PIN: Positional Insert Unlocks Object Localisation Abilities in VLMs
Supplementary Material

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

1. Extended analysis of caption-based VLMs

This section broadens the scope of our analysis of the localisation abilities of caption-based Vision-Language Models (VLMs) from the main paper. Our goal is to assess a wider range of prompts on more sample images. The study employs the same collection of VLMs as before, namely:

- GPT-4V [10]
- 7B version of BLIP-2 [8]
- 9B version of Flamingo [1, 2]
- Fromage [7]

Note that due to the undisclosed training data for GPT-4V [10], we cannot rule out its exposure to supervised object localisation training. Our expanded analysis includes three prompt types, designed to test the VLMs' abilities in various aspects of spatial understanding and object localisation. The prompts cover a spectrum of challenges, from generating bounding boxes around a specified object (shown in Fig. 2) to performing grid-based localisation (illustrated in Fig. 3) and determining relative positions (depicted in Fig. 4).

Generate bounding box Similar to the study in the main paper, we evaluate caption-based VLMs in their ability to generate a bounding box for the specified object. For this purpose, we applied the prompt from the main paper to more sample images, which are depicted at the top of Figure 2. Our observations indicate that only GPT-4V is capable of generating a bounding box that is approximately located near the object of interest, yet not with high precision; for example, the cat in Figure 2D. In contrast, all other VLMs, such as OpenFlamingo as shown in Figure 2B, complete the sentence with 'in the image,' without providing any bounding box informatiom. To further evaluate these VLMs, we added more detailed instructions to the prompt, such as 'in the format of $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$', which can be found in Figure 2E-H. We observe that even with more instructions, these VLMs are not able to provide any bounding box or positional information about the inquired object.

Grid-based localisation In this part, we evaluate the VLMs with a grid-based localisation task using two different grid styles. The first style uses a standard numbered grid (Fig. 3A-D), while the second uses a chessboard-style grid (Fig. 3E-H). In both cases, an 8x8 grid is overlaid on the images. The size of each grid cell varies to match the aspect ratio of the image. The goal is to evaluate the VLMs' ability to pinpoint the object location using the designated grid. We observe that only GPT4-V is able to list grid cells in its response, yet, for the numbered grid its response does not match the objects (e.g. the dog in Fig. 3B), and for it only roughly matches the object (e.g. the cat in Fig. 3H). The other models generally fail to provide accurate or relevant coordinates in response to the grid-based prompts. Their responses are often off-task, with Flamingo providing unrelated continuations (such as 'cells [...] of the brain'), Fromage repeating the prompt, and BLIP-2 sometimes not responding at all. This indicates a gap in these models' ability to understand and execute spatial tasks.

Relative position Here, we evaluate the VLMs' relative position abilities. For that, we task the models to identify an object relative to a center object (Fig. 4A-D). Therefore, we designed an artificial image with a pizza at the center, surrounded by a lemon to the left, a shark to the bottom, a cow to the right, and a dog above. We observe that BLIP-2 listed three random objects, regardless of the prompt. Fromage detects the objects to the left correctly Fig. 4A, yet, all other directions are wrong. OpenFlamingo responses are only about the pizza ignoring the surrounding objects. GPT-4V does answers correctly for all directions except for the one above the pizza Fig. 4B. We extend our study to ask VLMs how a specific object is placed relative to a red circle that is overlaid on the image (Fig. 4E-H). This is inspired by [11] which showed that red circles can be used for VLMs to direct their attention to a specific region. We observe that Fromage and BLIP-2 are not able to provide any meaningful responses. Instead, often these VLMs try to describe the

---

*Equal last author.

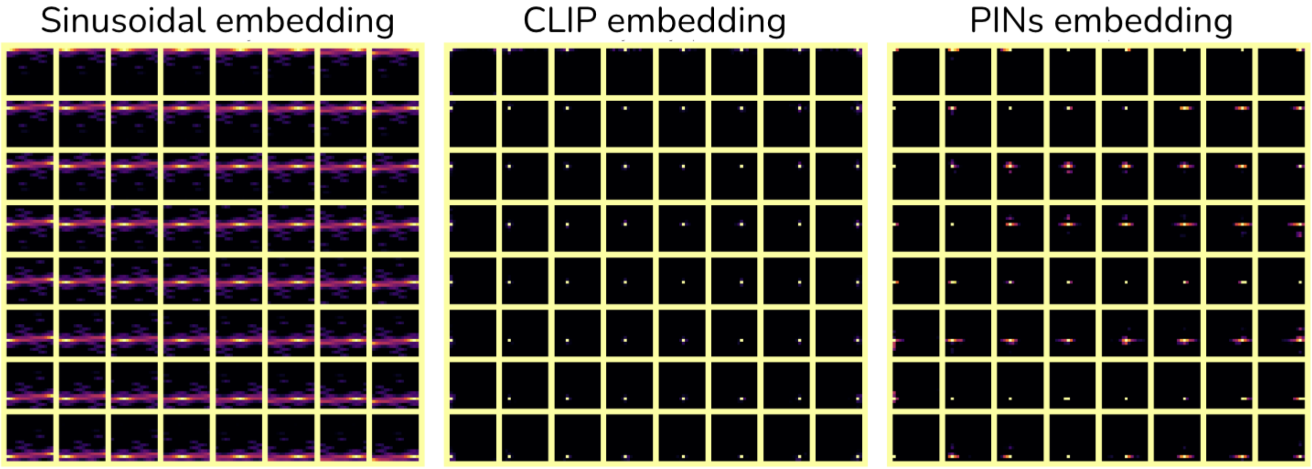


Figure 1. Visualization of pair-wise similarities of the raw sinusoidal embedding, the CLIP encoder's spatial embeddings and our learned PIN. Our embedding captures local positional information, making it effective for localisation.

absolute position of the object e.g.for Fromage Fig. 4C and BLIP-2 Fig. 4B. OpenFlamingo answers give indeed relative positional information, yet, most often wrong and in 3 of 4 cases 'on the left side'. Again, only GPT4-V is able to give roughly correct responses e.g. Fig. 4D, yet, Fig. 4A and C are partially and Fig. 4B is completely wrong. From that, we conclude that caption-based VLMs struggle with solving relative positional tasks indicating a lack of spatial understanding on the relative placement of objects.

Summary The extended analysis of caption-based VLMs reveals limitations in their spatial understanding and object localisation abilities. Among all evaluated models, only GPT-4V managed to generate responses that partially met the task criteria. Yet, due to the undisclosed training data for GPT-4V [10], we cannot rule out its exposure to supervised object localisation training. Despite varying prompt complexities and image scenarios, all other VLMs consistently underperform in tasks requiring precise localisation and relative positioning. The study's findings underscore a gap in the current capabilities of caption-based VLMs, highlighting their struggles with accurately interpreting and responding to spatially-oriented tasks. This motivated us to design the PIN module to unlock localisation abilities in the caption-based VLM Flamingo.

2. Additional ablations

Amount of objects to paste. Lastly, we evaluate the maximum number of objects, denoted as a_{max}, that are pasted onto the background for each image. Separate models are trained for 1, 2, 3, 4, and 5 allowed objects per image. The results are shown in Tab. 1 and the mIoU on the COCO dataset is reported for a maximum of 3, 4, and 5 objects per image. We observe a decrease in performance when too few objects are pasted during training (mIoU$_{\leq 3}$ of 0.24 vs. 0.35) as the VLM only focuses on the most salient object. Alternatively, pasting too many objects also decreases performance, especially for mIoU$_{\leq 5}$. With a_{max}=3 we strike a good balance between these two extremes, yielding the best accuracies across all mIoU values.

# pasted objects	mIoU$_{\leq 3}$	mIoU$_{\leq 4}$	mIoU$_{\leq 5}$
≤ 2	0.24	0.21	0.19
≤ 3	**0.35**	**0.31**	**0.29**
≤ 4	0.35	0.30	0.28
≤ 5	0.34	0.30	0.27

Table 1. Ablation on the number of objects being pasted during training on our synthetic data evaluated on COCO. Pasting with 1-3 objects works best across all mIoU scores.

Visualizing π In Fig. 1, we present a visualization of our learned input-independent feature vector π from the PIN module. Following ViT [5], we compute the cosine similarity for all pairings of the 16×16 patches. This results in a 16×16 grid visualization, where each cell shows the similarity between a specific patch with all other patches. For readability, we omitted every second patch, thus being a 8×8 plot. We also visualize the 1D sinusoidal embedding as it is the starting point for our PIN module. From Fig. 1, we find that this embedding only obtains the highest similarities with itself and among patches in the same row, a characteristic feature of the sinusoidal embeddings. Conversely, our learned embedding π demonstrates high similarity primarily within itself *and* its immediate neighboring patches, an attribute advantageous for localisation tasks, highlighting the similarity among the spatial locations. We also visualize the similarity for the raw CLIP vision encoder embeddings by averaging the similarities over 50 images. We observe that the embedding of the vision encoder does not contain any positional information as only one bright spot,

	# layers in ψ	S embedding	mIoU	mIoU$_M$	mIoU$_L$
(a)	1	sinusoidal	0.34	0.25	0.57
(b)	2	sinusoidal	**0.35**	**0.26**	**0.59**
(c)	3	sinusoidal	0.33	0.24	0.56
(d)	2	sinusoidal	0.35	0.26	0.59
(e)	2	learned	0.35	0.27	0.59

Table 2. Ablation on the number of layers in ψ and the type of positional embedding S used in PIN evaluated on COCO. The best performance is obtained with only 2 layers in ψ and sinusoidal vs learned positional embeddings for S leads to the same results.

the similarity with itself, can be found in each cell. In summary, our visualizations show that our learned embedding π successfully captures local positional information, making it particularly effective for tasks like localisation.

2.1. Depth of ψ

In this ablation, we analyse the impact of varying the number of layers in the feed-forward neural network ψ inside the PIN module. Table 2 (a) - (c) displays the results. Increasing the layer quantity results in a rise in parameters, advancing from 0.6M for one layer to 1.2M for two layers, and reaching 2.3M for three layers. We find that the optimal number of layers in ψ is 2, as evidenced by the highest mIoU scores across all categories. The findings indicate that a few learnable parameters are sufficient, aligning with the input-agnostic characteristics of the PIN module.

2.2. Sinusoidal vs learned

We investigate the effectiveness of the sinusoidal embedding [3] and compare it against a learned variant. As shown in Table 2 (d) - (e), both types of embeddings yield similar performance, with no significant difference in mIoU scores. Our goal is to incorporate spatial information into the VLM, for which the sinusoidal embedding is ideally suited. Its performance matches that of the learned version, which in theory provides greater adaptability and capacity for the model. Thus, the sinusoidal embedding with no learnable parameters is the optimal choice for our PIN module due to its efficiency and effectiveness in this context.

2.3. Choice of background

We ablate the choice of background images for our synthetic data generation. To this end, we compare the BG-20k [9] by using plain white background images on COCO in Tab. 3 rows (a-b). We observe a strong performance decrease in terms of IoU with white backgrounds, especially for medium-sized bounding boxes. We conjecture that the more realistic images in BG-20k contribute to a more robust spatial embedding π, enhancing localisation performance.

	Background	o_{max}	mIoU	mIoU$_M$	mIoU$_L$
(a)	White	0.5	0.24	0.12	0.48
(b)	BG-20k [9]	0.5	0.35	0.26	0.59
(c)	BG-20k [9]	0.0	0.33	0.26	0.56
(d)	BG-20k [9]	0.5	0.35	0.26	0.59

Table 3. Ablation on choice of background image and overlap between objects (o_{max}) on COCO. Realistic background images and allowing for overlap between the pasted objects improves localisation performance.

2.4. Overlap between objects

Lastly, we evaluate the effect of allowing for overlap o_{max} between pasted objects during training on our synthetic generated data on COCO. We compare two settings of no-overlap o_{max}=0.5 in Tab. 3, rows (c-d). We find that by creating more realistic generations by allowing for overlapping pasted objects, we obtain slightly better localisation performance, indicating a better learned PIN module.

3. Additional qualitative results

3.1. Visualization on RefCOCO

In Fig 5, we show zero-shot visual grounding results on RefCOCO of PIN with the OpenFlamingo VLM. The adapted VLM struggles with more complex scenarios(B and C), yet, it effectively handles simpler cases (F, G, H, J).

3.2. Visualization of PIN with BLIP-2

In Fig. 6, we visualize results when applying to the BLIP-2 VLM on 224×224, BLIP-2 (224), image resolution and 364×364, BLIP-2 (364), for PVOC [4]. The PIN trained with the higher image resolution BLIP-2 version is able to predict more accurate bounding boxes.

3.3. Visualizations on LVIS

Our adapted VLM demonstrates effective object localisation also on LVIS [6] as demonstrated in Fig. 7. Our model can localise multiple objects within a single image, as illustrated in Fig. 7A, D, E, and I. It also effectively identifies objects in unusual settings, such as a teddy bear in a tree (Fig. 7J) and a remote under a cat (Fig. 7H). These examples support the conclusion that our model extends its zero-shot capabilities to the task of object localisation.

3.4. Zero-shot visualizations on synthetic data

In Fig. 8, we demonstrate the zero-shot localisation capabilities of our VLM on our synthetic generated data. This visualization showcases the model's ability to accurately identify and localise multiple objects within an image, even in scenarios where pixel boundaries are not distinctly defined.

3.5. Visualizations of failure cases

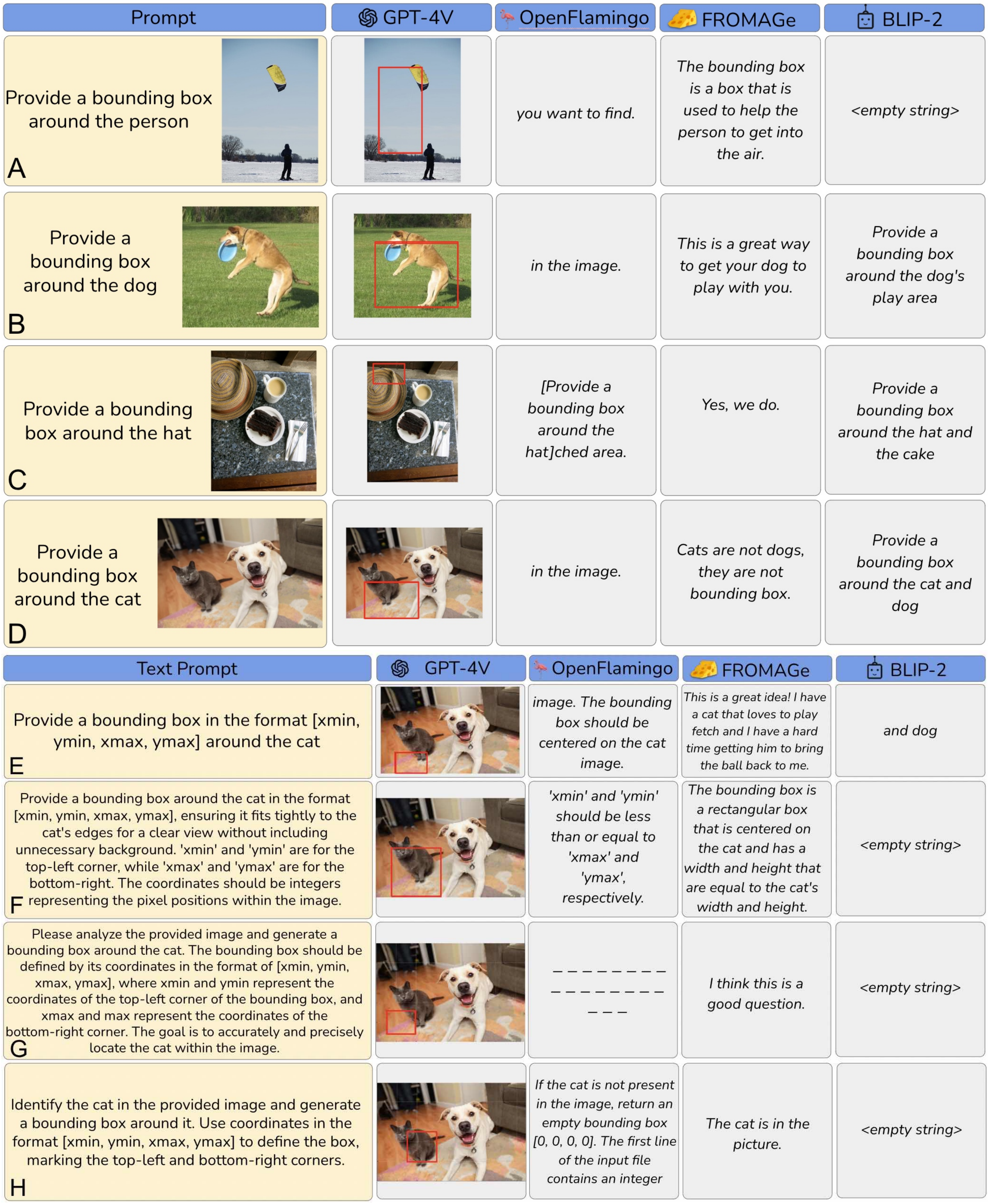
We visualize typical failure cases of our model in Fig. 9. As discussed in the limitation section, our model cannot effectively localise multiple instances from the same object due to our simplistic training procedure. We found that the model typically handles those cases by drawing a bounding box around all instances from the same class which can be seen in Fig. 9A-E. As we keep the original input resolution of the OpenFlamingo [2] VLM of 224, our model struggles to localise these objects with a tight bounding box (Fig. 9F-I) since the object spans only across a few pixels.

4. Limitations.

Owing to our simplistic training procedure and the caption-based pretraining focusing on big objects in relatively low-resolution images, our model struggles with generating tight bounding boxes, especially around smaller objects. Moreover, the PIN-enhanced VLM is trained to output a bounding box regardless if the object inquired is present in the image or not, limiting its use case as a general object detector. As a *no-bells-and-whistles* paper, we leave these challenges to future work.

5. Additional implementation details

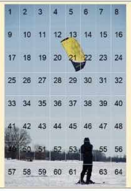
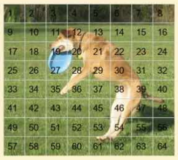
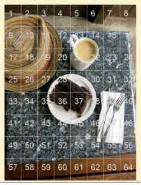
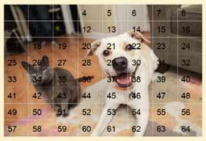
Our synthetic training and validation datasets are created from 1,116 object categories, based on LVIS, with overall 56,064 images generated by Stable Diffusion. These categories exclude those of COCO and PVOC to enable measuring truly zero-shot localisation performance. A different set of 81 categories (which includes the COCO and PVOC classes), amounting to 4,296 images, is reserved for zero-shot evaluation. The dataset averages 50.43 ± 12.11 images per object category. For pasting objects onto the background images, we find dividing the images into grids of 16×16 worked best for OpenFlamingo, 14×14 for BLIP-2, aligning with the shapes of the vision embedding. Thus, the network only needs to predict numbers between 0 and 224 in steps of the grid size, simplifying the task at hand. This also leads to bounding boxes not being perfectly precise around the inquired object, though, it has better performance than the model trained on a grid size matching image size. For RefCOCO, we extend our synthetic dataset with positional referral expressions. For that, we increase the likelihood of sampling the same object type to 0.7. We still randomly select one of the pasted objects for training, yet, when sampling an object for which its object type occurs multiple times in the image, we add a positional referral to it. These are computed by measuring the axes with the highest difference between the center points of the objects. Then, we extend the prompt with e.g. 'left person', or 'person on left' for left, right, top, and bottom.



Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
A Provide a bounding box around the person		you want to find.	The bounding box is a box that is used to help the person to get into the air.	<empty string>
B Provide a bounding box around the dog		in the image.	This is a great way to get your dog to play with you.	Provide a bounding box around the dog's play area
C Provide a bounding box around the hat		[Provide a bounding box around the hat]ched area.	Yes, we do.	Provide a bounding box around the hat and the cake
D Provide a bounding box around the cat		in the image.	Cats are not dogs, they are not bounding box.	Provide a bounding box around the cat and dog

Text Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
E Provide a bounding box in the format [xmin, ymin, xmax, ymax] around the cat		image. The bounding box should be centered on the cat image.	This is a great idea! I have a cat that loves to play fetch and I have a hard time getting him to bring the ball back to me.	and dog
F Provide a bounding box around the cat in the format [xmin, ymin, xmax, ymax], ensuring it fits tightly to the cat's edges for a clear view without including unnecessary background. 'xmin' and 'ymin' are for the top-left corner, while 'xmax' and 'ymax' are for the bottom-right. The coordinates should be integers representing the pixel positions within the image.		'xmin' and 'ymin' should be less than or equal to 'xmax' and 'ymax', respectively.	The bounding box is a rectangular box that is centered on the cat and has a width and height that are equal to the cat's width and height.	<empty string>
G Please analyze the provided image and generate a bounding box around the cat. The bounding box should be defined by its coordinates in the format of [xmin, ymin, xmax, ymax], where xmin and ymin represent the coordinates of the top-left corner of the bounding box, and xmax and max represent the coordinates of the bottom-right corner. The goal is to accurately and precisely locate the cat within the image.		_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	I think this is a good question.	<empty string>
H Identify the cat in the provided image and generate a bounding box around it. Use coordinates in the format [xmin, ymin, xmax, ymax] to define the box, marking the top-left and bottom-right corners.		If the cat is not present in the image, return an empty bounding box [0, 0, 0, 0]. The first line of the input file contains an integer	The cat is in the picture.	<empty string>

Figure 2. Analysis of localisation abilities of caption-based VLMs to provide a bounding box. A-D shows results with the same prompt on different sample images and E-H illustrates prompts with more instruction information on the same cat and dog image (D).

Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
The person is located at grid cells  A	*The person [...] is located at grid cells 50 to 52 horizontally, and from 41 to 44 vertically.*	*in the brain that are associated with the processing of auditory information.*	*The person is located at grid cells.*	*<empty string>*
The dog is located at grid cells  B	*The dog in the image [...] grid cells 25 through 28, 33 through 36, 41 through 44, and 49 through 52.*	*in the brain that are associated with the processing of spatial information.*	*<empty string>*	*with numbers*
The hat is located at grid cells  C	*The hat is located at grid cells 1 to 7 horizontally and 1 to 6 vertically.*	*in the left and right hemispheres of the brain.*	*The hat is located at grid cells.*	*<empty string>*
The cat is located at grid cells  D	*The cat in the image is situated at grid cells 26, 27, 33, and 34.*	*in the hippocampus, a region of the brain associated with spatial memory.*	*The cat is located at grid cells.*	*and the dog is in the center*

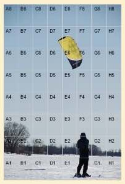
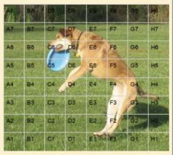
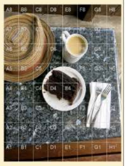
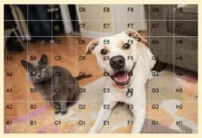
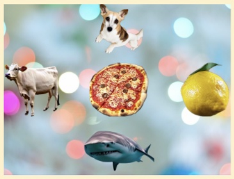
Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
Given an image with a chessboard grid overlay, the grid coordinates where the person is located are  E	*The person in the image is located at the coordinates that intersect at the row labeled "A" and the column labeled "3". So, the grid coordinates where the person is standing would be A3.*	*returned.*	*The person is in the upper left corner of the image.*	*shown*
Given an image with a chessboard grid overlay, the grid coordinates where the dog is located are  F	*To determine the grid coordinates where the dog is located, I'll [...] identify the squares that the dog covers. [...] Therefore, the grid coordinates where the dog is located would be from C3 to F6.*	*returned.*	*The dog is in the upper left corner of the grid.*	*the same as the grid coordinates where the dog is standing*
Given an image with a chessboard grid overlay, the grid coordinates where the hat is located are  G	*The hat in the image spans several grid coordinates. The main bulk[...]: A7, B7, C7 A6, B6, C6 A5, B5, C5* *Some parts of the hat's brim extend into the adjacent squares [...]*	*returned. The grid coordinates are returned as a list of integers*	*The hat is located at coordinates (0,0).*	*:*
Given an image with a chessboard grid overlay, the grid coordinates where the cat is located are  H	*The cat in the image is primarily located at the coordinates C3, C4, D3, and D4. Its body extends from B3 to E4, covering a larger area, but the main focus would be on the squares C3, C4, D3, and D4.*	*returned.*	*The cat is in the upper left corner of the grid.*	*shown*

Figure 3. Analysis of grid-based localisation of caption-based VLMs. A-D shows results with a numbered grid overlaid on the image. while E-H shows findings with a checkerboard-style grid.

Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
To the left of the pizza is a  A	*cow.*	*picture of a pizza.*	*A cow.*	*dog, a shark, and a fish*
Above the pizza is a B	*lemon.*	*picture of a pizza. The pizza is a picture of a pizza.*	*A pizza with a slice of cheese and a slice of tomato.*	*dog, a shark, and a pizza*
To the right of the pizza is a C	*lemon.*	*picture of a pizza.*	*A cow, a tree, and a pizza.*	*dog, a shark, and a fish*
Below the pizza is a D	*shark.*	*layer of mozzarella cheese, followed by a layer of tomato sauce, then a layer of pepperoni, then a layer of sausage, then [...]*	*A pizza with a slice of cheese and a slice of tomato.*	*dog, a shark, and a fish*

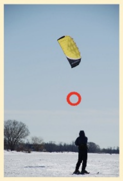
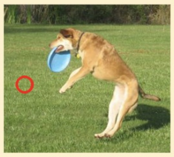
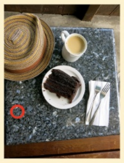
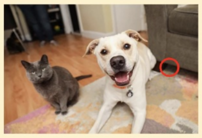
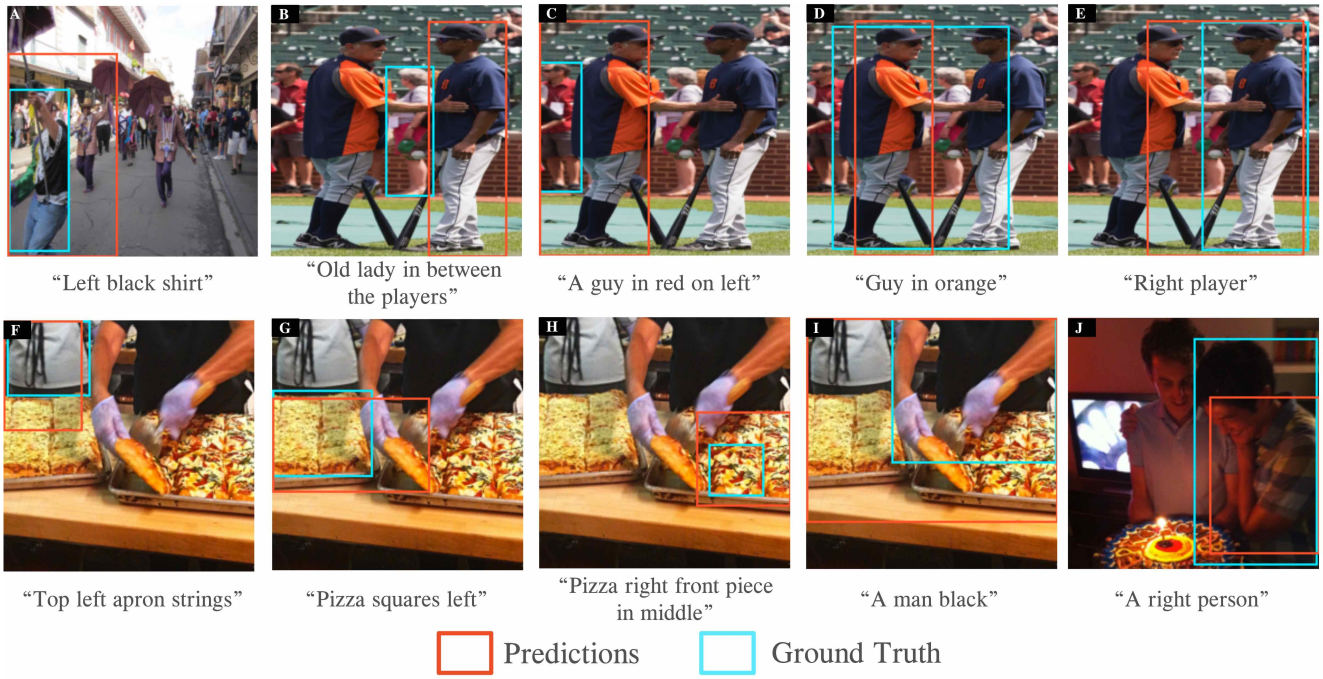
Prompt	GPT-4V	OpenFlamingo	FROMAGe	BLIP-2
Relative to the red circle, the person is located  E	*The person is located below and to the left of the red circle in the image*	*at the center of the circle.*	*The person is located in the red circle.*	*in the middle of the snow*
Relative to the red circle, the dog is located  F	*The dog is located to the left of and slightly above the red circle. It's in the middle of an action,[...].*	*on the left side of the red circle.*	*The dog is located in the red circle.*	*in the middle of the field*
Relative to the red circle, the hat is located  G	*The hat is located to the northwest relative to the red circle.*	*on the left side of the head.*	*The hat is located on the table in the corner of the room.*	*on the right side of the plate*
Relative to the red circle, the cat is located  H	*The cat is located to the left of the red circle.*	*on the left side of the red circle.*	*The cat is located in the red circle.*	*in the middle of the dog*

Figure 4. Analysis of relative position abilities of caption-based VLMs. In A-D, VLMs have to identify the object relative to the center one. In E-H, VLMs are tasked to provide the location relative to a red circle.



"Left black shirt" "Old lady in between the players" "A guy in red on left" "Guy in orange" "Right player"

"Top left apron strings" "Pizza squares left" "Pizza right front piece in middle" "A man black" "A right person"

Figure 5. Zero-shot visual grounding results on RefCOCO [12] of PIN with the OpenFlamingo [2] VLM. The adapted VLM struggles with more complex scenarios(B and C), yet, it effectively handles simpler cases (F, G, H, J).

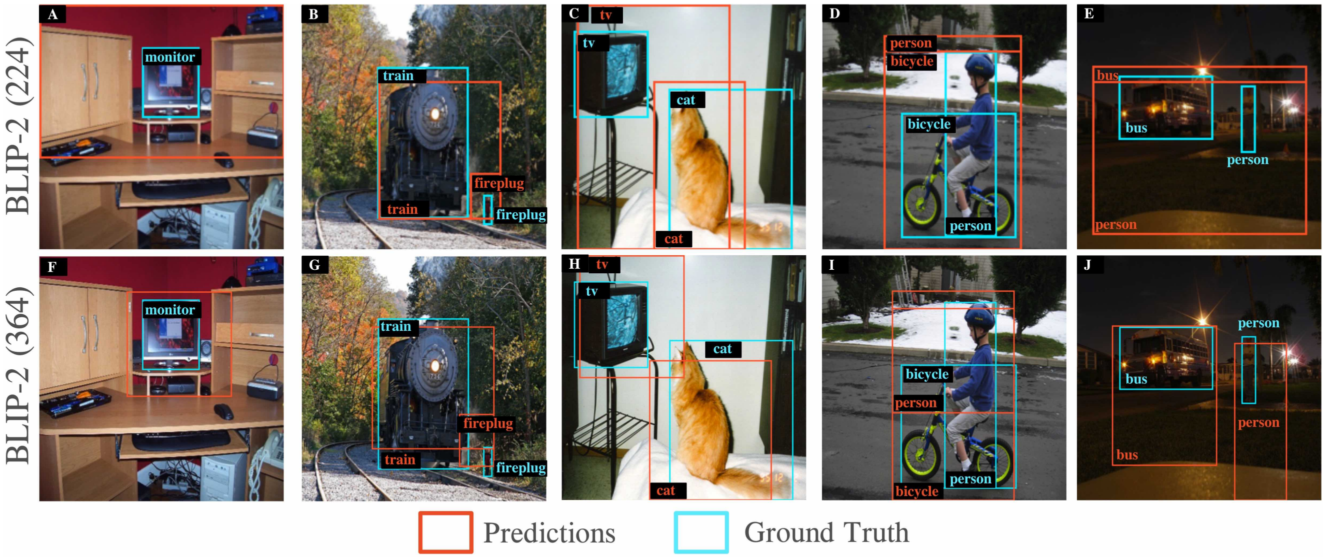


Figure 6. Object localisation results with BLIP-2 [8] on 224×224, BLIP-2 (224), image resolution and 364×364, BLIP-2 (364), on PVOC [4]. The PIN trained with the higher image resolution BLIP-2 version is able to predict more accurate bounding boxes.

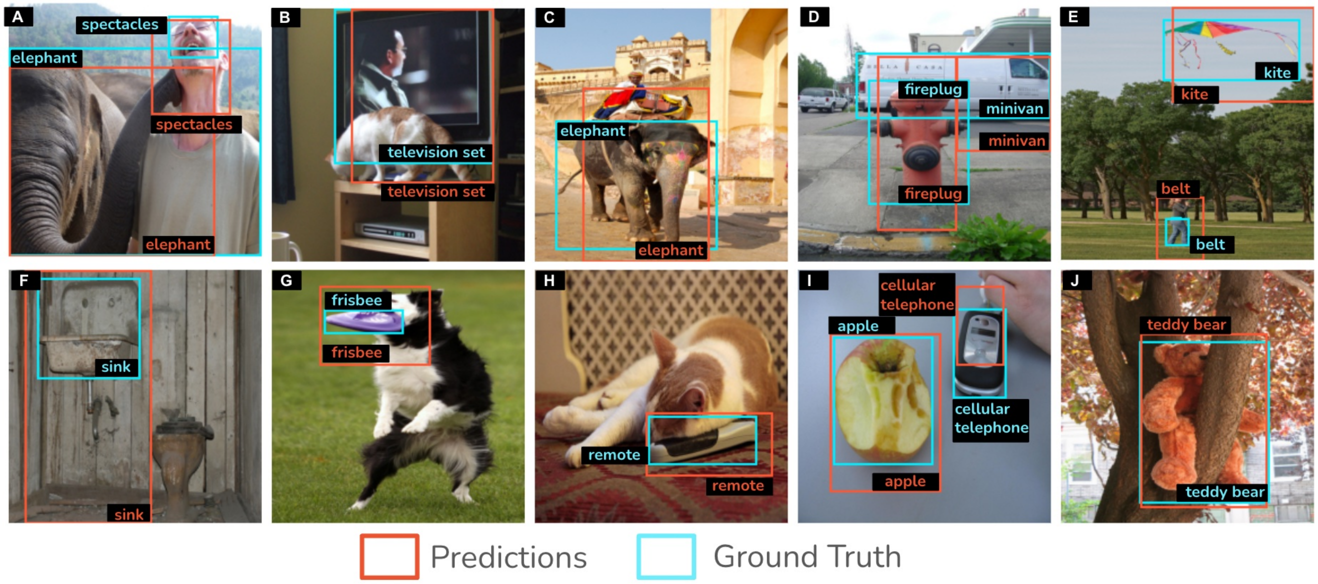


Figure 7. Object localisation results on LVIS [6] with the OpenFlamingo VLM.

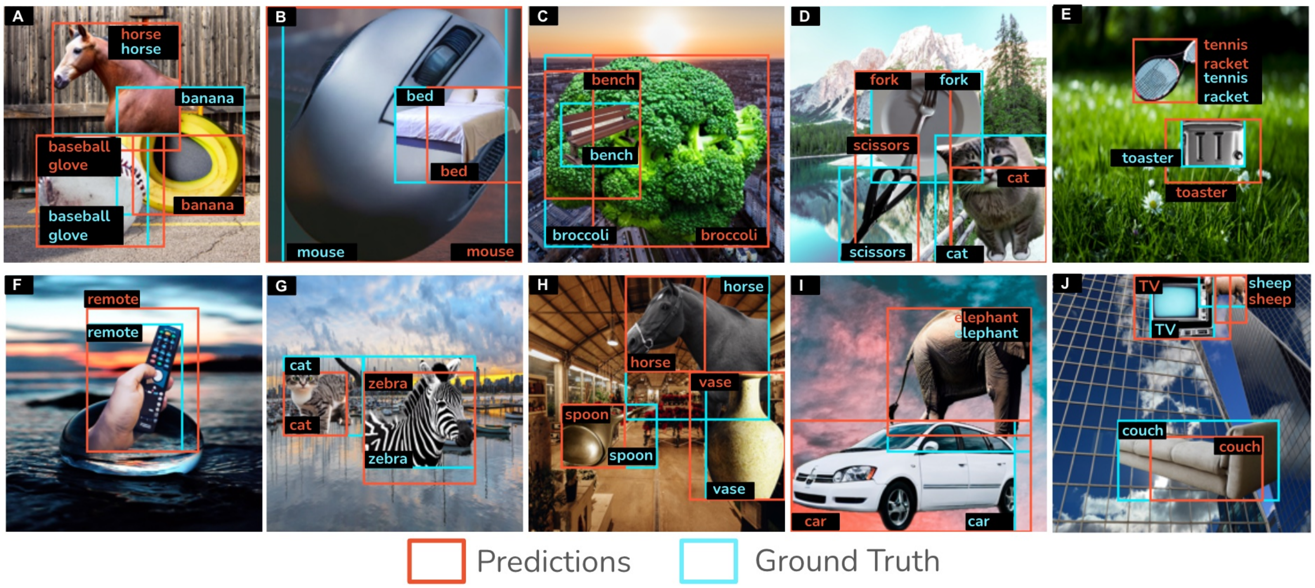


Figure 8. Zero-shot object localisation results on our synthetic data with the OpenFlamingo VLM.

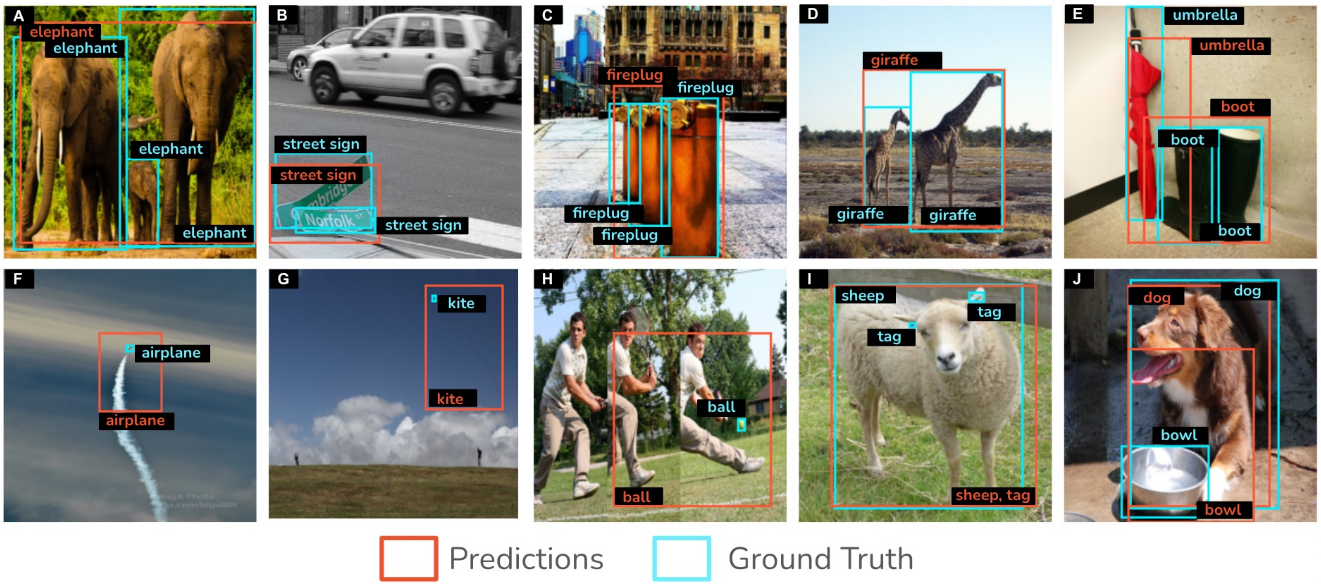


Figure 9. Typical failure cases: Due to the minimalistic design of our method, the PIN enhanced VLM cannot localise multiple instances of the same class (A-E). Often the VLM draws a bounding box around all objects of the same instance. Additionally, keeping the original input resolution of 224 from the VLM limits our ability to effectively manage very small objects (D-I).

References

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

[2] 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. 1, 4, 8

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 3

[4] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson W. H. Lau. Location-aware single image reflection removal. In *ICCV*, 2021. 3, 8

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[6] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 3, 9

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

[8] 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, 8

[9] 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. 3

[10] OpenAI. GPT-4 technical report. *CoRR*, 2023. 1, 2

[11] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does CLIP know about a red circle? visual prompt engineering for vlms. *CoRR*, 2023. 1

[12] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV (2)*, 2016. 8