CoTBal: Comprehensive Task Balancing for Multi-Task Visual Instruction Tuning

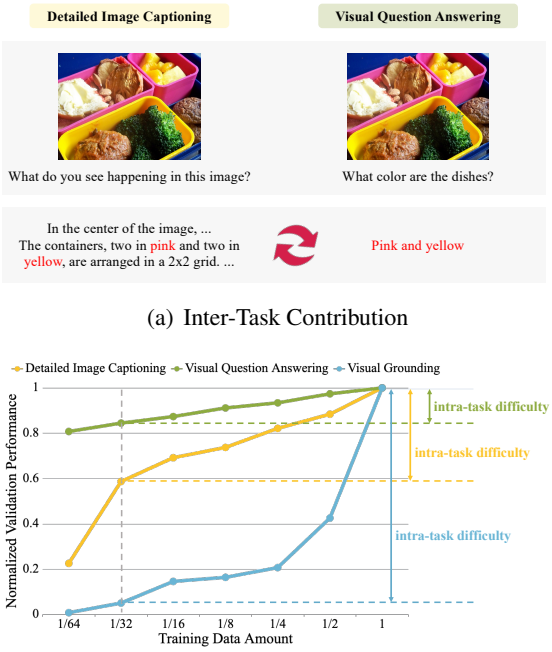
Anonymous ACL submission

Abstract

Visual instruction tuning is a key training stage of large multimodal models (LMMs). Nevertheless, the common practice of indiscriminately mixing instruction-following data from various tasks may result in suboptimal overall performance due to different instruction formats and knowledge domains across tasks. To mitigate this issue, we propose a novel **Co**mprehensive **T**ask **Bal**ancing (CoTBal) algorithm for multi-task visual instruction tuning of LMMs. To our knowledge, this is the first work that explores multi-task optimization in visual instruction tuning. Specifically, we consider two key dimensions for task balancing: (1) *Inter-Task Contribution*, the phenomenon where learning one task potentially enhances the performance in other tasks, attributable to the overlapping knowledge domains, and (2) *Intra-Task Difficulty*, which refers to the learning difficulty within a single task. By quantifying these two dimensions with performance-based metrics, task balancing is thus enabled by assigning more weights to tasks that offer substantial contributions to others, receive minimal contributions from others, and also have great intra-task difficulties. Experiments show that our CoTBal leads to superior overall performance in multi-task visual instruction tuning.

1 Introduction

Large multimodal models (LMMs) such as GPT-4V (Yang et al., 2023) and Gemini (Team et al., 2023) have attracted emerging attention for their ability to comprehend and reason across both visual and textual modalities. A key advancement in this field is visual instruction tuning (Liu et al., 2023b), which integrates visual encoders with large language models (LLMs) through specialized visual instructions and alignment modules. This innovative technique expands the inherent general-purpose capacities of LLMs to encompass the visual modality, significantly enhancing the training efficiency and effectiveness of LMMs. Approaches such as LLaVA (Liu et al., 2023b,a) and MiniGPT-4 (Zhu et al., 2023) have shown remarkable achievements through visual instruction tuning.



(a) Inter-Task Contribution

(b) Intra-Task Difficulty

Figure 1: Schematic illustrations of inter-task contributions and intra-task difficulties. (a) The red words indicate the overlapping knowledge domains among tasks, thereby enabling inter-task contributions. (b) The different curves correlating performance with training data amount reveal varying degrees of intra-task difficulties.

Typically, instruction-following data from various tasks are indiscriminately mixed for visual instruction tuning. However, simultaneous optimization across multiple tasks can lead to gradient conflicts (Yu et al., 2020) due to different instruction formats and knowledge domains across tasks, resulting in suboptimal overall performance. To magnitude this issue, based on the mixture of LoRA experts, Gou et al. (2023) utilizes distinct experts to learn conflicting tasks, which seems to be the unique work for multi-task visual instruc-

tion tuning. Note that multi-task learning (MTL) is mainly explored by designing model structures or optimization algorithms in previous works (Liu et al., 2019). The work of Gou et al. (2023) clearly falls into the first category of MTL. In contrast, we concentrate on applying the second category of MTL to visual instruction tuning in this paper.

Specifically, we propose a Generic Task Weighting (GTW) paradigm where losses are task-specific weighted and averaged at the token level. Based on the paradigm, we devise **Co**mprehensive **T**ask **Bal**ancing (CoTBal), a novel algorithm that balances multi-task visual instruction tuning according to both the inter-task contribution and the intra-task difficulty. On one hand, Figure 1(a) exemplifies that different tasks have overlapping knowledge domains, so that learning one task potentially enhances the performance in other tasks. The extent of this overlap varies, leading to differing degrees of inter-task contributions, which are quantified by the normalized validation performance of a model trained on one task and applied to others. On the other hand, Figure 1(b) shows that tasks exhibit distinct patterns of performance improvement with increasing training data amount. Tasks achieving near-optimal performance with a limited dataset are relatively simpler, while those requiring the full dataset for optimal performance have greater inherent learning difficulties. These intra-task difficulties are measured by the normalized validation performance gap between models trained on the full dataset and those trained on a mini subset of the same task. To achieve comprehensive task balancing for visual instruction tuning, we thus propose to assign more weights to three types of tasks: (1) tasks offering substantial contributions to others, (2) tasks receiving minimal contributions from others, and (3) tasks having great difficulties. These criteria are employed together in our CoTBal to obtain more balanced overall performance.

Briefly, our main contributions are three-fold:
(1) We propose the Generic Task Weighting (GTW) paradigm for multi-task visual instruction tuning. This is the first work that explores multi-task optimization in visual instruction tuning.
(2) We devise the Comprehensive Task Balancing (CoTBal) algorithm, which balances multi-task visual instruction tuning based on both the inter-task contribution and the intra-task difficulty.
(3) Experiments show that CoTBal outperforms existing methods, significantly improving overall performance while ensuring task balance.

2 Related Work

Multi-Task Learning. The purpose of Multi-task Learning (MTL) is jointly training a single model that can perform multiple tasks (Caruana, 1998; Ruder, 2017; Zhang and Yang, 2021; Vandenhende et al., 2021). Research in MTL is broadly divided into two categories: the first learns the correlations among tasks through model structures (Misra et al., 2016; Ma et al., 2018; Liu et al., 2019), and the second balances the joint training process of all tasks through optimization algorithms (Kendall et al., 2018; Lin et al., 2022; Sener and Koltun, 2018; Liu et al., 2021; Navon et al., 2022; Dai et al., 2023b). These two approaches are not mutually exclusive and can effectively complement each other (Liu et al., 2019). In this paper, we primarily focus on the multi-task optimization algorithm, which involves summing weighted losses or aggregating update gradients of all tasks.

Visual Instruction Tuning. Instruction tuning (Wei et al., 2021) is first explored in natural language processing, enabling large language models (LLMs) to follow textual instructions and accomplish unseen tasks (Zhang et al., 2023a; Ouyang et al., 2022; Wang et al., 2022). To extend the powerful capabilities of LLMs into multimodal domain, Liu et al. (2023b) introduces visual instruction tuning. This technique integrates visual encoders (Dosovitskiy et al., 2020) with LLMs (Touvron et al., 2023a,b) through specialized visual instructions and alignment modules, effectively constructing large multimodal models (LMMs) that can engage with vision-language information. Subsequently, a range of advanced approaches show robust performance on various visual tasks, focusing on two components: (1) training setting, which encompasses the selection of the alignment module (Zhu et al., 2023; Dai et al., 2023a; Bai et al., 2023) and the determination of trainable modules (Liu et al., 2023a; Ye et al., 2023), and (2) training data, characterized by its larger scale (Zhao et al., 2023), increased versatility (Zhang et al., 2023b; Li et al., 2023), and superior quality (Chen et al., 2023; Wang et al., 2023). However, Gou et al. (2023) observes that diverse tasks for visual instruction tuning focus on different perspectives, resulting in conflicts when trained together. To mitigate this, they propose the mixture of LoRA experts. In this paper, we tackle this challenge from a different angle by employing multi-task optimization, which assigns specific weights to each task.

3 Methodology

In this section, we start with a Generic Task Weighting (GTW) paradigm tailored for multi-task visual instruction tuning. Base on this paradigm, we elaborate on two key dimensions for task balancing: inter-task contribution balancing and intra-task difficulty balancing. These two dimensions are then integrated to formulate the final Comprehensive Task Balancing (CoTBal) algorithm.

3.1 Generic Task Weighting Paradigm

In current works involving visual instruction tuning, instruction-following data from various tasks are typically indiscriminately mixed for fine-tuning LMMs. The training loss is obtained by averaging the cross-entropy losses calculated across all valid tokens, as represented by the following formula:

$$L = \frac{\sum_{i=1}^{N}\sum_{j=1}^{S_i}\sum_{k=1}^{T_{ij}} -\log(p(t_{ijk}))}{\sum_{i=1}^{N}\sum_{j=1}^{S_i} T_{ij}}, \quad (1)$$

where N is the total number of tasks, S_i is the number of samples for Task i, T_{ij} is the number of valid tokens in the jth sample for Task i, and t_{ijk} is the kth valid token in the jth sample for Task i. However, this approach is incompatible with the task weighting paradigm of traditional multi-task optimization algorithms, where single-task losses are individually computed and aggregated through weighted summation to get the total loss. Therefore, we introduce the GTW paradigm, specifically tailored for multi-task visual instruction tuning. The training loss of GTW is defined as:

$$L_{GTW} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{S_i}\sum_{k=1}^{T_{ij}} -\lambda_i \log(p(t_{ijk}))}{\sum_{i=1}^{N}\sum_{j=1}^{S_i} \lambda_i T_{ij}}, \quad (2)$$

where λ_i denotes the weight of Task i. The losses are assigned task-specific weights and aggregated at the token level rather than at the sample or task level. GTW allows for more equitable consideration of each valid token, ensuring that the model is not biased towards certain tasks due to variations in sample sequence length or data amount across tasks. Besides, we also perform weighting in the denominator to enable a fair comparison with the indiscriminate data mixing approach (see Equation 1), where the weights are uniformly set to 1. The GTW paradigm is employed in our CoTBal algorithm, while also laying a solid foundation for subsequent studies.

3.2 Inter-Task Contribution Balancing

Although the focal points of distinct tasks vary in multi-task visual instruction tuning, a key shared objective exists: achieving more accurate comprehension and reasoning of visual information. As shown in Figure 1(a), the data of detailed image captioning on ShareGPT-4V (Chen et al., 2023) and visual question answering on VQAv2 (Goyal et al., 2017) both involve color information (pink and yellow dishes) in the image, which exemplifies the overlapping knowledge domains among tasks. Therefore, it is reasonable to hypothesize that different visual tasks could potentially provide mutual enhancement in their performance, which can be defined as the inter-task contribution. The extent of the overlapping knowledge domains varies, leading to differing degrees of inter-task contributions.

In practice, the inter-task contribution of Task i to Task j can be quantified by the validation performance for Task j of the model trained on Task i, which is normalized by the validation performance for Task j of the model trained on Task j itself. However, a model trained exclusively on one task may struggle to adhere to the instruction demands of other tasks. To address this, we incorporate mini subsets from all tasks into the training set, enabling the model to understand the instruction demands of each task. Therefore, the inter-task contribution of Task i to Task j can be calculated as:

$$C_{ij} = \frac{V_j(i+mini) - V_j(mini)}{V_j(j+mini) - V_j(mini)}, \quad (3)$$

where $V_j(i+mini)$ represents the validation performance for Task j of a model trained on the full dataset from Task i alongside mini subsets from other tasks, and $V_j(mini)$ signifies the validation performance for Task j of a model trained on mini subsets from all tasks. In the formula, $V_j(mini)$ is subtracted from both the numerator and the denominator to mitigate the impact of incorporating mini subsets from all tasks into the training set on the validation performance for Task j.

Furthermore, based on the accurate quantification of the inter-task contribution, we propose two task weighting strategies for inter-task contribution balancing. Firstly, we examine the average inter-task contribution of one given task to all other tasks as $C_{one2all}$, representing the extent to which this task assists all other tasks. The greater the assistance provided by one task to all other tasks, the more substantial its overall contribution to the entire training process of multi-task visual instruction

tuning. Therefore, tasks that have greater $C_{one2all}$ should be assigned more weights to enhance overall performance. The specific task weights $\boldsymbol{\lambda_{one2all}}$ can be computed as:

$$C_{one2all,i} = \frac{1}{N-1} \sum_{j \neq i} C_{ij}, \tag{4}$$

$$\boldsymbol{\lambda_{one2all}} = N \times \text{softmax}(\frac{\boldsymbol{C_{one2all}}}{T}), \tag{5}$$

where $C_{one2all,i}$ signifies $C_{one2all}$ for Task i and $\boldsymbol{C_{one2all}}$ represents the N-dimensional vector of $C_{one2all}$ for all tasks. T denotes the temperature hyperparameter that controls the degree of smoothness in the weight vector. Secondly, we consider the average inter-task contribution of all other tasks to one given task as $C_{all2one}$, denoting the degree to which this task receives benefits from all other tasks. If one task receives minimal benefits from other tasks, it tends to exhibit poorer performance compared to tasks that receive greater benefits. To maintain balanced overall performance, such type of tasks that have lower $C_{all2one}$ should also be assigned more weights. The specific task weights $\boldsymbol{\lambda_{all2one}}$ can be computed as:

$$C_{all2one,i} = \frac{1}{N-1} \sum_{j \neq i} C_{ji}, \tag{6}$$

$$\boldsymbol{\lambda_{all2one}} = N \times \text{softmax}(-\frac{\boldsymbol{C_{all2one}}}{T}), \tag{7}$$

where $C_{all2one,i}$ signifies $C_{all2one}$ for Task i and $\boldsymbol{C_{all2one}}$ represents the N-dimensional vector of $C_{all2one}$ for all tasks. T denotes the same temperature hyperparameter in Equation 5. Subsequently, we integrate the aforementioned two strategies to formulate the task weighting strategy for inter-task contribution balancing, where the task weights $\boldsymbol{\lambda_C}$ can be calculated as:

$$\boldsymbol{\lambda_C} = \frac{1}{2}(\boldsymbol{\lambda_{one2all}} + \boldsymbol{\lambda_{all2one}}). \tag{8}$$

3.3 Intra-Task Difficulty Balancing

In addition to the inter-task contribution, another critical aspect in multi-task visual instruction tuning is the intra-task difficulty, which refers to the inherent learning difficulty within each task. Tasks that achieve near-optimal performance with a limited dataset are considered to have poor intra-task difficulties. Conversely, tasks that require the full dataset to reach optimal performance are deemed to have great intra-task difficulties. As illustrated in Figure 1(b), different tasks exhibit distinct patterns of performance improvement with increasing training data amount. Arranged by increasing intra-task difficulty, the sequence of these three tasks is as follows: visual question answering on VQAv2 (Goyal et al., 2017), detailed image captioning on ShareGPT-4V (Chen et al., 2023) and visual grounding on RefCOCO (Kazemzadeh et al., 2014; Mao et al., 2016).

Practically, the intra-task difficulty for Task i is measured by the validation performance gap between a model trained on the full dataset and that trained on a mini subset from Task i, which is normalized by the validation performance of the former model. This metric offers a precise measure of potential performance degradation when using the mini subset of training data, thereby reflecting the inherent learning difficulty of the task. Notably, to ensure a fair measurement across each task, the ratio between the number of samples in the mini subset and the total number of samples in the full dataset should be kept consistent.

However, training extra models using both the full dataset and the mini subset from each task is necessary to obtain the intra-task difficulty, which will require additional time comparable to the training time of the final model. To alleviate this, we repurpose the models trained for computing inter-task contributions. Specifically, we substitute the model trained on the mini subset from Task i with that trained on mini subsets from all tasks, and replace the model trained solely on the full dataset from Task i with that trained on the full dataset from Task i alongside mini subsets from other tasks. Due to the minimal inter-task contributions of others tasks to Task i when compared to the contribution from Task i to Task i itself, the impact of mini subsets from other tasks on the validation performance for Task i is negligible. Therefore, this approach significantly reduces training time with minimal error. The intra-task difficulty for Task i is calculated as:

$$D_i = 1 - \frac{V_i(mini)}{V_i(i + mini)}, \tag{9}$$

where $V_i(mini)$ represents the validation performance for Task i of a model trained on mini subsets from all tasks, and $V_i(i + mini)$ denotes the validation performance for Task i of a model trained on the full dataset from Task i coupled with mini subsets from other tasks.

Moreover, owing to the varying intra-task difficulties across different tasks, treating each task

Algorithm 1 Overall Training Process of CoTBal

Input: N visual tasks, a pretrained LMM.
Output: a fine-tuned LMM.

```
Trained a model on mini subsets from all tasks;
for i = 1 to N do
    Trained a model on the full dataset from Task i and mini subsets from other tasks;
end for
for each Task i do
    for each other Task j do
        Compute inter-task contribution C_ij;
    end for
end for
for each Task i do
    Compute outwards contribution C_one2all,i;
    Compute inwards contribution C_all2one,i;
end for
Compute task weights λ_C using C_one2all and C_all2one for inter-task contribution balancing;
for each Task i do
    Compute intra-task difficulty D_i;
end for
Compute task weights λ_D using D for intra-task difficulty balancing;
Combine λ_C and λ_D to get final task weights λ_CoTBal for comprehensive task balancing;
Apply λ_CoTBal to fine-tune the final LMM using the GTW paradigm.
```

equally during the training process may result in underfitting of the more challenging tasks, despite the simpler ones being adequately trained or even overfitted. Therefore, we propose a weighting strategy that assigns more weights to tasks with greater intra-task difficulties. The task weights $\boldsymbol{\lambda_D}$ can be calculated as:

$$\boldsymbol{\lambda_D} = N \times \text{softmax}(\frac{\boldsymbol{D}}{T}), \tag{10}$$

where $\boldsymbol{D}$ represents the N-dimensional vector of intra-task difficulties for all tasks, and T is the same temperature hyperparameter used in Section 3.2.

3.4 Comprehensive Task Balancing

After individually establishing the strategies for inter-task contribution balancing and intra-task difficulty balancing, the final step involves integrating them to create the CoTBal algorithm. The algorithm is designed to synergistically leverage the strengths of both two balancing methods, thereby ensuring a more comprehensive and effective multi-task optimization process in visual instruction tuning. The specific task weights $\boldsymbol{\lambda_{CoTBal}}$ for comprehensive task balancing can be calculated as:

$$\boldsymbol{\lambda_{CoTBal}} = \alpha\boldsymbol{\lambda_C} + (1-\alpha)\boldsymbol{\lambda_D}, \tag{11}$$

where α is a hyperparameter that controls the relative influence of inter-task contribution balancing and intra-task difficulty balancing. The training process of CoTBal is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets. The training data of CoTBal includes a variety of datasets: ShareGPT4V (Chen et al., 2023), VQAv2 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), ChartQA (Masry et al., 2022), OCRVQA (Mishra et al., 2019), RefCOCO (Kazemzadeh et al., 2014; Mao et al., 2016) and ShareGPT (sha, 2023). The aforementioned visual datasets have various image domains and task types. Therefore, we treat each visual dataset as a distinct task, except for the RefCOCO dataset, which is bifurcated into two tasks: RefCOCO-caption and RefCOCO-grounding. The former generates descriptions for image regions defined by bounding boxes (bbox), while the latter produces bbox corresponding to described image regions. Besides, the ShareGPT dataset, only containing language conversation data, is also used as a training task to mitigate the phenomenon of model forgetting its inherent general language conversation capabilities.

Inspired by Liu et al. (2023a), we incorporate response format instructions into the data to clarify task requirements for the model and employ multiple data processing strategies to reduce training costs and ensure fairness, detailed as follows:
(1) For ShareGPT4V, the data is randomly partitioned into a validation set of 2k and a test set of 2k, with the remainder designated for training.
(2) For all VQA datasets and RefCOCO, data from the same training image are shuffled and merged into a single conversation.
(3) For RefCOCO, training conversations are segmented into parts, each with fewer than 10 turns.
(4) For OCRVQA, 80k conversations are sampled from the training set.
(5) For VQAv2, GQA and OCRVQA, 20k data are sampled from the validation set.
(6) For ShareGPT, invalid conversations are filtered out as Zheng et al. (2023), while long conversations that surpass 2048 tokens are truncated.

Tasks	Data Sizes	Response Format Instructions
ShareGPT	41k	–
ShareGPT-4V	98k	
VQAv2	83k	Answer the question using a single word or phrase.
GQA	72k	
ChartQA	18k	
OCRVQA	80k	
RefCOCO-caption	41k	Provide a short description for this region.
RefCOCO-bbox	41k	Provide the bounding box coordinate of the region this sentence describes.
Total	475k	

Table 1: Summary of training data sizes and response format instructions for each task.

The training data sizes and response format instructions for each task are presented in Table 1.

Evaluation Metrics. In the experiments, we first report the common evaluation metrics for each task: CIDEr (Vedantam et al., 2015) for image captioning tasks, Exact Match (EM) for visual question answering tasks, and Intersection over Union (IoU) for visual grounding tasks. Moreover, since multi-task visual instruction tuning aims to jointly improve performance across all tasks, we consider two metrics to comprehensively evaluate the effectiveness of methods: (1) $\Delta I\%$, the average per-task improvement, and (2) $\Delta E\%$, the average per-task error in test performance compared with models trained on individual tasks. These two metrics can be calculated as:

$$I_i = \frac{1}{K_i}\sum_{j=1}^{K_i}(-1)^{\delta_{ij}}\frac{M_{e,ij} - M_{b,ij}}{M_{b,ij}}, \tag{12}$$

$$\Delta I\% = \frac{1}{N}\sum_{i=1}^{N} I_i, \tag{13}$$

$$\Delta E\% = \frac{1}{N}\sum_{i=1}^{N} \min(0, I_i), \tag{14}$$

where N is the total number of tasks, I_i is the test performance improvement for Task i, K_i is the number of evaluation metrics for Task i, $M_{e,ij}$ is the value on the jth metric for Task i of the model trained by the evaluated method and $M_{b,ij}$ is that of the baseline model trained individually on Task i. δ_{ij} is an indicator function that is set to 0 if a higher value is better on the k-th metric for Task i, and 1 otherwise. The metric $\Delta E\%$ serves as an indicator of imbalance in model performance by focusing on the negative aspects of the performance improvement, i.e., where there is no improvement or even a decline in performance compared to baseline models. By aggregating these negative impacts across all tasks, $\Delta E\%$ provides a concise measure of how the method may disproportionately benefit some tasks at the expense of others, thus revealing the degree of performance imbalance.

Compared Methods. We compare the following methods: (1) our CoTBal algorithm; (2) Single-Task Learning (STL) baseline, training and testing independent models for each task; (3) Equal Weighting (EW), the indiscriminate data mixing approach which minimizes the loss in Equation 1 without task weighting; (4) Task-Level Aggregation (TLA), which averages the losses of valid tokens within each task, then calculates the mean loss across all tasks; (5) Random Loss Weighting (RLW) (Lin et al., 2021), which randomly assigns task weights; (6) Dynamic Weight Average (DWA) (Liu et al., 2019), which assigns more weights to tasks with less descending rates of training loss; (7) Improvable Gap Balancing version 1 (IGBv1) (Dai et al., 2023b), which assigns more weights to tasks with greater training losses. Method (5)-(7) are optimization algorithms in the traditional MTL framework, dynamically updating task weights in each training iteration. We apply the GTW paradigm to them for multi-task visual instruction tuning.

Note that we have not compared traditional gradient aggregation multi-task optimization algorithms. Such algorithms require computing update gradients via backpropagation for each task separately in each iteration, followed by the aggregation of gradients across all tasks. In multi-task visual instruction tuning, the large number of tasks and the massive volume of model parameters make this process impractical and excessively time-consuming.

Implementation Details. In the experiments, we fine-tune the pretrained LLaVA-v1.5-7B model on 8×A100 (80G) GPUs using the same training setting and code as Liu et al. (2023a). For our CoTBal algorithm, we perform task balancing across all seven visual instruction-following tasks, while directly assigning a weight of 1.0 to ShareGPT. The mini subset from each task is obtained by randomly sampling $1/32$th of the full dataset from that task. Both the temperature hyperparameter T and the control hyperparameter α are set to 0.5.

4.2 Multi-Task Evaluations

Table 2 presents the comparative results for multi-task instruction tuning. With the same foundational models and training data, CoTBal achieves the op-

Methods	ShareGPT4V test CIDEr↑	RefCOCO-caption Ref-test CIDEr↑	RefCOCO-caption Ref-testB CIDEr↑	RefCOCO-caption Refg-test CIDEr↑	VQAv2 test-dev EM↑	GQA test-bal EM↑	ChartQA test EM↑	OCRVQA test EM↑	RefCOCO-bbox Ref-test IoU↑	RefCOCO-bbox Ref-testB IoU↑	RefCOCO-bbox Refg-test IoU↑	$\Delta I\%\uparrow$	$\Delta E\%\downarrow$
STL	0.1285	0.4330	0.4658	0.6019	77.73	61.23	17.76	68.22	65.02	51.58	50.78		
EW	0.1411	0.4738	0.5591	0.5937	78.27	**62.20**	19.60	67.73	76.05	61.63	62.80	7.30	**0.10**
TLA	0.1144	**0.5083**	**0.5770**	0.5327	77.72	60.42	**22.36**	67.80	71.79	56.58	58.40	4.94	1.85
RLW	0.1388	0.4810	0.5571	0.5538	77.28	60.61	18.20	66.73	70.78	55.86	57.01	3.44	0.54
DWA	0.1225	0.4659	0.5470	0.6006	**78.28**	61.82	19.88	**67.87**	76.74	61.12	63.88	5.35	0.74
IGBv1	0.1349	0.4267	0.4824	**0.6017**	77.00	60.92	17.20	65.96	70.39	55.47	55.99	0.92	1.13
CoTBal	**0.1437**	0.4649	0.5724	0.5874	77.99	61.81	20.16	67.48	**82.62**	**67.38**	**69.19**	**9.45**	0.15

Table 2: Comparative results for multi-task visual instruction tuning. ↑ (↓) indicates that the higher (lower) the result, the better the performance. Ref-test and Ref-testB represent two test sets of Kazemzadeh et al. (2014), and Refg-test denotes the test set of Mao et al. (2016).

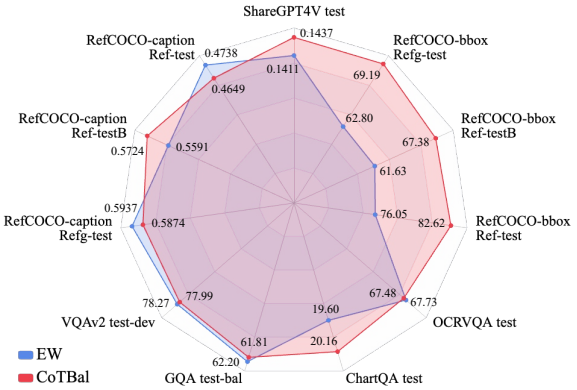
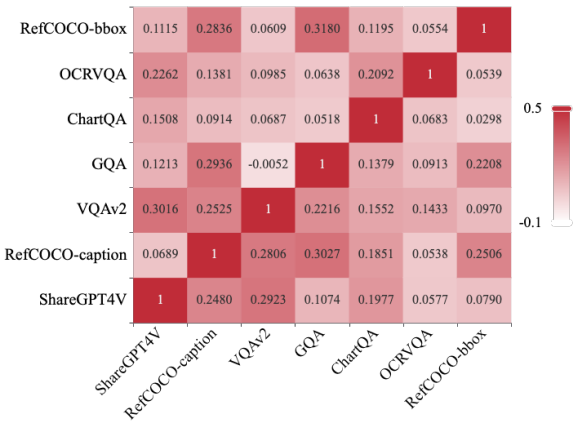
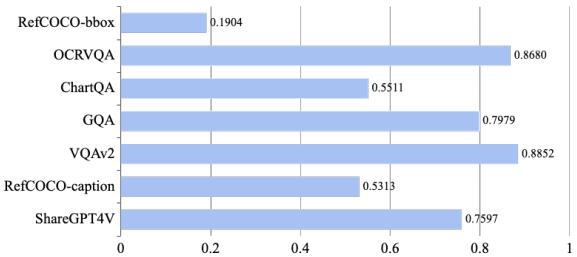


Figure 2: Performance comparison radar chart of the CoTBal method and the EW method.



(a) Heatmap of inter-task contributions.



(b) Histogram of intra-task difficulties.

Figure 3: Numerical visualizations of inter-task contributions and intra-task difficulties in the training process of the CoTBal algorithm.

timal average per-task performance improvement ($\Delta I\%$), alongside maintaining the near-lowest average per-task performance error ($\Delta E\%$). As shown in Figure 2, compared to the most commonly employed EW method, CoTBal significantly enhances the performance on ShareGPT4V, ChartQA and RefCOCO-bbox tasks while keeping competitive performance on other tasks. This validates the effectiveness of our algorithm in terms of both overall performance and the degree of performance imbalance. Specifically, as depicted in Figure 3, CoTBal effectively captures the variances in mutual contributions and inherent learning difficulties among these visual tasks, thereby providing appropriate task weights for the final model training, which maximally mitigates task conflicts.

Besides, we observe that TLA is significantly inferior to EW in both $\Delta I\%$ and $\Delta E\%$. TLA guarantees equality for each task in the final loss computation. However, variations in sample sequence length and data amount across different tasks may implicitly assign inappropriate task weights to the losses of valid tokens. The implicit weight is inversely related to the total number of valid tokens in each task, leading to poorer overall performance and a marked performance imbalance. This indicates that maintaining equality at the token level is more logical than preserving it at the task level, thereby demonstrating the viability of the GTW paradigm in multi-task visual instruction tuning.

As for the compared traditional multi-task optimization algorithms (RLW, DWA and IGBv1), directly applying them to visual instruction tuning yields suboptimal results in both $\Delta I\%$ and $\Delta E\%$. We contend that assigning task weights based on

Methods	$\Delta I\% \uparrow$	$\Delta E\% \downarrow$
EW	7.30	0.10
CoTBal ($T=2.0$)	8.25	0.12
CoTBal ($T=1.0$)	8.41	0.11
CoTBal ($T=0.5$)	9.45	0.15
CoTBal ($\boldsymbol{\lambda}_{one2all}$)	7.87	0.10
CoTBal ($\boldsymbol{\lambda}_{all2one}$)	7.05	0.05
CoTBal ($\boldsymbol{\lambda}_{C}$)	7.09	0.06
CoTBal ($\boldsymbol{\lambda}_{D}$)	10.39	0.30
CoTBal ($\boldsymbol{\lambda}_{CoTBal}$)	9.45	0.15
CoTBal (precise Difficulty)	9.20	0.16
CoTBal (real Difficulty)	9.45	0.15

Table 3: Ablation results for multi-task visual instruction tuning. T is the temperature hyperparameter, CoTBal ($\boldsymbol{\lambda}$) denotes the exclusive use of the specific $\boldsymbol{\lambda}$ for task weighting, and CoTBal (precise / real Difficulty) signifies the employment of the precise or real calculation approach for the intra-task difficulty.

training losses is imprecise, because the fine-tuning losses in large models fails to accurately reflect training progress. This is also the reason why both the inter-task contribution and the intra-task difficulty are quantified by performance-based metrics in our CoTBal algorithm.

4.3 Ablation Studies

As shown in Table 3, we analyze the impact of different training settings on model performance from three aspects: the temperature hyperparameter configuration, the task weighting strategy selection and the calculation approach for intra-task difficulties. The complete ablation results are presented in Appendix A. The compared methods include: EW; CoTBal ($T=2.0/1.0/0.5$) where the temperature hyperparameter T is set to 2.0, 1.0 or 0.5; CoTBal ($\boldsymbol{\lambda}_{one2all}/\boldsymbol{\lambda}_{all2one}/\boldsymbol{\lambda}_{C}/\boldsymbol{\lambda}_{D}/\boldsymbol{\lambda}_{CoTBal}$) where task weights are set as $\boldsymbol{\lambda}_{one2all}$, $\boldsymbol{\lambda}_{all2one}$, $\boldsymbol{\lambda}_{C}$, $\boldsymbol{\lambda}_{D}$ or $\boldsymbol{\lambda}_{CoTBal}$; and CoTBal (precise / real Difficulty) where the precise or real calculation approach for the intra-task difficulty is employed. Specifically, the precise calculation approach trains extra models using the full dataset and the mini subset from each task, while the real calculation approach repurposes the models trained for computing inter-task contributions to reduce additional training time.

In terms of the temperature hyperparameter configuration: CoTBal consistently outperforms EW in $\Delta I\%$, maintaining its superiority across all T values and enhancing its advantage as T decreases. The degree of task balancing increases as T decreases, leading to an improved $\Delta I\%$, which demonstrates the efficacy of comprehensive task balancing. Conversely, CoTBal exhibits a slight increase in $\Delta E\%$ as T decreases. When the degree of non-smoothness in task weights becomes excessive, tasks with significantly smaller weights inevitably underperform, resulting in the slight imbalance in performance.

In terms of the task weighting strategy selection: On the one hand, compared to the EW method, CoTBal ($\boldsymbol{\lambda}_{one2all}$) enhances $\Delta I\%$ while maintaining $\Delta E\%$ constant, due to its preference for tasks that offer substantial contributions to other tasks. On the other hand, CoTBal ($\boldsymbol{\lambda}_{all2one}$) significantly reduces $\Delta E\%$, mitigating the performance imbalance issue by prioritizing tasks that receive minimal benefits from other tasks. CoTBal ($\boldsymbol{\lambda}_{C}$) integrates the aforementioned two strategies, achieving more balanced $\Delta I\%$ and $\Delta E\%$. Moreover, CoTBal ($\boldsymbol{\lambda}_{D}$) markedly enhances $\Delta I\%$ by assigning more weights to tasks that have greater learning difficulties, yet concurrently exacerbates the performance imbalance issue. Finally, CoTBal ($\boldsymbol{\lambda}_{CoTBal}$) integrates all three strategies to maximize overall performance while mitigating the performance imbalance issue.

In terms of the calculation approach for the intra-task difficulty: The precise approach and the real approach exhibit similar levels of performance, with the real one even marginally surpassing the precise one in both $\Delta I\%$ and $\Delta E\%$. When calculating the intra-task difficulty of Task i, training with mini subsets from any other tasks has negligible impact on performance in Task i, hence our CoTBal algorithm employs the real calculation approach to significantly reduce training time while ensuring performance.

5 Conclusion

In this paper, we devise Comprehensive Task Balancing (CoTBal), the first multi-task optimization algorithm tailored for visual instruction tuning of LMMs. Specifically, we first propose the Generic Task Weighting (GTW) paradigm. Based on this paradigm, we then design three task weighting strategies according to the inter-task contribution and the intra-task difficulty. Our experiments demonstrate that CoTBal outperforms existing methods, including the indiscriminate data mixing approach, significantly improving overall performance while ensuring task balance.

Limitations

Although the proposed CoTBal algorithm enhances the performance of multi-task visual instruction tuning, it still presents two small drawbacks. Firstly, CoTBal necessitates extra time for the computation of the inter-task contribution and the intra-task difficulty. Specifically, the extra time is approximately $(1 + (N - 1)/32)$ times the duration needed to train the final model, where N is the number of tasks. Secondly, the measurement of the inter-task contribution and the intra-task difficulty could be further improved. It is rather an indirect metric assessed through validation performance, which may introduce slight noise. In our ongoing research, we will make further efforts on multi-task visual instruction tuning to overcome these drawbacks.

References

2023. Sharegpt. https://sharegpt.com/.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *ArXiv Preprint ArXiv:2308.12966*.

Rich Caruana. 1998. *Multitask learning*. Springer.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023. Sharegpt4v: Improving large multimodal models with better captions. *ArXiv Preprint ArXiv:2311.12793*.

W Dai, J Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. 2023a. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv Preprint ArXiv:2305.06500*.

Yanqi Dai, Nanyi Fei, and Zhiwu Lu. 2023b. Improvable gap balancing for multi-task learning. In *Uncertainty in Artificial Intelligence*, pages 496–506. PMLR.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv Preprint ArXiv:2010.11929*.

Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. 2023. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *ArXiv Preprint ArXiv:2312.12379*.

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.

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 Conference on Computer Vision and Pattern Recognition*, pages 6700–6709.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 787–798.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *ArXiv Preprint ArXiv:2306.00890*.

Baijiong Lin, YE Feiyang, Yu Zhang, and Ivor Tsang. 2022. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W Tsang. 2021. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *ArXiv Preprint ArXiv:2111.10603*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *ArXiv Preprint ArXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *ArXiv Preprint ArXiv:2304.08485*.

Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Towards impartial multi-task learning. In *International Conference on Learning Representations*.

Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1930–1939.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *ArXiv Preprint ArXiv:2203.10244*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition*, pages 947–952. IEEE.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pages 16428–16446. PMLR.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *ArXiv Preprint ArXiv:1706.05098*.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *ArXiv Preprint ArXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *ArXiv Preprint ArXiv:2302.13971*.

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

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3614–3633.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong, Weijia Li, Wei Li, Jiaqi Wang, et al. 2023. Vigc: Visual instruction generation and correction. *ArXiv Preprint ArXiv:2308.12714*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *ArXiv Preprint ArXiv:2212.10560*.

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. *ArXiv Preprint ArXiv:2109.01652*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *ArXiv Preprint ArXiv:2309.17421*, 9(1).

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *ArXiv Preprint ArXiv:2304.14178*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023a. Instruction tuning for large language models: A survey. *ArXiv Preprint ArXiv:2308.10792*.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023b. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *ArXiv Preprint ArXiv:2306.17107*.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

Bo Zhao, Boya Wu, and Tiejun Huang. 2023. Svit: Scaling up visual instruction tuning. *ArXiv Preprint ArXiv:2307.04087*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv Preprint ArXiv:2306.05685*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *ArXiv Preprint ArXiv:2304.10592*.

A Complete Results of Ablation Studies

We report the complete results of ablation studies for multi-task visual instruction tuning in Table 4.

Methods	ShareGPT4V	RefCOCO-caption			VQAv2	GQA	ChartQA	OCRVQA	RefCOCO-bbox				
	test	Ref-test	Ref-testB	Refg-test	test-dev	test-bal	test	test	Ref-test	Ref-testB	Refg-test	$\Delta I\% \uparrow$	$\Delta E\% \downarrow$
	CIDEr↑		CIDEr↑		EM↑	EM↑	EM↑	EM↑		IoU↑			
EW	0.1411	0.4738	0.5591	0.5937	78.27	62.20	19.60	67.73	76.05	61.63	62.80	7.30	0.10
CoTBal ($T=2.0$)	0.1433	0.4540	0.5642	0.5973	78.10	62.08	20.16	67.67	78.01	63.06	64.73	8.25	0.12
CoTBal ($T=1.0$)	0.1369	0.4605	0.5752	0.5948	78.15	62.09	20.32	67.71	80.61	65.00	66.78	8.41	0.11
CoTBal ($T=0.5$)	0.1437	0.4649	0.5724	0.5874	77.99	61.81	20.16	67.48	82.62	67.38	69.19	9.45	0.15
CoTBal ($\lambda_{one2all}$)	0.1448	0.4528	0.5763	0.6036	78.34	62.16	19.52	67.72	77.59	61.81	63.47	7.87	0.10
CoTBal ($\lambda_{all2one}$)	0.1333	0.4617	0.5520	0.5961	78.25	62.12	20.20	68.00	76.98	62.61	64.12	7.05	0.05
CoTBal (λ_{C})	0.1340	0.4645	0.5626	0.5934	78.39	62.27	20.04	67.92	77.46	61.92	63.67	7.09	0.06
CoTBal (λ_{D})	0.1455	0.4783	0.5706	0.5963	77.46	61.30	20.08	67.04	85.13	71.52	72.88	10.39	0.30
CoTBal (λ_{CoTBal})	0.1437	0.4649	0.5724	0.5874	77.99	61.81	20.16	67.48	82.62	67.38	69.19	9.45	0.15
CoTBal (precise Difficulty)	0.1345	0.4767	0.5604	0.5952	78.00	61.87	21.04	67.46	82.06	67.83	69.08	9.20	0.16
CoTBal (real Difficulty)	0.1437	0.4649	0.5724	0.5874	77.99	61.81	20.16	67.48	82.62	67.38	69.19	9.45	0.15

Table 4: Complete results of ablation studies for multi-task visual instruction tuning. ↑ (↓) indicates that the higher (lower) the result, the better the performance. Ref-test and Ref-testB represent two test sets of Kazemzadeh et al. (2014), and Refg-test denotes the test set of Mao et al. (2016).