PIN: Positional Insert Unlocks Object Localisation Abilities in VLMs

Michael Dorkenwald Nimrod Barazani Cees G. M. Snoek* Yuki M. Asano*
University of Amsterdam
https://quva-lab.github.io/PIN/

Abstract

Vision-Language Models (VLMs), such as Flamingo and GPT-4V, have shown immense potential by integrating large language models with vision systems. Nevertheless, these models face challenges in the fundamental computer vision task of object localisation, due to their training on multimodal data containing mostly captions without explicit spatial grounding. While it is possible to construct custom, supervised training pipelines with bounding box annotations that integrate with VLMs, these result in specialized and hard-to-scale models. In this paper, we aim to explore the limits of caption-based VLMs and instead propose to tackle the challenge in a simpler manner by i) keeping the weights of a caption-based VLM frozen and ii) not using any supervised detection data. To this end, we introduce an input-agnostic Positional Insert (PIN), a learnable spatial prompt, containing a minimal set of parameters that are slid inside the frozen VLM, unlocking object localisation capabilities. Our PIN module is trained with a simple next-token prediction task on synthetic data without requiring the introduction of new output heads. Our experiments demonstrate strong zero-shot localisation performances on a variety of images, including Pascal VOC, COCO, LVIS, and diverse images like paintings or cartoons.

1. Introduction

Vision-Language Models (VLMs) have shown remarkable results across diverse tasks, propelled by the advancements in Large Language Models (LLMs) [12, 15, 50]. Early works [21, 32, 42, 49, 63] used extensive image-caption data for end-to-end training, a trend later evolved by works like [4, 11, 26, 30, 31, 65], which efficiently integrated pretrained vision and language models through fusion networks to further enhance cross-modal understanding. Flamingo [4] demonstrates impressive multimodal in-context learning abilities. However, like many caption-based VLMs, it faces challenges in object localisation, a consequence of its training on web data.

*Equal last author.

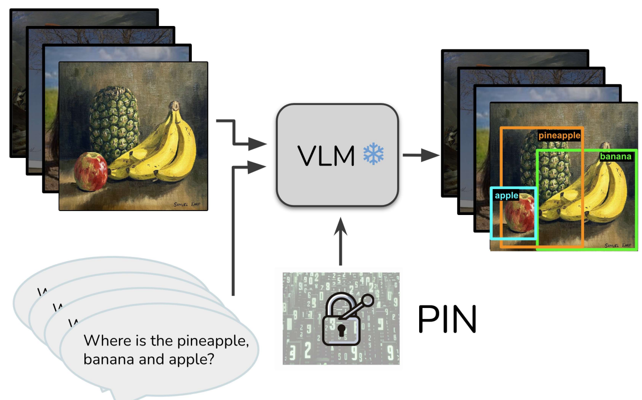

Figure 1. We learn a single Positional Insert (PIN) for unlocking zero-shot object localisation abilities in a frozen Vision Language Model (VLM) without adding any additional heads or requiring supervised datasets. Further output examples shown in Fig. 5 & 6.

Equipping VLMs with precise object localisation abilities is important for tasks like autonomous driving [1, 58, 59], assistive technology [61], and robotics [8, 14, 16]. Despite their proficiency in integrating visual-textual data, current image-caption training hinders accurate spatial understanding. Therefore, enhancing spatial comprehension in VLMs is key to enabling more nuanced and context-aware interactions.

One recent stream of research [9,38,53–55,60,62,66] focuses on developing unified *expert* Vision Language Models (VLMs) capable of performing a variety of tasks, including localisation, with a universal architecture. Although these models show impressive results across different tasks, their success largely depends on the availability of extensive task-specific, supervised data [9, 34, 54, 55]. Furthermore, [9, 38, 41, 53–55] require a large amount of compute for training. The setting we tackle in this paper is different. Our goal is to efficiently enable the localisation capabilities of VLMs while keeping their parameters untouched and *without* the need for localisation supervised datasets.

Our work aims to unlock the localisation abilities of caption-based VLMs by integrating spatial understanding into their existing zero-shot capabilities. We introduce a Positional Insert (PIN), a learnable spatial prompt designed

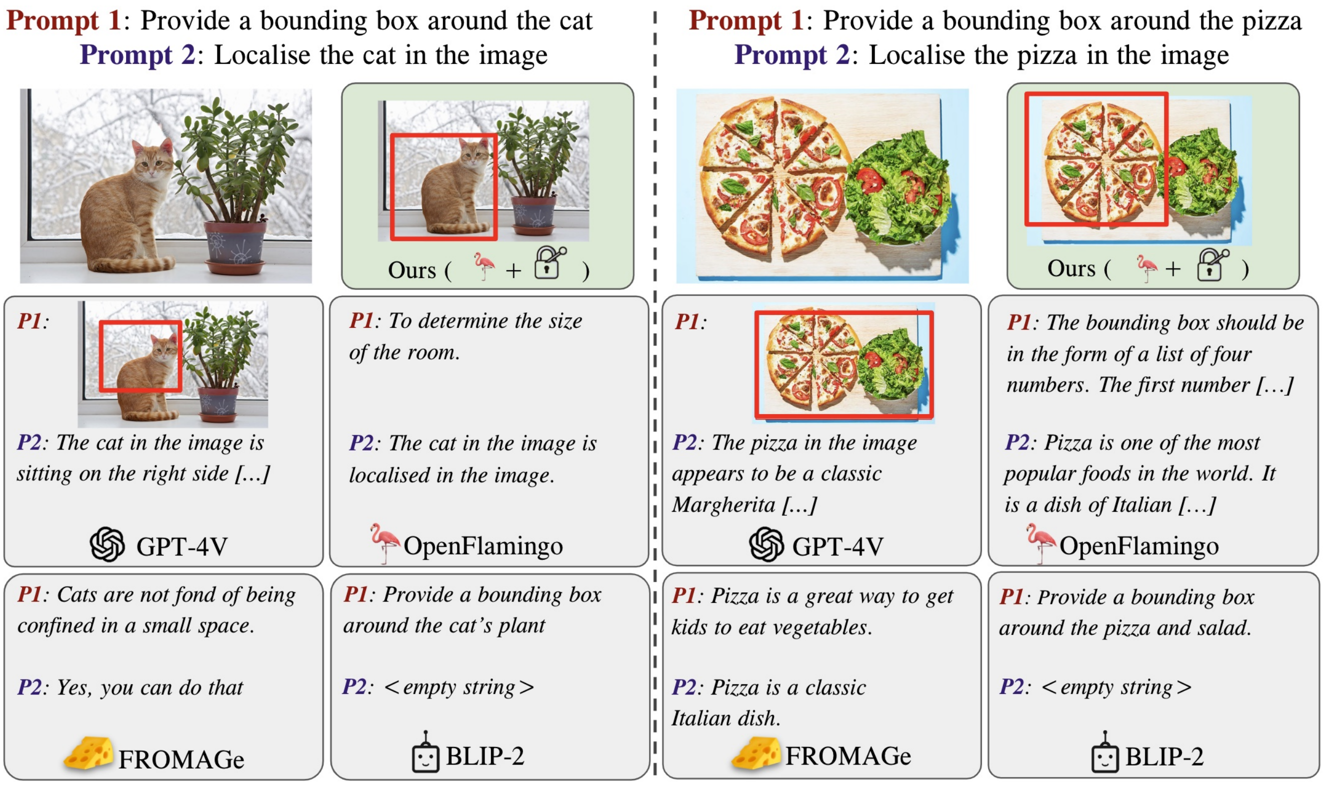

Figure 2. Examples from our analysis on localisation abilities of existing caption-based VLMs. GPT-4V [40] is the only model to return bounding boxes and by that roughly localised the object. All other VLMs struggle to easily localise the objects in the image. Further examples and different kinds of prompts are provided in the supplemental.

to infuse spatial awareness into VLMs without altering their pretrained weights. Our learned PIN is simply added to the vision encoder embedding and follows the VLMs forward pass from there, thereby not imposing any computational overhead. To train our PIN module effectively and without supervised data, we create a synthetic dataset composed of synthesized object renderings superimposed on background images, providing precise ground truth locations. We assess our approach on COCO [36], PVOC [13], LVIS [19], and RefCOCO [64]. Our findings reveal a significant enhancement in VLMs' object localisation abilities. Our contributions can be summarized as follows:

- We provide an analysis of the abilities of caption-based VLMs for object localisation.
- We propose PIN, a spatial prompt, to unlock the localisation abilities in caption-based VLMs.
- We demonstrate on the OpenFlamingo [5] and BLIP-2 [30] VLMs the ability to successfully localise objects on COCO, PVOC, LVIS, and other data.

2. Related Work

Caption-based Vision-Language Models. Large Language Models (LLMs) [12, 15, 40, 50] have not only been transformative for the field of natural language processing but have also significantly propelled the development of multimodal models. Initial works for Vision Language Models (VLMs) [2, 4, 11, 30, 31, 52, 56] concentrated on extensive image-text pretraining. These models typically undergo pretraining with vast collections of interleaved image-text data [46, 73]. Flamingo was a pioneer in merging a pretrained CLIP [42] image encoder with a pretrained LLM through a perceiver and gated cross-attention blocks, demonstrating strong multimodal in-context learning abilities. Given the image-text pretraining data containing descriptive captions for images, we categorize these VLMs as caption-based. This kind of pretraining naturally limits the spatial comprehension and expression abilities of those VLMs. In this paper, we present a new, simple, and efficient way designed to enable object localisation capabilities within these models.

Expert-based Vision-Language Models. Universal frameworks [10, 34, 38, 44, 60] have been introduced to unify architectures and training tasks by treating it as a language modeling problem conditioned on e.g. observed pixel inputs. Recent works [26, 29, 55, 67, 72] applied this to multimodal instruction-tuned data, promoting more intuitive human-model interactions for VLMs. The resulting *unified expert* VLMs are capable of handling diverse tasks. Many

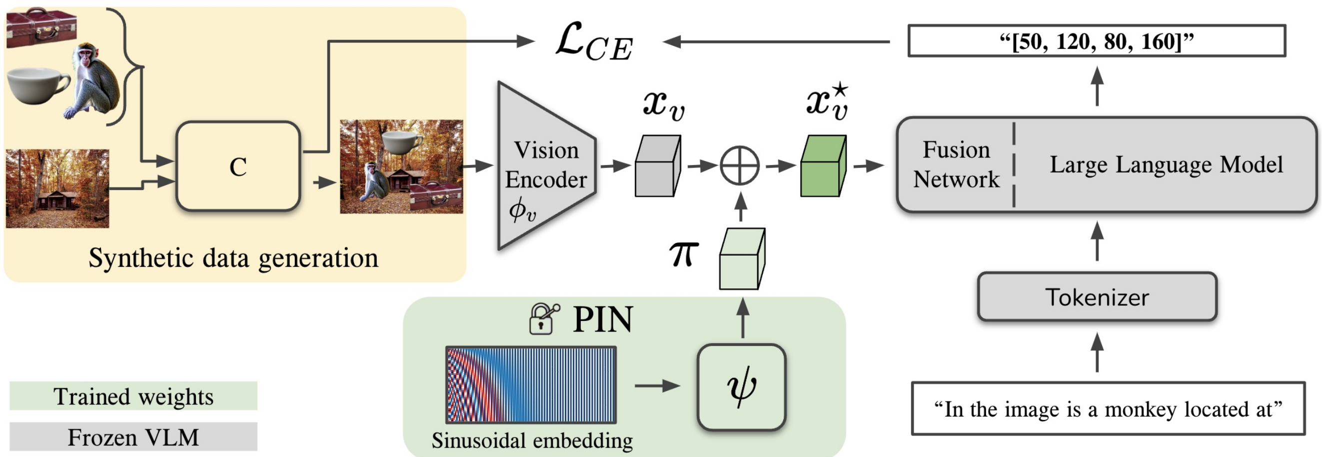

Figure 3. Schematic overview of our method. We generate synthetic training data by overlaying objects on background images using our composition function C. These images are then encoded, and our lightweight learnable spatial prompt vector π from the PIN module is added to their vision encodings x_v. Using the VLM's standard forward pass, a location text response is generated based on the input object name and the enhanced visual feature $x_v^\star$. The parameters of ψ in the PIN module are optimized with cross-entropy by comparing the generated text with the text describing the known object locations from the composition function C.

others [9, 38, 41, 53–55, 60, 62, 66] additionally target visual grounding tasks like localisation. Yet, those VLMs rely on large annotated localisation datasets [27, 36, 47, 64]. In addition, many of those works [38, 41, 54, 60, 66] require substantial amounts of compute to leverage this data. While these models exhibit impressive performance across various tasks, hence the name experts, their success hinges on large quantities of task-specific, supervised data and computational resources. Our work diverges from this path, seeking to unlock the object localisation capabilities of caption-based VLMs without relying on manually annotated datasets. We propose a more flexible and efficient strategy, exploring how far we can go without supervised data.

Visual Prompt Learning. Prompt Learning is a method originated from NLP [28, 35, 37] where prompts are viewed as continuous, task-specific vectors optimized during fine-tuning. This technique matches the performance of full fine-tuning but requires 1000 times fewer parameters, enhancing efficiency and reducing resource usage. Beginning works focused on adapting those methods to VLMs by adding learnable tokens to the language model [17, 69–71]. Subsequent works [7, 22, 23, 39, 57] extended them to the vision model and recently to both the vision and language branch [24]. However, these works have been applied to encoder-only models, such as CLIP [42], leaving their adaption to VLMs with a decoder unexplored. Motivated by these methods, we introduce a positional prompt for specifically targeting localisation in generative VLMs.

3. Localisation by Caption-based VLMs

Before discussing our proposed method, we first assess the object localisation capabilities of caption-based VLMs by analysing their textual responses given various prompts. We examine models such as GPT-4V [40], BLIP-2 [30], Flamingo [4, 5], and Fromage [26]. For that, we use prompts aimed at generating a bounding box response from these VLMs. Note that due to the undisclosed training data for GPT-4V [40], we cannot rule out its exposure to supervised object localisation training. We compare this against the publicly available 9B version of OpenFlamingo [5] and the 7B version of BLIP-2 [30]. An overview of the results and prompts can be found in Fig. 2. We find that among the evaluated VLMs, only GPT-4V [40] successfully returns bounding boxes that roughly localise the intended object. Other VLMs [5, 26, 30] are unable to provide any location information even in text form and instead are "chatty" (FROMAGe, OpenFlamingo) or return the input or provide no output (BLIP-2). In Sec. 5.1, we quantitatively evaluate the in-context learning abilities for localisation of the OpenFlamingo model. In the supplementary material, we broaden our study by examining a wider variety of prompts, specifically including those that do not require generating a bounding box, and by analyzing a larger number of samples. Yet, the conclusion remains the same as with the exemplary results in Fig. 2 that caption-based VLMs are unable to localise objects in a given image via textual responses.

4. Method

We tackle the shortcomings of caption-based Vision-Language Models (VLMs) in their ability to localise objects within images. To this end, we introduce a simple yet effective Positional Insert (PIN), designed to enhance the VLMs' object localisation capabilities without altering their existing parameters. An overview of our approach can be found in Fig. 3.

Preliminary. Vision-Language Models (VLMs) accept inputs composed of visual data such as images I alongside a textual input T. The visual component I is processed by a vision encoder ϕ_V producing a feature vector $x_v \in \mathbb{R}^{N_p \times D_v}$, where N_p denotes the number of patches and D_v the channel dimension. Similarly, the textual information T is tokenized, yielding textual embeddings $x_t \in \mathbb{R}^{M \times D_t}$, with M representing the amount of textual tokens and D_V the vocabulary size. The visual features x_v go through a fusion network F before being processed with the textual features x_t to produce a response text t_r=LLM$(F(x_v), x_t)$ by the Large Language Model.

4.1. PIN: Positional Insert

The Positional Insert is a learnable input-agnostic spatial feature vector and is inserted directly after the vision encoder ϕ_V. To instill spatial awareness into our PIN, we start with fixed positional embeddings of dimension d employing sinusoidal functions [51]

$$S[i, 2k] = \sin\left(\frac{\text{position}}{10000^{2k/d_{\text{model}}}}\right), \quad (1)$$

$$S[i, 2k+1] = \cos\left(\frac{\text{position}}{10000^{2k/d_{\text{model}}}}\right), \quad (2)$$

where i denotes the index of the position and k represents the index within the dimension of the embedding, with d_{model} as the dimensionality of the embedding space. The range for k extends from 1 to d_{model}. Each of the spatial sinusoidal vectors is further refined by a learnable, shallow feed-forward neural network ψ parametrized by θ, resulting in our PIN π=$\psi(S)$ with the output dimension matching the ones from the vision encoder $\pi \in \mathbb{R}^{M \times D_t}$. This learned embedding is then added to the output from the vision encoder x_v, resulting in the enriched visual feature representation

$$x_v^\star = x_v + \pi. \quad (3)$$

Training Objective. The PIN module's parameters θ of ψ are optimized via the text output produced by the large language model. This process requires no additional heads or projection layers, thus maintaining the model's simplicity and native natural language output. The model is trained with an input sequence consisting of a textual prompt $t_p \in T$ such as 'In the image is a <*obj*> located at' and is tasked to complete the sequence with the bounding box coordinates. For a given object name <*obj*>, present within the image, the model predicts a sequence of bounding box coordinates in the template of $t_r \in T$ like $[x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}]$ conditioned on the image features and the initial textual prompt. We employ a negative log-likelihood loss for the predicted tokens

$$\mathcal{L}_{CE}(\theta) = -\sum_{t=1}^{T} \log p_\theta \left(y_t | y_{<t}, x_v^\star\right), \quad (4)$$

where y_t corresponds to the target token at position t in the text, T is the total number of tokens to be predicted and $x_v^\star$ is the positional enhanced feature vector. Here p_θ is the probability assigned by the model to the correct token at position t, conditioned on the previous tokens $y_{<t}$, the visual features, and the textual prompt. This learning objective enables the easy adaption of pretrained VLMs for localisation without the dependency on specialized components like region proposal networks.

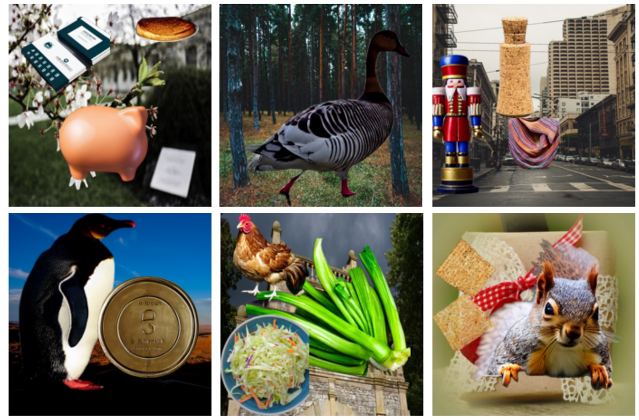

Figure 4. Sample images from our synthetic data generation.

4.2. Synthetic Data Generation

We do not rely on manually labeled data to unlock the positional information in the VLM. Instead, we generate our own synthetic data following [18,68] by utilizing Stable Diffusion [45] to synthesize objects from the LVIS [19] category list. The CLIP [43] module is used to sort out implausible images by removing those with a low CLIP [43] score, a matching score between the input image I and the textual information T. Note, since the vision encoder's weights remain unchanged, it is unlikely to overfit to any pasting artifacts. The composition function C overlays objects on randomly picked locations while considering the following constraints: the aspect ratio r of objects, minimal s_{min} and maximal s_{max} pasting sizes, the number of objects a_{max}, and the maximal overlap o_{max} w.r.t. already inserted objects. Given a background image $I_b \in I$, the composition function yields

$$(t_p, I_p) = C(I_b, r, a_{\text{max}}, s_{\text{min}}, s_{\text{max}}, o_{\text{max}}), \quad (5)$$

with a generated image $I_p \in I$ and the text $t_p \in T$ containing the object location for a randomly selected object by C. This process creates a self-generated supervision signal

	Method	PVOC$_{\leq 3 \text{ Objects}}$			COCO$_{\leq 3 \text{ Objects}}$			LVIS$_{\leq 3 \text{ Objects}}$		
		mIoU	mIoU$_M$	mIoU$_L$	mIoU	mIoU$_M$	mIoU$_L$	mIoU	mIoU$_M$	mIoU$_L$
OpenFlamingo [5]	*Baselines*									
	raw	0	0	0	0	0	0	0	0	0
	random	0.22±0.04	0.10±0.02	0.33±0.06	0.12±0.04	0.07±0.02	0.22±0.08	0.07±0.03	0.06±0.02	0.18±0.09
	2 context	0.19±0.11	0.08±0.05	0.30±0.18	0.10±0.08	0.06±0.04	0.18±0.16	0.04±0.06	0.03±0.04	0.10±0.15
	5 context	0.19±0.09	0.07±0.04	0.31±0.15	0.10±0.08	0.06±0.04	0.20±0.16	0.06±0.05	0.04±0.03	0.17±0.13
	10 context	0.20±0.11	0.06±0.03	0.32±0.18	0.09±0.07	0.05±0.04	0.17±0.14	0.05±0.05	0.03±0.03	0.15±0.14
	PEFT									
	CoOp on LLM	0.28	0.11	0.43	0.22	0.10	0.39	0.13	0.07	0.40
	VPT on F	0.34	0.16	0.51	0.26	0.15	0.47	0.19	0.14	0.48
	VPT on ϕ_V	0.42	0.21	0.61	0.33	0.22	0.57	0.23	0.19	0.56
	LoRA on ϕ_V	0.44	0.26	**0.62**	0.33	0.23	0.58	0.23	0.19	0.55
	PIN (ours)	**0.45**	**0.27**	**0.62**	**0.35**	**0.26**	**0.59**	**0.26**	**0.24**	**0.61**
BLIP-2 [30]	*PEFT*									
	VPT on F	0.33	0.12	0.51	0.27	0.12	0.50	0.18	0.11	0.47
	VPT on ϕ_V	0.32	0.12	0.50	0.26	0.11	0.48	0.17	0.10	0.46
	PIN (ours)	**0.44**	**0.24**	**0.63**	**0.34**	**0.22**	**0.60**	**0.26**	**0.23**	**0.60**

Table 1. Comparison on object localisation on a subset of PVOC [13], COCO [36] and LVIS [19] with up to 3 objects per image, yielding 3,582, 2,062 and 6,016 test images respectively. PIN improves on the OpenFlamingo in-context and PEFT baselines for both the OpenFlamingo and BLIP-2 VLM.

that is subsequently exploited in the training of PIN. Typical sample images can be found in Fig. 4.

5. Experiments

We apply our approach to the Flamingo [4] and BLIP-2 [30] VLM. More specifically we use the open-source version OpenFlamingo [5] for Flamingo. We evaluate the localisation abilities of our approach on a subset of COCO [36], PVOC [13], and LVIS [19] with up to 3 objects per image resulting in 3,582, 2,062 and 6,016 test images respectively. We use ground truth object names and localise those in a given image. We report numbers on the PVOC 2007, COCO, and LVIS evaluation set. The mean Intersection over Union (IoU) is reported quantifying the overlap between the true and predicted bounding box. We report this metric for all bounding boxes and additionally for medium and large bounding box sizes only. A bounding box is considered large if it is over 96×96 pixels, and medium if between 32×32 and 96×96 pixels. We keep OpenFlamingo and BLIP-2 in its native form, which uses image resolutions of 224, making it particularly difficult to localise small objects. For all experiments, we use the 3B parameter version with the instruction-tuned LLM of OpenFlamingo and the OPT 2.7B parameter version of BLIP-2.

Implementation details. The PIN module starts of from a 1D sinusoidal embedding [51] with 64 dimensions. From there a two-layer Multi-Layer-Perceptron is applied, each consisting of a fully connected (FC) layer, Layer Norm [6] and SwiGLU [48]. Lastly, a final FC layer is added to match the target vision encoder embedding dimension of 1024. The parameters of the PIN module are optimized with Adam [25] with a learning rate of 10^{-3}. We train our PIN module on $2 \times$ A6000 GPU for around two days. Overall, our PIN module consists of only around 1.2M parameters, *i.e.* around 0.04% of the VLM's size of 3B. Code will be released.

Synthetic dataset details. We follow X-Paste [68] to create our synthetic dataset using Stable Diffusion [45] version 1 generating 60 samples for each category in LVIS [19] resulting in around 70k object images. We exclude all categories overlapping with COCO [36] and PVOC [13] for training. For the background, we use images from the BG20-k [33] dataset on which we paste the objects. Following X-Paste's filtering procedure, we exclude all classes with less than $\leq$ 20 images remaining per class, as these classes might not be well-generated. For our composition function, we set the maximum allowed overlap to $o_{\text{max}}{=}0.5$, the number of images $a_{\text{max}}{=}3$, $r{=}r_{\text{orig}}$, $s_{\text{min}}{=}[0.3, 0.2, 0.1]$ and $s_{\text{max}}{=}[1.0, 1.0, 1.0]$, for up to three objects respectively.

5.1. Quantitative Results

Baselines. For comparison, we use OpenFlamingo's in-context learning version, configured with variable numbers of context images. To account for performance variation due to context image selection being sampled randomly, we execute each setup ten times and report the average and standard deviation. BLIP-2 is not able to do in-context learning due to the lack of interleaved image-text training data [30]. We select bounding boxes randomly from context images as a baseline to assess the in-context learning abil-

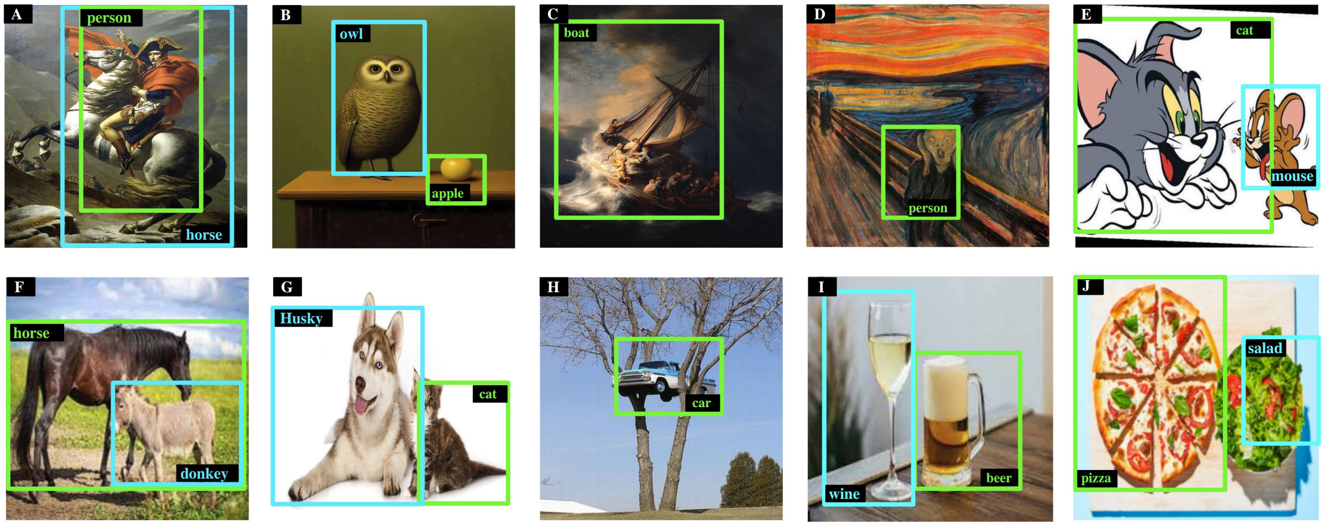

Figure 5. Localisation on a wide range of image types ranging from paintings, and comics to unique scenarios. Despite the varying image content, enhancing the OpenFlamingo caption-based VLM with our PIN shows strong localisation abilities.

ities. Additionally, we compare against other Parameter-Efficient Fine Tuning (PEFT) methods such as CoOp [70], using the strongest version with adding 16 learnable tokens to the input to the LLM. In addition, we append 100 learnable tokens in the spirit of Visual Prompt Learning (VPT) [22] to either the vision encoder ϕ or the Fusion network F (the same location where PIN is added). We also evaluate our method against finetuning the ViT vision encoder ϕ_V using LoRA [20] with α=16 and r=16.

Localisation on PVOC, COCO, and LVIS. From the results in Tab. 1, we first observe that our introduced PIN, when combined with OpenFlamingo, surpasses both the raw and the in-context learning versions of OpenFlamingo across all evaluated metrics, considerably. In particular, compared to the best OpenFlamingo in-context learning version, we improve in mIoU by a factor of 2$\times$ on PVOC and a factor of 3$\times$ on COCO. Notably, the PIN module achieves this without any exposure to COCO or PVOC classes during training, in contrast to the few-shot nature of in-context learning. The raw zero-shot OpenFlamingo variant fails to generate any meaningful bounding boxes, as visualized in Fig. 2. We observe that the random bounding box selector consistently performs better than the OpenFlamingo in-context learning version. This demonstrates that OpenFlamingo cannot leverage the positional information given by in-context bounding boxes to generate plausible bounding boxes for the query samples.

Furthermore, we also compare the adapted VLM with PIN for OpenFlamingo against other Parameter-Efficient Fine-Tuning (PEFT) methods. First, we observe low performance of CoOp. This is primarily because of the lack of spatial positional information in CoOp's adaptation. OpenFlamingo employs a perceiver resampler as a fusion network, which removes most positional information during caption-based pretraining. Thus, the CoOp adaption struggles to solve the localisation task. In contrast, our PIN outperforms this baseline considerably, as it can add positional information directly to the vision embedding during adaption. We also compare against a different PEFT baseline which follows Visual Prompt Tuning (VPT) [22], adding 100 learnable tokens to either the vision encoder ϕ_V or fusion network F. PIN outperforms the VPT baseline applied to the fusion network considerably and also the one applied to the vision encoder ϕ_V, especially for medium-sized bounding boxes (IoU$_M$). These findings demonstrate that PIN better incorporates positional information into the pretrained VLM. In addition, we also show PIN's necessity by comparing it against finetuning the vision encoder ϕ_V with LoRA [20]. PIN slightly outperforms the strong LoRA baseline while having 5$\times$ fewer parameters. We observe that the LoRA-adapted VLM can nearly perfectly solve our synthetic training examples, overfitting potentially to synthetic data artifacts. In contrast, PIN utilizes the strong concepts learned in the ViT without changing its weights, thus excluding the possibility of overfitting to synthetic data artifacts. We can also confirm the effectiveness of PIN on BLIP-2, outperforming again the other PEFT baselines. These findings demonstrate that PIN can effectively unlock localisation abilities in various VLMs beyond OpenFlamingo.

Grounding on RefCOCO. We also evaluate PIN on RefCOCO [64] Test-A split in a zero-shot manner, paving a new way for reporting model performance *without using any of its annotated training data*. To this end, we extend our synthetic dataset with positional expressions like 'left apple', 'monkey on the right' *etc*. With this simplistic

Figure 6. Object localisation results on PVOC [13] and COCO [36]. The PIN module unlocks spatial localisation in the caption-based OpenFlamingo [5] VLM.

OpenFlamingo [5]	P@0.3
+ Raw	0
+ In-context learning	3.7
+ PIN w/o positional referral	14.1
+ PIN w/ positional referral	26.4

Table 2. Evaluation on RefCOCO [64] Test-A. PIN shows decent grounding abilities without using any annotated training data, outperforming the in-context learning Flamingo baseline. Extending our synthetic dataset with positional referrals improves performance considerably.

setup, we achieve 26.4 P@0.3, indicating decent grounding abilities, compared to only 3.7 for the in-context learning Flamingo baseline. Extending our synthetic data with referral expression improves results considerably, by a factor of nearly 2. In the supplemental, we visualized our grounding predictions for RefCOCO. A limiting factor is the rather small 1B parameter LLM in OpenFlamingo, having trouble understanding more complex and longer referrals.

5.2. Qualitative Results

Localisation on diverse images. We also explore the object localisation abilities of our adapted VLM on a wide range of images, encompassing various domains such as comics and paintings, as illustrated in Fig. 5. Notably, our method demonstrates robust performance in localising distinct characters and objects, even amidst significant domain variations. For instance, it successfully identifies the cat and mouse in a comic image (Fig. 5E) and accurately locates the

	Method	PVOC$_{\leq 3 \text{ Objects}}$ mIoU	mIoU$_M$	mIoU$_L$	COCO$_{\leq 3 \text{ Objects}}$ mIoU	mIoU$_M$	mIoU$_L$
OpenFlamingo	*Generalization*						
	PIN (COCO)	**0.45**	**0.27**	**0.63**	**0.39**	**0.31**	**0.62**
	PIN (Synth.)	**0.45**	**0.27**	0.62	0.35	0.26	0.59
	Higher Resolution						
	PIN (224)	0.45	0.27	0.62	0.35	0.26	**0.59**
	PIN (448)	**0.47**	**0.30**	**0.65**	**0.37**	**0.29**	**0.59**
BLIP2	PIN (224)	0.44	0.24	0.63	0.34	0.22	0.60
	PIN (364)	**0.47**	**0.27**	**0.66**	**0.37**	**0.26**	**0.62**

Table 3. Ablating the image resolution and the choice of synthetic training data for PIN.

person in a painting (Fig. 5D), as well as the owl and apple in another (Fig. 5B). Additionally, our VLM showcases its ability to differentiate between closely related objects. This is evident in its distinguishing between a donkey and a horse (Fig. 5F), as well as between a glass of wine and a glass of beer (Fig. 5I). These observations lead us to conclude that our adapted VLM not only excels in localising objects across varied image types but also retains the strong zero-shot capabilities typical of caption-based VLMs.

Localisation on PVOC and COCO. The adapted VLM accurately localises objects of different sizes, as demonstrated in Fig. 6. *Variety in object sizes:* It identifies both large (person in Fig. 6Q) and small objects (bird in Fig. 6I; person in Fig. 6O). *Variety in object locations:* We also find that the enhanced VLM localises objects at various locations in an image, *e.g.* boxes near the bottom (Fig. 6C,E), top (Fig. 6B,M), left (Fig. 6F,R) and right (Fig. 6E,N). *Crowded and overlapping:* Additionally, our model effectively manages more complex situations such as more crowded scenes (train in Fig. 6C), partial occlusions (person riding a horse in Fig. 6D). *Multi-object:* Our method is capable of localising multiple objects within a single image, demonstrating its ability to recognize more than just the most salient object. This can be seen *e.g.* in Fig. 6Q for the person and toilet and in Fig. 6B for the person and the motorbike. Yet, the adapted model struggles with more confusing scenes yielding more loose bounding box predictions like the trail of the aeroplane in Fig. 6J. Similarly, for small bounding boxes, our approach cannot locate objects very precisely, *e.g.* the sofa and chair in Fig. 6H or sink in Fig. 6Q. Overall, we conclude that the model can extend its zero-shot abilities to the object localisation task. In the supplemental, we visualize results with the BLIP-2 VLM.

5.3. Ablations

Generalization of synthetic data. In Tab. 3 (*Generalization*), we delve deeper into the choice of training data on the zero-shot abilities of our PIN module. For that, we compare training PIN on either the COCO datasets or using the synthetic data in which all COCO and PVOC categories are excluded. As expected, we observe better performance for the PIN trained on COCO and evaluated on COCO. However, we observe equivalent performance when analyzing their generalization abilities to PVOC. From that, we conclude that synthetic data serves as a viable solution to adapt pretrained VLMs for object localisation while preserving their generalization capabilities.

Higher image resolution. In Tab. 3 (*Higher Resolution*), we analyze the impact of using higher image resolutions on the performance of PIN. All OpenFlamingo models are pretrained on a resolution of 224×224. To circumvent that, we extrapolate the frozen positional embeddings of the ViT, allowing our PIN to be trained at a resolution of 448×448. As expected, this leads to an improvement across all IoU metrics, particularly for medium-sized bounding boxes (mIoU$_M$). We scaled the size of the bounding box for medium M, and large L according to the increase in scale of the image resolution. Most VLMs of BLIP-2 are trained on an image resolution of 224×224, yet, caption finetuned VLMs are available on 364×364 image resolution. We visually compare the difference in Fig. **??** and observe tighter bounding boxes with the higher resolution VLM. Training PIN on a higher BLIP-2 resolution results in similar IoU improvements as for OpenFlamingo.

Impact of PIN on VLM's general abilities. We analyze the impact of applying PIN on the general abilities of the VLM using the VQAv2 [3] dataset. The base performance of OpenFlamingo is 44.1% when inserting PIN, the performance reduces to 34.3%, yet it does not compromise the VLM. Moreover, we compare this to the VLM adapted with the finetuned vision encoder. We observe a bigger reduction in performance with 33.4%. In addition, our PIN can be easily deactivated, thereby retaining the general VLM abilities, a flexibility not possible when finetuning the ViT.

6. Conclusion

In this work, we introduced PIN, a lightweight module that enables object localisation capabilities in a frozen VLM. We first showed the limited object localisation abilities of caption-based VLMs. Subsequently, we verified that these capabilities were enabled with our PIN module on OpenFlamingo and BLIP-2. Our zero-shot results across PVOC and COCO, various image types, and objects demonstrate that the strong performance of caption-based VLMs can be transferred to localisation.

Acknowledgement

This work is financially supported by Qualcomm Technologies Inc., the University of Amsterdam, and the allowance Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy.

References

[1] Wayve lingo-1. `https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/`. 1

[2] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A causal masked multimodal model of the internet. *CoRR*, 2022. 2

[3] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. VQA: visual question answering. *Int. J. Comput. Vis.*, 2017. 8

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 1, 2, 3, 5

[5] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *CoRR*, 2023. 2, 3, 5, 7

[6] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, 2016. 5

[7] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *CoRR*, 2022. 3

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong T. Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: vision-language-action models transfer web knowledge to robotic control. *CoRR*, 2023. 1

[9] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *CoRR*, 2023. 1, 3

[10] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey E. Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022. 2

[11] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. 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, and Weicheng Kuo. Pali: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 1, 2

[12] 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. *Journal of Machine Learning Research*, 2023. 1, 2

[13] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson W. H. Lau. Location-aware single image reflection removal. In *ICCV*, 2021. 2, 5, 7

[14] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan 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 Pete Florence. Palm-e: An embodied multimodal language model. In *ICML*, 2023. 1

[15] Tom B. Brown et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 2

[16] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *CoRR*, 2023. 1

[17] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *CoRR*, 2022. 3

[18] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 4

[19] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 4, 5

[20] 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 *ICLR*, 2022. 6

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV (33)*, 2022. 3, 6

[23] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV (35)*, 2022. 3

[24] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, 2023. 3

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2016. 5

[26] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *ICML*, 2023. 1, 2, 3

[27] 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. *Int. J. Comput. Vis.*, (1), 2017. 3

[28] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP (1)*, 2021. 3

[29] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *CoRR*, 2023. 2

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1, 2, 3, 5

[31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1, 2

[32] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 1

[33] Jizhizi Li, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. Bridging composite and real: Towards end-to-end deep image matting. *Int. J. Comput. Vis.*, (2), 2022. 5

[34] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 1, 2

[35] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP (1)*, 2021. 3

[36] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, 2014. 2, 3, 5, 7

[37] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, 2021. 3

[38] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. UNIFIED-IO: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2023. 1, 2, 3

[39] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022. 3

[40] OpenAI. GPT-4 technical report. *CoRR*, 2023. 2, 3

[41] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *CoRR*, 2023. 1, 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1, 2, 3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[44] Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Trans. Mach. Learn. Res.*, 2022. 2

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4, 5

[46] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 2

[47] 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 *ICCV*, 2019. 3

[48] Noam Shazeer. GLU variants improve transformer. *CoRR*, 2020. 5

[49] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP/IJCNLP (1)*, 2019. 1

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, 2023. 1, 2

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 4, 5

[52] 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. *Trans. Mach. Learn. Res.*, 2022. 2

[53] 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 *ICML*, 2022. 1, 3

[54] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *CoRR*, 2023. 1, 3

[55] Weihan Wang, Qingsong Lv, 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, and Jie Tang. Cogvlm: Visual expert for pretrained language models. *CoRR*, 2023. 1, 2, 3

[56] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022. 2

[57] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022. 3

[58] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, et al. On the road with gpt-4v (ision): Early explorations of visual-language model on autonomous driving. *CoRR*, 2023. 1

[59] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, Zheng Zhu, Shaoyan Sun, Yeqi Bai, Xinyu Cai, Min Dou, Shuanglu Hu, and Botian Shi. On the road with gpt-4v(ision): Early explorations of visual-language model on autonomous driving. *CoRR*, 2023. 1

[60] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *ECCV (36)*, 2022. 1, 2, 3

[61] Zongming Yang, Liang Yang, Liren Kong, Ailin Wei, Jesse Leaman, Johnell Brooks, and Bing Li. Seeway: Vision-language assistive navigation for the visually impaired. In *SMC*, 2022. 1

[62] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mplug-owl: Modularization empowers large language models with multimodality. *CoRR*, 2023. 1, 3

[63] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022. 1

[64] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV (2)*, 2016. 2, 3, 6, 7

[65] 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 *CVPR*, 2022. 1

[66] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022. 1, 3

[67] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey. *CoRR*, 2023. 2

[68] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, Weiming Zhang, and Nenghai Yu. X-paste: Revisiting scalable copy-paste for instance segmentation using CLIP and stablediffusion. In *ICML*, 2023. 4, 5

[69] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 3

[70] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 3, 6

[71] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. *CoRR*, 2022. 3

[72] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *CoRR*, 2023. 2

[73] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal C4: an open, billion-scale corpus of images interleaved with text. *CoRR*, 2023. 2