MiniGPT-v2: Large Language Model as a Unified Interface for Vision-Language Multi-task Learning

Jun Chen [1,2] Deyao Zhu[1] Xiaoqian Shen [1] Xiang Li [1] Zechun Liu [2] Pengchuan Zhang [2]
Raghuraman Krishnamoorthi [2] Vikas Chandra [2] Yunyang Xiong [2] Mohamed Elhoseiny [1]
[1] King Abdullah University of Science and Technology [2] Meta AI Research

Abstract

Large language models have shown their remarkable capabilities as a general interface for various language-related applications. Motivated by this, we target to build a unified interface for completing many vision-language tasks including image description, visual question answering, and visual grounding, among others. The challenge is to use a single model for performing diverse vision-language tasks effectively with simple multi-modal instructions. Towards this objective, we introduce MiniGPT-v2, a model that can be treated as a unified interface for better handling various vision-language tasks. We propose using unique identifiers for different tasks when training the model. These identifiers enable our model to better distinguish each task instruction effortlessly and also improve the model learning efficiency for each task. After the three-stage training, the experimental results show that MiniGPT-v2 achieves strong performance on many visual question-answering and visual grounding benchmarks compared to other vision-language generalist models. Our model and codes are available at https://minigpt-v2.github.io/

1. Introduction

Multi-modal Large Language Models (LLMs) have emerged as an exciting research topic with a rich set of applications in vision-language community, such as visual AI assistant, image captioning, visual question answering (VQA), and referring expression comprehension (REC). A key feature of multimodal large language models is that they can inherit advanced capabilities (e.g., logical reasoning, common sense, and strong language expression) from the LLMs. When tuned with proper vision-language instructions, multi-modal LLMs, specifically vision-language models, demonstrate strong capabilities such as producing detailed image descriptions, generating code, localizing the visual objects in the image, and even performing multi-modal reasoning to better answer complicated visual questions [1, 5, 8]. This evolution of LLMs enables interactions of visual and language inputs across communication with individuals and has been shown quite effective for building visual chatbots.

However, learning to perform multiple vision-language tasks effectively and formulating their corresponding multi-modal instructions present considerable challenges due to the complexities inherent among different tasks. For instance, given a user input *"tell me the location of a person"*, there are many ways to interpret and respond based on the specific task. In the context of the referring expression comprehension task, it can be answered with one bounding box location of the person. For the visual question-answering task, the model might describe their spatial location using human natural language. For the person detection task, the model might identify every spatial location of each human in a given image. To alleviate this issue and towards a unified approach, we propose a task-oriented instruction training scheme to reduce the multi-modal instructional ambiguity, and a vision-language model, MiniGPT-v2. Specifically, we provide a unique task identifier token for each task. For example, we provide a *[vqa]* identifier token for training all the data samples from the visual question answering tasks. In total, we provide six different task identifiers during the model training stages.

Our model, MiniGPT-v2, has a simple architecture design. It directly takes the visual tokens from a ViT vision encoder [3] and project them into the feature space of a large language model [7]. For better visual perception, we utilize higher-resolution images (448x448) during training. But this will result in a larger number of visual tokens. To make the model training more efficient, we concatenate every four neighboring visual tokens into a single token, reducing the total number by 75%. Additionally, we utilize a three-stage training strategy to effectively train our model with a mixture of weakly-labeled, fine-grained image-text datasets, and multi-modal instructional datasets, with different training focus at each stage.

To evaluate the performance of our model, we conducted extensive experiments on diverse vision-language tasks, including (detailed) image/grounded captioning, vision ques-

tion answering, and visual grounding. The results demonstrate that our MiniGPT-v2 can achieve SOTA or comparable performance on diverse benchmarks compared to previous vision-language generalist models, such as MiniGPT-4 [8], InstructBLIP [2], LLaVA [5] and Shikra [1].

2. Method

We start by introducing our vision-language model, MiniGPT-v2, then discuss the basic idea of a multi-task instruction template with task identifiers for training, and finally adapt our task identifier idea to achieve task-oriented instruction tuning.

2.1. Model Architecture

Our proposed model architecture, MiniGPT-v2, is shown in Fig. 1. It consists of three components: a visual backbone, a linear projection layer, and a large language model. We describe each component as follows:

Visual backbone. MiniGPT-v2 adapts the EVA [3] as our visual backbone model backbone. We freeze the visual backbone during the entire model training. We train our model with the image resolution 448x448, and we interpolate the positional encoding to scale with a higher image resolution.

Linear projection layer. We aim to project all the visual tokens from the frozen vision backbone into the language model space. However, for higher-resolution images such as 448x448, projecting all the image tokens results in a very long-sequence input (e.g., 1024 tokens) and significantly lowers the training and inference efficiency. Hence, we simply concatenate 4 adjacent visual tokens in the embedding space and project them together into one single embedding in the same feature space of the large language model, thus reducing the number of visual input tokens by 4 times. With this operation, our MiniGPT-v2 can process high-resolution images much more efficiently during the training and inference stage.

Large language model. MiniGPT-v2 adopts the open-sourced LLaMA2-chat (7B) [7] as the language model backbone. In our work, the language model is treated as a unified interface for various vision-language inputs. We directly rely on the LLaMA-2 language tokens to perform various vision-language tasks. For the visual grounding tasks that necessitate the generation of spatial locations, we directly ask the language model to produce textual representations of bounding boxes to denote their spatial positions.

2.2. Multi-task Instruction Template

When training a single unified model for multiple different tasks such as visual question answering, image caption, referring expression, grounded image caption, and region identification, the multi-modal model might fail to distinguish each task by just aligning visual tokens to language

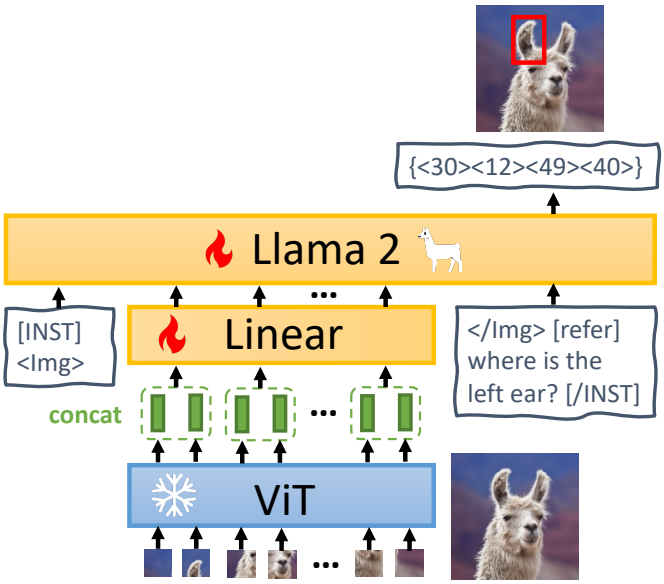

Figure 1. **Architecture of MiniGPT-v2.** The model takes a ViT visual backbone, which remains frozen during all training phases. We concatenate four adjacent visual output tokens from ViT backbone and project them into LLaMA-2 language model space via a linear projection layer.

models. For instance, when you ask "Tell me the spatial location of the person wearing a red jacket?", the model can either respond you the location in a bounding box format (e.g., $< X_{left} >< Y_{top} >< X_{right} >< Y_{bottom} >$) or describe the object location using natural language (e.g., upper right corner). To reduce such ambiguity and make each task easily distinguishable, we introduce task-specific tokens in our designed multi-task instruction template for training. We now describe our multi-task instruction template in more details.

General input format. We follow the LLaMA-2 conversation template design and adapt it for the multi-modal instructional template. The template is denoted as follows,

[INST] <Img> < ImageFeature> </Img> [Task Identifier] Instruction [/INST]

In this template, *[INST]* is considered as the user role, and *[/INST]* is considered as the assistant role. We structure the user input into three parts. The first part is the image features, the second part is the task identifier token, and the third part is the instruction input.

Task identifier tokens. Our model takes a distinct identifier for each task to reduce the ambiguity across various tasks. As illustrated in Table 1, we have proposed six different task identifiers for visual question answering, image caption, grounded image captioning, referring expression comprehension, referring expression generation, and phrase parsing and grounding respectively. For vision-irrelevant instructions, our model does not use any task identifier token.

Tasks	VQA	Caption	Grounded Caption	REC	REG	Object Parsing and Grounding
Identifiers	[vqa]	[caption]	[grounding]	[refer]	[identify]	[detection]

Table 1. Task identifier tokens for 6 different tasks, including visual question answering, image captioning, grounded image captioning, referring expression comprehension (REC), referring expression generation (REG), and object parsing and grounding

Spatial location representation. For tasks such as referring expression comprehension (REC), referring expression generation (REG), and grounded image captioning, our model is required to identify the spatial location of the referred objects accurately. We represent the spatial location through the textual formatting of bounding boxes in our setting, specifically: "$\{< \mathrm{X}_{left} >< \mathrm{Y}_{top} >< \mathrm{X}_{right} >< \mathrm{Y}_{bottom} >\}$". Coordinates for X and Y are represented by integer values normalized in the range [0,100]. $< \mathrm{X}_{left} >$ and $< \mathrm{Y}_{top} >$ denote the x and y coordinate top-left corner of the generated bounding box, and $< \mathrm{X}_{right} >$ and $< \mathrm{Y}_{bottom} >$ denote the x and y coordinates of the bottom-right corner.

2.3. Multi-task Instruction Training

We now adapt our designed multi-task instruction template for instruction training. The basic idea is to take instruction with task-specific identifier token as input for task-oriented instruction training of MiniGPT-v2. When input instructions have task identifier tokens, our model will become more prone to multiple-task understanding during training. We train our model with task identifier instructions for better visual aligment in three stages. The first stage is to help MiniGPT-v2 build broad vision-language knowledge through many weakly-labeled image-text datasets, and high-quality fine-grained vision-language annotation datasets as well (where we will assign a high data sampling ratio for weakly-labeled image-text datasets). The second stage is to improve the model with only fine-grained data for multiple tasks. The third stage is to finetune our model with more multi-modal instruction and language datasets for answering diverse multi-modal instructions better and behaving as a multi-modal chatbot.

Stage 1: Pretraining. To have broad vision-language knowledge, our model is trained on a mix of weakly-labeled and fine-grained datasets. We give a high sampling ratio for weakly-labeled datasets to gain more diverse knowledge in the first-stage.

For the weakly-labeled datasets, we use LAION, CC3M, SBU, and GRIT-20M from Kosmos v2 that built the dataset for referring expression comprehension (REC), referring expression generation (REG), and grounded image captioning.

For fine-grained datasets, we use datasets like COCO captionand Text Captions for image captioning, RefCOCO, RefCOCO+, and RefCOCOg for REC. For REG, we restructured the data from ReferCOCO and its variants, reversing the order from phrase $\rightarrow$ bounding boxes to bounding boxes $\rightarrow$ phrase. For VQA datasets, our training takes a variety of datasets, such as GQA, VQA-v2, OCR-VQA, OK-VQA, and AOK-VQA.

Stage 2: Multi-task training. To improve the performance of MiniGPT-v2 on each task, we only focus on using fine-grained datasets to train our model at this stage. We exclude the weakly-supervised datasets such as GRIT-20M and LAION from stage-1 and update the data sampling ratio according to the frequency of each task. This strategy enables our model to prioritize high-quality aligned image-text data for superior performance across various tasks.

Stage 3: Multi-modal instruction tuning. Subsequently, we focus on tuning our model with more multi-modal instruction datasets and enhancing its conversation ability as a chatbot. We continue using the datasets from the second stage and add instructional datasets, including LLaVA [5], Flickr30k dataset [6], our constructed mixing multi-task dataset, and the language dataset, Unnatural Instruction. We give a lower data sampling ratio for the fine-grained datasets from stage-2 and a higher data sampling ratio for the new instruction datasets.

3. Experiments

3.1. Quantitative Evaluation

We evaluate our model on VQA and referring expression comprehension benchmarks.

Visual question answering results. Table 2 presents our experimental results on multiple VQA benchmarks. Our results compare favorably to baselines including MiniGPT-4 [8], Shikra [1], LLaVA [5], and InstructBLIP [2] across all the VQA tasks. For example, on QKVQA, our MiniGPT-v2 outperforms MiniGPT-4, Shikra, LLaVA, and BLIP-2 by 20.3%, 10.6%, 3.4%, and 11.9%. These results indicate the strong visual question answering capabilities of our model. Furthermore, we find that our MiniGPT-v2 (chat) variant shows higher performance than the version trained after the second stage. On OKVQA, VSR, IconVQA, VizWiz, and HM, MiniGPT-v2 (chat) outperforms MiniGPT-v2 by 0.9%, 2.3%, 4.2%, 20.7%, and 0.6%. We believe that the better performance can be attributed to the improved language skills during the third-stage training, which is able to benefit visual question comprehension and response, especially on VizWiz with 20.7% top-1 accuracy increase.

Referring expression comprehension results. Table 3 compares our model to baselines on REC benchmarks. Our MiniGPT-v2 shows strong REC performance on RefCOCO, RefCOCO+, and RefCOCOg, performing better than other vision-language generalist models. MiniGPT-v2 outperforms OFA-L by over 8% accuracy across all tasks of Re-

Method	Grounding	OKVQA	GQA	VSR (zero-shot)	IconVQA (zero-shot)	VizWiz (zero-shot)	HM (zero-shot)
Flamingo-9B	✗	44.7	-	31.8	-	28.8	57.0
BLIP-2 (13B)	✗	45.9	41.0	50.9	40.6	19.6	53.7
InstructBLIP (13B)	✗	-	49.5	52.1	44.8	33.4	57.5
MiniGPT-4 (13B)	✗	37.5	30.8	41.6	37.6	-	-
LLaVA (13B)	✗	54.4	41.3	51.2	43.0	-	-
Shikra (13B)	✓	47.2	-	-	-	-	-
Ours (7B)	✓	56.9	**60.3**	60.6	47.7	32.9	58.2
Ours (7B)-chat	✓	**57.8**	60.1	**62.9**	**51.5**	**53.6**	**58.8**

Table 2. **Results on multiple VQA tasks.** We report top-1 accuracy for each task. Grounding column indicates whether the model incorporates visual localization capability. The best performance for each benchmark is indicated in **bold**.

Method	Model types	RefCOCO			RefCOCO+			RefCOCOg		Avg
		val	test-A	test-B	val	test-A	test-B	val	test	
UNINEXT	Specialist models	92.64	94.33	91.46	85.24	89.63	79.79	88.73	89.37	88.90
G-DINO-L		90.56	93.19	88.24	82.75	88.95	75.92	86.13	87.02	86.60
VisionLLM-H	Generalist models	-	86.70	-	-	-	-	-	-	-
OFA-L		79.96	83.67	76.39	68.29	76.00	61.75	67.57	67.58	72.65
Shikra (7B)		87.01	90.61	80.24	81.60	87.36	72.12	82.27	82.19	82.93
Shikra (13B)		87.83	91.11	81.81	**82.89**	**87.79**	74.41	82.64	83.16	83.96
Ours (7B)		**88.69**	**91.65**	**85.33**	79.97	85.12	**74.45**	**84.44**	**84.66**	**84.29**
Ours (7B)-chat		88.06	91.29	84.30	79.58	85.52	73.32	84.19	84.31	83.70

Table 3. **Results on referring expression comprehension tasks.** Our MiniGPT-v2 outperforms many VL-generalist models including VisionLLM, OFA and Shikra [1] and reduces the accuracy gap comparing to specialist models including UNINEXT and G-DINO.

fCOCO/RefCOCO+/RefCOCOg. Compared with a strong baseline, Shikra (13B), our model still shows better results, e.g., 84.29% vs 83.96% accuracy in average. These results provide direct evidence for the competing visual grounding capabilities of MiniGPT-v2. Although our model underperforms specialist models, the promising performance indicates its growing competence in visual grounding.

4. Conclusion

In this paper, we introduce MiniGPT-v2, a multi-modal LLM that can serve as a unified interface for various vision-language multi-tasking learning. To develop a single model capable of handling multiple vision-language tasks, we propose using distinct identifiers for each task during the training and inference. These identifiers help our model easily differentiate various tasks and also improve learning efficiency. Our MiniGPT-v2 achieves state-of-the-art results across many visual question answering and referring expression comprehension benchmarks. We also found that our model can efficiently adapt to new vision-language tasks, which suggests that MiniGPT-v2 has many potential applications in the vision-language community.

References

[1] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1, 2, 3, 4

[2] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2, 3

[3] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. 1, 2

[4] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 1

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2, 3

[6] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 3

[7] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2

[8] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3

MiniGPT-v2: Large Language Model as a Unified Interface for Vision-Language Multi-task Learning

Supplementary Material

E. Hallucination Evaluation

Throughout three training stages, we maintained a batch size of 4 and utilized the AdamW optimizer in conjunction with a cosine learning rate scheduler, setting the learning rate to 1e-4. Our visual backbone is EVA-CLIP, with the frozen weights. Notably, we trained the linear projection layer and performed efficient fine-tuning of the language model using LoRA. Specifically, we fine-tuned the W_q and W_v components with a rank (r) of 64 and a LoRA-alpha value equal 16. The entire model was trained with a consistent image resolution of 224×224 pixels, ensuring uniformity across all stages.

We measure the hallucination of our model on image description generation and compare the results with other vision-language baselines, including MiniGPT-4, mPLUG-Owl, LLaVA, and MultiModal-GPT. Following the methodology from [4], we use CHAIR to assess hallucination at both object and sentence levels. As shown in Table 4, we find that our MiniGPT-v2 tends to generate the image description with reduced hallucination compared to other baselines. We have evaluated three types of prompts in MiniGPT-v2. First, we use the prompt *generate a brief description of the given image* without any specific task identifier which tends to produce more detailed image descriptions. Then we provide the instruction prompt *[grounding] describe this image in as detailed as possible* for evaluating grounded image captions. Lastly, we prompt our model with *[caption] briefly describe the image*. With these task identifiers, MiniGPT-v2 is able to produce a variety of image descriptions with different levels of hallucination. As a result, all these three instruction variants have lower hallucination than our baseline, especially with the task specifiers of *[caption]* and *[grounding]*.

Method	CHAIR$_I$ ↓	CHAIR$_S$ ↓	Len
MiniGPT-4	9.2	31.5	116.2
mPLUG-Owl	30.2	76.8	98.5
LLaVA	18.8	62.7	90.7
MultiModal-GPT	18.2	36.2	45.7
MiniGPT-v2 (long)	8.7	25.3	56.5
MiniGPT-v2 (grounded)	7.6	12.5	18.9
MiniGPT-v2 (short)	**4.4**	**7.1**	**10.3**

Table 4. **Results on hallucination.** We evaluate the hallucination of MiniGPT-v2 with different instructional templates and output three versions of captions for evaluation. For the "long" version, we use the prompt *generate a brief description of the given image*. For the "grounded" version, the instruction is *[grounding] describe this image in as detailed as possible*. For the "short" version, the prompt is *[caption] briefly describe the image*.

	OKVQA	GQA	WizViz	VSR	IconVQA	HM	Average
w/o task identifier	50.5	53.4	28.6	57.5	44.8	56.8	48.6
w task identifier	**52.1**	**54.6**	**29.4**	**59.9**	**45.6**	**57.4**	**49.8**

Table 5. Task identifier ablation study on VQA benchmarks. With task identifier during the model training can overall improve VQA performances from multiple VQA benchmarks

F. Ablation on Task Identifier

We conduct ablation studies on the effect of the task identifier on the performance of MiniGPT-v2. We compare our model with the variant without using task identifiers on VQA benchmarks. Both models were trained on 4xA100 GPUs for 24 hours with an equal number of training steps for multiple vision-language tasks. Results in Table 5 demonstrate the performance on multiple VQA benchmarks and consistently show that token identifier training benefits the overall performance of MiniGPT-v2. Specifically, our MiniGPT-v2 with task-oriented instruction training achieves 1.2% top-1 accuracy improvement on average. These ablation results can validate the clear advantage of adding task identifier tokens and support the use of multi-task identifiers for multi-task learning efficiency.

G. Qualitative Results

We now provide the qualitative results for a complementary understanding of our model's multi-modal capabilities. Some examples can be seen in Fig. 2. Specifically, we demonstrated various abilities in the examples including a) object identification; b) detailed grounded image captioning; c) visual question answering; d) referring expression comprehension; e) visual question answering under task identifier; f) detailed image description; g) object parsing and grounding from an input text. More qualitative results can be found in the Appendix. These results demonstrate that our model has competing vision-language understanding capabilities. Moreover, notice that we train our model only with a few thousand of instruction samples on object parsing and grounding tasks at the third-stage, and our model can effectively follow the instructions and generalize on the new task. This indicates that our model has the flexibility to adapt on many new tasks.

Note that our model still occasionally shows halluci-

nations when generating the image description or visual grounding. e.g., our model may sometimes produce descriptions of non-existent visual objects or generate inaccurate visual locations of grounded objects. We believe training with more high-quality image-text aligned data and integrating with a stronger vision backbone or large language model hold the potential for alleviating this issue.

Figure 2. **Examples for various multi-modal capabilities of MiniGPT-v2.** We showcase that our model is capable of completing multiple tasks such as referring expression comprehension, referring expression generation, detailed grounded image caption, visual question answering, detailed image description, and directly parsing phrase and grounding from a given input text.