ViFT: Towards Visual Instruction-Free Fine-tuning for Large Vision-Language Models

Anonymous ACL submission

Abstract

Visual instruction tuning has become the predominant technology in eliciting the multimodal task-solving capabilities of large vision-language models (LVLMs). Despite the success, as visual instructions require images as the input, it would leave the gap in inheriting the task-solving capabilities from the backbone LLMs, and make it costly to collect a large-scale high-quality dataset. To address it, we propose ViFT, a visual instruction-free fine-tuning framework for LVLMs. In ViFT, we only require the text-only instructions and image caption data during training, to separately learn the task-solving and visual perception abilities. During inference, we extract and combine the representations of the text and image inputs, for fusing the two abilities to fulfill multimodal tasks. Experimental results demonstrate that ViFT can achieve state-of-the-art performance on several downstream benchmarks, with rather less training data. Our code and data will be publicly released.

1 Introduction

Recently, large vision-language models (LVLMs), built upon existing visual encoders (Dosovitskiy, 2020; Radford et al., 2021) and large language models (LLMs) (Brown, 2020; Zhao et al., 2023b), have gained widespread attention by demonstrating superior performance across diverse multimodal tasks (Du et al., 2022; Yin et al., 2023). To empower LVLMs with multimodal task-solving capabilities, a fundamental problem is to inherit and transfer the task-solving ability of LLMs into multimodal tasks (with image inputs). Recently, visual instruction tuning (Liu et al., 2024c,a) has emerged as the predominant framework to achieve this goal. Through fine-tuning on a variety of vision-language instruction-following data from different sources, LVLMs can directly learn the corresponding knowledge and generalize into other related tasks.

Despite its success, it is still necessary to continue scaling up the number of visual instruction data for fully learning multimodal advanced capabilities (*e.g.,* visual reasoning). However, there are two bottlenecks that greatly limit the scaling of visual instructions. First, due to the multimodal nature, visual instructions[1] need to incorporate both visual contents (*e.g.,* images or videos) and related language instructions, which makes the creation of large-scale visual instructions much more challenging compared to unimodal language instructions. Second, although existing work (Liu et al., 2024c; Zhu et al., 2023) has adopted the data synthesis strategy for visual instructions, the synthesized instructions might include unreliable information regarding the visual inputs, leading to the risks of potential performance decline.

Considering the above challenges, we rethink whether it is feasible to reduce the reliance on visual instruction data during training LVLMs. Existing LVLMs typically map visual inputs into the LLM's token space and then generate the text output based on it. If the visual inputs have been well perceived and aligned with text tokens, the LLM can comprehend the visual contents and leverage its inherent task-solving ability for tackling multimodal tasks. Therefore, LVLM's multimodal task-solving capability should be the combination of (1) the visual perception ability (for alignment) and (2) the task-solving ability from LLMs. Although it is hard and costly to synthesize extensive amount of high-quality visual instructions for learning the multimodal capabilities, it is promising to sufficiently learn the two individual abilities separately, thanks to the rich resources of natural language instructions (Wei et al., 2021; Teknium, 2023) and image caption data (Schuhmann et al., 2021; Chen

[1]Following prior works (Liu et al., 2024c), we exclude image captions from the scope of visual instructions, as they are designed for basic vision-language alignment, instead of learning advanced multimodal task-solving capabilities.

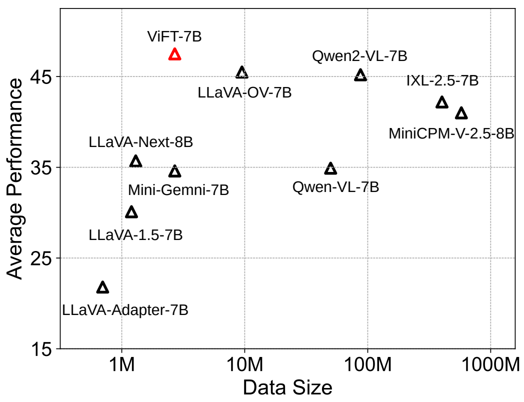


Figure 1: A comparison of ViFT with other instruction-tuned LVLM in terms of the training data size and average benchmark performance. ViFT is fine-tuned without any visual instruction data.

et al., 2024). Therefore, our goal is to ***disentangle and separately strengthen*** the two individual abilities during training, then ***combine them during inference*** to enhance LVLMs.

In this work, we propose a **V**isual **I**nstruction-**F**ree fine-**T**uning framework (ViFT) for training LVLMs. Concretely, we ensure that the two key abilities are separately optimized after extracting from the LVLM, and then combined during inference. In this way, we can take advantage of using available image caption and natural language instruction data, to better learn the two individual abilities. During inference, we extract the hidden states of the LVLM by using only the image and text parts from the input visual instruction, which are the ***disentangled representation vectors*** (Subramani et al., 2022; Turner et al., 2023) corresponding to the two individual abilities. Through the addition of two vectors, the LVLM can benefit from the improvement on the individual abilities and well fulfill multimodal tasks. Note that as ViFT does not require any visual instruction data for fine-tuning, it can better inherit the original abilities from LLMs, and avoid the high costs for synthesizing high-quality visual instructions, especially in new visual domains.

To study the effectiveness of our approach, we conduct extensive experiments on a series of benchmarks. Our approach outperforms the visual instruction tuning baseline across all seven evaluation tasks, achieving an average performance gain of 6.3% (47.5 vs. 41.2). When compared with the state-of-the-art LVLM, LLaVA-OneVision, our ViFT framework attains superior average performance (47.5 vs. 45.5), while using less than 30% amount of the training data, as shown in Figure 1. The primary contributions of this work can be summarized as followed:

- To the best of our knowledge, ViFT is the first instruction-free fine-tuning method with comparable performance to SOTA LVLMs.
- We specially designed the training and inference methods for disentangling and combining natural language task-solving and visual perception abilities, to efficiently improve the multimodal capabilities of LVLMs.
- Our ViFT is a low-cost approach for scaling data to improve LVLMs. Experimental results demonstrate the effectiveness of our approach on several benchmarks.

2 Related Work

Large Vision-Language Models. Large vision-language models (LVLMs) (Liu et al., 2024c,a) are capable of processing visual and textual inputs and tackling a variety of multimodal tasks. Currently, visual instruction tuning is the predominant framework for training LVLMs. By training on a large number of visual instructions, LVLMs can directly learn the task-solving capabilities for the corresponding multimodal tasks. Early studies (Liu et al., 2024c; Zhu et al., 2023) leverage LLMs to synthesize image-related GPT-style visual instructions. Subsequent studies leverage more advanced LVLMs (*e.g.,* GPT-4V) for higher-quality instruction synthesis (Du et al., 2023; Chen et al., 2024) and quantity scaling (Zhao et al., 2023a; Chen et al., 2025b). In addition to general instruction following, another line of works focus on the LVLM's visual reasoning capability (Zhang et al., 2024c; Shi et al., 2024; Gao et al., 2023) and the performance in other visual domains (*e.g.,* geometry (Shi et al., 2024; Gao et al., 2023), scientific (Saikh et al., 2022), and medical (Zhang et al., 2023a)). Despite its success, it's costly to synthesize high-quality visual instructions, particularly when adapting to diverse new visual domains and visual tasks.

Representation Engineering for LLMs. Our approach is closely related to studies of the representation engineering for LLMs (Zou et al., 2023; Turner et al., 2023), which aims to extract a compact vector from the LLM's intermediate representation (*e.g.,* hidden states). The extracted representations can be leveraged to manipulate the LLM's behaviour. An application of representation vectors is task arithmetic (Ilharco et al., 2022; Turner et al., 2023). Through feature engineering (*e.g.,* addition) of the representations, the LLM's behaviour on target tasks can be effectively controlled. Representation engineering is successfully implemented across various tasks, including style transfer (Subramani et al., 2022), knowledge editing (Hernandez et al., 2023), and sentiment control (Turner et al., 2023). Recent researches (Hendel et al., 2023; Liu et al., 2023) extend their application to in-context learning, where they are referred to as task vectors. In our study, we leverage the representations extracted from the LVLM to combine the individual abilities for solving multimodal tasks.

3 Preliminary

Existing LVLMs (Liu et al., 2024c; He et al., 2024b) generally consist of a pretrained visual encoder $f(\cdot)$ to process visual inputs (*e.g.,* images or videos), a connection layer $g(\cdot)$ for feature projection, and an LLM $p(\cdot)$ for autoregressive generation. During inference, given a visual instruction including an image input v and a text instruction query q, the image is first processed through visual encoder $f(\cdot)$ and connection layer $g(\cdot)$, producing visual tokens $X_v = [x_{v_1}, ..., x_{v_n}]$. These tokens are then prepended to the tokens of the text input X_q to compose the input of the LLM for autoregressively generating the target text. To train the LVLM for integrating the visual encoder and LLM, existing methods mainly incorporate two training stages: alignment pre-training and visual instruction tuning. The first stage only requires caption data and the second stage requires visual instructions.

In this work, we aim to skip the visual instruction tuning stage, and only train the model with captions and text instructions for disentangling and improving the task-solving and visual perception abilities. For model architecture, we follow LLaVA's design. Concretely, we adopt SigLIP as the visual encoder according to its suggestion (Liu et al., 2024b), and select Qwen2.5-7B-instruct (Yang et al., 2024) as our base LLM due to its remarkable performance. For connection layer, we follow the widely-used setting in current LVLMs (Liu et al., 2024c; Li et al., 2024b) that implement a simple 2-layer MLP.

4 Approach

In this section, we introduce ViFT, a visual instruction-free fine-tuning framework for LVLMs. Our main motivation is that the multimodal task-solving capability of LVLMs can be split into the task-solving ability of LLMs and the visual perception ability, which can be separately learned through text-only instructions and image caption data. In ViFT, we first collect the above data to fine-tune the LVLM for learning the two individual abilities, and then extract their corresponding representation to integrate the individual abilities during inference to tackle multimodal tasks. We show the overall framework in Figure 2.

4.1 Ability-Specific Fine-tuning

Previous LVLMs learn the multimodal task-solving capabilities by fine-tuning on visual instructions. In contrast, we propose to learn the task-solving and visual perception abilities separately, using text instructions and image caption data.

Text Instructions. We employ text instructions to facilitate the learning of task-solving ability. Specifically, we first sample text instructions from FLAN (Longpre et al., 2023) and OpenHermes (Teknium, 2023). These datasets encompass a broad range of natural language tasks, including daily dialogue, knowledge utilization, multi-hop reasoning, code synthesis, *etc*. We distill responses to these queries from Qwen-2.5-72B-instruct due to its remarkable performance in multiple real-world tasks. Additionally, we include 100K text instructions from Magpie-Qwen2.5-Pro (Xu et al., 2024a). We denote the text instruction dataset as $\mathcal{D}_{\text{text}} = \{q_i, r_i\}_{i=1}^{n_t}$, where q_i and r_i represent the input query and response.

Image Caption Data. Image caption data has been widely used to improve the cross-modal alignment ability of LVLMs, enabling the models to understand and process visual inputs. We first consider the large-scale caption dataset LAION (Schuhmann et al., 2021), which contains a variety of web images, and sample 1M image-caption pairs from it. As these web-collected captions may contain low-quality noisy data, we also collect high-quality captions from LLaVAR (Zhang et al., 2023b), ShareGPT-4V (Chen et al., 2025b), and ALLaVA (Chen et al., 2024) to improve data quality. Besides, we collect images from specific domains (*e.g.,* tables, graphs, documents) and caption them based on Qwen2-VL-7B (Wang et al., 2024b), to enhance the visual perception ability on these domains. The details of the collected visual data are presented in Appendix A. We denote

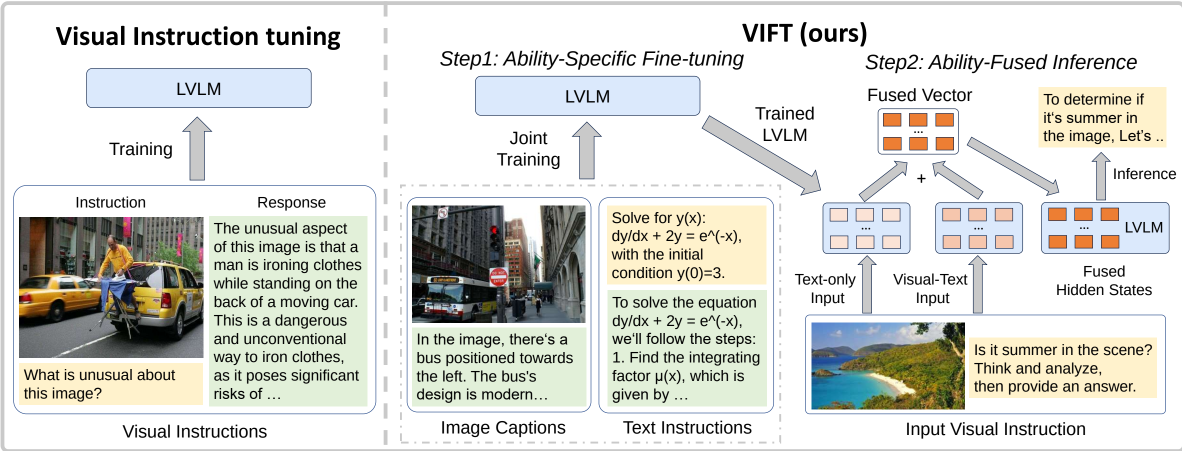


Figure 2: Compared to visual instruction tuning, ViFT first learns disentangled abilities through specific fine-tuning. During inference, given a visual instruction, we extract the disentangled ability representations through different modality inputs, and merge them into the fused representations for guiding the LVLM to generate the outputs.

the above caption data set as $\mathcal{D}_{\text{cap}} = \{v_i, r_i\}_{i=1}^{n_c}$, where v_i and r_i represent the image and caption respectively. We follow existing work (Liu et al., 2024c) to convert the caption data into instruction format to align with text instructions. Specifically, we randomly select a caption query q from a fixed query pool as its instruction. This results in a new caption dataset $\mathcal{D}'_{\text{cap}} = \{v_i, q_i, r_i\}_{i=1}^{n_c}$.

Training objective. Following previous LVLMs, we leverage an auto-regressive training objective for optimizing the parameters within the LVLM, denoted as:

$$\mathcal{L}(\theta) = -\sum_{j=1}^{N} \log \Pr(r_j | v, q, r_{<j}; \theta), \quad (1)$$

where N is the target sequence length. For text instructions, the condition of input image v is given as an empty set. In this way, we unify the learning objectives of the two kinds of data to support joint training. In application, due to the significant disparity in token length between captions and text instructions (as the image is converted to a long visual token sequence), we leverage a modality-specific batching strategy to prevent long padding sequences. By separately batching the text instructions and captions, this approach can accelerate the training process while improving the disentanglement of the two individual abilities.

4.2 Ability-Fused Inference via Disentangled Representations

After training, the task-solving and visual perception abilities are well learned. However, they cannot be combined via standard inference. Specifically, the model will elicit each individual ability for different modality inputs, as illustrated in Appendix E. We opt to address the problem via representation engineering (Subramani et al., 2022; Turner et al., 2023). The representations extracted from the model's hidden states are proven to be effective for manipulating its behavior (Subramani et al., 2022). More importantly, it enables the combination of different abilities through arithmetic operations, guiding the model to exhibit composite behavioral (Ilharco et al., 2022). Consequently, we can activate diverse abilities through different modality inputs, extract their corresponding representations, and then combine them via addition.

Extracting Disentangled Representations. We focus on the LLM part of the target LVLM as it plays a crucial role in the LVLM's behavior. The LLM consists of a stack of transformer layers. During inference, the input text will be first tokenized to a sequence of tokens $\boldsymbol{x} = [x_1, \ldots, x_n]$, where n denotes the sequence length. Then, the sequence will be processed through multiple layers, creating intermediate hidden states $\boldsymbol{h}^l(\boldsymbol{x}) = [\boldsymbol{h}^l(x_1), \ldots, \boldsymbol{h}^l(x_n)]$ at layer l. Notice that each input token will correspond to a hidden representation. For simplicity, we use $\boldsymbol{h}(\boldsymbol{x})$ to denote the hidden representations at all target layers. These representations will later be used to manipulate the model's behavior.

Task-Solving Ability Representations. Owing to our design in specific fine-tuning, the task-solving ability is mainly learned by text-only instructions. Thus, given a input visual instruction $I = \{q, v\}$, where q and v represent the text query and the paired image respectively, we can utilize the text part of the input visual instruction to elicit the task-solving ability from the LVLM. Although the text part is not sufficient for fulfilling the multi-modal task, it can still prompt the model to exhibit the task-solving behavior. Therefore, we aim to extract the representation for such ability. Concretely, we simply use the text instruction q as input, and extract the hidden representations across all target layers. Notably, for text-only inputs, the extracting process is the same for LLMs and LVLMs. We denote the extracted representations $\boldsymbol{h}(q)$ as the task-solving ability representation.

Visual Perception Ability Representations. We additionally utilize the image part v of the input visual instruction, to extract the representations for the visual perception ability. Here, we utilize the LVLM to process both the input image v and text q. The input image and text will be converted to a sequence of tokens. Next, we extract the hidden representations $\boldsymbol{h}(v, q)$ of the text part from all layers. In this way, as the text representations can attend to all image tokens, they contain the information from the image part. Besides, they will also have the same size as the task-solving ability representations, which do not need further alignment and also supports simple fusing strategies like addition operators.

Ability-Fused Inference. After extracting the two ability representations, we aim to combine them to activate the corresponding capabilities for tackling multimodal tasks. According to the studies of representation engineering (Subramani et al., 2022; Turner et al., 2023), intervening in the LLM's representation space can manipulate its behaviour. For instance, incorporating representations extracted from task-specific demonstrations enables the model to address the target task without training. In our case, we expect the model to exhibit both visual perception and task-solving abilities for tackling actual tasks. Therefore, we devise a simple but effective ability fusion strategy via weighted addition. In this way, we can easily combine the two abilities and control the fusion degree. Concretely, given a visual instruction with image v and text instruction q, the ability-fused representation is computed as:

$$\boldsymbol{h}'(v, q) = \alpha \boldsymbol{h}(v, q) + \beta \boldsymbol{h}(q) \qquad (2)$$

Here, α and β are two tunable weights. Given an image v and a text instruction q, we first extract the ability representations $\boldsymbol{h}(v, q)$ and $\boldsymbol{h}(q)$, and then compute the ability-fused representation $\boldsymbol{h}'(v, q)$. Next, during inference, we replace the hidden representation of the input text tokens with the fused ability representation, and autoregressively generate the output tokens. The entire generation process requires only one additional forward pass, and we will discuss the associated computational overhead in Section 6.

5 Experiment

5.1 Evaluation Benchmarks

To evaluate the performance of ViFT, we conduct experiments on seven public benchmark datasets: (1) MME (Fu et al., 2024) and MMMU (Yue et al., 2024) for evaluating visual perception and visual commonsense; (2) MathVista (Lu et al., 2023), Mathverse (Zhang et al., 2025a), Math-Vision (Wang et al., 2024a) and Olympiad-bench (He et al., 2024a) for evaluating visual reasoning; (3) LLaVA-Bench (Liu et al., 2024c) for evaluating visual instruction following. We ensure that the selected evaluation benchmarks have no domain overlap with ViFT's training data. Notably, for models that are only capable of generating direct answers, we employ chain-of-thought prompting to elicit its reasoning ability during evaluation.

5.2 Baselines

We compare ViFT with two types of baselines: (1) Visual instruction tuning (VIT). To ensure fair comparison for ViFT and VIT, we collect 2.7M (same amount as ViFT's training data) visual instructions from commonly-used public visual instruction datasets (LLaVA-Instruct (Liu et al., 2024a), ALLaVA-4V (Chen et al., 2024), SViT (Zhao et al., 2023a), M3IT (Li et al., 2023), and Vision-Flan (Xu et al., 2024b)). We then train a baseline LVLM (denoted as ViT-7B) with the collected VIT data under same settings for fair comparison. (2) Open-sourced LVLMs. We compare our ViFT-7B with several public open-sourced LVLMs with similar parameter size. These models include LLaVA-1.5 (Liu et al., 2024a), LLaVA-Next (Liu et al., 2024b), MiniCPM-V-2.5 (Yao et al., 2024), LLaMA-3.2-Vision (Meta, 2024), InternLM-XComposer (IXL-2.5) (Zhang et al., 2024a), Qwen2-VL (Wang et al., 2024b), and LLaVA-OneVision (Li et al., 2024a). Note that

Model	MME	MMMU	MVista*	MVerse*	MVision	OlyBench	LBench	Average
LLaVA-1.5-7B	64.5	35.7	25.6	12.1	8.5	2.8	61.8	30.1
LLaVA-Next-8B	68.1	43.1	41.0	13.9	14.1	3.7	66.0	35.7
MiniCPM-V-2.5-8B	72.3	45.8	46.6	20.5	14.1	5.1	82.7	41.0
LLaMA-3.2-Vision-11B	65.0	48.0	48.7	26.1	15.8	4.9	**83.1**	41.7
IXL-2.5-7B	**79.7**	42.9	54.4	27.2	14.8	5.9	70.2	42.2
Qwen2-VL-7B	79.1	52.0	58.3	30.5	17.7	8.4	70.1	45.2
LLaVA-OneVision-7B	76.1	46.6	**58.9**	31.0	18.1	6.7	81.0	45.5
VIT-7B	73.1	44.2	43.6	26.9	16.5	5.8	78.3	41.2
ViFT-7B (ours)	78.2	**52.8**	49.2	**34.8**	**24.0**	**12.1**	81.5	**47.5**

Table 1: A comparison between ViFT and other baseline models on seven benchmarks. VIT denotes our visual instruction tuning baseline. MVista, MVerse, MVision, OlyBench and LBench are short for MathVista, MathVerse, Math-Vision, OlympiadBench and LLaVABench, respectively. We use * to denote benchmarks that may have domain overlap with the training data of the baseline model. We report the normalized performance for MME. **Bold** and underline fonts indicate the best and second best performance, respectively.

Model	MVision	MME	MMMU
VIT-Qwen2.5-7B-SigLIP	16.5	73.1	44.2
ViFT-Qwen2.5-7B-SigLIP	**24.0**	**78.2**	**52.8**
VIT-Qwen2.5-14B-SigLIP	19.6	77.8	48.5
ViFT-Qwen2.5-14B-SigLIP	**26.2**	**79.4**	**54.6**
VIT-LLaMA3-8B-SigLIP	15.8	74.4	46.8
ViFT-LLaMA3-8B-SigLIP	**20.2**	**78.6**	**51.4**
VIT-Qwen2.5-7B-CLIP	16.8	71.4	44.0
ViFT-Qwen2.5-7B-CLIP	**19.6**	**75.4**	**49.8**

Table 2: The performance of ViFT across LVLMs with different model architectures.

many of the open-sourced LVLMs include much more training data compared to ViFT, and some models may have used training data from the same domain as the evaluation benchmarks, which may lead to unfair comparison. We explicitly mark them in our experimental results for clarity.

5.3 Implementation Detail

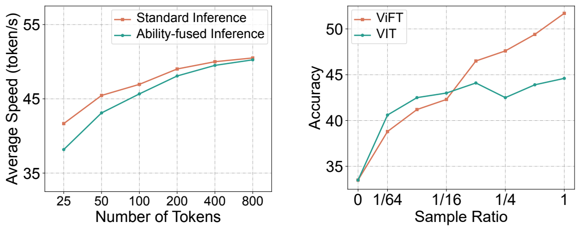
We adopt a two-stage training strategy: In the first stage, we train on web captions. In the second stage, we train on a mixture of high-quality captions and text instructions. This avoids the additional computational overhead caused by the significant length disparity between low-quality web captions and high-quality captions. We provide a comparison of these two strategies in Appendix D. We set the learning rate to 1e-5 for the LLM and vision encoder, and 2e-6 for the connector layer. The batch size is configured as 8 for each GPU. All models are trained for one epoch.

During inference, we only conduct ability fusion in the top 50% of layers. We set $\alpha = 1.0$ and $\beta = 0.1$ across all experiments. More detailed studies of the optimal hyperparameters and fusion layers are presented in Section 6.

5.4 Main Results

We present the results of ViFT and other baseline models in Table 1. Firstly, we observe that ViFT-7B outperforms the ViT-7B baseline across all benchmarks, with the most significant gains on visual commonsense and visual reasoning tasks. This demonstrates that our ViFT effectively inherits the LLM's language reasoning capability for visual reasoning tasks, substantially enhancing its performance. In contrast, the ViT paradigm tends to cause models to overfit to superficial generation patterns and fail to develop genuine visual reasoning capability. Moreover, due to the challenges in visual instruction synthesis, existing visual instruction datasets often contain low-quality samples that further disrupt the model's normal reasoning and perceptual capabilities, leading to its performance decline. The results prove that ViFT is more effective than VIT under fair comparison.

Secondly, compared to other open-sourced LVLMs, ViFT achieves the best performance on four out of all seven tasks. The only task where ViFT shows a notable performance gap is MathVista, which we attribute to the baseline LVLM's potential use of in-domain training data overlapping with the benchmark, providing significant performance advantages. Although similar issues may also exist in MathVerse, a benchmark for evaluating the models' geometric reasoning capability. ViFT can compensate for the lack of in-domain training data by inheriting reasoning capability from the backbone LLM, thereby achieving even stronger



(a) Efficiency Test

(b) Scaling Test

Figure 3: Efficiency test and scaling test for ViFT.

Model	MathVista	LLaVABench
ViFT	49.2	81.5
¬ Low-quality captions	48.5	80.9
¬ High-quality captions	42.1	66.1
¬ Text instructions	43.9	66.0
¬ AF inference	46.1	59.6

Table 3: The ablation of different training data component and inference strategy. AF inference indicates ability-fused inference.

performance. Considering the overall average performance, ViFT-7B outperforms the leading baseline LVLM, LLaVA-OneVision-7B (47.5 vs 45.5), despite being trained on a substantially smaller dataset (2.7M vs 9.5M). This demonstrates the effectiveness of our proposed framework.

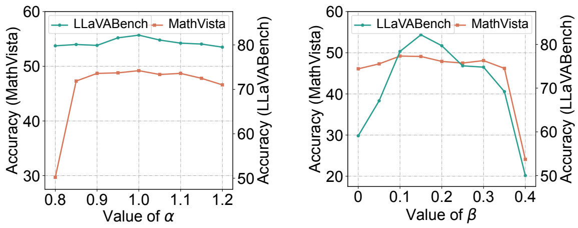
6 Further Analysis

Transferability Evaluation. We evaluate the transferability of ViFT across different model architectures. Specifically, we modify our base ViFT-7B model by replacing the backbone LLM (Qwen2.5-7B-Instruct) with LLaMA-3-8B-Instruct and Qwen2.5-14B-Instruct, and the visual encoder (SigLIP) with CLIP, while keeping all other experimental configurations unchanged. For all variant models, we train both VIT-based and ViFT-based models based on the same training data used in our main experiment. Then, under the same setting, we evaluate the performance of ViFT and VIT for each model variant. We present the results in Table 2. We can observe that ViFT consistently outperforms ViT across various downstream tasks, regardless of the model architecture. The results prove that ViFT is a general and robust approach, outperforming the VIT baseline across diverse model architectures.

Computation Complexity. We examine the additional time overhead of ability-fused inference compared to standard inference. The results are presented in Figure 3a. As we can observe, when generating short responses (*e.g.,* 25 tokens), utilizing ability-fused inference may introduce an 8% increase in computational overhead. However, as the generation length increases, the generation speeds for standard inference and ability-fused inference gradually converge. When generating more than 400 tokens, ability-fused inference almost doesn't introduce any additional computational overhead. This aligns with our expectations. For ability-fused inference, we merely introduce one additional forward pass during the entire generation process. Thus, while there is some discrepancy when generating short responses, such differences become negligible as generation length increases.

Data Scaling Test. We investigate the effect of data scaling for visual instruction tuning (VIT) and ViFT. Concretely, we randomly sample data subsets at different sampling ratios from ViFT's training data and the baseline visual instruction data, respectively. We then train LVLMs with these data subsets and evaluate their average performance on three benchmarks: MME, MMMU and MathVision. As shown in Figure 3b, we observe that data scaling consistently yields performance improvements for ViFT. This indicates that the enhancements in two individual abilities effectively propagate to improved fused multimodal task-solving capability. As for visual instruction tuning, the model achieves promising performance improvement with minimal data, but cannot yield significantly better results via scaling. This likely occurs because existing visual instructions primarily help models learn superficial styles, rather than improving actual multimodal task-solving capability. This further validates that ViFT demonstrates greater potential for performance improvement by leveraging existing large-scale, cost-effective data, compared to conventional approaches.

Ablation Study. We employ diverse training data components and inference strategy in our training framework. We present the ablation results in Table 3. First, we examine the impact of each data component for fine-tuning. We observe that removing high-quality captions or text instructions can result in severe performance decline. This indicates that the high-quality captions is important for enhancing the model's visual perception ability, which subsequently improves their multimodal task-solving capability. Text instructions are



(a) Impact of α (b) Impact of β

Figure 4: The impact of different hyperparameters.

Layers	MathVista	MathVision	LLaVABench
0-7	48.8	17.0	63.1
0-14	46.4	19.6	74.6
0-21	<u>49.0</u>	17.3	80.3
0-28	47.8	16.4	76.7
7-28	48.1	19.6	<u>81.2</u>
14-28	**49.2**	**24.0**	**81.5**
21-28	46.3	<u>22.8</u>	73.9

Table 4: The impact of fusion layer selection.

equally important as they preserve the LLM's inherent task-solving ability from multimodal training. In comparison, the impact of low-quality captions is relatively limited. Second, we study the effect of our proposed ability-fused inference. As we can observe, the model exhibits significant performance decline without ability-fused inference. This indicates that the individual abilities cannot be effectively combined through standard inference, and our proposed ability-fused inference successfully addresses this limitation.

Hyperparameter Tuning. We study the effect of different hyperparameter α and β on model performance. The results are presented in Figure 4. For α, we observe that as α increases, the model's performance initially increases and then decreases. While the performance on MathVista exhibits a sudden improvement at early stages, it remains relatively stable as α changes. The results confirm that $\alpha = 1.0$ represents an optimal choice, while small deviations do not significantly impact performance. Similar to α, the model performance exhibits an increase-then-decrease pattern as β varies. We observe a sudden decline when β reaches 0.4, indicating that a large β can result in abnormal behavior. Furthermore, we discover that the optimal β varies across different tasks. The optimal β for MathVista and LLaVA-Bench are 0.1 and 0.15, respectively. This demonstrates that different vision tasks may require varying levels of individual abilities, leading to task-specific optimal fusion ratios. However, the optimal fusion ratios across different tasks do not deviate significantly, and slight deviations from these ratios do not cause substantial performance degradation.

Fusion Layer Selection Analysis. We investigate the impact of layer selection for ability fusion. We examine two strategies: selecting layers from the top downward, or from the bottom upward (we refer the layers nearer to the LLM head as the top layers). The results are presented in Table 4. Our findings indicate that the top-down strategy consistently outperforms bottom-up selection. The reason might be that the LLM's top layers have more influence on the model's generation behavior (Geva et al., 2020, 2022), which makes ability fusion more effective at these layers. Also, recent studies (Chen et al., 2025a; Zhang et al., 2025b) demonstrate that visual information tends to aggregate with text tokens within the LVLM's early layers, and ability fusion in these layers may disrupt such a aggregation process, leading to declined performance. Moreover, we discover that selecting 50% of the layers from the top of the model downward yields the best performance, which makes it an optimal choice.

7 Conclusion

In this paper, we proposed an instruction-free fine-tuning framework ViFT, for enhancing the multimodal task-solving capabilities of LVLMs. Concretely, instead of using visual instructions, we only leveraged text instructions and image caption data, to separately learn the individual task-solving and visual perception abilities for the LVLM. After that, we extracted the representation vectors by using the model's hidden space for the disentangled abilities via different modality inputs, and combined them to guide the inference of the LVLM in multimodal tasks. With rather less training data, our trained model, ViFT-7B, achieved state-of-the-art performance among competitive LVLMs across various downstream benchmarks. Furthermore, based on our proposed framework, we can efficiently scale the vision data and text data to enhance the model's performance, which facilitates further advancements in this field.

8 Limitations

In this paper, we propose ViFT, a visual instruction-free fine-tuning framework for training LVLMs. While our approach achieves promising performance on downstream benchmarks, it still has some potential limitations. First, we prove that our approach can be enhanced by scaling the vision data and text data for training. Since there already exists well-established methods for efficiently synthesizing such data in large quantities (Yu et al., 2023; Zhou et al., 2024), there is still room for further improvement. Second, we utilize captions as the primary multimodal data to facilitate the learning of visual perception ability. While this approach represents the current mainstream practice, whether coarse-grained captions constitute the optimal data choice for visual perception learning across all vision domains remains an open research question. Third, apart from the visual reasoning and instruction following capabilities, ViFT has the potential to efficiently transfer more advanced capabilities to visual tasks (*e.g.,* long-thought reasoning (OpenAI, 2024; Guo et al., 2025)), and we will further explore this direction in subsequent work.

References

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. 2024. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2025a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2025b. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yifan Du, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, Jinpeng Wang, Chuyuan Wang, Mingchen Cai, Ruihua Song, and Ji-Rong Wen. 2023. What makes for good visual instructions? synthesizing complex visual reasoning instructions for visual instruction tuning. *arXiv preprint arXiv:2311.01487*.

Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. 2022. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*.

Chaoyou Fu, Yi-Fan Zhang, Shukang Yin, Bo Li, Xinyu Fang, Sirui Zhao, Haodong Duan, Xing Sun, Ziwei Liu, Liang Wang, et al. 2024. Mme-survey: A comprehensive survey on evaluation of multimodal llms. *arXiv preprint arXiv:2411.15296*.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. 2023. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 2024a. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024b. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

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

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. 2023. M3it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv preprint arXiv:2306.04387*.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024b. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024c. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2023. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Preprint*, arXiv:2301.13688.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI Blog. Retrieved December*, 20:2024.

OpenAI. 2024. Learning to reason with llms.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv:2402.14804*.

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

Jason Wei, Maarten Bosma, Vincent 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. In *International Conference on Learning Representations*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024a. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. 2024b. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. *arXiv preprint arXiv:2402.11690*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. 2024a. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*.

Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. 2024b. Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *The Twelfth International Conference on Learning Representations*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. 2025a. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. 2024c. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025b. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023a. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*.

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*.

Bo Zhao, Boya Wu, Muyang He, and Tiejun Huang. 2023a. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023b. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024. Jiuzhang3. 0: Efficiently improving mathematical reasoning by training small data synthesis models. *arXiv preprint arXiv:2405.14365*.

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*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Training Dataset

We utilize OpenHermes (Teknium, 2023) and FLAN (Longpre et al., 2023) as sources for text instruction queries. Open-Hermes comprises a diverse collection of text instructions from various sources and FLAN contains a substantial set of task-specific instructions. We anticipate these instruction sets will enhance the model's language capability in both general scenarios and complex reasoning tasks. Following query acquisition, we employ Qwen2-72B-instruct to distill specific instruction responses. This approach is adopted because the distilled responses demonstrate higher quality compared to the original responses. Additionally, since Qwen2-72B-instruct shares the same training data as our base LLM (though with different parameter scales), we hypothesize this alignment would better preserve the original language capabilities.

As for vision data, we collect extensive caption datasets encompassing both general and domain-specific vision domains. The details of the vision data are presented in Table 5.

Image Source	Domains	Numbers
ALLaVA	General	564976
LLaVAR	General	250000
ShareGPT	General	200000
COCO	General	82783
VG	General	158557
Synthdog	OCR	29765
GeoQA	Math	6027
CLEVR	Math	20000
FigureQA	Figure	20000
DocVQA	Document	10194
TabMWP	Table	20000
ChartQA	Chart	18317
DVQA	Diagram	30000
MMarxiv	Academic	54399
IconQA	Icons	18946
AI2D	Science	4903
ScienceQA	Science	6757

Table 5: Details of caption training data.

B Evaluation Datasets

We evaluate ViFT on four downstream benchmarks, the details of the benchmarks are as followed:

- *MathVista:* (Lu et al., 2023) it evaluates the LVLM's mathematical reasoning capabilities in multiple vision domains. It contains 6141 evaluation data samples, collected from 28 existing datasets and 3 newly created datasets.

- *MathVerse:* (Zhang et al., 2025a) it is an in-depth benchmark for evaluating LVLM's reasoning capability. It consists of 2612 math problems, and each problem is transformed into 6 distinct problem versions. We report the full performance (ALL) and the performance on vision-mini (V-mini) subset in our experiments.

- *MathVision:* (Wang et al., 2024a) it develops a comprehensive and challenging benchmark for evaluating the LVLM's advanced reasoning skill. It comprises 3040 high-quality mathematical problems derived from authentic mathematics competitions. These problems encompass several distinct mathematical disciplines and are categorized across five difficulty levels.

- *LLaVABench:* (Liu et al., 2024c) it evaluates the model's instruction-following capabilities across diverse visual scenarios, assessing its conversation, detailed description, and complex reasoning skills. Through carefully designed prompts, it measures the model's ability to generate accurate responses based on complex instructions in visual contexts.

- *MME:* (Fu et al., 2024) it evaluates the model's visual perception and visual commonsense reasoning abilities. Every instance in MME consists of one image and two binary questions. We evaluate all models on both perception and cognition splits of MME and report the normalized results.

- *MMMU:* (Yue et al., 2024) it is a comprehensive benchmark for evaluating the model's capability on visual commonsense and reasoning on massive multi-discipline tasks. It includes 11.5K meticulously collected questions spanning 30 subjects and 183 subfields.

- *OlympiadBench:* (He et al., 2024a) it consists of 8,476 bilingual multimodal problems for Olympic-level mathematics and physics competitions, which is extremely challenging and require high-level reasoning skills for LVLMs.

Model	# Captions	# Others	# Total
LLaVA-1.5	560K	665K	1.2M
LLaVA-Next	560K	760K	1.3M
MiniCPM-V-2.5	570M	8.3M	578M
IXL-2.5	> 400M	> 2M	> 402M
Qwen2-VL	> 87.5M	-	> 87.5M
LLaVA-OV	5.5M	4.0M	9.5M
ViFT	2.5M	0.2M	2.7M

Table 6: The statistics of training data for ViFT and other baseline LVLMs. Given that Qwen2-VL and IXL do not provide their specific training data volumes, we estimate the minimum data size based on the training data descriptions provided in their papers.

C Baselines

We compare ViFT with a number of existing open-source LVLMs. Notably, Although IXL-2.5 and Qwen2-VL achieve impressive performance, they are trained on a extensive multimodal datasets (exceeding 80M samples). LLaVA-OneVision, on the other hand, utilizes a relatively smaller but still substantial dataset of 9.5M samples. Compared to these models, our ViFT demonstrates superior data efficiency by requiring only 2.7M training samples. We present more detailed information of baseline LVLMs in Appendix C. Apart from the baseline LVLMs introduced above, we also include LLaVA-Adapter (Zhang et al., 2024b) and Mini-Gemini (Li et al., 2024b) in Figure 1. We report the training data size of these models in Table 6. For models without exact number of training data size in papers, we estimate the lower bound of the data size. For IXL-2.5, we report the incomplete training data size in their paper. For Qwen2-VL, we estimate the training data size by dividing the total training tokens with the max token length for each sample.

D Additional Experiments

D.1 One-stage training vs two-stage

During training, we adopt a two-stage training strategy to reduce cost. We investigate whether a two-stage training strategy produces significant different results compared to a one-stage approach. The results are presented in Table 7. As we can observe, single-stage training exhibits almost the same performance as two-stage training, yet is more efficient.

Strategy	Time	MathVista	MathV	LBench
Two-Stage	≈ 36h	49.2	24.0	81.5
One-Stage	≈ 47h	49.8	23.2	82.0

Table 7: Comparison of ViFT with different training strategy.

E Case Studies

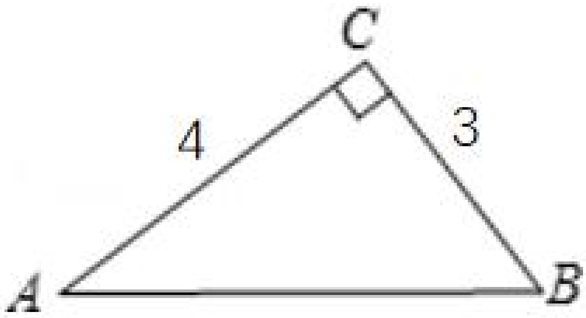
This section presents ViFT's generating behaviour under various experimental settings, and a comparison of ViFT with other baseline LVLMs. The detailed results are presented below.

Comparison of ViFT with Existing LVLMs. We present the results of ViFT and other SoTA LVLMs in Table 8, Table 9, and Table 10. As we can observe, many state-of-the-art baseline LVLMs fail to generate a high-quality reasoning path even when chain-of-thought prompt is leveraged, such as Qwen2-VL and LLaVA-OneVision. LLaMA-3.2-Vision-11B is the only baseline LVLM that is capable of generating relatively coherent chains of thought, but it may occasionally produce minor error during the reasoning process (*e.g.,* misinterprets the sine function), which results in an erroneous final result. InternLM-XComposer-2.5 sometimes generates promising reasoning path, but at other times produces shorter responses. For Qwen2-VL-7B and LLaVA-OneVision-7B, their reasoning is extremely short, leading to incorrect results. As this is actually a simple geometric problem, which should not pose a challenge for LLMs, the observed erroneous behavior is likely due to a ability degradation resulting from conflicts between knowledge learned from visual instruction tuning and the intrinsic abilities of LLM itself. Compared to these models, our ViFT is the only one that can almost fully inherit the original reasoning capabilities from its backbone LLM, generate high-quality reasoning paths, and ensure the accuracy and consistency of intermediate steps, ultimately achieving more correct results.

ViFT's Behaviour With or Without Ability-fused inference. We present the results of ViFT with or without ability-fused inference in Table 11, Table 12, and Table 13. First, we discover that the fine-tuned model will elicit diverse individual ability with different modality inputs, instead of a combined capability via standard inference. Specifically, it mostly produces a image description for multimodal inputs, regardless of the actual instruction. However, when we leverage the same instruction part as text-only input, the model follows the instruction and exhibits expected behavior, though it fails to provide a correct answer due to the missing image. Second, after implementing our ability-fused inference, ViFT maintains its visual perception capability while successfully leveraging the fused task-solving ability to generate correct responses, validating the effectiveness of our method.

ViFT's Behaviour With Different α Setting. We present the results of ViFT's behaviour with different α in Table 14. We observe that when α is set to a low value (0.8 in our case), ViFT rejects to answer the question for lack of visual information. This may be attributed to diminished visual perception ability caused by low value of α. As α gradually increases, the model begins to exhibit normal reasoning behaviour and generates accurate results. Within a certain range of α, the model's behavior remains notably consistent, showing no significant variations despite the changes of α values. We also notice that when α reaches 1.1, the model starts to first generate a image description, then answer the question. While the reasoning behavior remains largely consistent, this indicates that a larger α may enhance the model's focus on visual information.

ViFT's Behaviour With Different β Setting. The results of ViFT's behaviour with different β are presented in Table 15. We discover that when $\beta = 0$, the model exhibits a caption-only behaviour, similar to ViFT without ability fusion. As β increases, the model progressively demonstrates consistent reasoning behavior, aligning with the behavioral patterns observed during α variation. However, when β reaches 0.4, ViFT will output random strings, which indicates that a large β can result in unstable behaviour in the model. This also corresponds to the previous findings where the model's performance experiences a sudden and drastic decline once the beta value reaches a certain threshold.



Problem: Please first conduct reasoning, and then answer the question and provide the correct option letter, e.g., A, B, C, D, E, at the end.
As shown in the figure, then the value of sinB is equal to () Choices:
A:4/3
B:3/4
C:4/5
D:3/5
Ground Truth: C

LLaMA-3.2-Vision-11B:
Step 1: Understand the problem
The problem asks for the value of sinB in the given figure. To find this, we need to use trigonometric concepts, specifically the definition of sine in a right-angled triangle.
Step 2: Identify the triangle and its sides
From the figure, we can see that triangle ABC is a right-angled triangle with sides labeled as follows:
* Side AB = 3 * Side BC = 4 * Hypotenuse AC = 5
Step 3: Recall the definition of sine
The sine of an angle in a right-angled triangle is defined as the ratio of the length of the side opposite the angle to the length of the hypotenuse.
Step 4: Apply the definition of sine to find sinB
In this case, angle B is opposite side AB, and the hypotenuse is AC. Therefore, sinB = AB / AC.
Step 5: Calculate sinB
Using the values from step 2, we can calculate sinB as follows:
sinB = AB / AC = 3 / 5
Answer: D

InternLM-XComposer-2.5-7B:
To find the value of sinB, we need to use the definition of sine in a right-angled triangle. The sine of an angle in a right-angled triangle is the ratio of the length of the side opposite the angle to the length of the hypotenuse. In this case, side AC is opposite angle B, and AB is the hypotenuse.
Given that AC = 3 and AB = 4, we can calculate sinB as follows:
sinB = opposite/hypotenuse sinB = 3/4
Therefore, the correct answer is B: $\frac{3}{4}$.

Qwen2-VL-7B:
Solution: In right triangle triangle ABC, angle C=90°, AC=3, BC=4. According to the Pythagorean theorem, we have AB=5. Therefore, sinB=AC/AB=3/5. So, the answer is D.

Table 8: A comparison of ViFT with other baseline LVLMs (part1).

LLaVA-OneVision-7B:
Solution: Since angle C=90°, we have sinB=AC/AB=3/5. Therefore, the answer is D.

ViFT-7B:
To determine the value of sinB, we need to identify the sides of the triangle relative to angle B. The sine of an angle in a right-angled triangle is defined as the ratio of the length of the side opposite the angle to the length of the hypotenuse.
From the diagram: - Side AC (opposite to angle B) has a length of 4 units. - Side BC (adjacent to angle B) has a length of 3 units. - Side AB (the hypotenuse) is not explicitly given but can be calculated using the Pythagorean theorem if needed.
Using the Pythagorean theorem for triangle ABC:

$$AB^2 = AC^2 + BC^2$$

$$AB^2 = 4^2 + 3^2$$

$$AB^2 = 16 + 9$$

$$AB^2 = 25$$

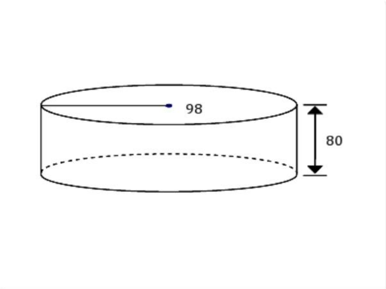
$$AB = 5$$

Now, we can calculate sinB:

$$\sin B = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{AC}{AB} = \frac{4}{5}$$

Therefore, the correct answer is C: $\frac{4}{5}$.

Table 8: A comparison of ViFT with other baseline LVLMs (part 1).



Problem: Please first conduct reasoning, and then answer the question and provide the correct answer.
Find the surface area of the given cylinder. All measurements in the diagram are in mm.
Round your answer to two decimal places.
Ground Truth: 109603.88

LLaMA-3.2-Vision-11B:
Step 1: Understand the problem
The problem asks for the surface area of a cylinder with a height of 80 mm and a radius of 98 mm.
Step 2: Identify the formula for the surface area of a cylinder
The formula for the surface area of a cylinder is:

$$\text{Surface Area} = 2\pi r^2 + 2\pi r h$$

where r is the radius and h is the height.
Step 3: Plug in the values given in the problem Given that the radius $r = 98$ mm and the height $h = 80$ mm, we can substitute these values into the formula.
Step 4: Calculate the surface area

$$\text{Surface Area} = 2\pi(98)^2 + 2\pi(98)(80)$$

$$\text{Surface Area} = 19208\pi + 15680\pi = 34888\pi \approx 34888 \times 3.14159 \approx 109541.31\,\text{mm}^2$$

Step 5: Round the answer to two decimal places Rounded to two decimal places, the surface area is approximately 109541.31 mm^2.

InternLM-XComposer-2.5-7B:
To find the surface area of a cylinder, we need to calculate the area of the two circular bases and the lateral surface area. The formula for the area of a circle is $A = \pi r^2$, where A is the area and r is the radius. First, let's find the area of one base:

$$A_{\text{base}} = \pi r^2 = \pi(98)^2 = 9624.6\,\text{mm}^2$$

Since there are two bases, we double the area of one base to get the total area of both bases:

$$A_{\text{total bases}} = 2 \times A_{\text{base}} = 2 \times 9624.6 = 19249.2\,\text{mm}^2$$

Next, we need to find the lateral surface area. The lateral surface area of a cylinder is given by the formula $A = 2\pi r h$, where h is the height of the cylinder.

$$A_{\text{lateral}} = 2\pi r h = 2\pi(98)(80) = 19200\,\text{mm}^2$$

Finally, we add the areas of the bases and the lateral surface area to get the total surface area of the cylinder:

$$A_{\text{total}} = A_{\text{total bases}} + A_{\text{lateral}} = 19249.2 + 19200 = 38449.2\,\text{mm}^2$$

Table 9: A comparison of ViFT with other baseline LVLMs (part 2).

Qwen2-VL-7B:
The surface area of a cylinder is given by the formula $2\pi rh + 2\pi r^2$, where r is the radius and h is the height. Substituting the given values, we get $2\pi \times 98 \times 80 + 2\pi \times 98^2$. Calculating this, we find the surface area to be approximately 23,567.44 mm². Therefore, the answer is 23567.44.

LLaVA-OneVision-7B:
The surface area of a cylinder is given by the formula: $2\pi r(h+r)$, where r is the radius and h is the height. Substituting the given values, we get: $2*3.1416*98*(98+80) = 2*3.1416*98*178 = 12210.4mm^2$ Therefore, the surface area of the cylinder is $12210.4mm^2$.

ViFT-7B:
To solve this problem, we need to analyze the given geometric configuration and use properties of circles and angles.
Given:
- There is a circle with center A.
- Points B, C, D, and E lie on the circumference of the circle.
- Angle BAC and angle EAD sum up to 180 degrees.
Analysis:
1. **Circle Properties**: Since A is the center of the circle, any line segment from A to a point on the circumference (e.g., B, C, D, or E) is a radius of the circle.
2. **Angle Sum Property**: The sum of opposite angles in a cyclic quadrilateral is 180 degrees. This means that if BAC and EAD are supplementary, they form a straight line when extended through the circle.
Steps to Solve:
To find the surface area of a cylinder, we use the formula:

$$A = 2\pi r(h + r)$$

where: - A is the surface area, - r is the radius of the base, - h is the height of the cylinder.
Given: - Radius (r) = 98 mm, - Height (h) = 80 mm.
Substitute these values into the formula:

$$A = 2\pi(98)(80 + 98)$$

$$A = 2\pi(98)(178)$$

$$A = 2\pi \times 17444$$

$$A = 34888\pi$$

Now, calculate the numerical value using $\pi \approx 3.14159$:

$$A \approx 34888 \times 3.14159$$

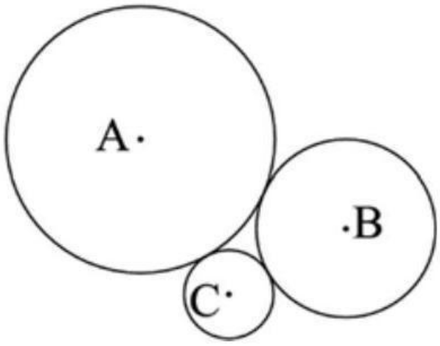
$$A \approx 109678.472$$

Rounding to two decimal places:

$$A \approx 109678.47$$

Therefore, the surface area of the cylinder is approximately 109678.47 square millimeters.

Table 9: A comparison of ViFT with other baseline LVLMs (part 2).



Problem: Please first conduct reasoning, and then answer the question and provide the correct answer. Three circles with centres A, B, C touch each other in pairs from the outside (see diagram). Their radii are 3,2 and 1. How big is the area of the triangle ABC?
Ground Truth: 6

LLaMA-3.2-Vision-11B:
To find the area of triangle ABC, we can use the formula for the area of a triangle given its base and height. The base of the triangle is the distance between points A and B, and the height is the distance from point C to line AB.
The distance between points A and B is equal to the sum of their radii, which is $3 + 2 = 5$. This is because the circles touch each other in pairs from the outside, so the distance between their centers is equal to the sum of their radii.
The distance from point C to line AB is equal to the radius of circle C, which is 1. This is because point C is on the circumference of circle C, and the radius of a circle is the distance from its center to any point on its circumference.
Now, we can use the formula for the area of a triangle:
Area = (base × height) / 2
Substituting the values we found earlier, we get:
Area = (5 × 1) / 2 = 5/2 = 2.5
Therefore, the area of triangle ABC is 2.5 square units.
Answer: 2.5

InternLM-XComposer-2.5-7B:
To find the area of triangle ABC, we can use Heron's formula, which states that for a triangle with sides of length a, b, and c, the area (A) is given by:

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where s is the semi-perimeter of the triangle, defined as:

$$s = \frac{a+b+c}{2}$$

In this case, the sides of the triangle are the distances between the centers of the circles. The distance between the centers of circles with radii 3 and 2 is the sum of their radii, so it is $3+2=5$. The distance between the centers of circles with radii 2 and 1 is $2+1=3$, and the distance between the centers of circles with radii 3 and 1 is $3+1=4$.
The semi-perimeter s of triangle ABC is therefore:

$$s = \frac{5+3+4}{2} = \frac{12}{2} = 6$$

Using Heron's formula, the area A of triangle ABC is:

$$A = \sqrt{6(6-5)(6-3)(6-4)} = \sqrt{6 \cdot 1 \cdot 3 \cdot 2} = \sqrt{36} = 6$$

Therefore, the area of triangle ABC is 6 square units.

Table 10: A comparison of ViFT with other baseline LVLMs (part 3).

Qwen2-VL-7B:
The lengths of the sides of triangle ABC are equal to the sums of the radii of the circles that touch at those points. Therefore, the lengths of the sides of triangle ABC are 3+2=5, 2+1=3, and 3+1=4. The semi-perimeter of triangle ABC is (5+3+4)/2=6. Therefore, the area of triangle ABC is $\sqrt{(6(6-5)(6-3)(6-4))} = 3\sqrt{5}$. Therefore, the answer is $3\sqrt{5}$.

LLaVA-OneVision-7B:
The answer is 4.

ViFT-7B:
To find the area of triangle ABC, we need to determine the lengths of its sides and then use Heron's formula or another suitable method.
Given: - Circle A has a radius of 3 units. - Circle B has a radius of 2 units. - Circle C has a radius of 1 unit.
Since the circles touch each other externally, the distance between the centers of any two circles is equal to the sum of their radii.
1. **Distance between centers A and B**:

$$AB = r_A + r_B = 3 + 2 = 5 \text{ units}$$

2. **Distance between centers A and C**:

$$AC = r_A + r_C = 3 + 1 = 4 \text{ units}$$

3. **Distance between centers B and C**:

$$BC = r_B + r_C = 2 + 1 = 3 \text{ units}$$

Now, we have the side lengths of triangle ABC: - $AB = 5$ units - $AC = 4$ units - $BC = 3$ units
We can use Heron's formula to calculate the area of triangle ABC. Heron's formula states that the area K of a triangle with sides of lengths a, b, and c is given by:

$$K = \sqrt{s(s-a)(s-b)(s-c)}$$

where s is the semi-perimeter of the triangle:

$$s = \frac{a+b+c}{2}$$

For our triangle:

$$s = \frac{5+4+3}{2} = \frac{12}{2} = 6 \text{ units}$$

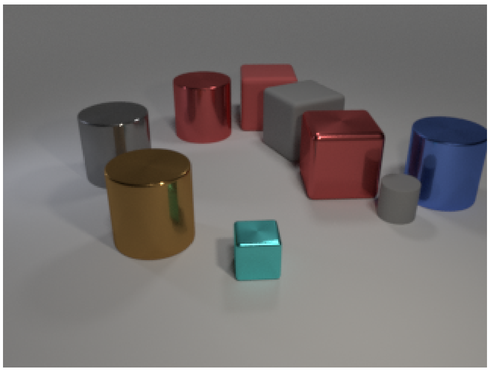
Substituting the values into Heron's formula:

$$K = \sqrt{6(6-5)(6-4)(6-3)} = \sqrt{6 \cdot 1 \cdot 2 \cdot 3} = \sqrt{36} = 6 \text{ square units}$$

Therefore, the area of triangle ABC is:

$$\boxed{6}$$

Table 10: A comparison of ViFT with other baseline LVLMs (part 3).



Problem: Subtract all small cyan cubes. Subtract all large yellow rubber cubes. How many objects are left?
Ground Truth: 8

ViFT With Vision-Text Inputs
The image depicts a collection of various geometric shapes, including cylinders and cubes, arranged on a flat surface. The objects vary in size, color, and material properties such as metallic or matte finishes. Here is a detailed description of the objects present:
1. **Cylinders**: - There are five cylindrical objects in total.
- Two large metallic cylinders: one gray and one red.
- One medium-sized metallic cylinder: gold-colored.
- One small metallic cylinder: blue.
- One small matte cylinder: gray.
2. **Cubes**: - There are four cube-shaped objects.
- One large metallic cube: red.
- One large matte cube: gray.
- One small metallic cube: cyan.
- One small matte cube: cyan.
3. **Arrangement**: - The objects are scattered across the surface without any specific pattern.
- The large metallic cylinders (gray and red) are positioned towards the back left side.
- The medium-sized metallic cylinder (gold) is located near the center-left.
- The small metallic cylinder (blue) is placed towards the right side.
- The small matte cylinder (gray) is situated close to the small metallic cylinder (blue).
- The large metallic cube (red) is positioned towards the front right.
- The large matte cube (gray) is located behind the large metallic cube (red).
- The two small metallic cubes (cyan) are placed in the foreground, with one slightly to the left and the other more centered.
- The two small matte cubes (cyan) are also in the foreground, with one slightly to the left and the other more centered.

ViFT With Text-only Inputs:
To solve this problem, we need to know the initial number of small cyan cubes and large yellow rubber cubes before any subtraction takes place. Since these values are not provided in the question, I will create a hypothetical scenario to illustrate how you would solve this problem.
Let's assume: - There are initially 10 small cyan cubes. - There are initially 5 large yellow rubber cubes. - No other types of objects are mentioned.
Step 1: Subtract all small cyan cubes. - Initial count: 10 small cyan cubes. - After subtraction: 10 - 10 = 0 small cyan cubes.
Step 2: Subtract all large yellow rubber cubes. - Initial count: 5 large yellow rubber cubes. - After subtraction: 5 - 5 = 0 large yellow rubber cubes.
Step 3: Calculate the total number of objects left. - Small cyan cubes left: 0 - Large yellow rubber cubes left: 0 - Total objects left: 0 + 0 = 0
Therefore, the final value is 0.

Table 11: An example of ViFT with or without ability-fused inference (part 1).

Problem: Is the number of big shiny cars in front of the red airliner greater than the number of big purple road bikes? Think first, then answer.
Ground Truth: Yes

ViFT Without Ability-Fused Inference:
In the image, there is one big shiny car located to the right side of the red airliner. There are no big purple road bikes present in the scene. Therefore, the number of big shiny cars (1) is not greater than the number of big purple road bikes (0).
Answer: (B) No

ViFT With Ability-Fused Inference:
To solve this problem, let's break it down step by step:
1. **Identify the objects**: We need to count the number of "big shiny cars" and "big purple road bikes" in relation to the red airliner.
2. **Counting the big shiny cars**: - There is one big shiny car (a purple SUV) in front of the red airliner.
3. **Counting the big purple road bikes**: - There are no big purple road bikes visible in the image.
4. **Comparison**: - The number of big shiny cars in front of the red airliner is 1. - The number of big purple road bikes is 0.
5. **Conclusion**: - Since 1 is greater than 0, the statement "Is the number of big shiny cars in front of the red airliner greater than the number of big purple road bikes?" is true.
Therefore, the correct answer is (A) Yes.

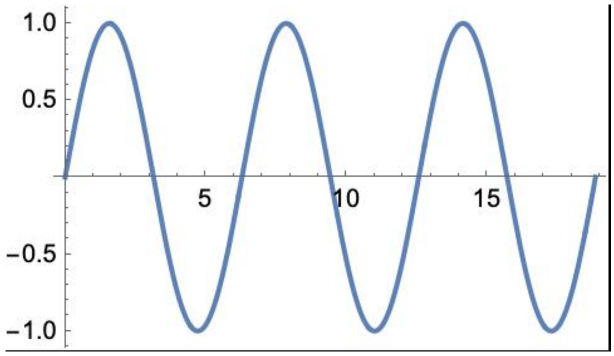
Table 12: An example of ViFT with or without ability-fused inference (part 2).

Problem: What material appears to make up the creature? Think first, then answer.

ViFT Without Ability-Fused Inference:
The image presents a 3D rendering of a creature that is a fusion of a camel and a flame. The creature stands on four legs, each ending in black hooves. Its body is predominantly red, with yellow flames dancing across its surface, giving it an appearance reminiscent of a living fire.
The creature's head is adorned with a pair of glasses, adding a touch of whimsy to its fiery form. The background is a simple gray gradient, which serves to highlight the vibrant colors of the creature. The creature is positioned centrally in the image, drawing the viewer's attention immediately to its unique blend of elements.
There are no discernible texts or other objects in the image. The relative position of the creature to the background suggests it is standing still, with no indication of movement or action. The image does not provide any additional context or narrative, leaving the interpretation open to the viewer.

ViFT With Ability-Fused Inference:
To determine what material appears to make up the creature in the image, let's analyze its characteristics:
1. **Color and Texture**: The creature has a vibrant red body with yellow and orange flames emanating from it. This suggests that the material might be something that glows or emits light, possibly indicating heat or energy.
2. **Shape and Structure**: The creature resembles an animal, specifically a camel, but with exaggerated features like large eyes and a flame-like appearance. Its legs are sturdy and end in black hooves.
3. **Material Properties**: - **Red Body**: The red color could indicate a material like clay, plastic, or even a painted surface. - **Flames**: The presence of flames suggests that the material could be designed to mimic fire or glow-in-the-dark properties. - **Sturdy Legs**: The legs appear solid and robust, which might suggest a material like metal or hard plastic.
4. **Contextual Clues**: Given the fantastical nature of the creature (a flaming camel), it is likely a representation rather than a real-life object. This implies that the material could be anything used for artistic purposes, such as clay, plastic, or even digital rendering.
5. **Lighting and Shadows**: The way the light interacts with the creature suggests a smooth, possibly glossy surface, which could be indicative of materials like plastic or polished stone.
Considering these points, the most plausible material would be a combination of clay or plastic for the body, with additional elements like paint or lighting effects to create the flame effect. The sturdy legs could be made of a different material, perhaps metal or hard plastic, to provide stability.
In conclusion, while the exact material cannot be definitively determined without further information, the creature seems to be crafted from a combination of clay/plastic for the body and possibly metal or hard plastic for the legs, with added visual effects to simulate flames.

Table 13: An example of ViFT with or without ability-fused inference (part 3).



Problem: What is this function most likely be? Think first, then answer.
Ground Truth: B. a trigonometric function

$\alpha = 0.8$**:**
However, without the actual graph or additional context, I cannot provide a specific identification of the function. The options given (polynomial, trigonometric, exponential, or logarithmic) suggest different types of functions that could fit the description. If you have more details about the graph or the function itself, please provide them so I can give a more accurate identification.

$\alpha = 0.9$**:**
To determine which type of function the given graph represents, let's analyze its characteristics:
1. **Shape**: The graph shows a smooth, periodic wave-like pattern with peaks and troughs that repeat at regular intervals.
2. **Amplitude**: The peaks and troughs appear to reach up to 1 unit above and below the x-axis.
3. **Periodicity**: The pattern repeats every 5 units along the x-axis.
Given these observations, we can infer the following:
- **Polynomial Functions**: These typically have a more complex shape with multiple turning points (peaks and troughs), but they do not exhibit the repetitive nature seen in the graph.
- **Trigonometric Functions**: These functions, such as sine or cosine, are known for their periodic behavior, which matches the observed pattern in the graph.
- **Exponential Functions**: These functions grow or decay exponentially and do not form a repeating pattern like the one shown.
- **Logarithmic Functions**: These functions also do not exhibit the periodicity seen in the graph.
Based on the analysis, the most likely function represented by the graph is a trigonometric function.
Therefore, the correct answer is (B) a trigonometric function.

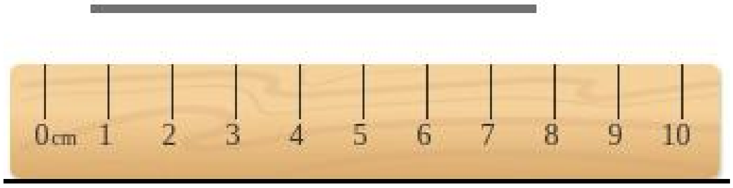
$\alpha = 1.0$**:**
To determine what type of function the given graph represents, let's analyze its characteristics:
1. **Shape**: The graph shows a smooth, periodic wave-like pattern with peaks and troughs that repeat at regular intervals.
2. **Amplitude**: The peaks and troughs appear to reach approximately ±1 on the y-axis.
3. **Periodicity**: The pattern repeats every 5 units along the x-axis, indicating a period of 5.
Given these observations, we can infer that the function is likely a sine or cosine function, which are fundamental trigonometric functions known for their wave-like behavior.
- **Polynomial Functions**: These typically have a smooth curve but do not exhibit the repetitive nature seen in the graph.
- **Exponential Functions**: These usually show rapid growth or decay and do not form a repeating pattern.
- **Logarithmic Functions**: These also do not exhibit the periodicity observed in the graph.
Therefore, based on the shape and periodic nature of the graph, the most likely function is a trigonometric function.
Answer: (B) a trigonometric function

Table 14: An example of ViFT with different α for ability fusion.

$\alpha = 1.1$**:**
The given graph depicts a periodic oscillation with a sinusoidal pattern. The function appears to be a sine or cosine wave, which are fundamental trigonometric functions. These functions are characterized by their smooth, repetitive curves that rise and fall in a regular manner.
To determine the specific type of trigonometric function, we can analyze the shape and properties of the curve:
1. **Periodicity**: The graph shows a repeating pattern every 5 units along the x-axis. This indicates that the period of the function is 5 units.
2. **Amplitude**: The maximum value of the function is 1, and it reaches this peak twice within each period (once at the top and once at the bottom).
3. **Shape**: The curve starts at zero, rises to its maximum positive value, then decreases back to zero, continues to its minimum negative value, and finally returns to zero before starting another cycle.
Given these characteristics, the function is best described as a trigonometric function, specifically a sine or cosine function. Since the graph starts at the origin and reaches its first peak at $x = 5$, it suggests that the function could be a sine function shifted horizontally.
Therefore, the most likely function represented in the graph is a trigonometric function.
Answer: (B)

$\alpha = 1.2$**:**
The given graph depicts a periodic oscillation with a sinusoidal pattern. The function appears to be a sine or cosine wave, which are fundamental trigonometric functions. These functions are characterized by their smooth, repetitive curves that rise and fall in a regular manner.
To determine the specific type of trigonometric function, we can analyze the shape and properties of the curve:
1. **Periodicity**: The graph shows a repeating pattern every 5 units along the x-axis. This indicates that the period of the function is 5 units.
2. **Amplitude**: The maximum value of the function is 1, and it reaches this peak twice within each period (once at the top and once at the bottom).
3. **Shape**: The curve starts at zero, rises to its maximum positive value, then decreases back to zero, continues to its minimum negative value, and finally returns to zero before starting another cycle.
Given these characteristics, the function is best described as a trigonometric function, specifically a sine or cosine function. Since the graph starts at the origin and reaches its first peak at $x = 5$, it suggests that the function could be a sine function shifted horizontally.
Therefore, the most likely function represented in the graph is a trigonometric function.
Answer: (B)

Table 14: An example of ViFT with different α for ability fusion.



Problem: Move the ruler to measure the length of the line to the nearest centimeter. How long is the line? Think first, then answer.
Ground Truth: 7

$\beta = 0$**:**
The image shows a ruler with measurements in centimeters, ranging from 0 to 10 cm. There is a black line above the ruler that spans from approximately 2 cm to just past the 6 cm mark. Therefore, the length of the line is about 5 centimeters.

$\beta = 0.1$**:**
To determine the length of the line using the ruler provided, we need to follow these steps:
1. **Identify the Scale**: The ruler has markings from 0 cm to 10 cm.
2. **Align the Line with the Ruler**: Place one end of the line at the 0 cm mark on the ruler.
3. **Measure the Length**: Extend the line along the ruler until it reaches the other end.
From the image, the line starts at the 0 cm mark and extends to just past the 7 cm mark but not reaching the 8 cm mark. Therefore, the line measures approximately 7 centimeters.
Answer: The line is about 7 centimeters long.

$\beta = 0.2$**:**
To determine the length of the line using the ruler provided, we need to follow these steps:
1. **Identify the Scale**: The ruler has markings from 0 cm to 10 cm.
2. **Align the Line**: Place the ruler such that one end of the line aligns with the 0 cm mark on the ruler.
3. **Measure the Length**: Extend the ruler along the line until it reaches the other end of the line.
Given the hint to provide an integer answer, let's assume the line extends to just past the 7 cm mark but not reaching 8 cm. Therefore, the length of the line would be approximately 7 centimeters.
Answer: 7

$\beta = 0.3$**:**
To determine the length of the line using the ruler provided, we need to follow these steps:
1. **Identify the Scale**: The ruler has markings from 0 cm to 10 cm.
2. **Align the Line**: Place the ruler such that one end of the line aligns with the 0 cm mark on the ruler.
3. **Measure the Length**: Extend the ruler along the line until it reaches the other end of the line.
Given the hint to provide an integer answer, let's assume the line extends to just past the 7 cm mark but not reaching 8 cm. Therefore, the length of the line would be approximately 7 centimeters.
Answer: 7

$\beta = 0.4$**:**
!!

Table 15: An example of ViFT with different β for ability fusion.