CAPTURE: Evaluating Spatial Reasoning in Vision Language Models via Occluded Object Counting

Atin Pothiraj Elias Stengel-Eskin Jaemin Cho Mohit Bansal
UNC Chapel Hill
{atin, esteng, jmincho, mbansal}@cs.unc.edu

Abstract

Recognizing and reasoning about occluded (partially or fully hidden) objects is vital to understanding visual scenes, as occlusions frequently occur in real-world environments and act as obstacles for spatial comprehension. To test models' ability to reason about multiple occluded objects, we introduce a novel task, ***C****ounting* ***A****modally for* ***P****atterns* ***T****hrough* ***U****nseen* ***RE****gions (*CAPTURE*), which requires a model to count objects arranged in a pattern by inferring how the pattern continues behind an occluder (an object which blocks parts of the scene).* CAPTURE *requires both recognizing visual patterns and reasoning, making it a useful testbed for evaluating vision-language models (VLMs) on whether they understand occluded patterns and possess spatial understanding skills. By requiring models to reason about occluded objects,* CAPTURE *also tests VLMs' ability to form world models that would allow them to fill in missing information.* CAPTURE *consists of two parts: (1)* CAPTURE^{real}*, with manually filtered images of real objects in patterns and (2)* CAPTURE^{synthetic}*, a controlled diagnostic with generated patterned images. We evaluate four strong VLMs (GPT-4o, Intern-VL2, Molmo, and Qwen2-VL) on* CAPTURE*, finding that models struggle to count on both occluded and unoccluded patterns. Crucially, we find that models perform worse with occlusion, suggesting that VLMs are also deficient in inferring unseen spatial relationships: even the strongest VLMs like GPT-4o fail to count with occlusion. In contrast, we find that humans achieve very little error on* CAPTURE*. We also find that providing auxiliary information of occluded object locations increases performance, underscoring that the model error comes both from an inability to handle occlusion as well as difficulty in counting in images.*[1]

1. Introduction

Inferring what lies behind different objects in occluded scenes is crucial for human perception, as it allows us to maintain a coherent understanding of our environment even when parts are hidden. The human visual system accomplishes this by integrating past experiences, context, and sensory inputs to reconstruct incomplete scenes [19, 27, 30, 45]. Meanwhile, recent advancements in vision-language models (VLMs) – especially in terms of visual and spatial reasoning – raise the question of whether these systems can perform similar inferential tasks. One way of measuring such capabilities is through amodal completion – the task of inferring the invisible parts of partially occluded objects; here, vision-only models are typically evaluated via dense prediction tasks like object segmentation and image inpainting [5]. However, this format is not well-suited for assessing VLMs, whose outputs consist of text tokens rather than pixel-level predictions. This raises a critical question: How can we quantify the ability of VLMs to form spatial world modeling [17] in the presence of occlusion?

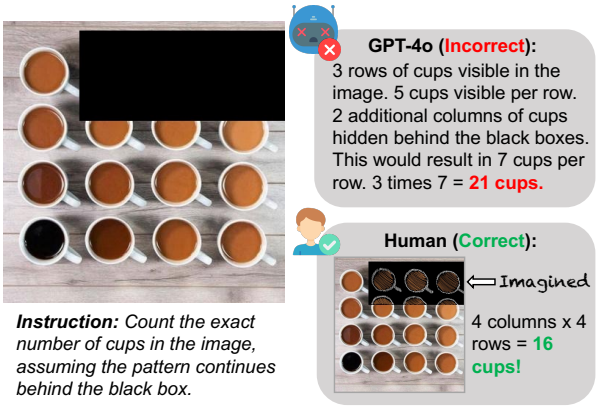

Figure 1. CAPTURE example with an output from GPT-4o. While people can easily infer the missing number of cups and correctly reason over occluded patterns, models generally struggle to reason over these occluded scenes.

To address this, we introduce **CAPTURE**, **C**ounting **A**modally for **P**atterns **T**hrough **U**nseen **RE**gions, a novel benchmark that tests a VLM's world modeling and spatial reasoning abilities through the task of *amodal counting*, where models are prompted to count occluded objects

[1] Code and data: https://github.com/atinpothiraj/CAPTURe

by amodally completing a pattern. CAPTURE focuses on counting as it provides an objective and easy-to-verify output by comparing predicted counts with ground truth values. Moreover, patterned objects appear in various real-world domains, especially in man-made environments like parking lots, cities, and warehouses, where counting objects is often required. Fig. 1 illustrates the CAPTURE task. We show a VLM an image where objects are placed in a regular pattern (e.g., a 4x4 grid) with some objects occluded, and ask the model to count the total number of objects in the image *assuming that the pattern continues behind the occlusion*. The task requires *handling occlusion*, *pattern recognition*, and *counting* skills that exist in humans from a fairly young age [27, 30, 45], thus humans can easily answer such questions – indeed, we find that people can complete CAPTURE tasks with almost no error.

CAPTURE consists of two subsets: $\text{CAPTURE}^{\text{real}}$ and $\text{CAPTURE}^{\text{synthetic}}$. As shown in Fig. 2, $\text{CAPTURE}^{\text{real}}$ contains real-world images and tests the ability of models to perform amodal counting in naturalistic contexts, while $\text{CAPTURE}^{\text{synthetic}}$ allows us to analyze specific factors by controlling different variables like color, shape, and number of objects. All images in CAPTURE contain a pattern of objects and a manually annotated occluding black box covering some objects. $\text{CAPTURE}^{\text{real}}$ contains 924 images with a diverse range of settings and objects, covering 92 different object types, while $\text{CAPTURE}^{\text{synthetic}}$ contains 1250 images across multiple attribute classes.

By combining vision encoders with large language models (LLMs), VLMs have the potential to reason in a zero-shot way about visual inputs. To put this ability to the test and measure VLMs' ability to reason about missing visual information, we evaluate four strong recent VLMs (GPT-4o, InternVL2, Molmo, and Qwen2VL) on CAPTURE. Our experiment results (Sec. 4) show that models generally struggle with the multiple aspects of the task, with high error rates on both $\text{CAPTURE}^{\text{real}}$ and $\text{CAPTURE}^{\text{synthetic}}$ for occluded and unoccluded images. In contrast, we find that humans can perform the task easily: whereas model performance deteriorates as more objects in images are occluded, humans complete the task almost perfectly. We also compare VLMs to a vision-only model trained to count visible objects; while this model generally outperforms VLMs, its error is directly tied to the number of occluded objects – the more objects are occluded, the higher its error will be.

By objectively measuring VLMs' spatial reasoning capabilities under occlusion, CAPTURE highlights an unexpected weakness in VLMs. We analyze this weakness by providing the model with additional clues and information. Specifically, we test to what degree the VLMs' failure stems from an inability to integrate visual information by providing it with a text-based representation of the visible objects in the image in the form of object coordinates; here, VLMs perform substantially better, indicating that their poor performance on CAPTURE stems partly from an inability to count objects in images, rather than an inability to count more generally. Our findings align with previous work, which similarly finds that VLMs struggle to count in images [22, 33, 42]. We also test the degree to which VLM errors stem from an inability to form a world model by providing it with auxiliary information (the coordinates of the occluded objects in text, or inpainting the occluded regions). We find that VLMs perform substantially better with this auxiliary information, suggesting that VLMs are partly limited by their inability to imagine the missing visual information. Addressing these gaps is critical for VLMs to function effectively in real-world scenarios, where visual reasoning often involves occlusions – whether counting stadium seats, components on production lines, or buildings in neighborhoods. We hope that our work will foster future research on improving the world modeling capabilities of VLMs.

2. CAPTURE

2.1. Task Overview

Input/output formulation. CAPTURE tests VLMs on occlusion reasoning, pattern recognition, and counting of both visible and occluded objects. VLMs already achieve high accuracy in classifying single, occluded objects [20]. Thus, we also argue that VLMs have the potential to perform well on CAPTURE's challenging task because their proficiency in handling occlusion ought to enable them to recognize occluded objects and reason accordingly. All images in CAPTURE contain a pattern. This makes the task solvable for models and people – if the objects were not placed in a pattern, it would be unreasonable to expect models to infer the position of the occluded objects. For example, given an image of a random pile of coins with a region occluded, it is not easy to infer whether the occluded region contains no coins or contains roughly the same amount as the rest of the pile. For this task, the patterns considered are all regular and fairly small, e.g. grids, circles, triangles, and other regular shapes – see Fig. 2 for further examples. The last step of CAPTURE is counting, asking the model to provide an objectively measurable output. In addition to VLMs, we also test COUNTGD [3], a state-of-the-art object detection-based counting method, finding that it fails to account for the occluded scenario, as its training entails solely predicting the visible, unoccluded objects in the image.

Metric. We use symmetric mean percent error (sMAPE) as the primary metric. sMAPE is given by:

$$\text{sMAPE} = 100 \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|} \tag{1}$$

where y_i represents the actual values, $\hat{y}_i$ represents the pre-

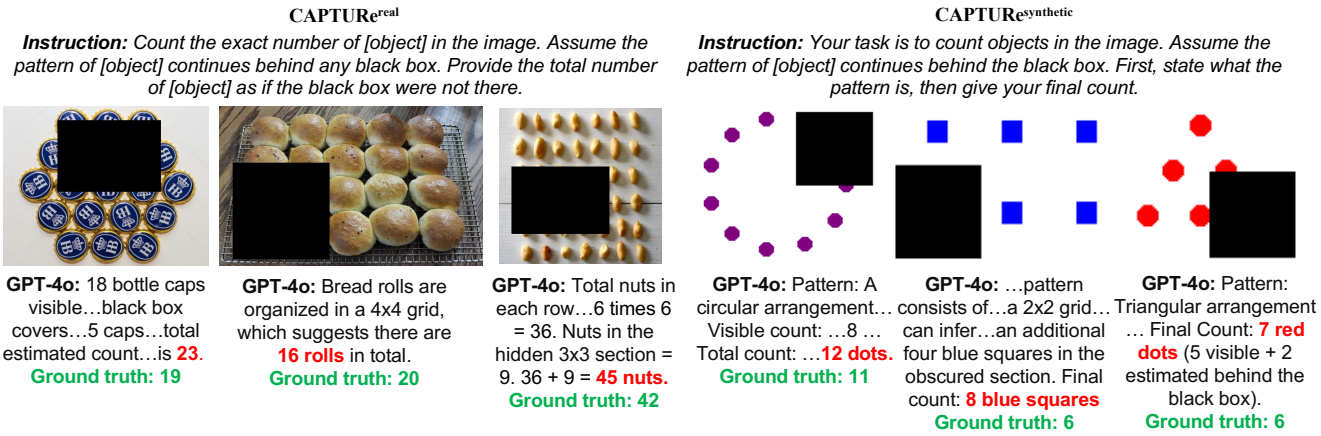

Figure 2. Example images with GPT-4o responses to CAPTURE^{real} and CAPTURE^{synthetic} occluded splits.

dicted values, and n is the number of observations. sMAPE is capped at 100%, providing a fixed range. This makes sMAPE ideal for challenging tasks like ours, as we can penalize responses that fail to produce an answer with a maximum error of 100%. For a justification of sMAPE over other metrics, see Appendix A.1.

2.2. Dataset

CAPTURE^{real}. We introduce a set of real images with patterns to test amodal counting in naturalistic settings. The original images and annotations come from the FSC-147 dataset [37], a diverse counting dataset with manual annotations for the number of target objects and all object bounding boxes in each image. FSC-147 contains a diverse array of objects, with 6146 real-world images across 147 object categories. We filter FSC-147 for images that contain identifiable and regular patterns of objects and manually overlay a black box to occlude some objects, resulting in 924 images. Filtering is first performed with GPT-4o and then manually verified; we also manually verify that determining objects despite the occlusion is feasible. For each example, we maintain both occluded and unoccluded versions. Further details on CAPTURE^{real} can be found in Appendix B.

CAPTURE^{synthetic}. While CAPTURE^{real} makes CAPTURE more applicable to real-world scenarios, each image is unique, making the data less controlled and challenging to draw clear conclusions about model performance. Images without background distractors, texture variance, and other potential visual obstacles provide a more controlled version of the task. Therefore, we create CAPTURE^{synthetic} to examine the task in a fully controlled environment. CAPTURE^{synthetic} comprises 1250 images of simple objects in patterns, where different variables are held constant or changed. We vary the following features:

1. **Object count**: varies from 5 to 15.
2. **Object**: can be either dots or squares.
3. **Arrangement/shape**: can be a rectangle, circle, or pyramid (where feasible based on object count).
4. **Location**: we consider five positions on the page: center, top-left, top-right, bottom-left, or bottom-right.
5. **Color**: we randomly choose one of 5 colors for all the objects in an image.

The CAPTURE^{synthetic} data is split similarly to the CAPTURE^{real} data; each configuration has a variant with an overlaid occluding box and one without.

	CAPTURE^{real}	CAPTURE^{synthetic}
# Images	924	1250
# Object Types	92	2
Avg. Occluded Obj.	13.97	2.73
Avg. Total Obj.	61.45	10.00
Strengths	Diverse Objects/Settings Naturalistic Realistic Context	Confounder-free Controllable Attributes Uniformly Distributed

Table 1. Statistics and strengths for CAPTURE splits.

2.3. Statistics and Examples

Fig. 2 shows examples from CAPTURE^{real} and CAPTURE^{synthetic} paired with their corresponding answers from GPT-4o and their ground truth answers. These examples show the range of objects and patterns in the dataset and highlight the task's feasibility for humans. Tab. 1 reports summary statistics for CAPTURE, including the number of images and object types, as well as the mean number of occluded and total objects in both splits of CAPTURE. The number of objects in CAPTURE^{real} is shown in Fig. 3, where most images have between 0 and 30 objects. On CAPTURE^{synthetic}, the maximum number of objects is 15, and CAPTURE^{synthetic} images generally have 1-6 occluded objects (shown in Fig. 4, as further occlusion could make the count unresolvable).

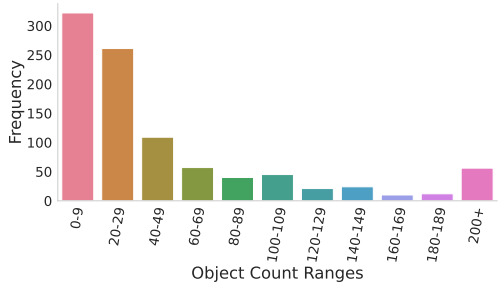

Figure 3. # of objects in CAPTURE^{real} images.

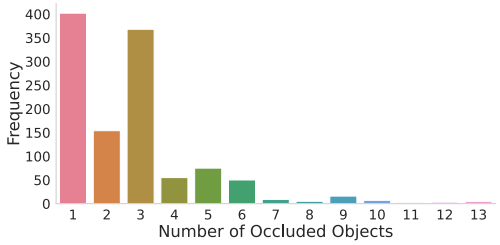

Figure 4. # of occluded objects in CAPTURE^{synthetic} images.

3. Experiment Setup

3.1. Models

We experiment with GPT-4o [28], Intern-VL2-Llama3-8B [9, 10], Qwen2-VL-7B [41], MiniCPM-o 2.6 [47], and Kimi-VL-A3B [40] for their high scores on other VLM tasks [29]. We add Molmo 7B-D [13], because of its ability to "point and count," giving it a potential advantage on CAPTURE. Specifically, Molmo is trained on millions of examples that directly ground text to 2D coordinates (or "points") in images. This allows Molmo to directly point to image coordinates and count more easily by pointing to several objects. All the VLMs feature a different language backbone and vision encoder to provide broad coverage of model architectures. To evaluate models, we provide the model with the name of the specific object to be counted and the explicit instruction to count fully visible objects and objects behind the occluding box (in the occluded images). For each model, we test ten prompts on a validation set of 100 images, selecting the best prompt for each model in each dataset section (CAPTURE^{real}/CAPTURE^{synthetic}) and for each environment (occluded/unoccluded). We provide the selected prompts in Appendix D.

3.2. Answer Generation and Extraction

Given the complex nature of CAPTURE, we allow models to generate open-ended responses and then subsequently extract answers. Further details (including the maximum number of tokens) can be found in Appendix A.2.

Answer extraction. Empirically, we found that constraining the output to a specific format for ease of analysis negatively impacted benchmark performance. Therefore, we instead prompt models to generate freely and extract the final output number using a separate answer extractor based on Llama 3.1 8B [1]. This answer extractor takes the output from the model as input and prompts it to extract a single number representing the final answer. The answer extractor also identifies if an output failed to converge on a singular number answer and assigns a label to these examples. We mark such incomplete/incoherent model generations as *'skipped'* questions and when calculating the error later, these responses are assigned the worst possible sMAPE score (100%). The answer extractor outputs were manually verified on 1000 outputs, and the extractor was found to be 100% accurate.

Human and object detection baselines. We also report the performance of humans and a recent counting model (COUNTGD [3]) as baselines to establish a point of reference for model performance. To confirm that humans can perform the CAPTURE task, we provided 100 randomly selected occluded examples each from the CAPTURE^{real} and CAPTURE^{synthetic} subsets to 3 undergraduate students with no prior knowledge of the task.

4. Results and Analysis

4.1. Main Results on CAPTURE^{real}

Models consistently struggle with counting and perform worse on occluded images. We run the VLMs on the occluded and unoccluded versions of CAPTURE to discern whether occlusion significantly impacts model performance. Tab. 2 shows that all models struggle with counting generally, performing poorly on both splits. Moreover, we see that **every model performs better on the unoccluded images.** On average, the models perform 6.28% worse in CAPTURE^{real} occluded images and 4.85% worse in CAPTURE^{synthetic} occluded images (in terms of absolute sMAPE), indicating increased difficulty from a standard counting task. The best model for both splits, GPT-4o, has an error rate of 14.75% on CAPTURE^{real} and a lower error rate of 9.71% on CAPTURE^{synthetic}. Across both the real and synthetic split, GPT-4o's error increases with occlusion, by 1.41% on the real data and 3.81% on the synthetic split. Interestingly, despite its fine-tuning on counting tasks, Molmo exhibits a sizable error rate of 32.5% on CAPTURE^{real} occluded images. The high error rates of VLMs indicate limited capabilities in visual understanding under occlusions, pattern recognition, and counting. We further analyze the source of these errors with oracle experiments in Sec. 4.3.

Humans complete the task with almost no error. Tab. 3, evaluated on a 100-example subset of each split, confirms that humans complete the task with ease despite occlusion,

Model	Error (%) [↓]			
	CAPTURE^{real}		CAPTURE^{synthetic}	
	Original	w/ Occlusion (Δ)	Original	w/ Occlusion (Δ)
GPT-4o	13.34	14.75 (+1.41)	5.90	9.71 (+3.81)
InternVL2	26.17	32.90 (+6.73)	16.44	17.57 (+1.13)
Molmo	25.90	32.49 (+6.59)	8.40	17.73 (+9.33)
Qwen2VL	18.96	29.33 (+10.37)	6.63	11.74 (+5.11)
MiniCPM-o 2.6	23.84	30.08 (+6.24)	17.06	19.00 (+1.94)
Kimi-VL-A3B	23.48	25.96 (+2.48)	16.91	18.07 (+1.16)
Avg. of 6 VLMs	21.95	27.59 (+5.64)	11.89	15.64 (+3.75)

Table 2. Results across VLMs on all splits of CAPTURE, with average error for each column. Metric: sMAPE (lower is better).

Model	Error (%) [↓]	
	CAPTURE^{real}	CAPTURE^{synthetic}
(Baseline)		
Human	3.79	0.92
(VLMs)		
GPT-4o	14.75	9.71
InternVL2	32.90	17.57
Molmo	32.49	17.73
Qwen2VL	29.33	11.74
Avg. of 4 VLMs	27.37	14.19

Table 3. Human baseline vs VLMs on CAPTURE^{real} and CAPTURE^{synthetic} (occluded split). Metric: sMAPE (lower is better).

with an sMAPE of 3.79% on CAPTURE^{real} and 0.92% on CAPTURE^{synthetic}. On the same subset of examples, **models performed 7 times worse on CAPTURE^{real} and 14 times worse on CAPTURE^{synthetic} than humans**, underscoring the gap between VLMs and humans in this task.

Object detection-based baseline outperforms VLMs. We attempt the task with a strong object detection-based model to highlight that a standard counting approach will experience a greater loss going from unoccluded to occluded environments, as it cannot capture any occluded objects, i.e. cannot reason. We choose COUNTGD [3], the top solution for unoccluded counting on FSC-147, on which it was trained. Because we draw our images from FSC-147's train and test sets, and COUNTGD trains on FSC-147, we only evaluate COUNTGD on the subset of our data sourced from the FSC-147 test set, consisting of 149 images.

We find that **COUNTGD deteriorates by 7.19% on occluded images**, increasing from 3.15% sMAPE to 10.34% as observed in Fig. 5. As expected, COUNTGD outperforms all VLMs on the unoccluded split as it is trained for counting on FSC-147. COUNTGD also outperforms the VLMs on the occluded split, reinforcing that only counting the visible objects is a hard-to-beat baseline. However, the drop in performance with occlusion is greater than the average

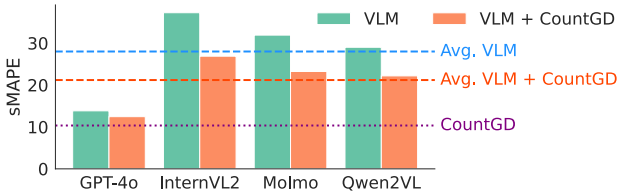

Figure 5. VLM vs. VLM + CountGD hybrid on questions from the CAPTURE^{real} (occluded split) that are not in COUNTGD training set. Metric: sMAPE (lower is better).

VLM's drop, highlighting a disadvantage of non-reasoning solutions on CAPTURE: their error is necessarily tied directly to the number of occluded objects and they cannot address the task on their own, whereas a VLM might be able to infer missing objects via reasoning.

Hybrid VLM counting systems improve performance. Finding that COUNTGD is far better at counting visible objects than VLMs, we leverage the advantage that COUNTGD has by feeding its visible object count information to the VLMs as part of the prompt. As expected, Fig. 5 illustrates that there is a considerable decrease in error when CountGD and the VLMs are combined. However, this hybrid system still performs worse than COUNTGD alone, indicating VLMs are still subpar even at counting just occluded objects (as further reinforced by Appendix C.3).

4.2. Effect of Data Factors on VLM Performance

Here, we use the CAPTURE^{synthetic} data (which can be controlled precisely and fully annotated) to examine which features correlate with model performance. We test the effect of the following variables on final performance: (1) Increasing the number of occluded objects; (2) Varying the pattern. We also investigate whether models can classify patterns, and to what degree models can predict the number of occluded objects only (rather than the total).

Models perform worse when more dots are occluded. In Fig. 6 (right), we observe that **error increases with re-**

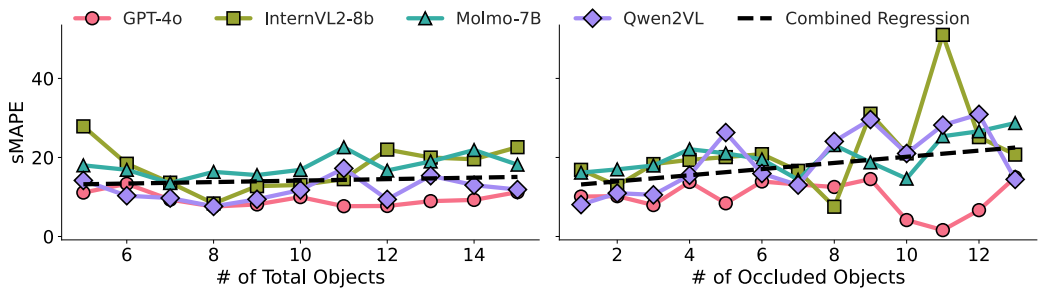

Figure 6. Effect of number of total objects in the image and number of occluded objects on sMAPE from CAPTURE^{synthetic} (occluded split). Metric: sMAPE (lower is better).

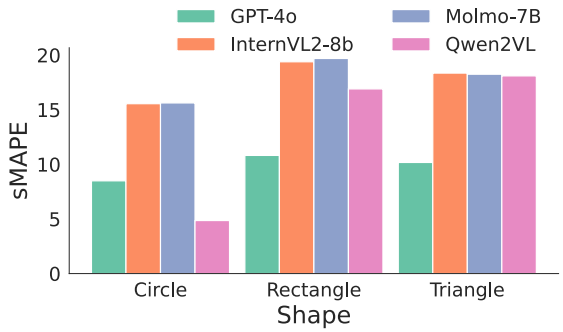

Figure 7. Effect of pattern type in CAPTURE^{synthetic} (occluded split) on sMAPE. Metric: sMAPE (lower is better).

spect to the number of occluded dots. However, Fig. 6 (left) also shows that performance is less affected by the total number of dots. This suggests that the task difficulty is more closely correlated with the difficulty of occlusion – i.e. the difficulty of the world modeling task – rather than the complexity of the pattern. Some models, such as GPT-4o, deviate from this trend, which has lower error on specific numbers. We further explore model bias in Appendix C.5.

Performance depends on pattern type. The controllability of CAPTURE^{synthetic} allows us to measure the effect of pattern type on performance. In Fig. 7, we find that model performance differs across shapes with some regularity: objects arranged in a circle generally have lower sMAPE than other shapes, across all models. Qwen2VL has an especially large decrease in error when given circular arrangements compared to rectangles or triangles.

Model	Accuracy (%) [↑]	
	Original	w/ Occlusion (Δ)
GPT-4o	84.00	78.52 (-5.48)
InternVL2	68.52	47.48 (-21.04)
Molmo	80.70	65.22 (-15.48)
Qwen2VL	88.35	86.43 (-1.92)
Avg. of 4 VLMs	80.39	69.41 (-10.98)

Table 4. VLM accuracy in identifying the correct pattern in CAPTURE^{synthetic}. Metric: accuracy (higher is better).

Models can identify patterns. To determine how much model errors can be attributed to a lack of pattern recognition ability, we formulate a separate task where models must recognize the pattern in the image on CAPTURE^{synthetic}. Here, we frame the task as multiple-choice, asking the model to select from the pattern types available (rectangle, triangle, or circle). Table 4 illustrates that all perform substantially better than random at this task, with most models except InternVL2 achieving accuracy above 80% in the unoccluded setting. As expected, the patterns were easier to identify in unoccluded scenarios, with models suffering an average accuracy drop of 10.95% in the occluded setting. Notably, GPT-4o and Qwen2VL have a fairly small drop in performance, suggesting they can generally capture the pattern even in the presence of occlusion.

4.3. Analysis with Auxiliary Information

In Sec. 4.1, we see that models broadly struggle with amodal counting. Here, we seek to disentangle whether this problem results from a failure to reason, the absence of a world model, or both by giving VLMs two different types of auxiliary information: *oracle* information and *predicted* information. Oracle information is ground truth and is directly pulled from CAPTURE's metadata, e.g., object locations. Predicted information generates new information from a completely separate model and gives it to the VLM. This information is not ground truth and is sourced from an external model, such as an image inpainting model, rather than the VLM. By giving the model auxiliary information in the form of reasoning and spatial clues, we can establish how much of each model's error results from an inability to handle occlusion rather than an inability to recognize and count visible objects.

Oracle setup. We test two oracles for CAPTURE^{real}'s occluded split based on its constituent subtasks: counting the visible objects and inferring/counting occluded objects. Both oracles provide the VLM with text-based coordinates

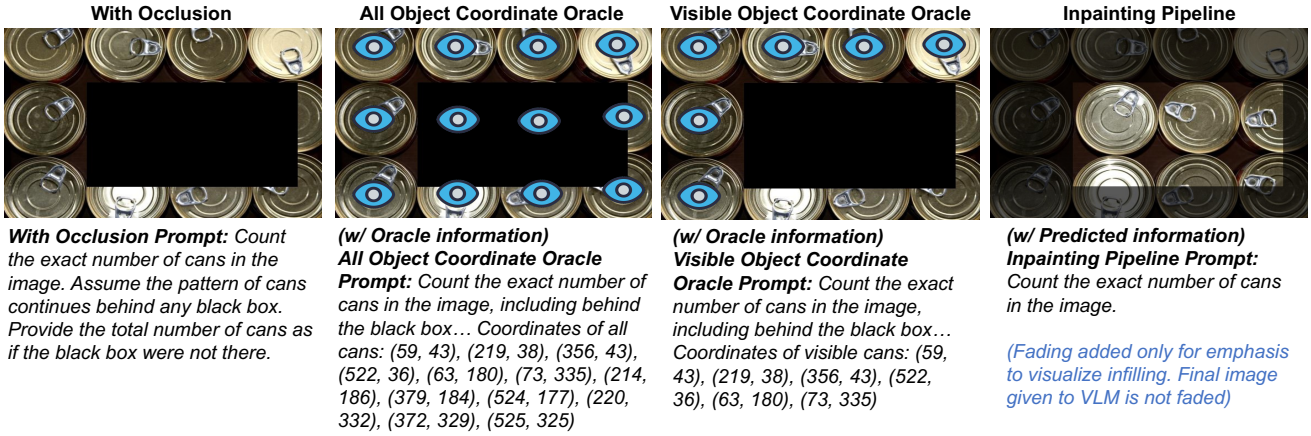

Figure 8. Example image and text inputs for experiments with auxiliary information experiments (Sec. 4.3). Blue eyes indicate objects for which the *All Object Coordinate Oracle* or *Visible Object Coordinate Oracle* extracts coordinates. The brighter part of the image represents the area which *Inpainting Pipeline* fills in. Example prompts are shown in italics. Blue eye overlays and faded parts of images are for demonstration purposes and are not passed with the image.

Model	Original	w/ Occlusion	Oracle Information		Predicted Information
			+ All Coordinates (Δ)	+ Visible (Δ)	+ Inpainting (Δ)
GPT-4o	13.34	14.75	2.93 (-11.82)	9.20 (-5.55)	15.89 (+1.14)
InternVL2	26.17	32.90	17.48 (-15.42)	25.13 (-7.77)	31.12 (-1.78)
Qwen2VL	18.96	29.33	9.62 (-19.71)	17.70 (-11.63)	22.64 (-6.69)
Avg. of 3 VLMs	19.49	25.66	10.01 (-15.65)	17.34 (-8.32)	23.22 (-2.44)

Table 5. Effect of auxiliary information on occluded CAPTURE^{real}. Δ = (Auxiliary Information) − (w/ Occlusion). Metric: sMAPE.

of objects in the image, simplifying the visual task by assuming the VLM effectively has a perfect visual system that can recognize and localize objects in the image. The first oracle, the *Visible Object Coordinate Oracle*, gives the VLM the coordinates of all unoccluded objects (encoded as text, as seen in Fig. 8) and instructs the model to estimate the number of occluded objects, count the number of visible object coordinates, and add the two. In other words, the model is given oracle information about what objects are visible, thus also revealing key information about the pattern. The second oracle, the *All Object Coordinate Oracle*, instead gives the model the coordinates of all objects. Here, the model only needs to count the coordinates in the prompt, eliminating the need to reason on the visual input. Note that Molmo is excluded in these tests because it contains a prompt limit that would truncate the list of coordinates. An example of the oracle inputs can be seen in Fig. 8.

Prediction setup. In this setting, we provide the VLM with an external world model representation predicted by another model. Specifically, we develop the *Inpainting Pipeline* to fill in the occluded region via a diffusion-based inpainting model and pass the inpainted image to the VLMs. For the inpainting model, we choose FLUX.1-Fill [dev], whose backbone FLUX.1 [dev] [21] is a top public model in the Text to Image Model Arena [7]. An example input to the VLM can be seen on the far-right of Fig. 8.

Providing visible or all object coordinates improves performance substantially. The results in Tab. 2 indicate that models struggle on CAPTURE, which requires identifying a pattern and counting both visible and occluded objects. Moreover, models generally struggle with counting even in unoccluded settings. Both oracles simplify the counting task: *All Object Coordinate Oracle* reduces the task to simply counting coordinates with no reasoning involved, and *Visible Object Coordinate Oracle* similarly simplifies the task for visible objects, while still requiring inferring occluded objects. Additionally, under *Visible Object Coordinate Oracle*, recognizing the pattern shifts from a visual reasoning task to an augmented math problem. Instead of visually reasoning about where objects are located, the VLM considers what patterns the coordinates could make. Translating this task into a text problem results in an average increase of 15% with all objects coordinate oracle; the errors LLMs make here are due to an inability to count in the text prompt, as opposed to weaknesses in handling occlusion (since all object coordinates are given), and the strongest

model, GPT-4o, achieves minimal error here. We also obtain an average increase of 8% with the visible objects coordinate oracle (shown in Tab. 5), possibly because it allows the more powerful LLM backbone (which is far larger than the vision model in all models tested) to complete the counting task. Taken together, these results suggest that there is much room for improvement in visual world modeling beyond text-based reasoning of VLMs.

Providing diffusion-based inpainting improves performance marginally. Similar to the object coordinate oracles, the Inpainting Pipeline (rightmost columns in Fig. 8 and Tab. 5) eliminates the need for world modeling and provides VLMs with an approximation of the image behind the occluder. With the inpainted images, VLM error decreases by almost 2% for InternVL2 and 7% for Qwen2VL compared to the original occluded images. GPT-4o's error increases on inpainted images by a small margin; we hypothesize that this may be because GPT-4o has one of the better world models (based on its superior performance), and thus does not improve further with the inpainted images. Moreover, every VLM still falls short of its unoccluded image performance, indicating that the diffusion model is not a perfect world model. Qualitatively, we find that the inpainting model sometimes fails to output the correct pattern.

5. Related Work

Spatial reasoning in visual question answering. Past work measures the spatial reasoning capabilities of VLMs in the form of visual question answering (VQA) [4, 16] benchmarks. SpartQA [26] asks VLMs to identify the spatial relation (e.g., above, behind, left of) between objects in synthetically created 2D images from NLVR [39]. More recent benchmarks test similar spatial relation understanding with real images [2, 24, 36]. While this past work asks models to provide a text description for a relation between two fully observed objects, CAPTURE measures the world modeling from a partially observed scene, thus requiring the handling of occlusion, pattern recognition, and counting. Together, these constitute a stricter test of spatial reasoning than typical VQA settings.

Amodal completion. Occlusions are common in natural scenes, and vision solutions for amodal completion have made significant progress in infilling occlusions [6, 38, 46]. The amodal completion task has evolved from simply completing a shape to filling in appearance (e.g., texture, color, etc.) to finally dealing with fine-grained order perception (multiple stacked occluded objects) [5]. Specifically in Qiu and Di [34], VLMs classify the hidden objects and extract fine details from occluded items. CAPTURE, however, presents a unique category of *patterned* amodal counting which requires inferring fully occluded objects based on a pattern rather than inferring occluded object wholes based on object parts. In other words, previous work has only attempted tasks that require amodal completion for one object at a time [31, 38, 46], whereas CAPTURE handles multiple objects. Multi-object amodal completion is crucial because in cluttered scenes, entire groups of objects are often occluded. Moreover, the output space of CAPTURE is language (rather than filling pixels).

Counting with vision-and-language models. Within the task of counting, the most similar application to CAPTURE is dense counting, where the objects to be counted occlude each other. There are many practical applications of such a task, like counting cells on a crowded slide [8], determining crop yields from densely-packed fields [43], or crowd counting [14, 44, 48]. Liang et al. [23] improved crowd counting with an augmented CLIP [35], i.e. also using VLMs for counting. Additionally, Jenkins et al. [18] introduced an amodal counting benchmark, presenting an occluded 3D counting task where models must count objects on retail shelves. However, our work differs in many ways, as Jenkins et al. [18] only counts retail shelves and uses LiDAR input. More broadly, dense counting focuses on overlapping objects rather than on counting objects arranged into patterns, which is the focus of CAPTURE.

6. Conclusion

We introduced CAPTURE, a novel benchmark for amodal counting that measures spatial reasoning capabilities under occlusion. CAPTURE is designed to assess VLMs' ability to form a robust world model and use that model for visual reasoning skills under occlusion. By testing counting, we cast the problem as a measurable task with an objective correct answer that also has real-world utility as VLMs become more broadly adopted. Our results suggest that **VLMs struggle to combine reasoning, counting, and world modeling** with low performance on occluded and unoccluded images. Our analysis indicates that models improve with oracle information about visible objects (simplifying the reasoning/counting tasks) and predicted information about the occluded objects (also simplifying world modeling), pointing to directions of model improvement.

Acknowledgments

This work was supported by DARPA ECOLE Program No. HR00112390060, NSF-CAREER Award 1846185, NSF-AI Engage Institute DRL-2112635, DARPA Machine Commonsense (MCS) Grant N66001-19-2-4031, ARO Award W911NF2110220, ONR Grant N00014-23-1-2356, Microsoft Accelerate Foundation Models Research (AFMR) grant program, and a Bloomberg Data Science PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

[1] AI@Meta. Llama 3.1 model card. *Github Model Card*, 2024. 4

[2] Haider Al-Tahan, Quentin Garrido, Randall Balestriero, Diane Bouchacourt, Caner Hazirbas, and Mark Ibrahim. Unibench: Visual reasoning requires rethinking vision-language beyond scaling. *arXiv preprint arXiv:2408.04810*, 2024. 8

[3] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *arXiv preprint arXiv:2407.04619*, 2024. 2, 4, 5

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 8

[5] Jiayang Ao, Qiuhong Ke, and Krista A Ehinger. Image amodal completion: A survey. *Computer Vision and Image Understanding*, 229:103661, 2023. 1, 8

[6] Jiayang Ao, Yanbei Jiang, Qiuhong Ke, and Krista A Ehinger. Open-world amodal appearance completion. *arXiv preprint arXiv:2411.13019*, 2024. 8

[7] Artificial Analysis. Text to image model arena, 2025. Accessed: April 10, 2025. 7

[8] Soumen Bera. Partially occluded object detection and counting. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pages 1–6. IEEE, 2015. 8

[9] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 4

[10] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 4

[11] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021. 11

[12] Nikolas Coupland. How frequent are numbers? *Language & Communication*, 31(1):27–37, 2011. 13

[13] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024. 4

[14] Zheyi Fan, Zihao Song, Di Wu, and Yixuan Zhu. Multi-branch segmentation-guided attention network for crowd counting. *Journal of Visual Communication and Image Representation*, 97:103964, 2023. 8

[15] Benito E Flores. A pragmatic view of accuracy measurement in forecasting. *Omega*, 14(2):93–98, 1986. 11

[16] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 8

[17] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018. 1

[18] Porter Jenkins, Kyle Armstrong, Stephen Nelson, Siddhesh Gotad, J Stockton Jenkins, Wade Wilkey, and Tanner Watts. Countnet3d: A 3d computer vision approach to infer counts of occluded objects. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3008–3017, 2023. 8

[19] Gaetano Kanizsa, Paolo Legrenzi, and Paolo Bozzi. *Organization in vision : essays on gestalt perception*. Praeger, 1979. 1

[20] Kaleb Kassaw, Francesco Luzi, Leslie M Collins, and Jordan M Malof. Are deep learning models robust to partial object occlusion in visual recognition tasks? *arXiv preprint arXiv:2409.10775*, 2024. 2

[21] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 7

[22] Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *arXiv preprint arXiv:2410.14669*, 2024. 2

[23] Dingkang Liang, Jiahao Xie, Zhikang Zou, Xiaoqing Ye, Wei Xu, and Xiang Bai. Crowdclip: Unsupervised crowd counting via vision-language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2893–2903, 2023. 8

[24] Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 2023. 8

[25] Baraka Jacob Maiseli. Optimum design of chamfer masks using symmetric mean absolute percentage error. *EURASIP Journal on Image and Video Processing*, 2019(1):74, 2019. 11

[26] Roshanak Mirzaee and Hossein Rajaby. Spartqa: A textual question answering benchmark for spatial reasoning. In *The 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2021)*, 2021. 8

[27] Ingrid R Olson, J Christopher Gatenby, Hoi-Chung Leung, Pawel Skudlarski, and John C Gore. Neuronal representation of occluded objects in the human brain. *Neuropsychologia*, 42(1):95–104, 2004. 1, 2

[28] OpenAI. Hello gpt-4o, 2024. 4

[29] OpenCompass Team. Openvlm leaderboard. https://huggingface.co/spaces/opencompass/open_vlm_leaderboard, 2024. Accessed: 2024-11-13. 4

[30] Yumiko OTSUKA, So KANAZAWA, and Masami K YAMAGUCHI. Development of modal and amodal completion in infants. *Perception (London. Print)*, 35(9):1251–1264, 2006. 1, 2

[31] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3931–3940. IEEE Computer Society, 2024. 8

[32] Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*, 2024. 12

[33] Muhammad Fetrat Qharabagh, Mohammadreza Ghofrani, and Kimon Fountoulakis. Lvlm-count: Enhancing the counting ability of large vision-language models. *arXiv preprint arXiv:2412.00686*, 2024. 2

[34] Wenmo Qiu and Xinhan Di. Occ-mllm: Empowering multimodal large language model for the understanding of occluded objects. *arXiv preprint arXiv:2410.01261*, 2024. 8

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 8

[36] Navid Rajabi and Jana Kosecka. Gsr-bench: A benchmark for grounded spatial reasoning evaluation via multimodal llms. *arXiv preprint arXiv:2406.13246*, 2024. 8

[37] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021. 3

[38] Kaziwa Saleh, Sándor Szénási, and Zoltán Vámossy. Mask guided gated convolution for amodal content completion. In *2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000321–000326. IEEE, 2024. 8

[39] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada, 2017. Association for Computational Linguistics. 8

[40] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang, Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, and Ziwei Chen. Kimi-VL technical report, 2025. 4

[41] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 4

[42] Wei-Yao Wang, Zhao Wang, Helen Suzuki, and Yoshiyuki Kobayashi. Seeing is understanding: Unlocking causal attention into modality-mutual attention for multimodal llms. *arXiv preprint arXiv:2503.02597*, 2025. 2

[43] Yiding Wang, Yuxin Qin, and Jiali Cui. Occlusion robust wheat ear counting algorithm based on deep learning. *Frontiers in Plant Science*, 12:645899, 2021. 8

[44] Yongjie Wang, Feng Wang, and Dongyang Huang. Dual-branch counting method for dense crowd based on self-attention mechanism. *Expert Systems with Applications*, 236:121272, 2024. 8

[45] Karen Wynn. Children's understanding of counting. *Cognition*, 36(2):155–193, 1990. 1, 2

[46] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Amodal completion via progressive mixed context diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9099–9109, 2024. 8

[47] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 4

[48] Lifang Zhou, Songlin Rao, Weisheng Li, Bo Hu, and Bo Sun. Multi-branch progressive embedding network for crowd counting. *Image and Vision Computing*, page 105140, 2024. 8