TinyGPT-V: Efficient Multimodal Large Language Model via Small Backbones

Zhengqing Yuan
Anhui Polytechnic University
Wuhu, China
zhengqingyuan@ieee.org

Zhaoxu Li
Nanyang Technological University
Singapore
zhaoxu.li@ntu.edu.sg

Lichao Sun
Lehigh University
Bethlehem, USA
lis221@lehigh.edu

Abstract

In the era of advanced multimodel learning, multimodal large language models (MLLMs) such as GPT-4V have made remarkable strides towards bridging language and visual elements. However, the closed-source nature and considerable computational demand present notable challenges for universal usage and modifications. This is where open-source MLLMs like LLaVA and MiniGPT-4 come in, presenting groundbreaking achievements across tasks. Despite these accomplishments, computational efficiency remains an unresolved issue, as these models, like LLaVA-v1.5-13B, require substantial resources. Addressing these issues, we introduce TinyGPT-V, a new-wave model marrying impressive performance with commonplace computational capacity. It stands out by requiring merely a 24G GPU for training and an 8G GPU or CPU for inference. Built upon Phi-2, TinyGPT-V couples an effective language backbone with pre-trained vision modules from BLIP-2 or CLIP. TinyGPT-V's 2.8B parameters can undergo a unique quantisation process, suitable for local deployment and inference tasks on 8G various devices. Our work fosters further developments for designing cost-effective, efficient, and high-performing MLLMs, expanding their applicability in a broad array of real-world scenarios. Furthermore this paper proposed a new paradigm of Multimodal Large Language Model via small backbones. Our code and training weights are placed at: https://github.com/DLYuanGod/TinyGPT-V and https://huggingface.co/Tyrannosaurus/TinyGPT-V respectively.

1 Introduction

Recently, with the advent of GPT-4V, an expansive multimodal large language model (MLLM), we've seen some impressive capabilities in vision-language understanding and generation [45]. That being said, it's pivotal to acknowledge that GPT-4V hasn't been released open-source, thus restricting universal usage and independent modifications. On the bright side, there has been a recent surge in open-source MLLMs, such as LLaVA and MiniGPT-4, which have demonstrated groundbreaking prowess in some tasks, surpassing GPT-4V in areas like image captioning (IC), visual question answering (VQA), and referring expression comprehension (REC) [8, 26, 27, 50]. For example, when put to the test on various visual grounding and question answering tasks, MiniGPT-v2 [6] emerges as a superior force compared to other conventional vision-language models.

Regardless of the substantial vision language capabilities exhibited by some open-source MLLMs, they still consume an excessive amount of computational resources during both training and inference stages. For example, LLaVA-v1.5-13B [26] used 8 A100 GPUs with 80GB memory over the course of 25.5 hours of training. Because the performance of large language models directly impacts the capabilities of MLLMs, their usage, such as LLaVA-v1.5-13B utilizing Vicuna-13b-v1.5 [49], and MiniGPT-v2 leveraging LLaMA2-7B-Chat [41], necessitates a substantial number of parameters of large language models to enhance performance in complex tasks like IC, VQA etc [50]. Therefore,

Preprint. Work in progress.

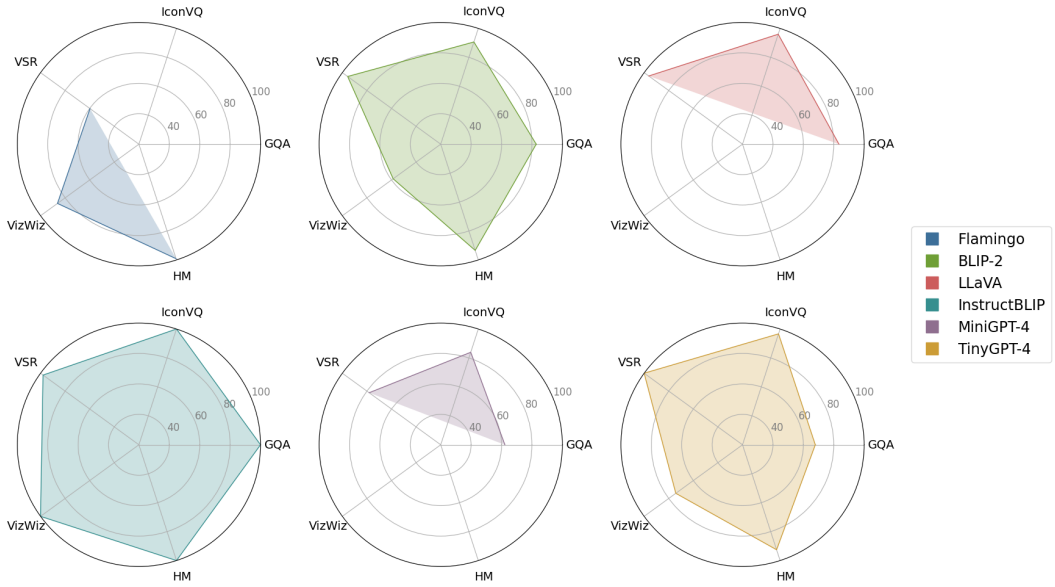


Figure 1: Compared to other general-purpose MLLMs, our TinyGPT-V achieves the same performance as 13B or 7B models in a variety of visual language tasks.

we require a large language model that can rival the performance of models such as LLaMA2 and Vicuna-v1.5, without the need for excessive GPU computational resources.

So a new model called TinyGPT-V is proposed which requires just 24G of GPU for training and only 8G of GPU or CPU for inference. It utilizes an advanced large language model, Phi-2 [19], which is constructed upon Phi [24], and reported to have outperformed the best effects of the 13B language models and it demonstrated similar or better results than models that are up to 25 times larger in scale. Regarding visual perception, we make use of the same pre-trained vision modules from BLIP-2 [23] or CLIP [35], which incorporates a ViT [10] as vision encoder as well as a mapping module. Following the training approach of MiniGPT, TinyGPT-V only applies the mapping module between the visual encoder and language model throughout the training process, while freezing all other parameters. TinyGPT-V uses the same dataset as MiniGPT-v2 in various stages of training such as LAION [37], Conceptual Captions [4, 39], SBU [33] etc. [25, 38, 18, 21, 29, 13, 31, 20, 46].

In our study, we observed that TinyGPT-V displays many qualities that mirror those present in GPT-4, benefiting greatly from the application of the Phi-2 model. Boasting just 2.8B parameters, TinyGPT-V's unique quantisation process makes it suitable for local deployment and inference tasks on 8G mobile devices. TinyGPT-V represents a substantial stride in reaching the equilibrium between unparalleled performance and maintaining the efficiency of MLLMs. Through our contributions, we strive to empower the community to engineer more cost-effective, efficient, and high-performance MLLMs for broad, real-world application scenarios.

2 Related Work

Advanced language model. The evolution of language models has been marked by significant milestones, starting with early successes like GPT2 [36] and BERT [9] in natural language processing (NLP). These foundational models set the stage for the subsequent development of vastly larger language models, encompassing hundreds of billions of parameters. This dramatic increase in scale has led to the emergence of advanced capabilities as seen in models like GPT-3 [2], Chinchilla [16], OPT [48], and BLOOM [44].These large language models (LLMs) have been instrumental in further advancements in the field. For instance, ChatGPT [32] and InstructGPT [34] leverage these powerful models to answer diverse questions and perform complex tasks such as coding. The introduction of open-source LLMs like LLaMA [41] has further propelled research in this area, inspiring subsequent developments like Alpaca [40], Vicuna [7]. These models fine-tune the LLaMA model with additional high-quality instruction datasets, showcasing the versatility and adaptability

of LLM frameworks.Among the most notable recent advancements are Phi [24] and its successor, Phi-2 [19]. These models have demonstrated exceptional performance, rivaling or even surpassing models up to 25 times larger in scale. This indicates a significant shift in the landscape of language modeling, emphasizing efficiency and effectiveness without necessarily relying on sheer size. Such developments mark a new era in the field of NLP, where smaller, more efficient models can achieve results comparable to their much larger counterparts, opening up new possibilities for application and research.

Multimodal language model. In recent years, the trend of aligning visual input to large language models for vision-language tasks has gained significant attention [5, 42, 1, 23, 28, 26, 50, 6]. Seminal works like VisualGPT [5] and Frozen [42], which utilized pre-trained language models for image captioning and visual question answering. This approach was further advanced by models such as Flamingo [1], which incorporated gated cross-attention mechanisms to align pre-trained vision encoders and language models, training on vast image-text pairs. BLIP-2 [23] introduced an efficient Q-Former for aligning visual and language modalities. These groundbreaking studies have paved the way for further innovations in the field, leading to the development of models like LLaVA [28] and MiniGPT4 [50], and their subsequent iterations, LLaVA-v1.5 [26], MiniGPT-v2 [6], ArtGPT-4 [47], instruction GPT-4 [43] and Instruction Mining [3]. These models have demonstrated advanced multimodal capabilities through instruction tuning, showcasing remarkable generalization abilities. Despite their powerful capabilities for visual-language tasks, these multimodal language models often require substantial computational resources. In contrast, TinyGPT-V represents a paradigm shift, harnessing a cost-effective and powerful small language model to achieve a robust, easily deployable model suitable for a variety of real-world vision-language applications. This approach underscores a move towards more efficient yet equally competent multimodal language modeling.

3 Method

We commence by proposing our vision-language model, TinyGPT-V, then conduct a discussion on the structure of the model and the organization of tasks, and finally introduce the training process for each stage.

3.1 Model Architecture

In this subsection, we present the architecture of TinyGPT-V, which consists of a visual coder Linear projection layer and a large language model.

Visual encoder backbone. EVA [11] of ViT serves as the visual foundation model in the TinyGPT-V adaptation as same as MiniGPT-v2. The visual foundation remains inactive throughout entire model training. Our model training operates at a picture resolution of 224x224 for Stage 1, 2 and 3 and 448x448 for Stage 4, and we amplify the positional encoding to scale up with an elevated image resolution.

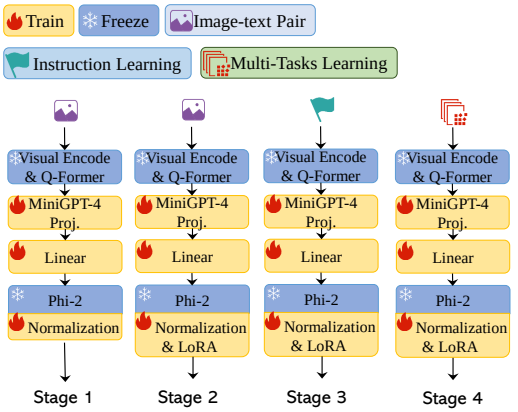
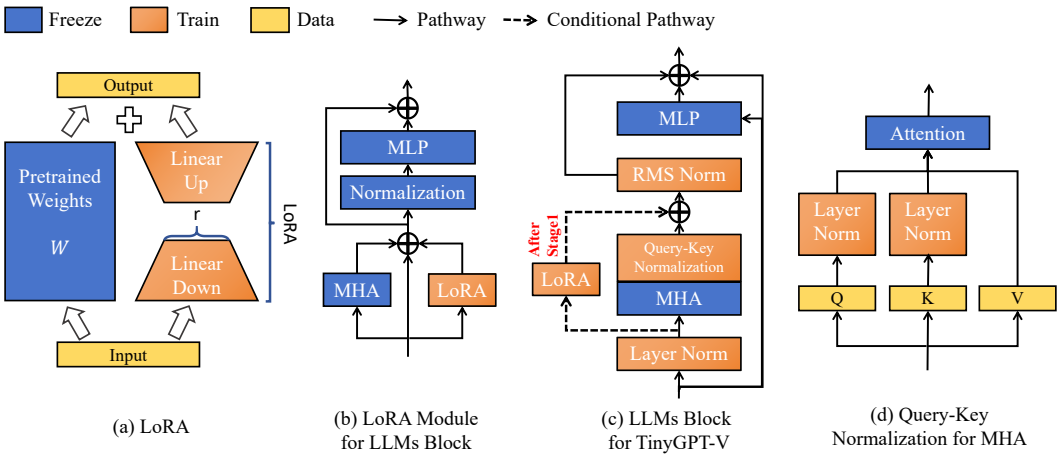


Figure 2: The training process of TinyGPT-V, the first stage is warm-up training, the second stage is pre-training, the third stage is instruction fine-tuning, and the fourth stage is multi-task learning.

Linear projection layers. The function of the Linear Projection layer is to embed the visual features extracted by the visual encoder into the language model. As well as making an effort to empower extensive language models to comprehend image-based information. Our employment of the Q-Former layer, sourced from the BLIP-2 [23] architecture, as the initial linear projection layer is driven by the goal to extract maximum functionality from the pre-trained BLIP system when deployed in visual language models. This approach substantially decreases the volume of parameters that require training stages. We utilize linear projection layers initialized with a Gaussian distribution as the



(a) LoRA (b) LoRA Module for LLMs Block (c) LLMs Block for TinyGPT-V (d) Query-Key Normalization for MHA

Figure 3: (a) represents the structure of LoRA, (b) represents how LoRA can efficiently fine-tune large language models (LLMs) in natural language processing, (c) represents the structure of LLMs for TinyGPT-V, and (d) represents the structure of QK Normalization.

second layer. The goal is to bridge the dimensionality gap between the Q-Former output and the language model's embedding layer, thus, better aligning the visual tokens with the relevant hidden space of the language model. As illustrated in Figure 2, to accelerate the training process of TinyGPT-V, we initially employ the pre-trained Linear projection from MiniGPT-4 (Vicuna 7B) as the foundational layer. Subsequently, we integrate an additional Linear layer projection to effectively bridge into the corresponding hidden space of the Phi-2 model.

Large lanuguage model backbone. We utilize the Phi-2 [19] model as the backbone for our TinyGPT-V large language model. Phi-2 is a 2.7 billion-parameter language model with excellent reasoning and language comprehension, demonstrating state-of-the-art performance among base language models with fewer than 13 billion parameters. In complex benchmarks, Phi-2 matches or outperforms most models 25 times larger. We rely squarely on Phi-2 linguistic tokens to execute several vision language operations. For vision anchoring assignments that require the creation of spatial locations, we explicitly request the linguistic model to generate textual depictions of bounding boxes to signify their geographical coordinates.

Normalization and LoRA for TinyGPT-V. In Section 4.3, we deduce that training smaller large-scale language models for transfer learning, particularly across different modalities (such as from text to image), presents significant challenges. Our investigations reveal that smaller models are particularly susceptible to NaN or INF values during multimodal data computation. This often results in a computational loss value of NaN, causing the initial batch forward propagation to fail. Additionally, a limited number of trainable parameters in these smaller models can contribute to gradient vanishing throughout the training process. To address these issues, as show in Figure 3 (c), we integrate the LLaMA-2 post-norm and input norm mechanisms, implementing RMS Norm after each Multi Head Attention Layer (MHA) to normalize the data for subsequent layers. We also update all Layer norms in the Phi-2 model to enhance training stability, as illustrated in the equation below.

$$\text{LayerNorm}_{input}(x_{hidden}) = \gamma \frac{x_{hidden} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{1}$$

Where, x_{hidden} is the input of this layer, μ and σ^2 are the mean and variance of the inputs to the layer, respectively, ϵ is a small number to prevent division by zero, γ and β are trainable parameters.

$$\text{RMSNorm}(x_{post}) = \frac{x_{post}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2 + \epsilon}} \tag{2}$$

where x_{post} is the input after MHA, N is the number of elements in the vector, and ϵ is a small constant added for numerical stability.

Furthermore, Henry et al. [15] have underscored the vital role of Query-Key Normalization in low-resource learning scenarios. Hence, as show in Figure 3 (d), we have incorporated Query-Key Normalization into the Phi-2 model, as detailed in the following equation.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{\text{LayerNorm}(Q)\text{LayerNorm}(K)^T}{\sqrt{d_k}}\right)V \tag{3}$$

where d_k denotes the dimension of Q or K.

The structure of the LoRA mechanism [17] is show in Figure 3 (a), which is an efficient fine-tuning method in parallel to the frozen pre-training weights as shown in Figure 3 (c), which does not increase the inference time consuming for large language models and is easier to optimise.

3.2 Multi-task Instruction Template

In order to mitigate the potential ambiguity when training a unified multi-modal model to handle various distinct tasks such as visual question answering, image captioning, referring expression comprehension, generation, and object parsing and grounding, we have used the MiniGPT-v2 tokens of task-specific within a multi-task instruction template. It is originating from the LLaMA-2 [41] conversation template and includes a general input format consisting of image features, a task identifier token, and an instruction input. It have six distinct task identifiers, each correlated to a specific task. For tasks requiring the model to identify spatial locations of referred objects, It utilize textual formatting of bounding boxes, with coordinates normalized within the range of 0 to 100. Overall, the unique task-specific tokens provided by MiniGPT-v2 facilitate disambiguation among tasks, allowing for more precise and accurate task execution.

3.3 Training Stages

In this subsection the three-stage training process of TinyGPT-V will be described.

Table 1: The full list of datasets used by TinyGPT-V during training.

Data types	Dataset	Stage 1	Stage 2	Stage 3	Stage 4
Image-text pair	LAION, CC3M, SBU	✓	✓	✗	✗
Instruction tuning	MiniGPT-4 Stage2 for CC & SBU	✗	✗	✓	✗
Caption	Text Captions	✗	✗	✗	✓
REC	RefCOCO, RefCOCO+, RefCOCOg, Visual Genome	✗	✗	✗	✓
VQA	GQA, VQAv2, OK-VQA, AOK-VQA	✗	✗	✗	✓
Multimodal instruction	LLaVA dataset, Flickr30k, Multi-task conversation	✗	✗	✗	✓
Langauge dataset	Unnatural Instructions	✗	✗	✗	✓

Warm-up training for the first training stage. During the initial pretraining stage, TinyGPT-V is taught vision-language understanding using a large library of aligned image-text pairs. The model identifies the output from the introduced projection layers as a soft prompt directing it to create relevant texts and to allow large language models to accept inputs from the image modality. The pretraining process uses a dataset combination of Conceptual Caption, SBU and LAION, involving 20000 training steps covering about 5 million image-text pairs.

Pre-training for the second training stage. Following the initial training phase, the large language model becomes equipped to process image modality inputs. To guarantee more consistent performance as the model transitions into the subsequent training stage, we re-employ the dataset from the first stage, specifically for training the LoRA module.

Human-like learning for the third training stage. We fine-tuned this TinyGPT-V model using a selection of image-text pairings from MiniGPT4 or LLaVA, which included instructions like "*###Human: <Img><ImageHere></Img> Take a look at this image and describe what you notice.###Assistant:*.". We used a uniform template inclusive of a randomly chosen prompt that improved the model's capacity for generating responses that were consistent and sounded more natural.

Multi-task learning in the fourth training stage. The fourth training stage of TinyGPT-V focuses on enhancing its conversation ability as a chatbot by tuning the model with more multi-modal instruction

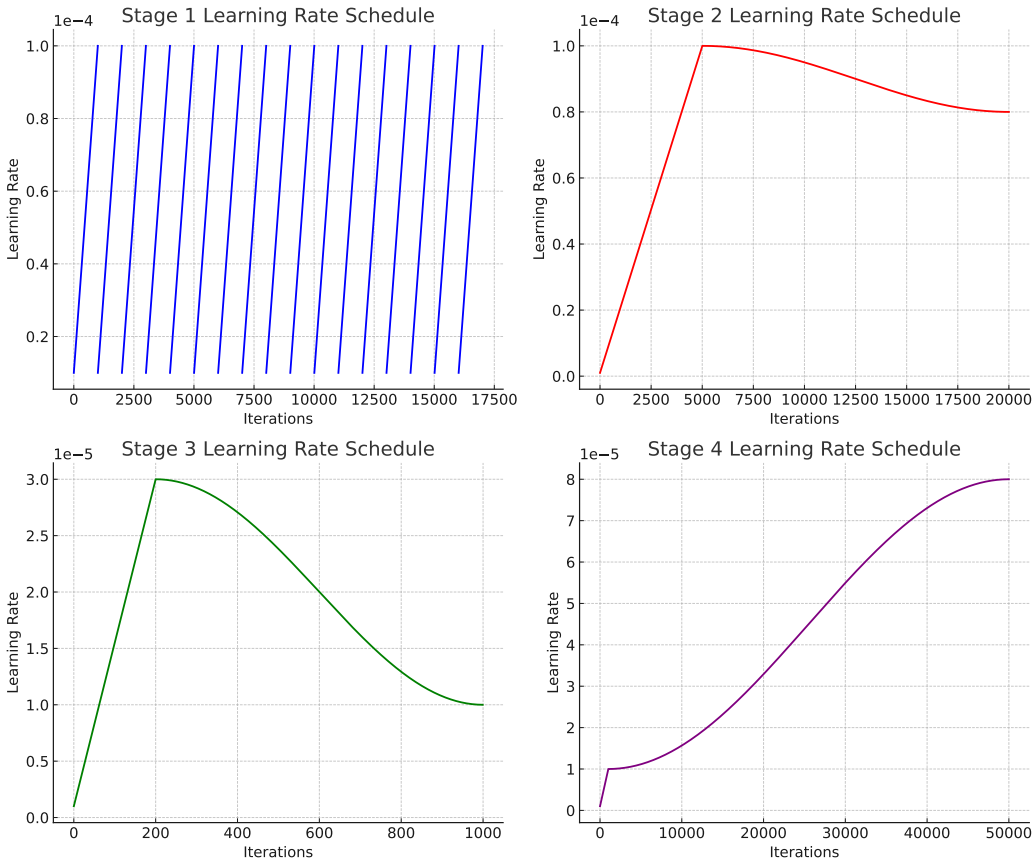


Figure 4: Changes in learning rate during the training stage of TinyGPT-V.

datasets as same as MiniGPT-v2. These datasets, as shown in Table 1, include LLaVA, Flickr30k, a mixing multi-task dataset, and Unnatural Instruction. The LLaVA dataset is utilized for multi-modal instruction tuning with detailed descriptions and complex reasoning examples. The Flickr30k dataset is used to improve grounded image caption generation and object parsing and grounding capabilities. Additionally, a mixing multi-task dataset is created to improve the model's handling of multiple tasks during multi-round conversations. Finally, to recover the language generation ability, the Unnatural Instruction dataset is added to the third-stage training of TinyGPT-V.

4 Experiments

In this section, we describe the training and evaluation methods in detail.

4.1 Training

Experimental setting. The experimental environment for this study was established with a single NVIDIA RTX 3090 GPU, equipped with a substantial 24GB of VRAM. The central processing was handled by an AMD EPYC 7552 48-Core Processor, offering 15 virtual CPUs. Memory allocation was set at 80GB, ensuring sufficient capacity for handling large datasets. The software environment was standardized on PyTorch version 2.0.0, with CUDA 11.8 support, facilitating optimized tensor operations on the GPU.

Training process. In our experimental process, we meticulously orchestrated the training of our model through four distinct stages, each characterized by specific learning rate strategies and loss profiles, as shown in Figure 4 and Figure 5.

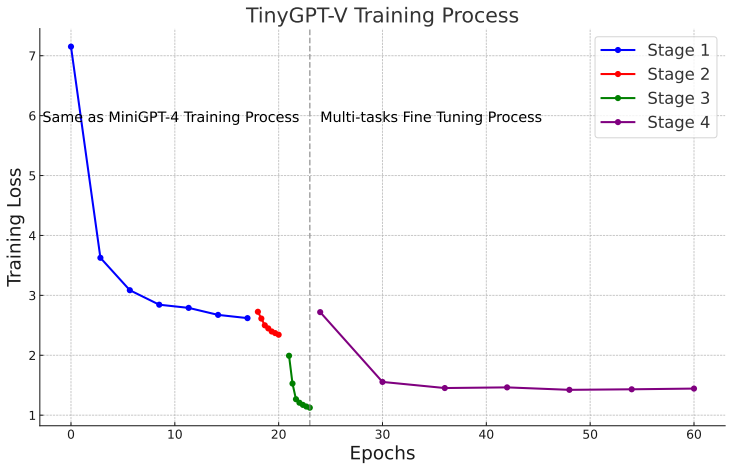


Figure 5: Changes in loss during the training stage of TinyGPT-V.

Stage 1: Spanning 17 epochs, with each epoch consisting of 1000 iterations, we employed a dynamic learning rate approach. The learning rate commenced at 1e-5 at the beginning of each epoch and gradually ascended to 1e-4 by the epoch's end. This pattern was consistently applied across all 17 epochs. The training loss exhibited a steady decline, starting from 7.152 and progressively tapering down to 2.620, reflecting the model's increasing proficiency in learning from the data. The purpose of this stage is to be able to make the Phi-2 model in TinyGPT-V react in some way to the input of the image modality. The alignment of text and image in the semantic space is done.

Stage 2: Comprising 4 epochs, each with 5000 iterations, this stage introduced the "linear_warmup_cosine_lr" [14, 12] learning rate schedule. We initiated a warmup phase of 5000 steps, where the learning rate linearly increased from 1e-6 (warmup_lr) to 1e-4 (init_lr), followed by a cosine decay down to a minimum learning rate of 8e-5. This phase saw a consistent reduction in loss, starting at 2.726 and culminating at 2.343. The purpose of this stage is to enable the LoRA module to play a role in multimodal data, further reducing the model's loss on image-text pairs and improving the model's ability to learn from the data.

Stage 3: This stage lasted for 5 epochs, each with 200 iterations. We maintained the "linear_warmup_cosine_lr" schedule, with a warmup phase of 200 steps. The learning rate began at 1e-6, ascending to 3e-5 (init_lr), before decaying to 1e-5 (min_lr). The loss values reflected significant improvements, starting at 1.992 and reducing to 1.125. The purpose of this stage is to allow TinyGPT-V to accept both verbal and image modal inputs and produce responses to them. After this stage of training TinyGPT-V has been able to perform most of the image answering tasks.

Stage 4: The final stage stretched over 50 epochs, each comprising 1000 iterations. We adhered to the "linear_warmup_cosine_lr" schedule with a 1000-step warmup phase. The learning rate was initiated at 1e-6, reaching up to 1e-5 (init_lr), and then experiencing a cosine decay to a minimum of 8e-5. The training loss values displayed a consistent downward trajectory, beginning at 2.720 and ultimately reaching as low as 1.399. The purpose of this stage is to allow TinyGPT-V to perform various tasks such as VQA or VSR tasks at the same time, increasing the generalization performance of TinyGPT-V on multimodal tasks.

4.2 Evaluation

Evaluation datasets. GQA [18] is a dataset for real-world visual reasoning and compositional question answering, featuring a powerful question engine that generates 22 million diverse reasoning questions. VSR [28] comprises over 10k natural text-image pairs in English, encompassing 66 types of spatial relations. IconQA [30] with 107,439 questions aimed at challenging visual understanding and reasoning in the context of icon images, encompassing three sub-tasks (multi-image-choice, multi-text-choice, and filling-in-the-blank). VizWiz [13] is a collection of more than 31,000 visual queries, each derived from a photo taken by a visually impaired individual using a smartphone, accompanied by a vocalized question regarding the image, and supplemented with 10 answers

Table 2: Comparative performance of TinyGPT-V and other MLLMs across multiple visual question answering benchmarks.

Method	Parameters	Grounding	GQA	VSR (zero-shot)	IconVQ (zero-shot)	VizWiz (zero-shot)	HM (zero-shot)
Flamingo	9B	✗	-	31.8	-	28.8	57.0
BLIP-2	13B	✗	41.0	50.9	40.6	19.6	53.7
LLaVA	13B	✗	41.3	51.2	43.0	-	-
Shikra	13B	✓	-	-	-	-	-
InstructBLIP	13B	✗	**49.5**	52.1	**44.8**	**33.4**	**57.5**
MiniGPT-4	13B	✗	30.8	41.6	37.6	-	-
Ours							
TinyGPT-V	2.8B	✓	33.6	**53.2**	43.3	24.8	53.2

sourced from a crowd for each query. The Hateful Memes dataset (HM) [22], developed by Facebook AI, is a comprehensive multimodal collection specifically designed for the detection of hateful content in memes, combining both image and text elements, and comprises over 10,000 newly created multimodal examples.

Visual question answering results. As shown in Table 2, it becomes evident that TinyGPT-V, a model with only 2.8 billion parameters, exhibits notably competitive performance across multiple benchmarks, closely rivaling models with nearly 13 billion parameters. Specifically, in the VSR (Visual Spatial Reasoning) zero-shot task, TinyGPT-V outshines its counterparts by securing the highest score of 53.2%. This is particularly impressive considering its parameter size is approximately 4.6 times smaller than other leading models such as BLIP-2, LLaVA, and InstructBLIP. In the GQA benchmark, while TinyGPT-V scores 33.6%, it lags behind the highest score achieved by InstructBLIP, which is 49.5%. However, TinyGPT-V shows robust performance in the IconVQ challenge, attaining a score of 43.3%, just 1.5% short of InstructBLIP's leading score of 44.8%. Similarly, in the VizWiz task, TinyGPT-V demonstrates commendable capabilities with a score of 24.8%, which, though not the highest, is notable given its reduced parameter count. In the context of the Hateful Memes (HM) dataset, TinyGPT-V matches InstructBLIP's top score of 57.5% with its own score of 53.2%, again underscoring its efficiency and capacity to compete with models of larger scales. Overall, TinyGPT-V's performance across these diverse and challenging benchmarks is striking, especially when considering its parameter efficiency

4.3 Ablation Study

As shown in Table 3, the full TinyGPT-V model achieves low loss across all stages, but the removal of key modules leads to significant training issues. Without the LoRA module, there's a gradient vanish starting from Stage 3. Omitting Input Layer Norm increases loss notably (to 2.839 in Stage 1) and causes gradient vanishing in Stage 4. Without RMS Norm, the model sees an elevated loss in Stage 1 (2.747) and faces early gradient vanishing in Stage 2. The absence of QK Norm results in immediate gradient vanish. This data clearly illustrates each module's crucial role in preventing gradient vanishing and maintaining low loss throughout the training process.

Table 3: Importance of each module in TinyGPT-V at each stage of training.

Method	Stage 1 Loss	Stage 2 Loss	Stage 3 Loss	Stage 4 Loss
TinyGPT-V	2.620	2.343	1.125	1.444
-LoRA	2.620	-	Gradient Vanish	-
-Input Layer Norm	2.839	2.555	1.344	Gradient Vanish
-RMS Norm	2.747	Gradient Vanish	-	-
-QK Norm	Gradient Vanish	-	-	-

Furthermore, our reveal a notable trend: the smaller the large language model used for transfer learning (particularly in transitioning from text-to-image modality), the more challenging the training process becomes. We observed a pronounced need for additional normalization layers to stabilize the training, especially when scaling down from larger models like Vicuna-13B to smaller ones like Phi-2 (2.7B) and Phi-1.5 (1.3B).

5 Conclusion

In this study, we introduce TinyGPT-V, a parameter-efficient MLLMs tailored for a range of real-world vision-language applications. Our model innovatively builds on the compact yet powerful Phi-2 small language model framework. This approach results in TinyGPT-V delivering exceptional outcomes in diverse benchmarks like visual question-answering and referring expression comprehension while keeping computational demands manageable. Remarkably, TinyGPT-V can be trained on a 24G GPU and deployed on an 8G device, demonstrating a significant advancement in creating cost-effective, efficient, and potent MLLMs. This paper marks a contribution towards crafting smaller, yet robust multimodal language models for practical, real-world use cases. We envision that our work will catalyze further explorations into developing compact MLLMs for diverse applications.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

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

[3] Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: When data mining meets large language model finetuning, 2023.

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts, 2021.

[5] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040, 2022.

[6] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[11] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale, 2022.

[12] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

[13] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[14] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks, 2018.

[15] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers, 2020.

[16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

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

[19] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, Mojan Javaheripi, Piero Kauffmann, Yin Tat Lee, Yuanzhi Li, Anh Nguyen, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Michael Santacroce, Harkirat Singh Behl, Adam Taumann Kalai, Xin Wang, Rachel Ward, Philipp Witte, Cyril Zhang, and Yi Zhang. Phi-2: The surprising power of small language models. https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/, 2023.

[20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[21] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020.

[22] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes, 2021.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[24] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[26] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[29] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

[30] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning, 2022.

[31] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[32] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022.

[33] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

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

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

[38] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.

[39] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[40] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[42] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[43] Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4, 2023.

[44] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[45] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision), 2023.

[46] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.

[47] Zhengqing Yuan, Xinyi Wang, Kun Wang, Lichao Sun, and Yanfang Ye. Artgpt-4: Towards artistic-understanding large vision-language models with enhanced adapter, 2023.

[48] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[49] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

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