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 has yet to be explored for vision and multimodal tasks. In this work, we introduce MULTIINSTRUCT, the first multimodal instruction tuning benchmark dataset that consists of 62 diverse multimodal tasks in a unified seq-to-seq format covering 10 broad categories. The tasks are derived from 21 existing open-source datasets and each task is equipped with 5 expert-written instructions. We take OFA (Wang et al., 2022a) as the base pre-trained model for multimodal instruction tuning, and to further improve its zero-shot performance, we explore multiple transfer learning strategies to leverage the large-scale NATURAL INSTRUCTIONS dataset (Mishra et al., 2022). Experimental results demonstrate strong zero-shot performance on various unseen multimodal tasks and the benefit of transfer learning from a text-only instruction dataset. We also design a new evaluation metric – *Sensitivity*, to evaluate how sensitive the model is to the variety of instructions. Our results indicate that fine-tuning the model on a diverse set of tasks and instructions leads to a reduced sensitivity to variations in instructions for each task[1].

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 pretraining (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 Vision-Language pre-trained models on multi-modal and vision tasks?

In this work, we propose MULTIINSTRUCT, the first benchmark dataset for multimodal instruction tuning with 62 diverse tasks from 10 broad categories, including Visual Question Answering (Goyal et al., 2017; Suhr et al., 2017), Commonsense Reasoning (Zellers et al., 2019; Xie et al., 2019), Visual Relationship Understanding (Krishna et al., 2017) and so on. We equipped each task with 5 instructions that are 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.

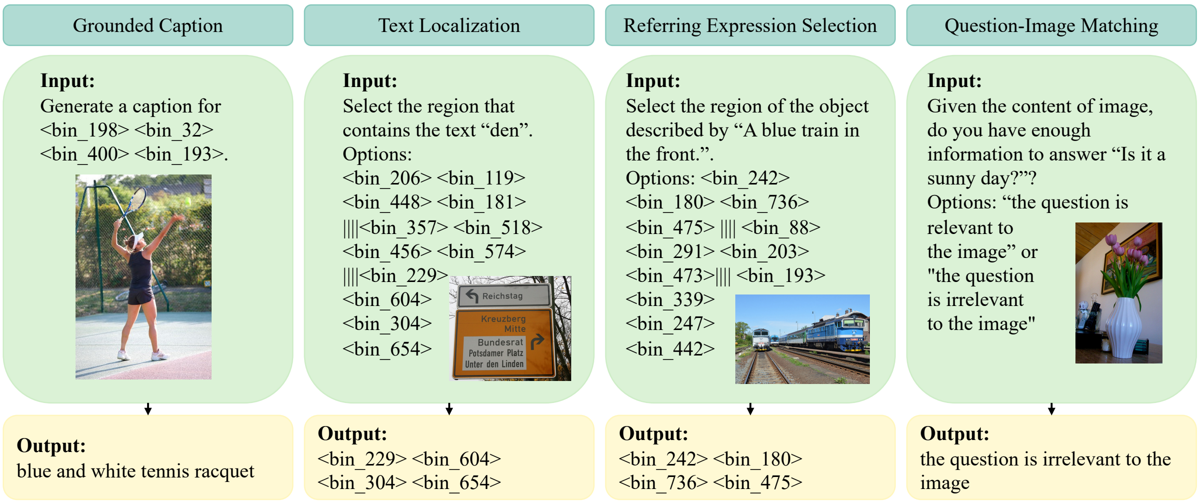
We use OFA (Wang et al., 2022a)[2], a unified model that is pre-trained on a diverse set of multimodal and unimodal tasks in a single Transformer-based sequence-to-sequence framework, as the base pre-trained multimodal language model, and fine-tune it on MULTIINSTRUCT. To utilize NATURAL INSTRUCTIONS (Mishra et al., 2022), a large-scale text-only instruction tuning dataset, we further explore two transfer learning strategies, in-

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

[1] The dataset, source code, and model checkpoints are publicly available at https://github.com/VT-NLP/MultiInstruct.

[2] We use OFA as it was the largest and most powerful open-source multimodal pre-trained model available at the time of our research while other stronger models didn't have publicly available checkpoints at that time.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11445–11465
July 9-14, 2023 ©2023 Association for Computational Linguistics



Input:
Generate a caption for
<bin_198> <bin_32>
<bin_400> <bin_193>.

Output:
blue and white tennis racquet

Input:
Select the region that
contains the text "den".
Options:
<bin_206> <bin_119>
<bin_448> <bin_181>
||||<bin_357> <bin_518>
<bin_456> <bin_574>
||||<bin_229>
<bin_604>
<bin_304>
<bin_654>

Output:
<bin_229> <bin_604>
<bin_304> <bin_654>

Input:
Select the region of the object
described by "A blue train in
the front.".
Options: <bin_242>
<bin_180> <bin_736>
<bin_475> |||| <bin_88>
<bin_291> <bin_203>
<bin_473>|||| <bin_193>
<bin_339>
<bin_247>
<bin_442>

Output:
<bin_242> <bin_180>
<bin_736> <bin_475>

Input:
Given the content of image,
do you have enough
information to answer "Is it a
sunny day?""?
Options: "the question is
relevant to
the image" or
"the question
is irrelevant
to the image"

Output:
the question is irrelevant to the image

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

cluding *Mixed Instruction Tuning* and *Sequential Instruction Tuning*. Experimental results demonstrate 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 suggested by previous studies (Webson and Pavlick, 2022; Liu et al., 2022b), PLMs are highly sensitive toward the wording and length of instructions. Thus, we propose a new metric – *Sensitivity*, which measures 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. The more tuning tasks and instructions for each task are introduced, the lower sensitivity tends to be achieved, and (2) transferring from a larger text-only instruction dataset can also significantly reduces the sensitivity of OFA.

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 significantly advanced the vision-language tasks. Several recent studies (Cho et al., 2021; Wang et al., 2022a,c; Lu et al., 2022) also started to build 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 it to image generation tasks by using a unified vocabulary for all text and visual tokens. BEIT-3 (Wang et al., 2022c) utilizes a novel shared Multiway Transformer network with a shared self-attention module to align different modalities and provide deep fusion. Building on the success of multimodal pretraining, our work focuses on improving the generalization and zero-shot performance on various unseen multimodal tasks through instruction tuning.

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 (Liu et al., 2021; Li and Liang, 2021; Han et al., 2022; Wang et al., 2022b; Sanh et al., 2022) aims to learn a task-specific prompt by reformulating the downstream tasks to the format that the model was initially trained on and has shown competitive performance across various natural language processing applications. As a special form of prompt tuning, in-context learning (Xie et al., 2021; Min et al., 2021) takes one or a few examples as the prompt to demonstrate the task. Instruction tuning (Wei et al., 2021) is another simple yet effective strategy to improve the generalizability of large language models. NATURAL INSTRUCTIONS (Mishra et al., 2022) is a meta-dataset containing diverse tasks with human-authored definitions, things to avoid, and demonstrations. It has shown effectiveness in improving the generalizability of language models even when the size is relatively small (e.g., BART_base) (Mishra et al., 2022; Wang et al., 2022d). 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

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, images, and text. These tasks are meant to teach machine learning models to perform various tasks such as object recognition, visual relationship understanding, text-image grounding, and so on by following instructions so that they can perform zero-shot prediction on unseen tasks. To build MultiInstruct, we first collect 34 tasks from the existing studies in visual and multimodal learning, covering Visual Question Answering (Goyal et al., 2017; Krishna et al., 2017; Zhu et al., 2016; Hudson and Manning, 2019; Singh et al., 2019; Marino et al., 2019), Commonsense Reasoning (Suhr et al., 2017; Liu et al., 2022a; Zellers et al., 2019; Xie et al., 2019), Region Understanding (Krishna et al., 2017), Image Understanding (Kafle and Kanan, 2017; Chiu et al., 2020), 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; Pham et al., 2021), Temporal Ordering tasks that are created from WikiHow[3], and Miscellaneous (Yao et al., 2022; Kiela et al., 2020; Das et al., 2017; Lin et al., 2014; Veit et al., 2016; Alam et al., 2022). Each of the 34 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 further examine the possibility of deriving new tasks based on the input and output of the original task to augment the task repository. For example, *Visual Grounding* requires the model to generate a caption for a given region in the image. We derive two additional tasks from it: *Grounded Caption Selection*, which is a simpler task 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. Compared with *Visual Grounding*, these two new tasks require different skills based on distinct input and output information. In this way, we further derived 28 new tasks from the 34 existing tasks. We divide all 62 tasks into 10 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 instances of existing tasks or reformulating them. Each new task is created with 5,000 to 5M instances. We split the 62 tasks into training and evaluation based on the following 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 select the challenging multimodal tasks that do not overlap with the training tasks for evaluation. Table 5 and Table 6 in Appendix A show the detailed statistics for the training and evaluation tasks in MultiInstruct and Tables 7 to 9 show 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 by giving them the specific goals of

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

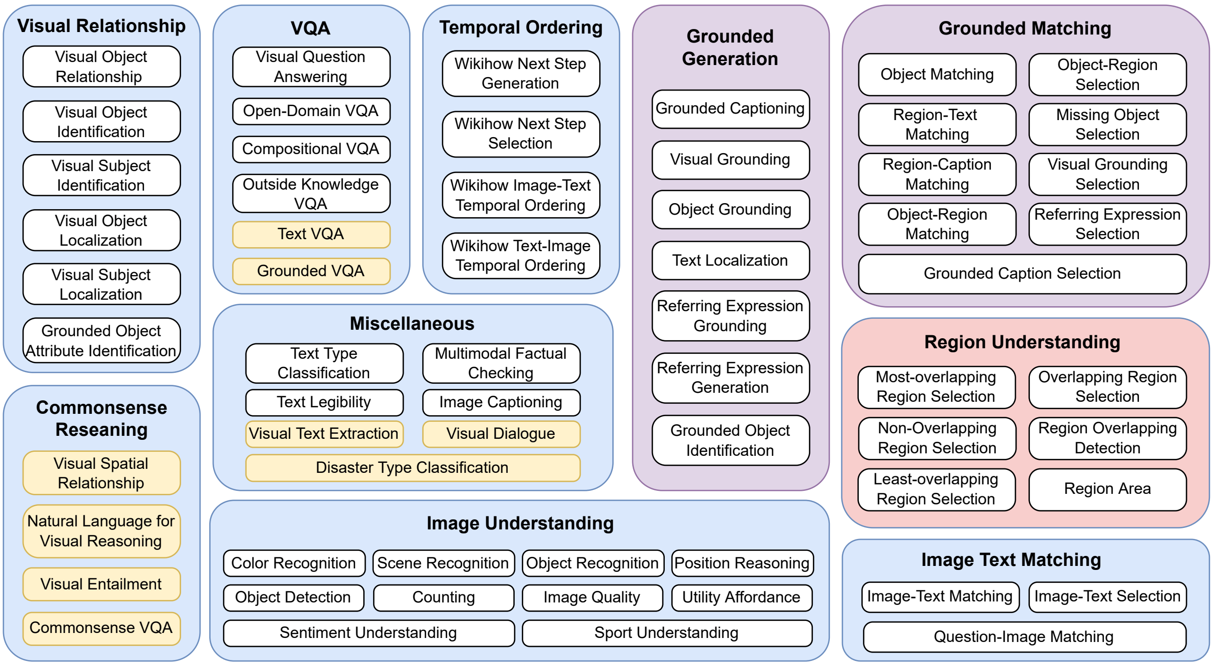


Figure 2: **Task Groups Included in MultiInstruct.** The yellow boxes represent tasks used for evaluation, while the white boxes indicate tasks used for training.

this task, the format of input data, and 10 example instances randomly sampled from the dataset. The information about the dataset is obtained from the dataset's README file or the publication that introduced the dataset. For newly derived tasks, we provide annotators with task descriptions along with 10 constructed example instances.

Step 2: to guarantee the quality of the instructions and that they effectively convey the intended tasks, we have each annotator review the instructions created by their peers, 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 consistency and avoid conflicts or repetition among instructions from different annotators, we have both annotators review the sets of instructions together, identifying any discrepancies or inconsistencies. If any are found, the annotators collaborate 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 1,000 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.

4 Problem Setup and Models

4.1 Problem Setup

We follow the same instruction tuning setting as the previous study (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 due to its unified architecture and flexible input-output modalities. We finetune it on our MULTIINSTRUCT dataset to demonstrate the effectiveness of instruction tuning. Specifically, we use the transformer-based encoder of OFA to encode the instruction along with all necessary information and an optional image, and predict the output with the transformer-based 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[4], we ensure that the evaluation tasks in MULTIINSTRUCT do not overlap with either the pre-training tasks in OFA nor the training tasks in MULTIINSTRUCT.

4.2 Transfer Learning from NATURAL INSTRUCTIONS

We notice that the scale of NATURAL INSTRUCTIONS (Mishra et al., 2022) is significantly larger than MULTIINSTRUCT, indicating the potential of transferring the instruction learning capability from the larger set of natural language tasks to multimodal tasks. We take 832 English tasks in NATURAL INSTRUCTIONS and explore several simple transfer-learning strategies:

Mixed Instruction Tuning ($\textbf{OFA}_{\textbf{MixedInstruct}}$) 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 ($\textbf{OFA}_{\textbf{SeqInstruct}}$) 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 fine-tune it on MULTIINSTRUCT to adapt the instruction learning capability to multimodal tasks. To maximize the effectiveness of the NATURAL INSTRUCTIONS dataset, we use all instances in English-language tasks to tune the model in the first training stage.

5 Experimental Setup

Evaluation Metrics We report the accuracy for classification tasks and ROUGE-L (Lin, 2004) for all generation tasks. For the region classification task, we compute the Intersection over Union (IoU) between the generated region and all regions in the options, select the option with the highest IoU as the prediction, and compute accuracy based on this prediction. If the predicted region has no intersection with any of the regions in the options, we treat this prediction as incorrect. For classification tasks where the answer is not a single-word binary classification, we also report ROUGE-L scores following Mishra et al. (2022), which treats all tasks as text generation problems. For each task, we conduct five experiments by evaluating the model using one of the five instructions in each experiment. We re-

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

port the mean and maximum performance and the standard deviation of the performance across all five experiments. We also compute the *aggregated performance* for each model based on the mean of the model's performance on all multimodal and NLP unseen tasks. We use Rouge-L as the evaluation metric for most tasks and accuracy for tasks that only have accuracy as a metric.

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 in the same task, 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 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 with task instructions: $\mathcal{D}^t = \{(I^t, x_j^t, y_j^t) \in \mathcal{I}^t \times \mathcal{X}^t \times \mathcal{Y}^t\}_{j=1}^N$, we formally define *sensitivity* as:

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

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 denoted by $\sigma_{i \in I^t}[\cdot]$ and $\mu_{i \in I^t}[\cdot]$, respectively.

Evaluation datasets We evaluate the models on nine unseen multimodal tasks: Text VQA (Singh et al., 2019), Grounded VQA (Zhu et al., 2016), Commonsense VQA (Zellers et al., 2019), Visual Entailment (Xie et al., 2019), Visual Spatial Reasoning (Liu et al., 2022a), Natural Language for Visual Reasoning (NLVR) (Suhr et al., 2017), Visual Text Extraction (Kiela et al., 2020), Visual Dialogue (Das et al., 2017), and Disaster Type Classification (Alam et al., 2022). These tasks belong to three task groups: Commonsense Reasoning, VQA, and Miscellaneous as shown in Figure 2. Tasks in the Commonsense Reasoning group have no overlap with any training task groups. Tasks in Miscellaneous do not share similarities with other tasks in the group. Although Text VQA and Grounded VQA belong to the VQA task group, they require additional skills such as extracting text from images or generating regions, making them fundamentally different from other tasks in VQA. In addition to multimodal tasks, we also evaluate the model on 20 NLP tasks collected from the test split of NATURAL INSTRUCTIONS.

Approaches for Comparison We denote the OFA finetuned on MULTIINSTRUCT as $\textbf{OFA}_{\textbf{MultiInstruct}}$, and compare it with the original pre-trained **OFA**[5], $\textbf{OFA}_{\textbf{TaskName}}$ which is fine-tuned on MULTIINSTRUCT but uses the task name instead of instruction to guide the model to make predictions, and several approaches that leverage the large-scale NATURAL INSTRUCTIONS dataset, including $\textbf{OFA}_{\textbf{NaturalInstruct}}$ which only fine-tunes OFA on NATURAL INSTRUCTIONS with instruction tuning, $\textbf{OFA}_{\textbf{MixedInstruct}}$ and $\textbf{OFA}_{\textbf{SeqInstruct}}$ that are specified in Section 4.2.

More details regarding the evaluation datasets, baseline approaches and training details can be found in Appendix B.

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 1 and 2. Our results indicate that $\text{OFA}_{\text{MultiInstruct}}$ 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 2, 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, the model is able to better interpret and follow the instructions to properly generate the expected output. Additionally, $\text{OFA}_{\text{MultiInstruct}}$ outperforms $\text{OFA}_{\text{TaskName}}$ on all unseen tasks, particularly on the Grounded VQA task, where $\text{OFA}_{\text{TaskName}}$ achieves nearly zero per-

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

Model	Commonsense VQA				Visual Entailment		Visual Spatial Reasoning		NLVR	
	RougeL		ACC		ACC		ACC		ACC	
	Max	Avg ± Std	Max	Avg ± Std	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	54.99	35.29 ± 22.21	56.06	52.10 ± 3.35
OFA$_{\text{TaskName}}$	48.99	-	29.01	-	55.70	-	53.76	-	55.35	-
OFA$_{\text{MultiInstruct}}$	**52.01**	**50.60** ± 1.12	**33.01**	31.17 ± 1.59	**55.96**	**55.06** ±0.76	**55.81**	**53.90** ±1.38	56.97	56.18 ± 0.95
Transfer Learning from Natural Instructions										
OFA$_{\text{NaturalInstruct}}$	27.15	14.99 ± 9.12	7.35	2.04 ± 3.01	33.28	14.86 ± 16.68	51.44	36.44 ± 20.72	56.06	35.98 ± 21.64
OFA$_{\text{MixedInstruct}}$	50.40	49.34 ± 1.04	31.31	30.27 ± 0.94	54.63	53.74 ± 0.97	55.13	52.61 ± 1.64	56.67	55.96 ± 0.48
OFA$_{\text{SeqInstruct}}$	50.93	50.07 ± 1.07	32.28	**31.23** ± 1.09	53.66	52.98 ± 0.56	54.86	53.11 ± 1.45	**57.58**	**56.63** ± 0.66

Table 1: **Zero-shot Performance on Multimodal Commonsense Reasoning.** The best performance is in **bold**.

Model	Text VQA		Grounded VQA		Visual Text Extraction		Visual Dialogue		Disaster Type Classification	
	RougeL		Acc		RougeL		RougeL		ACC	
	Max	Avg± Std	Max	Avg± Std	Max	Avg± Std	Max	Avg ± Std	Max	Avg ± Std
OFA	15.21	9.30 ± 5.42	0.02	0.00 ± 0.01	36.31	17.62 ± 16.82	45.46	28.71 ± 9.81	14.30	9.64 ± 4.34
OFA$_{\text{TaskName}}$	23.80	-	0.00	-	36.30	-	25.18	-	62.65	-
OFA$_{\text{MultiInstruct}}$	**27.22**	26.46 ± 0.83	**64.32**	47.22 ± 23.08	**74.35**	**62.43** ±11.56	**46.38**	32.91 ±7.59	64.88	56.00 ±12.96
Transfer Learning from Natural Instructions										
OFA$_{\text{NaturalInstruct}}$	5.59	5.40 ± 0.24	0.00	0.00 ± 0.00	5.65	1.24 ± 2.48	30.94	27.91 ± 2.16	56.64	38.21 ± 15.35
OFA$_{\text{MixedInstruct}}$	24.15	23.67 ± 0.47	63.79	**54.99** ± 18.16	62.43	46.56 ± 14.92	46.08	**38.02** ± 5.25	**68.31**	**64.31** ± 2.39
OFA$_{\text{SeqInstruct}}$	27.03	**26.67** ± 0.47	64.19	54.46 ± 15.96	71.63	60.62 ± 12.31	46.17	35.10 ± 6.92	64.46	57.89 ± 9.51

Table 2: **Zero-shot Performance on Question Answering and Miscellaneous.** The best performance is in **bold**.

formance. This suggests that the performance gain of OFA$_{\text{MultiInstruct}}$ mainly comes from instructions rather than multi-task training.

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 zero-shot performance on multimodal tasks. We observe that only fine-tuning OFA on Natural Instructions actually degrades the model's zero-shot performance on almost all multimodal tasks, as shown by comparing OFA$_{\text{NaturalInstruct}}$ and OFA in Table 1 and 2. One potential reason for this decline in performance is that during fine-tuning on the text-only dataset, the model learns to focus more on text tokens and attend less to image tokens. To verify this assumption, we compare the attention of text tokens on image tokens between OFA$_{\text{NaturalInstruct}}$ and other methods and observe that text tokens attend much less to image tokens after fine-tuning on the Natural Instructions dataset. The detailed explanations and analysis can be found in Appendix C.

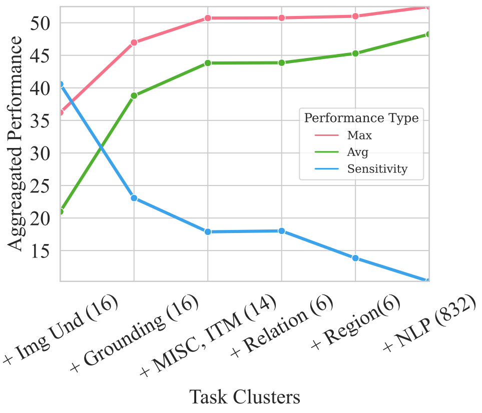


Figure 3: **Model Performance as the Number of Multimodal Instruction Task Clusters Increases.** The number in the parenthesis of each cluster denotes the number of tasks.

Another observation is that although our transfer learning methods do not lead to significant performance gains over OFA$_{\text{MixedInstruct}}$, both OFA$_{\text{SeqInstruct}}$ and OFA$_{\text{MixedInstruct}}$ achieve lower standard deviation on 6 out of 9 unseen multimodal tasks compared with OFA$_{\text{MultiInstruct}}$, demonstrating the potential benefits of the much larger text-only instruction datasets to multimodal instruction tuning.

6.3 Impact of Increasing Multimodal Instruction Task Clusters

To evaluate the impact of the number of tasks clusters for instruction tuning, we start with the task groups shown in Figure 2 and group them into five larger clusters: (1) `Img Und` (VQA + Image Understanding), (2) `Grounding` (Grounded Matching + Grounded Generation), (3) `MISC, ITM` (Temporal Ordering + Miscellaneous + Image Text Matching), (4) `Relation` (Visual Relationship),

# of Instructions	Aggregated Performance ↑	*Sensitivity* ↓
1 Instruction	42.81	24.62
5 Instructions	**47.82**	**10.45**

Table 3: **Effect of Different Number of Instructions.** Performance of OFA$_{\text{MultiInstruct}}$ finetuned on different numbers of instructions.

(5) `Region` (Region Understanding), together with (6) `NLP`, a collection of NLP tasks from NATURAL INSTRUCTIONS. We measure the change in both the aggregated performance and *sensitivity* of OFA$_{\text{MixedInstruct}}$ as we gradually add the task clusters for training.

As we increase the number of task clusters, we observe an improvement in both the mean and maximum aggregated performance and a decrease in *sensitivity*, as shown in Figure 3. Note that low *sensitivity* indicates that the model can produce consistent results despite variations in the wording of instructions. These results suggest that increasing the number of task clusters improves the model's performance on unseen tasks and leads to more consistent outputs. The results also support the effectiveness of our proposed MULTIINSTRUCT dataset.

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 *sensitivity* to variation in the instructions. To test this hypothesis, we train an OFA model on MULTIINSTRUCT with a single fixed instruction template per task and compare its performance with OFA finetuned on 5 different instructions. As shown in Table 3, OFA finetuned on 5 instructions achieves much higher aggregated 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 instruction tuning.

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

In Section 6.3 and 6.4, we have shown that the more tasks and instructions used for instruction tuning, the lower *sensitivity* the model will achieve toward the variations in 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 across all multimodal unseen tasks. The original OFA exhibits significantly higher sensitivity to variations in instructions compared to models fine-tuned on instruction datasets, indicating that multimodal instruction tuning significantly improves the model's capability on interpreting instructions, even with varying wordings. In addition, by transferring the large-scale NATURAL INSTRUCTIONS dataset to MULTIINSTRUCT, *sensitivity* is also reduced by a large margin, highlighting the benefit of fine-tuning the model on a larger instruction dataset, regardless of different formats and modalities.

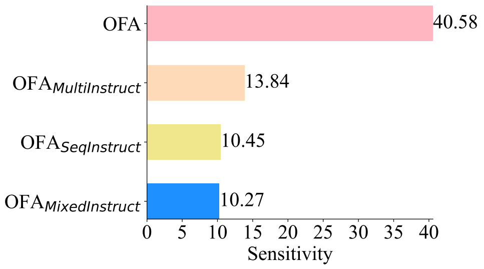


Figure 4: **Model *Sensitivity* on Unseen Evaluation Tasks.** Lower is better.

7 Zero-Shot Performance on NLP Tasks

So far, our focus has been on evaluating the zero-shot performance of multimodal tasks. In this section, we investigate the effect of multimodal instruction tuning on the performance of text-only tasks. To do this, we evaluate all our approaches on 20 natural language processing (NLP) tasks from the default test split in NATURAL INSTRUCTIONS[6]. The detailed task list can be found in Appendix B.2.

As shown in Table 4, OFA$_{\text{MultiInstruct}}$ outperforms OFA, despite the instruction tuning dataset and the unseen dataset are in different modalities. This suggests that multimodal instruction tuning can help improve the zero-shot performance on NLP tasks. In addition, we observe that OFA$_{\text{NaturalInstruct}}$ achieves the best performance on NLP tasks and OFA$_{\text{MixedInstruct}}$ is more effective in preserving the zero-shot capability gained from NATURAL INSTRUCTIONS on NLP tasks compared

[6] `https://github.com/allenai/natural-instructions`

Model	RougeL
OFA	2.25
$\text{OFA}_{\text{MultiInstruct}}$	12.18
Transfer Learning from NATURAL INSTRUCTIONS	
$\text{OFA}_{\text{NaturalInstruct}}$	**43.61**
$\text{OFA}_{\text{MixedInstruct}}$	43.32
$\text{OFA}_{\text{SeqInstruct}}$	30.79

Table 4: **Zero-shot Performance on NLP tasks.** The performance is reported in Rouge-L and the best performance is in **bold**.

to $\text{OFA}_{\text{SeqInstruct}}$. Based on the results in Tables 1, 2 and 4, we conclude that $\text{OFA}_{\text{MixedInstruct}}$ is able to achieve overall best aggregated performance on all multimodal and NLP tasks and shows much lower *sensitivity* towards variations in the wording of instructions, making it the most promising approach.

8 Conclusion

We present a new large-scale multi-modal 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 transfer learning techniques to leverage the much larger text-only NATURAL INSTRUCTIONS dataset and demonstrate its benefit. Moreover, we design a new evaluation metric *Sensitivity* to assess the model's sensitivity towards the variations in the wording of instructions. Results show that the model becomes less sensitive to these variations after being fine-tuned on a variety of tasks and instructions.

Limitations

Limitations of Data Collection Our proposed dataset only targets English language tasks. Future work should explore multimodal instruction tuning in a more diverse language setting and augment our MULTIINSTRUCT with multi-multilingual tasks. In addition, our current dataset mainly focuses on vision-language tasks. Datasets from more diverse modalities should be considered such as audio (Panayotov et al., 2015; Gemmeke et al., 2017; You et al., 2022) and video (Soomro et al., 2012; Ionescu et al., 2014). While we have built a novel multimodal instruction dataset containing 62 tasks, the number of tasks and associated instructions remains limited. To address this, future research could consider utilizing crowd-sourcing or automatic generation and augmentation techniques to increase the variety of instructions available.

Limitations of Experiments and Evaluation Our work is the first to explore instruction tuning on multimodal tasks and shows improved performance compared to baseline methods. However, there is still room for improvement, specifically in utilizing text-only instruction datasets. Future research could explore alternative architectures and stronger vision-language pre-trained models, or develop additional training loss functions to better utilize these unimodal instruction datasets. Additionally, we only used OFA as the baseline model as it was the largest open-source multimodal pre-trained model available when we conducted this research. As more and stronger multimodal pre-trained models being publicly available, it would be interesting to conduct a thorough comparison between models with different sizes. Finally, we take the first step to define *sensitivity* as a metric to evaluate the robustness of the models on understanding and following human-written instructions, which can be a potential standard metric for all the following instruction-tuning studies. However, it's only based on the variation of model performance across different instructions for the same task. In the future, we will consider more broad factors, e.g., the model's capability to understand different instructions for different tasks (Inter-task sensitivity), to further improve the *sensitivity* metric for instruction tuning.

Acknowledgments

This research is based upon work supported by the U.S. DARPA KMASS Program # HR001121S0034. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

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.

Firoj Alam, Tanvirul Alam, Md Hasan, Abul Hasnat, Muhammad Imran, Ferda Ofli, et al. 2022. Medic: a multi-task learning dataset for disaster image classification. *Neural Computing and Applications*, pages 1–24.

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.

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.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 776–780. IEEE.

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*.

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.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339.

Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE international conference on computer vision*, pages 1965–1973.

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. 2022a. Visual spatial reasoning. *arXiv preprint arXiv:2205.00363*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022b. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *CoRR*, abs/2205.05638.

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*.

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.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402.

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*, pages 2300–2344, Seattle, United States. 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*.

Chenyu You, Nuo Chen, Fenglin Liu, Shen Ge, Xian Wu, and Yuexian Zou. 2022. End-to-end spoken conversational question answering: Task, dataset and model. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1219–1232. Association for Computational Linguistics.

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 Defined in MultiInstruct

Table 5 shows the distribution of input and output modalities for both training and evaluation tasks in MultiInstruct, and Table 6 shows the detailed statistics for all the training and evaluation tasks separately. Tables 7 to 9 provide a comprehensive list of the 62 tasks included in MultiInstruct, along with one example of instruction for each task.

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

Table 5: 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 6: Detailed statistics in MultiInstruct.

B More Details for Experimental Setup

B.1 Multimodal Evaluation Datasets

Text VQA (Singh et al., 2019) requires models to read and reason about the text in an image to answer questions based on them.

Grounded VQA (Zhu et al., 2016) requires models to answer the questions about an image, with the answers being specific visual regions within the image.

Commonsense VQA (Zellers et al., 2019) requires the model to answer a multiple-choice question that requires commonsense reasoning about an image. Both the question and answers are presented in a combination of natural language and references to specific image regions within the image.

Visual Entailment (Xie et al., 2019) requires the model to determine whether the image semantically entails the text.

Natural Language for Visual Reasoning (NLVR) (Suhr et al., 2017) requires the model to answer a question that requires visual and set-theoretic reasoning on a synthetic image.

Visual Text Extraction is a new task derived from Hateful Memes (Kiela et al., 2020) dataset. This task requires the model to extract the text that appears in the image.

Visual Dialogue (Das et al., 2017) requires the model to answer a question given an image and a dialogue history.

Disaster Type Classification (Alam et al., 2022) requires the model to determine the disaster type based on the image.

B.2 NLP Evaluation Tasks

Below are the task names of the 20 NLP tasks that we used to test the zero-shot performance of all the methods. The 20 NLP tasks are from the default test split of the NATURAL INSTRUCTIONS dataset. During testing, we leverage the 'Definition' of the task as an instruction and prepend it with each input.

task1624_disfl_qa_question_yesno_classification,
task133_winowhy_reason_plausibility_detection,
task569_recipe_nlg_text_generation,
task1631_openpi_answer_generation,
task957_e2e_nlg_text_generation_generate,
task1386_anli_r2_entailment,
task393_plausible_result_generation,
task670_ambigqa_question_generation,
task890_gcwd_classification,
task1534_daily_dialog_question_classification,
task1388_cb_entailment,
task190_snli_classification,
task1533_daily_dialog_formal_classification,
task1598_nyc_long_text_generation,
task199_mnli_classification,
task1439_doqa_cooking_isanswerable,
task1409_dart_text_generation,
task1529_scitail1.1_classification,
task648_answer_generation,
task050_multirc_answerability

B.3 Approaches for Comparison

OFA (Wang et al., 2022a) denotes the original pre-trained OFA model without any fine-tuning. Here, we use OFA-large[8] 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$_{\textbf{TaskName}}$ is finetuned on MULTIINSTRUCT but it does not use the instructions we created for the tasks. Instead, we prepend the task name to each input and use a semicolon to separate the task name and the input. For a fair comparison, we still keep the two special tokens "`[Options]`" and "`||||`" for the option field.

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

OFA$_{\textbf{NaturalInstruct}}$ 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 evaluate this baseline on instruction templates that removed all specific tokens, including "`[Options]`" and "`||||`", since the model being tested has not been exposed to these specific tokens during instruction-tuning. We want to ensure that the evaluation is not biased in favor of models that have seen these tokens during training.

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

OFA$_{\textbf{SeqInstruct}}$ sequentially fine-tunes OFA on the large-scale NATURAL INSTRUCTIONS (Mishra

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

et al., 2022; Wang et al., 2022d) and MULTIINSTRUCT dataset with instruction tuning.

B.4 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. We run all the experiments once.

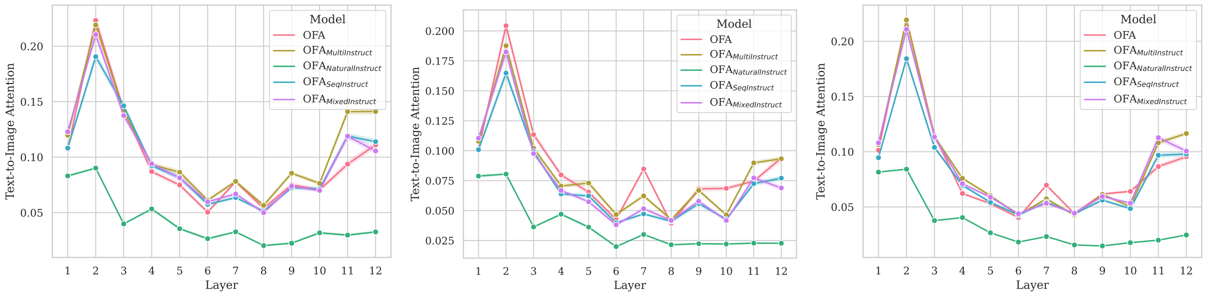
C Attention Analysis

In Section 6.1, we have demonstrated that fine-tuning OFA with NATURAL INSTRUCTIONS alone results in a decline in its zero-shot performance. In this section, we examine one possible reason for this decline by examining if fine-tuning the model on a text-only instruction dataset causes it to give less attention to image inputs.

To understand this, we conduct an analysis of the self-attention layers within the OFA encoder. The OFA encoder comprises 12 self-attention layers, each with 16 attention heads. We denote the input to self-attention layer l as $h^{(l)} = [x_1^{(l)}, \ldots, x_p^{(l)}, \ldots x_L^{(l)}]$, where L is the length of sequence. The input $h^{(0)} = [x_1^{(0)}, \ldots, x_I^{(0)}, x_{I+1}^{(0)}, \ldots x_{I+T}^{(0)}]$ to the first self-attention layer is actually the concatenation of image embeddings and text embeddings, where I, T is the length of image and text embeddings respectively. For ease of understanding and simplicity, we have altered the naming conventions and refer to $x_p^l, p = [1, ..., I]$ as image states and $x_p^l, p = [I + 1, ..., I + T]$ as text states.

For each self-attention layer, we first compute the attention given to the image states in relation to text states for each attention head. Specifically, for each text state as the query, we sum its attention scores on image states (i.e. the attention scores where the text state is the query and image states are the keys). We then compute the text-to-image attention across all text states. Finally, we average the text-to-image across all attention heads. This results in a text-to-image attention score for each self-attention layer.

Figure 5 illustrates the results of text-to-image attention scores on three unseen multimodal tasks: Text VQA, Visual Entailment, and Visual Text Extraction. The results on all three unseen tasks show that, in all self-attention layers of the OFA encoder, $\text{OFA}_{\text{NaturalInstruct}}$ has significantly lower text-to-image attention scores compared to other models. This decrease is particularly pronounced in the first two self-attention layers. This suggests that fine-tuning the model on a text-only instruction dataset leads to a reduction in the attention paid to image inputs, which may explain the decline in zero-shot performance.



(a) Text VQA (b) Visual Entailment (c) Visual Text Extraction

Figure 5: **Text-to-Image Attention of OFA Encoder.**

Category	Task Name	Dataset	Description	Exist
VQA	Open-Domain VQA	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>	✓
	Outside Knowledge VQA	OK-VQA (Marino et al., 2019)	Based on your knowledge, <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>	✓
Region Understanding	Most-Overlapping Region Selection	Visual Genome (Krishna et al., 2017)	Given the region <REGION>, decide which region in the options overlaps most with given region. <OPTION>	×
	Non-Overlapping Region Selection	Visual Genome (Krishna et al., 2017)	Which option does not share common area with <REGION>? <OPTION>	×
	Least-Overlapping Region Selection	Visual Genome (Krishna et al., 2017)	"Which option has the least shared area with <REGION>?<OPTION>	×
	Overlapping Region Selection	Visual Genome (Krishna et al., 2017)	Which region from options that has common area with <REGION>? <OPTION>	×
	Region Overlapping Detection	Visual Genome (Krishna et al., 2017)	Does <REGION1> share common area with <REGION2>? <OPTION>	×
	Region Area	Visual Genome (Krishna et al., 2017)	Compute the area of <REGION>.	×
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 53 multi-modal tasks, along with examples of the instructions for each task. The existing tasks are indicated with ✓, while the newly derived tasks are indicated using ×.

Category	Task Name	Dataset	Description	Exist
Image Understanding	Color Recognition	TDIUC (Kafle and Kanan, 2017)	Answer the question: <QUESTION> based on the color of an object. <OPTION>	✓
	Object Detection	TDIUC (Kafle and Kanan, 2017)	This task asks you to identify if an object appears in the image. <QUESTION><OPTION>	✓
	Object Recognition	TDIUC (Kafle and Kanan, 2017)	In this task you are asked a question about the type of an object in the image. <QUESTION><OPTION>	✓
	Scene Recognition	TDIUC (Kafle and Kanan, 2017)	Look at the environment in the image and answer the question accordingly. <QUESTION><OPTION>	✓
	Counting	TDIUC (Kafle and Kanan, 2017)	Question: <QUESTION> Please answer the question by counting the object mentioned in the question. <OPTION>	✓
	Sentiment Understanding	TDIUC (Kafle and Kanan, 2017)	Question: <QUESTION><OPTION> Please answer the question by interpreting the sentiment in the image.	✓
	Position Reasoning	TDIUC (Kafle and Kanan, 2017)	In this task, you need to analyze the position of objects in an image and answer the following question. <QUESTION><OPTION>	✓
	Utility Affordance	TDIUC (Kafle and Kanan, 2017)	Please take a look at the picture and answer the following question by thinking about what each object in the picture can be used for. <QUESTION><OPTION>	✓
	Sport Understanding	TDIUC (Kafle and Kanan, 2017)	There are some sports taking place in the image.<QUESTION><OPTION>	✓
	Image Quality	IQA (Chiu et al., 2020)	Select a reason from the options to explain why the image quality is bad. <OPTION>	✓
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-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>	×
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.	✓
	Image Captioning	MSCOCO (Lin et al., 2014)	Generate a sentence to describe the content of the image.	✓
Temporal Ordering	Wikihow Next Step Generation	WikiHow[7]	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 53 multi-modal tasks, along with examples of the instructions for each task. The existing tasks are indicated with ✓, while the newly derived tasks are indicated using ×.

Category	Task Name	Dataset	Description	Exist
VQA	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 (Zhu et al., 2016)	Which region is the answer to <QUESTION>? <OPTION>.	✓
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., 2022a)	Decide if the proposed spatial relationship between two objects in an image is "True" or "False"	✓
	Visual Entailment	SNLI-VE (Xie et al., 2019)	Can you conclude <TEXT> from the content of image? Select your answer from the options. <OPTION>	✓
	Commonsense Visual Question Answering	VCR (Zellers et al., 2019)	Look at the image and the regions in the question, <QUESTION>? <OPTION>.	✓
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>?	✓
	Disaster Type Classification	MEDIC (Alam et al., 2022)	What disaster happens in the image? <OPTION>	✓

Table 9: **Detailed Group of Evaluation Tasks Included in MULTIINSTRUCT.** The complete list of 9 multi-modal tasks, along with examples of the instructions for each task. The existing tasks are indicated with ✓, while the newly derived tasks are indicated using ×.

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 Pre-training Tasks in OFA.**

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 9

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract, section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

section 5, 6, 7, appendix B

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
appendix B.4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
appendix B.4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 5

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
section 3

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
section 3

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
We don't have paid participants involved

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
The data is publically available

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
The data is collected from publicly available benchmark datasets and there is no potential ethical issue.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.