Adaptive Task Balancing for Visual Instruction Tuning via Inter-Task Contribution and Intra-Task Difficulty

Yanqi Dai*
yanqidai@ruc.edu.cn
Renmin University of China
Beijing, China

Yong Wang
wangyong.lz@alibaba-inc.com
AMAP, Alibaba Group
Beijing, China

Zebin You
zebin@ruc.edu.cn
Renmin University of China
Beijing, China

Dong Jing
jingdong98@ruc.edu.cn
Renmin University of China
Beijing, China

Xiangxiang Chu
chuxiangxiang.cxx@alibaba-inc.com
AMAP, Alibaba Group
Beijing, China

Zhiwu Lu†
luzhiwu@ruc.edu.cn
Renmin University of China
Beijing, China

Abstract

Visual instruction tuning is a key training stage of large multimodal models. However, when learning multiple visual tasks simultaneously, this approach often results in suboptimal and imbalanced overall performance due to latent knowledge conflicts across tasks. To mitigate this issue, we propose a novel **A**daptive **T**ask **B**alancing approach tailored for **vis**ual instruction tuning (**VisATB**). Specifically, we measure two critical dimensions for visual task balancing based on validation performance: (1) *Inter-Task Contribution*, the mechanism where learning one task enhances the performance on others owing to shared knowledge across tasks, and (2) *Intra-Task Difficulty*, which denotes the inherent learning difficulty of a single task. Furthermore, we propose prioritizing three categories of tasks with greater weight: those that offer substantial contributions to others, those that receive minimal contributions from others, and those that present high learning difficulties. Among these three task weighting strategies, the first and third focus on improving overall performance, and the second targets the mitigation of performance imbalance. Extensive experiments on three benchmarks demonstrate that our VisATB approach consistently achieves superior and more balanced overall performance in visual instruction tuning. The data, code, and models are available at YanqiDai/VisATB.

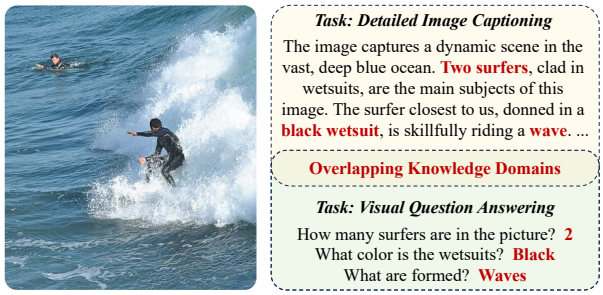
CCS Concepts

• **Computing methodologies** → **Natural language generation**; **Scene understanding**; **Object detection**; **Object recognition**.

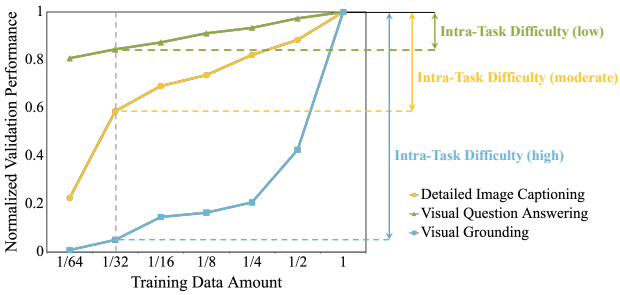
Keywords

LMMs, Visual Instruction Tuning, Task Balancing

ACM Reference Format:
Yanqi Dai, Yong Wang, Zebin You, Dong Jing, Xiangxiang Chu, and Zhiwu Lu. 2026. Adaptive Task Balancing for Visual Instruction Tuning via Inter-Task Contribution and Intra-Task Difficulty. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792371



(a) Inter-Task Contribution



(b) Intra-Task Difficulty

Figure 1: Schematic illustrations of inter-task contributions and intra-task difficulties. (a) The red words reveal that different tasks have overlapping knowledge domains, enabling inter-task contributions. (b) The different performance improvement trajectories w.r.t. training data amount reflect distinct degrees of intra-task difficulties.

1 Introduction

Large multimodal models (LMMs) [2, 32, 54] have garnered significant attention for their capability to understand and reason across both visual and textual modalities. A pivotal advancement in this field is visual instruction tuning [33], which integrates visual encoders with large language models (LLMs) using visual instruction-following data and specialized alignment modules. This innovative technique extends the robust, general-purpose capabilities of LLMs to the visual modality, substantially enhancing both the efficiency

*Part of this work was done during Yanqi Dai's internship at AMAP, Alibaba Group.
†Corresponding author.

This work is licensed under a Creative Commons Attribution 4.0 International License.

WWW '26, Dubai, United Arab Emirates

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792371

and effectiveness of LMM training. Many approaches, such as the LLaVA series [23, 31–33], have achieved remarkable results through visual instruction tuning.

To endow LMMs with diverse visual abilities, instruction-following data from multiple visual tasks are frequently combined indiscriminately for visual instruction tuning [5]. However, this approach faces a critical challenge: different tasks necessitate task-specific latent knowledge. For instance, the image captioning task demands comprehensive scene understanding and holistic description generation, whereas the visual grounding task emphasizes fine-grained localization of specific textual phrases. Consequently, simultaneous training across multiple tasks imposes a substantial learning burden on the model due to *task interference*, potentially resulting in suboptimal performance compared to training on each task individually [8, 14]. Manual specification of data mixing ratios or task weights heavily relies on extensive ablation studies and expert knowledge, rendering such approaches both resource-intensive and difficult to generalize across different model architectures and tasks.

To mitigate this issue, we perform a rigorous analysis of the task relationship and identify two key concepts: **inter-task contribution** and **intra-task difficulty**. First, as illustrated in Figure 1a, we observe that tasks often share overlapping knowledge domains, facilitating knowledge transfer that boosts performance in related tasks. The extent of these overlaps varies across tasks, leading to differing degrees of inter-task contributions. Second, Figure 1b reflects that different tasks demonstrate distinct performance improvement trajectories as training data increases. Specifically, tasks that achieve near-optimal performance with limited training data are regarded as relatively simple, whereas tasks that require extensive training data to reach optimal performance exhibit higher inherent learning difficulty.

Moreover, we introduce a novel **A**daptive **T**ask **B**alancing approach for **vis**ual instruction tuning (**VisATB**) based on the above two critical perspectives, integrating three task weighting strategies, each with its unique characteristics, as presented in Figure 2. **In the preparation stage**, we measure the inter-task contribution of one task to another task by training a model on one task and evaluating its normalized validation performance on the other task. Additionally, to quantify the intra-task difficulty of a target task, we estimate the normalized validation performance gap between a model trained on a mini subset of the task and one trained on the entire dataset or a sufficiently large subset. Subsequently, **in the task weight calculation stage**, we recommend assigning greater weight to tasks that (1) offer substantial contribution to others, (2) receive minimal contribution from others, and (3) present high learning difficulties. Among these three task weighting strategies, the first and third focus on improving overall performance, while the second aims to mitigate performance imbalance across tasks. *VisATB integrates them to achieve a more robust and balanced overall performance.* **In the final training stage**, we propose a Visual Instruction Task Weighting (VITW) paradigm tailored for visual instruction tuning, where losses are assigned task-specific weights and averaged at the token level. Building upon this paradigm, the final model is trained using the entire dataset of all tasks and the integrated task weight.

Our contributions can be summarized as follows:

(1) We identify two key concepts for visual task balancing: inter-task contribution and intra-task difficulty, and measure them based on validation performance.
(2) We introduce an Adaptive Task Balancing approach for visual instruction tuning (VisATB), which employs three distinct yet complementary task weighting strategies.
(3) We design a Visual Instruction Task Weighting (VITW) paradigm tailored for visual instruction tuning.
(4) Experiments on three benchmarks indicate that VisATB consistently outperforms existing approaches, achieving a more robust and balanced overall performance.

2 Method

In this section, we first introduce the Visual Instruction Task Weighting (VITW) paradigm tailored for visual instruction tuning. Building upon this paradigm, we analyze two crucial dimensions for visual task balancing: inter-task contribution and intra-task difficulty, and accordingly propose three task weighting strategies. Finally, we integrate these strategies to formulate the Adaptive Task Balancing (VisATB) approach.

2.1 Visual Instruction Task Weighting

Typically, data from various instruction-following tasks are mixed indiscriminately for visual instruction tuning. In this process, the training loss is computed as the average cross-entropy loss across all valid tokens, as expressed in the following equation:

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

where N denotes the task number, S_i indicates the sample number of Task i, T_{ij} represents the valid token number in the j-th sample for Task i, and t_{ijk} signifies the k-th valid token in the j-th sample for Task i. However, the traditional task weighting paradigm, which computes the total loss as a weighted average of individual task losses, is incompatible in this context. Differences in sequence length and sample size across tasks result in varying numbers of valid tokens for each task. Therefore, computing each task loss and averaging across all tasks introduces implicit weight to the losses of valid tokens, leading to biased learning across tasks.

To address this issue, we design a Visual Instruction Task Weighting (VITW) paradigm tailored for visual instruction tuning. The training loss of VITW is calculated as:

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

where λ_i signifies the weight of Task i. The losses of valid tokens are assigned task-specific weight and averaged at the token level, rather than at the task level, to ensure an equitable consideration for each valid token. It is a robust foundation for VisATB and has the potential to inform future research in visual instruction tuning.

2.2 Inter-Task Contribution Balancing

Although the focal points of different tasks may vary in visual instruction tuning, a central shared objective exists: enhancing the capability of understanding and reasoning about visual information. As presented in Figure 1a, the detailed image captioning

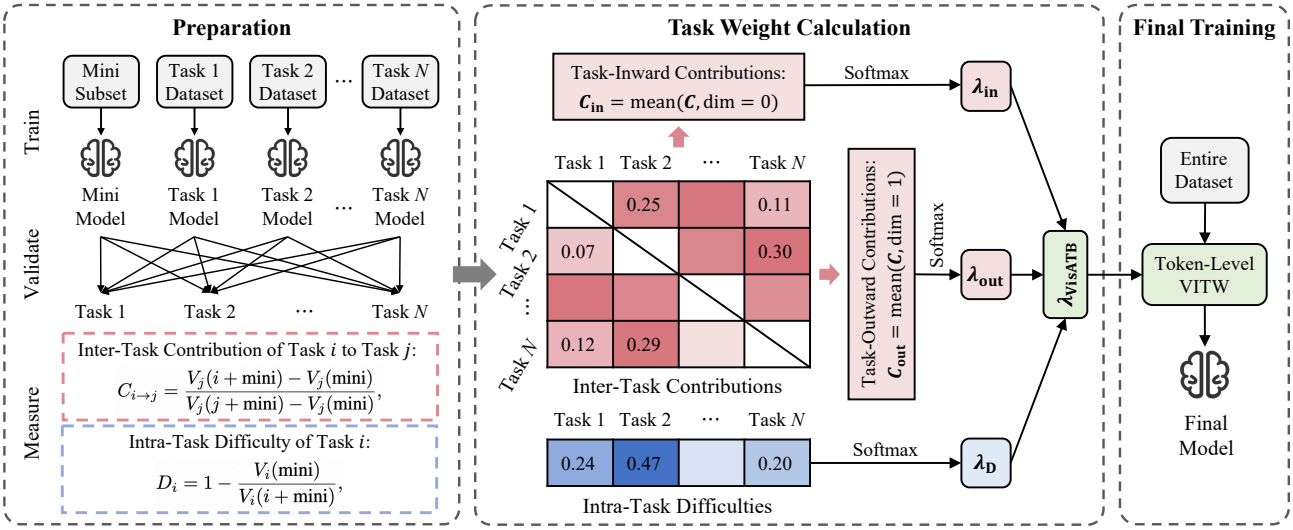


Figure 2: Overview of VisATB. In the preparation stage, we train models on the mini subset of all tasks and the dataset of each task, and validate their performance across all tasks to measure inter-task contribution and intra-task difficulty. In the task weight calculation stage, we compute three types of task weights and integrate them into the task weight λ_{VisATB}. In the final training stage, we utilize the entire dataset of all tasks and λ_{VisATB} to obtain the final model under the VITW paradigm.

data from ShareGPT4V [7] and the visual question answering data from VQAv2 [15] contain shared information for the same image, such as color, quantity, and category, exemplifying the overlapping knowledge domains among tasks. Therefore, learning one task can potentially enhance performance on others, through the mechanism we define as inter-task contribution, supported by the results in Section 3. The extent of inter-task contribution varies according to the degree of overlap in knowledge domains among tasks.

In practice, the inter-task contribution of Task i to Task j can be quantified as the normalized validation performance on Task j of a model trained on Task i, as follows:

$$C_{i\to j} = \frac{V_j(i+\text{mini}) - V_j(\text{mini})}{V_j(j+\text{mini}) - V_j(\text{mini})}, \tag{3}$$

where $V_j(i+\text{mini})$ denotes the performance on Task j of a model trained on the entire dataset or a large enough subset of Task i and mini subsets of all other tasks, and $V_j(\text{mini})$ indicates the performance on Task j of a model trained on mini subsets of all tasks. The large enough subsets are randomly sampled from the entire datasets, while the mini subsets are randomly sampled from these large enough subsets. To ensure fairness across tasks, each of these two sampling rates remains consistent across all tasks and is independent of the other. Notably, the mini subsets are added to the training data to ensure the model understands the instruction demands of all tasks. In the formula, $V_j(\text{mini})$ is subtracted from both the numerator and the denominator, which mitigates the influence of incorporating mini subsets into the training set on the validation performance on Task j.

Based on the precise quantification of the inter-task contribution, we propose two novel task weighting strategies for inter-task contribution balancing:

(1) Task-Outward Contribution Balancing: We describe the task-outward contribution, C_{out}, as the average inter-task contribution of a single task to all other tasks. This concept denotes the extent to which one task contributes to the performance of all other tasks. Tasks with higher C_{out} are more beneficial for overall training. Therefore, we propose assigning greater weight to tasks with higher C_{out} to improve collective performance across all tasks. Specifically, the task weight, $\boldsymbol{\lambda}_{\text{out}}$, for task-outward contribution balancing is computed as:

$$\boldsymbol{\lambda}_{\text{out}} = N \times \text{softmax}\left(\frac{\boldsymbol{C}_{\text{out}}}{T}\right), \text{ where } C_{\text{out},i} = \frac{\sum_{j\neq i} C_{i\to j}}{N-1}. \tag{4}$$

Here, $C_{\text{out},i}$ represents the task-outward contribution of Task i, $\boldsymbol{C}_{\text{out}}$ denotes the task-outward contribution vector of all tasks, and T is the temperature hyperparameter.

(2) Task-Inward Contribution Balancing: Conversely, we characterize the task-inward contribution, C_{in}, as the average inter-task contribution from all other tasks to a single task. This concept signifies the degree to which the performance of one task benefits from all other tasks. In comparison to tasks that benefit from more collaborative training, those with lower C_{in} are more likely to exhibit reduced performance. Therefore, we propose assigning greater weight to tasks with lower C_{in} to mitigate performance imbalance. Specifically, the task weight, $\boldsymbol{\lambda}_{\text{in}}$, for task-inward contribution balancing is calculated as:

$$\boldsymbol{\lambda}_{\text{in}} = N \times \text{softmax}\left(-\frac{\boldsymbol{C}_{\text{in}}}{T}\right), \text{ where } C_{\text{in},i} = \frac{\sum_{j\neq i} C_{j\to i}}{N-1}. \tag{5}$$

Here, $C_{\text{in},i}$ denotes the task-inward contribution of Task i, $\boldsymbol{C}_{\text{in}}$ signifies the task-inward contribution vector of all tasks, and T is the temperature hyperparameter.

2.3 Intra-Task Difficulty Balancing

Moreover, the inherent learning difficulty, termed intra-task difficulty, also varies substantially across tasks in visual instruction tuning. As presented in Figure 1b, different tasks exhibit distinct performance improvement trajectories w.r.t. increasing training

data amount. Tasks that achieve near-optimal performance with limited training data are considered to have lower intra-task difficulties, whereas tasks that require extensive training data to reach optimal performance demonstrate higher intra-task difficulties. For example, the tasks shown in Figure 1b are arranged in ascending order of intra-task difficulty as follows: visual question answering, detailed image captioning, and visual grounding.

In practice, the intra-task difficulty of Task i can be measured as the normalized validation performance gap on Task i between a model trained on a mini subset of Task i and one trained on the entire dataset or a large enough subset of the same task. Additionally, we repurpose the additional models trained for inter-task contribution balancing to reduce time costs. Consequently, the intra-task difficulty of Task i can be calculated as follows:

$$D_i = 1 - \frac{V_i(\text{mini})}{V_i(i + \text{mini})}, \tag{6}$$

where $V_i(i + \text{mini})$ indicates the performance on Task i of a model trained on the entire dataset or a large enough subset of Task i and mini subsets of all other tasks, and $V_i(\text{mini})$ represents the performance on Task i of a model trained on mini subsets of all tasks. Importantly, the impact of mini subsets from other tasks on the Task i validation performance is negligible compared to the impact of its own mini subset. Therefore, model repurposing can significantly reduce additional time costs while maintaining minimal error in the computation of intra-task difficulties.

Due to the varying degrees of intra-task difficulties across tasks, learning all tasks equally may result in underfitting on more challenging tasks, even when simpler ones are overfitted. Therefore, we propose assigning greater weight to tasks with higher D. Specifically, the task weight, $\boldsymbol{\lambda}_{\mathbf{D}}$, for intra-task difficulty balancing is computed as follows:

$$\boldsymbol{\lambda}_{\mathbf{D}} = N \times \text{softmax}(\frac{\boldsymbol{D}}{T}), \tag{7}$$

where $\boldsymbol{D}$ denotes the intra-task difficulty vector of all tasks, and T is the temperature hyperparameter.

2.4 VisATB: Adaptive Task Balancing

Finally, these three aforementioned task weighting strategies are integrated to formulate our VisATB approach. The specific task weight, $\boldsymbol{\lambda}_{\mathbf{VisATB}}$, for adaptive task balancing is computed as:

$$\boldsymbol{\lambda}_{\mathbf{VisATB}} = \alpha_{\text{out}}\boldsymbol{\lambda}_{\mathbf{out}} + \alpha_{\text{in}}\boldsymbol{\lambda}_{\mathbf{in}} + \alpha_{\text{D}}\boldsymbol{\lambda}_{\mathbf{D}}, \tag{8}$$

where α_{out}, α_{in}, and α_{D} denote the proportional coefficients, satisfying the constraint $\alpha_{\text{out}} + \alpha_{\text{in}} + \alpha_{\text{D}} = 1$.

In summary, tasks that offer substantial inter-task contributions to others and present high intra-task difficulties are assigned greater weight to achieve superior overall performance, while tasks that receive minimal inter-task contributions from others are assigned greater weight to mitigate performance imbalance.

3 Experiments

3.1 Experimental Setup

Benchmarks. We train and evaluate LMMs using three diverse multimodal benchmarks: a M^{3}IT Benchmark [25] comprising 17 training tasks and 1.2 million training samples, an Academic Benchmark [31] including 8 training tasks, and a Chat Benchmark [33] encompassing 3 training tasks. Moreover, we assess models on 7 unseen zero-shot tasks in the Academic Benchmark. A detailed description of the tasks and data preparation for these benchmarks is provided in Appendix A.

Compared Methods. We compare VisATB against the following baselines and methods: (1) Single-Task Learning (STL), where models are trained and tested on each single task; (2) Equal Weighting (EW), the most common approach, which minimizes the loss in Equation 1; (3) Task-Level Loss Aggregation (TLA), which calculates the loss within each task and then averages across all tasks; (4) Random Loss Weighting (RLW) [28]; (5) Dynamic Weight Average (DWA) [35]; and (6) Improvable Gap Balancing (IGBv1) [10]. Methods (4)-(6) are traditional task weighting methods, as described in Section 4. We leverage VITW to adapt them for visual instruction tuning. Additionally, gradient-based task weighting methods are excluded for comparison due to the substantial computational cost of aggregating gradients from large-scale model parameters.

Evaluation Metrics. We first report the common evaluation metrics for each visual task. Furthermore, we introduce two overall metrics: $\boldsymbol{\Delta I\%}$, average per-task improvement, and $\boldsymbol{\Delta E\%}$, average per-task error, in test performance on fine-tuned tasks compared to the STL baseline, which are calculated as follows:

$$\Delta I\% = \frac{1}{N}\sum_{i=1}^{N} I_i,\ \Delta E\% = \frac{1}{N}\sum_{i=1}^{N} \max(0, -I_i),$$
$$\text{where } I_i = \frac{1}{K_i}\sum_{j=1}^{K_i}(-1)^{\delta_{ij}}\frac{M_{\text{e},ij} - M_{\text{b},ij}}{M_{\text{b},ij}}. \tag{9}$$

Here, N represents the task number, K_i signifies the metric number for Task i, and I_i denotes the test performance improvement on Task i. $M_{\text{e},ij}$ and $M_{\text{b},ij}$ are the values on the j-th metric for Task i of the models trained by the evaluated method and the STL baseline. δ_{ij} is an indicator function, where $\delta_{ij} = 0$ if a higher value is better on the j-th metric for Task i, and $\delta_{ij} = 1$ otherwise. $\Delta I\%$ reflects the extent of overall performance improvement, while $\Delta E\%$ signifies the degree of performance imbalance. Moreover, to evaluate the generalizability of methods, we introduce $\boldsymbol{\Delta I_{\text{zero}}\%}$, average per-task improvement in test performance on zero-shot tasks compared to EW. The calculation of $\Delta I_{\text{zero}}\%$ is similar to that of $\Delta I\%$, except the STL baseline is replaced with the EW method.

Implementation Details. In the main experiments, we train the pretrained LLaVA-v1.5-7B model on eight NVIDIA A100 GPUs using the same settings as Liu et al. [31]. This choice is driven by LLaVA's well-established architecture and the accessibility of its pretrained models. In the M^{3}IT Benchmark, tasks are categorized into 5 groups following Li et al. [25]. We consider each task group as a whole and perform balancing at the group level. The temperature T is set as 1.0 in the M^{3}IT Benchmark, and 0.5 in the Academic Benchmark and the Chat Benchmark. The proportional coefficients of three task weighting strategies are consistently set as: $\alpha_{\text{out}} = 0.25$, $\alpha_{\text{in}} = 0.25$, and $\alpha_{\text{D}} = 0.5$.

To calculate inter-task contributions and intra-task difficulties, the entire datasets and 1/32nd mini subsets of tasks are utilized in the Academic Benchmark, which is guided by the training loss decline pattern, as detailed in Appendix B. In the Chat Benchmark, there are no specific constraints on the output format. Consequently,

Table 1: Comparative results on the M^{3}IT Benchmark. $\Delta I\%$ and $\Delta E\%$ are the average per-task improvement and error on fine-tuned tasks compared to the STL baseline.

Methods	Image Captioning			Classification						VQA			Reasoning			Generation		Overall	
	COCO	TCap	PCap	GOI	Text	INet	ITM	SVE	Moch	Shap	OCR	GQA	SQA	CLE	NL	VisD	M30k		
	CIDEr↑			EM↑						EM↑			EM↑			EM↑	GLEU↑	$\Delta I\%$↑	$\Delta E\%$↓
STL	1.024	0.751	0.179	90.7	100	98.0	98.9	81.2	84.6	63.7	56.5	48.2	73.4	57.4	72.4	42.8	0.358		
EW	**1.022**	0.747	0.250	91.2	100	**98.0**	99.3	81.8	85.6	66.0	54.9	52.4	79.6	59.9	71.1	42.5	**0.351**	3.51	0.49
TLA	1.001	0.739	0.219	**92.1**	100	97.7	99.2	81.6	85.6	65.3	**55.5**	49.5	76.5	60.4	69.0	42.2	0.342	1.43	0.97
RLW	1.016	0.748	0.245	90.8	100	97.4	99.2	81.8	79.8	64.5	54.5	50.3	76.4	55.1	69.2	41.6	0.340	1.21	1.59
DWA	1.021	**0.757**	0.243	89.8	100	97.5	99.3	81.5	79.0	67.5	**55.5**	**53.4**	80.6	59.9	70.9	42.4	0.351	3.08	0.91
IGBv1	1.006	0.727	0.227	89.7	100	97.1	99.1	76.1	76.4	63.9	50.6	45.4	72.2	52.8	63.0	40.8	0.343	-2.55	4.16
VisATB	1.019	0.756	**0.258**	91.7	100	97.9	**99.4**	**82.2**	**86.1**	**68.0**	55.1	52.7	**81.1**	**61.8**	**71.6**	**42.7**	**0.351**	**4.52**	**0.39**

Table 2: The task weights calculated in VisATB on the M^{3}IT Benchmark.

Task Weights	Cap.	Cls.	VQA	Reas.	Gen.
$\boldsymbol{\lambda}_{\mathbf{out}}$	0.9072	0.9978	1.1010	0.9927	1.0013
$\boldsymbol{\lambda}_{\mathbf{in}}$	0.9626	0.7375	1.2822	1.0286	0.9891
$\boldsymbol{\lambda}_{\mathbf{D}}$	1.1831	0.9421	0.9383	0.9967	0.9399
$\boldsymbol{\lambda}_{\mathbf{VisATB}}$	1.0590	0.9048	1.0649	1.0037	0.9675

only the entire datasets of tasks are required, without the need for mini subsets, simplifying the form of VisATB, as detailed in Appendix C. In the M^{3}IT Benchmark, the training data size of each task group is substantial; thus, we use 1/4th (large enough) subsets and 1/32nd mini subsets of tasks. In practice, guided by experimental analysis in Appendix G, we suggest that subsets containing more than 10k samples and trained for over 100 steps are sufficiently large to ensure effective training of VisATB.

3.2 Evaluation on the M^{3}IT Benchmark

Effectiveness of VisATB. The comparative results of the fine-tuned tasks on the M^{3}IT Benchmark are presented in Table 1. Utilizing the same pretrained model and training data, VisATB achieves the optimal performance in both $\Delta I\%$ and $\Delta E\%$. Specifically, VisATB attains the best performance on 11 out of 17 tasks and exhibits near-optimal performance on the remaining tasks, which demonstrates the effectiveness of VisATB in both improving overall performance and mitigating performance imbalance across diverse visual tasks. Notably, in the reasoning group, VisATB substantially outperforms all baselines, such as 81.1 on SQA and 61.8 on CLE, highlighting its potential strength in complex reasoning tasks.

Additionally, the specific task weights calculated in VisATB are presented in Table 2. In the classification task group, although the tasks are assigned relatively small weights (0.9048), VisATB still demonstrates strong performance, attaining the optimal results on 4 out of 6 tasks and ranking second-best on the remaining tasks. This counterintuitive finding suggests an important insight: for certain relatively simple tasks, directly increasing their weight may not lead to further performance improvements due to potential overfitting or saturation effects. Instead, our approach of prioritizing tasks that contribute more significantly to these simple ones (as reflected in high task-outward contributions) has the potential to transcend individual task performance limits by facilitating the acquisition of new, transferable knowledge across the task space.

Validity of VITW. We compare TLA with EW to systematically evaluate the validity of our VITW paradigm in visual instruction tuning. The results provide compelling evidence for the necessity of token-level loss aggregation. Specifically, TLA performs substantially worse than EW in both overall metrics: it achieves only 1.43% in $\Delta I\%$ compared to 3.51% of EW, and exhibits a more pronounced performance imbalance with a $\Delta E\%$ of 0.97% versus 0.49% of EW. As discussed in Section 2.1, this performance degradation stems from the implicit weight bias introduced by TLA, which is inversely proportional to the number of valid tokens in each task. For example, the image captioning task group, which naturally involves longer textual descriptions and thus a larger number of valid tokens per sample, receives lower implicit weight in the optimization process, thereby leading to the poorest performance. These findings underscore the validity of our VITW paradigm, which ensures equitable treatment of all valid tokens by computing weighted losses at the token level before aggregation.

Limitation of Traditional Task Weighting Methods. Directly applying traditional task weighting methods, including RLW, DWA, and IGBv1, in visual instruction tuning yields markedly inferior performance. These results underscore a fundamental limitation of traditional methods, which balance tasks solely based on training losses, when applied to visual instruction tuning: training losses fail to serve as reliable indicators of actual task learning progress and generalization capacity in LMMs. In contrast, the validation performance-based measurement in VisATB provides a more accurate reflection of the model's actual capabilities and learning trajectories across tasks, leading to superior task weight determination and significant performance improvements.

Quantitative Analysis of the Time Cost of VisATB. An important practical consideration for any task balancing method is its computational overhead. We provide a comprehensive analysis of the additional training time required by VisATB. The extra training time of VisATB is approximately $(R_{\text{large}} + N \times R_{\text{mini}})$ times the training duration of the final model, where N denotes the task number, R_{large} and R_{mini} represent the sampling rates of large enough subsets and mini subsets. In the M^{3}IT Benchmark, with $N = 5$, $R_{\text{large}} = 1/4$, and $R_{\text{mini}} = 1/32$, the additional training time is about $0.25 + 5 \times 0.03 = 0.40$ times that of the final model.

Table 3: Comparative results of the fine-tuned tasks on the Academic Benchmark. $\Delta I\%$ and $\Delta E\%$ are the average per-task improvement and error on fine-tuned tasks compared to the STL baseline.

Methods	ShareGPT4V CIDEr↑	Ref-caption CIDEr↑	VQAv2 EM↑	GQA EM↑	ChartQA EM↑	OCRVQA EM↑	Ref-bbox IoU↑	**Overall** $\Delta I\%$↑	**Overall** $\Delta E\%$↓
STL	0.1285	0.4658	77.73	61.23	17.76	68.22	51.58		
EW	0.1411	0.5591	78.27	**62.20**	19.60	67.73	61.63	8.75	**0.10**
TLA	0.1144	**0.5770**	77.72	60.42	**22.36**	67.80	56.58	6.65	1.85
RLW	0.1388	0.5571	77.28	60.61	18.20	66.73	55.86	4.95	0.54
DWA	0.1225	0.5470	**78.28**	61.82	19.88	**67.87**	61.12	6.34	0.74
IGBv1	0.1349	0.4824	77.00	60.92	17.20	65.96	55.47	0.17	1.13
VisATB	**0.1437**	0.5724	77.99	61.81	20.16	67.48	**67.38**	**11.29**	0.15

Table 4: Comparative results of the zero-shot tasks on the Academic Benchmark. $\Delta I_{\text{zero}}\%$ is the average per-task improvement on zero-shot tasks compared to the EW method.

Methods	TextVQA↑	POPE↑	MME↑	SQA↑	MMBench↑	SEED^{I}↑	MM-Vet↑	**Overall ($\Delta I_{\text{zero}}\%$↑)**
EW	53.96	86.83	**1524.41**	60.74	48.54	54.13	29.00	0.00
TLA	50.74	**86.94**	1426.51	55.86	**52.92**	46.67	28.70	-3.73
RLW	52.74	86.29	1504.06	59.99	52.58	56.54	28.70	0.90
DWA	**54.05**	86.76	1484.26	60.53	49.31	54.75	28.90	-0.07
IGBv1	52.09	86.86	1497.02	57.79	39.95	47.26	**29.30**	-5.63
VisATB	53.87	86.73	1501.86	**61.07**	51.46	**58.05**	**29.30**	**1.87**

However, this overhead exhibits favorable scaling properties:

(1) *Sublinear scaling with data size*: As the total dataset size increases, both R_{large} and R_{mini} can be proportionally reduced while maintaining sufficient samples for reliable measurement (as discussed in Section G, subsets with >10k samples trained for >100 steps are generally sufficient). This ensures that the absolute additional cost does not increase significantly with dataset scale.

(2) *Task grouping strategy*: In scenarios with a large number of tasks, e.g., the M^3IT Benchmark with 17 tasks, VisATB can leverage task clustering to perform balancing at the group level. By clustering semantically or structurally similar tasks (e.g., all classification tasks into one group), we can reduce N from 17 to 5, substantially decreasing the $(N \times R_{\text{mini}})$ component of the overhead. It can be performed using expert knowledge about task similarities or automated clustering approaches based on task characteristics.

Consequently, VisATB can effectively function across diverse scales and scenarios without substantially increasing complexity or incurring prohibitive time costs, making it a practical solution for real-world visual instruction tuning applications.

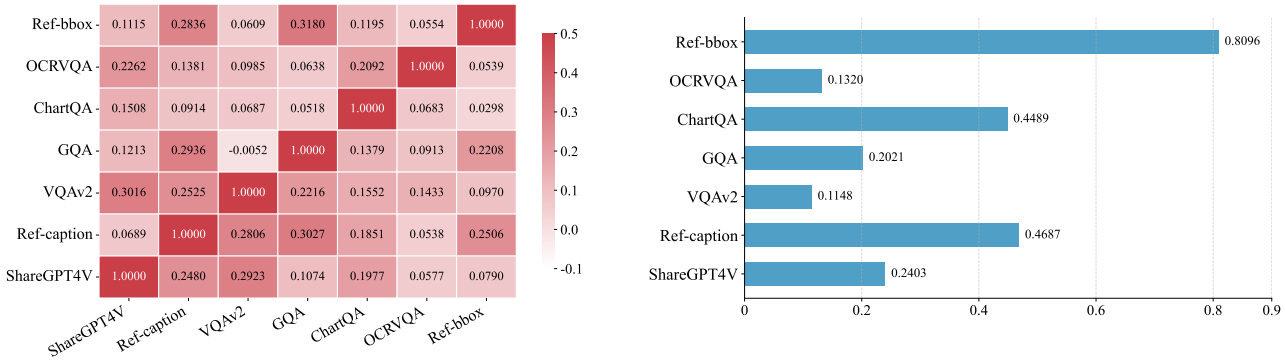
3.3 Evaluation on the Academic Benchmark

The comparative results of the fine-tuned tasks on the Academic Benchmark are summarized in Table 3. Overall, VisATB achieves the highest $\Delta I\%$ while maintaining a near-minimal $\Delta E\%$, indicating both enhanced overall performance and mitigated task imbalance. Compared with EW, VisATB delivers substantial gains on Ref-bbox, Ref-caption, ChartQA, and ShareGPT4V, while preserving competitive results on the remaining tasks. These results further demonstrate the effectiveness of VisATB. Additionally, TLA and all traditional task weighting methods perform worse than EW and VisATB in both $\Delta I\%$ and $\Delta E\%$, further underscoring the validity of VITW and the limitation of traditional methods.

Generalization on Zero-Shot Tasks. The comparative results of the zero-shot tasks on the Academic Benchmark are presented in Table 4, including single-word QA (TextVQA, POPE, MME), multiple-choice QA (SQA, MMBench, SEED^{I}), and open-ended QA (MM-Vet). Overall, VisATB achieves the optimal $\Delta I_{\text{zero}}\%$. Compared to EW, VisATB exhibits significantly superior performance on multi-choice QA tasks while maintaining comparable results on other tasks. Importantly, the training corpus contains no visual multiple-choice QA-type data at all. Nevertheless, VisATB substantially enhances the model's zero-shot generalization to such tasks, demonstrating its ability to induce transferable capabilities beyond the supervised domains. Additionally, TLA demonstrates markedly lower values than EW in $\Delta I_{\text{zero}}\%$, further supporting the validity of VITW.

Visual Analysis of VisATB. The inter-task contributions and intra-task difficulties of VisATB are visualized in Figure 3, and the resulting task weights are detailed in Table 5. As shown in Figure 3a, the heatmap of inter-task contributions demonstrates substantial heterogeneity across task pairs: certain tasks (e.g., Ref-caption and VQAv2) provide strong positive contributions to others, while some (e.g., OCRVQA) receive minimal benefit from others. Meanwhile, Figure 3b illustrates that intra-task difficulties vary significantly: tasks like grounding and captioning are notably harder to learn compared to simpler tasks such as basic visual QA. This highlights the necessity of considering both inter-task contributions and intra-task difficulties for effective task balancing, and the effectiveness of VisATB to capture these differences across tasks.

Model Independence of VisATB. To verify the general effectiveness of our approach independent of specific models, we further



(a) Heatmap of inter-task contributions. **(b) Histogram of intra-task difficulties.**

Figure 3: Visualizations of inter-task contributions and intra-task difficulties calculated in VisATB on the Academic Benchmark.

Table 5: The task weights calculated in VisATB on the Academic Benchmark.

Task Weights	ShareGPT4V	Ref-caption	VQAv2	GQA	ChartQA	OCRVQA	Ref-bbox
$\boldsymbol{\lambda_{\text{out}}}$	1.0223	1.0782	1.0888	0.9815	0.8592	0.9588	1.0112
$\boldsymbol{\lambda_{\text{in}}}$	0.9727	0.8722	1.0345	0.9456	0.9649	1.1532	1.0569
$\boldsymbol{\lambda_{\text{D}}}$	0.7195	1.1361	0.5598	0.6666	1.0920	0.5794	2.2466
$\boldsymbol{\lambda_{\text{VisATB}}}$	0.8585	1.0557	0.8107	0.8151	1.0020	0.8177	1.6403

Table 6: Comparative results on the Academic Benchmark using various pretrained models.

Models	Methods	$\Delta I\%\uparrow$	$\Delta E\%\downarrow$	$\Delta I_{\text{zero}}\%\uparrow$
LLaVA-v1.5-13B	EW	4.55	0.66	0.00
	VisATB	**6.89**	**0.29**	**0.38**
Qwen2-VL-2B	EW	-2.68	4.08	0.00
	VisATB	**-1.32**	**3.70**	**1.00**

fine-tune the pretrained LLaVA-v1.5-13B [31] and Qwen2-VL-2B [50] models on the Academic Benchmark. The overall results are presented in Table 6, with detailed results and settings in Appendix D. Across both scales and architectures, VisATB consistently improves $\Delta I\%$ and $\Delta I_{\text{zero}}\%$ while reducing $\Delta E\%$ compared to EW, demonstrating its robustness to backbone variation.

3.4 Evaluation on the Chat Benchmark

The comparative results of the fine-tuned tasks on the Chat Benchmark are presented in Table 7, which includes three loosely structured tasks: general conversation (Conv.), detailed description (Detail.), and complex reasoning (Complex.). This setting reduces overt format conflicts across tasks and serves to assess whether VisATB still offers benefits when structural incompatibilities are minimal. Overall, VisATB outperforms all methods except TLA in $\Delta I\%$, while maintaining the lowest value in $\Delta E\%$. Compared to EW, VisATB yields substantial gains on Conv. (+7.4 absolute, 52.2 vs. 44.8) and performs slightly better on Detail. (60.7 vs. 59.6), while sustaining the state-of-the-art Complex. performance (83.4). These results indicate that in a chat-style setting, VisATB can still enhance underfitting tasks without compromising the performance of well-performing tasks.

Table 7: Comparative results on the Chat Benchmark. $\Delta I\%$ and $\Delta E\%$ are the average per-task improvement and error on fine-tuned tasks compared to the STL baseline.

Methods	Conv.↑	Detail.↑	Complex.↑	Overall $\Delta I\%\uparrow$	Overall $\Delta E\%\downarrow$
STL	43.6	49.3	80.9		
EW	44.8	59.6	**83.4**	8.91	**0.00**
TLA	**57.2**	61.9	79.2	**18.22**	0.70
RLW	46.0	56.7	82.9	7.66	**0.00**
DWA	46.1	**65.9**	82.8	13.92	**0.00**
IGBv1	39.9	64.2	81.7	7.58	2.83
VisATB	52.2	60.7	**83.4**	15.31	**0.00**

Analysis of TLA and Traditional Methods. TLA achieves the best $\Delta I\%$ but at the cost of an increase in $\Delta E\%$ (0.70 vs. 0.00 of EW). Because the implicit weight introduced by TLA is unregulated, it is unstable across various scenarios and prone to performance imbalance. Additionally, traditional task weighting methods exhibit inferior performance in $\Delta I\%$, except for DWA, which outperforms EW but still falls short of VisATB. These findings further indicate the necessity of our VITW paradigm and the insufficiency of traditional methods for visual instruction tuning.

3.5 Ablation Studies

As presented in Table 8 (with full results provided in Appendix E), we ablate our VisATB on the Academic Benchmark from three perspectives: task weighting strategies, temperatures, and calculation approaches for the intra-task difficulty. The compared methods include: EW; VisATB ($\boldsymbol{\alpha} = [\alpha_{\text{out}}, \alpha_{\text{in}}, \alpha_{\text{D}}]$), where the proportional coefficients for the three task weighting strategies are set to different values; VisATB ($T = 2.0/1.0/0.5$), where the temperature T is set

Table 8: Ablation results on the Academic Benchmark. $\boldsymbol{\alpha} = [\alpha_{\text{out}}, \alpha_{\text{in}}, \alpha_{\text{D}}]$ is the vector of proportional coefficients for the three task weighting strategies; T is the temperature hyperparameter; and 'precise/real Diff' denotes the use of the precise or real calculation approach for intra-task difficulty.

Methods	$\Delta I\%\uparrow$	$\Delta E\%\downarrow$	$\Delta I_{\text{zero}}\%\uparrow$
EW	8.75	0.10	0.00
VisATB ($\boldsymbol{\alpha} = [1, 0, 0]$)	9.67	0.10	**1.36**
VisATB ($\boldsymbol{\alpha} = [0, 1, 0]$)	8.45	**0.05**	-3.96
VisATB ($\boldsymbol{\alpha} = [0, 0, 1]$)	**12.21**	0.30	0.63
VisATB ($\boldsymbol{\alpha} = [0.50, 0.50, 0]$)	8.58	**0.06**	2.32
VisATB ($\boldsymbol{\alpha} = [0.33, 0.33, 0.33]$)	8.77	0.32	**3.04**
VisATB ($\boldsymbol{\alpha} = [0.25, 0.25, 0.50]$)	**11.29**	0.15	1.87
VisATB ($T = 2.0$)	9.92	0.12	0.50
VisATB ($T = 1.0$)	10.24	**0.11**	0.52
VisATB ($T = 0.5$)	**11.29**	0.15	**1.87**
VisATB (precise Diff)	10.75	0.16	**2.61**
VisATB (real Diff)	**11.29**	**0.15**	1.87

as 2.0, 1.0 or 0.5; and VisATB (precise/real Diff), where the precise or real calculation approach for intra-task difficulty is used. The precise calculation approach trains the additional models to precisely quantify the intra-task difficulty, as detailed in Appendix F, while the real calculation approach repurposes the models trained for inter-task contribution balancing to reduce the time cost.

Task Weighting Strategies. We first isolate each strategy component of VisATB. As mentioned in Section 2.4, $\boldsymbol{\lambda}_{\text{out}}$ and $\boldsymbol{\lambda}_{\text{D}}$ focus on improving overall performance, while $\boldsymbol{\lambda}_{\text{in}}$ aims to mitigate performance imbalance. Compared to EW, activating only the $\boldsymbol{\lambda}_{\text{out}}$ weight (VisATB ($\boldsymbol{\alpha} = [1, 0, 0]$)) raises $\Delta I\%$ from 8.75 to 9.67 and improves $\Delta I_{\text{zero}}\%$ to 1.36, while keeping $\Delta E\%$ unchanged at 0.10. Utilizing only the $\boldsymbol{\lambda}_{\text{in}}$ weight (VisATB ($\boldsymbol{\alpha} = [0, 1, 0]$)) effectively reduces $\Delta E\%$ to 0.05, albeit with a slight drop in $\Delta I\%$ and $\Delta I_{\text{zero}}\%$. Relying solely on the $\boldsymbol{\lambda}_{\text{D}}$ weight (VisATB ($\boldsymbol{\alpha} = [0, 0, 1]$)) produces the largest boost in $\Delta I\% = 12.21$ and a moderate improvement in $\Delta I_{\text{zero}}\%$ but at the cost of a slight increase in $\Delta E\%$. These underscore the effectiveness of all three task weighting strategies, each focusing on distinct yet complementary aspects.

When combining these task weighting strategies, the proportional coefficients can be adjusted to achieve more favorable Pareto trade-offs. Specifically, VisATB ($\boldsymbol{\alpha} = [0.50, 0.50, 0]$) integrates two inter-task contribution balancing strategies, resulting in a balance between $\Delta I\%$ and $\Delta E\%$, while also enhancing $\Delta I_{\text{zero}}\%$. Moreover, VisATB ($\boldsymbol{\alpha} = [0.33, 0.33, 0.33]$) combines all three strategies, leading to improvements in $\Delta I\%$ and $\Delta I_{\text{zero}}\%$. To further enhance overall performance, VisATB ($\boldsymbol{\alpha} = [0.25, 0.25, 0.50]$) slightly increases the value of α_{D}, achieving significantly higher $\Delta I\%$ and $\Delta I_{\text{zero}}\%$ than EW, while also maintaining nearly the lowest $\Delta E\%$. Notably, we observe that combining strategies generally leads to superior performance in $\Delta I_{\text{zero}}\%$ than employing any single strategy. This finding demonstrates the effectiveness of comprehensively considering multiple aspects of task balancing for generalization to unseen zero-shot tasks.

Temperatures. VisATB increasingly outperforms EW in both $\Delta I\%$ and $\Delta I_{\text{zero}}\%$ as T decreases, while exhibiting a slightly higher $\Delta E\%$ at lower values of T. As the temperature in the softmax functions of Equations 4, 5, and 7 decreases, the weight distribution becomes progressively sharper. If the sharpness in task weight is excessively high, tasks with too small weights may inevitably underperform, leading to a slight performance imbalance. In practice, we recommend using the lowest temperature T that ensures all task weights remain within the range of 0.5 to 2.0 to avoid over-balancing.

Calculation Approaches for Intra-Task Difficulty. As discussed in Section 2.3, the objective of the real approach is to reduce the time cost with minimal error. VisATB (precise Diff) and VisATB (real Diff) exhibit comparable performance levels, with VisATB (real Diff) even showing a slight advantage in $\Delta I\%$ and $\Delta E\%$. Meanwhile, the real calculation approach enables a reduction of approximately $(R_{\text{large}} + R_{\text{mini}})$ times the training duration of the final model, where R_{large} and R_{mini} represent the sampling rates of large enough subsets and mini subsets, respectively. This observation underscores the efficacy of our real calculation approach.

4 Related Work

Visual Instruction Tuning. Instruction tuning is first proposed in NLP, enabling LLMs to follow textual instructions and accomplish various tasks [51]. Moreover, to extend the powerful abilities of LLMs into the multimodal domain, Liu et al. [33] introduce visual instruction tuning. This innovative technique integrates LLMs with visual encoders using visual instruction-following data and alignment modules. Subsequently, a series of improved approaches demonstrate robust performance on visual tasks, respectively focusing on model structures [5, 8, 9, 14, 27, 44, 59], training settings [31, 53], and training data [7, 24, 49, 56, 57]. Particularly, to mitigate visual task conflicts, Dai et al. [9] adaptively adjust sampling probabilities based on task data sizes, and several recent studies design the mixture of LoRA experts structure [8, 14, 44]. In this paper, we propose tackling this challenge from an alternative perspective by adaptive task weighting.

Task Weighting. Adaptive task weighting is commonly employed in CV, which assigns task weights based on losses or gradients to balance the joint training process of tasks [3, 6, 20, 30, 34, 41, 43]. For example, Lin et al. [28] assign task weight randomly; Liu et al. [35] prefer tasks with lower loss decline rates; and Dai et al. [10] favor tasks with higher improvable gaps.

5 Conclusion

In this paper, we introduce an Adaptive Task Balancing approach for visual instruction tuning (VisATB). Specifically, we design a token-level Visual Instruction Task Weighting (VITW) paradigm. Building upon this paradigm, we analyze two crucial dimensions for visual task balancing: inter-task contribution and intra-task difficulty. Accordingly, we propose three distinct yet complementary task weighting strategies. Extensive experiments demonstrate that VisATB outperforms existing methods, achieving a more robust and balanced overall performance.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (62376274, 62437002).

References

[1] 2023. ShareGPT. https://sharegpt.com. Accessed: 2024-8-01.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Idan Achituve, Idit Diamant, Arnon Netzer, Gal Chechik, and Ethan Fetaya. 2024. Bayesian uncertainty for gradient aggregation in multi-task learning. *arXiv preprint arXiv:2402.04005* (2024).

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).

[6] Hao Ban and Kaiyi Ji. 2024. Fair resource allocation in multi-task learning. *arXiv preprint arXiv:2402.15638* (2024).

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*. Springer, 370–387.

[8] Shaoxiang Chen, Zequn Jie, and Lin Ma. 2024. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160* (2024).

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. *arXiv preprint arXiv:2305.06500* (2023).

[10] Yanqi Dai, Nanyi Fei, and Zhiwu Lu. 2023. Improvable Gap Balancing for Multi-Task Learning. In *Uncertainty in Artificial Intelligence*. PMLR, 496–506.

[11] 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*. 326–335.

[12] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459* (2016).

[13] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv preprint arXiv:2306.13394* (2023).

[14] 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* (2023).

[15] 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*. 6904–6913.

[16] 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*. 6700–6709.

[17] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2901–2910.

[18] Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-vil: A dataset and benchmark for natural language explanations in vision-language tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1244–1254.

[19] 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*. 787–798.

[20] 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*. 7482–7491.

[21] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 317–325.

[22] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125* (2023).

[23] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326* (2024).

[24] 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. *Advances in Neural Information Processing Systems* 36 (2023), 28541–28564.

[25] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. 2023. M^3IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning. *arXiv preprint arXiv:2306.04387* (2023).

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

[27] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, et al. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* (2024).

[28] 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* (2021).

[29] 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 *Computer Vision–ECCV 2014: 13th European Vonference, Zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 740–755.

[30] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-Averse Gradient Descent for Multi-task Learning. *Advances in Neural Information Processing Systems* 34 (2021), 18878–18890.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26296–26306.

[32] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/. Accessed: 2024-8-01.

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual Instruction Tuning. *Advances in Neural Information Processing Systems* 36 (2024).

[34] 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*.

[35] 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*. 1871–1880.

[36] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2024. Mmbench: Is your multi-modal model an all-around player?. In *European conference on computer vision*. Springer, 216–233.

[37] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. *Advances in Neural Information Processing Systems* 35 (2022), 2507–2521.

[38] 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*. 11–20.

[39] 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* (2022).

[40] 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*. IEEE, 947–952.

[41] 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*. PMLR, 16428–16446.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (2015), 211–252.

[43] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. *Advances in Neural Information Processing Systems* 31 (2018).

[44] Leyang Shen, Gongwei Chen, Rui Shao, Weili Guan, and Liqiang Nie. 2024. Mome: Mixture of multimodal experts for generalist multimodal large language models. *arXiv preprint arXiv:2407.12709* (2024).

[45] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 742–758.

[46] 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*. 8317–8326.

[47] 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)*. 217–223.

Table 9: Training data sizes and response format instructions for the fine-tuned tasks on the Academic Benchmark. The total training data size is 475k.

Tasks	Sizes	Response Format Instructions
ShareGPT	41k	–
ShareGPT4V	98k	
Ref-caption	41k	Provide a short description for this region.
VQAv2	83k	Answer the question using a single word or phrase.
GQA	72k	
ChartQA	18k	
OCRVQA	80k	
Ref-bbox	41k	Provide the bounding box coordinate of the region this sentence describes.

Table 10: Response format instructions for the zero-shot test tasks on the Academic Benchmark.

Tasks	Response Format Instructions
TextVQA	Answer the question using a single word or phrase.
POPE	
MME	
SQA^I	Answer with the option's letter from the given choices directly.
MMBench	
$SEED^I$	
MM-Vet	–
LLaVA-Bench	

[48] 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* (2016).

[49] Bin Wang, Fan Wu, Xiao Han, Jiahui Peng, Huaping Zhong, Pan Zhang, Xiaoyi Dong, Weijia Li, Wei Li, Jiaqi Wang, et al. 2024. Vigc: Visual instruction generation and correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 5309–5317.

[50] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191* (2024).

[51] 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* (2021).

[52] Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. 2023. End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2733–2743.

[53] 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* (2023).

[54] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review* 11, 12 (2024), nwae403.

[55] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490* (2023).

[56] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107* (2023).

[57] Bo Zhao, Boya Wu, Muyang He, and Tiejun Huang. 2023. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087* (2023).

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems* 36 (2024).

[59] 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* (2023).

A Task Information and Data Preparation

We train and evaluate LMMs on the following benchmarks:

M^3IT Benchmark. The tasks in the M^3IT Benchmark are carefully selected from the M^3IT dataset [25], a large-scale multimodal instruction tuning dataset. All curated tasks have training and validation sets. If no test set is provided, the validation set is randomly divided into two equal parts, one for validation and the other for testing. Following the task clustering of Li et al. [25], the tasks are categorized into five distinct groups. The task grouping and task list in the group are shown as follows:

(1) Image Captioning: MS-COCO (COCO) [29], TextCaps (TCap) [45], and Image-Paragraph-Captioning (PCap) [21].
(2) Classification: COCO-GOI (GOI) [29], COCO-Text (Text) [48], ImageNet (INet) [42], COCO-ITM (ITM) [29], e-SNLI-VE (SVE) [18], and Mocheg (Moch) [52].
(3) Visual Question Answering (VQA): Shapes VQA (Shap) [4], OCRVQA (OCR) [40], and GQA [16]
(4) Reasoning: ScienceQA (SQA) [37], CLEVR (CLE) [17], and NLVR (NL) [47].
(5) Generation: Visual Dialog (VisD) [11], and Multi30k (M30k) [12].

Chat Benchmark. The tasks in the Chat Benchmark are introduced by Liu et al. [33], including general conversation (Conv.), detailed description (Detail.), and complex reasoning (Complex.). We utilize LLaVA-Bench-COCO for validation and $LLaVA^W$ for testing.

Academic Benchmark. The tasks in the Academic Benchmark encompass ShareGPT [1], ShareGPT4V [7], Ref-caption, Ref-bbox [19, 38], VQAv2 [15], GQA [16], ChartQA [39], and OCRVQA [40]. Among these, the Ref-caption task involves generating captions for image regions defined by bounding boxes, while the Ref-bbox task aims to predict the bounding boxes corresponding to the described image regions. The testB set of Kazemzadeh et al. [19], the test-dev set of Goyal et al. [15], the test-dev-balanced set of Hudson and Manning [16], and the test sets of other tasks are used for testing their corresponding tasks. The weight of ShareGPT is set as 1.0.

Furthermore, we present 7 zero-shot metrics employed by Liu et al. [31] to evaluate the generalization of methods: TextVQA [46], POPE [26], MME [13], ScienceQA (SQA) [37], MMBench [36], SEED-Bench-IMG ($SEED^I$) [22], and MM-Vet [55].

The training data sizes and response format instructions for fine-tuned tasks are depicted in Table 9, while the response format instructions for zero-shot test tasks are presented in Table 10. Moreover, we employ multiple data processing and splitting strategies to reduce the computational cost and ensure the evaluation reliability:

(1) For ShareGPT4V, data is randomly partitioned, with 2k allocated to a validation set, 2k to a test set, and the remainder reserved for training.
(2) For all VQA tasks and RefCOCO, training data from the same image are merged into a single conversation.
(3) For RefCOCO, training conversations are divided into segments, each containing fewer than 10 turns.

Table 11: Complete results of the fine-tuned tasks on the Academic Benchmark using various pretrained models. $\Delta I\%$ and $\Delta E\%$ are the average per-task improvement and error on fine-tuned tasks compared to the STL baseline.

Models	Methods	ShareGPT4V CIDEr↑	Ref-caption CIDEr↑	VQAv2 EM↑	GQA EM↑	ChartQA EM↑	OCRVQA EM↑	Ref-bbox IoU↑	Overall $\Delta I\%$↑	Overall $\Delta E\%$↓
LLaVA-v1.5-13B	STL	0.1407	0.5329	78.78	62.54	18.96	69.86	65.50		
	EW	0.1351	0.6008	**79.51**	**63.05**	22.28	**69.41**	68.44	4.55	0.66
	VisATB	**0.1391**	**0.6224**	79.33	63.00	**22.76**	69.22	**73.35**	**6.89**	**0.29**
Qwen2-VL-2B	STL	0.1370	0.8402	81.11	64.55	64.36	73.40	27.89		
	EW	0.1305	**0.6658**	80.22	**64.41**	63.36	**73.28**	**30.62**	-2.68	4.08
	VisATB	**0.1480**	0.6456	**80.24**	64.23	**64.04**	72.93	30.46	**-1.23**	**3.70**

Table 12: Complete results of the zero-shot tasks on the Academic Benchmark using various pretrained models. $\Delta I_{\text{zero}}\%$ is the average per-task improvement in test performance on zero-shot tasks compared to the EW method.

Models	Methods	TextVQA↑	POPE↑	MME↑	SQA↑	MMBench↑	SEED^{I}↑	MM-Vet↑	Overall ($\Delta I_{\text{zero}}\%$↑)
LLaVA-v1.5-13B	EW	**57.52**	**86.50**	1556.48	64.18	**58.25**	64.56	35.40	0.00
	VisATB	57.06	86.47	**1567.57**	**65.90**	57.65	**64.58**	**35.80**	**0.38**
Qwen2-VL-2B	EW	**73.45**	**88.41**	1438.46	52.49	**70.45**	**74.08**	**35.70**	0.00
	VisATB	72.74	**88.41**	**1445.53**	**64.35**	69.85	73.69	30.80	**1.00**

Figure 4: Training loss of EW in the Academic Benchmark.

(4) For OCRVQA, 80k conversations are randomly sampled from the training set.
(5) For VQAv2, GQA and OCRVQA, 20k data are randomly sampled from the validation set.
(6) For ShareGPT, invalid conversations are filtered out following Zheng et al. [58], and conversations that surpass 2048 tokens are truncated.

B The Sampling Rate for Mini Subsets

As discussed in Section 2.2, mini subsets enable the model to understand the instruction demands of all tasks. Due to the remarkable few-shot learning capabilities of large models, incorporating a small amount of training data can significantly enhance the model's adherence to the task instructions. In our experiments on the Academic Benchmark, the mini subset from each task is obtained by randomly sampling 1/32nd of the entire dataset from that task. As shown in Figure 4, this sampling ratio is informed by the decline pattern of training loss, which exhibits a rapid decrease during the initial 100+ steps, followed by a gradual reduction until reaching the final 3,700+ steps. This pattern suggests that the model effectively grasps the instruction demands in the initial phase, with the subsequent phase dedicated mainly to acquiring the knowledge embedded within the data. Practically, we recommend ensuring that the mini subset from each task contains at least 1,000 data points, and the total number of training steps for all mini subsets exceeds 100.

C The Simpler Form of VisATB in the Chat Benchmark

In the Chat Benchmark, there are no specific constraints on the output format. Consequently, only the entire datasets of tasks are required, without the need for mini subsets, simplifying the form of VisATB. Specifically, the inter-task contribution of Task i to Task j can be calculated as:

$$C_{i\to j} = \frac{V_j(i) - V_j(\text{base})}{V_j(j) - V_j(\text{base})}, \tag{10}$$

where $V_j(i)$ represents the validation performance on Task j of a model trained on Task i, $V_j(j)$ denotes the validation performance on Task j of a model trained on Task j itself, and $V_j(\text{base})$ signifies the validation performance on Task j of a pretrained base model. Additionally, the intra-task difficulty of Task i can be computed as:

$$D_i = 1 - \frac{V_i(\text{mini}_i)}{V_i(i)}, \tag{11}$$

where $V_i(\text{mini}_i)$ denotes the validation performance on Task i of a model trained on the mini subset of Task i, and $V_i(i)$ signifies the validation performance on Task i of a model trained on Task i itself.

D Detailed Results and Settings on Various Pretrained Models

The detailed results on the Academic Benchmark using various pretrained models are shown in Tables 11 and 12. The temperature T is set as 0.5 on LLaVA-v1.5-13B and 1.0 on Qwen2-VL-2B.

Table 13: Detailed results of ablation studies for the fine-tuned tasks on the Academic Benchmark.

Methods	ShareGPT4V CIDEr↑	Ref-caption CIDEr↑	VQAv2 EM↑	GQA EM↑	ChartQA EM↑	OCRVQA EM↑	Ref-bbox IoU↑	**Overall** $\Delta I\%$↑	$\Delta E\%$↓
EW	0.1411	0.5591	78.27	62.20	19.60	67.73	61.63	8.75	0.10
VisATB ($\boldsymbol{\alpha}=[1,0,0]$)	0.1448	0.5763	78.34	62.16	19.52	67.72	61.81	9.67	0.10
VisATB ($\boldsymbol{\alpha}=[0,1,0]$)	0.1333	0.5520	78.25	62.12	20.20	68.00	62.61	8.45	**0.05**
VisATB ($\boldsymbol{\alpha}=[0,0,1]$)	0.1455	0.5706	77.46	61.30	20.08	67.04	71.52	**12.21**	0.30
VisATB ($\boldsymbol{\alpha}=[0.50,0.50,0]$)	0.1340	0.5626	78.39	62.27	20.04	67.92	61.92	8.58	**0.06**
VisATB ($\boldsymbol{\alpha}=[0.33,0.33,0.33]$)	0.1321	0.5591	78.12	62.08	19.68	66.68	66.08	8.77	0.32
VisATB ($\boldsymbol{\alpha}=[0.25,0.25,0.50]$)	0.1437	0.5724	77.99	61.81	20.16	67.48	67.38	**11.29**	0.15
VisATB ($T=2.0$)	0.1433	0.5642	78.10	62.08	20.16	67.67	63.06	9.92	0.12
VisATB ($T=1.0$)	0.1369	0.5752	78.15	62.09	20.32	67.71	65.00	10.24	**0.11**
VisATB ($T=0.5$)	0.1437	0.5724	77.99	61.81	20.16	67.48	67.38	**11.29**	0.15
VisATB (precise Diff)	0.1345	0.5604	78.00	61.87	21.04	67.46	67.83	10.75	0.16
VisATB (real Diff)	0.1437	0.5724	77.99	61.81	20.16	67.48	67.38	**11.29**	**0.15**

Table 14: Detailed results of ablation studies for the zero-shot tasks on the Academic Benchmark.

Methods	TextVQA↑	POPE↑	MME↑	SQA↑	MMBench↑	SEED^{I}↑	MM-Vet↑	**Overall ($\Delta I_{\text{zero}}\%$↑)**
EW	53.96	86.83	1524.41	60.74	48.54	54.13	29.00	0.00
VisATB ($\boldsymbol{\alpha}=[1,0,0]$)	54.40	86.87	1499.87	60.86	51.03	56.79	29.00	**1.36**
VisATB ($\boldsymbol{\alpha}=[0,1,0]$)	54.19	86.49	1511.93	59.59	39.09	48.14	30.60	-3.96
VisATB ($\boldsymbol{\alpha}=[0,0,1]$)	53.87	86.68	1487.54	60.74	49.91	55.55	29.50	0.63
VisATB ($\boldsymbol{\alpha}=[0.50,0.50,0]$)	54.73	86.53	1518.02	60.79	51.89	57.83	29.50	2.32
VisATB ($\boldsymbol{\alpha}=[0.33,0.33,0.33]$)	54.07	86.47	1514.76	61.26	52.92	57.39	30.80	**3.04**
VisATB ($\boldsymbol{\alpha}=[0.25,0.25,0.50]$)	53.87	86.73	1501.86	61.07	51.46	58.05	29.30	1.87
VisATB ($T=2.0$)	54.07	86.66	1498.19	61.02	49.66	54.89	29.30	0.50
VisATB ($T=1.0$)	54.13	86.45	1499.72	60.29	49.48	56.21	29.10	0.52
VisATB ($T=0.5$)	53.87	86.73	1501.86	61.07	51.46	58.05	29.30	**1.87**
VisATB (precise Diff)	54.25	86.77	1504.47	60.83	51.72	57.47	30.80	**2.61**
VisATB (real Diff)	53.87	86.73	1501.86	61.07	51.46	58.05	29.30	1.87

E Detailed Ablation Results

The detailed ablation results on the Academic Benchmark are presented in Tables 13 and 14.

F The Precise Calculation Approach for Intra-Task Difficulty

In the precise calculation approach for intra-task difficulty, the intra-task difficulty of Task i can be calculated as follows:

$$D_i = 1 - \frac{V_i(\text{mini}_i)}{V_i(i)}, \tag{12}$$

where $V_i(\text{mini}_i)$ denotes the validation performance on Task i of a model trained on the mini subset of Task i, and $V_i(i)$ signfies the validation performance on Task i of a model trained on Task i itself.

G The Sampling Rate for Sufficiently Large Subsets

As validated by experimental results in Sections 3.2 and 3.3, 1/4th subsets in the M^3IT Benchmark and the entire datasets in the Academic benchmark are sufficient for VisATB to accurately measure inter-task contribution and intra-task difficulty. Specifically, the 1/4th subset of VQA in the M^3IT Benchmark contains 14k samples, while the entire dataset of ChartQA in the Academic benchmark comprises 18k samples. Therefore, we recommend that subsets containing more than 10k samples and trained for over 100 steps are sufficiently large to ensure effective training of VisATB.