Transfer Visual Prompt Generator across LLMs

Ao Zhang [1] **Hao Fei** [1*] **Yuan Yao** [2*] **Wei Ji** [1] **Li Li** [1] **Zhiyuan Liu** [2] **Tat-Seng Chua** [1]
[1] Sea-NExT Joint Lab, School of Computing, National University of Singapore
[2]Department of Computer Science and Technology, Tsinghua University
zhanga6@outlook.com haofei37@nus.edu.sg yaoyuanthu@163.com

Project: https://vpgtrans.github.io

Abstract

While developing a new vision-language LLM (VL-LLM) by pre-training on tremendous image-text pairs from scratch can be exceedingly resource-consuming, connecting an existing LLM with a comparatively lightweight visual prompt generator (VPG) becomes a feasible paradigm. However, further tuning the VPG part of the VL-LLM still suffers from indispensable computational costs, i.e., requiring thousands of GPU hours and millions of training data. One alternative solution is to transfer an existing VPG from any existing VL-LLMs for the target VL-LLM.

In this work, we for the first time investigate the VPG transferability across LLMs, and explore a solution to reduce the cost of VPG transfer. We first study the VPG transfer across different LLM sizes (e.g., small-to-large), and across different LLM types, through which we diagnose the key factors to maximize the transfer efficiency. Based on our observation, we design a two-stage transfer framework named **VPGTrans**, which is simple yet highly effective. Through extensive experiments, we demonstrate that VPGTrans helps significantly speed up the transfer learning process without compromising performance. Remarkably, it helps achieve the VPG transfer from BLIP-2 OPT$_{2.7B}$ to BLIP-2 OPT$_{6.7B}$ with over 10 times speed-up and 10.7% training data compared with connecting a VPG to OPT$_{6.7B}$ from scratch. Further, a series of intriguing findings and potential rationales behind them are provided and discussed. Finally, we showcase the practical value of our VPGTrans approach, by customizing two novel VL-LLMs, including VL-LLaMA and VL-Vicuna, with recently released LLaMA and Vicuna LLMs. With our VPGTrans, one can build a novel high-performance VL-LLM at affordably lower cost.

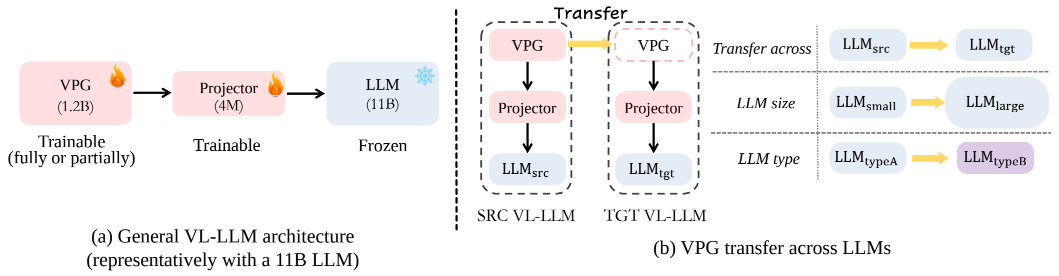


Figure 1: (a) The general architecture of VL-LLMs, *e.g.* , BLIP-2 [18], Flamingo [2] and PaLM-E [11], including a visual prompt generator (VPG), a linear projector and a backbone LLM. To tune the VL-LLM, typically, only the VPG and the projector are updated, while the LLM is kept fully frozen. (b) This work investigates the VPG transferability across LLMs, including different LLM sizes and LLM types.

*Corresponding Author: Hao Fei, Yuan Yao

1 Introduction

Background. Recent years have witnessed a great rise in large-scale language models (LLMs) on ushering the human-like artificial intelligence. By scaling up the model size (*e.g.* , from 11B to 175B), LLMs have been empowered with the amazing capability of human language understanding. Text-based LLMs are further enhanced by associating with other modalities such as vision, leading to the vision-language LLMs (VL-LLMs), *e.g.* , BLIP-2 [18], Flamingo [2], GPT-4 [6] for multimodal dialog system, and PaLM-E [11] for embodied AI system. To construct a VL-LLM, a visual prompt generator (VPG) module (*cf.* Fig. 1) that produces soft prompts for the input images/videos is added[2] at the bottom of the backbone LLM for bridging the gap between vision and language modalities. Currently, such architecture has been frequently adopted by many popular VL-LLMs. For example, BLIP-2 pre-train a CLIP-ViT [23] combined with a Q-Former as VPG; PaLM-E explores several different architectures, *e.g.* ViT [10] or OSRT [25] as VPG. To obtain the final VL-LLM, the VPG needs to be tuned. Without updating the whole architecture on the on-demand data, ideally, the large backbone LLM can be kept fully frozen, with only the comparatively light VPG module being (fully or partially) updated. [3]

Motivation. However, building a VL-LLM is inevitably computational costing. Because even though the backbone LLM is frozen, tuning the light VPG and projector parts still requires additional huge computation resources to cover the LLM load that accounts for the main overhead. For example, tuning the VPG and projector for BLIP2-FlanT5$_{\text{XXL}}$ still needs over 600 A100-GPU hours on over 100 million image-text pairs, where the LLM takes up 70% of the GPU memory consumption. Thus, whenever developing a new VL-LLM with any newly-emerging LLM, it can be prohibitively expensive to train the VPG and projector. Hopefully, transferring a pre-trained VPG (which is the main body of trainable parts) from an existing VL-LLM to a novel LLM instead of training from scratch[4], offers a promising solution for relieving the cost issue. Intuitively, all the VL-LLMs literally can share the same VPG infrastructure and utility,[5] which makes the VPG transferring theoretically feasible. In this work, we thus investigate the potential of transferring VPG across LLMs.

Proposal. Specifically, this paper examines the transferability of VPG across LLMs 1) with different sizes (in the same type), *i.e.* , *transfer across LLM size*, and 2) across different types of LLMs, *i.e.* , *transfer across LLM type*, as illustrated in Fig. 1.

- [*Transfer across LLM size*]. It has been the typical practice for LLM-related research [6] to validate the training strategy and the hyperparameter on smaller models (*e.g.* OPT$_{\text{2.7B}}$) and then scale up to larger ones (*e.g.* OPT$_{\text{6.7B}}$). It thus is worth exploring whether a VPG trained on a smaller LLM can be transferred to a larger LLM, resulting in reduced computational costs & data, and maintaining comparable performance.
- [*Transfer across LLM type*]. With a well-tuned VPG for a type of LLM, it is interesting to see if the VPG can be further transferred to other types of LLMs even with different architectures (*e.g.* decoder *v.s.* encoder-decoder). If the transfer can be achieved, how to make it more efficient?

We conduct a series of exploratory analyses (*cf.* §3.1) to explore these two questions and also identify the key factors for transfer efficiency. Based on our observations, we design a two-stage transfer learning framework (*cf.* §3.2), namely **VPGTrans**, that includes a projector warm-up and direct fine-tuning. For stage-1, we find that the projector warm-up can effectively reduce the training step for adapting a pre-trained VPG to a newly coming LLM, and avoid potential performance drop in the adaptation. To achieve an efficient warm-up, the projector will be initialized with the help of two LLMs' word embedding converter and then trained with an extremely large learning rate. For stage-2, there is a normal fine-tuning of both VPG and projector. Despite its simplicity, VPGTrans is able to significantly speed up the VPG-transfer process without hurting the performance.

Results and Findings. Via extensive experiments on the above two scenarios (*cf.* §4 & §5), we gain the following key observations.

[2]Also including a linear projector for dimension matching.

[3]Note that Flamingo also inserts some tunable parameters into the LLM part, but recent works [18, 11] found that freezing LLM is more efficient.

[4]It is not a rigorous expression, because the VPG is typically a pre-trained model, like CLIP [23]. We use it for simplicity in this paper.

[5]while the projector can not be shared.

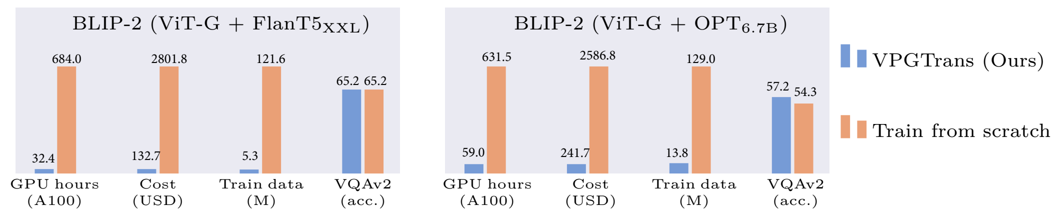


Figure 2: Comparing the cost between *training VPG from scratch* vs. *transferring VPG via our VPGTrans strategy*. Note the LLM via VPGTrans is FlanT5$_{\text{XL}\rightarrow\text{XXL}}$ and OPT$_{2.7\text{B}\rightarrow6.7\text{B}}$, respectively.

- VPGTrans helps achieve comparable or better performance with at most 10 times acceleration for the small-to-large transfer across LLMs in the same type, while also enabling training with less data and resulting in more stable training.
- VPGTrans can also achieve at most 5 times acceleration for the transfers between different model types.
- Notably, our VPGTrans helps achieve a **BLIP-2 ViT-G OPT$_{2.7\text{B}\rightarrow6.7\text{B}}$** transfer with **less than 10% of the GPU hours** and **10.7% training data** required for the original model training.
- Furthermore, our framework can even outperform the original BLIP-2 OPT$_{6.7\text{B}}$ on most of the datasets, with a **+2.9** improvement on VQA(v2) and a **+3.4** improvement on OKVQA.

Our investigation further reveals some intriguing findings, for which we provide possible explanations.

- When transferring across different model sizes: 1) transfer learning from a smaller LLM to a larger LLM does not compromise final performance; and 2) the VPG transfer from a smaller LLM (say A) to a larger LLM (say B) roughly follows a counterintuitive principle of "*the smaller A's size the better*".
- Furthermore, when transferring across different models, effective VPG transfer is only achievable between large LLMs, while small LLMs are not suitable for this purpose using our VPGTrans.

Contributions. In this study, we for the first time show that effective VPG transfer across LLMs can be achieved, suggesting that it is possible to build a new VL-LLM with considerably lower cost, as seen in Fig. 2. To summarize, we make the following key contributions:

- *Effective approach*. We investigate the key factors for VPG-transfer efficiency, and propose a two-stage transfer framework VPGTrans. The approach helps achieve highly-efficient VPG transfer across LLMs meanwhile with less training data and even task improvements.
- *Intriguing findings*. By exploring the VPG transfer across LLMs, we reveal several intriguing and key findings, and shed light on the possible explanations that will enlighten further research in this direction.
- *Open source*. We showcase how to customize a novel GPT-4-like VL-LLM with our VPGTrans (*cf.* §6), and release two multimodal-version VL-LLMs: VL-LLaMA and VL-Vicuna. All codes and models are open at `https://github.com/VPGTrans/VPGTrans`.

2 Preliminary

This section will first give a briefing on the existing prevailing VL-LLMs, and then elaborate on the settings of the exploratory analyses of these VL-LLMs.

2.1 VL-LLM

Architecture. As illustrated in Fig. 1, current VL-LLMs mostly adopt a common architecture, including a visual prompt generator (VPG), a projector, and a backbone LLM. Typically, VPG takes as inputs images/videos, and encodes the visual input into a fixed length of soft prompts. Then, a linear projector is adopted to align the soft prompt's dimension to LLM's word embedding dimension. Finally, the LLM will generate sentences based on the input features passed from the soft prompt. We list some of the recent representative VL-LLMs in Table 1.

Training Paradigm. Given a VL-LLM, typically the VPG and linear projector will be trained, fully or partially. For example, PaLM-E updates all of the parameters of VPG in the pre-training stage,

Table 1: VL-LLMs architectures and pre-training paradigm. †: it is a GPT-2-like LLM with relative position embeddings.

VL-LLMs	**VPG**	**VPG Trainable**	**LLM**	**LLM Trainable**
Frozen [29]	NF-ResNet-50	NF-ResNet-50	GPT-2-like†	No
Flamingo [2]	ViT+Resampler	Resampler	Chinchilla	Xattn-Dense
PaLM-E [11]	ViT/OSRT	All	PaLM	No
BLIP-2 [18]	EVA-CLIP + Q-Former	Q-Former	OPT/Flan-T5	No

while BLIP-2 and Flamingo freeze the ViTs and tune their Q-Former and Resampler, respectively. As the main part of the whole architecture, the LLM is mostly frozen during the training. Flamingo is an exception, which adds a small portion of trainable gated Xattn-Dense layers in its LLM (10B for Flamingo-80B). However, the later works [18, 11] show that freezing all of the LLM can also achieve excellent zero-shot performance but with significantly reduced computational cost, which leads the trend into frozen LLM. For example, BLIP-2 FlanT5$_{\text{XXL}}$ (11B) can achieve better zero-shot VQAv2 performance compared with Flamingo-80B. Thus, in this paper, we mainly focus on VPG transfer across frozen LLMs.

Notations. Formally, given a source VL-LLM to be transferred, we denote its LLM and VPG as LLM_{src} and VPG_{src}, and denote the correspondence parts at the target side as LLM_{tgt} and P_{tgt}.

2.2 Experiment Settings

Architecture. Among all the VL-LLMs in Table 1, currently only BLIP-2 is open-sourced with model access. Thus we adopt BLIP-2's architecture and training paradigm. In our exploration experiments, we consider using the VPG that consists of a CLIP-ViT *large*, and a Q-Former that has already undergone a BLIP-like pre-training (the 1st stage pre-training in BLIP-2's paper [18]).

Training Data. For all of the exploration experiments, we adopt a combination of human-annotated COCO caption dataset [19] and web image-text pairs SBU dataset [22], which results in a total of 1.4 million image-text pairs.

Transfer Direction. For the small-to-large model transfer among the same type LLMs, we investigate: 1) OPT (decoder-only) series including 125M, 350M, 1.3B, and 2.7B, and 2) FlanT5 (encoder-decoder) ranging *base, large*, and *XL*. For the transfer across different types of LLMs, we consider the one between OPT and FlanT5 of similar sizes.

Evaluation. To evaluate the performance of VL-LLMs, we choose five benchmark datasets:

- COCO caption [19]. The task requires the model to generate a textual caption describing the input image. Note that the evaluation on COCO is not a standard zero-shot setting, as the training set of COCO caption is also included in our pre-training corpus.
- NoCaps [1]. Different from COCO captions that mainly focus on limited regular visual concepts, NoCaps aims at testing models' ability on generating captions with much more diverse visual concepts.
- VQAv2 [4]. VQAv2 assesses the capability of generating open-ended answers based on the visual facts in the given image. Due to the extremely large amount of image numbers for VQAv2 validation set, we split a subset with 20,000 questions for fast evaluation.
- GQA [12]. GQA requires multi-hop reasoning based on the visual image inputs. To produce correct answers, a model needs to have a strong scene-level understanding. Similar to VQAv2, GQA is also an open-ended QA dataset.
- OKVQA [21]. Different from VQAv2 and GQA, OKVQA requires not only extracting the information in the visual content but also associating it with the necessary knowledge.

We report the CIDEr for all caption tasks and accuracy for all VQA tasks.

Implementation Details. We follow the same implementation details of BLIP-2, via the open code.[6] Concretely, we use FP16 for OPT and BFloat16 for FlanT5 for the model training. For the learning rate, we first conduct a linear warm-up from 1e-6 to 1e-4, and then use a cosine learning

[6]`https://github.com/salesforce/lavis`

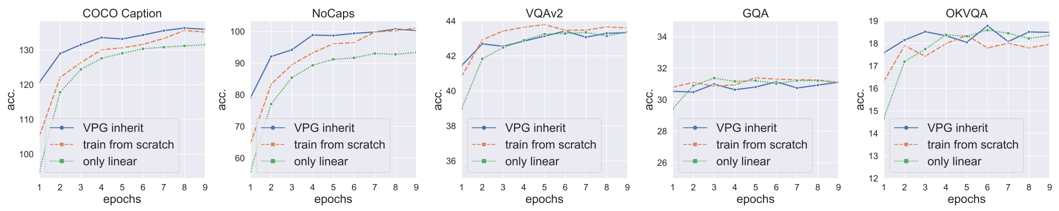


Figure 3: Comparisons between i) inheriting VPG from $\text{OPT}_{125\text{M}}$ and training it with randomly initialized projector for $\text{OPT}_{350\text{M}}$ and ii) training VPG and randomly initialized projector for $\text{OPT}_{350\text{M}}$ from scratch, and iii) training only the linear projector after inheritance.

rate schedule with the minimal lr=1e-5 for 10 epochs. Due to the limited data amount, we slightly decrease the batch size, which we find beneficial for the final performance. Specifically, we set the batch size of 1,728 and 1,152 for OPT and FlanT5-based models, respectively. Gradient accumulation is applied to achieve the target batch size.

3 Maximizing the Transfer Efficiency with a Two-stage Transfer Strategy

In this section, we first identify the key factors for maximizing transfer efficiency, based on which we then motivate our solution for better transfer.

3.1 Exploratory Analysis: Identifying Key Factors for VPG Transfer

Via selected experiments of small-to-large transfer among OPT models, we can obtain the following key observations. Note that more systematical comparisons are conducted in the later section (*cf.* §4).

• **Inheriting the trained VPG can accelerate training.** To demonstrate this, we compare the convergence rates of VPG training on $\text{OPT}_{350\text{M}}$ from scratch, and inheriting VPG trained on $\text{OPT}_{125\text{M}}$. The patterns are shown in Fig. 3. Overall, we find that inheriting VPG trained on $\text{OPT}_{125\text{M}}$ accelerates convergence, particularly for two caption tasks. However, for datasets requiring fine-grained visual perception such as VQAv2 and GQA, inheriting VPG from a smaller model may hinder performance. We hypothesize that tuning VPG with a randomly initialized linear projector will compromise the existing fine-grained visual perception ability of VPG. The possible reason can be that, the VPG is typically a pre-trained model with powerful visual perception ability, and thus updating based on the gradient passed through a random projector will mislead the VPG at the initial steps [2]. Fortunately, this problem can be resolved by warming up the linear projector prior to the joint tuning of VPG and the projector (*cf.* 3.1). Note that this hypothesis may also be applicable for training from scratch, but we mainly focus on the VPG transfer in this paper.

• **Warming up the linear projector can prevent performance drop and expedite VPG training.** To verify this, we first conduct a warm-up training of the linear projector for 3 epochs, during which both VPG and LLM are frozen. Subsequently, we jointly train VPG and the projector and plot the performance curve in Fig. 4 (the warm-up process is not included in this figure). The results show that the performance drop observed in Fig.3 can be avoided in Fig. 4. Additionally, we observe that the warm-up training leads to fewer training steps required for VPG and projector joint training. However, we must emphasize that warming up is a costly step. In the case of a large LLM, such as 6.7B, the trainable parameters of BLIP-2's VPG will account for less than 10% of the total parameters. In this condition, there is no significant difference of the cost between training VPG+projector and only projector. We will elaborate on how to accelerate the linear projector warm-up in our later discussion (*cf.* 3.1).

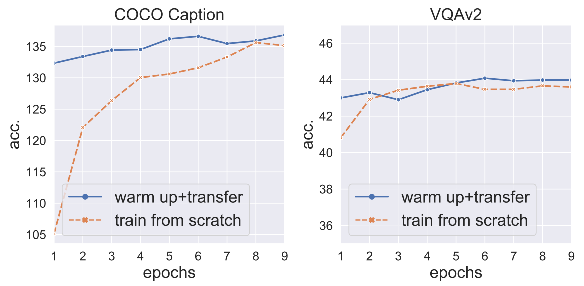


Figure 4: First warming-up then transferring can avoid performance drop on VQAv2 and accelerate convergence for COCO caption.

• **Merely tuning the projector can not achieve the best performance.** Before coming to the warm-up acceleration, we want to clarify that merely tuning the projector is insufficient for achieving the best performance. Notably, as shown in Fig. 3, significant performance gaps are observed between the "*only linear*" (green curve) and "*train from scratch*" (orange curve) approaches for COCO caption and NoCaps. Therefore, if the goal is to build a multimodal conversation robot using carefully collected dialog data, training only the linear projector is insufficient to align with the provided data.

• **Initializing LLM_{tgt}'s projector with the help of the word converter can accelerate the linear projector warm-up.** In fact, the VPG and projector trained on LLM_{src} have already learned how to map the visual content to LLM_{src}'s understandable soft prompt. If we can convert the LLM_{src}'s soft prompt to LLM_{tgt}'s soft prompt, we can directly merge the converter with the projector trained on OPT_{src}. One natural idea is to leverage the word embeddings of both models as a proxy for the soft prompt. The intuition behind the scene is that, the soft prompt works in the same format as normal sentences *i.e.* , concatenated with word embeddings as input. As demonstrated in Fig. 5, we observe a common pattern where the last token of soft prompts is closest to EOS, while the middle tokens represent the image content. Such phenomenon indicates a similarity between soft prompts and word embeddings.

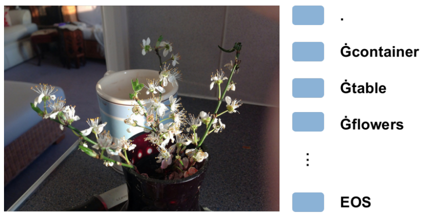


Figure 5: Interpreting generated soft prompts for $\text{OPT}_{125\text{M}}$ with nearest words.

To validate our hypothesis, we conduct experiments in a challenging scenario where we transfer from $\text{OPT}_{125\text{M}}$ to $\text{OPT}_{1.3\text{B}}$. By training a linear word embedding converter with cosine similarity (*cf.* §3.2(b)), we can initialize the projector for $\text{OPT}_{1.3\text{B}}$ by merging the projector for $\text{OPT}_{125\text{M}}$ and converter. We observe that the initialization can accelerate the linear projector warm-up process, as shown in Table 2. The model with word embedding converter initialization converges faster than the one without initialization, and the 2nd epoch's performance of the *w/ init.* approach is almost the same as the third epoch's performance of the *w/o init.* approach.

• **Word embedding converter can not replace a trained linear projector.** While soft prompts share some similarities with word embeddings, they are not identical. For instance, the norm of soft prompts is typically around 10 times the average norm of word embeddings. It is important to note that the merged linear projector cannot replace the warm-up training, as using only the linear projector initialization yields a random performance. We believe that a better understanding of how prompt works will further benefit the VPG's transfer learning.

Table 2: Comparison between linear projector warm-up with/without word embedding initialization. The metric is COCO caption's CIDEr.

Epoch	w/ init.	w/o init.
1	130.2	126.1
2	132.7	131.6
3	133.4	132.8

• **Linear projector warm-up enables fast convergence with an extremely large learning rate.** To determine the most efficient transfer practice, we experiment with training the projector using different learning rates. Surprisingly, we find that the linear projector enables fast convergence with an extremely large learning rate, whereas such a large learning rate will cause a crash for VPG training. Specifically, when we set the learning rate to 5 times the original value, the COCO caption's CIDEr score can reach 133.1 with only 1 epoch training, which is similar to the results in Table 2 obtained after 3 epochs training with the original learning rate. Furthermore, the linear projector is robust to changes in the learning rate. Although further increasing the learning rate to 10 times does not yield any additional acceleration, the linear projector can converge without crashing during training.

3.2 A Two-stage VPG Transfer Framework

By connecting all the dots as discussed above in §3.1, we now design our two-stage VPGTrans framework for more efficient VPG transfer. As shown in Fig. 6, the stage-1 of VPGTrans performs projector warm-up and the stage-2 carries out direct fine-tuning. Our results demonstrate that, our approach is simple yet can significantly speed up the transfer learning process without compromising performance. Some more detailed results are given in the later sections (*cf.* §4 & 5).

▶ **Stage-1: Projector Warm-up.**

(a) Inherit VPG. We first initialize the VPG for LLM_{tgt} with the VPG trained on LLM_{src}.

(b) Projector Initialization. Then, we initialize the projector for LLM_{tgt} with a merged projector from the projector of LLM_{src} and a linear word converter. Formally, we define the linear projector of

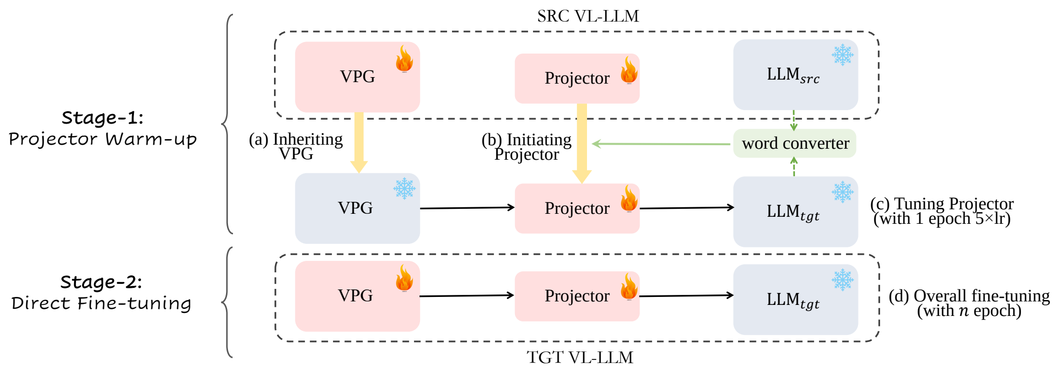


Figure 6: Our two-stage VPGTrans framework. **Stage-1** is to first (a) inherit the VPG of LLM$_{\text{src}}$ and (b) initialize the projector by merging the projector of LLM$_{\text{src}}$ and word converter. (c) Then the projector will be warmed up for 1 epoch with a large learning rate. **Stage-2** is to (d) train the VPG and projector jointly for n epochs.

LLM$_{\text{src}}$ as $f_s(x) = W_s x + b_s$, the linear projector for LLM$_{\text{tgt}}$ as $f_t(x) = W_t x + b_t$, and the word converter as $g_w(x) = W_w x + b_w$.

The word converter is a linear layer trained with text-only caption data to convert from the LLM$_{\text{src}}$'s word embeddings to LLM$_{\text{tgt}}$'s word embeddings. We experiment with optimizing losses based on cosine similarity or Euclidean distance, and observe no significant difference between the two losses. Thus we simply use cosine similarity in our experiments. In cases where LLM$_{\text{src}}$ and LLM$_{\text{tgt}}$ use different tokenization methods, we optimize based on the overlapped tokens. Formally, for every given token k, we denote its word embeddings of LLM$_{\text{src}}$ and LLM$_{\text{tgt}}$ as x_s and x_t. Then, we minimize the loss:

$$l = 1 - sim(g_w(x_s), x_t)\,. \tag{1}$$

Once we obtain the word converter $g_w(\cdot)$, we can easily merge it with the projector of LLM$_{\text{src}}$ as:

$$f_t(x) = f_s(g_w(x)) = W_s(W_w x + b_w) + b_s\,, \tag{2}$$

resulting in f_t's weight and bias as $W_t = W_s W_w$ and $b_t = W_s b_w + b_s$.

(c) Warm-up Training. Then, we only train the projector in this stage with a frozen VPG and LLM. Specifically, we train the projector for 1 epoch with 5 times of learning rate.

► Stage-2: Direct Fine-tuning.

(d) Direct Fine-tuning. In the final step, we conduct joint training of VPG and projector for n epochs with a normal learning rate.

4 Exp-I: Transfer across Different Model Sizes

In this section, we conduct experiments to systematically illustrate the effectiveness of our VPGTrans and analyze the relationship between transfer efficiency and model size. For simplicity, we use **TaS** to represent the transfer across different model sizes.

4.1 Experimental Settings

In this part, we introduce baselines and transfer variants. For details about training data and implementation details, please refer to the experiment settings in the Preliminary (*cf.* 2.2).

Baselines. We mainly compare our VPGTrans with *training from scratch* (TFS) and *VPG inheritance* (VPGInherit), where we report their zero-shot performance on the aforementioned 5 tasks. For our VPGTrans, the word converter training only requires updating a linear layer on tokenized text data and typically takes less than 15 minutes on 1 A100 GPU with less than 15G GPU memory. Meanwhile, the stage-1 projector warm-up is slightly faster than training both VPG and projector due to the frozen VPG. Therefore, we consider the whole stage-1 training as the 1st epoch, where the 1st epoch in Fig. 7 of VPGTrans is actually our stage-1's result.

Table 3: The speed-up rate of our VPGTrans compared with *training from scratch* (TFS). The speed-up rate is the quotient of the epochs used to achieve the *best performance on a specific dataset* (BP) for TFS and the epochs used to surpass BP by VPGTrans. The symbol "-" means VPGTrans can not achieve better performance than BP.

Transfer	COCO Caption	NoCaps	VQAv2	GQA	OKVQA
$\text{OPT}_{125\text{M}\rightarrow350\text{M}}$	1.7	3.0	1.0	5.0	5.0
$\text{OPT}_{125\text{M}\rightarrow1.3\text{B}}$	9.0	10.0	9.0	-	2.0
$\text{OPT}_{350\text{M}\rightarrow1.3\text{B}}$	4.5	5.0	9.0	2.0	2.0
$\text{OPT}_{125\text{M}\rightarrow2.7\text{B}}$	10.0	10.0	2.0	2.0	3.0
$\text{OPT}_{350\text{M}\rightarrow2.7\text{B}}$	10.0	10.0	2.0	-	3.0
$\text{OPT}_{1.3\text{B}\rightarrow2.7\text{B}}$	3.3	3.3	2.0	-	1.5
$\text{FlanT5}_{\text{base}\rightarrow\text{XL}}$	1.0	1.1	3.0	4.0	2.0
$\text{FlanT5}_{\text{XL}} \rightarrow \text{OPT}_{2.7\text{B}}$	5.0	5.0	2.0	2.0	3.0
$\text{OPT}_{2.7\text{B}} \rightarrow \text{FlanT5}_{\text{XL}}$	1.7	2.0	-	2.0	-

Transfer Variants. We conducted experiments on transfer learning using 1) the OPT model across four different sizes: 125M, 350M, 1.3B, and 2.7B, and 2) the FlanT5 model across three sizes: *base, large*, and *XL*. However, we encountered significant instability during training with FlanT5-large. As a result, we mainly present the transfer results between FlanT5-base and FlanT5-XL in most conditions. We also describe how to adapt the VPGTrans to train FlanT5-large by adjusting some hyperparameters (*cf.* §4.3).

4.2 VPGTrans Enabling Faster Convergence without Performance Drop under TaS

First of all, as shown in Fig. 7, our VPGTrans can consistently accelerate the model convergence. Especially for COCO caption and NoCaps that requires more training steps to converge, our VPGTrans (green line) can be higher than the other two lines (blue and orange lines). To give a quantitative evaluation of the speed-up rate, we show the speed-up rate in Table 3. The speed-up rate is calculated by considering the number of epochs reduced to achieve the best TFS performance on a particular dataset. Formally, given a dataset D, TFS obtains the best performance p on D at epoch e_{tfs}, whereas VPGTrans first achieves a better performance than p at epoch e_{vt}. The speed-up rate on D is given by $\frac{e_{tfs}}{e_{vt}}$. According to Table 3, our VPGTrans can achieve at least 4 times speed-up on 40% of Transfer-Task variants. Furthermore, for the two caption datasets, which take a long time to converge, our VPGTrans $\text{OPT}_{125\text{M}\rightarrow2.7\text{B}}$ delivers a 10 times speed-up.

Following we provide further important observations with respect to the efficiency transfer by VPGTrans.

• **More thorough convergence leads to better performance on caption tasks.** Fig. 7 indicates that the caption tasks generally necessitate more training steps to attain the best performance. Particularly for larger models such as $\text{OPT}_{1.3\text{B}}$, $\text{OPT}_{2.7\text{B}}$, and $\text{FlanT5}_{\text{XL}}$, 10 epochs of training are inadequate to achieve optimal performance. In such cases, our VPGTrans significantly accelerates model training, leading to better performance than TFS.

• **The smaller the LLM_{src}, the easier the transfer.** We notice an interesting phenomenon that when transferring to a given LLM_{tgt}, both the convergence rate and optimal performance are roughly inversely proportional to the size of LLM_{src}. For example, as shown in Table 3, the $\text{OPT}_{125\text{M}\rightarrow2.7\text{B}}$ and $\text{OPT}_{350\text{M}\rightarrow2.7\text{B}}$ have much higher speed-up rate than $\text{OPT}_{1.3\text{B}\rightarrow2.7\text{B}}$ on all of the datasets. Meanwhile, as demonstrated in Fig. 7, the optimal performance of $\text{OPT}_{125\text{M}\rightarrow2.7\text{B}}$ and $\text{OPT}_{350\text{M}\rightarrow2.7\text{B}}$ are better than $\text{OPT}_{1.3\text{B}\rightarrow2.7\text{B}}$ on 3 VQA tasks.

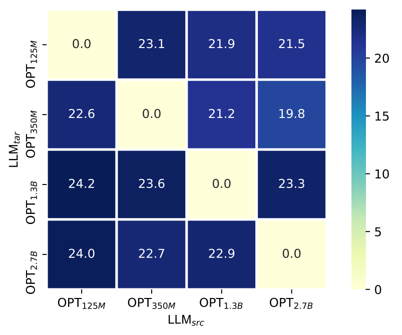


Figure 8: The confusion matrix of cross-size VPG transfer. Only linear layers are trained and models are tested on COCO caption with SPICE metric to compare the VPGs trained on different LLM_{src}.

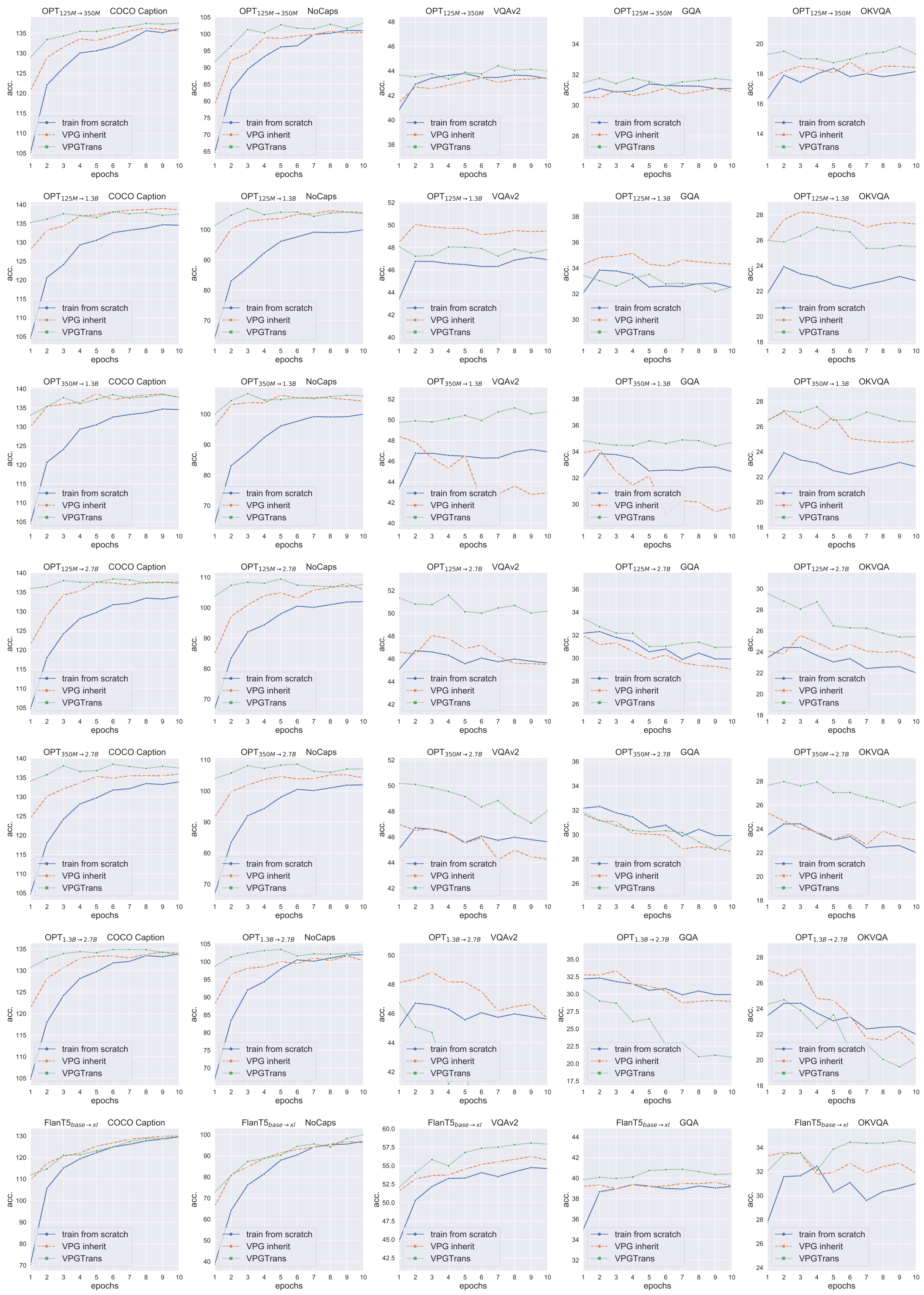


Figure 7: Comparison between different methods across 7 TaS variants on 5 tasks. Note that the performance of all tasks is zero-shot evaluation.

Table 4: Testing VPGTrans on scaled-up scenarios.

Models	VQAv2 val	GQA test-dev	OKVQA test	speed-up	training data
BLIP-2 ViT-G OPT$_{6.7B}$	54.3	**36.4**	36.4	1.0	129M
BLIP-2 ViT-G OPT$_{2.7B \rightarrow 6.7B}$ (**ours**)	**57.2**	36.2	**39.8**	**10.7**	**13.8M**
BLIP-2 ViT-G FlanT5$_{XXL}$	65.2	44.7	**45.9**	1.0	121.6M
BLIP-2 ViT-G FlanT5$_{XL \rightarrow XXL}$ (**ours**)	**65.2**	**45.0**	45.0	**21.1**	**5.3M**

We hypothesize that training VPG on larger LLM will have a worse influence on VPG's existing fine-grained perception ability, due to the enlarging embedding dimensions. To validate our hypothesis, we fix the VPG weight and only tune linear projectors to test VPGs trained with different LLM$_{src}$ through cross-size transfer. The SPICE [3] metric on COCO caption is used to evaluate the VPG's visual perception ability, whereas SPICE is mainly designed for checking the objects, attributes, and relationships in the generated captions. As shown in Fig. 8, for each row, given the LLM$_{tar}$, the performance of VPG trained on smaller LLM$_{src}$ can outperform the larger ones for most conditions, which indicates a better visual perception ability of VPG trained on smaller LLM$_{src}$. Therefore, **adapting a VPG from a smaller LLM which is less affected, is helpful to take fewer steps to reach the TFS's best performance and achieve even better performance.**

4.3 VPGTrans Enabling Stable Training under TaS

As we illustrated before, FlanT5$_{large}$ training is extremely unstable. Even when the learning rate is adjusted to one-tenth of its original value, the model does not converge or shows a very slow convergence rate after 4 epochs. However, we find that by lowering the learning rate for stage-2 training, our VPGTrans can achieve stable training on FlanT5$_{large}$. We plot the performance curve of the COCO caption in Fig. 9.

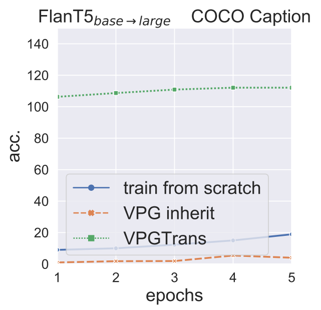


Figure 9: Comparison of training FlanT5$_{large}$ using different methods.

4.4 VPGTrans Enabling Training with Less Data under TaS

We empirically find that TaS can reduce the requirement for the amount of training data. By reducing the training data to only COCO, we find no obvious of performance drop. However, we want to stress that the retained data should be of high quality, which means that if the same number of SBU data is retained, the performance especially for captioning will drop. The conclusion can also be found in Table 4. We refer the readers to the next subsection for more details.

4.5 Scale-up Experiments

To validate the effectiveness of our VPGTrans on the real-world application level, we conduct experiments on transferring from BLIP-2 ViT-G OPT$_{2.7B}$ to OPT$_{6.7B}$ and from BLIP-2 ViT-G FlanT5$_{XL}$ to FlanT5$_{XXL}$.

Experimental Settings. We try to imitate BLIP-2's pre-training data composition. First of all, two human-annotated datasets **COCO** and **VG** are used. **SBU** is also used. Then, BLIP-2 uses BLIP to generate captions for the 115M web images and rank them with CLIP ViT-L/14. We also adopt similar synthetic data from **Laion-COCO**.[7] We report the concrete number of data we use in Table 4. For the stage-1 training, we keep the same as the previous validation experiments where COCO and SBU are used for warm-up with a 5 times the learning rate. Then, we use COCO, VG, and Laion-COCO for the stage-2 training. Note that we have tried to include Laion-COCO and VG for the stage-1 training, but found no obvious difference and thus use COCO and SBU just for simplicity.

Speed-up with non-degenerated performances. As illustrated before, the caption tasks are not strict zero-shot settings, and BLIP-2 does not report the same "zero-shot" performance in their paper, thus we only compare the zero-shot results with BLIP-2 on 3 VQA tasks. As shown in Table 4,

[7]`https://laion.ai/blog/laion-coco`

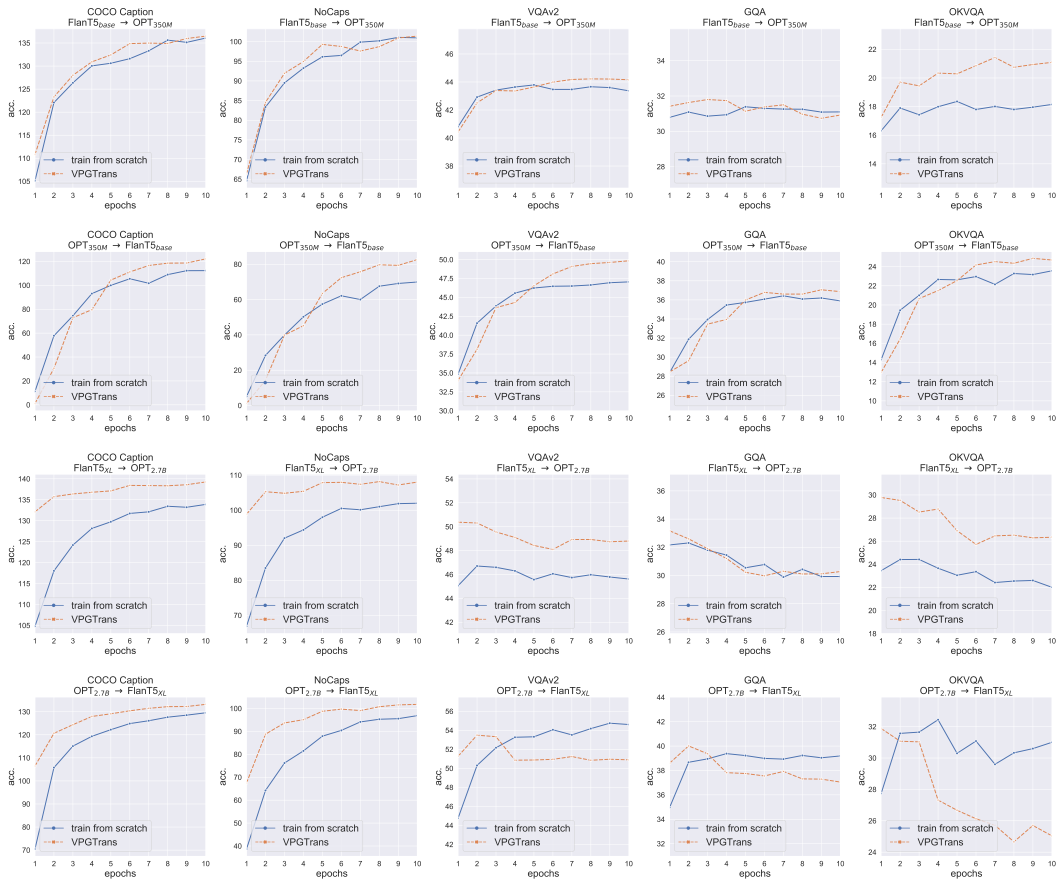


Figure 10: Comparison between different methods across 4 TaT variants on 5 tasks. Note that the performance of all tasks is zero-shot evaluation.

we can see that **(1) $\mathbf{OPT_{2.7B \to 6.7B}}$:** our VPGTrans achieves a 10.7 times speed-up with only 10.7% training data, while the performance on VQAv2 and OKVQA have over 2 points improvement. **(2) $\mathbf{FlanT5_{XL \to XXL}}$:** VPGTrans can achieve 21.1 times speed-up with less than 5% training data while achieving the same performance on VQAv2, higher performance on GQA and slightly lower performance on OKVQA. Note that continuing training the $\mathrm{FlanT5_{XL \to XXL}}$ only shows improvement on VQAv2 and GQA but does not show an improvement on OKVQA. Thus, we do not show a model with higher OKVQA with more training steps.

5 Exp-II: Transfer across Different Model Types

In this section, we further investigate the transfer across different model types. For simplicity, we mark this type of transfer as **TaT**.

5.1 Experimental Settings

In this part, we introduce baselines and transfer variants. For details about training data and implementation details, please refer to the experiment settings in the Preliminary (*cf.* 2.2).

Baselines. We mainly compare our VPGTrans with *training from scratch* (TFS), where we report their zero-shot performance on the aforementioned 5 tasks. Other details (*cf.* 4.1) are totally the same with TaS experiments.

Transfer Variants. We conducted experiments on transfer learning between 1) $\mathrm{OPT_{350M}}$ and $\mathrm{FlanT5_{base}}$, and 2) $\mathrm{OPT_{2.7B}}$ and $\mathrm{FlanT5_{XL}}$.

Table 5: Performance of customized VL-LLMs.

Models	VQAv2 val	GQA test-dev	OKVQA test
BLIP-2 OPT$_{2.7B}$	53.5	34.6	31.7
BLIP-2 OPT$_{6.7B}$	54.3	36.4	36.4
BLIP-2 OPT$_{2.7B\rightarrow6.7B}$ **(ours)**	57.2	36.2	**39.8**
VL-LLaMA$_{7B}$ **(ours)**	**58.1**	**37.5**	37.4
BLIP-2 FlanT5$_{XXL}$	65.2	44.7	**45.9**
BLIP-2 FlanT5$_{XL\rightarrow XXL}$ **(ours)**	**65.2**	**45.0**	45.0

5.2 VPGTrans Enabling Faster Convergence only on Large LLMs under TaT

• **There is no speed-up of TaT on small LLMs.** An interesting finding is that, on TaT our VPGTrans does not show speed-up for small models, and even shows a degeneration of training speed on in the initial several epochs. As shown in Fig. 10, when transferring from FlanT5$_{base}$ to OPT$_{350M}$, the VPGTrans shows a similar convergence rate with TFS, where the curves have almost coincided in the first 4 datasets. When transferring from OPT$_{350M}$ to FlanT5$_{base}$, the convergence speed of VPGTrans is even slower than TFS in the initial several epochs.

• **Speed-up of VPGTrans happens in large LLMs.** However, when moving to the large LLMs like OPT$_{2.7B}$ and FlanT5$_{XL}$, there is an obvious speed-up. As shown in Table 3, we can see at least 2 times speed-up when transferring from FlanT5$_{XL}$ to OPT$_{2.7B}$. We empirically find that **the soft prompts for larger LLM are more linear transferrable among different LLM types**. As shown in Fig. 10, when transferring between FlanT5$_{base}$ and OPT$_{350M}$, the VGPTrans' 1st epoch results on two caption datasets are limited, where only a linear operation can be trained. The result of OPT$_{350M}$ $\rightarrow$ FlanT5$_{base}$ on the COCO caption is even near zero. By contrast, the VPGTrans' 1st epoch results of OPT$_{2.7B}$ $\leftrightarrow$ FlanT5$_{XL}$ are obviously higher than TFS. We hypothesize that larger LLM typically learned more generalizable text embeddings and share more similarity among relative word distances, which enables an easier VPG transfer.

6 Customizing New VL-LLMs with Any LLMs

Above we thoroughly certify the efficacy of our proposed VPGTrans approach for higher efficient transfer of VPG. In this section, we illustrate how to apply the VPGTrans framework for VPG transfer to customize new VL-LLMs with any LLMs. We representatively demonstrate the building of two novel VL-LLMs: VL-LLaMA and VL-Vicuna.

6.1 VL-LLaMA

LLaMA [28] is a recently opened large language model, that featured better performance with smaller parameter amounts. By applying our VPGTrans, we can equip the LLaMA model with a VPG to perceive the visual information.

Specifically, we transfer the VPG from BLIP-2 OPT$_{6.7B}$ to LLaMA$_{7B}$, and build a VL-LLaMA$_{7B}$. The implementation details are the same as the above scale-up experiment (*cf.* §4.5). As shown in Table 5, our customized VL-LLaMA$_{7B}$ can achieve the best performance on VQAv2 and GQA among all of the decoder-only LLMs' BLIP-2 variants. Also, for OKVQA data, it outperforms the original BLIP-2 OPT$_{6.7B}$ by 1 point.

6.2 VL-Vicuna

A more exciting application of our VPGTrans is to build a GPT-4 [6] style multimodal conversation chatbot. To achieve our goal, we employ Vicuna as our base LLM. Vicuna [8] is built based on LLaMA and gets supervised instruction fine-tuning by using conversation data.

Similarly, we also transfer the VPG from BLIP-2 OPT$_{6.7B}$. However, image-caption pairs trained models typically prefer to output short text. Therefore, we further tune our VPG and projector on MiniGPT-4 [34]'s 3,439 self-instruct data instances to better align with the conversation scenario.

We compare our VL-Vicuna with MiniGPT-4 in Fig. 11. MiniGPT-4 also employs a BLIP-2's VPG but only trains linear projectors with millions of image-text pairs and 3,439 self-instruct data instances. We can see that our VPGTrans-built VL-Vicuna can result in better visual perception ability. For example, for the 1st example in Fig. 11, MiniGPT-4 can not count the number of people correctly which also brings a bad influence to the following conversation. By contrast, our VL-Vicuna can not only correctly count the number of people in the image but also describe the roles of these 3 people.

Moreover, we show some failure cases of our VL-Vicuna. As shown in Fig. 12, our VL-Vicuna falsely recognize the flower decoration on the cake as a vase of flower and can not read the time from the clock, which indicates an improvement space for the visual perception ability.

7 Related Work

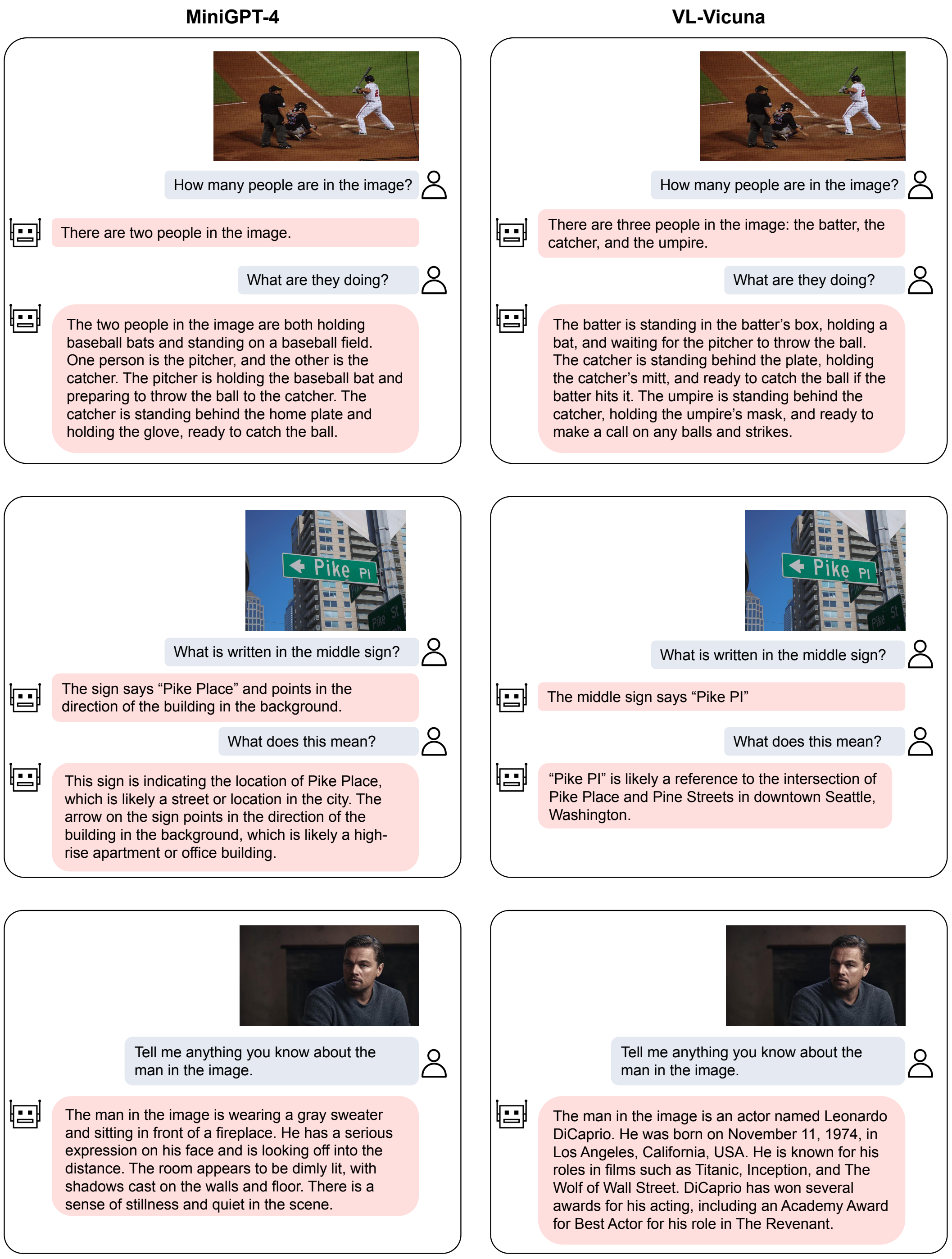
7.1 Vision and Language Models

Vision and language models (VLMs) aims at understanding visual and textual information with a single model. Previously, the VLM mainly employs a pre-trained object detector as its feature extractor and conducts unsupervised training on a huge amount of image-text pairs. For example, VL-Bert [26], ViL-Bert [20] and Uniter [7] adopt Faster-RCNN [24] to extract image information into object features and take advantage of the masked language model as their pre-training task. Later, due to the prevalence of vision transformer (ViT), the VLM paradigm is turned into end-to-end training with a ViT as the visual encoder. The representative works include ALBEF [16], BLIP [17] and BEIT-v3 [30], which show the state-of-the-arts supervised training performance on a wide range of downstream tasks.

Recently, LLMs have shown their remarkable capability as zero/few-shot learners [5] and a series of emergent abilities [31] like in-context learning [5], and chain-of-thoughts reasoning [32]. A new paradigm, i.e., VL-LLMs are created by associating the VLM or pure vision encoders with LLMs. As we illustrated before, the VLM or visual encoders are typically able to convert the input vision signals into LLM-understandable soft prompts, and thus we call them VPG. The VL-LLMs advance in inheriting the great potentials of the backbone LLMs, and thus are capable of achieving excellent zero/few-shot performances [2, 18] on downstream tasks or be equipped with visual planning ability [11]. However, connecting the VPG to the existing LLMs with further tuning is costly. Even the BLIP-2 [18], targeted at efficient training, will take over 600 A100 GPU hours on over 100M image-text pairs for its largest model. With this regard, our proposed VPGTrans can effectively reduce the cost of building new VL-LLMs with the help of existing ones.

7.2 Prompt Transfer

In this paper, we investigate the VPG transfer, where the soft prompt is to represent the content of specific inputs like images and videos. In addition to the content prompt, the more explored soft prompt is the task prompt [14, 33, 13], where a sequence of soft prompts are tuned to assist the pre-trained models to achieve better performance on specific tasks. There have already been some works exploring the transferability of task prompts. For example, Su et al. [27] conducts a series of experiments to illustrate the transferability across tasks and models. Specifically, Su et al. [27] find that the transfer between similar tasks is beneficial for training speed-up and better performance. Lester et al. [15] proposes to recycle soft prompts across models with vocab-to-vocab transformations, or linear-combination transformations. Note that our word converter initialization is similar to the idea of vocab-to-vocab transformations. However, we do not observe a zero-shot transfer in our experiments like them, which indicates a potential difference between the content prompts and task prompts. Another way of soft prompt transfer [9] is to conduct prompt tuning on discrete prompts, and thus the discrete prompts can be directly shared across models. Different from these task prompts transfer works, our VPG transfer scenario actually suffers from fewer limitations. For example, the task soft prompts transfer suffers from the dimension change problem, where the main body of the trainable parameters should be processed. However, our VPG (the main trainable parameters) can naturally be shared among LLMs with different embedding dimensions and leave the dimension change problem to a simple projector with ignorable parameters.



MiniGPT-4

How many people are in the image?

There are two people in the image.

What are they doing?

The two people in the image are both holding baseball bats and standing on a baseball field. One person is the pitcher, and the other is the catcher. The pitcher is holding the baseball bat and preparing to throw the ball to the catcher. The catcher is standing behind the home plate and holding the glove, ready to catch the ball.

What is written in the middle sign?

The sign says “Pike Place” and points in the direction of the building in the background.

What does this mean?

This sign is indicating the location of Pike Place, which is likely a street or location in the city. The arrow on the sign points in the direction of the building in the background, which is likely a high-rise apartment or office building.

Tell me anything you know about the man in the image.

The man in the image is wearing a gray sweater and sitting in front of a fireplace. He has a serious expression on his face and is looking off into the distance. The room appears to be dimly lit, with shadows cast on the walls and floor. There is a sense of stillness and quiet in the scene.

VL-Vicuna

How many people are in the image?

There are three people in the image: the batter, the catcher, and the umpire.

What are they doing?

The batter is standing in the batter’s box, holding a bat, and waiting for the pitcher to throw the ball. The catcher is standing behind the plate, holding the catcher’s mitt, and ready to catch the ball if the batter hits it. The umpire is standing behind the catcher, holding the umpire’s mask, and ready to make a call on any balls and strikes.

What is written in the middle sign?

The middle sign says “Pike Pl”

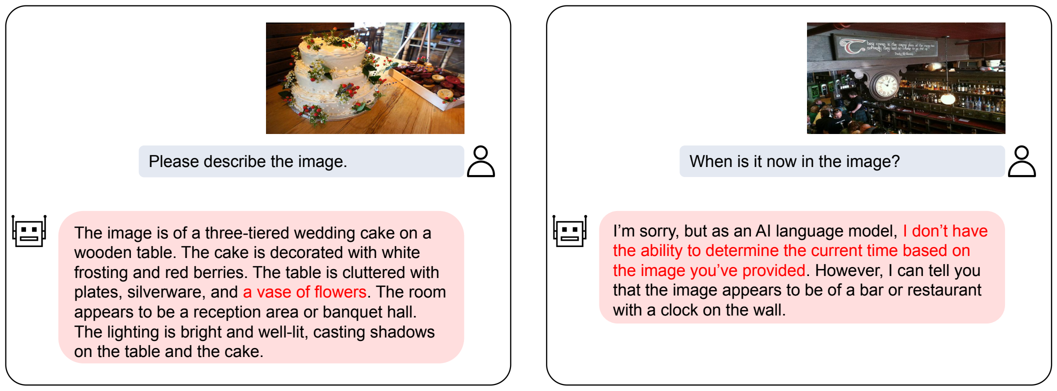
What does this mean?

“Pike Pl” is likely a reference to the intersection of Pike Place and Pine Streets in downtown Seattle, Washington.

Tell me anything you know about the man in the image.

The man in the image is an actor named Leonardo DiCaprio. He was born on November 11, 1974, in Los Angeles, California, USA. He is known for his roles in films such as Titanic, Inception, and The Wolf of Wall Street. DiCaprio has won several awards for his acting, including an Academy Award for Best Actor for his role in The Revenant.

Figure 11: Comparison between MiniGPT-4 and our VL-Vicuna.


Please describe the image.

The image is of a three-tiered wedding cake on a wooden table. The cake is decorated with white frosting and red berries. The table is cluttered with plates, silverware, and a vase of flowers. The room appears to be a reception area or banquet hall. The lighting is bright and well-lit, casting shadows on the table and the cake.

When is it now in the image?

I'm sorry, but as an AI language model, I don't have the ability to determine the current time based on the image you've provided. However, I can tell you that the image appears to be of a bar or restaurant with a clock on the wall.

Figure 12: Failure cases of our VL-Vicuna.

8 Conclusion

Transferring an existing VPG from any existing VL-LLMs is a feasible solution to build a new VL-LLM while avoiding huge computational costs. In this work, we conduct a throughout investigation to the problem of VPG transferability across LLMs. We first explore the key factors for maximizing the transfer efficiency under the VPG transfer across different LLM sizes (e.g., small-to-large), and across different LLM types. Based on the key observations we propose a novel two-stage transfer framework, namely VPGTrans, which can help achieve comparable or better performance while significantly reducing the training costs. With VPGTrans, we achieve the VPG transfer from BLIP-2 OPT$_{2.7B}$ to BLIP-2 OPT$_{6.7B}$ with over 10 times speed-up and 10.7% training data, compared with connecting a VPG to OPT$_{6.7B}$ from scratch. Moreover, a list of important findings and possible reasons behind them are shown and discussed. Finally, we demonstrate the practical value of our VPGTrans, by customizing new VL-LLMs via VPG transfer from existing VL-LLMs.

Potential Impact and Limitations

Our VPGTrans is designed for building new VL-LLMs with lower computational cost, i.e., shorter training time and less training data. With an already pre-trained VL-LLM, VPGTrans enables fast VPG transfer to build either a larger VL-LLM or a VL-LLM with different type of LLM. We hope VPGTrans can facilitate teams in LLM communicty to customize their VL-LLMs without costing too much. There are also possible limitations of the current version of VPGTrans. The first one is that our VPGTrans should rely on any already-aligned VPG. The second potential limitation is that the VPGTrans-built VL-LLMs still suffer from the common problems of the VL-LLMs. For example, the VL-Vicuna may make up some sentences with falsely recognized visual facts. It is worth exploring associating our VPGTrans with training safer VL-LLMs.

Acknowledgments and Disclosure of Funding

This research is supported by the Sea-NExT Joint Lab.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.

[2] 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.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[5] 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.

[6] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pages 104–120. Springer, 2020.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://vicuna.lmsys.org`.

[9] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

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

[11] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[12] 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.

[13] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022.

[14] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[15] Brian Lester, Joshua Yurtsever, Siamak Shakeri, and Noah Constant. Reducing retraining by recycling parameter-efficient prompts. *arXiv preprint arXiv:2208.05577*, 2022.

[16] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[20] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[21] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[22] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

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

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[25] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in Neural Information Processing Systems*, 35:9512–9524, 2022.

[26] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[27] Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, et al. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, 2022.

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[29] 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.

[30] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[31] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[32] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[33] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[34] 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.