GRiT: A Generative Region-to-text Transformer for Object Understanding

Jialian Wu[1,3⋆], Jianfeng Wang[2], Zhengyuan Yang[2], Zhe Gan[4], Zicheng Liu[3], Junsong Yuan[1], and Lijuan Wang[2]

[1] State University of New York at Buffalo
[2] Microsoft
[3] Advanced Micro Devices
[4] Apple

Abstract. This paper presents a Generative RegIon-to-Text transformer, GRiT, for object understanding. The spirit of GRiT is to formulate object understanding as <region, text> pairs, where region locates objects and text describes objects. Specifically, GRiT consists of a visual encoder to extract image features, a foreground object extractor to localize objects, and a text decoder to generate natural language for objects. With the same model architecture, GRiT describes objects via not only simple nouns, but also rich descriptive sentences. We define GRiT as open-set object understanding, as it has no limit on object description output from the model architecture perspective. Experimentally, we apply GRiT to dense captioning and object detection tasks. GRiT achieves superior dense captioning performance (15.5 mAP on Visual Genome) and competitive detection accuracy (60.4 AP on COCO test-dev). Code is available at `https://github.com/JialianW/GRiT`

1 Introduction

Great efforts have been made in object detection task [1, 20, 27, 47]. To answer 'what the object is?', object detection models classify objects among a *closed-set* of object classes, where class names are usually simple nouns. Recent open-vocabulary object detectors [12, 18, 44, 45] exploit fertile vision and language data to enable object detectors to recognize object classes that do not exist in object detection data. Open-vocabulary object detectors, however, still has to first know what object classes are there in inference phase so as to define a closed-set of class name embeddings to achieve object classification. The "open-vocabulary" only means some of the class name embeddings, so-called novel classes, are not associated with object detection bounding box data during training. As illustrated in Fig. 2 (a), these closed-set frameworks behave like performing a multiple-choice question, choosing the most likely answer from a limited number of candidates.

In contrast, humans understand objects in an *open-set* configuration that does not limit the number of object categories and can therefore learn new objects

⋆ Work was done when the author interned at Microsoft.

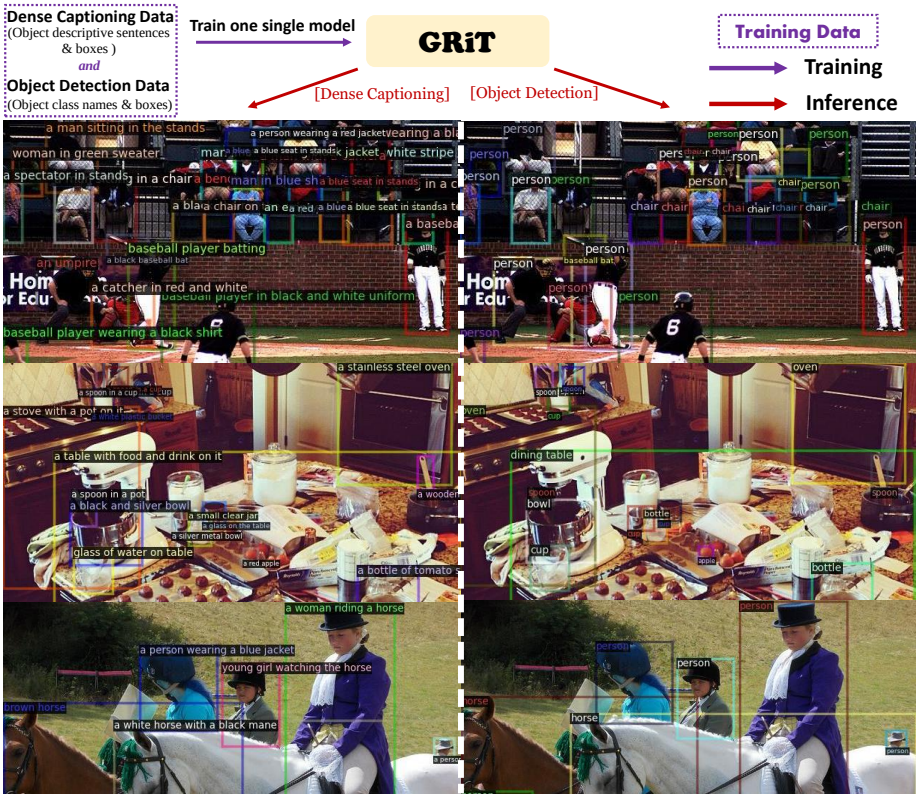

Fig. 1: GRiT's object understanding pipeline with multi-task training. GRiT is an open-set object understanding framework that localizes objects and generates object description texts in *free-form*. The right figures show the predictions from the model trained on object detection data and dense (object) captioning data together. During inference, the trained GRiT can generate either simple class names for object detection task, or rich descriptive sentences for dense captioning task, instructed by two special tokens `[ObjectDet]` and `[DenseCap]`, respectively.

effortlessly. Also, humans perceive auxiliary information associated with the object to improve understanding, *e.g.*, color, shape, and action expressed through adjectives and verbs. Toward human-like object understanding, we propose a Generative RegIon-to-Text transformer, coined as GRiT. GRiT is not classifying objects into categories but generating natural language for objects. It does not need to define a list of categories as shown in Fig. 2 (b), and can provide more information about an object as shown in Fig. 1.

Given an input image, GRiT localizes all presented objects and generates object descriptions for each of them in *free-form*. Specifically, GRiT consists of three main components: a visual encoder, a foreground object extractor, and a text decoder. The visual encoder extracts image features, on which the foreground

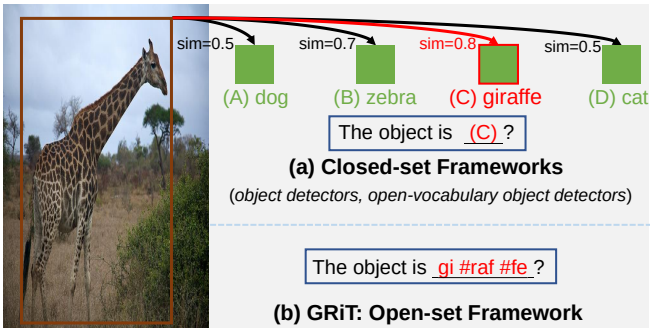

Fig. 2: GRiT *vs.* Closed-set object recognition models. Closed-set approaches classify objects among a predefined set of categories by choosing the class embedding that has the maximum similarity to the given object feature. In contrast, GRiT directly generates text tokens to spell the words that describe the object. In this example, "`giraffe`" is spelled by three text tokens generated by GRiT.

object extractor detects foreground object regions and crops object features. Taking object features as input, the text decoder autoregressively generates text tokens to describe the given object, where each word is tokenized by one or multiple text tokens (*a.k.a.* sub-words) by the WordPiece [28, 34] model. In addition to simple noun category names (*e.g.*, cat, giraffe), GRiT can also generate rich descriptive sentences, providing more information as shown in Fig. 1. In this way, GRiT achieves general object understanding that can unify various region-level tasks into a single framework, *e.g.*, object detection and dense (object) captioning.

GRiT is an *open-set* object understanding framework which means it has unlimited words to describe objects, as any word can be represented by a combination of text tokens. We note that our open-set object understanding is different from the open-vocabulary object detection, where the former is to generate texts for objects while the latter is to classify objects into categories. The open-set object understanding is more universal and more challenging than the closed-set object detection. It is also more friendly as data grows and evolves, as it continuously learns new object concepts without adapting model architecture.

With the same architecture, GRiT can train on the short-description task (*e.g.*, object detection) and the long-description task (*e.g.*, dense captioning) separately or together. Joint training on short- and long-description tasks without any adaptations can confuse the model to generate descriptions for the correct task. To solve this issue, we add a special token `[task]` to control GRiT's text decoder to predict task-specific object descriptions.

In summary, this paper presents a generative and open-set object understanding framework. On Visual Genome dense captioning [16], GRiT obtains 15.5 mAP which surpasses standard (non-LLM) dense captioning models. On COCO object detection [22], GRiT achieves 60.4 AP, which is comparable to the closed-set

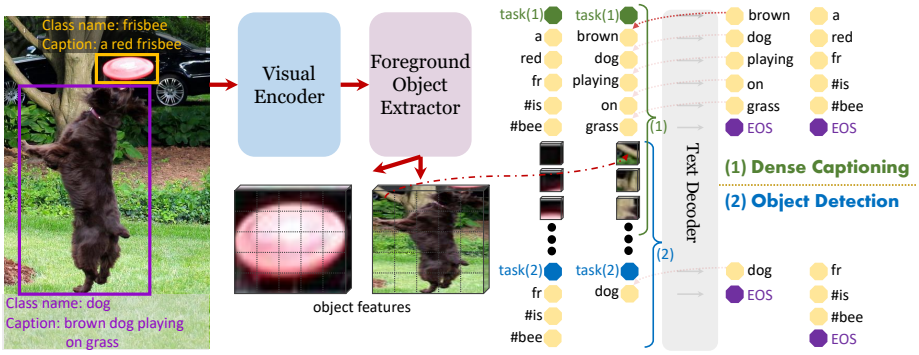

Fig. 3: Overview of GRiT. Given an input image, the visual encoder extracts image features, from which the foreground object extractor predicts object boxes. Object features are derived by cropping image features using object boxes. Taking object features as input, the text decoder autoregressively generates text tokens one-by-one in the task-i style, instructed by a begin token $\texttt{[task]}_i$.

standard object detectors. We hope our work can inspire more future works on this generative object understanding.

2 Related Work

Continuous progress has been made in standard object detection [1–3, 27, 40, 48]. Despite the excellent object localization and classification accuracy, these frameworks are designed to recognize a fixed set of object categories.

Recent open-vocabulary object detection [12, 18, 44, 45] breaks the fixed category limit. The goal is to recognize object categories that are not in object detection datasets, using the knowledge learned from large-scale vision and language data. For example, RegionCLIP [44] exploits the CLIP [26] trained on hundreds of millions of image-text data to match region features with text embeddings, where the knowledge in CLIP helps to annotate region-text pairs not labeled in object detection data. ViLD [12] makes use of the CLIP language embeddings and distills the CLIP vision knowledge into its own visual backbone. More straightforwardly, Detic [45] treats the full image as a whole box and learns directly from the large-scale image labels. These open-vocabulary object detectors can vary the size of the category set and recognize more categories beyond object detection datasets. However, the category set still needs to be closed and predefined by humans in order to construct a contrastive matrix with the object regions and have the model choose a category from it to label the object. Besides, they do not show the ability to generate descriptive sentences.

In this work, we are inspired by the generative image-to-text transformer [14, 33, 38]. The generative methods produce free-form words and sentences to achieve various image understanding tasks like image captioning, question answering, and classification. Our GRiT extends the spirit to region-level object understanding,

aiming to generate object descriptions in free-form with not only class names but also rich descriptive sentences. GRiT inherits the open-set feature of the generative methods, which does not need humans to define a category list and *just spells out* object descriptions on its own.

In contrast to standard dense captioning models [15, 19, 30, 36, 37], GRiT is a general object understanding framework unifying both object detection and dense captioning. In architecture, GRiT is advanced by the simple yet successful generative image-to-text transformer. This enables our state-of-the-art dense captioning performance without the need for complex object relation and context modeling as in previous dense captioning models. Recent multimodal large language models [42, 43] also show excellent performance in captioning objects. Different from LLM-based models, GRiT's language model part is small and is trained from scratch (for the MAE pertaining scheme described in Sec. 4.1).

3 GRiT

3.1 Architecture

As illustrated in Fig. 3, GRiT comprises three major components: a visual encoder, a foreground object extractor, and a text decoder. GRiT is end-to-end in both training and inference.

Visual Encoder. Given an input image, a visual encoder is applied to obtain image features. Our visual encoder consists of a backbone network, and a feature pyramid that is proven helpful for object detection [20,21]. Following the image-to-text transformer, we use ViT [10] as the backbone network for main experiments. Different from image tasks, object understanding favors high-resolution input images, which can consume huge GPU memory in training ViT's self-attention. Therefore, we divide ViT's feature maps into non-overlapped windows with the size of 14×14, and compute self-attention only within the windows following [20]. A relative positional encoding is added during the window self-attention as in [23]. To exchange information across the windows, four evenly selected ViT blocks keep the original ViT self-attention scheme which computes global self-attention across all positions on the feature map. ViT extracts image features throughout in a single scale without hierarchies, *e.g.*, $\frac{1}{16}\times$ image size, which is however incompatible with vanilla FPN [21]. To build feature pyramid on top of ViT, we follow the idea of simple feature pyramid [20] that produces multi-scale features by up/down-sampling from the last feature map of ViT. In this way, we construct five scales of feature maps $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}\}$ for our feature pyramid.

Foreground Object Extractor. On top of the feature pyramid, our foreground object extractor detects foreground objects with bounding boxes and objectness scores. The foreground object extractor borrows the architecture of two-stage object detectors [1, 27, 46], comprised by a proposal generator and an RoI head. The proposal generator produces a large amount of proposal boxes. Then, the RoI head refines the box position and predicts an objectness score indicating the confidence that the box contains a foreground object. Thus, it is a binary classifier

in the RoI head for foreground/background classification, which is different from the multi-category classifier in standard object detectors. Lastly, the foreground object extractor removes highly overlapped boxes by NMS and the boxes with low objectness scores.

Text Decoder. Our text decoder is the core part of GRiT for understanding and describing objects in free-form. The text decoder takes object features as input and generates text tokens to describe the given object. To derive object features, we use object boxes produced by the foreground object extractor to crop image features to a fixed size, *e.g.*, 14×14, and then flatten them into 1D vectors. Object features encode not only object appearance but also image context thanks to the global self-attention blocks in ViT. This is important for dense captioning task where object descriptions may contain context descriptions in other image regions as well, as shown by the example of "young girl watching the horse" in Fig. 1. To convert words into text tokens, we use the BERT's WordPiece tokenizer [9, 28, 34], where any word can be represented by a combination of text tokens from the overall 30,522 tokens of vocabulary. For example, the class name "`giraffe`" in COCO is converted into three text tokens "`gi`", "`#raf`", and "`#fe`".

The text decoder is achieved by a 6-layer transformer following the image-to-text transformer GIT [33]. Our text decoder, equipped with a begin token `[task]`, produces object descriptions by generating text tokens one-by-one in an autoregressive way until an end token `[EOS]`. In each step, object features and previously generated tokens, including the begin token, are concatenated as input to the transformer. Both object features and text tokens are embedded into the same dimensions before feeding into the transformer, where we also add a positional encoding to the text embeddings. A seq2seq attention mask is applied to make sure text tokens are attending to object features and only previous text tokens, and object features are only attending to themselves. At the end of the transformer, a linear layer projects the text embeddings into 30,522 text token logits, and a softmax is applied afterward to yield the score for each text token. The text token with the highest score is kept. To predict the class name "`giraffe`", GRiT needs to consecutively generate the three tokens "`gi`", "`#raf`", and "`#fe`". Thanks to this flexible application of text tokens, GRiT achieves an *open-set* object understanding that can describe whatever we provide in training.

Different tasks may have varied styles of object descriptions. For example, object detection task interprets objects by short class names, while dense captioning task describes objects with rich descriptive sentences including object attributes, quantity, or actions. Jointly training them can confuse the model in inference, not knowing which style of object descriptions it should generate. To solve this issue, we define a set of begin tokens $\{\texttt{[task]}_i\}_{i=1}^{T}$ for jointly training tasks with different styles of object descriptions. T is the number of different styles of tasks. In training, we select $\texttt{[task]}_i$ as the begin token when the object description annotation is from the task-i. In this way, during inference, $\texttt{[task]}_i$ can inform the trained model to generate descriptions in the task-i style.

3.2 Training

The training loss of GRiT consists of two major parts $L = L_o + L_t$, where L_o and L_t are for the foreground object extractor and text decoder, respectively. L_o is the same as the standard object detector loss that includes box losses and classification losses for both the proposal generator and RoI head. L_t is achieved by the language modeling (LM) loss as follows:

$$L_t = \frac{1}{N+1} \sum_{i=1}^{N+1} \text{CE}(y_i, p(y_i|o, y_{0,\ldots,i-1})), \tag{1}$$

where $p(y_i|o, y_{0,\ldots,i-1})$ is the predicted score for the i-th text token given the object features o and previously generated text tokens $y_{0,\ldots,i-1}$. N is the number of text tokens in the given object description. y_0 and y_{N+1} are the begin token and end token, respectively. CE is the cross-entropy loss with a label smoothing of 0.1. Note that the text decoder loss L_t is only imposed on foreground objects predicted by the foreground object extractor.

3.3 Inference

Beam Search. Standard object detectors may yield multiple object class labels for one box to improve performance. To enable a similar mechanism in GRiT, we employ a beam search algorithm in the text decoder, which is commonly used in image captioning. Specifically, we select the top k text tokens in terms of their scores when generating the first token of object description in addition to the begin token, where k is the beam size. The text decoder then continues to decode k object descriptions following these k text tokens. In experiments, we find $k = 3$ is sufficient for object detection on COCO. We do not use beam search for dense captioning.

Object Scoring. GRiT rates object predictions by objectness score from the foreground object extractor and an object description score from the text decoder. Since an object description may contain multiple text tokens, its score is computed by averaging the scores of all text tokens. The final object confidence score is computed by multiplying the square roots of these two scores.

4 Experiments

To evaluate GRiT's general object understanding capability, we experiment on the COCO dataset [22] for object detection task and the Visual Genome (VG) dataset [16] for dense (object) captioning task.

COCO. COCO contains 80 object classes and all class names are nouns. Each class name in COCO is encoded by 1∼3 text tokens. We train on COCO 2017 train and evaluate on COCO 2017 val and 2017 test-dev. **Evaluation Metric:** Object detection is evaluated by COCO box AP and AR. As our approach imposes no hard constraint in generating the class names, we remove boxes whose class names are not in COCO during evaluation.

Visual Genome. We use VG v1.0 train set and test set for training and evaluation. Following the original paper [15], we pre-process VG data to discard object descriptions with more than 15 words and convert symbols into English words, *e.g.*, ° → "`degree`". The pre-processing leaves 77,396 images for the train set and 5,000 images for the test set. There are ∼4 million annotated region descriptions with over 50,000 unique words in the train set. The annotations contain some typographical mistakes like "`tranportation`" → "`transportation`", which we don't perform further processing as GRiT can be fault-tolerant to some extent. Different from COCO object detection, object descriptions in VG have adjectives and verbs in addition to nouns, describing object attributes, actions, etc. **Evaluation Metric:** Similar to object detection metric, dense captioning measures an mAP across a range of thresholds for both localization and description accuracy, following [15]. For localization, it uses box IoU thresholds of .3, .4, .5, .6, .7. For language description, a METEOR score [17] with thresholds of 0, .05, .1, .15, .2, .25 is used. The mAP is averaged by the APs across all pairwise of these two types of thresholds.

Since COCO and VG have inconsistent box annotations as we will discuss in Sec. 4.6, we train GRiT separately on these two datasets in comparison with state-of-the-arts and ablations, in order to fairly compare with other single-task methods and reliably study GRiT's properties.

4.1 Implementation Details

Visual Encoder. We employ ViT-B, ViT-L, and ViT-H [10] as the backbone of the visual encoder unless otherwise specified. The input image patch size is 16×16. A layer-wise learning rate decay [5,20] of 0.7/0.8/0.9 is set for ViT-B/L/H. For feature pyramid, the feature maps of $\{\frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$ scales are constructed in the way of simple feature pyramid [20]. The feature maps of $\{\frac{1}{64}, \frac{1}{128}\}$ scales are built by downsampling from that of $\frac{1}{32}$ scale following [46].

Foreground Object Extractor. We use CenterNet [47] as the proposal generator. It generates 2000 and 256 proposal boxes in training and testing, respectively. The RoI head is achieved by a 3-stage Cascade R-CNN [1]. The object box used by the text decoder is predicted from the last stage. The number of classes of each stage classifier is set to 2, *i.e.*, foreground and background, where our objectness score is computed by averaging the foreground scores of all the three stages. For COCO object detection, soft NMS is applied in testing, and a mask head is added for multi-task training.

Training. In experiments, we explore two pre-training schemes: 1) MAE pre-training: The ViT backbone is initialized from the self-supervised MAE [13] trained on ImageNet-1K [8], while the rest of the model parameters are randomly set; 2) GIT pre-training: The ViT backbone and text decoder are initialized from the pre-trained image-to-text transformer GIT [33] and the rest are randomly set. We use MAE pre-training for most experiments by default unless otherwise specified. For the model initialized from GIT pre-training, we finetune on the downstream tasks by 90k iterations with a training batch size of 32. We find MAE pre-training can alleviate overfitting and benefit from more training epochs

Method	mAP
JIVC [36]	9.31
ImgG [19]	9.25
COCD [19]	9.36
COCG [19]	9.82
CAG-Net [37]	10.51
TDC+ROCSU [30]	11.49
ControlCap [43]	18.2
GRiT_{MAE} (Ours)	15.48
GRiT_{GIT} (Ours)	15.52

Table 1: Comparison with state-of-the-art dense captioning models on Visual Genome. GRiT_{MAE} refers to the model initialized by MAE pre-training scheme. GRiT_{GIT} is initialized from the GIT model that is re-pretrained on CC3M and CC12M [31] datasets removing the VG dataset. Our results are based on ViT-B. Gray color indicates the LLM-based model.

as discussed in [20]. Thus, for the model initialized from MAE pre-training, we increase finetuning iterations to 180k and batch size to 64. We use the AdamW optimizer [24] with a learning rate of 8×10^{-5} and the cosine learning rate decay schedule. In training, the input image size is 1024×1024 pixels resized by the large-scale jittering [11]. The testing image size is 800×1300 pixels.

4.2 Comparison to State-of-the-Arts on VG

We evaluate the dense captioning performance in Table 1. GRiT achieves state-of-the-art performance compared to standard dense captioning models but loses to ControlCap [43] which is an LLM-based model.

4.3 Comparison to State-of-the-Arts on COCO

As shown in Table 2, we compare GRiT with the state-of-the-art object detectors on COCO. Generally, all the methods use the visual backbone pre-trained on ImageNet (IN). To deliver the best performance, some models are also finetuned on extra object detection datasets, *e.g.*, Object365 [29], before finetuning on COCO. The results of the state-of-the-art object detectors are cited from their best model settings. Therefore, some listed models may be achieved in different training and testing configurations than others. For example, GLIPv2 [41] makes use of Object365 plus four object detection datasets [41], image-text datasets CC [31] and SBU [25], and grounding datasets GoldG [41]. DyHead [7] utilizes a larger input image size with 2000 pixels at maximum. CenterNet2 [46] adopts BiFPN [32], DCN [6], and a larger input image size of 1560×1560 pixels. GRiT performs comparably with the state-of-the-art closed-set object detectors, which is remarkable in view of the challenge of our open-set way. It demonstrates GRiT's open-set generative region-to-text model can be a new promising formulation for object detection that is previously solved in the closed-set way.

Model	Backbone	2017 val			2017 test-dev		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Closed-Set framework:							
Deformable DETR [48]	ResNeXt-101	-	-	-	50.1	69.7	54.6
EfficientDet-D7x [32]	EfficientNet-B7	54.4	-	-	55.1	74.3	59.9
CenterNet2 [46]	Res2Net	-	-	-	56.4	74.0	61.6
HTC++ [3, 23]	Swin-L	57.1	-	-	57.7	-	-
DyHead [7]	Swin-L	58.4	-	-	58.7	77.1	64.5
ViT-Adapter-L [4]	ViT-L	58.4	-	-	58.9	-	-
Soft Teacher* [35]	Swin-L	60.1	-	-	-	-	-
ViTDet [20]	ViT-H	60.4	-	-	-	-	-
DINO* [40]	Swin-L	63.1	-	-	63.2	-	-
GLIPv2† [41]	Swin-H	-	-	-	60.6	-	-
Open-Set framework:							
GRiT (Ours)	ViT-B	53.6	71.6	58.2	53.8	71.8	58.7
GRiT (Ours)	ViT-L	56.3	73.8	61.4	56.6	74.5	61.8
GRiT (Ours)	ViT-H	58.8	76.6	64.4	59.0	76.9	64.5
GRiT* (Ours)	ViT-H	60.3	78.1	65.7	60.4	78.1	66.0

Table 2: Comparison with state-of-the-art object detectors on COCO. All results are reported under single-scale testing. The reference results are cited from the best-performing models in their papers. * indicates the model is pretrained on Object365. † indicates GLIPv2 is pretrained on Object365, FourODs, GoldG, and CC15M+SBU.

4.4 Ablation Studies

In this section, we perform ablation experiments on object detection task to study GRiT's properties.

GRiT *vs.* Closed-set Object Detector: GRiT achieves object detection in an *open-set* way, which is more difficult than the closed-set way of standard object detectors. To measure the performance gap between these two strategies, we build a closed-set object detector by replacing GRiT's text decoder with the closed-set multi-category classifier as used in standard object detectors. The rest of the model settings are exactly the same as GRiT's settings. As shown in Table 3b, GRiT is comparable to the closed-set object detector with a 0.8 AP gap, which once again validates GRiT's open-set framework is promising to be a new formulation for object detection.

Object Feature Size: We experiment with different sizes of object features input to the text decoder. As shown in Table 3a, 49 feature vectors achieve similar performance to 196 feature vectors, which indicates the text decoder is robust to the number of input object features.

Beam Search: Standard object detector outputs multiple class labels for one box by its multi-category classifier. Similarly, we use beam search to output multiple class name texts for each box as described in the method section. As shown in Table 3c, beam search improves object detection metric, especially for recall, and beam size=3 is a good trade-off between accuracy and inference time.

Size	AP	AP_{50}	AP_{75}
7×7	50.8	68.4	55.6
14×14	**50.9**	68.6	55.6

(a) Object feature size. GRiT is not sensitive to the number of object features.

Method	AP	AP_{50}	AP_{75}
Closed-set OD	**51.7**	70.0	56.4
GRiT	50.9	68.6	55.6

(b) GRiT vs. Closed-set object detector. Closed-set OD follows the same setting as GRiT but replaces the text decoder with a closed-set classifier as in standard OD.

Beam size	AP	AR@1	AR@10
1	50.0	37.0	61.4
2	50.8	38.0	63.6
3	**50.9**	38.3	64.0
5	**50.9**	**38.4**	**64.1**

(c) Beam search. Beam search improves especially recall by labeling one box with more than one class name.

Training progress	0-60k	60k-80k	80k-90k	AP
Training object classes	60	60	80	49.1
	60	80	80	50.4
	80	80	80	**50.9**

(d) Incremental training. GRiT seamlessly learns new object classes that are added in the middle of training.

Beam size	Objectness	Description	AP	AR@1	AR@10
1	✓		49.1	36.9	61.3
	✓	✓	**50.0**	**37.0**	**61.4**
3	✓		19.7	32.3	61.3
	✓	✓	**50.9**	**38.3**	**64.0**

(e) Object scoring. Object description score is crucial to remove false alarms when there is more than one label for a box.

Table 3: Ablation studies on COCO 2017 val. All models are based on ViT-B trained by 90k iterations with a batch size of 32.

Object Scoring: To rate object predictions, we combine both objectness score from the foreground object extractor and object description score from the text decoder. GRiT is always equipped with objectness score due to its function of removing background boxes. As shown in Table 3e, description score improves 0.9 AP when beam size=1. However, the model without description score fails when beam size=3, *i.e.*, outputting three class names each box. The reason is that all the three classes share the same confidence score though at least two of them are false positives. This has a mild impact on recall but leads to significantly worse precision and AP.

Incremental Training: GRiT is open-set and capable of generating unlimited number of words. As data evolve, one can add new object classes or concepts in the middle of training without adapting any architecture. We simulate this use case on COCO in Table 3d, where we start training with 60 classes and add the remaining 20 classes in the middle of training. Compared to the model that is trained on all classes throughout, we achieve similar results when adding the rest of the classes in the last one-third of training. GRiT performs reasonably even in the case where we supplement the remaining classes in the last one-ninth of training.

Method	*Pre-training* Task (Data)	Parameters	Backbone	AP
GIT [33]	Language Modeling (Image-text pairs)	backbone, text decoder	ViT-B	52.0
			ViT-L	52.7
			Coswin-H	54.8
MAE [13]	Image Reconstruction (ImageNet-1K)	backbone	ViT-B	53.6
			ViT-L	56.3
			ViT-H	58.8

Table 4: GRiT pre-training. MAE pre-training outperforms GIT pre-training. GIT adopts Coswin-H [39] as the visual backbone, so we adjust our backbone accordingly. Results are evaluated on COCO 2017 val.

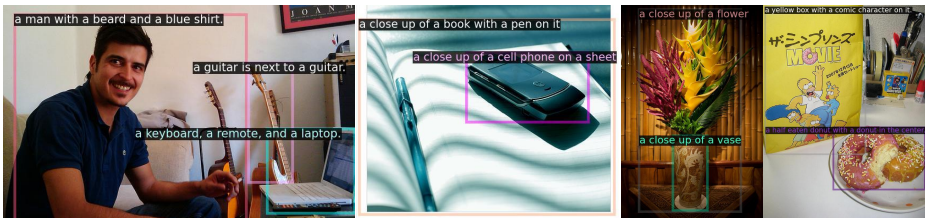

Fig. 4: Zero-shot object understanding predictions.

Pre-training: We study GRiT with different pre-training schemes. As shown in Table 4, MAE pre-training shows better performance than GIT pre-training. MAE pre-training is to recover the masked patches in the image, which may exhibit a stronger localization capability helping object detection task. GIT pre-training focuses more on the image-level representation and language modeling, which shows slightly better performance on dense captioning task, as in Table 1.

4.5 Zero-shot Object Understanding

GRiT follows the image-to-text transformer's design for the network structure of the visual encoder backbone and text decoder (GIT [33] here in this paper). We explore whether GRiT can achieve zero-shot object understanding by simply using the image-to-text transformer's trained parameters without finetuning on object description annotations. To this end, we initialize GRiT by the GIT model fine-tuned on COCO image captioning task. Then, we finetune our feature pyramid and foreground object extractor on COCO detection data (no use of the class names) while keeping GIT's parameters fixed, such that GRiT is able to generate object boxes. To align with GIT's parameters, object features are cropped from the last feature map of the visual backbone rather than the feature pyramid. This zero-shot object understanding result is shown in Fig. 4. We see that the model generates various object-level descriptions for different regions in the same image though GIT is trained on the image-level description task. While,

Training Task	Testing Task	
	ObjectDet	DenseCap
ObjectDet, DenseCap	50.7	14.37
ObjectDet	53.8	-
DenseCap	-	15.48

Table 5: Jointly *vs.* Separately training object detection and dense captioning. Object detection and dense captioning are evaluated on COCO 2017 test-dev and VG test, respectively. All experiments are based on ViT-B.

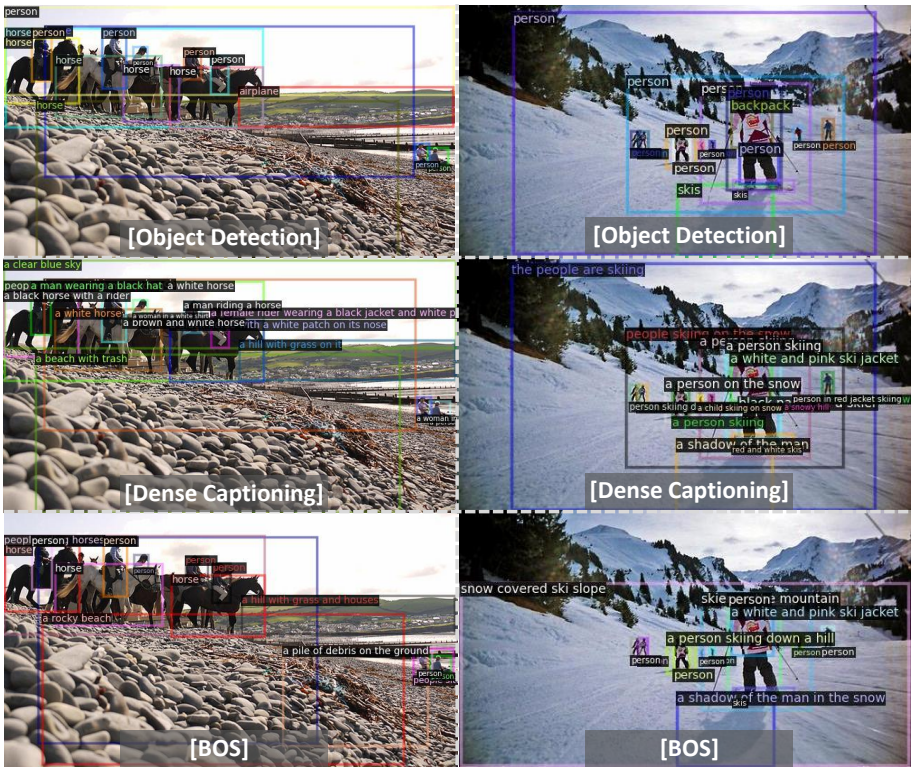

Fig. 5: Predictions from the models jointly trained on object detection and dense captioning. Row 1&2: the model using two begin tokens `[ObjectDet]` and `[DenseCap]`. Row 3: the model using only one begin token `[BOS]`. To generate object descriptions in a specific style, it is necessary to instruct the model with a unique begin token for that style.

we also notice in experiments that there are images for which this zero-shot scheme does not work well. We hypothesize the issue is that GIT's text decoder parameters do not well recognize the cropped features as GIT's text decoder is trained with the entire image's features as input.

4.6 Joint Object Detection and Dense Captioning

Since GRiT is a general object understanding model and can generate any style of object descriptions in the same framework, we jointly train a model on both object detection (short-description task) and dense captioning (long-description task). As shown in Table 5, we compare the jointly trained model with two models that are trained on each task separately. We find that the separately trained model outperforms the jointly trained model. The main cause is that the COCO and VG datasets are not in consensus about box annotations. All boxes in COCO are specific objects, while VG has lots of "scene boxes" covering a whole scene and describing multiple objects together. This leads to many false positives when testing COCO object detection. For example, as shown in Fig. 5, there are several large boxes predicted by the model focusing on a whole scene rather than a specific object. These boxes are regarded as false positives in COCO but they are meaningfully annotated in VG, for example, "a beach with trash". This annotation style difference can also cause low recall when testing VG dense captioning as those "scene boxes" are suppressed during training when images come from COCO dataset. We believe GRiT is capable of achieving both tasks in the same trained model without performance drops if the box data have no disagreement.

As discussed in the method section, we instruct GRiT to generate task-specific descriptions by the begin tokens $\{\texttt{[task]}_i\}_{i=1}^{T}$ when jointly training the tasks with different styles of descriptions. To demonstrate this adaptation is the key to multi-task object understanding in one trained model, we compare to the model that is jointly trained on both tasks using only one begin token `[BOS]`. As shown in the last row of Fig. 5, the model with only `[BOS]` token cannot generate consistent object descriptions in the same image. Some objects are described in the way of dense captioning, while others are simply described by COCO class names. We also notice that COCO-background regions are more likely described by descriptive sentences because such regions only exist in the VG dataset during training. In contrast, the model informed by $\texttt{[task]}_i$ token correctly generates descriptions in the style we request.

Lastly, we inspect the generated captions by the joint trained model on 5,000 COCO test set images. We find that 38% of the object instances produced a caption description that is outside COCO vocabulary and without any COCO vocabulary words.

5 Conclusion

This work presents a general and open-set object understanding framework, GRiT. GRiT formulates object understanding as region-text pairs, which is capable of unifying various region/object-level tasks in a single paradigm. GRiT is end-to-end from image feature extraction to foreground object detection to object description generation. Extensive experiments on object detection and dense captioning demonstrate the effectiveness and generality of GRiT.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
3. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al.: Hybrid task cascade for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4974–4983 (2019)
4. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
5. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
7. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7373–7382 (2021)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
11. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2918–2928 (2021)
12. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021)
13. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
14. Hu, X., Gan, Z., Wang, J., Yang, Z., Liu, Z., Lu, Y., Wang, L.: Scaling up vision-language pre-training for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17980–17989 (2022)
15. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4565–4574 (2016)
16. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**(1), 32–73 (2017)

17. Lavie, A., Agarwal, A.: METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In: Proceedings of the Second Workshop on Statistical Machine Translation. pp. 228–231. Association for Computational Linguistics, Prague, Czech Republic (Jun 2007), https://aclanthology.org/W07-0734
18. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
19. Li, X., Jiang, S., Han, J.: Learning object context for dense captioning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8650–8657 (2019)
20. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. arXiv preprint arXiv:2203.16527 (2022)
21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
25. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. Advances in neural information processing systems **24** (2011)
26. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
28. Schuster, M., Nakajima, K.: Japanese and korean voice search. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5149–5152. IEEE (2012)
29. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019)
30. Shao, Z., Han, J., Marnerides, D., Debattista, K.: Region-object relation-aware dense captioning via transformer. IEEE Transactions on Neural Networks and Learning Systems (2022)
31. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)

32. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
33. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
34. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
35. Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z.: End-to-end semi-supervised object detection with soft teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3060–3069 (2021)
36. Yang, L., Tang, K., Yang, J., Li, L.J.: Dense captioning with joint inference and visual context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2193–2202 (2017)
37. Yin, G., Sheng, L., Liu, B., Yu, N., Wang, X., Shao, J.: Context and attribute grounded dense captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6241–6250 (2019)
38. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:2205.01917 (2022)
39. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)
40. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
41. Zhang, H., Zhang, P., Hu, X., Chen, Y.C., Li, L.H., Dai, X., Wang, L., Yuan, L., Hwang, J.N., Gao, J.: Glipv2: Unifying localization and vision-language understanding. arXiv preprint arXiv:2206.05836 (2022)
42. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., Luo, P.: Gpt4roi: Instruction tuning large language model on region-of-interest. arXiv preprint arXiv:2307.03601 (2023)
43. Zhao, Y., Liu, Y., Guo, Z., Wu, W., Gong, C., Ye, Q., Wan, F.: Controllable dense captioner with multimodal embedding bridging. arXiv preprint arXiv:2401.17910 (2024)
44. Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L.H., Zhou, L., Dai, X., Yuan, L., Li, Y., et al.: Regionclip: Region-based language-image pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16793–16803 (2022)
45. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. arXiv preprint arXiv:2201.02605 (2022)
46. Zhou, X., Koltun, V., Krähenbühl, P.: Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461 (2021)
47. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
48. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)