GiT: Towards Generalist Vision Transformer through Universal Language Interface

Haiyang Wang[1,2*], Hao Tang[1*], Li Jiang[3,2✉], Shaoshuai Shi[2], Muhammad Ferjad Naeem[4], Hongsheng Li[5], Bernt Schiele[2], and Liwei Wang[1✉]

[1]Peking University [2]Max Planck Institute for Informatics
[3] The Chinese University of Hong Kong, Shenzhen
[4] ETH Zurich [5] The Chinese University of Hong Kong
{wanghaiyang@stu, tanghao@stu, wanglw@cis}.pku.edu.cn
jiangli@cuhk.edu.cn {sshi, schiele}@mpi-inf.mpg.de
ferjad.naeem@vision.ee.ethz.ch hsli@ee.cuhk.edu.hk

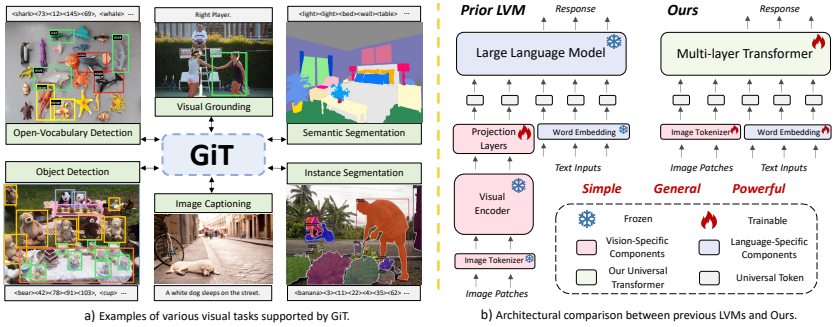

Fig. 1: ***G**eneralist **Vi**sion **T**ransformer*. a) Examples of tasks supported by GiT. b) Architectural comparison between previous LVMs (*e.g.*, LLaVA [55]), and ours. GiT seamlessly handles various vision-centric tasks, particularly fine-grained visual perception, via a universal language interface using a plain transformer (*e.g.*, ViT and GPT).

Abstract. This paper proposes a simple, yet effective framework, called GiT, simultaneously applicable for various vision tasks only with a vanilla ViT. Motivated by the universality of the Multi-layer Transformer architecture (*e.g.*, GPT) widely used in large language models (LLMs), we seek to broaden its scope to serve as a powerful vision foundation model (VFM). However, unlike language modeling, visual tasks typically require specific modules, such as bounding box heads for detection and pixel decoders for segmentation, greatly hindering the application of powerful multi-layer transformers in the vision domain. To solve this, we design a universal language interface that empowers the successful auto-regressive decoding to adeptly unify various visual tasks, from image-level understanding (*e.g.* captioning), over sparse perception (*e.g.* detection), to dense prediction (*e.g.* segmentation). Based on the above designs, the entire model is composed solely of a ViT, without any specific additions, offering a remarkable architectural simplification. GiT is a multi-task visual model, jointly trained across five representative benchmarks without

* Equal contribution. ✉ Corresponding author.

task-specific fine-tuning. Interestingly, our GiT builds a new benchmark in generalist performance, and fosters mutual enhancement across tasks, leading to significant improvements compared to isolated training. This reflects a similar impact observed in LLMs. Further enriching training with 27 datasets, GiT achieves strong zero-shot results over various tasks. Due to its simple design, this paradigm holds promise for narrowing the architectural gap between vision and language. Code and models will be available at `https://github.com/Haiyang-W/GiT`.

Keywords: Unified Visual Modeling · Multi-Task Learning

1 Introduction

Developing a universal model capable of completing various tasks aligned with human intention is a long standing goal in Machine Learning. In language processing, the emergence of LLMs [1,69,82,102] opens up a promising route, which only employs several stacked transformer layers for adaptable task management with minimal prompts. In this paper, we explore this simple multi-layer transformer [84] architecture in visual modeling, seamlessly integrating numerous vision tasks with a universal language interface, aiming to close the architecture gap between vision and language.

The Machine Learning community is undergoing a paradigm shift with the rise of foundation models (*e.g.*, GPT [9], BERT [43], DALL-E [71]) trained on massive data, enabling the sharing of conceptual knowledge, and offering seamless adaptability to diverse downstream tasks. Language models [9, 43, 82] have greatly benefited from this recently, thanks to a homogenized representation (*i.e.*, input and output are uniformly represented as token sequence). State-of-the-art models like GPT4 [65], LLaMA [82], PaLM2 [1] and Gemini [81] have shown an unprecedented ability to follow human instructions and solve open-ended tasks. Thanks to their success, this architecture is potentially viewed [8,72] as a general framework for other machine learning tasks beyond NLP.

Motivated by this opportunity, the community has developed several large vision models, such as LLaVA [55], Unified-IO [59] and OFA [88], by leveraging vision features [30,38] as foreign language of open-source LLMs [70,80,82]. However, this progress still retained task-specific designs, including diverse visual encoders [88, 104], perception heads [49], RPN [49], and specific target representations [59]. Task-specific modules require intricate designs for each task a model needs to solve, potentially hindering progress towards a general vision model. Moreover, these task-specific designs typically involve numerous separate training stages [89], complicating model scaling across different tasks. We argue that an alternative general-purpose framework could employ lightweight components through a more universal input-output interface, and allocate most of the model resources to learning a general model across these tasks.

Previous attempts [3, 7, 28, 51, 55, 90, 104] on large visual modeling predominantly focused on the image-level vision-language domain, mainly due to its straightforward integration into LLMs by viewing the image as a foreign language. This approach often overlooks the incorporation of classical perception

Table 1: Columns from left to right display task source examples, dataset counts, total samples, percentages, and multi-task sampling rates, then task modalities. Highlighted rows summarize statistics for similar task groups. See appendix for the complete list.

	Example Sources	Size				Input Modalities		Output Modalities		
		Dataset	Size	Percent	Weight	Text	Image	Text	Sparse	Dense
Image-Level		**10**	**11.4m**	**67.1**	**40**	✓	✓	✓	✓	-
Image Captioning	*CC12M [14], VG [46], SBU [66]*	5	11.3m	66.6	30	-	✓	✓	-	-
Visual Grounding	*RefCOCO [100], Flickr30k [68]*	5	115k	0.7	10	✓	✓	-	✓	-
Object-Level		**11**	**5.2m**	**30.9**	**40**	-	✓	-	✓	✓
Object Detection	*Objects365 [75], COCO [54]*	8	3.8m	22.6	20	-	✓	-	✓	-
Instance Segmentation	*OpenImages [48], LVIS [35]*	4	1.4m	7.9	20	-	✓	-	✓	✓
Pixel-Level		**6**	**322k**	**2.0**	**20**	-	✓	-	-	✓
Semantic Segmentation	*COCOStuff [12], ADE20K [103]*	6	322k	2.0	20	-	✓	-	-	✓
All Tasks		**27**	**17m**	**100**	**100**	✓	✓	✓	✓	✓

tasks, such as detection and segmentation. Developing a unified framework for fundamental visual perception has proven to be quite challenging since it requires the model to predict multiple outputs with different formats in parallel, with annotations varying widely in representations, ranging from coarse-grained image level to fine-grained pixel level. For example, detection yields variable numbers of bounding boxes, segmentation produces binary masks, and image captioning generates textual answers. These drawbacks make it difficult to design a single model simultaneously applicable across all visual tasks.

Recent developments in LLMs [4,9,64,65] have shown the potential of Transformer [84] being a universal computation architecture. Inspired by this, we introduce GiT, a vision foundation model that can handle diverse vision-centric tasks. As shown in Figure 1, compared to previous unified models [59, 88, 89], our method features a minimalist design, comprising just several Transformer layers without any vision-specific additions other than the patch projection layers, closely aligning with LLM architectures. Similar to language modeling, all visual tasks are structured into an auto-regressive framework through a universal language interface. Specifically, our targets are expressed as token sequences using a unified representation, relying solely on a standard vocabulary without involving extra tokens [72,89]. To be compatible with various visual tasks across different perceptual scales, we introduce a flexible multi-task template for task unification. It partitions the whole image into N subregions by grid sampling and concurrently processes each subregion with efficient parallel decoding.

The above designs facilitate multi-task training of our model across five representative benchmarks without task-specific fine-tuning. As shown in Table 3 and 4, leveraging shared parameters and representation, our model achieves strong generalist results and mirrors the multi-task capabilities of LLMs [4]. Tasks with overlapping abilities can mutually enhance each other, leading to significant gains over separate training (see §5.2 for more analysis). To further enhance generalizability, we incorporate 27 standard visual datasets into training (see Table 11), resulting in strong zero- and few-shot performances on unseen data.

In particular, our work makes the following contributions:

- *Foundational framework for unified visual modeling.* We introduce a simple visual modeling paradigm with a straightforward multi-layer transformer, greatly simplifying the model design. Our model integrates various vision-centric tasks, especially the often-neglected object- and pixel-level tasks, via an efficient universal language interface.

Table 2: Summary of architecture configuration. Shared parameters account for over 98% of the whole model. The parameter of text embedding is excluded because it operates in a zero-computation index manner.

Model	Multi-Modal Tokenizers			Multi-layer	Layer	Total
	Text	Image	Out-of-vocab	Transformer	Number	Parameter
GiT $_{\text{Base}}$	0	0.4%	1.8%	97.8%	18 (12+6)	131M
GiT $_{\text{Large}}$	0	0.2%	1.1%	98.7%	30 (24+6)	387M
GiT $_{\text{Huge}}$	0	$< 0.1\%$	0.8%	99.1%	38 (32+6)	756M

Table 3: Abilities required for each task and the performance improvements after multi-task training. † means polygon-based segment [93, 94], different from the popular mask-based methods [36].

Task	Image	Language	Segment	Localization	*Improve* (single→multi)
Detection	✓	-	-	✓	+***1.6***@AP
InsSeg	✓	-	✓†	✓	+***1.6***@AP$_{50}$, +***0.2***@AP$_{75}$
Grounding	✓	✓	-	✓	+***2.5***@Acc
Caption	✓	✓	-	-	+***4.7***@CIDEr
SemSeg	✓	-	✓	-	+***0.1***@mIoU

- *Multi-task ability like LLMs.* Weight-sharing and unified learning objectives enable us to obtain the multi-task capability as observed in LLMs, achieving the best and mutually enhanced generalist performance over five benchmarks.
- *Strong generalizability.* Fully embracing the one-stage joint training strategy as used in LLMs, our model is trained on 27 publicly available datasets, achieving strong zero- and few-shot performance across various tasks.

2 Related Work

Multi-layer Transformer [84] has emerged as a universal learning architecture, becoming a cornerstone in most LLM frameworks. Notable LLMs like GPT series [4, 9, 64, 65, 67, 69], as well as LLaMA [82], PaLM [1], and OPT [102] have made significant advances in this domain. Beyond language, plain transformer also has proven effective in 2D vision with ViT [30], 3D vision via DSVT [85], multimodal imaging in UniTR [86]. Despite their success, these straightforward transformers are often limited to feature encoding and require task-specific modules, greatly hindering the progress toward a general learner. To solve this, we aim to broaden the scope of multi-layer transformer, moving beyond their conventional encoder-only function to an LLM-like visual modeling. Our model employs several transformer layers for various visual tasks with a universal language interface, narrowing the architectural gap between the vision and language.

Vision Foundation Model excels in handling diverse visual tasks within a unified architectural framework. Motivated by the success of seq2seq models in NLP, innovations like OFA [88], Flamingo [3], LLaVA [55] and Gato [72] have reframed vision-language tasks as sequence generation problems, which is further developed by Unified-IO [59], Pix2Seq v2 [22], and VisionLLM [89] to process spatial information across more tasks. However, these approaches face challenges such as inefficient inference from non-parallel decoding [22] or the complexity of vision-specific additions [49, 59, 89], slowing progress towards a universal vision model. Moreover, they often lack LLMs' multi-task capabilities, where joint training yields superior performance compared to individual training.

3 Universal Language Interface

In this section, we propose a simple universal language interface that integrates five fundamental visual tasks, ranging from image, over object to the pixel level,

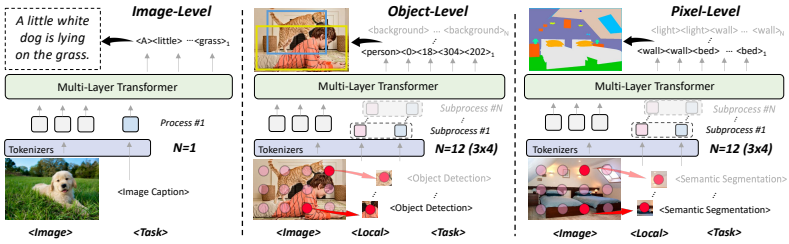

Fig. 2: Task-level customization spans from image- to object- and pixel-level, setting N to 1, 625 (25×25), and 1764 (42×42), in real implementation. Red point means localized visual token, generated by image bilinear interpolation at its grid point. Task prompt is text, converted into a token via text and out-of-vocabulary representation.

into the successful auto-regressive framework. All our targets are expressed as token sequences via a unified representation (§3.1), and then organized by a general multi-task template (§3.2), which partitions the fine-grained visual perception into a series of parallel-decoded subproblems. Figure 2 illustrates the multi-task input templates for three tasks, namely image captioning (image-level task, left), object detection (object-level task, middle) and semantic segmentation (pixel-level task, right). Further technical details are provided below.

3.1 Unified Input and Output Representation

To support various modalities such as images, language, bounding boxes, masks, *etc*, it's essential to represent them in a unified space. To achieve this, we straightforwardly project the input image and text into patch and language token embeddings. Following this, all targets are represented via a universal language interface and tokenized entirely based on a standard vocabulary [92].

Text representation. Vision-language tasks often require text processing, like image captioning, where a natural language description is generated based on the given image. To handle it, we follow the practice of BERT [43], texts are transformed into WordPiece [92] subwords, with a ∼30,000 token vocabulary, and then embedded via a lookup table into a learnable embedding space. Position encodings are added to indicate local positions within time steps.

Out-of-vocabulary representation. Visual perception typically relies on complex textual concepts comprised of multiple pieces, such as "traffic light" and "20 47", the category name and numerical value used in object detection. As discussed in [52,89], using multiple tokens to represent them is inefficient. 1) Adding separators like </c>"traffic light"</c> to identify categories will extend sequence length, particularly impractical for dense prediction tasks. 2) Varying token length for multi-piece words leads to inconsistent decoding steps, necessitating complex and rule-based post-processing to achieve reliable outcomes. To tackle this problem, some solutions [59,72,89] introduce new tokens of category and number terms while facing challenges when considering token capacity constraints. Instead of expanding the vocabulary, we treat multi-piece concepts as continuous text and compress them into a single token as follows,

$$\begin{aligned} \mathcal{I}_0, \mathcal{I}_1 &= \text{Tokenizer}(\text{“traffic light”}), && \mathcal{I} \text{ is the token index}, \\ \mathcal{F}_0, \mathcal{F}_1 &= \text{Attention}(\text{TE}(\mathcal{I}_0) + \text{PE}(0), \text{TE}(\mathcal{I}_1) + \text{PE}(1)), && \mathcal{F}_{\text{traffic light}} = \mathcal{F}_0, \end{aligned} \tag{1}$$

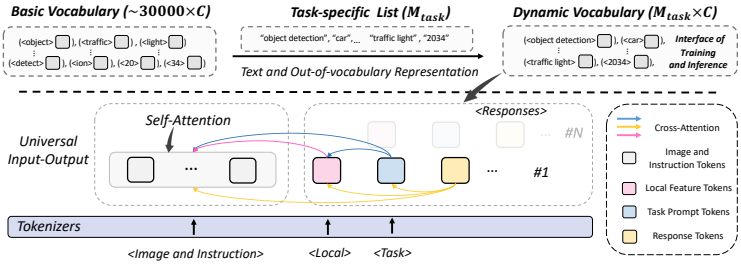

Fig. 3: Our multi-task formulation is broadly illustrated as processing four types of user inputs: image patches, instructive language tokens, and N parallel point-based subprocesses, each with its interpolated local image feature and task identifier for efficient parallel visual prediction. As for the language interface, we use a basic vocabulary, a specific vocabulary list required by the current task, and the task-agnostic out-of-vocabulary module (§3.1) to dynamically create vocabulary sets for each task.

where Attention$(\cdot)$ is a single-layer attention, TE$(\cdot)$ and PE$(\cdot)$ are text and position embedding functions. Our approach offers an alternative solution for handling any out-of-vocabulary terms without expanding the basic vocabulary, which greatly simplifies the post-processing for achieving effective perception.

Sparse representation. In the context of sparse object-level perceptions such as object detection [13, 34] and instance segmentation [37], which generate various category and location representations (for example, bounding boxes and instance masks), we propose a standardized output format. This format is defined as a tuple (C, P), where C represents the category label, and $P{=}\{x_i, y_i\}_{i=1}^{N}$ denotes a set of N points that identify the object's location. To align with the format of linguistic tokens, both the class and location targets are tokenized by the prior text and out-of-vocabulary representation. Following VisionLLM [89], continuous coordinates are uniformly discretized into integers within `[-range, range]`. A bounding box is formulated with four points as $\{x_1, y_1, x_2, y_2\}$, representing its top-left and bottom-right coordinates, while instance mask defines its fine-grained region via multiple points along the boundary [93, 94].

Dense representation. Various perceptual tasks, such as semantic segmentation [58, 74], require models to generate dense outputs, often involving per-pixel predictions. To handle these tasks, we start by converting per-pixel labels into unified tokens. For example, semantic classes [54] are firstly tokenized by text and out-of-vocabulary representation. Then, these dense labelings are flattened into a 1D sequence in raster order, represented autoregressively, similar to iGPT [20].

Image representation. Images are converted into a non-overlapping 16×16 patch sequence in raster order and then embedded to tokens with a trainable linear projection and a learnable positional embedding, as done in ViT [30].

3.2 Multi-Task Template with Parallel Decoding

Prior to constructing the templates, we first divide 2D visual understanding into three distinct categories, each defined by their perceptual granularity and output representation. Our focus encompasses five core tasks for training and analysis: 1) *Image-level* tasks, exemplified by Image Captioning and Visual Grounding,

2) *Object-Level* tasks like Object Detection and Instance Segmentation, and 3) *Pixel-Level* tasks such as Semantic Segmentation. Then, we introduce a unified seq2seq framework that seamlessly integrates various task formulations, from purely visual to those involving language, enabling flexible task customization.

General Formulation. Inspired by well-established language models, we adapt the widely accepted instruction template of LLMs to the vision community (*e.g.*, vision-language and spatial-aware visual perception). As shown in Figure 2 and 3, the instructional template is defined as follows,

$$\underbrace{\text{<Image><Instruction>}}_{\text{shared global observation}} \underbrace{\begin{cases} \text{<LocalFeature}_1\text{><Task}_1\text{> : <Response}_1\text{>} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \text{<LocalFeature}_N\text{><Task}_N\text{> : <Response}_N\text{>.} \end{cases}}_{\text{multi-track local observations and responses}} \tag{2}$$

In our template, user input is structured into four segments. The first comprises image patches, as done in ViT. The second involves instruction inputs, like language expression used for visual grounding. For the third and fourth segments, targeting efficient object- and pixel-level visual perception like simultaneously predicting multiple bounding boxes as in traditional object detection, we partition the task into N parallel local subprocesses by grid sampling, as shown in Figure 2. Each subprocess works with a local image token, created by bilinearly interpolating image features based on its grid point position, and a pure text task identifier, converted into a single token via text and out-of-vocabulary representation. For Vision-Language tasks, we set N to 1, while for vision-centric tasks like detection and segmentation, N is adjustable to match the required prediction resolution. These designs allow our method to flexibly handle nearly all 2D vision tasks. Notably, some segments are optionally required by different tasks, *e.g.*, image captioning only requires image inputs and a task prompt.

In contrast to the traditional encoder and decoder setups, we employ various mask matrices to determine the token representation context. As shown in Figure 3, our method processes inputs (*i.e.*, image and instruction) by applying bidirectional self-attention, similar to a typical encoder. Importantly, we enable image-to-text attention to enhance its ability of text-conditioning image processing (see Table 7). As for computing local and task prompts, and target prediction of each subprocess, we use left-to-right unidirectional attention for modeling causal relations, in line with decoder-only autoregressive approach.

Image-Level. The definition for image-level tasks such as image captioning and visual grounding is straightforward, closely mirroring the NLP tasks. Following previous vision-language methods, we set N to 1 and structure the token sequence of image captioning as {<`image`> "*image captioning*": <`text`>}, and visual grounding as {<`image`> <`instruction`> "*visual grounding*": <`bbox`>}.

Object-Level. Developing a generative framework that adeptly manages classical object-level perception tasks, including object detection and instance segmentation, presents a significant challenge. It demands a model capable of concurrently generating all the bounding boxes and masks. To address this, as shown in Figure 2, we introduce a point-based parallel decoding framework designed for visual prompt perception. It starts by sampling a grid of N points across the

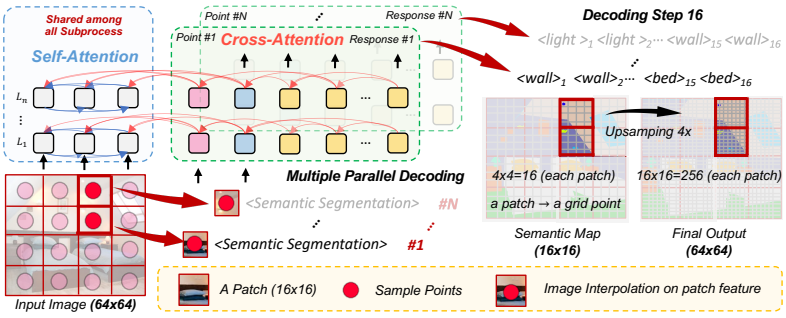

Fig. 4: An illustration of pixel-level multiple parallel decoding. Consider a 64×64 image divided into 16 patches, where each patch is 16×16. With N=16 and a decoding step of 16 per subprocess, each grid point covers one patch to predict a 4×4 semantic map, which is then upsampled 4× to the original size for the final result.

image, where N is set to 625, corresponding to a 25 × 25 sampling resolution for 1120 × 1120 images. Following this, we conduct generative perception at each point using the format: {<image> <local feature> <task identifier>: <sparse response>}. <image> is the patch tokens shared by all grid subprocesses. <sparse response> indicates our chosen object-level sparse representation as detailed in §3.1. Notably, if the point is in the negative part, <background> token will be predicted.

An example of detection for a grid point: {<image><local feature> "*object detection*": <c><x$_1$><y$_1$><x$_2$><y$_2$>}, where <c> is the class label, and (<x$_1$><y$_1$><x$_2$><y$_2$>) indicate the box points' offsets from the grid points.

Pixel-Level. The auto-regressive decoding paradigm [9, 65, 69] struggles with high-dimensional outputs, particularly in cases like computing all pixel semantic categories in a single sequence, incurring considerable computational overhead. Earlier efforts [59, 63] attempted to alleviate this using compressed tokens via VQ-VAE [83]. However, this approach compromised the pure language interface and introduced intricate modules. To tackle this issue, as illustrated in Figure 4, we convert per-pixel labels into linguistic tokens and further divide the image into N uniform sub-regions, just like object-level tasks. Specifically, for segmentation tasks, we set N to 1764 to achieve a 42×42 perceptual resolution for images sized 672 × 672. Each subprocess independently conducts sequential pixel-level predictions in parallel, leading to enhanced efficiency.

An example of semantic segmentation for a single track with 16 decoding steps: {<image> <local feature> "*semantic segmentation*": <c$_1$> <c$_2$> ··· <c$_{15}$> <c$_{16}$>}, where <c$_i$ > is the i-th class token of each sub-region.

4 Training

4.1 Architecture: Multi-layer Transformer

By employing the universal language interface, we formulate a diverse array of 2D vision tasks as sequences of discrete input and output tokens. This method

has paved the way for extending the successful architectures (such as Multi-layer Transformers [9, 69, 84]) in Large Language Models, to unified visual modeling.

Building on the visual foundations, we leverage the structure of window-based ViT [30, 53], identical to the visual encoder used in SAM [45], for both linguistic sequences and high-resolution images. A few global attention blocks are evenly integrated into the model for feature propagation. Notably, within the window attention layer, each patch token only interacts with grid points located in the same window. Our approach can be built upon such a common structure (*i.e.*, ViT) without architectural changes, enhancing the framework's universality.

Benefiting from the above designs, our architecture can allocate the most of computational parameters ($> 98\%$) to general inference, complemented by a few lightweight modules for diverse modality inputs, as shown in Table 2.

4.2 Multi-Task and Universal Training

GiT undergoes joint training across various tasks and datasets. Our goal is to assess the capability of a unified model to handle multiple tasks simultaneously. Thus, we refrain from task-specific fine-tuning, despite prior studies demonstrating its potential to enhance task performance.

Various Tasks and Datasets. To build a singular unified model for diverse perception and V&L tasks, we construct an analyzable multi-task benchmark comprising the most representative datasets across five fundamental tasks we previously identified, spanning from image- to pixel-level visual understanding. To enhance the model's adaptability, we augment the benchmark by integrating 27 datasets from 16 publicly accessible data sources, as listed in Table 11.

Joint Multi-Task Training. We jointly train GiT on the above multi-task benchmark by mixing samples from these datasets. As detailed in Table 11, to prevent overshadowing tasks with smaller data during joint training and avoid potential performance drops, we uniformly sample from all tasks (1/5), regardless of their data sizes. In universal settings where tasks span multiple domains, sampling inside each task is balanced across scenarios like daily life, indoor, and outdoor. Within these domains, datasets are sampled in proportion to their size.

Regarding the learning objective, different tasks require distinct vocabularies. For example, visual grounding uses numerical coordinates, whereas segmentation involves semantic concepts. To tackle this problem, as illustrated in Figure 3, we approach all tasks as the next token generation problem using standard CrossEntropy loss, while employing a task-specific vocabulary. This allows for dynamically controlling vocabulary sets, adapting to the unique requirements of each task during both training and inference phases.

Scaling Models. We adopt a variant of ViT [30] similar to SAM [45], augmented with six extra transformer layers and text embeddings used in BERT [43] to improve non-visual modality processing (refer to Table 9). To study the dependence of performance on model scale, we introduce three different sizes of model built up on ViT-B, -L, and -H, with parameters ranging from 131M to 756M, detailed in Table 2. The initial layers inherit parameters pretrained by SAM, while the new layers start with random initialization.

Table 4: Results on standard vision-centric benchmarks. "single-task" refers to models trained on each task separately, while "multi-task" indicates models trained jointly across all selected benchmarks. "⋆" denotes the model is capable of the task, though no number is reported. "-" means incapability in that specific task. "†" indicates that the generalist model embedded previous task-specific models to enhance performance. GiT stands out as the first generalist model to support all listed vision tasks, delivering competitive outcomes without task-specific adaption. Following [17,49], some generalist models that only report results with task-specific fine-tuning are not included, *e.g.*, OFA [88] and X-Decoder [108]. We highlight the top-1 entries of one-stage multi-task generalist models and joint training improvements with **bold** font. Specific module counts exclude non-computational ones, like index-based text tokenizers.

Methods	Specific Modules		#Params	Object Detection			Instance Seg			Semantic Seg	Captioning		Grounding
	Examples	Num		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	mIoU(SS)	BLEU-4	CIDEr	Acc@0.5
Specialist Models													
Faster R-CNN-FPN [73]	ResNet,RPN	5	42M	40.3	61.0	44.0	-	-	-	-	-	-	-
DETR-DC5 [13]	ResNet,Encoder	5	41M	43.3	63.1	45.9	-	-	-	-	-	-	-
Deformable-DETR [106]	ResNet,Encoder	5	40M	45.4	64.7	49.0	-	-	-	-	-	-	-
Pix2Seq [21]	ResNet,Encoder	3	37M	43.0	61.0	45.6	-	-	-	-	-	-	-
Mask R-CNN [36]	ResNet,RPN	6	46M	41.0	61.7	44.9	37.1	58.4	40.1	-	-	-	-
Polar Mask [93]	ResNet,FPN	5	55M	-	-	-	30.5	52.0	31.1	-	-	-	-
Mask2Former [25]	ResNet,Decoder	5	44M	-	-	-	43.7	-	-	47.2	-	-	-
VL-T5 [26]	Faster R-CNN	3	440M	-	-	-	-	-	-	-	34.5	116.5	-
UNITER [24]	Faster R-CNN	4	303M	-	-	-	-	-	-	-	-	-	81.4
MDETR [40]	RoBERTa,DETR	6	188M	-	-	-	-	-	-	-	-	-	86.8
Generalist Models (Pre-training + MultiTask-Tuning)													
UniTab [97]	Encoders	4	185M	-	-	-	-	-	-	-	⋆	115.8	88.6
Pix2Seq v2 [22]	ViT,Decoder	2	132M	46.5	⋆	⋆	38.2	⋆	⋆	-	34.9	⋆	⋆
Unified-IO$_{XL}$ [59]	VQ-VAE	4	2.9B	-	-	-	-	-	-	⋆	⋆	122.3	⋆
Shikra-13B [17]	ViT,Vicuna	3	13B	-	-	-	-	-	-	-	⋆	117.5	87.8
Generalist Models (MultiTask-Training)													
Uni-Perceiver [107]	None	1	124M	-	-	-	-	-	-	-	32.0	⋆	⋆
Uni-Perceiver-MoE [105]	None	1	167M	-	-	-	-	-	-	-	33.2	⋆	⋆
Uni-Perceiver-V2 [49]	Mask DINO,Swin	8	308M	58.6†	⋆	⋆	50.6†	⋆	⋆	-	35.4	116.9	⋆
VisionLLM-R50 [89]	Deform-DETR	6	7B	44.6	64.0	48.1	25.1	50.0	22.4	-	31.0	112.5	80.6
GiT-B$_{\text{single-task}}$	None	1	131M	45.1	62.7	49.1	31.4	54.8	31.2	47.7	33.7	107.9	83.3
GiT-B$_{\text{multi-task}}$	None	1	131M	46.7	64.2	50.7	31.9	56.4	31.4	47.8	35.4	112.6	85.8
Improvement (single→multi)				***+1.6***	***+1.5***	***+1.6***	***+0.5***	***+1.6***	***+0.2***	***+0.1***	***+1.7***	***+4.7***	***+2.5***
GiT-L$_{\text{multi-task}}$	None	1	387M	51.3	69.2	55.9	35.1	61.4	34.7	50.6	35.7	116.0	88.4
GiT-H$_{\text{multi-task}}$	None	1	756M	**52.9**	**71.0**	**57.8**	**35.8**	**62.6**	**35.6**	**52.4**	**36.2**	**118.2**	**89.2**

5 Experiments

5.1 Experimental Settings

Multi-Task Datasets. To facilitate in-depth analysis and fair evaluation, we built an analyzable multi-task benchmark, choosing one of the most representative datasets for each task. To ensure consistency and enable comparison with VisionLLM [89], we retained the same datasets they used for the four vision-centric tasks: COCO2017 [54] for object detection and instance segmentation, COCO Caption [23] for image captioning, and the RefCOCO series [60,100] for visual grounding. For the semantic segmentation not included in VisionLLM, we employed the widely used ADE20K dataset [103].

Extended Datasets. To showcase the universality of our unified framework, we enhanced our multi-task benchmark by integrating more standard and publicly available datasets from vision-language and visual perception (see §4.2).

Training and Evaluation Details. To illustrate the flexibility and efficacy of our model, we established three training paradigms: single-task, multi-task, and universal setting. In single-task training, the focus is on optimizing performance

Table 5: Zero shot results. "⋆" and "-" follow Table 4. † are the performance reproduced based on the mmdetection [16]. "universa" extends the multi-task setting by including a broader array of datasets, as detailed in §4.2.

Methods	Specific Modules Examples	Num	#Params	Object Detection Cityscapes [27]	Instance Seg Cityscapes [27]	Semantic Seg Cityscapes [27]	Semantic Seg SUN RGB-D [78]	Captioning nocaps [2]
Supervised								
Faster R-CNN-FPN [73]	ResNet,RPN	5	42M	40.3	-	-	-	-
Mask R-CNN [36]	ResNet,RPN	6	46M	40.9	36.4	-	-	-
DeepLabV3+ [19]	ResNet,Decoder	3	63M	-	-	80.9	⋆	-
Mask2Former [25]	ResNet,Decoder	5	44M	-	-	80.4	⋆	-
TokenFusion [91]	Segformer,YOLOS	4	-	-	-	⋆	48.1	-
Zero-Shot Transfer								
GLIP-T [52]	Swin,Dy-Head	5	156M	28.1†	-	-	-	-
Grounding DINO-T [56]	Swin,DINO	6	174M	31.5†	-	-	-	-
BLIP-2 (129M) [50]	ViT-G,Qformer	4	12.1B	-	-	-	-	**15.8**
ReCo+ [77]	DeiT-SIN	4	46M	-	-	24.2	⋆	-
XDecoder-T [108]	FocalNet,Encoder	4	165M	-	16.0	47.3	34.5	⋆
GiT-$B_{\text{multi-task}}$	None	1	131M	21.8	14.3	34.4	30.9	9.2
GiT-$B_{\text{universal}}$	None	1	131M	29.1	17.9	56.2	37.5	10.6
GiT-$L_{\text{universal}}$	None	1	387M	32.3	**20.3**	58.0	39.9	11.6
GiT-$H_{\text{universal}}$	None	1	756M	**34.1**	18.7	**61.8**	**42.5**	12.6

on individual benchmarks. Multi-task training, on the other hand, targets the development of a general learner across five selected datasets. Drawing from the insights in Uni-Perceiver v2 [49], we adopt an unmixed sampling strategy (*i.e.*, sampling one task per iteration) for faster and more stable training, However, our framework is also compatible with in-batch mixing strategies [59, 107] as suggested by recent studies. Universal training expands our approach to incorporate 27 comprehensive benchmarks introduced in §4.2. All models leverage AdamW [44] optimizer with a cosine annealing schedule, setting the initial learning rate to 0.0002 and weight decay to 0.05. The largest models of the universal setting are trained on 96 NVIDIA A100 GPUs for 320k iterations.

All experiments are evaluated on the selected datasets using standard protocols and test split. Due to the limited space, more details are in Appendix.

5.2 In-distribution Benchmarking

We evaluate our model's in-distribution performance on various vision-centric tasks, comparing it with both task-specific and advanced generalist models. It relies solely on a stacked multi-layer transformer, adapting to various tasks only through instruction and post-processing changes.

Comparison with Specialist Models. We compare our single-task model with well-established specialist baselines in Table 4. Our model demonstrates the ability to perform various vision-centric tasks individually within the same framework, narrowing the performance gap with specialized models. It achieves comparable results in most tasks (*e.g.*, detection: 45.1 vs. 45.4 of Deformable-DETR [106], semantic segmentation: 47.7 vs. 47.2 of Mask2Former [25]), but slightly underperforms in instance segmentation. This is typical for polygon-based methods, which often yield lower results than mask manner. Our model improves by +0.9 against PolarMask [93], a leading polygon-based method.

Notably, to maintain a universal interface, our method only uses the basic label assignments, without employing the latest enhancement techniques, leaving

Table 6: Few shot results of out-distributed domains. We conduct this experiment based on weights pretrained in the universal stage. "⋆" , "-" and † follow Table 5.

Methods	Specific Modules		Medical Imaging@mDice	Remote Sensing@mIoU		Human Centric@mAP	
	Examples	Num	DRIVE [79]	LoveDA [87]	Potsdam [39]	WIDERFace [96]	DeepFashion [57]
Supervised							
U-Net [74]	None	1	81.4	⋆	⋆	-	-
AerialFormer [95]	Encoder,Stem	3	-	54.1	89.1	-	-
RetinaFace [29]	ResNet,FPN	5	-	-	-	52.3	-
Mask R-CNN [36]	ResNet,RPN	6	-	-	-	⋆	59.9
Few-Shot Transfer							
Faster RCNN [73]	ResNet,RPN	4	-	-	-	25.4†	14.9†
DeepLabV3 [18]	ResNet,ASPP	3	32.1†	20.3†	24.2†	-	-
GiT-$B_{\text{multi-task}}$	None	1	34.3	24.9	19.1	17.4	23.0
GiT-$B_{\text{universal}}$	None	1	51.1	30.8	30.6	31.2	38.3
GiT-$L_{\text{universal}}$	None	1	55.4	34.1	37.2	33.4	49.3
GiT-$H_{\text{universal}}$	None	1	**57.9**	**35.1**	**43.4**	**34.0**	**52.2**

huge room for performance gains. For example, label assignment used in detection closely mirrors Deformable-DETR [106]. Adopting more advanced strategies like DINO's contrastive DeNoising [101] could further improve our results.

Comparison with Generalist Models. Some generalist models [17, 22, 59, 88] employ a two-stage training process, initially leveraging large-scale, task-relevant datasets like image-text pairs or diverse perception data, and then undergoing single- or multi-task downstream tuning within the same framework to enhance performance. Our GiT fully embraces the more challenging one-stage joint training, popularized in LLMs, that blends all data for unified modeling followed by direct downstream evaluation, without any task-specific adaptation.

Table 4 shows that our model not only adeptly manages dense prediction but also outperforms the former leading generalist model, VisionLLM [89], across all tasks, with 50× fewer parameters and a much simpler framework.

Table 4,5,6 show that scaling our model greatly improves multitask, zero- and few-shot performance, sometimes even matching supervised approaches.

Discussion about multi-task capacity. Table 4 reveals that GiT-$B_{\text{multi-task}}$ outperforms GiT-$B_{\text{single-task}}$, showing notable improvements in each task after joint training on five standard datasets. As observed in Table 3, multi-task training typically boosts performance when tasks share the same capabilities but are less effective otherwise. This pattern is clearly observed in the shared localization ability across detection, visual grounding, and instance segmentation. Conversely, specialized skills, like fine-grained dense prediction in semantic segmentation and polygon-based regression in instance segmentation don't see significant gains from multi-tasking.

5.3 Out-of-distribution Analysis

Zero-Shot Transfer. After large-scale multi-task training, GiT is readily assessed on a variety of novel data sources. To demonstrate this capability, we conducted zero-shot evaluations on three established datasets across five configurations, addressing four vision tasks beyond visual grounding. These evaluations span a range of contexts, from indoor environments like SUN RGB-D [78],

Table 7: Ablation of modality experts and text conditioning on GiT-$B_{\text{multi-task}}$, using multiple FFN for multimodal learning and image-to-text attention in visual grounding.

Modality Experts	Text Conditioning	Detection@AP	Ins Seg@AP	Sem Seg@mIoU(SS)	Caption@CIDEr	Grounding@Acc(0.5)
		46.1	31.4	47.8	111.8	78.6
✓		46.2	31.6	47.7	112.2	78.7
	✓	46.7	31.9	47.8	112.6	85.8

Table 8: Ablation study between encoder-decoder and decoder-only architecture.

Methods	Enc Layer	Dec Layer	Detection@AP	Ins Seg@AP	Sem Seg@mIoU(SS)	Caption@CIDEr	Grounding @Acc(0.5)
GiT-$B_{\text{multi-task}}$	12	6	46.3	31.6	46.9	110.8	84.8
GiT-$B_{\text{multi-task}}$	0	18	46.7	31.9	47.8	112.6	85.8

outdoor scenes such as Cityscapes [27], and daily life like nocaps [2]. We report mIoU and SPICE [5] for semantic segmentation and captioning, mAP for object detection and instance segmentation.

As shown in Table 5, our universal models achieve the best results in nearly all tasks. With comparable parameters, GiT-$B_{\text{universal}}$ surpasses X-Decoder [108] on Cityscapes (+8.9) and SUN RGB-D (+3.0) on semantic segmentation, and shows similar advantages in instance segmentation and object detection. Scaling the model further enhances its zero-shot capabilities, nearing supervised performance. BLIP-2 [50] outperforms GiT-H on nocaps, likely attributed to its integration with pretrained language models and extensive training data (129M). Notably, to our knowledge, GiT is the first generalist model to achieve zero-shot performance across various domains and tasks.

Few-Shot Transfer. GiT demonstrates rapid adaptation to out-of-distribution data sources. We conducted a comprehensive few-shot evaluation on five datasets in medical imaging (*i.e.*, DRIVE [79]), remote sensing (*i.e.*,LoveDA [87] and ISPRS [39]), and human-centric scenarios (*i.e.*, WIDERFace [96] and DeepFashion [57]). Our approach follows the N-way K-shot [32] setting (*i.e.*, K=5) and directly fine-tune the pre-trained model on support sets [10].

In our segmentation analysis, we choose DeeplabV3 as our baseline, which aligns with the dataset (*i.e.*, ADE20K) used for training our multi-task variant. We observed that both GiT $_{\text{multi-task}}$ and DeeplabV3 perform poorly in the few-shot setting. However, after large-scale universal training, GiT-$B_{\text{universal}}$ demonstrates significantly improved generalization. This trend is mirrored in detection tasks, underscoring that our universal model structure and training approach greatly enhances generalization capabilities.

5.4 Ablation Study

Decoder-only Architecture. Our model follows the GPT's decoder-only design, though its advantages over encoder-decoder frameworks are not well-explored. We transformed GiT-B's initial 12 layers into an encoder for image and text, excluding target tokens. Table 8 shows that the encoder-decoder paradigm underperforms decoder-only models in all five tasks, particularly in semantic segmentation with a -0.9 drop. This might be due to decoder-only models allocating more layers (18 vs 6) for processing target tokens.

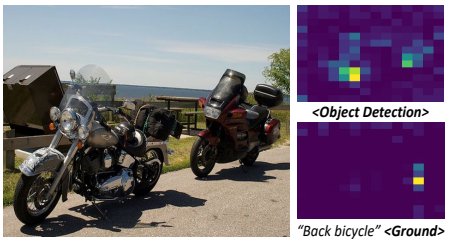

Fig. 5: Visualizations on cross-attention between task token and image, with yellower colors indicating higher responses.

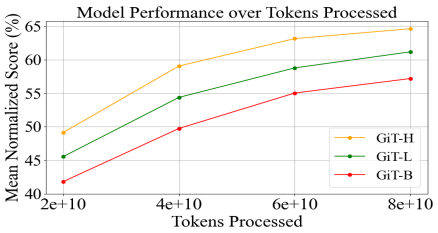

Fig. 6: Model size scaling law results. In-distribution performance as a function of tokens processed for 3 model scales.

Number of New Layers. Table 9 shows adding just one new layer can significantly boost performance, improving mAP by 2.6, likely due to the difference between image input and language targets. Involving more layers continues to improve results, with gains leveling off after six layers.

Table 9: Ablation study of new layer on GiT-B$_{\text{single-task}}$.

New Layers	Detection@AP
0	40.2
1	42.8
2	43.8
3	44.6
6	45.1

Modality Experts. Although employing multiple FFN as modality experts is a commonly used practice [6, 105] for multimodal processing, Table 7 shows no notable performance gains in our approach, leading us to exclude this design due to its increased parameters and inference latency.

Text Conditioning. In our visual grounding task with image and text inputs, we enable image-to-text attention during network forwarding. Table 7 shows that this method markedly improves performance in a multi-task setting, likely due to its enhanced differentiation between detection and visual grounding tasks. These two tasks function at distinct image scales (*i.e.*, 1120 and 224), where the former involves identifying multiple boxes, while the latter involves generating a single box guided by text. Moreover, this approach may help the model capture image-text relationships, boosting the ability of instruction-following.

Scaling Law Analysis. Figure 6 presents an in-distribution performance of our universal model against its parameter count, offering insights into the potential enhancements with expanding model capacity. We plot performance progression for three model sizes based on a composite score averaging key metrics from all tasks, showing significant gains with increased scale at a consistent token count.

6 Conclusion

In this paper, we introduce GiT, a simple yet powerful vision foundation model that utilizes only a vanilla ViT to integrate diverse visual tasks through a universal language interface. Mirroring multi-task abilities as observed in LLMs, GiT establishes new benchmarks in generalist performance. With training across 27 datasets, GiT becomes the first generalist model to excel in zero- and few-shot tasks across diverse domains using shared parameters, showcasing the foundational role of the multi-layer transformer in computer vision.

References

1. Aakanksha, C., Sharan, N., Jacob, D., Maarten, B., Gaurav, M., Adam, R., Paul, B., Won, C.H., Charles, S., Sebastian, G., et al.: Palm: Scaling language modeling with pathways. JMLR (2023)
2. Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., Anderson, P.: Nocaps: Novel object captioning at scale. In: ICCV (2019)
3. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. In: NeurIPS (2022)
4. Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., Ilya, S., et al.: Language models are unsupervised multitask learners. OpenAI blog (2019)
5. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: Semantic propositional image caption evaluation. In: ECCV (2016)
6. Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O.K., Aggarwal, K., Som, S., Piao, S., Wei, F.: Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In: NeurIPS (2022)
7. Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., Taşırlar, S.: Introducing our multimodal models (2023), `https://www.adept.ai/blog/fuyu-8b`
8. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: NeurIPS (2020)
10. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017)
11. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
12. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
13. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
14. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR (2021)
15. Chen, C., Borgeaud, S., Irving, G., Lespiau, J.B., Sifre, L., Jumper, J.: Accelerating large language model decoding with speculative sampling. arXiv preprint arXiv:2302.01318 (2023)
16. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
17. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)

18. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CVPR (2017)
19. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
20. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: ICML (2020)
21. Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G.: Pix2seq: A language modeling framework for object detection. In: ICLR (2022)
22. Chen, T., Saxena, S., Li, L., Lin, T.Y., Fleet, D.J., Hinton, G.E.: A unified sequence interface for vision tasks. NeurIPS (2022)
23. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
24. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: ECCV (2020)
25. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
26. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. In: ICML (2021)
27. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
28. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. NeurIPS (2023)
29. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: CVPR (2020)
30. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
31. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
32. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. PMLR (2017)
33. Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. NeurIPS (2020)
34. Girshick, R.: Fast r-cnn. In: ICCV (2015)
35. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
36. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
37. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
38. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
39. III/4, I.W.: ISPRS 2D Semantic Labeling Contest, `https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx`
40. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: ICCV (2021)
41. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)

42. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP (2014)
43. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
44. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2015)
45. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV (2023)
46. 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. IJCV (2017)
47. Kuhn, H.W.: The hungarian method for the assignment problem. NRL (1955)
48. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)
49. Li, H., Zhu, J., Jiang, X., Zhu, X., Li, H., Yuan, C., Wang, X., Qiao, Y., Wang, X., Wang, W., et al.: Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. In: CVPR (2023)
50. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. ICML (2023)
51. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
52. 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: CVPR (2022)
53. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: ECCV (2022)
54. 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: ECCV (2014)
55. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS (2023)
56. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
57. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR (2016)
58. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
59. Lu, J., Clark, C., Zellers, R., Mottaghi, R., Kembhavi, A.: UNIFIED-IO: A unified model for vision, language, and multi-modal tasks. In: ICLR (2023)
60. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR (2016)
61. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
62. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017)
63. Ning, J., Li, C., Zhang, Z., Wang, C., Geng, Z., Dai, Q., He, K., Hu, H.: All in tokens: Unifying output space of visual tasks via soft token. In: ICCV (2023)
64. OpenAI: Chatgpt (2022), `https://openai.com/blog/chatgpt`
65. OpenAI: Gpt-4 technical report (2023)

66. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. NeurIPS **24** (2011)
67. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. NeurIPS **35** (2022)
68. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV (2015)
69. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
70. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR (2020)
71. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML (2021)
72. Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J.T., et al.: A generalist agent. TMLR (2022)
73. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NeurIPS (2015)
74. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
75. 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: ICCV (2019)
76. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018)
77. Shin, G., Xie, W., Albanie, S.: Reco: Retrieve and co-segment for zero-shot transfer. In: NeurIPS (2022)
78. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: CVPR (2015)
79. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. TMI (2004)
80. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca` (2023)
81. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
82. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
83. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: NeurIPS (2017)
84. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
85. Wang, H., Shi, C., Shi, S., Lei, M., Wang, S., He, D., Schiele, B., Wang, L.: Dsvt: Dynamic sparse voxel transformer with rotated sets. In: CVPR (2023)
86. Wang, H., Tang, H., Shi, S., Li, A., Li, Z., Schiele, B., Wang, L.: Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In: ICCV (2023)

87. Wang, J., Zheng, Z., Ma, A., Lu, X., Zhong, Y.: Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. In: NeurIPS (2021)
88. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: ICML (2022)
89. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. NeurIPS (2023)
90. Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O.K., Singhal, S., Som, S., et al.: Image as a foreign language: Beit pretraining for all vision and vision-language tasks. CVPR (2023)
91. Wang, Y., Chen, X., Cao, L., Huang, W., Sun, F., Wang, Y.: Multimodal token fusion for vision transformers. In: CVPR (2022)
92. 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)
93. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. In: CVPR (2020)
94. Xu, W., Wang, H., Qi, F., Lu, C.: Explicit shape encoding for real-time instance segmentation. In: ICCV (2019)
95. Yamazaki, K., Hanyu, T., Tran, M., Garcia, A., Tran, A., McCann, R., Liao, H., Rainwater, C., Adkins, M., Molthan, A., et al.: Aerialformer: Multi-resolution transformer for aerial image segmentation. arXiv preprint arXiv:2306.06842 (2023)
96. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: CVPR (2016)
97. Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., Wang, L.: Unitab: Unifying text and box outputs for grounded vision-language modeling. In: ECCV (2022)
98. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. In: ICLR (2024)
99. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020)
100. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV. Springer (2016)
101. 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. In: ICLR (2022)
102. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022)
103. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR (2017)
104. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)

105. Zhu, J., Zhu, X., Wang, W., Wang, X., Li, H., Wang, X., Dai, J.: Uni-perceiver-moe: Learning sparse generalist models with conditional moes. NeurIPS (2022)
106. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. ICLR (2020)
107. Zhu, X., Zhu, J., Li, H., Wu, X., Li, H., Wang, X., Dai, J.: Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In: CVPR (2022)
108. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., et al.: Generalized decoding for pixel, image, and language. In: CVPR (2023)

In our supplementary, we provide detailed information including model design specifics in §A, dataset summaries in §B, along with in-depth training, inference and evaluation procedures in §C and D. Additional ablation experiments are included in §E. §F details specific modules used in comparative methods. Qualitative results across different datasets and tasks are in §G. Lastly, limitations, negative societal impacts, and a comparison with Fuyu-8B are in §H.

A Implementation details

Window Attention. Our window attention is adapted from the SAM [45] variant of ViT [30]. Following SAM, after patch embedding, images are downsampled by a factor of 16, and windows are defined with a size of 14×14. The primary distinction from the original lies in how we handle multi-track local observations and responses in the parallel training stage, such as grid-wise prompts (*i.e.*, local image token, task identifier) and their outputs. To manage these multi-track elements, we merge them into a sequence and append them after the shared observation. Consequently, the input to window attention consists of multiple parts, requiring a customized attention mask to ensure grid independence while enabling autoregressive prediction, as detailed in Figure 7. Within each subprocess group (*i.e.*, those associated with the same grid), interactions are left-to-right unidirectional attention. Moreover, tokens belonging to different subprocesses are isolated, preventing them from accessing each other's information.
Global Attention. In tasks that require object- and pixel-level analysis, the large number of local predictions creates significant memory and computational burdens, especially in global attention layers, where processing attention across all grid points can be unnecessary and inefficient. Therefore, for such tasks, we have optimized the global attention layer to focus only on the shared global observations (*i.e.*, input image and text), eliminating the need to compute targets for each grid. Table 10 shows that this strategy slightly impacts performance but greatly decreases computation time. However, in captioning and visual grounding with a 224 image size, which involves only one window and a single global response, this optimization is unnecessary.

Table 10: Performance of semantic segmentation by single-task training with our accelerated global attention. It significantly reduces the computational cost with slight performance drops.

Global Attention	mIoU	Training Time
Normal	47.9	51h
Accelerated	47.7	35h

Out-of-vocabulary Representation. We encode multi-piece out-of-vocabulary concepts to a single token. This is achieved through a streamlined approach that utilizes only one attention layer combined with absolute positional encoding. As

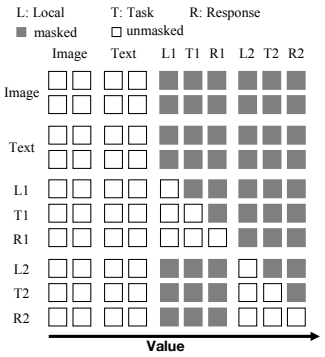

Fig. 7: Attention mask visualization.

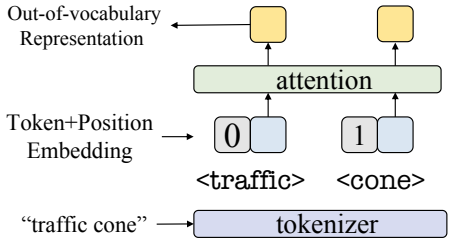

Fig. 8: Out-of-vocabulary representation.

shown in Figure 8, "traffic cone" is tokenized as <traffic><cone>. The corresponding text embeddings, augmented with positional encoding, are input into the attention layer, allowing each word to interact with the rest. We select the first output token as the final representation for multi-word concepts like "traffic cone". For single-word concepts, we use the original text embedding directly.

Background Representation. Given that each dataset contains distinct positive and negative classes, utilizing text labels like <background> to denote negative classes could lead to ambiguity when training across multiple datasets. Therefore, we employed a unique encoding approach for the background class,

$$\mathcal{F}_{\text{background}} = -\sum_{i=0}^{N-1} \mathcal{F}_i / N \tag{3}$$

where $\mathcal{F}_i$ is the representation of i-th positive class and N denotes the total number of categories. This approach makes the cosine similarity between tokens of a positive class and those assigned to the background class typically negative. Its superior performance in zero-shot scenarios highlights its effectiveness.

Resolution and Coordinate Discretization. For our experiments, we use different image resolutions tailored to specific tasks: 1120 × 1120 pixels for object detection and instance segmentation, 672 × 672 pixels for semantic segmentation, and 224 × 224 pixels for image captioning and visual grounding. To encode spatial positions as discrete tokens, we discretize the image coordinates into a set number of intervals. Specifically, we determine the number of these intervals to be double the resolution of the input image. For instance, with an input image of 224 × 224 pixels, we divide the coordinate space into 448 discrete intervals.

B Extended Datasets

B.1 In-distribution Datasets

During universal training, a total of 27 datasets from 16 publicly accessible data sources are used, with sizes and weights detailed in Table 11. Note that the actual quantities in web-sourced caption datasets (CC3M [76], CC12M [14], SBU Captions [66]) are fewer than the original number reported due to inactive links.

Table 11: Universal training dataset details. Columns from left to right indicate dataset size, proportion to total data, assigned group number, and sampling weight. Weights are evenly distributed across the tasks. Different scenarios within each task (*e.g.*, daily life, autonomous driving) create individual groups with equal weights. Sampling weights in groups are set based on dataset sizes.

Dataset	Size	Percent (%)	Group ID	Weight (%)
Object Detection	3.8M	22.55	-	20.00
Objects365 [75]	1.7M	9.98	0	3.22
OpenImages [48]	1.7M	9.98	0	3.22
LVIS [35]	164K	0.96	0	0.23
nuImages [11]	93K	0.55	1	6.66
Pascal VOC 2007 [31]	10K	0.06	2	0.37
Pascal VOC 2012 [31]	11K	0.06	2	0.22
COCO 2017 [54]	164K	0.96	2	6.07
Instance Segmentation	1.4M	8.34	-	20.00
LVIS [35]	164K	0.96	3	0.76
OpenImages [48]	1M	5.87	3	5.90
nuImages [11]	93K	0.55	4	6.66
COCO 2017 [54]	164K	0.96	5	6.66
Semantic Segmentation	322K	1.89	-	20.00
COCO-Stuff [12]	164K	0.96	6	6.28
Pascal Context [61]	10K	0.06	6	0.38
nuImages [11]	93K	0.55	7	4.84
BDD100K [99]	10K	0.06	7	0.52
Mapillary Vistas [62]	25K	0.15	7	1.30
ADE20K [103]	20K	0.12	8	6.67
Image Caption	11.3M	66.54	-	20.00
CC3M [76]	1.8M	10.57	9	1.74
CC12M [14]	7.8M	45.79	9	6.96
SBU Captions [66]	800K	4.70	9	0.58
Visual Genome [46]	770K	4.52	9	0.71
COCO Caption [23]	164K	0.96	10	10.00
Visual Grounding	115K	0.68	-	20.00
RefCOCO [42]	20K	0.12	11	4.00
RefCOCO+ [42]	20K	0.12	11	4.00
RefCOCOg [60]	25K	0.15	11	4.00
RefCLEF [42]	20K	0.12	12	4.00
Flickr30K [68]	30K	0.18	13	4.00
All	17M	100	-	100

COCO. The MS COCO dataset, or Microsoft Common Objects in Context [54], is a comprehensive dataset for object detection, segmentation, key-point detection, and captioning. It includes over 330K images, with annotations for more than 220K, featuring 1.5 million objects across 80 categories. Each image has five sentence descriptions and 250K pedestrians are annotated with keypoints. The initial release in 2014 has 164K images in training (83K), validation (41K), and test (41K) sets. In 2017, the training/validation split changed to 118K/5K.
Objects365. Objects365 [75] is a vast object detection dataset, comprising 365 object categories and boasting over 2 million training images along with 30 million annotated bounding boxes. This dataset presents diverse objects in different scenarios, providing a robust benchmark for challenging object detection tasks.
OpenImages. Open Images [48] is a dataset with about 9 million images, each annotated with image-level labels, object bounding boxes, segmentation masks, visual relationships, localized narratives, and point-level labels. Covering 20,638 image-level labels, 600 object classes with 16 million bounding boxes, and 2.8 million segmentation masks, it stands as a valuable resource in computer vision.
LVIS. LVIS [35] (Large Vocabulary Instance Segmentation) is a dataset tailored for instance segmentation tasks, providing approximately 2 million high-quality segmentation masks across over 1000 entry-level object categories within a dataset of 164,000 images. This dataset was created to tackle the Zipf distribution commonly observed in natural images, making it an invaluable resource for researchers and developers working on instance segmentation tasks dealing with a large vocabulary of objects.
Pascal VOC 2007. The Pascal VOC 2007 [31] dataset serves as a crucial resource for real-world object recognition, featuring 20 object classes. With 9,963 photos and 24,640 labeled samples, thoughtfully split for balanced training/validation and testing, it stands as a versatile dataset supporting various tasks, including classification, detection, segmentation, and person layout.
Pascal VOC 2012. Pascal VOC 2012 [31] is a valuable dataset for recognizing objects in real-world settings. It encompasses 20 object classes and includes 11,530 images with 27,450 ROI-tagged objects and 6,929 segmentations, serving as a prominent benchmark in computer vision.
nuImages. The nuImages [11] dataset complements the nuScenes [11] for autonomous driving by providing 93,000 2D annotated images, with 1.2 million camera images from past and future timestamps. It is part of the nuScenes ecosystem and focuses on panoptic and multi-annotation aspects. The dataset covers various driving scenarios, including diverse conditions such as rain, snow, and night. It also offers temporal dynamics with 2 Hz spaced images. The annotations encompass 800,000 foreground objects with instance masks and 100,000 semantic segmentation masks.
ADE20K. The ADE20K [103] semantic segmentation dataset comprises 20,000 scene-centric images meticulously annotated at the pixel level for both objects and object parts. Encompassing 150 semantic categories, it includes items like sky, road, and specific objects such as person, car, and bed. The dataset is divided into 20,210 training, 2,000 validation, and 3,000 testing images.
COCO-Stuff. The COCO-stuff [12] dataset holds significance for diverse scene

understanding tasks, such as semantic segmentation, object detection, and image captioning. Derived by augmenting the original COCO dataset, which initially prioritized object annotations, it addresses the oversight of stuff annotations. Spanning 164,000 images, the COCO-stuff dataset includes 172 categories, incorporating 80 things, 91 stuff, and 1 unlabeled class.

Pascal Context. The PASCAL Context [61] dataset extends the PASCAL VOC 2010 [31] detection challenge by providing pixel-wise labels for all training images. Encompassing over 400 classes, which include the original 20 classes from PASCAL VOC segmentation, these classes are categorized into objects, stuff, and hybrids. To address the sparsity of many object categories, a common practice involves using a subset of 59 frequently occurring classes.

BDD100K. BDD100K [99] is a large dataset with 100K videos, providing over 1,000 hours of driving experience and 100 million frames. It includes annotations for road objects, lane markings, drivable areas, and detailed instance segmentation. For road object detection and drivable area segmentation challenges, there are 70,000 training and 10,000 validation images. For full-frame semantic segmentation, there are 7,000 training and 1,000 validation images.

Mapillary Vistas. Mapillary Vistas [62] is a large-scale street-level image dataset with 25,000 high-resolution images. Featuring annotations for 66 object categories, including instance-specific labels for 37 classes, it adopts a dense and fine-grained annotation style using polygons. The dataset primarily focuses on semantic image segmentation and instance-specific image segmentation, aiming to advance visual road-scene understanding.

CC3M. Conceptual Captions, known as CC3M [76], features an extensive collection of around 3.3 million images, each meticulously paired with descriptive captions. Extracted from Alt-text HTML attributes associated with web images, these captions undergo an automated pipeline for quality assurance. This makes the dataset highly versatile, catering to a diverse range of natural language processing and image understanding tasks.

CC12M. Conceptual 12M [14] (CC12M) is a dataset specifically created for vision-and-language pre-training. It consists of a substantial 12 million image-text pairs. Unlike some other datasets with restrictive requirements, CC12M relaxes its data collection pipeline to enhance dataset scale and diversity. It has been shown to provide state-of-the-art results in vision-and-language tasks, particularly in long-tail visual recognition, making it a valuable resource for research and development in this field.

SBU Captions. The SBU Captions dataset [66] is a collection of 1 million images and their associated captions sourced from Flickr, primarily used for training image captioning models. It provides diverse real-world images and textual descriptions, serving as a valuable resource for research in computer vision and natural language processing.

Visual Genome. Visual Genome [46] is a comprehensive dataset with 108,077 images, richly annotated with 5.4 million region descriptions, 1.7 million visual question answers, 3.8 million object instances, 2.8 million attributes, and 2.3 million relationships. This dataset is designed to provide detailed information about images, including objects, attributes, and the relationships between them.

COCO Caption. COCO Captions [23] consists of 1.5 million captions for 330,000 images, with five captions for each image in the training and validation sets. The "Karpathy split", a widely used subset of this dataset created by Andrej Karpathy, involves merging the train and val sets from the raw dataset, creating a new validation set by selecting 5,000 images from the original val set, and an additional 5,000 images are used to form a test set.
RefCOCO. The RefCOCO [42], RefCOCO+ [42], and RefCOCOg [60] datasets were generated through the ReferitGame, a two-player game where one participant describes a segmented object in an image using natural language, and the other participant identifies the correct object. In RefCOCO, there are no language restrictions on referring expressions, whereas in RefCOCO+, location words are prohibited. These datasets concentrate on appearance-based descriptions, such as "the man in the yellow polka-dotted shirt," rather than perspective-dependent ones. RefCOCO comprises 142,209 referring expressions for 50,000 objects in 19,994 images, and RefCOCO+ contains 141,564 expressions for 49,856 objects in 19,992 images.
RefCLEF. RefCLEF [42], also known as ReferIt, consists of 20,000 images sourced from the IAPR TC-12 dataset, accompanied by segmented image regions from the SAIAPR-12 dataset. The dataset is evenly split into two sections: one with 10,000 images designated for training and validation, and another with 10,000 images for testing. The training and validation portion includes a total of 59,976 entries, each consisting of an image, a bounding box, and a description. Test set is slightly larger, featuring 60,105 entries with the same type of data.
Flickr30K. Flickr30K [68] is a widely recognized dataset used for sentence-based image descriptions. It features 31,783 images depicting everyday activities and events, each accompanied by a descriptive caption. This dataset serves as a standard benchmark for studying the relationship between linguistic expressions and visual media.

B.2 Out-distribution Datasets

Cityscapes. Cityscapes [27] is a large dataset for understanding urban scenes, featuring semantic, instance-wise, and pixel-level annotations across 30 classes grouped into 8 categories. It comprises around 5,000 finely annotated images and 20,000 coarsely annotated ones, recorded in various cities under different conditions. This dataset is valuable for tasks related to urban scene analysis.
SUN RGB-D. The SUN RGB-D dataset [78] comprises 10,335 RGB-D images of room scenes, each with depth and segmentation maps. It's annotated for 700 object categories and divided into training and testing sets with 5,285 and 5,050 images, respectively. This dataset addresses the need for large-scale 3D annotations and metrics for scene understanding tasks. It includes data from four sensors, with extensive annotations for 2D and 3D object boundaries, orientations, room layout, and scene categories, enabling advanced algorithm training and cross-sensor bias study.
nocaps. The nocaps [2] dataset pushes image captioning models to grasp a wider array of visual concepts from diverse data origins. Comprising 166,100 human-

generated captions for 15,100 images sourced from OpenImages, the dataset integrates different training data, including COCO image-caption pairs and OpenImages labels and bounding boxes, with a specific emphasis on describing objects.
DRIVE. The DRIVE [79] dataset used for retinal vessel segmentation consists of 40 color fundus images, including 7 displaying abnormal pathology. Captured during diabetic retinopathy screenings in the Netherlands, these images were taken with a Canon CR5 camera featuring a 45-degree field of view. The dataset is split into a training set (20 images) and a testing set (20 images), each accompanied by a circular field of view (FOV) mask. Expert manual segmentations are provided for assessment in the training set, while the testing set includes two observer-based segmentations, with the first observer's results considered as the ground truth for evaluation.
LoveDA. The LoveDA [87] dataset comprises 5987 high-resolution remote sensing images (0.3 m) from urban and rural areas in Nanjing, Changzhou, and Wuhan. It targets semantic segmentation and domain adaptation tasks, offering challenges such as multi-scale objects, complex backgrounds, and inconsistent class distributions, aiming to address diverse geographical environments.
ISPRS Potsdam. The ISPRS Potsdam [39] dataset comprises 38 patches with true orthophotos (TOP) and digital surface models (DSM) having a 5 cm ground sampling distance. The TOP images are available in various channel compositions (IRRG, RGB, RGBIR), and DSM files contain 32-bit float values representing heights. Some patches have normalized DSMs, indicating heights above the terrain. Ground truth labels are provided for a portion of the data, with the rest reserved for benchmark testing.
WIDER Face. The WIDER Face [96] dataset is a comprehensive face detection benchmark dataset, consisting of 32,203 images with a diverse range of 393,703 labeled faces. These images exhibit variations in scale, pose, and occlusion. The dataset is categorized into 61 event classes, with 40% for training, 10% for validation, and 50% for testing. Evaluation follows the PASCAL VOC dataset metric.
DeepFashion. The DeepFashion [57] dataset is a comprehensive collection of around 800,000 fashion images, accompanied by extensive annotations. These annotations include 46 fashion categories, 1,000 descriptive attributes, bounding boxes, and landmark information. The dataset covers a broad spectrum of fashion images, from well-posed product photos to real-world consumer snapshots.

C Training

C.1 Implementation Details

Training schemes. For single-task training, GiT-$B_{\text{single-task}}$ is typically trained using a batch size of 24 for 120,000 iterations on 8 NVIDIA A100 GPUs (40GB), following a cosine annealing schedule. In multi-task joint training on five datasets, GiT-$B_{\text{multi-task}}$ undergoes training with the same batch size and GPU number for more iterations (*i.e.*, 640,000). The large and huge model variants require more GPU memory for training and are therefore trained on 12 and 24 GPUs, respectively. For large-scale universal training, we train all models using a batch

Table 12: Performance of grid sampling on object detection with 25 × 25 grid resolution.

Sample Number	mAP	Training Time
625	45.3	47h
250	45.1	20h

size of 96 across 320,000 iterations. This process is conducted on setups of 32, 48, and 96 GPUs, resulting in total training times of 3, 5, and 7 days, respectively.
Custom learning rate. For the layers without pretraining, we applied the standard base learning rate. In contrast, the layers that had been pretrained used progressively increasing learning rates. This strategy begins with a learning rate that is 0.1 times the base rate for the first pretrained layer, gradually escalating to a full 1.0 times the base rate by the final pretrained layer. We argue this method enhances the integration of pretrained and newly trained weights, leading to better overall performance of the model.
Grid generation and sampling. We adjust the grid sizes according to the level of detail required by each task. For object detection and instance segmentation, we work with 5 × 5 grids in each window, while for semantic segmentation, we increase the grid size to 14 × 14. To illustrate, in object detection, an input image of 1120 × 1120 pixels is represented by a 25 × 25 grids, and in semantic segmentation, a 672 × 672 pixels is represented by a 42 × 42 grids. Computing losses for every point on these grids would demand excessive computational resources, particularly for semantic segmentation. To manage this, we employ a strategy of sampling specific grid points during training, selecting a predetermined number of points with a focus on including positive samples and supplementing with negative samples as needed. Specifically, for object detection and instance segmentation, we choose 10 points out of 25 in each window, and for semantic segmentation, we select 32 points out of 196. As shown in Table 12, this method effectively reduces computational costs without significant performance drops.

C.2 Label Assignment

Object Detection. Our approach employs the well-established Hungarian matching algorithm [47] for label assignment calculation. For each grid point, we compute its normalized L1 distance to the centers of all boxes as the matching cost.
Instance Segmentation. Similar to object detection, instance segmentation targets are determined by computing the L1 distance between bounding box centers and grid positions. Polar coordinates with 24 rays, inspired by PolarMask [93], are employed for mask representation. The mass center of an object is calculated using its annotated polygon boundaries. Grid points classified as positive must accurately predict object category, bounding box, centroid, and distances from the mass center to boundary points.
Semantic Segmentation. Expanding upon ViT, we generate patch features (42 × 42) by downsampling the image (672 × 672) via a factor of 16. Given the dense prediction nature of semantic segmentation, we align the grid point size with the patch feature size. To alleviate computational load, we downsample

Table 13: The evaluation results of the models after universal training on five standard vision-centric benchmarks.

Methods	#Params	Object Detection			Instance Seg			Semantic Seg	Captioning		Grounding
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	mIoU(SS)	BLEU-4	CIDEr	Acc@0.5
$GiT\text{-}B_{universal}$	131M	44.4	61.2	48.1	30.3	53.0	30.0	44.6	33.6	108.3	84.2
$GiT\text{-}L_{universal}$	387M	50.2	67.6	54.6	33.1	58.4	32.7	48.1	36.2	117.5	86.0
$GiT\text{-}H_{universal}$	756M	53.3	71.2	58.3	35.9	62.6	36.1	53.0	37.7	124.2	88.3

original mask annotations (672 × 672) by a factor of 4, resulting in annotations of size 168 × 168, which is four times larger than the grid size. Subsequently, each grid point autonomously predicts segmentation annotations for 16 positions within a 4 × 4 square centered around it.

Image Captioning. In our image captioning process, we tokenize each caption into a fixed-length sequence of 20 tokens. If the caption length is shorter than 20 tokens, we pad it with termination symbols to ensure uniformity.

Visual Grounding. In visual grounding tasks, each query directly targets a specific bounding box, removing the necessity to align boxes with grid points.

C.3 Data Augmentation

Object Detection and Instance Segmentation. For object-level perception tasks, images undergo preprocessing steps. Initially, images are horizontally flipped with a 0.5 probability. Subsequently, two methods are employed to achieve a fixed input size. The first method involves direct resizing of the image to dimensions of 1120 × 1120, disregarding the original aspect ratio. The second method randomly resizes the image to one of three size pairs: (400, 4200), (500, 4200), or (600, 4200), while preserving the original aspect ratio. Following resizing, the image is cropped to a size of (384, 600) and then resized again to 1120 × 1120 pixels.

Semantic Segmentation. In semantic segmentation, specific preprocessing steps are applied to images to ensure their size is standardized and to increase diversity. Initially, images are acquired with a size of 672 × 672 pixels, employing random selection between two methods. The first method directly resizes the image to 672 × 672, disregarding the original aspect ratio. The second method involves scaling the image to sizes ranging from 100% to 200% of 672, again without preserving the original aspect ratio. Following this, a random crop is applied to ensure the image size remains 672 × 672 pixels. Moreover, to augment image diversity, two additional operations are performed with a 50% probability: horizontal flipping and photometric distortions. These steps collectively contribute to a more robust dataset for segmentation tasks.

Image Captioning. As for this task, we initiate preprocessing with a dynamic crop, varying size ratio in [0.08, 1.0] and aspect ratio in [3/4, 4/3] in relation to the original image. Following this crop, the image is resized to 224×224 dimensions. Additionally, there is a 50% probability of horizontally flipping the image for further augmentation.

Visual Grounding. Visual grounding augmentation includes color adjustments

Table 14: Universal training evaluation results on detection, instance segmentation, and visual grounding datasets.

Methods	Object Detection@AP					Grounding@Acc				Instance Seg@AP
	Objects365 [75]	OpenImages [48]	LVIS [35]	VOC0712 [31]	nuImages [11]	RefCOCO+ [42]	RefCOCOg [60]	Flickr30K [68]	RefCLEF [42]	LVIS [35]
GiT-B$_{\text{universal}}$	17.7	43.4	12.3	79.0	44.5	72.5	76.9	71.0	72.2	8.4
GiT-L$_{\text{universal}}$	25.5	51.6	17.3	83.6	47.2	73.9	78.9	72.7	74.5	11.4
GiT-H$_{\text{universal}}$	31.9	57.7	21.7	84.9	50.0	78.3	80.7	77.5	75.8	14.8

Table 15: Evaluation of universal training on segmentation datasets, with all results measured using the mIoU metric.

Methods	COCO-Stuff [12]	Pascal Context [61]	BDD100K [99]	Mapillary Vistas [62]
GiT-B$_{\text{universal}}$	42.6	56.8	57.8	23.0
GiT-L$_{\text{universal}}$	46.0	60.4	59.3	25.4
GiT-H$_{\text{universal}}$	49.1	63.3	61.5	28.9

with a 50% probability, enabling changes in brightness, contrast, saturation, and hue. Subsequently, the image undergoes a random crop within a relative range of (0.8, 0.8) of the original size. Finally, we resize the image to 224×224 without keeping the original aspect ratio.

D Evaluation

D.1 Auto-regressive Decoding

We tailor unique decoding rules for various tasks based on task templates. For example, in object detection, using the template <c><x$_1$><y$_1$><x$_2$><y$_2$>, the category is decoded in the first position, drawing from a vocabulary containing all categories in the dataset. The subsequent four positions decode numerical values, drawing from a vocabulary of discretized locations. Table 16 illustrates the fixed decoding step number for all tasks, with no terminator token required except for image captioning. In image captioning, predictions following the terminator are disregarded during inference.

D.2 Inference Speed

In Table 17, we present the inference speed of GiT-B across five tasks, measured on a single NVIDIA A100 GPU with a batch size of 1. Due to our adherence to the auto-regressive decoding paradigm commonly seen in NLP, we inherit the drawback of slow inference speed. This limitation becomes more pronounced in high-resolution object-level and semantic segmentation tasks that necessitate per-pixel predictions. However, we contend that leveraging multiple parallel decoding has significantly improved our method's speed, bringing it to an acceptable level. As shown in Table 18, our approach demonstrates comparable segmentation speed to SAM. Given that our structure and prediction approach closely align with LLM, the inference acceleration techniques [15] employed for LLM also hold promise for enhancing our method.

Table 16: Decoding steps for all five tasks.

Task	Object Detection	Instance Segmentation	Semantic Segmentation	Image Captioning	Visual Grounding
Decoding Step	5	31	16	20	4

Table 17: Inference speed of GiT-B on A100.

Task	Resolution	Grid Number	Decoding Step	FPS
Object Detection	1120 × 1120	625	5	2.5
Instance Segmentation	1120 × 1120	625	31	0.7
Semantic Segmentation	672 × 672	1764	16	1.5
Image Captioning	224 × 224	1	20	3.2
Visual Grounding	224 × 224	1	4	8.1

Table 18: Latency comparison with SAM on semantic segmentation task.

Method (ADE20K)	Resolution	#Params	FPS
SAM-B [41]	672 × 672	90M	1.6
GiT-B	672 × 672	131M	1.5

D.3 Benchmarking Setup

Multi-Task Learning. On the multi-task datasets, we conducted evaluations on the validation sets, except for COCO Caption [12], where we used the Karpathy split [41] for evaluation on the test set.

Universal Learning. We evaluate our universal models on several key datasets. Table 13 presents their performance on representative datasets for five tasks. However, due to the less frequent sampling of these analyzable multi-task datasets during universal training, their performance slightly lags behind models trained on multi-task benchmark. For further performance insights on other datasets, refer to Tables 14 and 15. Notably, for image captioning, all datasets except COCO Caption are entirely used in training, obviating the need for extra evaluation.

Few-shot Learning. We adopt the classical N-way K-shot [32] setting to create a support set for few-shot evaluation. In this setup, for each class in the dataset, we extract k samples labeled with the corresponding class, resulting in the selection of N×K samples. By default, K is set to 5. As depicted in Table 19, we sample varying quantities of support sets depending on the number of categories in each dataset. Each experiment, by default, iterates 100 times on the support set. However, due to the limited size of the support set in WIDERFace [96], we reduce the iteration count to 50 times to mitigate the risk of overfitting. All few-shot training is conducted with a fixed learning rate of 2e-4.

We select Faster R-CNN [73] and DeepLabV3 [19], two classic methods, as comparative baselines. In the case of Faster R-CNN, we employ the version with ResNet-50 as the backbone, utilizing pre-trained weights from the COCO [54] dataset. For DeepLabV3, we opt for the version with ResNet-101 as the backbone, leveraging pre-training on the ADE20K [103] dataset.

E More ablation studies

Text Conditioning. In visual grounding, we incorporate image-to-text attention during network forwarding, enhancing task differentiation between detection and visual grounding. Table 20 demonstrates that incorporating text conditioning results in a modest improvement of +0.6 in visual grounding when trained

Table 19: Few shot datasets.

Dataset	Size	Category Number	Support Set Size	Training Iters
DRIVE [79]	40	2	10	100
LoveDA [87]	5,987	7	35	100
ISPRS Potsdam [39]	5,472	6	30	100
WIDERFace [96]	32,203	1	5	50
DeepFashion [57]	800,000	15	75	100

Table 20: Ablation of text conditioning on visual grounding task.

Models	Text Conditioning	Acc@0.5
GiT-$B_{\text{single-task}}$		82.7
GiT-$B_{\text{single-task}}$	✓	83.3
GiT-$B_{\text{multi-task}}$		78.6
GiT-$B_{\text{multi-task}}$	✓	85.8

Table 21: Ablation of beam number on image captioning task.

Beam Number	BLEU-4	CIDEr
1	33.1	106.9
2	33.5	107.2
3	33.7	107.9
5	33.7	107.6

independently. However, its impact becomes more significant in multi-task training, showing a remarkable enhancement of +7.2, aligning with our hypothesis.

Beam Search. Table 21 demonstrates how performance varies with different beam sizes in beam search. We observe an improvement as the beam size increases from 1 to 2, but performance stabilizes between 2 and 5, with only a minor drop in CIDEr. Given that larger beam sizes lead to longer inference times, we have selected a default beam size of 2.

Mass Center and Ray Number. Table 22 presents an ablation of instance segmentation settings. Utilizing the mass center yields better results than the box center, probably because the box center might fall outside the object. Employing 36 rays slightly improves performance but at the cost of significant training time.

F Specific Modules of Comparison Methods

In Table 23, we outline the specific modules and parameter quantities utilized for method comparison. Many methods, regardless of whether they are specialist or generalist models, incorporate task-specific modules and modality-specific encoders in their designs. In contrast, our approach is characterized by its simplicity, as it does not rely on such intricate designs.

G Visualization

Task Visualization. In Figure 9, we visualize an example for each task, showcasing the image input, text-formatted predictions, and the visualization of the prediction results from left to right. For simplicity, we selected a few examples of local responses predicted by the model and listed their corresponding text-formatted predictions.

Zero-shot Visualization. In Figure 10, we showcase qualitative examples of predictions on zero-shot datasets made by GiT-$H_{\text{universal}}$. Notably, our model accurately predicts missing annotations in some cases. For instance, in Cityscapes

Table 22: Ablation on instance segmentation settings.

Box Center	Mass Center	Ray Number	mAP	Training Time
✓		24	29.0	32h
✓		36	29.2	49h
	✓	24	31.4	32h
	✓	36	31.7	49h

Table 23: Specific modules and their corresponding parameter quantities for the methods used for comparison. The parameter of text embedding is excluded because it operates in a zero-computation index manner.

Methods	Specific Modules	Num	#Params
Specialist Models			
Faster R-CNN-FPN [73]	ResNet,FPN,RPN,ClassficationHead,RegressionHead	5	42M
DETR-DC5 [13]	ResNet,Encoder,Decoder,ClassficationHead,RegressionHead	5	41M
Deformable-DETR [106]	ResNet,Encoder,Decoder,ClassficationHead,RegressionHead	5	40M
Mask R-CNN [36]	ResNet,FPN,RPN,RPNHead,ClassficationHead,RegressionHead	6	46M
Polar Mask [93]	ResNet,FPN,ClassficationHead,CenternessHead,RegressionHead	5	55M
Mask2Former [25]	ResNet,PixelDecoder,TransformerDecoder,ClassficationHead,MaskHead	5	44M
Pix2Seq [21]	ResNet,Encoder,Decoder	3	37M
UNITER [24]	Faster R-CNN,Project Layer, Encoder,Decoder	4	303M
VILLA [33]	Faster R-CNN, Encoder,Decoder	3	369M
MDETR [40]	CNN,RoBERTa,Image Adapter, Text Adapter,Encoder,Decoder	6	188M
VL-T5 [26]	Faster R-CNN, Encoder,Decoder	3	440M
DeepLabV3+ [19]	ResNet,Decoder,Auxiliary Head	3	63M
TokenFusion [91]	Segformer,YOLOS,Fusion Module,GroupFree	4	79M
U-Net [74]	Encoder,Decoder,Decode Head	3	8M
AerialFormer [95]	Transformer Encoder, CNNs Stem, Multi-Dilated CNNs Decoder	3	114M
RetinaFace [29]	ResNet,FPN,ClassficationHead,RegressionHead,ContextModule	5	30M
Generalist Models			
UniTab [97]	Image Encoder,Text Encoder, Multimodal Encoder, Decoder	4	185M
Uni-Perceiver [107]	None	1	124M
Uni-Perceiver-MoE [105]	None	1	167M
Uni-Perceiver-V2 [49]	ResNet,RPN,Mask DINO,RoBERTa,Decoder,ClassficationHead,RegressionHead,MaskHead	8	308M
Pix2Seq v2 [22]	ViT,Decoder	2	132M
Unified-IO$_{XL}$ [59]	VQ-VAE Encoder,VQ-VAE Decoder,Encoder,Decoder	4	2.9B
Shikra-13B [17]	ViT,Vicuna,Image Adapter	3	13B
Ferret-13B [98]	ViT,Vicuna,Visual Sampler,KNN	4	13B
VisionLLM-R50 [89]	ResNet,Language-Guided Image Tokenizer,Encoder,Decoder,Alpaca-7B	5	7B
GLIP-T [52]	Swin,FPN,Text Encoder,Dy-Head,Fusion Module	5	431M
Grounding DINO-T [56]	Swin,DINO,BERT,Feature Enhancer,Decoder,Query Selection	6	174M
BLIP (129M) [51]	ViT-L,BERT,Image-grounded Text Encoder, Image-grounded Text Decoder	4	583M
BLIP-2 (129M) [50]	ViT-G,Qformer,Adapter,LLM	4	12.1B
ReCo+ [77]	DeiT-SIN,CLIP,DenseCLIP,DeepLabV3+	4	46M
XDecoder(T) [108]	FocalNet,Encoder,Decoder,Latent Query	4	165M

detection, it correctly identifies unannotated bicycles and vehicles, even under low-light conditions. A similar accuracy is observed in SUN RGB-D segmentation, where the model detects all chairs, although only two are annotated. In Cityscapes segmentation, despite the dataset's bias of excluding self-owned vehicles from annotation, our model demonstrates exceptional generalization by correctly classifying these vehicles, relying on minimal information and without dataset-specific fine-tuning.

Few-shot Visualization. Figure 11 provides visual representations of the qualitative predictions made by GiT-H$_{\text{universal}}$ on few-shot datasets. These examples highlight the remarkable performance of our model in situations with limited data, emphasizing its potential for applications across diverse domains.

H Discussion

Comparison with Fuyu-8B. Compared to Fuyu-8B [7], which focuses on well-explored vision-language tasks, our GiT extends the scope of the multi-layer transformer to often-overlooked object and pixel-level tasks with a universal language interface. To achieve it, we design a flexible parallel decoding template using point prompts for task unification across various perceptual scales. The local image prompt is also introduced to enhance fine-grained perception ability.
Comparison with adapter-based methods. Our method provides an alternative solution for LVMs. Unlike previous fine-tuning efforts with LLMs, we aim to close the architectural gap between vision and language. Moreover, our GiT allows easy end-to-end implementation without module-specific design, greatly simplifying the training process and model scaling.
Limitations. Constrained by training data limited to five selected tasks with relatively straightforward task prompts, GiT struggles to generalize to entirely new tasks in zero-shot settings. Task-level zero-shot remains challenging, even for capable LLMs. GiT closely aligns with it and inherits this limitation. However, our GiT shows strong extendibility in task unification, potentially supporting various other tasks by incorporating relevant data.
Negative Societal Impact. Our largest model necessitates 7 days of training on 96 A100 GPUs, leading to considerable carbon emissions. Furthermore, the generated content might reflect biases from the training data, stemming from a lack of alignment with human preferences.

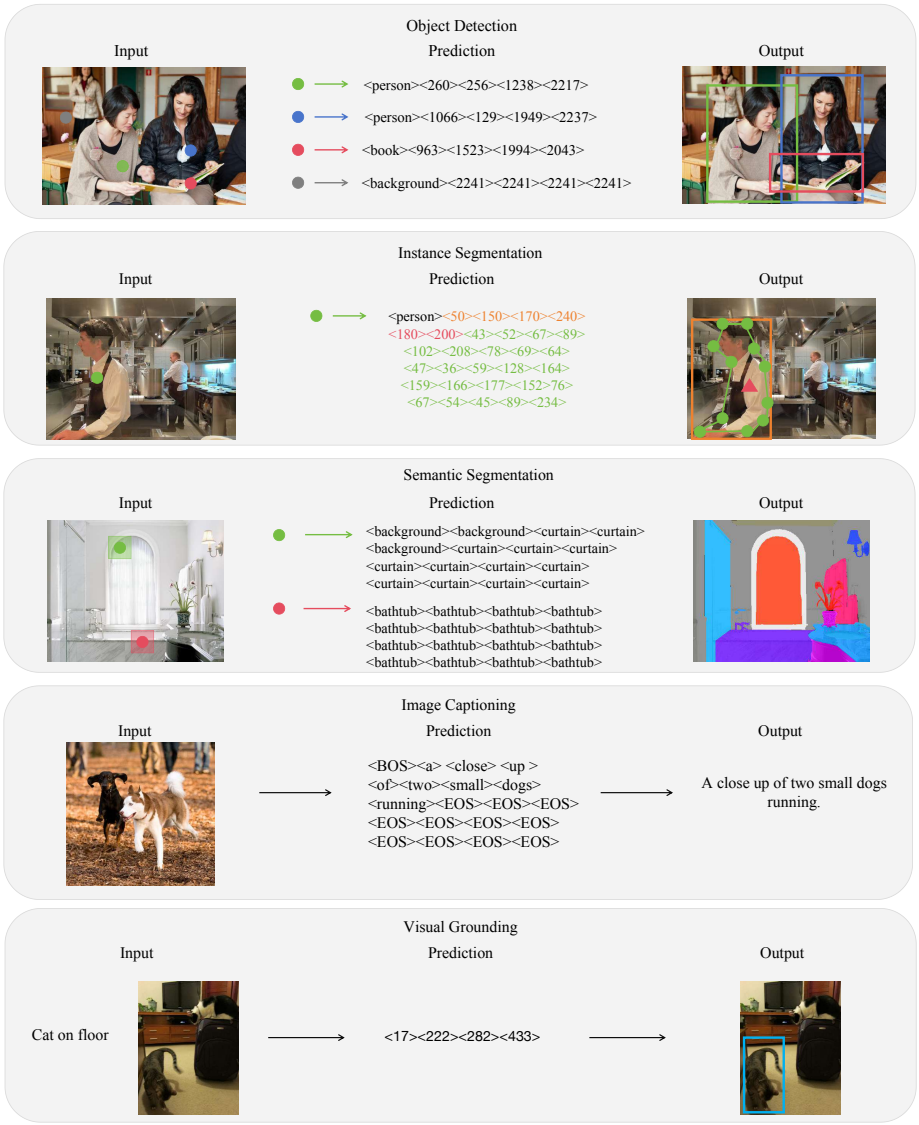

Fig. 9: Visualization of five standard vision-centric tasks.

Fig. 10: Qualitative results on zero-shot datasets. Zoom in for better viewing.

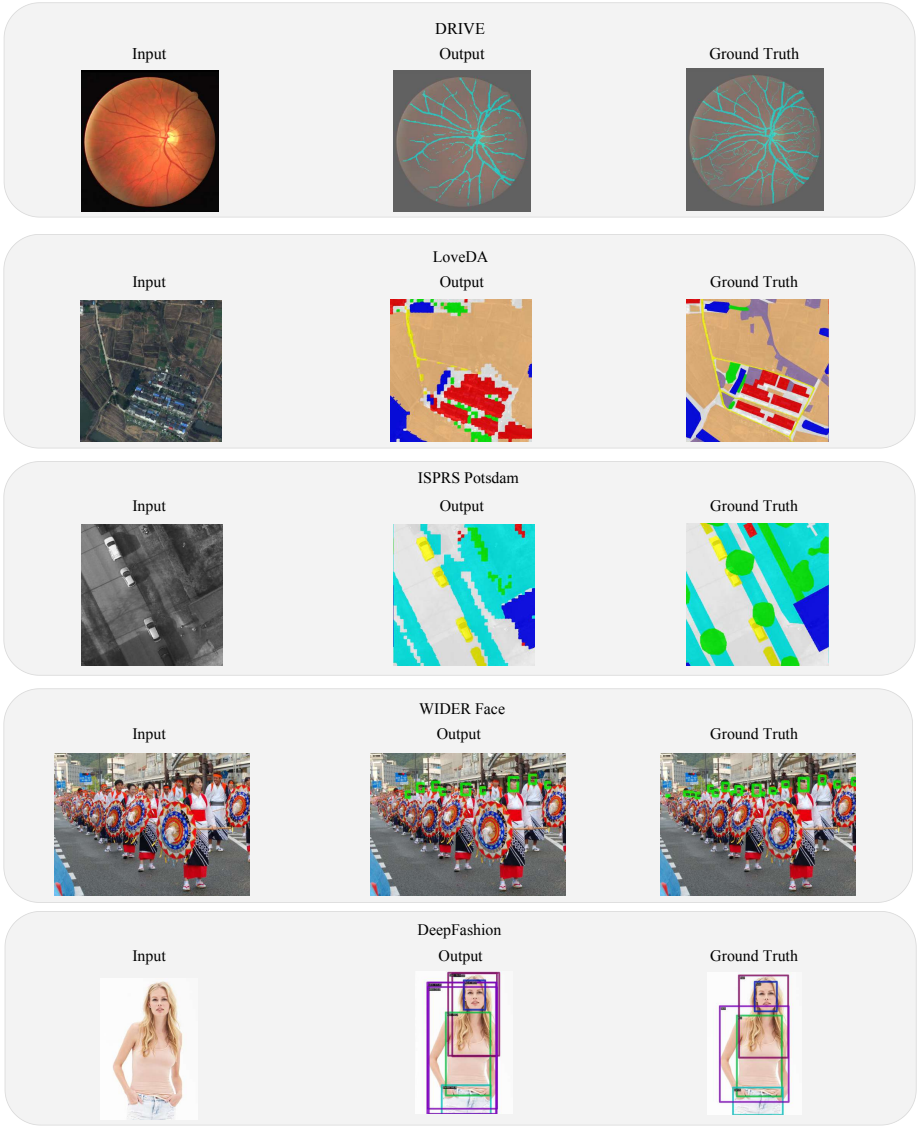

Fig. 11: Qualitative results on few-shot datasets. Zoom in for better viewing.