Omni-SMoLA: Boosting Generalist Multimodal Models with Soft Mixture of Low-rank Experts

Jialin Wu Xia Hu Yaqing Wang Bo Pang Radu Soricut
Google Research
{jialinwu, xiahu, yaqingwang, bopang, rsoricut}@google.com

Abstract

Large multi-modal models (LMMs) exhibit remarkable performance across numerous tasks. However, generalist LMMs often suffer from performance degradation when tuned over a large collection of tasks. Recent research suggests that Mixture of Experts (MoE) architectures are useful for instruction tuning, but for LMMs of parameter size around O(50-100B), the prohibitive cost of replicating and storing the expert models severely limits the number of experts we can use. We propose Omni-SMoLA, an architecture that uses the Soft MoE approach to (softly) mix many multimodal low rank experts, and avoids introducing a significant number of new parameters compared to conventional MoE models. The core intuition here is that the large model provides a foundational backbone, while different lightweight experts residually learn specialized knowledge, either per-modality or multimodally. Extensive experiments demonstrate that the SMoLA approach helps improve the generalist performance across a broad range of generative vision-and-language tasks, achieving new SoTA generalist performance that often matches or outperforms single specialized LMM baselines, as well as new SoTA specialist performance.

1. Introduction

Large multimodal models (LMMs) [6–8, 14, 33, 53] demonstrate remarkable performance on a variety of tasks including visual question answering, image captioning, visual document understanding, etc. To date, the best performance on most of these tasks is achieved by so-called *specialist* LMMs, but their large scale makes it impractical to deploy a multitude of such specialists at once. As a result, so-called *generalist* LMMs emerge as an obvious choice, where such a model is trained and deployed to handle a wide range of tasks using the same set of model parameters.

Building a single generalist model to solve multiple tasks remains challenging. A straightforward approach is to fine-tune the model parameters with supervised data representing multiple tasks. However, recent research suggests that it causes non-negligible performance degradation compared to the performance of a single-task specialist [6]. It is likely that, even though these tasks share the same configuration of modalities (e.g., image + text as input, text as output), what the model needs to solve for is significantly diverse – for instance, some tasks require recognizing the fine-grained identity of visual content, others may rely on world-knowledge outside of the visual scene, while others require reading and understanding texts from images.

Recent work [48] show that Mixture-of-Experts (MoE) models stand to benefit more from instruction tuning compared to dense models, and serve as good candidate architectures for building generalist large language models. Intuitively, this should work well because different expert modules can specialize and handle different tasks. However, there is an obvious issue with applying the MoE design on Transformer blocks for large-scale models: the different transformer blocks result in replicating the model parameters using high-rank experts. This creates a situation in which the scale of each expert model block compared to their dense-model counterparts is much more limited.

In this work, we address the aforementioned limitations by introducing Omni-SMoLA, an architecture that efficiently mixes many multi-modal low rank experts. Using this architecture, we demonstrate strong capabilities for adapting pretrained models to tackle specialized tasks. The core intuition is that a large pretrained (or instruction-tuned) model provides a foundational backbone of capabilities (we denote this model by θ^*), while different lightweight experts learn additional specializations (which can be knowledge, style, or capabilities). In particular, for the modalities considered in this paper (text & vision), the Omni-SMoLA architecture consists of three sets of experts, focusing on text tokens, visual tokens and multimodal tokens, respectively, in order to satisfy different needs from various tasks.

In general, the SMoLA design has several important proprieties. First, due to its adoption of the low rank expert design [43] and unlike conventional MoE transformer models [15, 18, 30, 48], the total parameter count is not proportional to the product of expert counts and the parameter counts in each expert as the backbone still contains majority of the parameters. This allows it to bypass the limitation on the number of experts used, which helps achieve better generalist performance. Second, this design is potentially compatible with any large model architecture, either dense or MoE. And, last but not least, it allows for the freedom to potentially adopt different model architectures between the pretraining stage and multi-task learning (or instruction tuning) stage.

We evaluate the Omni-SMoLA approach on a variety of settings, starting from PaLI-3 [7] (a 5B LMM) and PaLI-X [6] (a 55B LMM), models that have current state-of-the-art (SOTA) performance across a wide range of vision-language benchmarks. The settings include various image captioning tasks and visual question answering tasks, and we experiment with possible combinations in terms of model specialization. We find that: (1) Omni-SMoLA achieves better average performance compared to full-model fine-tuning baselines for both PaLI-3 and PaLI-X; our experiments show that it achieves new SoTA results on multiple vision-language benchmarks, both under generalist settings and under specialist settings; (2) the performance improves with the introduction of the Omni experts, and also increases with the number of experts; (3) in spite of the added modules and a large number of experts per module, the inference speed is only slightly slower compared the base models, indicating the efficiency of this design.

2. Related Work

2.1. Large Multi-modal Models

Inspired by the success of Large Language Model [4, 10, 12], there is a growing interest of building large multi-modal models (LMMs) [7, 8, 14, 33] that is designed to understand both vision and language signals simultaneously [14, 32]. The main approach is to integrate a pre-trained image encoder, which represents images as a sequence of continuous embeddings, to autoregressive language model [6–8, 14, 33, 53]. PaLI series of works [6–8] integrate pretrained ViT models [13] to encoder-decoder language framework. PaLM-E [14] incorporate vision encoders as sensor modalities to language model and enable the model to process multiple images in text sentences in a flexible way. BLIP-2 [33] propose a lightwieght Querying Transformer to leverage frozen pre-trained image encoder and language model for multimodal tasks.

2.2. Parameter-Efficient Fine-Tuning

Recently the success of scaling up model size encourage the development of larger language models [6, 9, 44, 52]. Meanwhile, parameter-efficient fine-tuning [2, 22, 24, 42, 45, 51, 52] aims to discover a more efficient solution to adapt large models to particular downstream tasks. Instead of full model fine-tuning which updates the entire set of model parameters, parameter-efficient fine-tuning updates or adds a relatively small number of parameters and leaves the rest of model parameters fixed [52]. FISH Mask[51] applies a fixes sparse mask on model parameters and only updates mask-selected parameters. Adapters [2, 22, 42, 45] inserts new trainable dense layers into Transformer and leave the original model parameters frozen. Prefix-tuning [34] and prompt-tuning [31] freeze parameters of the model and learn continuous prompts. LoRA [24] injects trainable low-rank decomposition matrices into every layer of Transformer and freezes the pretrained language model parameters. In particular, LoRA shows outstanding capability to achieve competitive or even better performance than fine-tuning with only 0.1% trainable parameters [24, 57]

2.3. Mixture-of-Experts for Multitask Learning

Mixture-of-Experts (MoE) architectures are centered around enhancing conditional computation capabilities and scale parameters in neural architectures such as Transformers. The MoE transformer models [17, 29, 46, 61] typically employ N feed-forward networks, referred to as "experts". Each of these experts has its unique set of trainable weights, enabling them to craft distinct representations for each input token based on contextual information. Multi-task learning (MTL), a popular ML topic for many years, aims at finding solutions to simultaneously improving performance on multiple tasks of interests [5, 35]. Recently, mixture-of-experts (MoE) [25, 26, 47] approaches have become a promising approach for MTL [16], benefiting from its strategy of separating the parameter space and adopt relevant model part to different tasks.

Inspired by these advances, there is an increasing interest in investigating the application of MoEs in Transformer-based large models. Some methods adopt MoE in Transformer structure of large language models [15, 18, 30, 48]. Gshard [30] introduces the idea of scaling Transformer in LMMs with MoE layers, in which the feed forward layer of every other Transformer is replaced by a Sparsely-Gated MoE layer. This MoE Transformer structure is then used in [15] to develop a family of Decoder-only language models, and [48] which finds MoE modified LLM models benefits more from instruction tuning than dense LLMs.

Some other methods explore combining MoE with parameter-efficient fine-tuning. AdaMix [56] proposes a mixture-of-adapters mechanism to improve per-task tuning performance. The most relevant work is the concurrent research [60] that introduces mixture of LoRA by weighted summing of different LoRA outputs. While conceptually similar, our SMoLA approach differs by having significantly lower computational cost, and also allowing hundreds of experts to handle single and multiple modalities with negligible inference speed cost. We find that scaling to hundreds of experts is crucial to attaining improved generalist performance.

3. Methodology

3.1. Preliminaries

Low-rank Adaptation (LoRA). Low-Rank Adaptation (LoRA) [23] is a technique designed to enhance the adaptability of pretrained transformer models to new tasks with a minor increase in trainable parameter counts. It can be applied on any linear layers, offering great compatibility with recent large models.

We denote $\mathtt{W} \in \mathtt{R}^{\mathtt{d_1} \times \mathtt{d_2}}$ as the weight matrix for a linear layer from the large model. LoRA introduces two low-rank matrices $\mathtt{W}^{\mathtt{in}} \in \mathtt{R}^{\mathtt{r} \times \mathtt{d_1}}$ and $\mathtt{W}^{\mathtt{out}} \in \mathtt{R}^{\mathtt{d_2} \times \mathtt{r}}$ for each layer, where $\mathtt{r} \ll \mathtt{min}\{\mathtt{d_1}, \mathtt{d_2}\}$. The $\mathtt{W}^{\mathtt{in}}$ and $\mathtt{W}^{\mathtt{out}}$ are consecutively applied to the input of the linear layer to project the input to a low rank space and then project back to the output space. The adapted weights $\mathtt{W}'$ can be represented as $\mathtt{W}' = \mathtt{W} + \mathtt{W}^{\mathtt{out}}\mathtt{W}^{\mathtt{in}}$. As the rank of $\mathtt{W}^{\mathtt{in}}$ and $\mathtt{W}^{\mathtt{out}}$ is limited by $\mathtt{r}$ and typically much smaller than $\mathtt{d_1}$ and $\mathtt{d_2}$, the LoRA approach serves as a compact and efficient adaptation mechanism.

Soft Mixture of Experts (Soft MoE). We briefly recap the Soft MoE model in this section (details can be found in [43]). The core idea is to learn a dispatcher module that can dispatch input tokens to different experts, and a combiner module that can combine the results from all the experts and project them back to the original token space.

We denote the input to the transformer block as $\mathbf{X} \in \mathtt{R}^{\mathtt{N} \times \mathtt{d_1}}$, consisting of $\mathtt{N}$ tokens. Soft MoE introduces a routing matrix $\Phi \in \mathtt{R}^{\mathtt{E} \times \mathtt{d_1}}$ that corresponds to $\mathtt{E}$ experts. The dispatcher and combiner are represented by Eq. 1 and 2: $\mathtt{norm}$ denotes l2 normalization and α is a learnable scalar.

$$\mathbf{D} = \mathtt{softmax}(\alpha \cdot \mathtt{norm}(\Phi)\mathtt{norm}(\mathbf{X})^T, \mathtt{axis{=}1}) \quad (1)$$

$$\mathbf{C} = \mathtt{softmax}(\alpha \cdot \mathtt{norm}(\Phi)\mathtt{norm}(\mathbf{X})^T, \mathtt{axis{=}0}) \quad (2)$$

Each expert model $f_{\mathtt{i}}$ (usually MLP Blocks) operates on the corresponding slice of dispatched inputs $\tilde{x}_i = (\mathbf{DX})_{\mathtt{i},:}$ to produce $\tilde{y}_{\mathtt{i}} = f_{\mathtt{i}}(\tilde{x}_{\mathtt{i}})$. Then, the combiner $\mathbf{C}$ projects the output $\tilde{\mathbf{Y}} = [\tilde{y}_0, \tilde{y}_1, ...\tilde{y}_{\mathtt{E-1}}]$ to the token space $\mathbf{Y} = \mathbf{C}^T\tilde{\mathbf{Y}}$.

3.2. SMoLA Block

Conventional MoE design employs high rank experts in their MLP blocks that directly learn to handle different inputs. Therefore, these experts are parameter-heavy and require expensive pretraining. The SMoLA approach relies on adding (to an original base model denoted as θ^*) experts that use a Soft MoE architecture, while simultaneously avoiding significantly increasing the parameter count by soft-mixing many zero-initialized *low-rank* experts. Intuitively, the original base model θ^* serves as a foundational backbone, and the additional low-rank experts serve as "specialists" that gather additional specialized knowledge and handle different use cases.

The base model θ^* can be initialized with either pretrained (raw), multitask-tuned, or instruction-tuned checkpoints. Using a raw checkpoint provides a more general backbone, while a multitask-tuned checkpoint provides a backbone focused on a required skill-set of the involved tasks – we consider the decision of whether to use one or the other as a backbone to be application-dependent. Our choice for Soft MoE [43] to instantiate the SMoLA block follows from the desirable properties this architecture exhibits: fully differentiable, with no token dropping, and no expert balance issues.

The right part of Fig 1 presents a SMoLA block. SMoLA operates on linear layers for the maximum flexibility and compatibility. We denote $\mathtt{W}^*$ ($\mathtt{W}^* \in \mathtt{R}^{\mathtt{d_1} \times \mathtt{d_2}}$) as the weight matrix of a linear layer in the base model θ^* and $\mathbf{X} \in \mathtt{R}^{\mathtt{N} \times \mathtt{d_1}}$ as the input with N tokens. Following [43], we introduce the routing matrix $\Phi \in \mathtt{R}^{\mathtt{E} \times \mathtt{d_1}}$ and compute the dispatcher $\mathbf{D} \in \mathtt{R}^{\mathtt{E} \times \mathtt{N}}$ and the combiner $\mathbf{C} \in \mathtt{R}^{\mathtt{E} \times \mathtt{N}}$ using Eq. 1 and 2 for the $\mathtt{E}$ experts.

SMoLA adopts a LoRA-inspired approach for the expert blocks. We introduce trainable low-rank matrices $\mathtt{W}_i^{\mathtt{out}}$, $\mathtt{W}_i^{\mathtt{in}}$ for the i-th expert, producing the output $\tilde{y}_i$ as in Eq. 3.

$$\tilde{y}_i = \mathtt{W}_i^{\mathtt{out}}\mathtt{W}_i^{\mathtt{in}}(\mathbf{DX})_{i,:}^T \quad (3)$$

Then, the output of the SMoLA $\mathbf{Y}$ combines the outputs of each expert and the original linear outputs, as in Eq. 4.

$$\mathbf{Y} = \mathbf{X}\mathtt{W}^* + \mathbf{C}^T[\tilde{y}_0, \tilde{y}_1, ...\tilde{y}_{\mathtt{E-1}}] \quad (4)$$

3.3. Omni-SMoLA

By default, SMoLA blocks take as inputs all the tokens, regardless of their modality (denoted by $\mathcal{S}$MoLA$_{\mathtt{MM}}$ in the next section). However, we note that various multimodal tasks may place a different emphasis on how different modalities are used. For example, image captioning relies more on the visual tokens, VQA tasks on text-heavy images and using upstream OCR focuses more on text, while natural-image VQA must rely on both the visual and text tokens.

Inspired by [55], SMoLA can be seamlessly configured to only adapt tokens for selected modalities. We denote the SMoLA blocks that only take visual tokens or text tokens as $\mathcal{S}$MoLA$_{\mathtt{V}}$ or $\mathcal{S}$MoLA$_{\mathtt{T}}$, respectively. $\mathcal{S}$MoLA$_{\mathtt{MM}}$ refers to the SMoLA blocks that take both visual and text tokens. As shown in Figure 1, Omni-SMoLA (denoted by $\mathcal{S}$MoLA$_{\mathtt{O}}$ in the next section) combines via sum the original backbone outputs with the outputs of $\mathcal{S}$MoLA$_{\mathtt{MM}}$ and the concatenated outputs of $\mathcal{S}$MoLA$_{\mathtt{V}}$ and $\mathcal{S}$MoLA$_{\mathtt{T}}$.

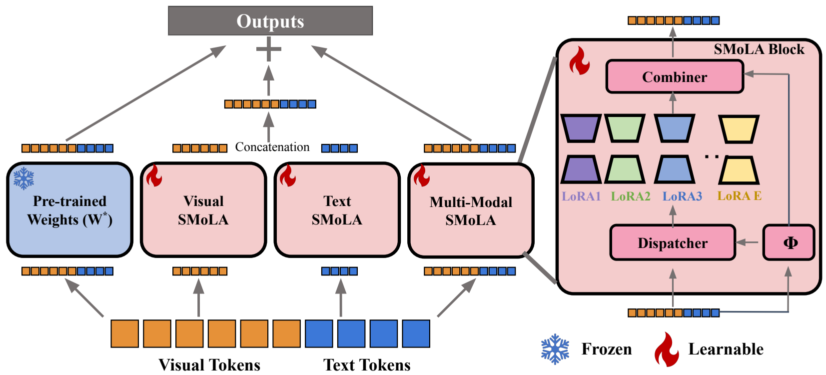

Figure 1. Omini-SMoLA model architecture contains three SMoLA blocks that take as input visual tokens, text tokens and multimodal tokens. Each such block employs a large set of low-rank experts.

3.4. The Properties of Omni-SMoLA

Parameter Efficiency and Time Complexity. The integration of LoRA and Soft MoE results in a combination that achieves a substantial reduction in the number of parameters required for adaptation, compared to traditional MoE [30]. The low-rank matrices introduced by LoRA are of significantly lower dimensionality than the full-rank feedforward matrices, ensuring that the parameter increase is minimal (and controlable via the rank hyperparameter). Not only does this lead to a leaner model, but it also reduces memory requirements, making it feasible to increase the number of experts to enhance performance.

Moreover, the inference cost of applying Omni-SMoLA is negligible. Let $\mathtt{d}_{\mathtt{max}}$ denote $\mathtt{max}\{\mathtt{d}_1, \mathtt{d}_2\}$ and $\mathtt{r}$ denote the rank per expert, the time complexity of SMoLA blocks per-layer is $O(\mathtt{ENd}_{max} + \mathtt{E}(\mathtt{d}_1 + \mathtt{d}_2)\mathtt{r})$. For one single layer, it increases the cost from $O(\mathtt{Nd}_1\mathtt{d}_2)$ to $O(\mathtt{Nd}_1\mathtt{d}_2 + \mathtt{ENd}_{max} + \mathtt{E}(\mathtt{d}_1 + \mathtt{d}_2)\mathtt{r})$. The number of expert $\mathtt{E}$ is always much smaller than $\mathtt{min}\{\mathtt{d}_1, \mathtt{d}_2\}$, while the rank $\mathtt{r}$ (typically a small integer like 4) is much smaller than the input tokens length, especially for multimodal settings where a single high resolution image may easily be responsible for thousands of visual tokens.

Alternative Scaling Dimension. Traditional scaling methods in neural networks often involve increasing the size of the model, either by adding more layers or increasing the dimensionality of the existing layers. The proposed method, on the other hand, introduces an alternative scaling dimension. By leveraging sparse activation and parameter-efficient adaptation, the proposed method achieves scaling through increasing the number of the low-rank experts, which in turn does not result in a severe increase of total model parameter size.

Extensibility for Future Growth. The design of the proposed method inherently supports extensibility, accommodating future growth and adaptations with ease. As the requirements of a task evolve, additional low-rank specialist modules can be seamlessly integrated into the architecture, enhancing the model's capability without necessitating a complete overhaul. This stands in stark contrast to traditional scaling methods, which often require predefined dimensions and layer numbers, limiting the model's adaptability to changing scenarios.

4. Experiments

4.1. Experimental setups

Training Mixtures. We considers three mixtures:

- *Image Captioning mixture*: COCO captions[1] [27] , Textcaps [49], VizWiz-Cap [21].
- *VQA mixture*: VQAv2[2] [19], OK-VQA [37], VizWiz-VQA [20], ST-VQA [3], TextVQA [50], OCRVQA [41], InfoVQA [40], DocVQA [39], ChartQA [38], AI2D [28].
- *Full mixture*: combines the Image Captioning mixture and the VQA mixture.

By default, we use the full mixture in our experiments to simulate the scenario of mixing a wide variety of different tasks. The only exception is Sec. 4.3.6, where we measure the effect of using more focused mixtures.

Task Prompts. We do not use benchmark specific prompts in order to achieve better versatility of the generalist models. Following [6] and [7], we use `Generate the alt_text in {lang} at 0:` as the captioning prompt and `Answer in en: {question}` as the VQA prompt.

[1]In keeping with the multilingual nature of PaLI models, here we used a variant of the original English-only COCO captions that included translated captions for an additional 35 languages.

[2]Included translated questions for an additional 13 languages.

	Model	COCO	NoCaps†	VQAv2	OKVQA	A-OKVQA†		Sci-QA†	TallyQA†	
		Kar.-test	val	test-dev	val	DA	MC	test	simple	complex
Specialist	GIT2 [54]	145.0	126.9	81.7	-	-	-		-	-
	BEiT-3 [55]	147.6	-	84.2	-	-	-		-	-
	PaLM-E [14]	138.7	-	80.0	66.1	-	-		-	-
	InstructBLIP [11]	-	123.1	-	62.1	62.1	73.4	90.7	-	-
	PaLI-X [6]	149.2	126.3	86.0	66.1	-	-		86.0	75.6
	CogVLM [58]	148.7	128.3	84.7	64.7	-	-	92.7	-	-
Generalist	Unified-IO [36]	122.3	100.0	77.9	54.0	45.2	-	-	-	-
	Qwen-VL [1]	-	121.4	79.5	58.6	-	-	67.1	-	-
	CogVLM [58]	147.0	126.2	83.4	58.9	-	-	-	-	-
	PaLI-3$_{FT}$	144.4	120.3	82.5	56.2	59.0	78.7	55.2	80.4	65.4
	SMoLA$^{48}_{O}$-PaLI-3$_{FT}$	146.5	120.3	83.6	58.2	59.8	79.3	55.8	81.8	65.1
	PaLI-X$_{FT}$	148.7	125.6	84.4	60.7	63.9	84.0	67.2	83.8	71.8
	SMoLA$^{48}_{O}$-PaLI-X$_{FT}$	**149.8**	126.1	85.0	62.4	**65.3**	**84.1**	67.8	83.3	70.7

Table 1. Results on natural image captioning and question answering including COCO Captions (Karpathy split), NoCaps, VQAv2, OKVQA, A-OKVQA, ScienceQA and TallyQA test split with end-to-end modeling without OCR pipeline input. Bold and underlined numbers highlight best performance and best generalist performance, respectively. † denotes that there are no training examples from these datasets during training (i.e. out-domain). The numbers in bracket denote the further per-task LoRA tuned performances. We use the same SMoLA$^{48}_{O}$-PaLI-X$_{FT}$ and SMoLA$^{48}_{O}$-PaLI-3$_{FT}$ to handle inferences in Table 1 and Table 2.

	Model	Text Caps	VizWiz Cap	Text VQA	VizWiz VQA	ST VQA	OCR VQA	Info VQA	Doc VQA	AI2D	Chart VQA
		val	test	test	test-dev	test	test	test	test	test	test
***without** OCR pipeline input*											
	Specialist SOTA	158.8 [7]	122.7 [6]	79.5[7]	76.4[58]	84.1[7]	76.7[7]	57.8[7]‡	87.6[7]‡	81.2[6]	70.9[6]
Generalist	Unified-IO [36]	-	-	-	57.4	-	-	-	-	-	-
	Qwen-VL [1]	-	-	63.8	-	-	75.7	-	65.1	62.3	65.7
	CogVLM [58]	151.3	-	68.1	-	-	74.1	-	-	-	-
	mPLUG-DocOwl [59]	111.9	-	52.6	-	-	-	38.2	62.2	-	57.4
	SMoLA$^{48}_{O}$-PaLI-3$_{FT}$	156.7	119.8	-*	70.4	83.8	72.8	52.4	84.5	75.6	68.9
	SMoLA$^{48}_{O}$-PaLI-X$_{FT}$	144.6	120.3	70.5	71.7	78.9	71.6	49.2	80.1	**81.4**	**71.3**
***with** OCR pipeline input*											
	Specialist SOTA	161.0 [7]	125.7 [6]	80.8 [6]	76.4[58]	85.7[7]	77.8[7]	62.4[7]	88.6[7]	81.4[6]	72.3[6]
	SMoLA$^{48}_{O}$-PaLI-3$_{FT}$	159.3	120.4	-*	71.0	85.9	73.9	57.3	87.4	75.5	68.9
	SMoLA$^{48}_{O}$-PaLI-X$_{FT}$	154.7	124.6	**81.1**	73.8	**86.0**	74.9	**65.6**	**90.6**	**81.4**	**73.8**

Table 2. Results on benchmarks more focused on text understanding capabilities. Bold and underlined numbers highlight SOTA performance and SOTA generalist performance, respectively. ‡ marks specialist results with a higher resolution of 1064 where SMoLA used 812. We use the same SMoLA$^{48}_{O}$-PaLI-X$_{FT}$ and SMoLA$^{48}_{O}$-PaLI-3$_{FT}$ to handle inferences with and without OCR pipeline input in Table 1 and Table 2. *Results are missing because test server is not available.

Base Models. We build SMoLA models on top of two variants of PaLI models: PaLI-X [6] and PaLI-3 [7]. PaLI models use contrastively pretrained ViT modules as the visual encoder to produce visual embeddings for input images; these visual embeddings are then concatenated with text embeddings and passed to the encoder-decoder backbone. PaLI-X is a large-scale multimodal model that contains around 55B parameters. We only experimented with using the full-mixture in PaLI-X based experiments, where we adopted a resolution of 672. PaLI-3 is a more nimble variant. It is still highly performant with just around 5B parameters, achieving SOTA results on a broad range of image captioning and VQA tasks that require text understanding capabilities from images. For PaLI-3 based experiments, we use a resolution of 812 for the full mixture and the image captioning mixture, and 1064 for the VQA mixture.

Notation and implementation. We use SMoLA$^{E}_{Y}$-PaLI-3|X$_{RAW|LoRA|FT}$ to denote the config choices for SMoLA:

- `E` denotes the number of experts for each individual modality and for multimodal experts.
- `Y` denotes the SMoLA's modality configuration: `MM` or `O`.
- base model: PaLI-3 vs PaLI-X
- SMoLA's initial checkpoint can be either the `RAW` checkpoint of the base model, the base model tuned using `LoRA` on a given training mixture, or full-model fine-tuned (`FT`)

Model	COCO Cap	Text Cap	VizWiz Cap	VQA v2	OK VQA	Text VQA	VizWiz VQA	ST VQA	OCR VQA	Info VQA	Doc VQA	AI2D	Chart VQA	Avg. δ
	K.test	val	test	test-dev	val	val*	test-dev	test	test	test	test	test	test	
***with** OCR pipeline input, except for COCO Cap, VQAv2, OKVQA*														
PaLI-3 Specialist	145.9	161.0	120.3	85.0	60.1	78.3	72.2	85.7	77.8	62.4‡	88.6‡	75.2	69.5	0.00
SMoLA$_{\mathrm{O}}^{48}$-PaLI-3$_{\mathtt{RAW}}$	144.4	159.1	118.7	82.6	56.2	79.1	70.6	85.5	73.3	55.1	86.6	73.8	67.6	-2.26
PaLI-3$_{\mathtt{FT}}$	146.2	**161.0**	121.1	82.5	56.4	78.7	69.9	84.9	72.7	54.3	85.9	72.8	65.8	-2.31
SMoLA$_{\mathrm{MM}}^{96}$-PaLI-3$_{\mathtt{FT}}$	145.7	159.3	**121.4**	83.4	56.7	80.0	71.5	85.6	73.6	56.7	87.3	75.2	69.2	-1.26
SMoLA$_{\mathrm{O}}^{48}$-PaLI-3$_{\mathtt{FT}}$	**146.5**	159.3	120.4	83.6	58.2	**80.1**	71.0	**85.9**	73.9	57.3	87.4	**75.5**	68.9	-1.07
	K.test	val	test	test-dev	val	test	test-dev	test	test	test	test	test	test	
PaLI-X Specialist	149.2	159.6	125.7	86.0‡	66.1‡	80.8‡	74.6‡	84.5‡	77.3‡	54.8‡	86.8‡	81.4‡	72.3‡	0.00
PaLI-X$_{\mathtt{LoRA}}$	147.3	159.3	125.1	83.5	57.4	78.9	69.6	84.8	72.3	61.4	88.3	78.8	70.9	-1.65
SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{LoRA}}$	148.6	158.8	125.2	84.7	60.8	80.3	73.1	85.2	74.2	64.8	90.1	80.2	73.0	-0.01
PaLI-X$_{\mathtt{FT}}$	148.7	157.0	125.3	84.4	60.7	79.6	72.2	84.7	73.5	62.4	88.2	80.7	70.2	-0.88
SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{FT}}$	**149.8**	154.7	124.6	85.0	62.4	**81.1**	73.8	**86.0**	74.9	**65.6**	**90.6**	**81.4**	**73.8**	**+0.38**

Table 3. Ablation results on image captioning and question answering benchmarks. Bold and underlined numbers highlight best performance and best generalist performance, respectively. ‡ denotes the specialist results with a higher resolution of 1064 for PaLI-3 and 756 resolution for PaLI-X, where we uses 812 for PaLI-3 series and 672 for PaLI-X series. *We use val split as TextVQA test server is broken.

using the training mixture. We use a rank of 128 for LoRA tuning on all linear layers on the PaLI encoder.[3]

For simplicity, we assign the same number of experts to each SMoLA block and use a rank of 4 per expert. SMoLA is applied on all the linear layers in the attention and MLP modules in PaLI encoder blocks. For example, SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{FT}}$ with full-mixture denotes starting with PaLI-X finetuned on the full-mixture, and then SMoLA-tuned on the same mixture using 48 visual-token experts, 48 text-token experts, and 48 multimodal-token experts.

Checkpoint selection. We monitor the scores on the validation splits[4] every 500 iterations with at most 1,024 examples for each task and select the checkpoint with maximum average validation scores.

4.2. Main Results

In this section, we present our main experimental results using the full mixture. Recall that the full mixture contains both image captioning and VQA tasks. We report SMoLA results on the natural image tasks (as well as "out-domain" tasks not included in the training mixture) in Table 1, and results on tasks that focus on understanding texts in images in Table 2. While results are split into these two tables for easier consumption, they are from the same SMoLA-based generalist models trained on one single mixture.

First, note that the generalist PaLI-X$_{\mathtt{FT}}$ (PaLI-X finetuned on the full-mixture) under-performs its specialist counterparts (PaLI-X finetuned for each task individually) on all the benchmark datasets shown in Table 1. Applying SMoLA over PaLI-X$_{\mathtt{FT}}$ outperformed the base generalist model across the board. It effectively shortened the gap to specialist performances, and notably introduced a new SOTA CIDEr score of 149.8 on COCO captioning, outperforming all the specialist models for that task.

It is important to note that Table 1 presents results for both "in-domain" tasks that are included in the training mixture (COCO captioning, VQAv2, and OKVQA), as well as "out-domain" tasks (those marked with †). The in-domain tasks simulate usecases where we are interested in serving one single model for a set of known tasks. The out-domain tasks simulate usecases where we want to apply a generalist model to unseen tasks in a zero-shot setting. The trend we noted above holds for both cases: SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{FT}}$ outperforms base model PaLI-X$_{\mathtt{FT}}$ for both in-domain and out-domain tasks on average. Overall, SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{FT}}$ achieves new SoTA generalist results for all except NoCaps and TallyQA, and furthermore beating fine-tuned specialist models for COCO (in-domain) and A-OKVQA (out-domain). While PaLI-3$_{\mathtt{FT}}$ based models overall under-performs PaLI-X$_{\mathtt{FT}}$ based models on this set of tasks, SMoLA nonetheless improves the base model performance consistently, demonstrating the effectiveness of this technique for both large- and small-scale models.

Table 2 presents SMoLA results on text-heavy tasks in two experimental setups: (a) relying solely on a model's text understanding capabilities from the raw pixels (*without* OCR input), and (b) including tokens extracted by an upstream OCR module as part of the text input (*with* OCR input). In the with-OCR setting, SMoLA$_{\mathrm{O}}^{48}$-PaLI-X$_{\mathtt{FT}}$ shows remarkable results: with one single model, it outperforms specialist SoTA performance on 6 out of 10 datasets, yielding new SoTA performance for TextVQA, ST-VQA, In-

[3] LoRA with rank 512 did not achieve better overall performance.

[4] We use the Pix2Struct validation split for AI2D.

Model	VQAv2	OK VQA	Text VQA	VizWiz VQA	ST VQA	OCR VQA	Info VQA	Doc VQA	AI2D	Chart VQA	Avg. δ
	test-dev	val	test	test-dev	test	test	test	test	test	test	
without *OCR pipeline input*											
PaLI-3 Specialist	85.0	60.1	79.5	71.9	84.1	76.7	57.8	87.6	75.2	70.0	0.0
PaLI-3$_{\texttt{FT}}$	82.1	57.9	79.8	69.2	84.0	72.5	55.9	87.6	74.2	68.0	-1.53
SMoLA$_{\texttt{O}}^{48}$-PaLI-3$_{\texttt{FT}}$	83.4	57.7	**80.0**	70.8	84.0	73.4	57.3	**87.8**	**75.9**	**70.1**	-0.46
with *OCR pipeline input*											
PaLI-3 Specialist	-	-	80.8	72.2	85.7	77.8	62.4	88.6	75.2	69.5	
PaLI-3$_{\texttt{FT}}$	-	-	81.7	70.0	85.5	73.6	59.8	88.8	74.7	67.3	
SMoLA$_{\texttt{O}}^{48}$-PaLI-3$_{\texttt{FT}}$	-	-	**82.2**	72.0	**85.8**	74.6	61.1	**89.3**	**76.0**	**70.4**	

Table 4. Generalist results using the VQA mixture. Bold numbers highlight the results outperforming single specialized PaLI-3 baselines, and underlined numbers presents the results outperform multi-task fine-tuned baselines.

foVQA, DocVQA, AI2D and ChartQA. It also improves over the base model PaLI-x$_{\texttt{FT}}$ (see Section 4.3.1). This indicates that SMoLA is effective in enabling joint processing of information across different modalities: text situated in image, as well as text tokens extracted by the upstream OCR module. In the without-OCR setting, SMoLA$_{\texttt{O}}^{48}$-PaLI-3$_{\texttt{FT}}$ is able to take advantage of PaLI-3's strong text understanding capability and achieves SOTA generalist score on TextCaps, ST-VQA, InfoVQA, and DocVQA.

4.3. Ablation Studies

4.3.1 Different base models

In Section 4.2, we see strong performance from SMoLA$_{\texttt{O}}^{48}$-PaLI-X$_{\texttt{FT}}$, starting from a strong checkpoint (full-model PaLI-X finetuned). In this section, we examine the effect of switching to PaLI-X$_{\texttt{LoRA}}$, which is LoRA-tuned on the mixture and easier to obtain for large models. As shown in Table 3, compared to their corresponding base models, we find SMoLA helps both PaLI-X$_{\texttt{LoRA}}$ and PaLI-X$_{\texttt{FT}}$ to achieve better overall results, obtaining +1.64 and +1.26 improvements on average, respectively. While it is slightly weaker than SMoLA$_{\texttt{O}}^{48}$-PaLI-X$_{\texttt{FT}}$, which outperforms per-task fine-tuned specialist models by an average of 0.38 points, SMoLA$_{\texttt{O}}^{48}$-PaLI-X$_{\texttt{LoRA}}$ still achieves competitive performance versus the specialist models (on average only a difference of 0.01 point). It is worth noting that the SMoLA design improves PaLI-X$_{\texttt{FT}}$ by +2.4 points on DocVQA, +3.2 points on InfoVQA, and +3.6 points on ChartQA, which all involve comprehending rich text and symbols in images. We note some performance drop on the TextCaps task, possibly due to overfitting and unambiguous intention for image captioning tasks when the same prompt is used for TextCaps and natural-image descriptions.

Table 3 also shows other ablation results on using different starting checkpoints (θ^*) for SMoLA. Similar observation holds for the PaLI-3-based models. For instance, applying SMoLA to the raw checkpoint ($i.e.$ PaLI-3$_{\texttt{RAW}}$) achieves better overall score than full model fine-tuning baseline PaLI-3$_{\texttt{FT}}$, and applying SMoLA to PaLI-3$_{\texttt{FT}}$ brings it more competitive against PaLI-3 specialists, outperforming per-task finetuned baselines on 4 benchmarks. One exception is InfoVQA where the specialist uses a higher resolution.

4.3.2 Effect of Using Multi-Modal Experts

We validate the omni experts design by comparing the average performance of using 48 experts on each combination of modalities ($i.e.$ SMoLA$_{\texttt{O}}^{48}$-PaLI-3$_{\texttt{FT}}$) to using 96 experts on all tokens ($i.e.$ SMoLA$_{\texttt{MM}}^{96}$-PaLI-3$_{\texttt{FT}}$). These two variants introduce the same additional FLOPS during inference. As shown in Table 3, SMoLA$_{\texttt{O}}^{48}$-PaLI-3$_{\texttt{FT}}$ has a slightly edge in terms of average performance. This suggests that for the same amount of extra compute, there can be a slight advantage to allow modality-dependent SMoLA blocks.

4.3.3 Effect of Scaling Up the Expert Counts

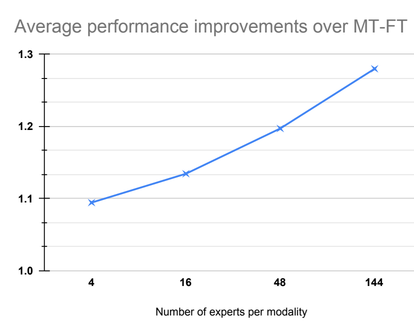

Figure 2. Average results of increasing number of experts.

We study the effect of scaling up the expert counts using SMoLA$_{\texttt{O}}^{\texttt{E}}$-PaLI-3$_{\texttt{FT}}$. Figure 2 plots the average improvements over PaLI-3$_{\texttt{FT}}$ using 4, 16, 48, and 144 experts per modality. The scores are averaged across the tasks presented in Table 3 on validation splits[5] except for InfoVQA. With only 4 experts per modality, SMoLA already yields around +1.1 improvements over PaLI-$3_{\texttt{FT}}$. Scaling up the experts counts further improves the performance: 16, 48 and 144 experts provide 1.14, 1.2, 1.27 average points gain.

4.3.4 Further LoRA tuning to Push SOTA

We note there that the SMoLA$_{\texttt{O}}^{48}$-PaLI-$\text{X}_{\texttt{FT}}$ generalist can be considered a strong foundational model. With further per-task LoRA tuning (using a rank of 4), the SMoLA$_{\texttt{O}}^{48}$-PaLI-$\text{X}_{\texttt{FT}}$ specialists achieve better results than the SMoLA$_{\texttt{O}}^{48}$-PaLI-$\text{X}_{\texttt{FT}}$ generalist, yielding new SOTA results on 9 benchmark datasets: COCO caption, OKVQA, DocVQA, InfoVQA, AI2D, ChartQA, A-OKVQA, ScienceQA and TallyQA (Table 5). These new SOTA results indicate the extensibility of the Omni-SMoLA design.

Model	Split	PaLI-X	SOTA	Ours
COCO	K.test	149.2	149.2 [6]	**152.1**
VQA v2	test-dev	86.0	**86.0** [6]	85.7
OKVQA	val	66.1	66.1 [6]	**66.7**
VizWiz-VQA	test-dev	74.6	**76.4** [58]	75.9
OCRVQA	test	77.3	**77.8** [7]	75.7
DocVQA	test	86.8	88.6 [7]	**90.8**
InfoVQA	test	54.8	62.4 [7]	**66.2**
AI2D	test	81.4	81.4 [6]	**82.5**
ChartQA	test	72.3	72.3 [6]	**74.6**
A-OKVQA	DA (val)	-	62.1 [11]	**70.2**
	MC (val)	-	73.4 [11]	**88.2**
ScienceQA	test	-	92.7 [58]	**94.7**
TallyQA	simple	86.0	86.0 [6]	**86.3**
	complex	75.6	75.6 [6]	**77.1**

Table 5. Further LoRA tuning SMoLA$_{\texttt{O}}^{48}$-PaLI-$\text{X}_{\texttt{FT}}$

4.3.5 Inference Speed Comparison

We compare the inference speed by measuring the number of processed examples per second (eps) for the PaLI-$3_{\texttt{FT}}$ model and SMoLA$_{\texttt{O}}^{48}$-PaLI-$3_{\texttt{FT}}$ with a resolution of 812 in batch mode (size 128) using beam decoding (beam size 4) We use COCO caption as the evaluation task where the length of outputs are around 10 tokens on average. We sample 18 forward batches to compute the statistics. PaLI-$3_{\texttt{FT}}$ processed 31.29 ± 0.63 examples per second and SMoLA$_{\texttt{O}}^{48}$-PaLI-$3_{\texttt{FT}}$ processed 30.85 ± 0.70 examples per second, yielding only 1.4% slow-down when using 48 experts in each SMoLA block, on all linear layers in the PaLI encoder.

4.3.6 Effects of different training mixtures

We evaluate SMoLA on the VQA and captioning mixture with PaLI-3 in order to exam its effectiveness when all training tasks are under the same umbrella of either VQA or captioning. We adopt a resolution of 1064 for the VQA mixture and 812 for the captioning mixture, and finetune the PaLI-3 raw checkpoints on each mixture as baselines.

VQA Mixture As shown in Table 4, while still underperforming the per-task fine-tuned specialist models by 0.46, SMoLA improves over the PaLI-$3_{\texttt{FT}}$ baseline by +1.07 on average. In particular, it helps the base model on most of the tasks with and without OCR inputs except for a 0.2 performance drop on OK-VQA. The significant performance improvements over the PaLI-$3_{\texttt{FT}}$ baseline on InfoVQA and ChartQA persist as observed when training with the full mixture. Furthermore, it also helps the PaLI-3-based model to achieve a new SOTA result of 82.2 on TextVQA.

Image Captioning Mixture Table 6 summarizes results of applying SMoLA to PaLI-$3_{\texttt{LoRA}}$ and PaLI-$3_{\texttt{FT}}$ using the image captioning mixture. In these experiments, we use a resolution of 812. We observe similar trends as in the case of using the full mixture: PaLI-$3_{\texttt{FT}}$ outperforms PaLI-$3_{\texttt{LoRA}}$ on average, indicating the insufficiency of LoRA tuning on a wide range of tasks; and SMoLA helps both the full model fine-tuned and LoRA baseline achieve better average performance. The SMoLA$_{\texttt{O}}^{48}$-PaLI-$3_{\texttt{FT}}$ outperforms the per-task fine-tuned specialist models by 0.64 on average, and sets a new SOTA for a generalist image-captioning system.

Model	COCO	TextCap		VizWizCap		Avg.
		ocr×	*ocr*✓	*ocr*×	*ocr*✓	δ
	K. test	val	val	test	test	
PaLI-3 Specialist	145.9	158.8	161.0	119.6	120.3	0.0
PaLI-$3_{\texttt{LoRA}}$	143.6	158.6	161.3	118.8	120.5	-0.56
SMoLA$_{\texttt{O}}^{48}$-PaLI-$3_{\texttt{LoRA}}$	143.9	160.6	**162.6**	119.1	120.8	+0.28
PaLI-$3_{\texttt{FT}}$	145.0	**159.9**	160.9	120.3	**120.9**	+0.28
SMoLA$_{\texttt{O}}^{48}$-PaLI-$3_{\texttt{FT}}$	**146.5**	159.5	161.7	**120.5**	120.6	**+0.64**

Table 6. Generalist results using the image captioning mixture. Bold and underlined numbers highlight best performance and best generalist performance, respectively.

5. Conclusion

In this work, we present Omni-SMoLA, a multimodal architecture that mixes many multi-modal experts efficiently and achieves both high specialist and generalist performance. In contrast to previous models for which we see performance degradation on average when training the models on a wide range of tasks, we show that the SMoLA low-rank experts are able to model different skills and task, and overall improve the performance of a generalist model. This finding indicates that simple LMM fine-tuning is suboptimal for handling a wide range of tasks, and that pairing the act of fine-tuning with specifically-designed architecture changes leads to better performing models.

[5] We use test split for AI2D as there are only 120 examples in val split.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 5

[2] Ankur Bapna and Orhan Firat. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019. 2

[3] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 4

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901. Curran Associates, Inc., 2020. 2

[5] Rich Caruana. Multitask learning. *Machine learning*, 28: 41–75, 1997. 2

[6] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023. 1, 2, 4, 5, 8

[7] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023. 2, 4, 5, 8

[8] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[10] Aaron Daniel Cohen, Adam Roberts, Alejandra Molina, Alena Butryna, Alicia Jin, Apoorv Kulshreshtha, Ben Hutchinson, Ben Zevenbergen, Blaise Hilary Aguera-Arcas, Chung ching Chang, Claire Cui, Cosmo Du, Daniel De Freitas Adiwardana, Dehao Chen, Dmitry (Dima) Lepikhin, Ed H. Chi, Erin Hoffman-John, Heng-Tze Cheng, Hongrae Lee, Igor Krivokon, James Qin, Jamie Hall, Joe Fenton, Johnny Soraker, Kathy Meier-Hellstern, Kristen Olson, Lora Mois Aroyo, Maarten Paul Bosma, Marc Joseph Pickett, Marcelo Amorim Menegali, Marian Croak, Mark Díaz, Matthew Lamm, Maxim Krikun, Meredith Ringel Morris, Noam Shazeer, Quoc V. Le, Rachel Bernstein, Ravi Rajakumar, Ray Kurzweil, Romal Thoppilan, Steven Zheng, Taylor Bos, Toju Duke, Tulsee Doshi, Vincent Y. Zhao, Vinodkumar Prabhakaran, Will Rusch, YaGuang Li, Yanping Huang, Yanqi Zhou, Yuanzhong Xu, and Zhifeng Chen. Lamda: Language models for dialog applications. In *arXiv*. 2022. 2

[11] 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. *arXiv preprint arXiv:2305.06500*, 2023. 5, 8

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 2

[13] 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. In *International Conference on Learning Representations*, 2021. 2

[14] Danny Driess, F. Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Ho Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Peter R. Florence. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023. 1, 2, 5

[15] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi

Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022. 2

[16] Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^{3}vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35:28441–28457, 2022. 2

[17] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021. 2

[18] William Fedus, Barret Zoph, and Noam M. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2021. 2

[19] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 4

[20] 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. 4

[21] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *Computer Vision – ECCV 2020*, pages 417–434, Cham, 2020. Springer International Publishing. 4

[22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2

[23] 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. *arXiv preprint arXiv:2106.09685*, 2021. 3

[24] 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. In *International Conference on Learning Representations*, 2022. 2

[25] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991. 2

[26] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1993. 2

[27] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 4

[28] Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. *ArXiv*, abs/1603.07396, 2016. 4

[29] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2

[30] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam M. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *ArXiv*, abs/2006.16668, 2020. 2, 4

[31] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2

[32] Chunyuan Li. Large multimodal models: Notes on cvpr 2023 tutorial. *arXiv preprint arXiv:2306.14895*, 2023. 2

[33] 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. 1, 2

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 2

[35] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019. 2

[36] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-IO: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 5

[37] 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. 4

[38] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of ACL*, 2022. 4

[39] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 4

[40] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 4

[41] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 4

[42] Jonas Pfeiffer, Andreas Ruckle, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and

Iryna Gurevych. Adapterhub: A framework for adapting transformers. Association for Computational Linguistics, 2020. 2

[43] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023. 1, 3

[44] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021. 2

[45] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 2

[46] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2

[47] Noam M. Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ArXiv*, abs/1701.06538, 2017. 2

[48] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. Mixture-of-experts meets instruction tuning:a winning combination for large language models, 2023. 1, 2

[49] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. TextCaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758, 2020. 4

[50] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 4

[51] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021. 2

[52] Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, et al. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860, 2023. 2

[53] 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. 1, 2

[54] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *TMLR*, 2022. 5

[55] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *CoRR*, abs/2208.10442, 2022. 3, 5

[56] Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning, 2022. 2

[57] Yaqing Wang, Jialin Wu, Tanmaya Dabral, Jiageng Zhang, Geoff Brown, Chun-Ta Lu, Frederick Liu, Yi Liang, Bo Pang, Michael Bendersky, et al. Non-intrusive adaptation: Input-centric parameter-efficient fine-tuning for versatile multimodal modeling. *arXiv preprint arXiv:2310.12100*, 2023. 2

[58] Wenmeng Yu Wenyi Hong Ji Qi Yan Wang Junhui Ji Zhuoyi Yang Lei Zhao Xixuan Song Jiazheng Xu Bin Xu Juanzi Li Yuxiao Dong Ming Ding Jie Tang Weihan Wang, Qingsong Lv. Cogvlm: Visual expert for pretrained language models, 2023. 5, 8

[59] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. mplug-2: A modularized multi-modal foundation model across text, image and video. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, pages 38728–38748. PMLR, 2023. 5

[60] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning, 2023. 2

[61] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. Taming sparsely activated transformer with stochastic experts. *arXiv preprint arXiv:2110.04260*, 2021. 2

A. Comparison between SMoLA and LoRA

We study the effect of scaling up the expert counts for SMoLA$^{\mathtt{E}}_{\mathtt{O}}$-PaLI-3$_{\mathtt{FT}}$ versus using a higher rank of LoRA on top of PaLI-3$_{\mathtt{FT}}$. The metrics are averaged across the tasks presented in Table 3 in the main paper on the validation splits[6] except for InfoVQA. With 4, 16, 48, and 144 experts per modality, SMoLA$^{\mathtt{E}}_{\mathtt{O}}$-PaLI-3$_{\mathtt{FT}}$ models achieve 1.1, 1.14, 1.2, 1.27 improvements over the PaLI-3$_{\mathtt{FT}}$ baseline.

We experiment with another LoRA tuning stage on PaLI-3$_{\mathtt{FT}}$ with the rank $\mathtt{r}$ of 32, 128, 384, 1536 that have the same extra expert model parameters with SMoLA$^{\mathtt{r/4}}_{\mathtt{MM}}$-PaLI-3$_{\mathtt{FT}}$ and share the same extra compute with SMoLA$^{\mathtt{r/8}}_{\mathtt{O}}$-PaLI-3$_{\mathtt{FT}}$. These LoRA tuning brought 0.58, 0.86, 0.79 and 0.59 gain over the PaLI-3$_{\mathtt{FT}}$ baseline. The performance is saturated with a rank of 128 and further scaling up rank leads to worse performance, where SMoLA provides an alternative expert count scaling dimension that achieves better overall performance.

A.1. Effective Rank for LoRA

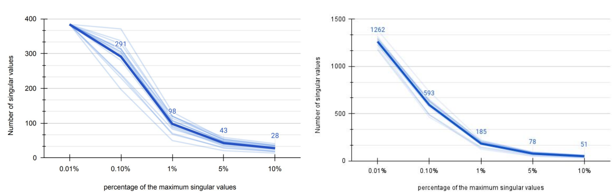

Figure 3. Number of singular values that are greater than different thresholds for LoRA weights. The left is for LoRA with rank 384 and the right is for LoRA with rank 1536.

We investigate on how many ranks for LoRA have the potentials of highly contributing to the outputs to better understand the reason that a higher rank does not improve the overall results. Specifically, we compute the singular values for the combined weights of $\mathtt{W}^{\mathtt{out}}\mathtt{W}^{\mathtt{in}}$ and present that for the LoRA with rank 384 and 1536 on the output layer of each attention module for the encoder blocks in PaLI-3$_{\mathtt{FT}}$ in Figure 3. The light colored lines plot the number of singular values that are greater than 0.01%, 0.1%, 1%, 5%, 10% of the largest singular value for all the encoder blocks and the bold line plots the averaged singular value counts. For LoRA with rank 1536, there are only about 78 (5%) and 185 (12%) singular values greater than 5% and 1% of the largest singular value, the remaining dimensions would make little contribution to the output with normalized inputs.

B. Visualization on the routing matrix Φ

We present a heat map visualization of $\Phi\Phi^T$ in Figure 4. The more similar $\Phi\Phi^T$ to an identity matrix, the more potential that the routing matrix Φ routes the input to different experts. The observations are (a) the $\Phi\Phi^T$ for text experts less closer to the identity matrices, indicating that some experts would have more similar routing processes. This is primarily because we consider VQA with short questions and image captioning with the same prompt where text queries are easy to process. (b) the $\Phi\Phi^T$ for early layers in vision and multimodal experts are more closer to the identity matrices, indicating the demand of SMoLA design to handle more diverse shallow representations.

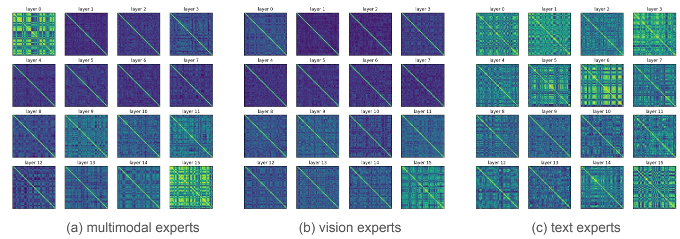

Figure 4. Heat maps for $\Phi\Phi^T$ for different modalities in SMoLA$^{\mathtt{48}}_{\mathtt{O}}$-PaLI-3$_{\mathtt{FT}}$

[6] We use test split for AI2D as there are only 120 examples in val split.