MULTIINSTRUCT: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning

Zhiyang Xu*, **Ying Shen***, **Lifu Huang**
Computer Science Department
Virginia Tech
{zhiyangx, yings, lifuh}@vt.edu

Abstract

Instruction tuning, a new learning paradigm that fine-tunes pre-trained language models on tasks specified through instructions, has shown promising zero-shot performance on various natural language processing tasks. However, it's still not explored for vision and multimodal tasks. In this work, we introduce MULTIINSTRUCT, the first multimodal instruction tuning benchmark dataset that consists of 47 diverse multimodal tasks covering 11 broad categories. Each task is designed at least with 5,000 instances (input-out pairs) from existing open-source datasets and 5 expert-written instructions. We take OFA (Wang et al., 2022a) as the base pre-trained model for multimodal instruction tuning, and to improve its performance, we explore multiple transfer learning strategies to leverage the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022). Experimental results demonstrate its strong zero-shot performance on various unseen multimodal tasks and the benefit of transfer learning from text-only instructions. We also design a new evaluation metric – *Sensitivity*, to evaluate how sensitive the model is to the variety of instructions. Our results indicate that the model is less sensitive to the varying instructions after finetuning on a diverse set of tasks and instructions for each task.

1 Introduction

With the advances in large-scale pre-trained language models (PLMs), recent studies have explored various efficient learning paradigms (Brown et al., 2020; Liu et al., 2021; Wei et al., 2021; Xie et al., 2021) to generalize PLMs to new tasks without task-specific tuning. Among these, instruction tuning (Wei et al., 2021) has achieved significant success in zero-shot learning on natural language processing tasks. By fine-tuning a PLM on tasks described through instructions, instruction tuning allows the model to learn to understand and follow the instructions to perform predictions on unseen tasks. Recent advancement in multimodal pre-training (Wang et al., 2022a; Alayrac et al., 2022; Bao et al., 2022; Wang et al., 2022c) has shown the potential of jointly interpreting text and images in a shared semantic space, which further leads us to ask: can the instruction tuning be leveraged to improve the generalizability of PLMs on vision and multi-modal tasks?

In this work, we propose MULTIINSTRUCT, the first benchmark dataset for multimodal instruction tuning with 47 diverse tasks from 11 broad categories. MULTIINSTRUCT covers most of the multimodal tasks that require visual understanding and multimodal reasoning, such as Visual Question Answering (Goyal et al., 2017; Zhu et al., 2016; Suhr et al., 2017), Image Captioning (Lin et al., 2014), Image Generation (Changpinyo et al., 2021), Visual Relationship Understanding (Krishna et al., 2017) and so on. For each task, we create at least 5,000 instances (i.e., input-output pairs) and 5 instructions that are manually written by two experts in natural language processing. As shown in Figure 1, we formulate all the tasks into a unified sequence-to-sequence format in which the input text, images, instructions, and bounding boxes are represented in the same token space.

We use OFA (Wang et al., 2022a), a unified model that is pre-trained on a diverse set of cross-model and unimodal tasks in a single Transformer-based sequence-to-sequence framework, as the base pre-trained multimodal language model, and fine-tune it on MULTIINSTRUCT. Considering that the scale of NATURAL INSTRUCTIONS (Mishra et al., 2022), a text-only instruction tuning dataset, is much larger than MULTIINSTRUCT, we further explored several transfer learning strategies, including *Mixed Instruction Tuning*, *Sequential Instruction Tuning*, and *Adapter-based Sequential Instruction Tuning*. Experimental results demon-

* Zhiyang Xu and Ying Shen contributed equally to this work.

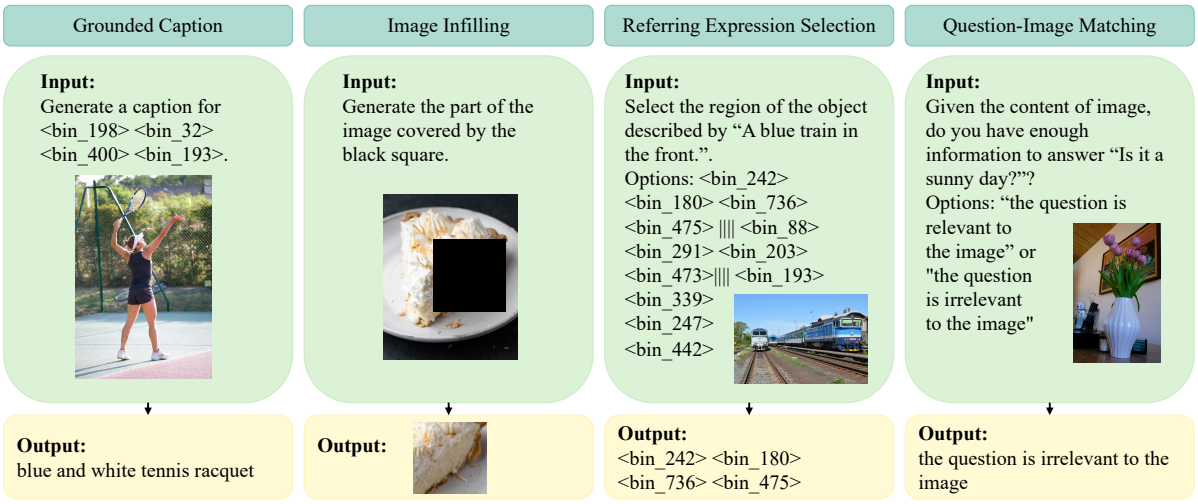

Figure 1: **Example Instances from MULTIINSTRUCT for Four Tasks.**

strate strong zero-shot performance on various unseen multimodal tasks with instruction tuning and the potential of further improving it by leveraging large-scale text-only instruction datasets.

As instruction tuning highly relies on the interpretation of instructions for various tasks, we also develop a new metric – *Sensitivity*, to measure how sensitive the model is toward the variety of instructions for the same task. Experimental results demonstrate that (1) instruction tuning significantly reduces the sensitivity of OFA to the varying wording of instructions, and (2) it's also beneficial to reduce the sensitivity by adopting a diverse set of instructions and tasks for instruction tuning.

2 Related Work

Multimodal Pretraining Multimodal pretraining (Tan and Bansal, 2019; Cho et al., 2021; Singh et al., 2022; Alayrac et al., 2022; Wang et al., 2022a; Li et al., 2022b,a) has gained increasing attention in recent years and significantly advanced the downstream vision-language tasks. Several recent studies (Cho et al., 2021; Wang et al., 2022a,c; Lu et al., 2022) also started to explore building a unified pre-training framework to handle a diverse set of cross-modal and unimodal tasks. Among them, VL-T5 (Cho et al., 2021) tackles vision-and-language tasks with a unified text-generation objective conditioned on multimodal inputs, while OFA (Wang et al., 2022a) further extends the text-generation objective to image generation tasks by using a unified vocabulary for all text and visual tokens. Similarly, Unified-IO (Lu et al., 2022) uses a unified vocabulary to homogenize all input and output modalities but applies the architecture to a wider set of classical computer vision tasks. BEIT-3 (Wang et al., 2022c) uses a unified masked modality modeling approach to process both unimodal and multimodal data. It utilizes a novel shared Multiway Transformer network with a shared self-attention module that is able to learn how to align different modalities. Compared with all these studies, our work focuses on creating a large-scale multimodal instruction tuning benchmark dataset and improving the generalization and zero-shot performance on various unseen multimodal tasks.

Efficient Language Model Tuning To improve the generalizability and adaptivity of large-scale pre-trained language models, various efficient language model tuning strategies have been proposed recently. Prompt tuning aims to learn a task-specific prompt while keeping most of the parameters of the model freezed (Liu et al., 2021; Li and Liang, 2021; Han et al., 2022; Wang et al., 2022b). It requires reformulating the downstream tasks to the format that the model was initially trained on, and has shown competitive performance in a wide variety of applications in natural language processing. As a special form of prompt tuning, in-context learning takes one or a few examples as the prompt to demonstrate the task. Xie et al. (2021) provide theoretical insights to explain why in-context learning works. MetaICL (Min et al., 2021) is a new meta-training framework for few-shot learning, in which a pre-trained language model is fine-tuned to perform in-context learning on a large set of training tasks. Min et al. (2022) show that in-context learning does not rely on the correctness of

demonstrations but instead benefits from knowing the output label space through empirical studies.

Instruction tuning is another simple yet effective strategy to improve the generalizability of large language models. Wei et al. (2021) first propose FLAN to fine-tune language models on various tasks described by natural language instructions and show significant zero-shot performance on unseen tasks. Instruct-GPT (Ouyang et al., 2022) first finetunes GPT-3 (Brown et al., 2020) on crowdsourcing instructions with supervised learning and then continues to optimize it via human feedback. Natural-Instructions (Mishra et al., 2022; Wang et al., 2022d) is a meta-dataset containing 61 distinct tasks with human-authored definitions, things to avoid, and demonstrations. Researchers (Mishra et al., 2022; Wang et al., 2022d) have shown that NATURAL INSTRUCTIONS can greatly improve the generalizability of language models even when the size is relatively small (e.g., BART_base). Webson and Pavlick (2022) investigate whether the language model understands the task instructions in the way humans do and show that with irrelevant and misleading instructions, the models still achieve similar performance as good instructions in the few-shot learning setting. InstructDial (Gupta et al., 2022) applies instruction tuning to the dialogue domain and shows significant zero-shot performance on unseen dialogue tasks. While these studies have been successful in text-only domains, it has not yet been extensively explored for vision or multimodal tasks.

3 MULTIINSTRUCT

MULTIINSTRUCT is a new multi-modal instruction tuning dataset that consists of 47 tasks derived from 54 datasets and each task has five unique instructions. Figure 1 shows four example multimodal instructions. In the following section, we introduce the multimodal instruction formatting (Section 3.3), the process of data construction, and the statistics of the data.

3.1 Multimodal Task and Data Collection

The MULTIINSTRUCT dataset is designed to cover a wide range of multimodal tasks that require reasoning among regions, image, and text. These tasks are meant to teach machine learning models to learn fundamental skills such as object recognition, visual relationship understanding, text-image grounding, and so on. To build MULTIINSTRUCT, we first collect 26 tasks from the existing studies in visual and multimodal learning that require these skills, covering Visual Question Answering (Goyal et al., 2017; Krishna et al., 2017; Zhu et al., 2016; Suhr et al., 2017; Liu et al., 2022; Hudson and Manning, 2019; Singh et al., 2019; Marino et al., 2019), Image Captioning (Lin et al., 2014), Grounded Generation (Krishna et al., 2017; Yu et al., 2016; Lin et al., 2014), Image-Text Matching (Lin et al., 2014; Goyal et al., 2017), Grounded Matching (Krishna et al., 2017; Veit et al., 2016; Yu et al., 2016), Visual Relationship (Krishna et al., 2017), Image Attributes (Chiu et al., 2020), Image Generation (Changpinyo et al., 2021), Commonsense Reasoning (Zellers et al., 2019; Xie et al., 2019), Temporal Ordering tasks that are created from WikiHow[1], and Miscellaneous (Yao et al., 2022; Kiela et al., 2020; Das et al., 2017). Each of the 26 tasks can be found with one or multiple open-source datasets, which are incorporated into MULTIINSTRUCT. Details of each task and their corresponding datasets are shown in Tables 7 to 9 in Appendix.

For each of these tasks, we examined the possibility of decomposing the task into simpler or related tasks to further test and improve the model's capabilities. For example, *Visual Grounding* requires the model to generate a caption for a given region in the image. We derive two additional tasks that are related to this complex skill: *Grounded Caption Selection*, which is a simpler skill that requires the model to select the corresponding caption from multiple candidates for the given region, and *Visual Grounding Selection*, which requires the model to select the corresponding region from the provided candidate regions based on a given caption. These two tasks are meant to improve the model's capability on similar but simpler skills that are needed to complete the *Visual Grounding* task. In this way, we further derived 21 new additional tasks from the 26 existing tasks. We divide all 47 tasks into 11 broad categories as shown in Figure 2.

For the existing tasks, we use their available open-source datasets to create instances (i.e., input and output pairs) while for each new task, we create its instances by extracting the necessary information from the instances of existing tasks or reformulating them. In this way, each new task is created with 5,000 to 5M instances. We split the 47 tasks into training and evaluation based on the following

[1] https://www.wikihow.com.

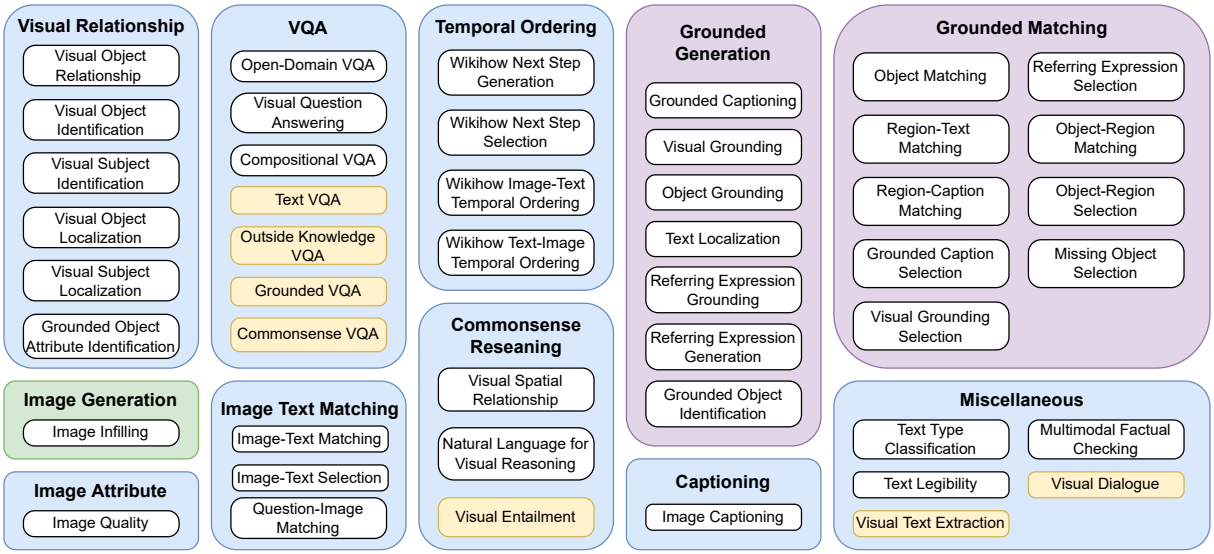

Figure 2: **Task clusters included in MULTIINSTRUCT.** The yellow boxes denote tasks used for evaluation and the white boxes denote tasks used for training.

criteria: (1) we take the tasks that are similar to the pre-training tasks of OFA (Wang et al., 2022a) for training; and (2) we also take the tasks that require simple or common skills for training and complex tasks for evaluation, based on the judgment of two expert researchers in Natural Language Processing. To make the training instances of different tasks to be balanced, we sample a maximum of 10,000 instances for each training task while keeping all the instances of evaluation tasks for testing. Tables 7 to 9 show the details of all training and evaluation tasks along with their corresponding datasets.

3.2 Task Instruction Creation

We first provide a definition for "*instruction*" used in MULTIINSTRUCT. An *instruction* is defined with a template that describes how the task should be performed and contains an arbitrary number of placeholders, including `<TEXT>`, `<REGION>` and `<OPTION>`, for the input information from the original task. For example, in the instruction of the Grounded Captioning task, "Generate a caption for `<REGION>`", `<REGION>` is the placeholder for region-specific information. Note that the placeholder `<OPTION>` is only used in classification tasks and for some tasks, the input may also include an image that is not included in the instruction and will be fed as a separate input to the model. Figure 1 provides several instruction examples for the tasks included in MULTIINSTRUCT.

To produce high-quality instructions that accurately convey the intended tasks, we employ an iterative annotation process involving two expert annotators who have a thorough understanding of the task and the dataset.

Step 1: each annotator first writes 2-3 instructions for each task. These annotators are provided with clear and detailed information about the task and the dataset, including the specific goals of this task, the format of the input data, and 10 example instances randomly sampled from the dataset. The details about the dataset are typically obtained from either the dataset's README file or the original publication that introduced the dataset. For tasks that are newly derived, we provide task descriptions along with 10 constructed example instances to the annotators.

Step 2: to guarantee the quality of the instructions and make sure that they are effective for the intended tasks, we have each annotator review the instruction created by the other, checking if they can clearly understand and identify the intended task by just reading the instruction. If any issues are identified, the reviewing annotator provides suggestions and works with the original annotator to revise the instructions.

Step 3: to ensure the instructions from different annotators do not conflict with or repeat each other, we have both annotators review the sets of instructions together and identify any discrepancies or inconsistencies. If any are found, the annotators work together to resolve them and create a final set of instructions that accurately and clearly describe the task. In this way, each task will be created with

5 high-quality instructions.

Step 4: we repeat steps 1-3 to create 5 instructions for each of the training and evaluation tasks. Finally, both annotators review each task and its instructions and filter out the task that is not representative or overlaps with other tasks.

3.3 Multimodal Instruction Formatting

To unify the processing of various input/output data types, we follow the method from OFA (Wang et al., 2022a), which involves representing images, text, and bounding box coordinates as tokens in a unified vocabulary. Specifically, we apply byte-pair encoding (BPE) (Sennrich et al., 2016) to encode the text input. For the target image, we apply VQ-GAN (Esser et al., 2021) to generate discrete image tokens through image quantization. To represent regions or bounding boxes of an image, we discretize the four corner coordinates into location tokens such as "<bin_242> <bin_180> <bin_736> <bin_475>" where each location token "<bin_NUM>" represents a quantized coordinate obtained by dividing the image into 1000 bins. This approach allows us to convert different types of input into a unified vocabulary.

All tasks in MULTIINSTRUCT can then be formulated as natural language sequence-to-sequence generation problems, where the input includes: (1) an image (if there is no input image, a black picture is used as the input); and (2) an instruction where the placeholders such as `<TEXT>`, `<REGION>` or `<OPTION>` are filled with specific information of each input instance. Notably, for the `<OPTION>` of the instructions for classification tasks, we introduce two special tokens for this field: "`[Options]`" to mark the beginning of the option field and "`||||`" to delimit the given options. We concatenate all the options with "`||||`" in the option field and the model will directly generate one option from them. Figure 1 provides several examples of the formulated input and illustrates how the original data input is combined with the instruction in the MULTIINSTRUCT.

3.4 Statistics of MULTIINSTRUCT

Table 1 shows the distribution of input and output modalities for both training and evaluation tasks in MULTIINSTRUCT, and Table 2 shows the detailed statistics for all the training and evaluation tasks separately.

Input modality			Output Modality			# of Training	# of Testing
Image	Text	Region	Image	Text	Region		
✓			✓			1	0
✓				✓		1	0
✓	✓			✓		14	5
✓		✓		✓		9	1
✓		✓			✓	2	0
✓	✓				✓	3	1
✓	✓	✓		✓		9	0
✓	✓	✓			✓	1	0

Table 1: Distribution of input and output modalities for all the tasks in MULTIINSTRUCT.

	Train	Eval
Average # of Tokens per Instruction	14.67	9.37
Averaged # of Character per Instruction	85.78	58.77
Average Levenshtein Distance of Instructions	63.63	54.74
# of Instructions per Task	5	5
# of Classification Tasks	21	3
# of Generation Tasks	19	4
# of Existing Tasks	19	7
# of Created Datasets	21	0

Table 2: Detailed statistics in MULTIINSTRUCT.

4 Problem Setup and Models

4.1 Problem Setup

We follow the same instruction tuning setting as previous studies (Wei et al., 2021) and mainly evaluate the zero-shot learning capabilities of the fine-tuned large language models. Specifically, given a pre-trained multimodal language model M, we aim to finetune it on a collection of instruction tasks T. Each task $t \in T$ is associated with a number of training instances $\mathcal{D}^t = \{(I^t, x_j^t, y_j^t) \in \mathcal{I}^t \times \mathcal{X}^t \times \mathcal{Y}^t\}_{j=1}^N$, where x_j^t denotes the input text, image, region, and options if provided, y_j^t denotes the output of each instance, and I^t represents the set of five task instructions written by experts. The input information from x_j^t will be used to fill in the placeholders in the instruction.

We use OFA (Wang et al., 2022a) as the pre-trained multimodal model as it's a unified architecture with flexible input-output modalities. We further finetune it on our MULTIINSTRUCT dataset to demonstrate the effectiveness of instruction tuning. Specifically, as OFA is a transformer-based seq2seq framework, we use its transformer encoder to encode the instruction with all the necessary filled information and an optional image and predict the output with the transformer decoder. Given that the training dataset contains many tasks, we mix all the training instances from these tasks and randomly shuffle them. For each instance, we also randomly sample an instruction template for each batch-based training. Note that, though some of

the training tasks in MULTIINSTRUCT are similar to the pre-training tasks of OFA[2], we ensure that the evaluation tasks in MULTIINSTRUCT do not overlap with either the pre-training tasks in OFA or the training tasks in MULTIINSTRUCT.

4.2 Transfer Learning from NATURAL INSTRUCTIONS

We notice that the scale of NATURAL INSTRUCTIONS (Mishra et al., 2022; Wang et al., 2022d) is significantly larger than MULTIINSTRUCT, which shows the potential of transferring the instruction learning capability from the larger set of natural language tasks (i.e., 894 English tasks defined in NATURAL INSTRUCTIONS (Mishra et al., 2022)) to multimodal tasks. Here, we explore several simple strategies:

Mixed Instruction Tuning: We combine the instances of NATURAL INSTRUCTIONS and MULTIINSTRUCT and randomly shuffle them before finetuning OFA with instructions. Note that, each task in NATURAL INSTRUCTIONS is just associated with one instruction while for each instance from MULTIINSTRUCT, we always randomly sample one instruction from the five instructions for each instance of training.

Sequential Instruction Tuning Inspired by the Pre-Finetuning approach discussed in Aghajanyan et al. (2021), we propose a two-stage sequential instruction tuning strategy where we first fine-tune OFA on the NATURAL INSTRUCTIONS dataset to encourage the model to follow instructions to perform language-only tasks, and then further finetune it on MULTIINSTRUCT to adapt the instruction learning capability to multimodal tasks. To best leverage the NATURAL INSTRUCTIONS dataset, we use all instances in English-language tasks to tune the model in the first training stage.

Adapter-based Sequential Instruction Tuning Tuning the whole large pre-trained language model is computationally expensive and impractical as the size of the model and training tasks/instances grows. Considering this, we further propose a parameter-efficient instruction tuning strategy by introducing adapters (Houlsby et al., 2019). Specifically, we insert an adapter after the feedforward layer in each transformer layer in the decoder and encoder of OFA. The adapters are based on the multi-layer-perceptron structure which consists of a feedforward layer, followed by a non-linear activation layer and another feedforward layer. To reduce the number of training parameters, we downsize the input features size by four times which is 256 given the 1024 feature dimension in OFA. During training, we freeze the parameters of OFA and sequentially optimize the parameters of the adapter on NATURAL INSTRUCTIONS and MULTIINSTRUCT. During inference, we directly apply the adapter-based OFA to perform zero-shot prediction on various unseen tasks.

[2]Table 10 in Appendix lists the multimodal tasks and dataset used in OFA pre-training.

5 Experimental Setup

5.1 Evaluation Metrics

We report the accuracy for classification tasks and ROUGE-L (Lin, 2004) for all generation tasks and some classification tasks following Mishra et al. (2022), which treats all tasks as text generation problems. For each task, we evaluate the model using one of the five instructions, resulting in a total of five experiments. We report the mean and maximum performance across all five experiments. We also compute the *aggregated performance* for each model where we take the average of the model's performance score on all unseen tasks. For most tasks, we use Rouge-L as the performance score. For tasks that only have accuracy as a metric, we use accuracy to compute the aggregated performance.

In addition, as instruction tuning mainly relies on the instructions to guide the model to perform prediction on various unseen multimodal tasks, we further propose to evaluate how sensitive the model is to the variety of human-written instructions, which has not been discussed in previous instruction tuning studies but is necessary to understand the effectiveness of instruction tuning. We thus further design a new metric as follows:

Sensitivity which refers to the model's capability of consistently producing the same results, regardless of slight variations in the wording of instructions, as long as the intended task remains the same. Specifically, for each task $t \in T$, given its associated instances $\mathcal{D}^t = \{(x_j^t, y_j^t) \in \mathcal{X}^t \times \mathcal{Y}^t\}_{j=1}^N$ and task instructions I^t, we define the Intra-task sensitivity as:

$$\mathbb{E}_{t \in T}\left[\frac{\mu_{i \in I^t}\left[\mathbb{E}_{(x,y)\in\mathcal{D}^t}[\mathcal{L}(f_\theta(i,x),y)]\right]}{\sigma_{i \in I^t}\left[\mathbb{E}_{(x,y)\in\mathcal{D}^t}[\mathcal{L}(f_\theta(i,x),y)]\right]}\right]$$

where $\mathcal{L}$ denotes the evaluation metric such as accuracy or ROUGE-L, $f_\theta(\cdot)$ represents the multimodal instruction-tuned model. The standard deviation and mean of the model's performance across all instructions are represented by $\sigma_{i \in I^t}[\cdot]$ and $\mu_{i \in I^t}[\cdot]$, respectively.

5.2 Approaches for Comparison

OFA (Wang et al., 2022a), which denotes the original pre-trained OFA model without any fine-tuning. Here, we use OFA-large[3] which contains 472M parameters and was trained on 8 tasks shown in Table 10. As reported in Wang et al. (2022a), OFA has demonstrated certain zero-shot capability on unseen multimodal tasks.

OFA$_{\text{MultiInstruct}}$, which only fine-tunes OFA on our newly introduced MULTIINSTRUCT dataset with instruction tuning.

OFA$_{\text{NaturalInstruct}}$ only fine-tunes OFA on the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022; Wang et al., 2022d) with instruction tuning. To ensure a fair comparison, we evaluated this baseline on instruction templates that had removed all specific tokens, such as "`[Options]`" and "`||||`". This was done because the models being tested had not been exposed to these specific tokens during the instruction-tuning process. We want to ensure that the evaluation was not biased in favor of models that had seen these tokens during training.

OFA$_{\text{MixedInstruct}}$ fine-tunes OFA on the mix of the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022; Wang et al., 2022d) and MULTIINSTRUCT dataset with instruction tuning.

OFA$_{\text{SeqInstruct}}$ sequentially fine-tunes OFA on the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022; Wang et al., 2022d) and MULTIINSTRUCT dataset with instruction tuning.

OFA$_{\text{AdapterInstruct}}$ freezes the parameters of the OFA and only optimizes the parameters of the adapter sequentially on the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022; Wang et al., 2022d) and MULTIINSTRUCT dataset with instruction tuning.

[3] https://ofa-beijing.oss-cn-beijing.aliyuncs.com/checkpoints/ofa_large.pt

5.3 Training Details

We set the maximum length of input tokens to 1024 and the maximum target length to 512. For image preprocessing, we strictly follow the process in the OFA. Please refer to the original paper for more details. We train the models on 8 Nvidia A100 GPUs with a batch size 8 per GPU, a learning rate of 1e-05, and float16 enabled for 3 epochs for all the setups and datasets.

6 Results and Discussion

6.1 Effectiveness of Instruction Tuning on MULTIINSTRUCT

We evaluate the zero-shot performance of various approaches on all the unseen evaluation tasks, as shown in Table 3, 4 and 5. Our results indicate that OFA$_{\text{MultiInstruct}}$ significantly improves the model's zero-short performance over the original pre-trained OFA model across all unseen tasks and metrics, demonstrating the effectiveness of multimodal instruction tuning on MULTIINSTRUCT. As seen in Table 3, OFA achieves extremely low (nearly zero) zero-shot performance on the Grounded VQA task, which requires the model to generate region-specific tokens in order to answer the question. By examining the generated results, we find that OFA, without instruction tuning, failed to follow the instruction and produce results that contain region tokens. However, by fine-tuning OFA on MULTIINSTRUCT with instruction tuning, the model is able to better interpret and follow the instruction to properly generate the expected output.

6.2 Impact of Transfer Learning from NATURAL INSTRUCTIONS

One key question in multimodal instruction tuning is how to effectively leverage the large-scale text-only NATURAL INSTRUCTIONS dataset to enhance the instruction guidance and zero-shot performance on multimodal tasks. We observe that simply fine-tuning OFA on NATURAL INSTRUCTIONS actually degrades the model's zero-shot performance, as shown by comparing OFA$_{\text{NaturalInstruct}}$ and OFA. There are several potential reasons for this decline in performance. One potential cause is the misalignment in the embedding space between the vision and language modules, as the language module is updated more frequently than the vision module. Another possibility is that fine-tuning the

Model	Outside Knowledge VQA				Text VQA				Grounded VQA	
	RougeL		ACC		RougeL		ACC		ACC	
	Max	Avg ± Std	Max	Avg ± Std	Max	Avg ± Std	Max	Avg ± Std	Max	Avg ± Std
OFA	31.54	24.13 ± 10.33	28.38	21.19 ± 8.89	15.21	9.30 ± 5.42	12.32	7.96 ± 4.20	0.02	0.00 ± 0.01
$OFA_{MultiInstruct}$	32.55	30.31 ± 1.81	28.76	26.88 ± 1.63	**25.13**	20.31 ± 5.75	**20.62**	17.29 ± 4.31	63.85	42.03 ± 29.43
Transfer Learning from NATURAL INSTRUCTIONS										
$OFA_{NaturalInstruct}$	14.51	13.82 ± 0.57	12.21	11.56 ± 0.65	5.59	5.40 ± 0.24	4.18	3.78 ± 0.27	0.00	0.00 ± 0.00
$OFA_{MixedInstruct}$	32.15	30.95 ± 0.96	28.06	27.02 ± 0.71	19.07	17.37 ± 1.72	15.52	14.41 ± 1.10	62.23	49.62 ± 21.55
$OFA_{SeqInstruct}$	**32.97**	**31.70** ± 1.14	**28.91**	**27.72** ± 1.04	24.76	**23.61** ± 1.37	20.48	**19.61** ± 0.93	**64.20**	50.20 ± 22.56
$OFA_{AdapterInstruct}$	30.84	27.16 ± 2.98	26.91	23.16± 3.31	23.91	19.61 ± 4.33	19.56	15.90 ± 3.59	63.15	**53.51** ± 16.62

Table 3: **Zero-shot Performance on Question Answering datasets.** The best performance is underscored and in bold and the second best is in bold.

Model	Commonsense VQA				Visual Entailment	
	RougeL		ACC		ACC	
	Max	Avg ± Std	Max	Avg ± Std	Max	Avg± Std
OFA	17.93	14.97 ± 4.30	0.73	0.40 ±0.29	49.99	41.86 ± 10.99
$OFA_{MultiInstruct}$	**51.23**	**50.54** ± 0.66	31.91	**31.08** ± 1.05	**54.81**	**54.26** ± 0.66
Transfer Learning from NATURAL INSTRUCTIONS						
$OFA_{NaturalInstruct}$	27.15	14.99 ± 9.12	7.35	2.04 ± 3.01	33.28	14.86 ± 16.68
$OFA_{MixedInstruct}$	49.95	48.77 ± 1.19	30.49	29.47 ± 0.93	53.94	53.07 ± 0.99
$OFA_{SeqInstruct}$	50.80	49.79 ± 1.31	**32.00**	30.86 ± 1.13	52.71	51.86 ± 0.93
$OFA_{AdapterInstruct}$	47.64	46.36 ± 1.42	26.77	25.38 ± 1.09	47.19	43.84 ± 3.82

Table 4: **Zero-shot Performance on Multimodal Commonsense Reasoning.** The best performance is in **bold**.

Model	Visual Text Extraction		Visual Dialogue			
	RougeL		RougeL		ACC	
	Max	Avg± Std	Max	Avg ± Std	Max	Avg ± Std
OFA	36.31	17.62 ± 16.82	45.46	28.71 ± 9.81	37.63	23.97 ± 8.05
$OFA_{MultiInstruct}$	56.56	28.16 ± 25.62	**46.60**	34.80 ± 6.61	**38.47**	29.31 ± 5.13
Transfer Learning from NATURAL INSTRUCTIONS						
$OFA_{NaturalInstruct}$	5.65	1.24 ± 2.48	30.94	27.91 ± 2.16	25.95	23.23 ± 1.92
$OFA_{MixedInstruct}$	41.66	27.07 ± 10.13	46.01	38.49 ± 4.71	37.93	**32.16** ± 3.63
$OFA_{SeqInstruct}$	57.84	**49.16** ± 10.38	46.59	**36.50** ± 6.06	**38.47**	30.69 ± 4.73
$OFA_{AdapterInstruct}$	**58.11**	38.38 ± 23.98	46.17	31.30 ± 8.39	38.00	25.59 ± 7.06

Table 5: **Zero-shot Performance on Miscellaneous.** The best performance is in **bold**.

model on a text-only instruction dataset may make it pay less attention to image and region inputs.

Another observation is that simply training OFA on the mixture of MULTIINSTRUCT and NATURAL INSTRUCTIONS does not lead to the same level of performance improvement compared with OFA only trained on MULTIINSTRUCT, which is possibly due to the imbalance of the training tasks from these two datasets. On the other hand, $OFA_{SeqInstruct}$ achieves similar if not better performance compared with $OFA_{MultiInstruct}$, demonstrating the potential benefit of the much larger text-only instruction datasets to multimodal instruction tuning. Compared with $OFA_{SeqInstruct}$ which fine-tunes the whole parameter set of OFA, roughly 473M parameters, $OFA_{AdapterInstruct}$ only tunes 12.6M parameters (2.7% of the parameters tuned in $OFA_{SeqInstruct}$) but achieves comparable or slightly worse performance on all the unseen multiple tasks. This suggests the potential of $OFA_{AdapterInstruct}$ as a more parameter-efficient method for instruction tuning, especially when the size of the pre-trained multimodal language model or training tasks and datasets is very large.

6.3 Number of Multimodal Instruction Tasks

To evaluate the impact of the number of multimodal instruction tasks on multimodal instruction tuning, we measure the change in both the aggregated performance and intra-task sensitivity of $OFA_{MultiInstruct}$ as we vary the number of instruction tasks. The results, presented in Figure 3, show that as the number of multimodal instruction tasks increases, the aggregated performance improves

and the sensitivity decreases. This is a desirable outcome, as low intra-task sensitivity indicates that the model can produce consistent results despite variations in the wording of the instructions. The results also support the effectiveness of our proposed MULTIINSTRUCT dataset.

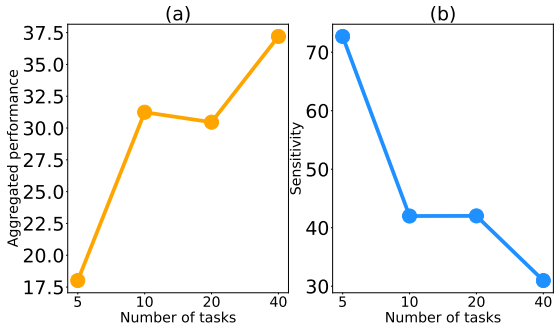

Figure 3: **Effect of Different Numbers of Multimodal Instruction Tuning Tasks.** (a) Shows the aggregated performance vs. the number of tasks. (b) Shows the proposed *Sensitivity* vs. the number of tasks.

6.4 Effect of Diverse Instructions on Instruction Tuning

We hypothesize that using a diverse set of instructions for each task during multimodal instruction tuning can improve the model's zero-shot performance on unseen tasks and reduce its intra-task sensitivity to variation in the instructions. To test this hypothesis, we train an OFA model on MULTIINSTRUCT but with a single fixed instruction template per task, and compare its performance with the one that is tuned on 5 different instructions. As shown in Table 6, OFA which is finetuned on 5 instructions significantly improves the overall zero-shot performance on all evaluation tasks and shows lower sensitivity. These results demonstrate the effectiveness of increasing the diversity of instructions and suggest that future work could explore crowd-sourcing or automatic generation strategies to create even more diverse instructions for multimodal instruction tuning.

# of Instructions	Aggregated Performance ↑	*Sensitivity* ↓
1 Instruction	34.66	39.86
5 Instructions	37.20	30.97

Table 6: Results of $\text{OFA}_{\text{MultiInstruct}}$ finetuned on different numbers of instructions.

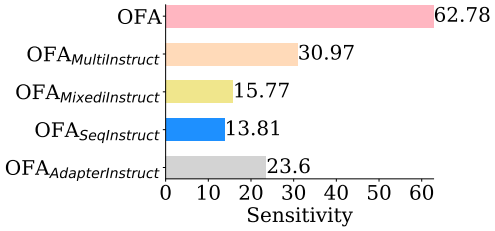

Figure 4: **Averaged *Sensitivity* of each model on all the unseen evaluation tasks.** Lower is better.

6.5 Effect of Fine-tuning Strategies on Model *Sensitivity*

In Section 6.3 and Section 6.4, we have shown that the more tasks and instructions that are used for instruction tuning, the lower sensitivity the model will achieve toward the varying instructions for each task. We further investigate the impact of fine-tuning and transfer learning strategies on model sensitivity. Figure 4 shows the averaged *sensitivity* of each model on all the multimodal unseen tasks. The original OFA shows much higher sensitivity to the variety of instructions than the models that are fine-tuned on instruction datasets, suggesting that the multimodal instruction tuning significantly improves the model's capability on interpreting the instructions even with varying wordings. In addition, by transferring the large-scale NATURAL INSTRUCTIONS dataset to MULTIINSTRUCT, the sensitivity is also reduced by a large margin, demonstrating the benefit of fine-tuning the model on a larger instruction dataset, regardless of different formats and modalities.

7 Conclusion and Future Work

In this paper, we present a new large-scale multimodal instruction tuning benchmark dataset – MULTIINSTRUCT, which covers a wide variety of vision and multimodal tasks while each task is associated with multiple expert-written instructions. By finetuning OFA (Wang et al., 2022a), a recently state-of-the-art multimodal pre-trained language model, on MULTIINSTRUCT with instruction tuning, its zero-shot performance on various unseen multimodal tasks is significantly improved. We also explore several transferring learning techniques to leverage the much larger text-only NATURAL INSTRUCTIONS dataset and demonstrate its benefit.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. Beit: Bert pre-training of image transformers. In *ICLR 2022*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568.

Tai-Yin Chiu, Yinan Zhao, and Danna Gurari. 2020. Assessing image quality issues for real-world problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3646–3656.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 326–335.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P. Bigham. 2022. Improving zero and few-shot generalization in dialogue through instruction tuning.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in Neural Information Processing Systems*, 33:2611–2624.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Hao Li, Jinguo Zhu, Xiaohu Jiang, Xizhou Zhu, Hongsheng Li, Chun Yuan, Xiaohua Wang, Yu Qiao, Xiaogang Wang, Wenhai Wang, and Jifeng Dai. 2022a. Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. *CoRR*, abs/2211.09808.

Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022b. Lavender: Unifying video-language understanding as masked language modeling. *arXiv preprint arXiv:2206.07160*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2022. Visual spatial reasoning. *arXiv preprint arXiv:2205.00363*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2022. Unified-io: A unified model for vision, language, and multi-modal tasks.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work?

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. 2021. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.

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

Sijia Wang, Mo Yu, and Lifu Huang. 2022b. The art of prompting: Event detection based on type specific prompts. *arXiv preprint arXiv:2204.07241*.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2022c. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *CoRR*, abs/2208.10442.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022d. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *CoRR*, abs/2109.01652.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *CoRR*, abs/2111.02080.

Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. 2022. End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models. *arXiv preprint arXiv:2205.12487*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

A Tasks in MULTIINSTRUCT

Category	Task Name	Dataset	Description	Exist
VQA	Open-Domain	VQAv2 (Goyal et al., 2017), Visual Genome (Krishna et al., 2017)	Answer the question <QUESTION> based on the content of the given image.	✓
	VQA	Visual7w (Zhu et al., 2016)	Answer a visual question <QUESTION> by selecting an answer from given options. <OPTION>	✓
	Compositional VQA	GQA (Hudson and Manning, 2019)	Answer a compositional question based on the content of the given image. Question: <QUESTION>	✓
Grounded Generation	Grounded Captioning	Visual Genome (Krishna et al., 2017)	Given the region <REGION> in the image, generate a caption for that region.	✓
	Visual Grounding	Visual Genome (Krishna et al., 2017)	Given a caption <TEXT> for some region in the image, identify the region and generate its bounding box.	✓
	Grounded Object Identification	MSCOCO (Lin et al., 2014)	Identify the type of an object in <REGION>.	✓
	Object Grounding	MSCOCO (Lin et al., 2014)	What are the regions containing the object [TEXT]?	×
	Referring Expression Grounding	RefCOCO (Yu et al., 2016)	Locate a region in an image based on the referring expression [TEXT].	✓
	Referring Expression Generation	RefCOCO (Yu et al., 2016)	Generate the referring expression for an object in region <REGION>.	✓
	Text Localization	COCO-Text (Veit et al., 2016)	Select a region from options that contain the text <TEXT> in the image. <OPTION>	✓
Captioning	Image Captioning	MSCOCO (Lin et al., 2014)	Generate a sentence to describe the content of the image.	✓
Image-Text Matching	Image-Text Matching	MSCOCO (Lin et al., 2014)	Decide if the text matches the image.	×
	Question-Image Matching	VQAv2 (Goyal et al., 2017)	Decide if the image contains an answer to the question <QUESTION>.	×
	Image-Text Selection	MSCOCO (Lin et al., 2014)	Select the text that best matches the image. <OPTION>	×
Grounded Matching	Region-Caption Matching	Visual Genome (Krishna et al., 2017)	Decide if the caption matches the given region <REGION> in the image.	×
	Grounded Caption Selection	Visual Genome (Krishna et al., 2017)	Given a region <REGION> in the image, select a caption from given options for that region. <OPTION>	×
	Visual Grounding Selection	Visual Genome (Krishna et al., 2017)	Given a caption <TEXT> for some region in the image, select the region from the options. <OPTION>	×
	Referring Expression Selection	RefCOCO (Yu et al., 2016)	Select a region from options based on the referring expression <TEXT>. <OPTION>	×
	Object-Region Matching	MSCOCO (Lin et al., 2014)	Does region <REGION> contain the object <TEXT>?	×
	Object-Region Selection	MSCOCO (Lin et al., 2014)	Select the region containing the given object <TEXT>. <OPTION>	×
	Object Matching	MSCOCO (Lin et al., 2014)	Do objects in region <REGION1> and region <REGION2> have the same type?	×
	Missing Object Selection	MSCOCO (Lin et al., 2014)	Select an object from options that does not appear in any of the given regions <REGION>. <OPTION>	×
	Region-Text Matching	COCO-Text (Veit et al., 2016)	Does region <REGION> contain the text <TEXT>?	×

Table 7: **Detailed group of training tasks included in MULTIINSTRUCT.** The complete list of 40 multi-modal tasks, along with examples of the instructions for each task.

Category	Task Name	Dataset	Description	Exist
Visual Relationship	Object Relationship	Visual Genome (Krishna et al., 2017)	What is the relationship between the subject in region <REGION1> and object in region <REGION2>?	✓
	Visual Object Identification	Visual Genome (Krishna et al., 2017)	Given the subject in region <REGION>, what is the object that has a relationship <TEXT> with that subject?	×
	Visual Subject Identification	Visual Genome (Krishna et al., 2017)	Given the object in region <REGION>, what is the subject that has a relationship <TEXT> with that object?	×
	Visual Object Localization	Visual Genome (Krishna et al., 2017)	Given the subject in region <REGION>, where is the object in the image that has relationship <TEXT> with the subject?	×
	Visual Subject Localization	Visual Genome (Krishna et al., 2017)	Given the object in region <REGION>, where is the subject in the image that has relationship <TEXT> with the object?	×
	Grounded Image Attribute Identification	VAW (Pham et al., 2021)	Decide which option is the attribute of the object in the region <REGION>. <OPTION>	✓
Image Attributes	Image Quality	IQA (Chiu et al., 2020)	Select a reason from options to explain why the image quality is bad. <OPTION>	✓
Image Generation	Image Infilling	Conceptual-12M (Changpinyo et al., 2021)	Fill in the missing part of the image.	✓
Miscellaneous	Multimodal Factual Checking	MOCHEG (Yao et al., 2022)	Decide if the claim can be supported by the given image and the context.	✓
	Text Legibility	COCO-Text (Veit et al., 2016)	Decide if the text in the given region is legible.	✓
	Text Type Classification	COCO-Text (Veit et al., 2016)	Read the text in the given region and determine the type of text from options.	✓
Commonsense Reasoning	Natural Language for Visual Reasoning	NLVR (Suhr et al., 2017)	Decide if the sentence <TEXT> correctly describes the geometric relationships of objects in a synthesized image.	✓
	Visual Spatial Reasoning	VSR (Liu et al., 2022)	Decide if the proposed spatial relationship between two objects in an image is "True" or "False"	✓
Temporal Ordering	Wikihow Next Step Generation	WikiHow [4]	For task <TASK>, given the history steps and the current step with its corresponding image, what is the next step for this task? <HISTORY>	×
	Wikihow Next Step Selection	WikiHow	For task <TASK>, select the immediate next step to the step specified by the image.	×
	Wikihow Text-Image Temporal Ordering	WikiHow	For the task <TASK>, given the current step <STEP>, decide if the content of the image is the next or previous step.	×
	Wikihow Image-Text Temporal Ordering	WikiHow	For the task <TASK>, given the current step specified by the image, decide if the step <STEP> is the next or previous step.	×

Table 8: **(Continued) Detailed group of training tasks included in MULTIINSTRUCT.** The complete list of 40 multi-modal tasks, along with examples of the instructions for each task.

Category	Task Name	Dataset	Description	Exist
VQA	Outside Knowledge VQA	OK-VQA (Marino et al., 2019)	Based on your knowledge, <QUESTION>?	✓
	Text VQA	Text VQA (Singh et al., 2019)	There is some text on the image. Answer <QUESTION> based on the text in the image.	✓
	Grounded VQA	Visual7W	Which region is the answer to <QUESTION>? <Options>.	✓
	Commonsense Visual Question Answering	VCR (Zellers et al., 2019)	Look at the image and the regions in the question, <QUESTION>? <Options>.	✓
Commonsense Reasoning	Visual Entailment	SNLI-VE (Xie et al., 2019)	Can you conclude *<TEXT>* from the content of image? Select your answer from the options. <OPTIONS>	✓
Miscellaneous	Visual Text Extraction	Hateful Memes (Kiela et al., 2020)	What is the text written on the image?	×
	Visual Dialogue	Visual Dialogue (Das et al., 2017)	Given the image and the dialog history below: <HISTORY> <QUESTION>?	✓

Table 9: **Detailed group of evaluation tasks included in MULTIINSTRUCT.** The complete list of 7 multi-modal tasks along with the examples of the instructions.

Dataset Name	Task Name
Conceptual Caption 12M (CC12M)	Image Captioning
Conceptual Captions (CC3M)	Image Captioning
MSCOCO image captions (COCO)	Image Captioning
Visual Genome Captions (VG Captions)	Image Captioning
VQAv2	Visual Question Answering
VG-QA (COCO)	Visual Question Answering
GQA (VG)	Visual Question Answering
RefCOCO	Visual Grounding
RefCOCO+	Visual Grounding
RefCOCOg	Visual Grounding
VG captions	Visual Grounded Captioning
OpenImages	Object Detection
Object365	Object Detection
VG	Object Detection
COCO	Object Detection
OpenImages	Image Infilling
YFCC100M	Image Infilling
ImageNet-21K	Image Infilling

Table 10: Multimodal tasks used to pre-train OFA