arXiv:2303.02506v1 [cs.LG] 4 Mar 2023

Prismer: A Vision-Language Model with An Ensemble of Experts

Shikun Liu[1,2*] Linxi Fan[2] Edward Johns[1]
Zhiding Yu[2] Chaowei Xiao[2,3] Anima Anandkumar[2,4]

[1]Imperial College London [2]NVIDIA [3]ASU [4]Caltech

Abstract

Recent vision-language models have shown impressive multi-modal generation capabilities. However, typically they require training huge models on massive datasets. As a more scalable alternative, we introduce Prismer, a data- and parameter-efficient vision-language model that leverages an ensemble of domain experts. Prismer only requires training of a small number of components, with the majority of network weights inherited from readily-available, pre-trained domain experts, and kept frozen during training. By leveraging experts from a wide range of domains, we show that Prismer can efficiently pool this expert knowledge and adapt it to various vision-language reasoning tasks. In our experiments, we show that Prismer achieves fine-tuned and few-shot learning performance which is competitive with current state-of-the-art models, whilst requiring up to two orders of magnitude less training data. Code is available at `https://github.com/NVlabs/prismer`.

1 Introduction

Large pre-trained models have demonstrated exceptional generalisation capabilities across a wide range of tasks. However, these capabilities come at a hefty cost in terms of computational resources required for training and inference, as well as the need for large amounts of training data. In the language domain, models with hundreds of billions of learnable parameters typically require a compute budget on the yottaFLOP scale [18, 8, 7, 65].

The problems in vision-language learning are arguably more challenging. This domain is a strict super-set of language processing, whilst also requiring extra skills unique to visual and multi-modal reasoning. For example, many image captioning and visual question answering problems require the model to be capable of fine-grained object recognition, detection, counting, and 3D perception [4, 14]. A typical solution is to use a massive amount of image-text data to train one giant, monolithic model that learns to develop these modality-specific skills from scratch, simultaneously, and within the same generic architecture.

*Corresponding Author: shikun.liu17@imperial.ac.uk. Work done during an internship at NVIDIA.

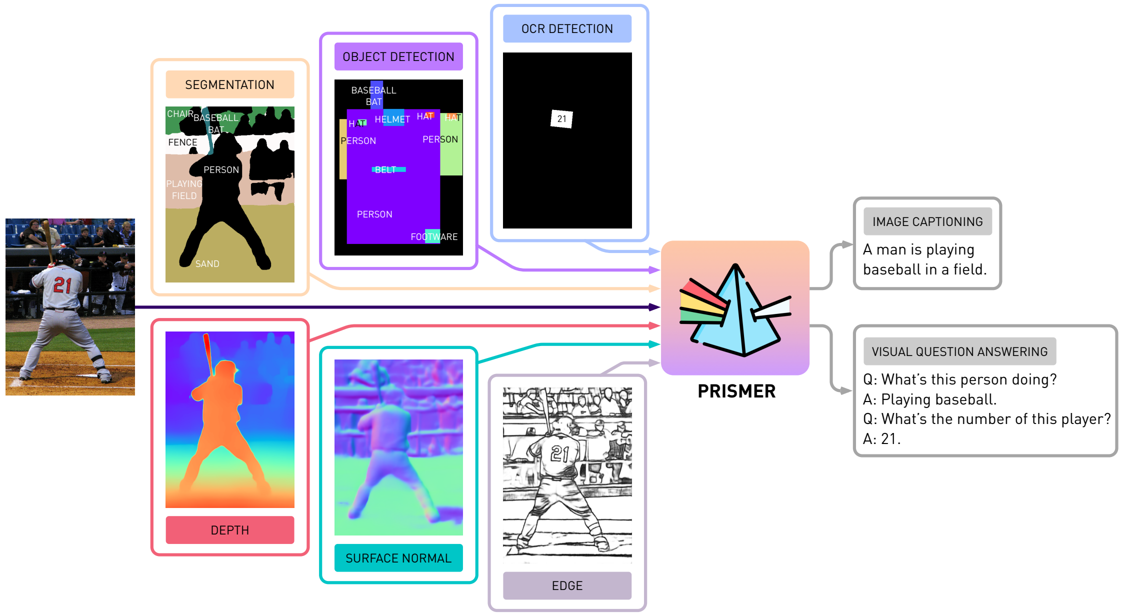


Figure 1: **Prismer model overview.** Prismer is a data-efficient vision-language model that leverages diverse pre-trained experts through its predicted multi-modal signals. It can perform vision-language reasoning tasks such as image captioning and visual question answering. The analogy is with an optical prism: Prismer splits a single reasoning task into diverse domain-specific reasoning.

Instead, we investigate an alternative approach to learn these skills and domain knowledge via *distinct and separate sub-networks*, referred to as "experts". As such, each expert can be optimised independently for a specific task, allowing for the use of domain-specific data and architectures that would not be feasible with a single large network. This leads to improved training efficiency, as the model can focus on *integrating* specialised skills and domain knowledge, rather than trying to learn everything at once, making it an effective way to *scale down* multi-modal learning.

To achieve this, we propose Prismer[1], a visually conditioned autoregressive text generation model, trained to *better use diverse pre-trained domain experts* for open-ended vision-language reasoning tasks. Prismer's key design elements include i) powerful vision-only and language-only models for *web-scale knowledge* to construct our core network backbones, and ii) modality-specific vision experts encoding multiple types of visual information, including *low-level vision signals* such as depth, and *high-level vision signals* such as instance and semantic labels, as a form of *auxiliary knowledge*, directly from their corresponding network outputs. All expert models are individually pre-trained and frozen, and are connected through some lightweight trainable components which contribute to roughly 20% of the total network parameters.

Despite Prismer being trained on only 13M publicly available image/alt-text data examples, it shows strong multi-modal reasoning performance in tasks such as image captioning, image

[1]The model name "Prismer" draws from the analogy to an optical prism which breaks a white light into a spectrum of colours, and here we break down a single reasoning task into diverse domain-specific reasoning.

classification, and visual question answering, competitive with many state-of-the-art vision-language models [3, 80, 82], that were trained with one or two orders of magnitude more data. Finally, we conduct an in-depth analysis of Prismer's learning behaviours and observe some encouraging properties. For example, i) Prismer exhibits *strong robustness against the inclusion of noisy experts*, and ii) the learning performance also scales favourably with increases in both the *quantity* or *quality* of experts.

2 Related Work

Vision-Language Models (VLMs) Inspired by the breakthrough of transformers in the language domain [78, 23], early works aimed to model the vision-language relationship using a shared network based on transformers in a *single-stream* design [1, 15, 45, 74]. These works usually leverage a pre-trained object detector, encoding images as sequences of *visual words*, parameterised by object- or region-level features. Prismer takes a slightly different approach by using pre-trained models to provide their output predictions as auxiliary signals, whilst still relying on the original images to encode visual features.

Another line of works encodes vision and language features in separate networks in a *dual-stream* design, where the vision-only and language-only features are aligned through contrastive learning [64, 89, 34, 43]. These works typically focus on close-ended multi-modal alignment tasks such as image-text classification and retrieval. In contrast, Prismer's vision encoder also aligns its vision features with the language embedding through pre-training with contrastive learning, but with a greater emphasis on multi-modal generation tasks.

Both single- and dual-steam VLMs in the past years have often been pre-trained with a combination of multiple objectives, such as masked language modelling, masked region modelling, word-region alignment, visual grounding and more [1, 17, 42, 43, 54]. These multiple objectives can make the training process more complex and require careful balancing of the different losses. Prismer adopts a different approach, aligning with recent developments in VLMs that focus on language generation, and only require a single autoregressive training objective [80, 82, 32]. Despite the reduced complexity, training these large-scale VLMs is data intensive and computationally demanding, often requiring billions of training data. To overcome these challenges, Prismer leverages powerful pre-trained domain expert models for data-efficient training. Unlike another set of works that prioritise in-context capability by conditioning on a large frozen language model with no task-specific fine-tuning [26, 77, 3], Prismer focuses on fine-tuned performance with an emphasis on parameter efficiency, using smaller but diverse pre-trained models.

Multi-task and Auxiliary Learning Multi-task learning and auxiliary learning aim to train models to predict multiple modalities (such as semantic segmentation, object detection, and depth estimation) from a single input, thereby improving the performance across one or multiple tasks. This is often achieved through the design of effective multi-task networks that balance task-shared and task-specific features [50, 56, 75, 85], or through the explicit modelling of task relationships [48, 49, 57, 87, 27]. Prismer also employs multiple modalities, similar to these methods, but only uses them solely as input, serving as auxiliary knowledge. Prismer is more related to works such as [5, 29], which utilise pre-trained experts to create pseudo labels for multi-task self-training. However, whilst those methods focus on learning task-agnostic features through multi-task supervision, Prismer focuses purely on multi-modal reasoning with a single-task objective.

Unifying Pre-trained Experts The utilisation of diverse pre-trained domain experts for multi-modal reasoning has been investigated in previous studies. Socratic models [88] use language as a one-way communication interface to connect different pre-trained experts. However, this design is limited to multi-modal reasoning within the domains on which the pre-trained experts were trained, and errors predicted by previous experts can be carried forward to future experts. On the other hand, PIC [44] addresses this issue by using a group of pre-trained experts to evaluate each expert's prediction and reach a consensus through an iterative closed-loop communication. Whilst both methods perform multi-modal reasoning in a zero-shot manner without any training, Prismer utilises a unified architecture design to enhance the integration and sharing of information between the pre-trained experts.

Finally, we would like to note the difference between "Mixture of Experts (MoE)" [68, 59, 55] and "Ensemble of Experts" defined in Prismer. In MoE, the "experts" are sub-modules in a single network, interconnected through their corresponding gating networks, encoding *implicit knowledge* guided by a shared training objective. Conversely, in Prismer, the "experts" are independently pre-trained models, encoding *explicit knowledge* based on their pre-trained tasks or domains.

3 Prismer: Open-ended Reasoning with Multi-modal Knowledge

In this section, we introduce the Prismer model, a type of vision-language generative model that takes multi-modal signals as input, and outputs free-form text.

3.1 Model Overview

The design of the Prismer model is illustrated in Fig. 2. Prismer is an encoder-decoder transformer model [78] that leverages a library of existing pre-trained experts. It consists of a vision encoder and an auto-regressive language decoder. The vision encoder takes an RGB image and its corresponding multi-modal labels as input (*e.g.* depth, surface normal, segmentation labels, predicted from the frozen pre-trained experts), and outputs a sequence of RGB and multi-modal features. The language decoder is then conditioned on these multi-modal features via cross attention, and produces a sequence of text tokens.

One of the key advantages of the Prismer model is its exceptional data efficiency during training. This is achieved by leveraging *a combined power of strong domain-specific experts*, resulting in a significant reduction in the number of GPU hours required to achieve comparable performance to other state-of-the-art vision-language models. Prismer is built on top of existing pre-trained vision-only and language-only backbone models — this allows us to tap into the vast amount of *web-scale knowledge* already stored in these pre-trained parameters. Additionally, we also extend the vision encoder to accept multi-modal signals — this enables it to better capture semantics and information about the input image through the help of the *generated multi-modal auxiliary knowledge*. For example, we expect "text-reading" problems can

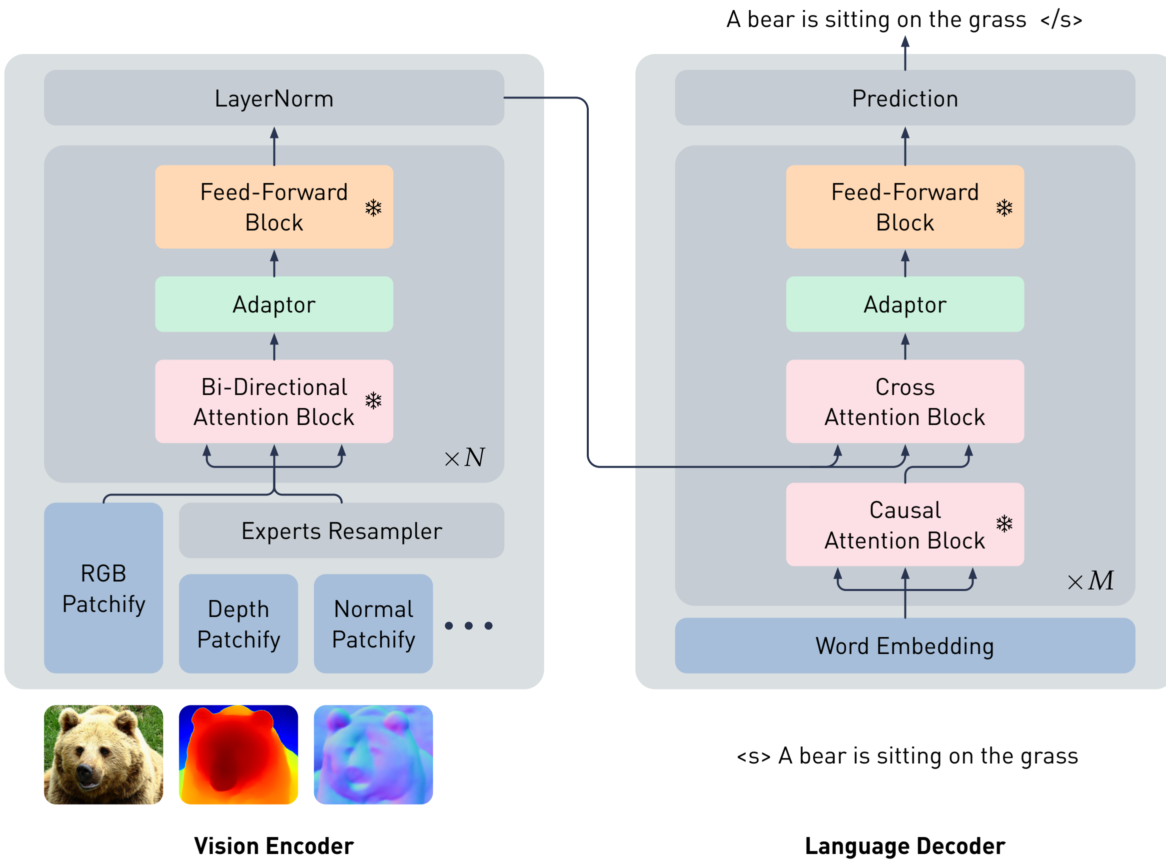


Figure 2: **Prismer architecture design overview.** Prismer has two main trainable components: the Experts Resampler that converts variable multi-modal signals to a fixed number of outputs, and the Adaptor that enhances the model's expressivity for vision-language reasoning. To ensure that the model takes advantage of the rich domain-specific knowledge encoded in the pre-trained experts, the majority of network weights are frozen during training, as represented by ❄.

be easily solved by leveraging an OCR detection expert; and "object-recognition" problems can be easily solved by leveraging an object detection expert. A visualisation of all expert labels we included in Prismer is shown in Fig. 1 and is further explained in Sec. 3.2.

Prismer is designed to fully leverage pre-trained experts whilst keeping the number of trainable parameters to a minimum. To do this, the majority of the network weights of the pre-trained experts are frozen to maintain the *integrity of their learned knowledge* and prevent catastrophic forgetting [38, 39]. To link the multi-modal labels as well as the vision and language parts of Prismer, we insert two types of parameter-efficient trainable components: *Experts Resampler* and *Adaptor*. The Experts Resampler is used in the vision encoder to map a variable length of multi-modal signals to a sequence of multi-modal features with a *fixed length*. The Adaptors are inserted in each transformer layer of the vision and language parts of the model to better adapt the pre-trained experts to new tasks and modalities.

Prismer is a *generative* model, and we re-formulate all vision-language reasoning tasks as a *language modelling or prefix language modelling* problem. For example, given the multi-modal tokens (encoded from an input RGB image and its multi-modal labels) and a question as the prefix, the model generates the answer for the visual question answering task; given the multi-modal tokens, the model generates its caption for the image captioning task. Once we have a prefix prompt, we may either sample the output text in an autoregressive manner, as in an *open-ended* setting; or we may rank the log-likelihood from a fixed set of completions, as in a *closed-ended* setting.

3.2 Pre-trained Experts

In Prismer, we include two types of pre-trained experts:

Backbone Experts The vision-only and language-only pre-trained models, which are responsible for encoding images and texts into a meaningful sequence of tokens. Both models are required to be based on the transformer architecture [78], so we that can easily connect them with a few trainable components of similar designs. To preserve their rich domain-specific knowledge encoded in the network parameters, the majority of the weights are frozen during pre-training.

Modality Experts The models that can produce task-specific labels depending on their training datasets. In Prismer, we include up to 6 modality experts all from the vision domain, encoding three *low-level* vision signals: depth, surface normals, and edge; and three *high-level* vision signals: object labels, segmentation labels, and text labels. These modality experts are treated as *black-box predictors*, and their predicted labels are used as input for the Prismer model. As a result, all network weights of the modality experts are frozen, and they can have *any design*.

We apply modality-specific post-processing on these predicted labels, transforming them to a $\mathbb{R}^{H\times W\times C}$ tensor (here H, W, C represent image height, width and channels respectively. *e.g.* $C = 1$ for depth and edge label, and $C = 3$ for surface normals label). For all expert labels encoding high-level semantic signals, we tile each pixel with its corresponding text embedding from a pre-trained CLIP text model [64], and then we further apply PCA to down-sample the dimensionality to $C = 64$ for efficient training. The detailed descriptions of all modality experts, including their pre-trained datasets and the architecture design, are listed in Table 1.

3.3 Key Architectural Components

Modality-Specific Convolutional Stem All expert labels are first processed with randomly initialised convolution layers to map them to the same dimensionality. Specifically, we apply 5 convolutional layers and each is composed of a small $[3 \times 3]$ kernel, which is shown to perform better than a single convolutional layer but with a larger kernel in the original Vision Transformer design [24], consistent with the finding in [84]. The convolutional stem is designed to be modality-specific, which we have found to yield superior performance in comparison to a shared design in a multi-task learning setting [50, 56].

Task	Dataset	Model	Params.	Post-Processing
Semantic Segmentation	COCO-Stuff [9]	Mask2Former [16]	215M	Tile each pixel with its corresponding label parametrised by CLIP text embedding.
Object Detection	COCO [46] + Objects365 [71] + OpenImages [41] + Mapillary [58]	UniDet [92]	120M	Tile each pixel with its corresponding label parametrised by CLIP text embedding. The labels for the overlapping pixels are further determined by the depth expert.
Text Detection	ICDAR 2015 [36]	CharNet [51]	89M	Tile each pixel with its corresponding text parametrised by CLIP text embedding.
Depth Estimation	MIX-6 [67]	DPT [67]	123M	Re-normalised to $[-1, 1]$.
Surface Normal	ScanNet [20]	NLL-AngMF [6]	72M	Re-normalised to $[-1, 1]$.
Edge Detection	BIPED [63]	DexiNed [63]	35M	Re-normalised to $[-1, 1]$.

Table 1: **The detailed description of modality experts**. We provide a detailed description of each modality expert including its pre-trained dataset, parameter size, model name and type and post-processing strategy.

For high-level semantic labels such as those in object detection, semantic segmentation, and OCR detection, we down-sample the resolution by a factor of 4 to conserve running memory. Furthermore, for each object instance, we add a trainable and randomly sampled embedding to distinguish among different object instances. The size of this instance embedding is set to 128, which corresponds to the maximum possible number of object instances to be present in a single image. For RGB images, we simply process with the pre-trained convolutional stem defined by the original vision backbone. All modality expert embeddings, including RGB, are then added with a pre-trained positional embedding before being further processed by transformer layers.

Experts Resampler The computational complexity of self-attention is *quadratically proportional* to the number of input patches. And therefore, the vision encoder can easily require tremendous memory when including a large number of modality experts. To address this issue, we propose *Experts Resampler*, which takes a *variable* number of expert labels as input and outputs a *fixed* number of embeddings, illustrated in Fig. 3 Left. Such design produces a *constant* memory for the self-attention computation in the vision encoder, as well as the vision-text cross attention in the language decoder (shown in Fig. 2), independent of the inclusion of a different number of experts. Inspired by the design in the Perceiver [33] and the Flamingo model [3], the Experts Resampler learns a pre-defined number of latent input queries, to cross-attend a flattened embedding concatenated from all multi-modal features. The Resampler then compresses the multi-modal features into a much smaller number of tokens equal to the number of learned latent queries, as a form of *auxiliary knowledge distillation*. We design keys and values to be a concatenation for both multi-modal features and the learned latent queries, which is shown to be more effective, consistent with the design in the Flamingo model [3].

Lightweight Adaptor We insert one lightweight adaptor into each transformer layer of both vision and language backbones in order to improve Prismer's expressivity and conditioning on multi-modal features, illustrated in Fig. 3 Right. The adaptor has an encoder-decoder

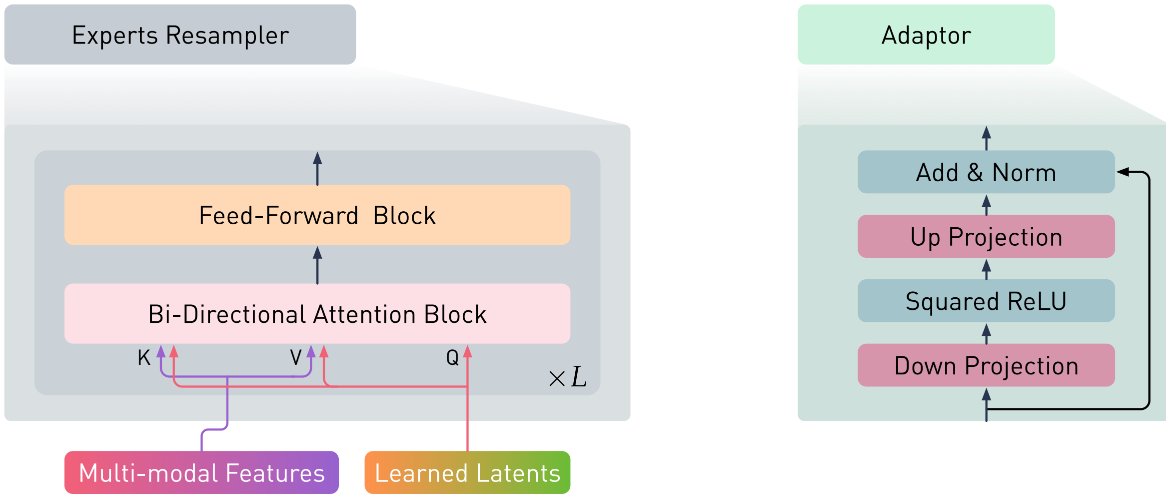


Figure 3: **Design details in Experts Resampler and Adaptor.** Left: The Experts Resampler takes multi-modal features with variable length as input, and outputs a fixed number of tokens via cross attention. Right: The Adaptor has a residual connection to the input and two fully-connected layers, that down-projects the input features to a smaller bottleneck dimension and then up-projects back to the original dimension.

design, which has proven to be successful for efficient transfer learning in the NLP domain [31, 62]. It first down-projects the input features into a smaller dimension, applies a non-linearity, and then up-projects the features back to the original input dimension. We choose the non-linearity function to be squared ReLU [73] – a simple and parameter-free function that delivers strong training stability. With the residual connection, we initialise all adaptors with near-zero weights to approximate the identity function. Combined with a standard cross attention block in the language decoder, the model is able to smoothly transition from the domain-specific vision-only and language-only backbones to a vision-language model during pre-training with paired image-text data.

The model performance, memory usage and time complexity for other design choices are systematically evaluated and ablated in Sec. 5.2.

3.4 Training Objective

For simplicity, we train Prismer with a *single* objective — to predict the next text token autoregressively. Following the standard encoder-decoder architecture, the vision encoder predicts the multi-modal features z, and the language decoder learns to maximise the conditional likelihood of the paired text caption y with its length T under the forward autoregressive factorisation: $L = -\sum_{t=1}^{T} \log p(y_t | y_{<t}, z)$.

In practice, our *one-time* pre-processing step of collecting multi-modal expert labels is computationally cheap and fast with data parallelism. The single generative objective then only requires one forward pass to compute gradients, which is significantly more efficient and streamlined than many other VLMs that may require a multi-stage and/or multi-step pre-training [42, 43, 81, 25, 15], with multiple objectives and data sources. However, because our model only focuses on multi-modal language generation, it is less suitable for multi-modal discriminative tasks such as image-text retrieval and visual entailment, which are the focus of other types of VLMs [28, 15, 34].

4 Experiments

4.1 Prismer Model Variants

In addition to Prismer, we also introduce a model variant named PrismerZ, which solely relies on the power of strong backbone experts and is trained with *zero* modality experts. PrismerZ has the same architectural design as the original Prismer but without the Experts Resampler. PrismerZ simplifies the data inference process as it only requires RGB images, making it more efficient and applicable to a wider range of applications. Prismer is less efficient in data inference due to the need for data processing on expert labels, but as we will show, it has better predictive performance.

Both Prismer and PrismerZ utilise ViT [24] pre-trained by CLIP [64] as the frozen vision encoder, and RoBERTa [52] as the frozen language decoder. We have alternatively tried using two other popular open-sourced decoder-only autoregressive language models: OPT [91] and BLOOM [69], but early experiments showed that they did not perform as well.

We experiment with two model sizes, `BASE` and `LARGE`. The `BASE` model is built on top of `ViT-B/16` and $\texttt{RoBERTa}_{\texttt{BASE}}$, and the `LARGE` model is built on top of `ViT-L/14` and $\texttt{RoBERTa}_{\texttt{LARGE}}$. In Prismer, we apply the same Experts Resampler with roughly 50M parameters in both model sizes. The detailed architecture details are summarised in Table 2.

	Resampler		Vision Encoder			Language Decoder			Trainable Params.	Total Params.
	Layers	Width	Backbone	Layers	Width	Backbone	Layers	Width		
$\text{Prismer}_{\texttt{BASE}}$	4	768	ViT-B/16	12	768	$\texttt{RoBERTa}_{\texttt{BASE}}$	12	768	160M	980M
$\text{Prismer}_{\texttt{LARGE}}$	4	1024	ViT-L/14	24	1024	$\texttt{RoBERTa}_{\texttt{LARGE}}$	24	1024	360M	1.6B
$\text{PrismerZ}_{\texttt{BASE}}$	-	-	ViT-B/16	12	768	$\texttt{RoBERTa}_{\texttt{BASE}}$	12	768	105M	275M
$\text{PrismerZ}_{\texttt{LARGE}}$	-	-	ViT-L/14	24	1024	$\texttt{RoBERTa}_{\texttt{LARGE}}$	24	1024	270M	870M

Table 2: **Prismer and PrismerZ architecture details.** We report the backbone we choose for each architecture size, along with its corresponding number of layers and width. We also report the number of trainable parameters and total parameters for each architecture. We count the total parameters required for data inference, which include the additional 6 modality experts with a combined parameter size of 654M parameters in our Prismer model.

4.2 Training and Evaluation Details

Pre-training Datasets We construct our pre-training data from the following datasets: two in-domain datasets: COCO [46] and Visual Genome [40]; and three web datasets: Conceptual Captions [72], SBU captions [61], and a much noisier Conceptual 12M [10]. The web datasets are pre-filtered and re-captioned by a pre-trained image captioner [42]. The pre-training datasets include 11M unique images or 12.7M image/alt-text pairs.[2] All datasets are available publicly and have been widely used for pre-training many VLMs [43, 42, 15].

Optimisation and Implementation All our models are trained with AdamW optimiser [53] with a weight decay of 0.05. Since only a small proportion of the model parameters are trainable, model sharding is only applied during fine-tuning on large-resolution images. Specifically, we employ ZeRO Stage 2 technique [66], which enables the sharding of optimiser states and parameter gradients across all GPU instances. Additionally, we also apply Automatic Mixed Precision (AMP) with `fp16` precision to further reduce training time. For more details on our data processing techniques and hyper-parameter choices, please refer to Appendix A. An analysis of training costs compared to other vision-language models can be found in Appendix B.

Evaluation Setting We evaluate the performance of our models through language modelling, which is a more challenging task than discriminative learning (particularly in VQA tasks), and aligns with that used in other vision-language generative models [42, 3, 80, 13]. For example, the model must accurately generate all text tokens for a question (which is on average 2.2 tokens per question in the VQAv2 dataset [4] as reported in [80]), rather than just one correct prediction as required in discriminative models.

Specifically, we evaluate image captioning tasks in an open-ended setting, and we apply beam search with a beam size of 3 for text generation. A prefix prompt of "A picture of" is added to the input text for fined-tuned image captioning tasks, similar to previous studies such as in [82, 42, 64], which was shown to improve the quality of image captions. We evaluate both VQA and image classification tasks in a close-ended setting, by ranking the per-token log-likelihood from a pre-defined answer list.

4.3 Results on Vision-Language Benchmarks

Fine-tuned Performance on COCO Caption, NoCaps and VQAv2 We fine-tune our models on COCO Caption dataset [14] on a widely adopted Karpathy split [37], with the standard cross-entropy loss, and without metric-specific optimisation [79]. We evaluate the fine-tuned models on the COCO Caption Karpathy test split and NoCaps [2] validation set. We also evaluate our models on the VQAv2 dataset [4], with additional training samples from Visual Genome [40] following [42]. We compare our models with prior state-of-the-art VLMs that are mostly pre-trained on image-text data for a fair comparison. We sort all VLMs by their model sizes and report the results in Table 3.

The results show that both Prismer and PrismerZ achieve superior performance considering their model sizes, which suggests that the strong backbone experts are primarily responsible for good generalisation. However, the modality experts provide an additional boost in performance, particularly in image captioning tasks (such as a 6 CIDEr score increase in the NoCaps out-of-domain set in the `BASE` model) and in the `LARGE` model variant (such as a 1 VQAv2 accuracy increase in the `LARGE` model). Both Prismer$_{\texttt{BASE}}$ and Prismer$_{\texttt{LARGE}}$

[2] This is slightly less than the theoretical number which should be 14M unique images. It is because some image URLs in the web datasets are not valid during the time we downloaded the datasets.

	Pre-train (# Pairs)	COCO Caption				NoCaps				VQAv2	
		B@4	M	C	S	In	Near	Out	Overall	test-dev	test-std
OSCAR$_{\texttt{BASE}}$ [45]	6.5M	36.5	30.3	123.7	23.1	83.4	81.6	77.6	81.1	73.2	73.4
VinVL$_{\texttt{BASE}}$ [90]	8.9M	38.2	30.3	129.3	23.6	103.7	95.6	83.8	94.3	76.0	76.1
GIT$_{\texttt{BASE}}$ [80]	10M	**40.4**	30.0	131.4	23.0	100.7	97.7	89.6	96.6	72.7	-
BLIP$_{\texttt{BASE}}$ [42]	129M	39.7	-	133.3	-	**111.8**	**108.6**	111.5	**109.6**	**78.3**	**78.3**
LEMON$_{\texttt{BASE}}$ [32]	200M	40.3	30.2	133.3	23.3	107.7	106.2	107.9	106.8	-	-
PrismerZ$_{\texttt{BASE}}$	12.7M	39.7	31.1	133.7	24.1	108.7	107.8	105.8	107.5	76.6	-
Prismer$_{\texttt{BASE}}$	12.7M	40.1	**31.1**	**135.1**	**24.1**	108.8	108.3	**111.7**	109.1	76.8	77.0
OSCAR$_{\texttt{LARGE}}$ [45]	6.5M	37.4	30.7	127.8	23.5	85.4	84.0	80.3	83.4	73.4	73.8
VinVL$_{\texttt{LARGE}}$ [90]	8.9M	38.5	30.4	130.8	23.4	-	-	-	-	76.5	76.6
GIT$_{\texttt{LARGE}}$ [80]	20M	**42.0**	30.8	**138.5**	23.8	107.7	107.8	102.5	106.9	75.5	-
BLIP$_{\texttt{LARGE}}$ [42]	129M	40.4	-	136.7	-	114.9	112.1	**115.3**	113.2	-	-
LEMON$_{\texttt{LARGE}}$ [32]	200M	40.6	30.4	135.7	23.5	**116.9**	**113.3**	111.3	**113.4**	-	-
PrismerZ$_{\texttt{LARGE}}$	12.7M	40.0	31.2	135.7	24.2	112.3	111.2	112.8	111.8	77.5	-
Prismer$_{\texttt{LARGE}}$	12.7M	40.4	**31.4**	136.5	**24.4**	114.2	112.5	113.5	112.9	**78.4**	**78.5**
LEMON$_{\texttt{HUGE}}$ [32]	200M	41.5	30.8	139.1	24.1	118.0	116.3	120.2	117.3	-	-
SimVLM$_{\texttt{HUGE}}$ [82]	1.8B	40.6	33.7	143.3	25.4	113.7	110.9	115.2	112.2	80.0	80.3
GIT [80]	0.8B	44.1	31.5	144.8	24.7	129.8	124.1	127.1	125.5	78.6	78.8
GIT-2 [80]	12.9B	44.1	31.4	145.0	24.8	126.9	125.8	130.6	126.9	81.7	81.9
CoCa [86]	4.8B	40.9	33.9	143.6	24.7	-	-	-	122.4	82.3	82.3
PaLI [13]	1.6B	-	-	149.1	-	-	-	-	127.0	84.3	84.3

Table 3: **Fine-tuned performance on COCO Caption (Karpathy split), NoCaps (validation set) and VQAv2.** Both Prismer and PrismerZ achieve superior performance in all three datasets compared to other VLMs with similar model sizes. Prismer can achieve competitive performance on par with VLMs that are trained with orders of magnitude more data. {B@4, M, C, S} refer to BLEU@4, METEOR, CIDEr, SPICE respectively. {In, Near, Out} refer to in-domain, near-domain and out-of-domain respectively.

achieve comparable image captioning performance to BLIP [42] and LEMON [32], despite being trained on 10 and 20 times less data, respectively. Additionally, the Prismer$_{\texttt{LARGE}}$ model has achieved VQAv2 accuracy comparable to GIT [80], despite being trained on 60 times less data. Whilst we acknowledge a noticeable performance gap between Prismer and the current state-of-the-art VLMs (such as CoCa [86], GIT-2 [80] and PaLI [13]), these models require substantially higher training costs and access to large-scale private training data.

Zero-shot Performance on Image Captioning Our generative pre-training approach allows for zero-shot generalisation, where the models can be directly applied to image captioning tasks without additional fine-tuning. In Fig. 4 Left, we show that Prismer achieves competitive performance on the NoCaps dataset compared to SimVLM [82], whilst using 140 times less training data. Additionally, we notice that the zero-shot performance of Prismer models even surpasses the fine-tuned performance of certain VLMs such as OSCAR [45] and VinVL [90], as shown in Table 3.

We present a list of example captions generated by Prismer in Table 4. The results show that both Prismer$_{\texttt{BASE}}$ and Prismer$_{\texttt{LARGE}}$ are capable of generating captions that are semantically coherent and aligned with the visual content of the images. Notably, Prismer$_{\texttt{LARGE}}$ generates captions of higher quality compared to Prismer$_{\texttt{BASE}}$, exhibiting a deep understanding of fine-grained object semantics such as brand recognition (*e.g.* Mercedes, CK One), and cultural concepts (*e.g.* vintage drawing, tango), indistinguishable to human-written captions.

	COCO Caption			
	B@4	M	C	S
ZeroCap [76]	2.6	11.5	14.6	5.5
MetaLM [30]	24.5	22.5	82.2	15.7
VLKD [21]	25.8	23.1	85.1	16.9
Flamingo [3]	-	-	84.3	-
CapDec [60]	26.4	25.1	91.8	-
Prismer$_{\texttt{BASE}}$	36.1	29.3	122.6	22.9
Prismer$_{\texttt{LARGE}}$	39.5	30.4	129.7	23.8

	NoCaps	
	C	S
FewVLM [35]	47.7	9.1
MetaLM [30]	58.7	8.6
VLKD [21]	63.6	12.8
SimVLM$_{\texttt{LARGE}}$ [82]	102.2	-
SimVLM$_{\texttt{HUGE}}$ [82]	110.4	-
Prismer$_{\texttt{BASE}}$	87.5	13.0
Prismer$_{\texttt{LARGE}}$	107.9	14.8

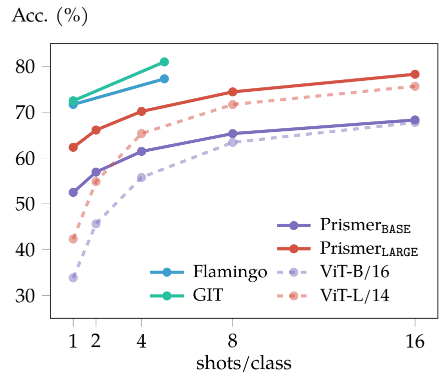


Figure 4: **Results on zero-shot image captioning and few-shot ImageNet classification.** Left: Prismer achieves state-of-the-art zero-shot image-captioning results on COCO Caption (Karpathy test) and NoCaps (validation set), on par with SimVLM, despite being trained on 160 times less data. Right: Prismer significantly improves few-shot performance compared to its corresponding vision backbone. However, Prismer still underperforms GIT and Flamingo which are trained on significantly more data.

Few-shot Performance on ImageNet Classification Finally, we fine-tune and evaluate Prismer on ImageNet dataset [22] in a few-shot setting. Following the approach outlined in [64], we convert the classification task into a language modelling problem by mapping each unique category to a template caption: "A photo of a [CLASS NAME]", and we then score all captions using the log-likelihood estimated by our model. Unlike Flamingo [3] which performs few-shot classification via in-context examples without gradient updates, we perform few-shot classification via lightweight fine-tuning following [80]. This is more similar to the standard linear probe setting, by considering the entire language decoder as an image classifier. Accordingly, we also compare with the few-shot linear probe performance of Prismer's original vision backbones ViT-B/16 and ViT-L/14 [24], as reported in [70, 64].

From the results shown in Fig. 4 Right, we observe that Prismer underperforms GIT [80] and Flamingo [3], which both have stronger vision backbones and are pre-trained on significantly more data. However, Prismer still outperforms its original vision backbones ViT-B and ViT-L by a large margin, especially in a very few-shot setting. This suggests that Prismer's generalisation abilities are enhanced by the multi-modal training data and expert labels, and its performance can likely be improved further by using an even stronger vision backbone.

5 Additional Analysis

We conduct experiments to probe Prismer carefully and discover some interesting abilities in Sec. 5.1. And we also ablate various architectural components and training strategies in Sec. 5.2. To speed up training, all experiments are conducted with the `BASE` model on a combined dataset of the Conceptual Captions and SBU, consisting of a total of 3M data. All experiments are evaluated on the VQAv2 `test-dev` split in a smaller $[224 \times 224]$ resolution.

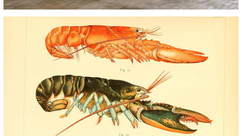
	Ground-Truth	Prismer$_{\texttt{BASE}}$	Prismer$_{\texttt{LARGE}}$
	*1. A clear bottle of CK cologne is full of liquid.* *2. The bottle of perfume is made by Calvin Klein.*	*A bottle of alcohol sitting next to a computer keyboard.*	*A bottle of ck one next to a computer keyboard.*
	*1. A statue has a large purple headdress on it.* *2. A woman decorated in fashioned clothing and relics.*	*The woman is wearing a black dress.*	*A mannequin dressed in a black dress with feathers on her head.*
	*1. A new white car with the door open is in a showroom full of people.* *2. A shiny white mercedes car is on display.*	*A white car on display at a car show.*	*A white mercedes car on display at an auto show.*
	*1. Large piece of meat with slices of pineapple with cherries being held on with toothpicks on blue and white plate.* *2. A cake has several slices of pineapple and cheries in them.*	*Pineapples on a plate.*	*Pineapple upside down cake on a blue and white plate.*
	*1. A man and woman is dancing as a crowd watches them in the distance.* *2. A woman in a red dress dancing with a bald man wearing black.*	*A couple of people that are standing in the dirt.*	*A couple dancing tango in front of a crowd.*
	*1. Two illustrations of lobster colors are shown as Fig. 21 and Fig. 22.* *2. A drawing of a lobster and a lobster.*	*Colored drawing of two lobsters on pink paper.*	*A vintage illustration of lobsters from the 19th century.*
	*1. Man in skydiving gear giving two thumbs up with skydivers in the sky behind him.* *2. Person giving double thumbs up sign while others are parachuting in the background.*	*Man wearing a blue and purple jacket.*	*A man wearing a helmet and goggles with parachutes in the background.*

Table 4: **Visualisation of zero-shot image captioning on NoCaps.** Prismer$_{\texttt{LARGE}}$ produces more detailed and semantically coherent captions than Prismer$_{\texttt{BASE}}$, showing an understanding of fine-grained object recognition and abstractions. Results are *not* cherry-picked.

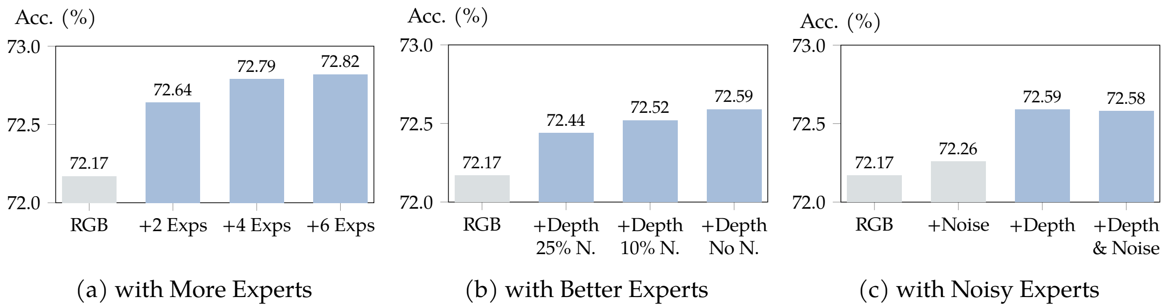


Figure 5: **Prismer's VQAv2 accuracy with different types and the number of experts.** Prismer has shown that its performance improves with an increase in the number and quality of modality experts. Additionally, Prismer also demonstrates its strong robustness to noisy experts, making it a practical and effective multi-modal learning strategy.

5.1 Intriguing Properties of Prismer

More Experts, Better Performance We observe that the performance of Prismer improves with the addition of more modality experts, as shown in Fig. 5a. This is because more experts provide a greater diversity of domain knowledge to the model. However, we also note that the performance of the model eventually plateaus, which suggests that additional modality experts beyond a certain number do not provide any extra gains.

Better Experts, Better Performance To evaluate the impact of expert quality on Prismer's performance, we construct a *corrupted* depth expert by replacing a certain number of predicted depth labels with random noise sampled from a Uniform Distribution. As shown in Fig. 5b, Prismer's performance improves as the quality of the depth expert improves. This is intuitive as better experts provide more accurate domain knowledge, allowing the model to perceive more accurately.

Robustness to Noisy Experts Our results also show that Prismer maintains performance even when including experts that predict noise, as shown in Fig. 5c. Interestingly, adding noise can even result in a non-trivial improvement compared to training on RGB images alone, which can be considered as a form of implicit regularisation. This property allows the model to safely include many experts *without degrading the performance*, even when the expert is *not necessarily informative*. Therefore, Prismer presents a more effective learning strategy than the standard multi-task or auxiliary learning methods, which either require exploring task relationships [49, 27, 87] or designing more advanced optimisation procedures [48, 57].

5.2 Architecture Design and Training Details

Adaptor Design and Size In our ablation study of adaptor designs, as shown in row (i) and (ii) of Table 5, we find that the most straightforward adaptor design, consisting of a standard residual connection and an encoder-decoder structure, performs the best. We have experimented with more intricate designs, such as adding an additional adaptor at the end of each transformer layer or incorporating a learnable gating mechanism akin to the one shown

Ablated Component	Our Setting	Changed Setting	Params. (Rel.)	Step Time (Rel.)	VQAv2 (Acc.)
Prismer$_{\texttt{BASE}}$ (our setting with reduced training)			**1.00**	**1.00**	**72.79**
(i) Adapter Design	Residual MLP	Residual MLP ×2	1.04	1.02	72.36
		Gated Residual MLP	1.03	1.03	70.54
(ii) Adapter Bottleneck Dim.	1	1/2	0.95	0.96	72.52
		1/4	0.93	0.93	71.66
(iii) Resampler Design	Experts Perceiver	Random Sampling	0.91	0.96	72.24
		Full Perceiver	1.00	0.90	65.05
		Dual Perceiver	1.08	1.02	71.56
(iv) Resampler Layers	4	1	0.94	0.93	70.61
		2	0.96	0.96	72.39
		6	1.04	1.01	72.78
(v) Resampler Latents	64	32	1.00	0.95	72.44
		128	1.00	1.01	70.28
		256	1.00	1.06	68.07
(vi) Pre-training	Freeze Vision and Lang.	Freeze Vision Only	1.00	1.07	70.49
		Freeze Lang. Only	1.00	1.05	67.77
		All Parameters	1.00	1.15	68.13
(vii) Fine-tuning	Freeze Vision	Freeze Vision and Lang.	1.00	1.00	71.36
		Freeze Lang. Only	1.00	1.00	70.37
		All Parameters	1.00	1.00	68.69

Table 5: **Ablation studies for architecture components and training strategies.** We perform ablation studies to evaluate the impact of different architectural components and training strategies on the VQAv2 `test-dev` performance. We compare the performance of our default setting to other design and training options. The number of parameters and pre-training step time of the changed setting relative to the default setting are reported. To ensure a fair comparison, all experiments are evaluated using a reduced amount of training data and 3 modality experts: depth, normal and segmentation.

in [47], but both have resulted in inferior performance. Furthermore, we observe that a larger bottleneck hidden size for the single adaptor has led to improved performance.

Resampler Design and Multi-modal Sampling Strategy In our ablation study of Experts Resampler designs and various strategies for encoding multi-modal signals, as shown in row (iii) - (v) of Table 5, we find that using lightweight designs for the resampler layers and latent variables is crucial for stable training. Our experiments also show that using a non-learnable random sampling approach resulted in a slightly lower performance compared to using a learnable resampler. We have also attempted to optimise the resampler by receiving all input signals, including RGB information, but this approach has also resulted in a significant decline in performance. Finally, incorporating an extra resampler at the end of the vision encoder is not beneficial, though it may help in reducing and keeping a constant memory usage independent of the image resolutions, it ultimately leads to a decrease in performance.

The Effect of Frozen Backbones In our experiments on pre-training and fine-tuning whilst freezing different parts of the model, as shown in row (vi) and (vii) of Table 5, we find that freezing pre-trained parameters is essential for achieving strong performance and avoiding over-fitting and catastrophic forgetting of the learned knowledge.[3] Freezing these parameters also saves a significant amount of GPU memory. Even when fine-tuning on different downstream tasks, we find that freezing the vision encoder is beneficial (while allowing the resampler and adaptors to be trainable). This observation is consistent with the findings in [89], which demonstrates that fine-tuning only the language model with a frozen vision model can produce a much stronger zero-shot vision-language retrieval performance.

6 Conclusions, Limitations and Discussion

In this paper, we have introduced Prismer, a vision-language model designed for reasoning tasks. Prismer is parameter-efficient and utilises a small number of trainable components to connect an ensemble of diverse, pre-trained experts. By leveraging these experts, Prismer achieves competitive performance in image captioning, VQA, and image classification benchmarks, comparable to models trained on up to two orders of magnitude more data.

For full transparency, we now discuss some limitations of Prismer during our implementation and explore potential future directions for this work.

Multi-modal In-context Learning Zero-shot in-context generalisation is an emergent property that only exists in very large language models [8, 83]. In this work, we build Prismer on top of a small-scale language model with the main focus on parameter-efficient learning. Therefore, it does not have the ability to perform few-shot in-context prompting by design.

Zero-shot Adaptation on New Experts We experiment with inference on a pre-trained Prismer with a different segmentation expert pre-trained on a different dataset. Although we apply the same language model to encode semantic labels, Prismer shows limited adaptability to a different expert with a different set of semantic information, which leads to a notable performance drop.

Free-form Inference on Partial Experts Similarly, we discover that Prismer entangles its multi-modal features from all experts we include during pre-training. Therefore, only having a partial number of experts during inference will lead to a notable performance drop. We attempt to use a different training objective such as masked auto-encoding [5], to design Prismer to reason on an arbitrary number of experts, but it eventually leads to a degraded fine-tuned performance.

Representation of Expert Knowledge In our current design of Prismer, we convert all expert labels into an image-like 3-dimensional tensor via modality-specific post-processing for simplicity. There are other efficient methods to represent expert knowledge, such as converting object detection labels into a sequence of text tokens [11, 12]. This may potentially lead to a stronger reasoning performance and a more stable training landscape in future works.

[3] We assume the size of our pre-training multi-modal data is significantly smaller than the original pre-training data used to train the backbone experts.

References

[1] What does BERT with vision look at?", author = "li, liunian harold and yatskar, mark and yin, da and hsieh, cho-jui and chang, kai-wei. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2020.

[2] 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 International Conference on Computer Vision (ICCV)*, 2019.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[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 International Conference on Computer Vision (ICCV)*, 2015.

[5] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multimodal multi-task masked autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[6] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[7] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[8] 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*, 2020.

[9] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[12] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J. Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[13] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[14] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[15] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[16] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[18] 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.

[19] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on clip via vision-language knowledge distillation. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2022.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers), 2019.

[24] 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. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[25] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. Magma–multimodal augmentation of generative models through adapter-based finetuning. *arXiv preprint arXiv:2112.05253*, 2021.

[27] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[28] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[29] Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[30] Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. Language models are general-purpose interfaces. *arXiv preprint arXiv:2206.06336*, 2022.

[31] 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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[32] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[34] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[35] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2022.

[36] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[37] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[38] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2018.

[39] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 2017.

[41] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision (IJCV)*, 2020.

[42] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[43] 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 (NeurIPS)*, 2021.

[44] Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[45] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[46] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[47] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[48] Shikun Liu, Andrew J Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[49] Shikun Liu, Stephen James, Andrew J Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *Transactions on Machine Laerning Research (TMLR)*, 2022.

[50] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] Wei Liu, Chaofeng Chen, and Kwan-Yee Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2018.

[52] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[54] 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 (NeurIPS)*, 2019.

[55] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *The Artificial Intelligence Review*, 2014.

[56] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[57] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[58] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[59] Hien D Nguyen and Faicel Chamroukhi. Practical and theoretical aspects of mixture-of-experts modeling: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018.

[60] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip. *arXiv preprint arXiv:2211.00575*, 2022.

[61] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[62] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[63] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[64] 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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[65] 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.

[66] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.

[67] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[68] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

[70] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset

for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[71] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[72] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2018.

[73] David So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V Le. Searching for efficient transformers for language modeling. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[74] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[75] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[76] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[77] 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 (NeurIPS)*, 2021.

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[79] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[80] 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. *Transactions on Machine Laerning Research (TMLR)*, 2022.

[81] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[82] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[83] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Laerning Research (TMLR)*, 2022.

[84] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[85] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[86] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Laerning Research (TMLR)*, 2022.

[87] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[88] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

[89] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[90] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

[92] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

A Detailed Training Strategy and Hyper-parameters

All models are pre-trained with $[224 \times 224]$ image resolution, and we evaluate the models on three types of vision-language reasoning tasks: image captioning, visual question answering (VQA), and image classification. We fine-tune the models with a larger resolution $[480 \times 480]$ on image captioning and VQA tasks, and $[384 \times 384]$ on image classification tasks. Automated data augmentation [19] is applied for both pre-training and fine-tuning. A list of the hyper-parameters used in the experiments can be found in Table 6.

	Pre-training	COCO / NoCaps	VQAv2	ImageNet
Optimiser	AdamW			
LR Schedule	Cosine annealling to zero			
Weight Decay	0.05			
Warmup Steps	2000	0	0	0
Initial LR	$3/1 \cdot 10^{-4}$ (B / L)	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$
Resolution	224	480	480	384
Epochs	20	3	10	20
Batch Size	1024	256	512	64

Table 6: **The detailed list of hyper-parameters and training strategy.** To ensure reproducibility, we have included a list of all hyper-parameters used in our experiments. These same hyper-parameters are applied to both the `BASE` and `LARGE` model variants.

B Comparison of Training Cost

Prismer is highly efficient in terms of the training cost. The largest model variant, Prismer$_{\texttt{LARGE}}$, only requires 8 days of training on 32 NVIDIA V100 GPUs. This is significantly more efficient than previous state-of-the-art VLMs such as SimVLM [82] which requires 5 days of training on 2048 TPUv3, GIT-2 [80] which requires 1.5 months of training on 1500 NVIDIA A100s, and Flamingo [3] which requires 2 weeks of training on 1536 TPUv4. A detailed breakdown of the pre-training cost can be found in Table 7.

	Model Params.	Pre-training Data (# Image-Text Pairs)	Pre-training Cost (# PFlops Days)
BLIP$_{\texttt{LARGE}}$	583M	129M	$22.2^{\ddagger}$
SimVLM$_{\texttt{HUGE}}$	630M	1.8B	$30.2^{\ddagger}$
GIT	681M	0.8B	$45.8^{\ddagger}$
PaLI	17B	2.3B	450
Flamingo	80B	2.3B	$1.4\text{K}^{\dagger}$
GIT-2	5.1B	12.9B	$5.5\text{K}^{\dagger}$
Prismer$_{\texttt{BASE}}$	980M	12.7M	0.66
Prismer$_{\texttt{LARGE}}$	1.6B	12.7M	1.9

Table 7: **Training cost of vision-language models.** We compare the training cost of Prismer with several other vision-language models using the approximation method from [8]. The symbol † represents the training cost approximated by [13], and ‡ represents the training cost approximated by us.