for the text decoder. The model structure and the pretraining dataset follow GIT_B, except that the decoder transformer follows BERT_B (Devlin et al., 2018) for Base and BERT_L for Large.

Decoder Size	Initialization	COCO	TextCaps	VizWiz-Captions
Base	Random	127.5	61.7	68.0
	BERT_B	126.9	59.7	67.0
Large	Random	126.6	54.1	65.0
	BERT_L	123.9	52.3	63.6

domains, which mitigates the drawback of the shared parameters. With the self-attention, the image token can be attended with each other for a better representation during decoding. In all experiments, we use the former architecture.

G.3 Initialization of the text decoder

We randomly initialized the decoder. One question is whether the pre-trained weights from text corpus can give a better performance. Table 26 shows the comparison results. We study two decoder sizes: Base (following BERT_B Devlin et al. (2018) with 12 layers) and Large (following BERT_L with 24 layers). As we can see, the text corpus pretrained checkpoint shows no or even worse improvement than the random initialization in both Base and Large sizes. This observation is also in consistent with Table 9 of (Wang et al., 2020). The reason could be that the pretrained weights has no knowledge of the image signals, but this is the key for the VL model. Another observation is that larger decoder also exhibits no benefit, which may be attributed to the fact of more difficulty with a larger transformer. Here, the discussion is based on the fact that the weights are learnable, and thus may not be applicable to the case with frozen parameters, where the initial weight is important, e.g. (Alayrac et al., 2022; Zeng et al., 2022; Xie et al., 2022; Wang et al., 2022c). This is also not applicable to the case where the input is not image features, but the text, e.g. (Lin et al., 2021b).

Table 27: Ablation study with different initialization strategies for the image encoder under the setting of GIT_B. *Supervised* is pretrained with the classification task on ImageNet; *self-supervised* is MAE on ImageNet.

	COCO	TextCaps	VizWiz-Captions
CLIP	131.4	64.9	61.2
Supervised	122.1	47.0	58.3
self-supervised	123.4	44.9	51.6
Random	89.0	36.5	38.1

G.4 Initialization of the image encoder

In our design, the image encoder is initialized from the contrastive pretraining. Table 27 shows the comparison with other initialization methods. The setting follows GIT_B. The image encoder is the base-sized version of ViT, which is initialized from the CLIP model, from the supervised pretraining (classification task on ImageNet), from the self-supervised pretraining (MAE (He et al., 2022a) on ImageNet), or randomly initialized. From the results, we can clearly observe the higher performance with the CLIP pretrained weights. Compared with the supervised/self-supervised, we note that the pretraining datasets for the image encoder are different here due to the availability of these weights. Although it is unclear whether the pretraining dataset is more important than the task or vice versa, we choose the contrastive pretraining as the pretraining dataset is also easy to scale up, and the result is better. For randomly initialization, we observe significant lower performance. The reason could be the small scale of the pretraining set (10M image-text pairs in the set-up of GIT_B). A larger dataset may reduce the gap, but it may require longer training iterations. We leave how to effectively train the model from scratch as future work.

Table 28: Effectiveness of the intermediate fine-tuning on VQA tasks. The gain is large when the target training data scale is small.

	OCR-VQA	VQAv2	TextVQA	VizWiz-QA	ST-VQA
w/o inter. FT	67.6	78.0	59.0	66.6	66.9
w/ inter. FT	67.8	78.6	59.9	68.0	69.1
Δ	0.2	0.6	0.9	1.4	2.2
Train data	166K	122K	22K	20K	17K

G.5 Intermediate fine-tuning on VQA

For VQA, we conduct the intermediate fine-tuning by combining multiple VQA datasets. Table 28 shows the performance comparison with direct fine-tuning without the intermediate fine-tuning. From the results, we can see the intermediate fine-tuning improves the performance for all tasks, and the improvement is more if the target training data scale is small.

G.6 Bias study over gender and skin

Motivated by Zhao et al. (2021), we investigate the bias of our captioning model as follows. Zhao et al. (2021) provides the gender type (male or female) and the skin type (light or dark) for the COCO 2014 test images containing people. As we use the Kapathy split, we first collect the overlapped images between the Kapathy test and the images with well-defined gender and skin annotations in Zhao et al. (2021). Then, we evaluate the performance on the subset images of each category. To measure the bias, we calculate the normalized performance difference (NPD). For example of the gender, we first obtain the metric (e.g. CIDEr) on the images annotated with *male* (C_1) and on the images with *female* (C_2). Then, NPD is $|C_1 - C_2|/(C_1 + C_2)$. With no bias, C_1 equals C_2 and NPD is 0. If the model performs well on one group but totally fails on the other group (metric is 0), NPD is 1. Table 29 shows the result, and we can see that the bias ranges only from 0.7% to 5.3% across all metrics.

Table 29: Normalized performance difference for gender (male vs female) and skin (light vs dark), the annotation of which is provided in Zhao et al. (2021). The value ranges from 0 to 1, and 0 means no bias at all. The lower, the better.

	B4	M	C	S
Gender	0.7%	0.9%	2.0%	2.1%
Skin	4.2%	2.3%	5.3%	2.2%

G.7 Scene text in pre-training data

We show in the main paper that a considerable amount of pre-training samples contain scene text descriptions. Fig. 23 groups the pre-training samples with scene text from CC12M and the downloaded by different scenarios. Samples (1-5) show the associated text which contains the scene text in a natural language way. This is in line with the requirement of scene-text related tasks, *TextCaps*. (6-10) show examples of long pieces of texts. The pre-training samples also describe scene text in stylized fonts (11-15), leading to GIT's ability in robust scene text recognition. (16-20) contain pre-training examples with low-quality images and occluded/blurry/curved scene texts. In addition to the scene text pre-training samples, pre-training datasets also contain descriptions of diverse entities, *e.g.*, the "apple logo" in (21), banknotes in (22), celebrity "Biden" in (23), landmark "empire state building" in (24), and product "beats headphone" in (25). The pre-training data plays a critical role in GIT's capability in scene text description and informative caption generation.

References

Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *CVPR*, 2019.

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *ICCV*, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

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 *CVPR*, 2018.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019.

Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *CVPR*, 2022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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.

David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *AAAI*, 2019.

Shaoxiang Chen, Wenhao Jiang, Wei Liu, and Yu-Gang Jiang. Learning modality interaction for temporal sentence localization and event captioning in videos. In *ECCV*, 2020a.

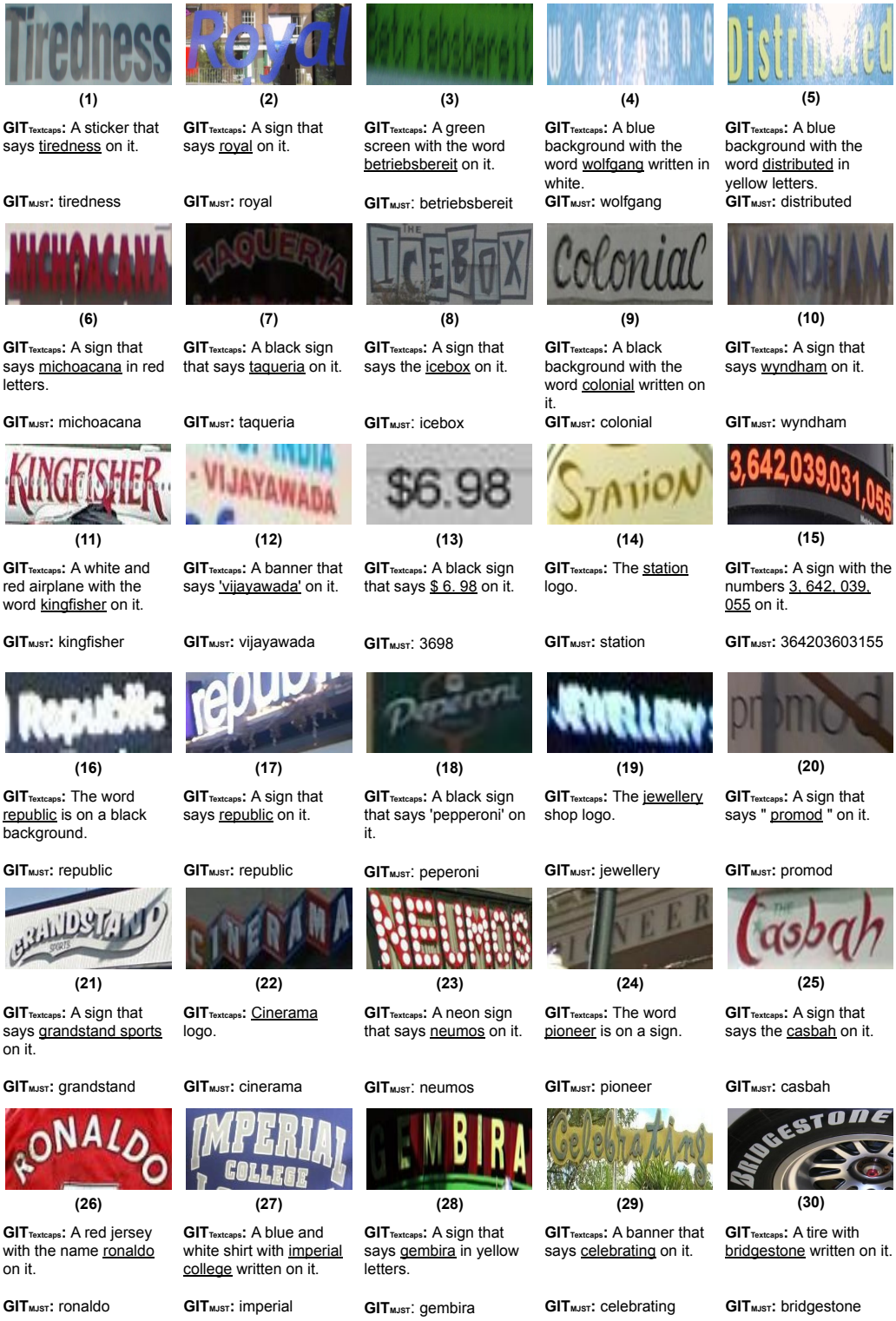


Figure 22: Grouped scene text recognition predictions on all six experimented benchmarks. $\text{GIT}_{\text{Textcaps}}$ and GIT_{MJST} are model variants finetuned with TextCaps caption data (Sidorov et al., 2020), and scene text recognition data MJ+ST (Jaderberg et al., 2014; 2016; Gupta et al., 2016), respectively. (1-5) IC13 (Karatzas et al., 2013). (6-10) SVT (Wang et al., 2011). (11-15) IIIT (Mishra et al., 2012). (16-20) IC15 (Karatzas et al., 2015). (21-25) SVTP (Phan et al., 2013). (26-30) CUTE (Risnumawan et al., 2014).

(1)

The new design natural men's perfume bottle body spray form fine fragrance mist.

(2)

GT Caption: parsonsfield performs at the 2019 celebrate mansfield festival.

(3)

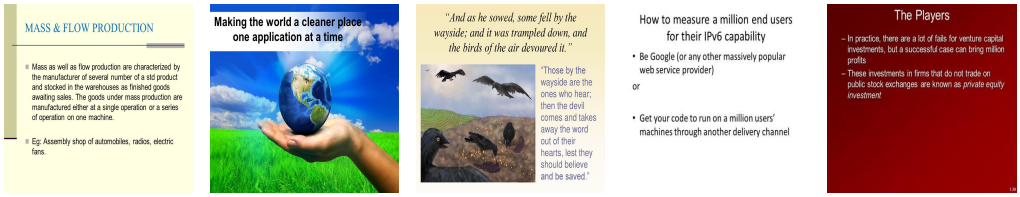
The super mario bros. monopoly game board.

(4)

Review: Kirkland Speyside Single Malt Scotch Whisky 20 Years Old Sherry Cask Finish.

(5)

Mitex Mobi UHF Mobile Radio.



(6)

Mass & flow production mass as well as flow production are characterized by the manufacturer of several number of a std product and stocked in the warehouses as finished goods awaiting sales.

(7)

Making the world a cleaner place one application at a time.

(8)

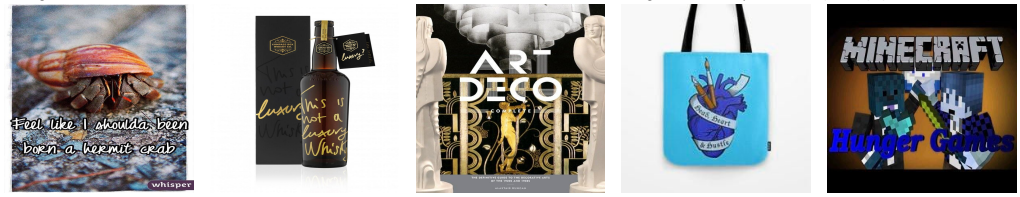
And as he sowed, some fell by the wayside; and it was trampled down, and the birds of the air devoured it.

(9)

How to measure a million end users for their IPv6 capability Be Google (or any other massively popular web service provider) or Get your code to run on a million users\' machines through another delivery channel.

(10)

1-39 The Players \\u2013In practice, there are a lot of fails for venture capital investments, but a successful case can bring million profits \\u2013These investments in firms that do not trade on public stock exchanges are known as private equity investment.



(11)

Feel like i shoulda been born a hermit crab.

(12)

By all counts, this is a luxury whisky. but compass box wants consumers to decide for themselves.

(13)

Art deco, considered by many to be the most glamorous decorative arts style.

(14)

Head, heart & hustle Tote Bag.

(15)

Minecraft Hunger Games w/ Mitch Game #107 Lava and Ice!

(16)

The Hobbit: An Unexpected Journey Poster.

(17)

Affliction-cain-velasquez-ufc-155-walkout-shirt.

(18)

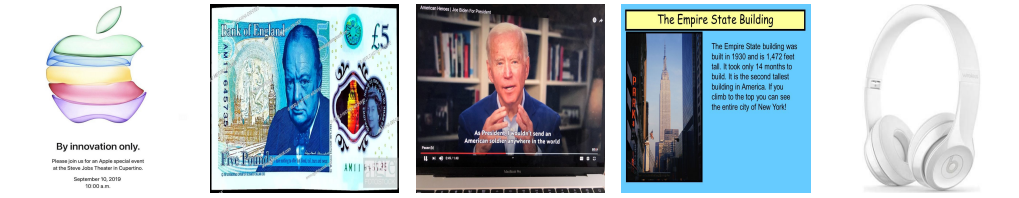
2005 Upper Deck Hall of Fame Essential Enshrinement Silver #CY3 C.Yastrzemski Sleeves.

(19)

St. Louis Cardinals starting pitcher Jake Westbrook throws during the first inning of a spring ……

(20)

KEEP CALM AND CONSTRUCT A BIVOUAC - Personalised Babygrow.



(21)

The flowing translucent apple logo used on the press invitation for the september 10, 2019 event.

(22)

Stock photo: the bank of england £5 note, is a banknote of the pound sterling. ……

(23)

Democratic presidential candidate Biden in a video released from his home on march 31.

(24)

The empire state building the empire state building was built in 1930 and is 1,472 feet tall.

(25)

Bluetooth Beats buy beats by dr dre 3 wireless bluetooth headphones.

Figure 23: Grouped representative pre-training samples from CC12M and the downloaded images. We highlight the informative descriptions in underline. (1-5) Pre-training samples with scene text descriptions. (6-10) Long pieces of scene texts. (11-15) Scene texts in stylized fonts. (16-20) Hard samples of blurry, occluded, or curved scene texts. (21-25) Pre-training data also contains diverse knowledge on logos, banknotes, celebrities, landmarks, products, *etc.*.

Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

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*, 2020b.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Marcella Cornia, Lorenzo Baraldi, Giuseppe Fiameni, and Rita Cucchiara. Universal captioner: Long-tail vision-and-language model training through content-style separation. *arXiv preprint arXiv:2111.12727*, 2021.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*, 2014.

Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. An empirical study of training end-to-end vision-and-language transformers. *arXiv preprint arXiv: 2111.02387*, 2021.

Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, 2021a.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lin Liang, Zhe Gan, Lijuan Wang, Yezhou Yang, and Zicheng Liu. Injecting semantic concepts into end-to-end image captioning. *arXiv preprint arXiv:2112.05230*, 2021b.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. Compressing visual-linguistic model via knowledge distillation. In *ICCV*, 2021c.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET : End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

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.

Chenyu Gao, Qi Zhu, Peng Wang, Hui Li, Yuliang Liu, Anton van den Hengel, and Qi Wu. Structured multimodal attentions for textvqa. *arXiv preprint arXiv:2006.00753*, 2020.

Xuchao Gong, Hongji Zhu, Yongliang Wang, Biaolong Chen, Aixi Zhang, Fangxun Shu, and Si Liu. Multiple transformer mining for vizwiz image caption. In *2021 VizWiz Grand Challenge Workshop*, 2021.

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 *CVPR*, 2017.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pp. 2315–2324, 2016.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. *arXiv preprint arXiv:2002.08565*, 2020.

Wei Han, Hantao Huang, and Tao Han. Finding the evidence: Localization-aware answer prediction for text visual question answering. In *COLING*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022a.

Yue He, Chen Chen, Jing Zhang, Juhua Liu, Fengxiang He, Chaoyue Wang, and Bo Du. Visual semantics allow for textual reasoning better in scene text recognition. In *AAAI*, 2022b.

Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *ICCV*, 2019.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, 2020.

Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. *arXiv preprint arXiv:2111.12233*, 2021a.

Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. VIVO: surpassing human performance in novel object captioning with visual vocabulary pre-training. In *AAAI*, 2021b.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, 2019.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 2016.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

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

Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *AAAI*, 2020.

Yash Kant, Dhruv Batra, Peter Anderson, Alexander G. Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for textvqa. In *ECCV*, 2020.

Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015.

Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

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

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

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

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for multimodal video question answering. *IJCV*, 2021.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 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.

Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022a.

Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021a.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022b.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020a.

Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. Value: A multi-task benchmark for video-and-language understanding evaluation. In *NeurIPS*, 2021b.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *ACL*, 2021c.

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. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020b.

Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *PAMI*, 2019.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, 2004.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. *arXiv preprint arXiv:2111.13196*, 2021a.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

Xudong Lin, Gedas Bertasius, Jue Wang, Shih-Fu Chang, Devi Parikh, and Lorenzo Torresani. Vx2text: End-to-end learning of video-based text generation from multimodal inputs. In *CVPR*, 2021b.

Fen Liu, Guanghui Xu, Qi Wu, Qing Du, Wei Jia, and Mingkui Tan. Cascade reasoning network for text-based visual question answering. In Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (eds.), *ACM MM*, 2020a.

Sheng Liu, Zhou Ren, and Junsong Yuan. Sibnet: Sibling convolutional encoder for video captioning. *IEEE TPAMI*, 2020b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yu Liu, Lianghua Huang, Liuyihang Song, Bin Wang, Yingya Zhang, and Pan Pan. Enhancing textual cues in multi-modal transformers for vqa. In *2021 VizWiz Grand Challenge Workshop*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

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

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

Pengyuan Lyu, Chengquan Zhang, Shanshan Liu, Meina Qiao, Yangliu Xu, Liang Wu, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Maskocr: Text recognition with masked encoder-decoder pretraining. *arXiv preprint arXiv:2206.00311*, 2022.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, 2019.

Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011.

Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *CVPR*, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021.

Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, 2013.

Yixuan Qiao, Hao Chen, Jun Wang, Yihao Chen, Xianbin Ye, Ziliang Li, Xianbiao Qi, Peng Gao, and Guotong Xie. Winner team mia at textvqa challenge 2021: Vision-and-language representation learning with pre-trained sequence-to-sequence model. *arXiv preprint arXiv:2106.15332*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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. *J. Mach. Learn. Res.*, 2020.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 2014.

Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. Look before you speak: Visually contextualized utterances. In *CVPR*, 2021.

Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. *arXiv preprint arXiv:2201.08264*, 2022.

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.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. 2021.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

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

Mingkang Tang, Zhanyu Wang, Zhaoyang Zeng, Fengyun Rao, and Dian Li. Clip4caption ++: Multi-clip for video caption. *arXiv preprint arXiv:2110.05204*, 2021.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

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*, 2022a.

Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*, 2019a.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. Minivlm: A smaller and faster vision-language model. *arXiv preprint arXiv:2012.06946*, 2020.

Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. UFO: A unified transformer for vision-language representation learning. *arXiv preprint arXiv:2111.10023*, 2021a.

Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022b.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019b.

Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. *arXiv preprint arXiv:2205.10747*, 2022c.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021b.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yujia Xie, Luowei Zhou, Xiyang Dai, Lu Yuan, Nguyen Bach, Ce Liu, and Michael Zeng. Visual clues: Bridging vision and language foundations for image paragraph captioning. *arXiv preprint arXiv:2206.01843*, 2022.

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.

Guanghui Xu, Shuaicheng Niu, Mingkui Tan, Yucheng Luo, Qing Du, and Qi Wu. Towards accurate text-based image captioning with content diversity exploration. In *CVPR*, 2021.

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

Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing inter-modality: Visual parsing with self-attention for vision-language pre-training. *arXiv preprint arXiv:2106.13488*, 2021a.

Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing inter-modality: Visual parsing with self-attention for vision-language pre-training. In *NeurIPS*, 2021b.

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

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *CVPR*, June 2022a.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. *arXiv preprint arXiv:2111.12085*, 2021b.

Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei A. F. Florêncio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021c.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3081–3089, 2022b.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2014.

Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, 2020.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

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. *arXiv preprint arXiv:2111.11432*, 2021.

Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, 2020.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: multimodal neural script knowledge models. In *NeurIPS*, 2021.

Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, 2019.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. In *CVPR*, 2021a.

Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020.

Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Ying Shan, Bing Li, Ying Deng, and Weiming Hu. Open-book video captioning with retrieve-copy-generate network. In *CVPR*, 2021b.

Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *ICCV*, 2021.

Qi Zheng, Chaoyue Wang, and Dacheng Tao. Syntax-aware action targeting for video captioning. In *CVPR*, 2020.

Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *AAAI*, 2020.

Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

Xinxin Zhu, Longteng Guo, Peng Yao, Shichen Lu, Wei Liu, and Jing Liu. Vatex video captioning challenge 2020: Multi-view features and hybrid reward strategies for video captioning. *arXiv preprint arXiv:1910.11102*, 2019.