StateVLM: A State-Aware Vision-Language Model for Robotic Affordance Reasoning

Xiaowen Sun[1*], Matthias Kerzel[1†], Mengdi Li[2†], Xufeng Zhao[1], Paul Striker[1], Stefan Wermter[1]

[1]Department of Informatics, University of Hamburg, Hamburg, 20146, Germany.

[2]King Abdullah University of Science and Technology, Thuwal, 23955, Saudi Arabia.

*Corresponding author(s). E-mail(s): xiaowen.sun@uni-hamburg.de;
Contributing authors: matthias.kerzel@uni-hamburg.de;
li_mengdi@hotmail.com; xfz.zhao@gmail.com;
paul.jonas950@gmail.com; stefan.wermter@uni-hamburg.de;
†These authors contributed equally.

Abstract

Vision-language models (VLMs) have shown remarkable performance in various robotic tasks, as they can perceive visual information and understand natural language instructions. However, when applied to robotics, VLMs remain subject to a fundamental limitation inherent in large language models (LLMs): they struggle with numerical reasoning, particularly in object detection and object-state localization. To explore numerical reasoning as a regression task in VLMs, we propose a novel training strategy to adapt VLMs for object detection and object-state localization. This approach leverages auxiliary regression head outputs to compute an Auxiliary Regression Loss (ARL) during fine-tuning, while preserving standard sequence prediction at inference. We leverage this training strategy to develop StateVLM (State-aware Vision-Language Model), a novel model designed to perceive and learn fine-grained object representations, including precise localization of objects and their states, as well as graspable regions. Due to the lack of a benchmark for object-state affordance reasoning, we introduce an open-source benchmark, Object State Affordance Reasoning (OSAR), which contains 1172 scenes with 7746 individual objects and corresponding bounding boxes. Comparative experiments on adapted benchmarks (RefCOCO, RefCOCO+, and RefCOCOg) demonstrate that ARL improves model performance by an average of 1.6% compared to models without ARL. Experiments

on the OSAR benchmark further support this finding, showing that StateVLM with ARL achieves an average of 5.2% higher performance than models without ARL. In particular, ARL is also important for the complex task of affordance reasoning in OSAR, where it enhances the consistency of model outputs.

Keywords: Vision-language models, Referring expression comprehension, Visual perception, Object-state understanding, Affordance reasoning

1 Introduction

Vision-language models (VLMs) [1, 2] for robotics are not only capable of natural language understanding but also of visual perception. In robotic tasks such as manipulation, human-robot interaction, and autonomous driving, precise localization and understanding of objects within a scene are essential. However, VLMs applied to robotics remain subject to a fundamental limitation inherent in large language models (LLMs): they struggle with numerical reasoning, particularly object detection [3, 4] and object-state localization [5–7].

Most important VLM architectures adopt LLMs as their backbone and integrate visual perception modules within them [8–16]. LLMs are originally trained on sequence prediction. The existing VLMs focused on object detection tasks use a sequence-based output format, which is the LLMs' default. For instance, Pix2Seq [17] is an object detection framework that predicts a sequence of discrete tokens that correspond to object descriptions (e.g., object bounding boxes and class labels, $y_{\texttt{min}} = 9$ $x_{\texttt{min}} = 7$ $y_{\texttt{max}} = 67$ $x_{\texttt{max}} = 98$ `train`.....). State-of-the-art VLMs also formulate object detection as a sequence prediction task, including SPHINX [8], Shikra [9], LLAVA-G [12], and others. However, such a representation is suboptimal because discrete symbolic tokens require sequence modeling, whereas continuous numerical outputs are better modeled with regression. **We hypothesize that using this Pix2Seq training paradigm for numerical tasks may result in inefficient learning**.

To explore object detection as a regression task in VLMs, Zhang et al. proposed the Pix2Emb method, NExT-Chat, which introduces a box decoder for continuous numerical outputs as embeddings [18]. The key distinction between the Pix2Seq and Pix2Emb methods lies in their outputs: the Pix2Seq method generates sequences only, whereas Pix2Emb method generates both sequences and embeddings. However, despite adopting a similar training data regime for detection training, NExT-Chat performs slightly worse than the Pix2Seq method, Shikra-7B [9], on referring expression comprehension (REC). Zhang et al. propose two hypotheses to explain this gap. First, achieving an optimal balance between sequence and embedding losses is challenging, whereas Pix2Seq methods do not encounter this issue, as they optimize only the sequence loss. Second, LLMs are not pre-trained on regression tasks, which may further increase training difficulty. To further investigate this open question in VLMs, we propose a less intrusive strategy than the Pix2Emb method for adapting VLMs to object detection and object-state localization.

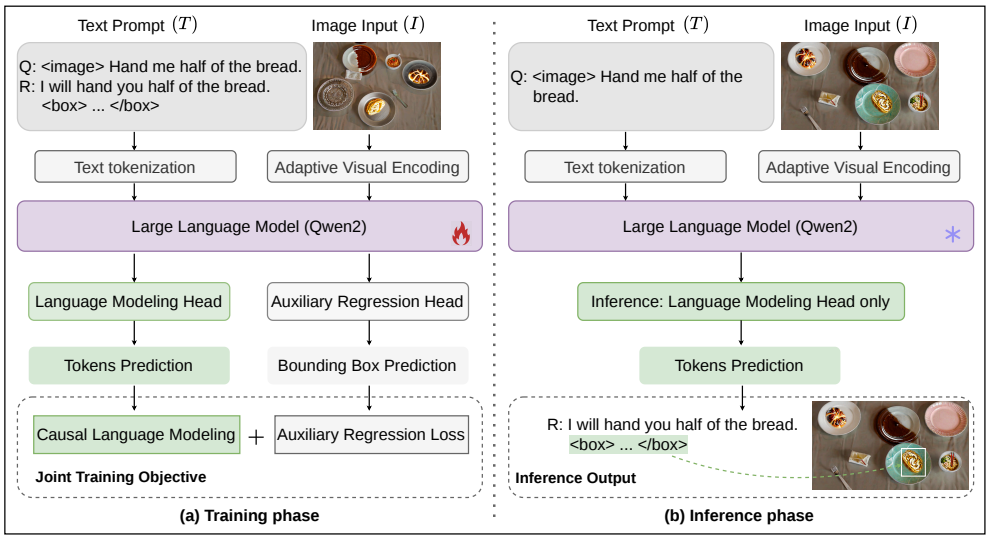


Fig. 1: During the **training phase (a)**, a specially designed auxiliary regression head converts sequence predictions into embedding predictions for the computation of an auxiliary regression loss, which is jointly optimized with causal language modeling. During the **inference phase (b)**, however, the model continues to rely on the language modeling head to generate sequences composed of text and bounding boxes, since the LLM backbone of the VLM is originally trained for sequence prediction.

Previously, there have been some benchmarks that focused on object-state recognition and classification [19–22] and fewer studies on object-state localization [23, 24]. Object-state localization is essential for robotic tasks such as object manipulation, which require affordance reasoning at the object-state level [25, 26]. State-level reasoning is significantly more challenging than category-level reasoning because object states are dynamic and can change over time [21, 27, 28]. For instance, two dirty plates may require different grasping areas depending on the specific locations of the dirt. Similarly, a knife can be entirely clean or dirty, or its handle and blade may exhibit different cleanliness conditions. In some causal scenarios, if a knife is accidentally dropped into food, it may have a clean blade but a dirty handle. In this special case, the knife's blade becomes the appropriate grasping area. Therefore, to address the lack of a benchmark for object-state affordance reasoning, we introduce an open-source benchmark that focuses on object-state localization and affordance reasoning.

Our primary contributions in this paper are summarized as follows:

1. We propose StateVLM, a novel model designed to perceive and learn fine-grained object representations, including precise localization of objects and their states, as well as graspable regions.
2. To further investigate object detection as a regression task in VLMs, we propose a novel strategy to adapt VLMs to object detection. The model uses an auxiliary

regression head output to compute an Auxiliary Regression Loss (ARL) during training, while during inference it proceeds with standard sequence prediction, as illustrated in Fig. 1.

3. Comparative experiments on StateVLM demonstrate that this ARL significantly improves the StateVLM's convergence during fine-tuning, thereby enhancing its ability to learn object location features. The experiments were conducted on the adapted referring expression comprehension (REC) task. For this task, we used the following datasets: RefCOCO, RefCOCO+ [6], and RefCOCOg [7].
4. To address the lack of a benchmark for object-state affordance reasoning, we introduce an open-source benchmark OSAR to the research community, focused on object-state localization and affordance reasoning. It comprises both complex and simple scenes, totaling 1172 scenes, and contains 7746 individual objects, 25401 referring expressions, and corresponding bounding boxes. LoRA fine-tuning of StateVLM on the OSAR benchmark shows the effectiveness of ARL in improving the model's state-aware capabilities.

The remainder of this paper is structured as follows: Section 2 reviews existing approaches to tackle object-state understanding and reasoning and highlights the limitations of current methods. Section 3 describes the task definition and provides a summary of our proposed benchmark, named OSAR. The architecture and training procedure of StateVLM are detailed in Section 4. Section 5 presents and analyzes the experimental results, then discusses our findings, highlighting the role of the auxiliary regression loss in improving performance. Finally, Section 6 concludes the paper and suggests directions for future work.

2 Related Work

2.1 VLMs for Referring Expression Comprehension

REC (Referring Expression Comprehension) is a fundamental region-level image understanding task that seeks to identify and localize a target object described by a natural language expression within a given scene [6, 7]. In recent years, State-of-the-art VLMs have achieved remarkable performance on this task [8–16], significantly surpassing traditional non-VLM approaches [29–34]. This improvement can largely be attributed to the integration of visual representations or embeddings into LLMs. Due to their training on massive and diverse text corpora, LLMs exhibit strong linguistic interpretation abilities and extensive commonsense reasoning, both of which are crucial for advancing embodied AI and robotic perception. Nevertheless, these models demand substantial computational resources and prolonged multi-stage training processes, which present notable practical challenges.

The computational resources and training times of the most representative VLMs for the REC task are different. Depending on the number of transformer layers in the LLM and the scale of the visual backbone, they typically contain either 7B(illion) or 13B parameters. Furthermore, the models adopt several different training configurations. First, they are trained on computational clusters of varying sizes, such as 8, 32, or 256×A100 GPUs. Second, they employ various training strategies, including

one-stage, two-stage, or multi-stage training. Third, training durations are reported inconsistently, with some measured in wall-clock time and others in training steps.

Therefore, it is difficult to draw a rigorous scientific conclusion because we lack controlled variables [35]. It remains unclear whether simply combining two distinct data distributions (discrete text and continuous location data) results in inefficient training for bounding box coordinate prediction.

2.2 Affordance Reasoning: From Object Categories to Properties and States

Object affordance refers to the range of actions an agent can perform with an object, as perceived through its properties [26, 36]. A robust understanding of object categories (e.g., 'a cup', 'a plate', and 'an apple') provides a foundational basis for such reasoning. Building on the object categories, a model can further infer an object's potential affordances by leveraging finer-grained information, object attributes.

Object attributes, as generalizable properties, are central to this reasoning process. For instance, instead of relying solely on category recognition, Yang et al. [28] developed a robotic grasping method based directly on object attributes. Similarly, Attr-POMDP [37] presents an attribute-guided formulation of a partially observable Markov decision process for task disambiguation. In the Octopi system [38], the authors selected hardness, roughness, and bumpiness as key physical attributes for physical reasoning. Another prominent direction in affordance reasoning, particularly for robotic manipulation, involves physically grounded methods [39–41]. PhyGrasp [42] is designed to accurately assess the physical properties of object parts to determine optimal grasping poses. This finer-grained information is complex, and some of its values are dynamic for any given object.

We ground the concept of an object state in object-oriented programming [43], where it is defined as follows: "The state of an object encompasses all of the (usually static) properties of the object, plus the current (usually dynamic) values of each of these properties." Therefore, in our context, the properties refer to object properties (usually static), such as color, shape, size, weight, texture, cleanliness, and rigidity. The values of these properties are dynamic, meaning they can change over time or remain constant, and different properties may change at different rates. Our method centers on understanding an object's state and reasoning about its affordances, which has been less explored so far.

2.3 The Missing Link Between Object-State and Manipulation

Previous benchmarks have primarily focused on object-state recognition and classification [19]. For example, the cooking state recognition challenge [20] focused on 18 types of objects and their corresponding states, such as diced, grated, or creamy paste. Another dataset [21] classified whether a container was full or half empty. Furthermore, the Object State Detection Dataset [22] includes a larger number of objects and state classes.

Another set of datasets focuses on detecting or anticipating object state changes. For example, a subset of Ego4D [27] addresses object state change classification and

detection. Ego4D-OSCA [44] is a dataset for anticipating object state changes from Ego4D video sequences. The OSCaR dataset [24] focuses on object state captioning and state change representation. State-aware Keypoint Trajectories (SKT) [23] provides a synthetic dataset encompassing a broad spectrum of garment configurations, ranging from flat to deformed and folded states. However, none of these datasets explicitly focus on how an object's state influences its manipulation.

Overall, further investigation is needed to determine whether simply combining discrete and continuous data distributions leads to inefficient training for bounding-box coordinate prediction in VLMs. Designing a multimodal dataset is essential for analyzing this hypothesis at the object-state level, especially for affordance-based reasoning for object manipulation.

3 Multimodal Dataset for Object State Affordance Reasoning

Robotic manipulation can be decomposed into macro- and micro-level tasks. The macro-level task involves task planning, where the robot identifies objects and reasons about their destinations. The micro-level task involves motion planning for grasping and placing the object, as shown in Fig. 2. Our previous study, OSSA [45], focused on the macro-level task of robotic task planning. However, this study did not address object-state localization and affordance reasoning, both of which are crucial for robotic manipulation. To investigate this missing aspect, we propose and make available a novel benchmark, **Object State Affordance Reasoning (OSAR)**.

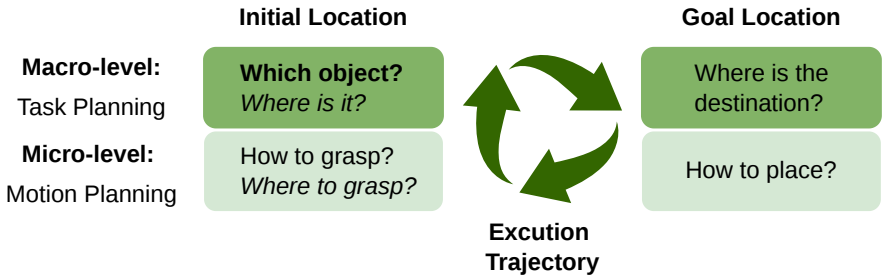


Fig. 2: Decomposition of robotic manipulation into macro- and micro-level tasks. The macro-level task involves task planning, identifying objects, and their destinations. The micro-level task involves motion planning, grasping, and placing the objects. While an object's state can affect all steps, we focus on its impact on localization, specifically determining the target object, its position, and the grasping location.

Table 1: Dialog templates in OSAR. The placeholder [target] is instantiated with the state of objects present in the corresponding scenes, such as 'empty plate', 'dirty plate', or 'bowl with noodles'. The bounding box coordinates $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$ denote the object location in the object detection task and the grasp location in the affordance reasoning task, where (x_1, y_1) corresponds to the top-left corner and (x_2, y_2) corresponds to the bottom-right corner.

Task	**Conversations**	
Types	**Roles**	**Content**
Object detection (object-state localization)	User:	"Show me the [target]."
	StateVLM:	"response: Here is the [target]. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."
	User:	"Where is the [target]?"
	StateVLM:	"response: Here is the [target]. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."
	User:	"Where is the location of the [target]?"
	StateVLM:	"response: Here is the location of the [target]. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."
Affordance reasoning (object grasp prediction)	User:	"Hand me the [target]."
	StateVLM:	"response: Sure. I will hand you the [target]. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."
	User:	"Pass me the [target]."
	StateVLM:	"response: Alright, I will pick up the [target] for you. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."
	User:	"Give me the [target]."
	StateVLM:	"response: Okay, I will give you the [target]. $\langle \text{box} \rangle x_1, y_1, x_2, y_2 \langle /\text{box} \rangle$."

3.1 Task Definition

Object Detection: Object-State Localization

Classic referring expression comprehension benchmarks, including RefCOCO, RefCOCO+, RefCOCOg [6, 7], and Flickr30K [46], focus on object category-level understanding, spatial identification, and grounding. However, they do not explicitly address object-state expressions used to refer to object locations, which are crucial for robotic manipulation. To address this gap, we construct a new referring expression comprehension dataset for object-state localization. The expressions provide explicit descriptions of the target object with respect to its state and spatial location. The dialog format is shown in Table 1, where the expressions are instantiated using object states present in the corresponding scenes, such as 'empty plate', 'dirty plate', and 'bowl with noodles', as illustrated in Table 2 and Fig. 4.

Affordance Reasoning: Object Grasp Prediction

In contrast to object-state localization, object grasp prediction requires consideration of physical common sense. For instance, in everyday kitchen scenarios, references are typically made to the object itself rather than to its container. During physical interaction, the decision of whether to grasp a container or its contents depends on the properties and state of the contained item. Specifically, one would grasp a container (e.g., a bowl) to pass noodles, whereas passing an apple does not require a container. Accordingly, we introduce the second task, termed object grasp prediction, whose format is identical to that of object detection (see Table 1 for affordance reasoning), except that the predicted bounding box corresponds to the region intended for grasping.

Table 2: Ambiguous opposites and synonyms in OSAR: Tableware states reflect the shifting semantics of use and cleanliness.

Object Category	Physical Condition	Semantic Interpretation
Plate, Bowl Cup, Mug, Glass	Empty, no residue	clean / unused
	Empty with little residue	used / dirty
	With / holding contents	used / dirty
Bottle	Cap open	possibly used (depends on liquid level)
	Cap closed / unopened	possibly unused (depends on liquid level)
Spoon, Fork, Knife	No residue	clean / unused
	Has residue	dirty / used
Napkin	Has residue Unfolded	used / dirty
	No residue Folded neatly	clean / unused

3.2 Benchmark Dataset

3.2.1 Visual Scenes

We use the Stable Diffusion model [47] to generate images and render them in Blender[1]. In total, we selected 1172 images based on quality, comprising 780, 267, and 125 images for templates (a), (b), and (c), respectively, as illustrated in Fig. 3. From each scene, we selected 11.2% of the images as test samples. The final dataset comprises 7746 individual objects, of which 11.17% belong to the test set. These objects were in different states within their respective categories, as shown in Fig. 4.

(a) (b) (c)

Fig. 3: Example scenes in OSAR: (a) **Simple scenes** are defined as those with only one object from each category. (b) and (c) **Complex scenes** are defined as those with multiple objects from each category in various states. The green box marks the ideal grasp region; this is an example of how an object's state affects its affordance, as the area covered with food should be avoided when grasping.

[1] https://www.blender.org/

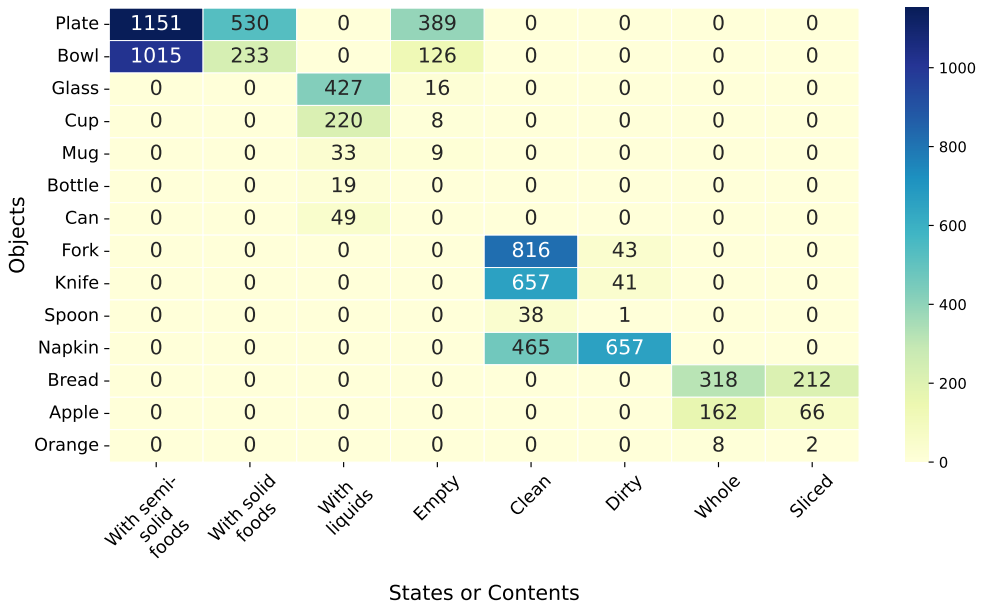


Fig. 4: Object statistics in OSAR. Semi-solid foods usually include sauces, pasta, soup, noodles, and creams. Solid foods typically refer to various states of apples, bread, and oranges. Liquids encompass a variety of drinks, beverages, and juices.

3.2.2 Annotation Rules

When assessing the state of kitchen tools, multiple dimensions must be considered, including contents, usage, hygiene, physical condition, position, accessibility, and operational status. OSAR specifically focuses on `clearing the table` after a meal. Our scope within this task is defined as follows: containers are analyzed based on their contents, specifically whether they are empty or contain food, and what type of food they contain. Cutlery and napkins are evaluated based on hygiene and readiness for usage; they are categorized simply as clean or dirty. For object-state localization, we generated two distinct expressions for each object based on the ambiguous opposites and synonyms listed in Table 2. Overall, we obtained 25,401 expressions and 76,203 dialog instances.

3.2.3 Evaluation Metric

Our main challenges lie in accurately localizing and predicting bounding boxes for object states and grasp regions. Following prior studies [9, 14, 16, 18], we adopt the standard Intersection over Union (IoU) [48] as our evaluation metric. The standard IoU is a widely used measure in computer vision for tasks such as object detection and segmentation. Let $B_p \subset \mathbb{R}^2$ denote the predicted bounding box and $B_g \subset \mathbb{R}^2$ the

corresponding ground-truth box. The intersection and union are defined as

$$I = B_p \cap B_g, \qquad U = B_p \cup B_g.$$

The standard IoU is

$$\mathrm{IoU}(B_p, B_g) = \frac{|I|}{|U|},$$

where $|\cdot|$ denotes the area measure. Standard IoU is bounded in [0, 1].

4 Proposed Method: StateVLM

4.1 Overall Structure

We propose StateVLM, a state-aware vision-language model that uses MiniCPM-V [49] as its backbone. MiniCPM-V is designed for edge devices and offers strong commonsense reasoning capabilities, which are essential for robotic tasks. However, it lacks inherent object detection functionality. These characteristics make it an ideal testbed for isolating and validating our hypothesis regarding object, object-state localization and affordance reasoning. We propose a less intrusive strategy than Pix2Emb, NExT-Chat [18], to fine-tune StateVLM, as shown in Fig. 1.

Specifically, during the training phase, StateVLM jointly optimizes the language generation and auxiliary bounding box prediction objectives. Given an image (I) and a text prompt (T), StateVLM generates a sequence of tokens Y that includes both text and numerical tokens, together with the auxiliary predicted bounding box ($\hat{B}$):

$$(Y, \hat{B}) = \texttt{StateVLM}(I, T).$$

During the inference phase, StateVLM follows the standard Pix2Seq-style autoregressive decoding process and generates a sequence of tokens (Y):

$$(Y) = \texttt{StateVLM}(I, T).$$

4.2 StateVLM Modules

4.2.1 BackBone

The backbone network is adopted from MiniCPM-V 2.6 [49], which comprises three key modules: a visual decoder, a shared compression layer, and an LLM. The visual decoder utilizes an adaptive visual encoding approach, SigLip-400M [50], to tokenize the image inputs into visual tokens. The shared compression layer uses a perceiver-resampler structure [11] with a single cross-attention layer to compress the large number of visual tokens into a fixed set of 96 tokens. Finally, the compressed visual tokens and the text input are fed into the LLM (Qwen2-7B [51]). Therefore the backbone encoder can be formulated as a function that takes the image (I) and text input (T) and produces hidden representations (H):

$$H = \texttt{Qwen}(\texttt{PerceiverResampler}(\texttt{SigLIP}(I)), T)$$

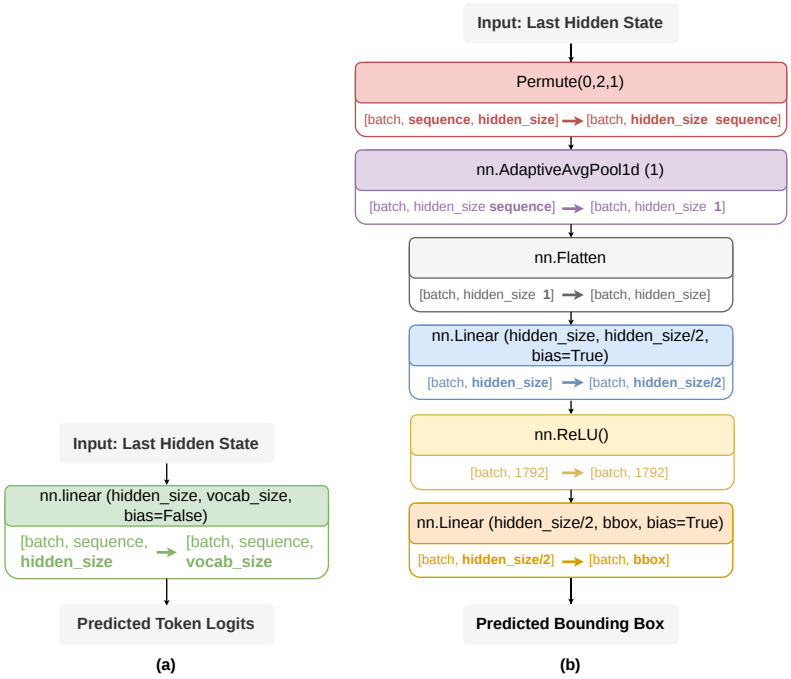


Fig. 5: (a) Language modeling head; (b) Auxiliary regression head. The language modeling head corresponds to the default autoregressive training objective used in Qwen2-7B and is utilized for sequence generation in StateVLM. The auxiliary regression head represents the proposed auxiliary training objective for StateVLM.

4.2.2 Language Modeling Head

Conditioned on the hidden representations (H), the language modeling head generates a sequence of tokens ($Y = Y_1, Y_2, \ldots, Y_T$):

$$P(Y \mid I, T) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, H).$$

The language modeling head is the default autoregressive training objective for Qwen2-7B [51], which consists of a specific linear layer (see Fig. 5a). This language modeling head works jointly with the backbone to train the model using a causal self-attention mask to predict the next token in a sequence based on all previous tokens.

4.2.3 Auxiliary Regression Head

In parallel, we propose an auxiliary regression head to predict the bounding box coordinates ($\hat{B}$) from the hidden representations (H):

$$\hat{B} = f_{\text{ARL}}(H).$$

The auxiliary regression head is a lightweight feed-forward linear head that transforms the sequence output into continuous values. As illustrated in Fig. 5b, an `AdaptiveAvgPool1d` layer is used to perform global average pooling over the temporal (sequence) dimension, reducing each hidden-state sequence to a single vector. The resulting vector is flattened into a one-dimensional vector using a `Flatten` layer. Next, a fully connected layer projects this vector to a lower-dimensional intermediate representation. `ReLU` non-linearity is applied to introduce non-linear transformations, enabling the model to capture complex relationships in the data. Finally, another fully connected layer maps the intermediate representation to a 4-dimensional output corresponding to the four continuous numbers, which are the predicted location coordinates $[x_1, y_1, x_2, y_2]$.

4.3 Joint Training Objective

4.3.1 Overall Training Objective

The overall training objective of StateVLM is a weighted combination of the *causal language modeling (CLM)* loss and *auxiliary regression loss (ARL)*:

$$\mathcal{L}_{\text{CLM+ARL}} = \alpha\mathcal{L}_{\text{CLM}} + \beta\mathcal{L}_{\text{ARL}},$$

$\alpha = 0.2$ and $\beta = 0.8$ are the weights of $\mathcal{L}_{\text{CLM}}$ and $\mathcal{L}_{\text{ARL}}$, which balance these two losses and ensure they play equivalent feedback roles in model tuning. For the value selection, we tried different combinations of α and β and found that this combination yields the best performance on the validation set.

4.3.2 Causal Language Modeling

The backbone, Qwen2-7B, is trained autoregressively using the CLM objective, predicting each token conditioned on all previous tokens via a causal self-attention mask. Given a token sequence $[Y_1, \ldots, Y_T]$ and vocabulary size V, with model logits $z_t \in \mathbb{R}^V$ at timestep t, the *CLM loss* is

$$\mathcal{L}_{\text{CLM}} = -\sum_{t=1}^{T-1} \log \frac{\exp(z_{t,Y_{t+1}})}{\sum_{v=1}^{V} \exp(z_{t,v})},$$

where $z_{t,v}$ is the predicted logit for the token v and Y_{t+1} is the corresponding ground-truth token.

4.3.3 Auxiliary Regression Loss

To enhance numerical reasoning and improve bounding box prediction, we incorporate an *ARL*. This loss jointly supervises both the location and shape of predicted

bounding boxes by combining *Least Absolute Deviations (L1) loss* [52] and *Generalized Intersection Over Union (GIoU) loss* [48].

Let $B_p \subset \mathbb{R}^2$ denote the predicted bounding box and $B_g \subset \mathbb{R}^2$ the corresponding ground-truth box. The *L1 loss* is computed as

$$\mathcal{L}_{L1} = \|B_p - B_g\|_1.$$

The *GIoU loss* is based on the standard Intersection over Union (IoU) metric, which measures the overlap between two bounding boxes, as we discussed in Section 3.2.3. Because IoU does not capture spatial discrepancies when the boxes do not overlap, GIoU introduces an additional penalty term that accounts for the normalized area outside the union of the two bounding boxes but within their smallest enclosing box. Let C denote the smallest enclosing box that contains both B_p and B_g. The GIoU is defined as

$$\text{GIoU}(B_p, B_g) = \text{IoU}(B_p, B_g) - \frac{|C| - |B_p \cup B_g|}{|C|},$$

GIoU loss is calculated as

$$\mathcal{L}_{\text{GIoU}} = 1 - \text{GIoU}(B_p, B_g).$$

The total bounding box loss is a weighted sum of the two components:

$$\mathcal{L}_{\text{ARL}} = \gamma \mathcal{L}_{L1} + \delta \mathcal{L}_{\text{GIoU}},$$

$\gamma = 0.2$ and $\delta = 0.8$ follows the ratio utilized in DETR [53] and NExT-Chat [18].

5 Experiments

5.1 Experimental Overview

We conduct two experiments. Experiment 1 examines our hypothesis that Seq2Pix training paradigms for numerical tasks may lead to inefficient learning. Experiment 2 investigates methods for improving the object-state awareness capabilities of VLMs.

5.2 Implementation Details

We implement StateVLM using PyTorch 2.1.2 and torchvision 0.16.2[2], together with the HuggingFace Transformers library version 4.40.0.[3] The model is initialized with weights from the pretrained MiniCPM-V 2.6[4]. We perform full fine-tuning of the entire model in the first experiment on large-scale public benchmarks. For the second experiment, we modify the model for parameter-efficient fine-tuning using LoRA [54] on our proposed small-scale dataset. All experiments are conducted on four NVIDIA A100 GPUs (80GB each). We employ DeepSpeed 0.14.5[5] to optimize distributed training and use an automatically warmed-up learning rate of 1e-6 to avoid unstable parameter updates during the early stages of optimization for both training regimes. Due

[2] https://pytorch.org/
[3] https://huggingface.co/docs/transformers/index
[4] https://github.com/QAdottech/MiniCPM-V
[5] https://www.deepspeed.ai/

to GPU memory constraints, the batch size is set to 24 for full fine-tuning and 48 for LoRA fine-tuning on four NVIDIA A100 GPUs, where LoRA enables larger batches through reduced memory overhead. Following prior state-of-the-art approaches [8–16], we evaluate predicted bounding boxes using the IoU metric (see Section 3.2.3) with a threshold of 0.5, reported as Acc@0.5.

5.3 Experiment 1: Full Fine-Tuning on Adapted Public Benchmarks

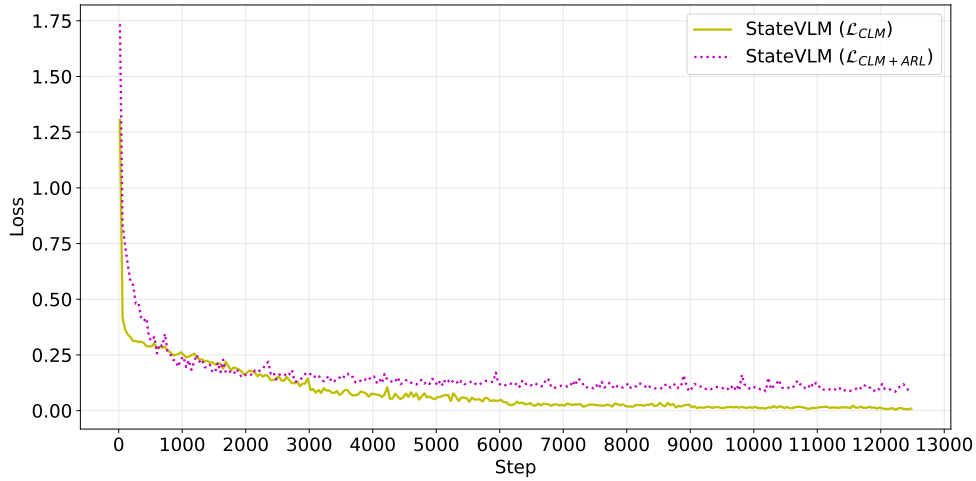
To verify our hypothesis that the current training paradigm for numerical tasks may lead to inefficient learning, we conduct comparative experiments on the adapted REC benchmarks: RefCOCO, RefCOCO+ [6], and RefCOCOg [7]. RefCOCO is split into (107859, 10834, 5657, 5095) samples for the (train, validation, testA, testB) sets, respectively. RefCOCO+ contains (107376, 10758, 5726, 4889) samples for the corresponding splits, while RefCOCOg consists of (72369, 4896, 9602) samples for the (train, validation, test) splits, respectively. We fine-tune our models on the training split and report performance on the corresponding validation and test sets (i.e., testA and testB for RefCOCO and RefCOCO+, and the test set for RefCOCOg).

We first assess the performance of the backbone model, MiniCPM-V, as a baseline. Then, we conduct two groups of full fine-tuning experiments: one in which the model is trained solely with the standard *CLM* objective and another in which it is trained using the *CLM* combined with the *ARL* objective described in Section 4.3. For the distinction, we denote them as StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$), respectively.

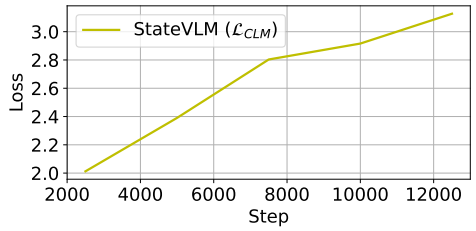
5.3.1 Results and Analysis

We observe that the training loss varies with the number of steps for each model. StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) show a similar trend, where the losses gradually decrease as the number of training steps increases, as shown in Fig. 6a. The validation loss for StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) aligns with the training loss trend (see Fig. 6c). However, the validation loss for StateVLM ($\mathcal{L}_{\text{CLM}}$) continues to increase with the number of training steps (see Fig. 6b). StateVLM ($\mathcal{L}_{\text{CLM}}$) with 5000 training steps outperforms both the 2500-step and 7500-step models. We suspect that StateVLM ($\mathcal{L}_{\text{CLM}}$) begins to overfit after 5000 steps of training on the current training data.

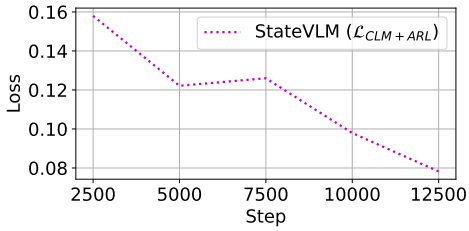
Furthermore, we evaluate the fine-tuned models StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) on the validation and test splits of RefCOCO, RefCOCO+, and RefCOCOg. We used a fixed random seed for evaluation to ensure deterministic model outputs. Consequently, all runs yield identical results, and no deviation metrics are applicable. We demonstrate the average performance of the two models at 5000, 10000, and 15000 training steps on the RefCOCO, RefCOCO+, and RefCOCOg benchmarks in Fig. 7. StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) performance improves significantly from 5000 to 15000 steps and is better than StateVLM ($\mathcal{L}_{\text{CLM}}$) after 10000 steps. StateVLM ($\mathcal{L}_{\text{CLM}}$) performance, however, remains stable or even decreases over the training, which is consistent with the validation loss curve in Fig. 6b.



(a)



(b)



(c)

Fig. 6: StateVLM loss curves: (a) StateVLM ($\mathcal{L}_{\text{CLM}}$ and $\mathcal{L}_{\text{CLM+ARL}}$) training loss progression over steps, (b) Validation loss for StateVLM ($\mathcal{L}_{\text{CLM}}$), and (c) Validation loss for StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$).

The concrete performance of StateVLM ($\mathcal{L}_{\text{CLM}}$) on the subset of RefCOCO, RefCOCO+, and RefCOCOg splits is illustrated in Fig. 8. The performance of StateVLM ($\mathcal{L}_{\text{CLM}}$) remains stable between 5000 and 10000 steps and decreases at 15000 steps, indicating that it reaches its maximum performance at 5000 steps. We continue training the StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) until 25000 steps and the performance gradually improves over the training steps on RefCOCO, RefCOCO+, and RefCOCOg, as shown in Fig. 9.

The performance of StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) is illustrated in Table 3, alongside existing models on the adapted REC task. StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) outperforms StateVLM ($\mathcal{L}_{\text{CLM}}$), demonstrating the effectiveness of the auxiliary regression loss in improving the model's performance on bounding box prediction tasks. Compared with the Pix2Emb-based method NExT-Chat, StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) achieves superior performance. Comparing the Pix2Seq-based method

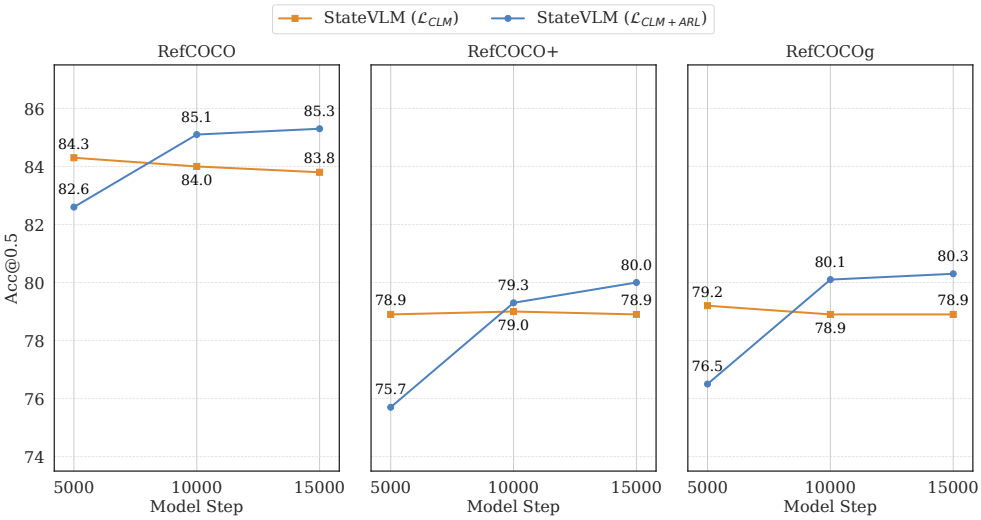


Fig. 7: Average performance of StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) across benchmarks over training Steps.

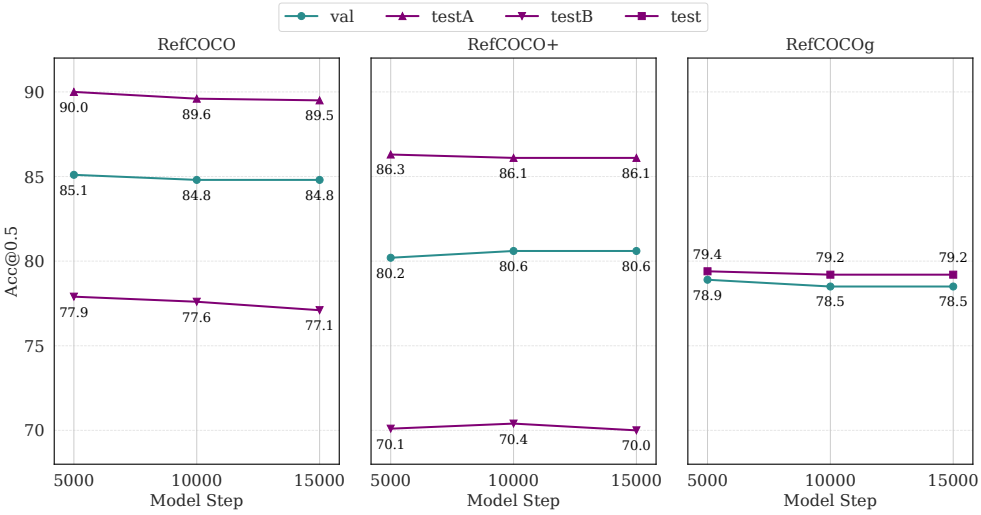


Fig. 8: StateVLM ($\mathcal{L}_{\text{CLM}}$) performance changes over training steps.

StateVLM ($\mathcal{L}_{\text{CLM}}$) with Shikra, StateVLM ($\mathcal{L}_{\text{CLM}}$) achieves lower performance than Shikra. As the training regime, computing resources, and training time vary, we cannot directly infer why StateVLM ($\mathcal{L}_{\text{CLM}}$) underperforms Shikra. However, StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) achieves performance comparable to Shikra, demonstrating the effectiveness of the auxiliary regression loss. On average, the StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) achieves a 1.6% performance improvement over the StateVLM ($\mathcal{L}_{\text{CLM}}$). Overall, these results

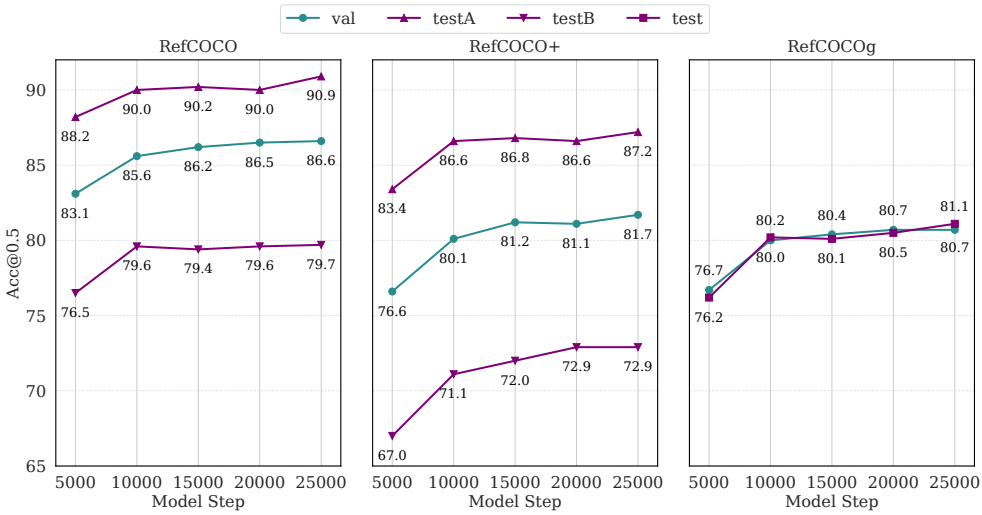


Fig. 9: StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) performance changes over training steps.

Table 3: Comparative performance comparison in adapted REC benchmarks (%). The evaluation metric is Acc@0.5.

VLMs Type	Methods	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val	test
Pix2Emb	NExT-Chat [18]	85.5	90.0	77.9	77.2	84.5	68.0	80.1	79.8
Pix2Seq	Shikra [9]	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2
	Baseline (MiniCPM-V)	16.2	19.6	13.4	12.4	15.7	9.9	7.2	6.7
	StateVLM ($\mathcal{L}_{\text{CLM}}$)	85.1	90.0	77.9	80.2	86.3	70.1	78.9	79.4
	StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$)	86.6	90.9	79.7	81.7	87.2	72.9	80.7	81.1
	Improvements (Avg 1.6)	+1.5	+0.9	+1.8	+1.5	+0.9	+2.8	+1.8	+1.7

demonstrate that incorporating an auxiliary regression loss can significantly enhance the performance of VLMs on bounding box prediction tasks, validating our hypothesis about the limitations of the current training paradigm for object localization tasks.

Ablation Study

We conducted several ablation studies to identify the components essential for achieving the final performance.

Auxiliary Regression Head Design: One way is to reduce the two linear layers to a single linear layer. Additionally, we tried different activation functions within this auxiliary regression head, testing `GELU` and `Sigmoid`.

Text Prompts: We tested two different text prompts to understand the effect of text content on model performance:

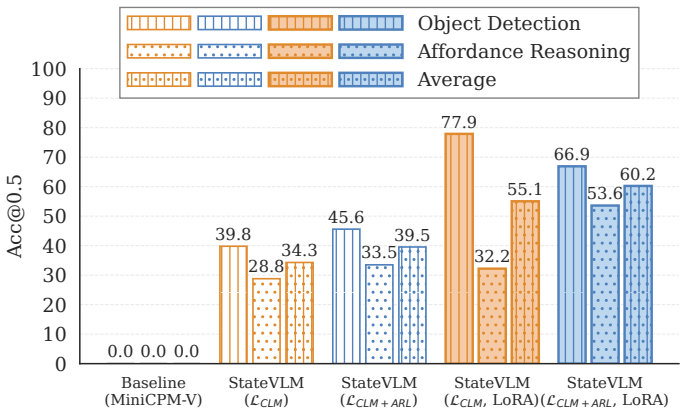


Fig. 10: Comprehensive performance comparison on OSAR.

- Simple Prompt: "A chat between a human and an AI that understands visuals. Follow instructions."
- Concrete Prompt: "A chat between a human and an AI that understands visuals. In images, [x, y] denotes points: top-left is [0, 0], bottom-right is [1, 1]. Increasing x moves right; y moves down. A bounding box is defined as $[x_1, y_1, x_2, y_2]$ where (x_1, y_1) is the top-left corner and (x_2, y_2) is the bottom-right corner. Follow instructions."

Overall, the combination of the regression head architecture that we presented, along with the simple prompt, achieved the best performance.

5.4 Experiment 2: LoRA Fine-Tuning on Proprietary Dataset

To investigate the potential object-state awareness of VLMs, we introduce a small-scale and well-annotated dataset, OSAR, as detailed in Section 3. We first evaluate the performance of the baseline model, MiniCPM-V, and the models that we obtained from previous experiments: StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) on our proposed dataset. Due to the cost of dataset generation and annotation, OSAR is relatively small-scale, which is not sufficient for full fine-tuning. Therefore, we further fine-tune these models using the LoRA strategy, obtaining the models StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA).

5.4.1 Results and Analysis

A systematic experimental evaluation on the proposed tasks, object detection and affordance reasoning, is shown in Fig. 10. The baseline model, MiniCPM-V 2.6, fails to perform these tasks, as it is not trained for object detection. StateVLM ($\mathcal{L}_{\text{CLM}}$) and ($\mathcal{L}_{\text{CLM+ARL}}$), obtained from the previous experiment, achieve scores of 34.3% and 39.5% on OSAR, respectively. StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) achieves better performance than StateVLM ($\mathcal{L}_{\text{CLM}}$), demonstrating the effectiveness of the auxiliary regression

Table 4: Exception rates(%). N/A indicates Not Applicable because LoRA fine-tuning was evaluated only on OSAR, not on the adapted REC datasets.

Model	Adapted REC Datasets			Proprietary Dataset	
	RefCOCO	RefCOCO+	RefCOCOg	Object Detection	Affordance Reasoning
StateVLM ($\mathcal{L}_{\text{CLM}}$)	0.21	0.15	0.17	0.10	0.06
StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$)	0.00	0.00	0.00	0.00	0.00
StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA)	N/A	N/A	N/A	3.45	33.36
StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA)	N/A	N/A	N/A	0.00	0.00

loss in improving the model's performance on numerical prediction tasks. This finding is consistent with the conclusion of the first experiment.

After applying the same-pattern LoRA fine-tuning strategy on OSAR, the performance of StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA) improve to 55.1% and 60.2%, respectively. StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA) performs better than StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA). This result further demonstrates that the auxiliary regression loss is effective in improving the model's performance on numerical prediction tasks, which is consistent with the conclusion of the first experiment.

Notably, the performance of StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA) on the object detection task is lower than that of StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA), which is inconsistent with the conclusion of the first experiment. However, StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA) performs better than StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA) on the affordance reasoning task, which is consistent with the conclusion of the first experiment.

The affordance reasoning task is more complex and challenging than the object detection task. The object detection task only requires the model to predict the bounding box of the object, whereas the affordance reasoning task requires the model to predict the bounding box of the grasping area, which is sometimes only a part of the object. In previous full fine-tuning benchmarks (RefCOCO, RefCOCO+, and RefCOCOg), the bounding boxes generally enclose the whole object rather than just a part of it. Therefore, StateVLM ($\mathcal{L}_{\text{CLM}}$, LoRA) quickly learns to predict the bounding box of the whole object, improving from 39.8% to 77.9% after LoRA fine-tuning. However, it is limited to learn to predict the bounding box of the grasping area, improving only from 28.8% to 32.2% after LoRA fine-tuning. It also exhibits a high exception rate of 33.36% on the affordance reasoning task, indicating that the model does not fully understand the complexity of the task and to predict bounding boxes in a consistent format.

In contrast, StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$, LoRA) is able to learn to predict both the bounding box of the whole object and the grasping area simultaneously. It achieves a zero exception rate on both tasks, indicating that the model learns to understand the tasks and to produce bounding boxes in a consistent format. The exception rates of all models on the subtasks are shown in Table 4. Models trained with the auxiliary regression loss achieve zero exception rates, further demonstrating its effectiveness in improving model performance and output consistency on numerical tasks. We conclude that the auxiliary regression loss enhances the model's ability to predict bounding boxes in a consistent, predefined format, which is particularly beneficial for the affordance reasoning task.

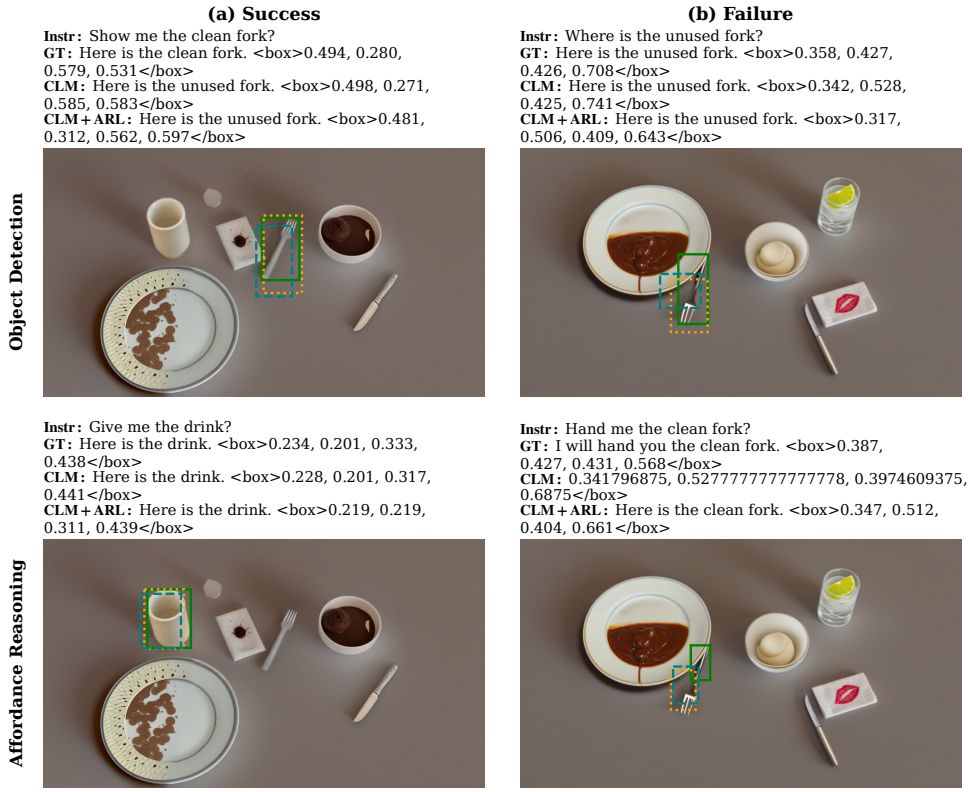


Fig. 11: Qualitative comparison of StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) on grounded object detection and affordance reasoning examples. Ground-truth boxes are shown in green, while model predictions are shown in orange (StateVLM ($\mathcal{L}_{\text{CLM}}$)) and teal (StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$)).

5.4.2 Visualization

To provide a more direct and intuitive observation of the performance of StateVLM ($\mathcal{L}_{\text{CLM}}$) and StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$), we visualize representative successful and failed instances of predicted bounding boxes from these two models on object detection and affordance reasoning tasks, as shown in Fig. 11. For object detection, the goal is to predict bounding boxes covering entire objects. Both models generally succeed on this task but exhibit reduced accuracy on cutlery (e.g., knives, forks, and spoons). The predicted boxes for these cases are often biased toward one side of the object. This suggests a potential interference from affordance reasoning, as cutlery also represents the most challenging category for that task. For affordance reasoning, the objective is to localize grasping regions. StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$) demonstrates improved understanding of affordances for liquids and semi-solid foods (e.g., sauces and drinks), accurately predicting grasping regions for their containers. In contrast, StateVLM ($\mathcal{L}_{\text{CLM}}$) frequently produces invalid prediction formats or does not capture the intended task.

Both models struggle with plates and cutlery, failing to correctly localize their grasping regions. Overall, incorporating the auxiliary regression loss improves affordance reasoning while a significant gap remains between current performance and practical deployment, motivating further research.

6 Conclusion and Future Work

In this paper, we propose StateVLM, a novel model designed to perceive and learn fine-grained object representations, including precise localization of objects and their states, as well as graspable regions. We propose a less intrusive strategy than Pix2Emb for adapting StateVLM to numerical tasks under limited computational resources. Specifically, during the training phase, StateVLM utilizes the output of the auxiliary regression head to compute an auxiliary regression loss (ARL) for detection training. During the inference phase, StateVLM continues with standard sequence prediction. Comparative results on adapted REC tasks (RefCOCO, RefCOCO+, RefCOCOg) demonstrate that this ARL improves the performance of StateVLM on REC tasks compared to training without it, thereby validating our hypothesis about the limitations of the current training paradigm of Pix2Seq for numerical tasks. StateVLM's performance on the proposed OSAR benchmark indicates that its latent state-aware capabilities for object-state localization and affordance reasoning can be further enhanced through LoRA tuning on a small-scale and well-annotated dataset. Additionally, we find that this ARL also enhances the consistency of StateVLM outputs, which is particularly important for complex tasks such as affordance reasoning.

There remain challenges to our proposed approach for real-world embodied agents. First, affordance is a broad and multifaceted concept, and we hypothesize that more extensive and carefully annotated datasets are required to advance this field. Although we leveraged a diffusion model to generate the dataset, human verification is still necessary. Second, our evaluation focused on distinguishing objects in different states, and we did not investigate ambiguous or vague instructions in depth. Addressing these challenges will require the development of novel approaches that extend beyond efficient fine-tuning strategies. However, we demonstrate that incorporating an ARL significantly improves StateVLM performance in object detection, and that a small-scale, well-annotated dataset further enhances StateVLM's state-awareness, thereby improving object-state localization and affordance reasoning. With the publication of this research, we will release the code and dataset to facilitate further research in this area.

Acknowledgements. The authors gratefully acknowledge support from the China Scholarship Council (CSC) and the German Research Foundation DFG under project CML (TRR 169). The authors also thank Tianyu Liu for his advice on the experimental design and Jae Hee Lee for his valuable feedback.

Declarations

- **Conflict of Interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- **Author Contributions** Xiaowen Sun conceived the study, designed and implemented the method, curated the dataset, conducted the analysis, and drafted the manuscript. Matthias Kerzel generated dataset images and contributed to discussions. Mengdi Li contributed to refining the concept and conducting the analysis. Xufeng Zhao contributed to the analysis. Paul Striker assisted Xiaowen Sun with dataset annotation. Stefan Wermter supervised Xiaowen Sun and provided overall guidance. All authors reviewed and edited the manuscript.
- **Ethics Approval** Not applicable.
- **Data Availability** Available upon request.
- **Code Availability** Available upon request.

References

[1] Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., Dai, J.: VisionLLM: large language model is also an open-ended decoder for vision-centric tasks. In: Proceedings of the 37th International Conference on Neural Information Processing Systems, pp. 61501–61513. Curran Associates Inc., Red Hook, NY, USA (2023)

[2] Guo, Q., De Mello, S., Yin, H., Byeon, W., Cheung, K.C., Yu, Y., Luo, P., Liu, S.: RegionGPT: Towards region understanding vision language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13796–13806 (2024)

[3] Yang, C., Guan, Z., Wang, X., Ma, W.: EFNet: enhanced activation and fine-grained information auxiliary network for salient object detection. Applied Intelligence **56**(4), 96 (2026)

[4] Jia, Z., Wang, S., Zheng, J., Han, X., Tang, Y., Sun, F.: AFFNet: Adaptive feature fusion network for defect detection of industrial product surface. Applied Intelligence **56**(2), 59 (2026)

[5] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2425–2433 (2015). IEEE

[6] Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 787–798 (2014)

[7] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11–20 (2016)

[8] Lin, Z., Liu, D., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Shao, W., Chen, K., Han, J., Huang, S., Zhang, Y., He, X., Qiao, Y., Li, H.: SPHINX: A mixer of weights, visual embeddings and image scales for multimodal large language models. In: European Conference on Computer Vision (ECCV), LXII, pp. 36–55. Springer, Berlin, Heidelberg (2024)

[9] Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal LLM's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)

[10] Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)

[11] Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 (2023)

[12] Zhang, H., Li, H., Li, F., Ren, T., Zou, X., Liu, S., Huang, S., Gao, J., Leizhang, Li, C., Yang, J.: LLaVA-grounding: Grounded visual chat with large multimodal models. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) European Conference on Computer Vision, Part XLIII, pp. 19–35. Springer, Milan, Italy (2024)

[13] Pramanick, S., Han, G., Hou, R., Nag, S., Lim, S.-N., Ballas, N., Wang, Q., Chellappa, R., Almahairi, A.: Jack of all tasks, master of many: Designing general-purpose coarse-to-fine vision-language model. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14076–14088 (2024)

[14] You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.-F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. In: The Twelfth International Conference on Learning Representations (2024)

[15] Zhang, H., You, H., Dufter, P., Zhang, B., Chen, C., Chen, H.-Y., Fu, T.-J., Wang, W.Y., Chang, S.-F., Gan, Z., Yang, Y.: Ferret-v2: An improved baseline for referring and grounding with large language models. In: First Conference on Language Modeling (2024)

[16] Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., *et al.*: CogVLM: Visual expert for pretrained language models. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C. (eds.) Advances

in Neural Information Processing Systems, vol. 37, pp. 121475–121499. Curran Associates, Inc., Red Hook, NY, USA (2024)

[17] Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G.: Pix2seq: A language modeling framework for object detection. In: International Conference on Learning Representations (2022)

[18] Zhang, A., Yao, Y., Ji, W., Liu, Z., Chua, T.-S.: NExT-Chat: an LMM for chat, detection, and segmentation. In: Proceedings of the 41st International Conference on Machine Learning, pp. 60116–60133. JMLR.org, Vienna, Austria (2024)

[19] Gouidis, F., Papoutsakis, K.E., Patkos, T., Argyros, A.A., Plexousakis, D.: Exploring the impact of knowledge graphs on zero-shot visual object state classification. In: Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), vol. 2, pp. 738–749 (2024)

[20] Jelodar, A.B., Sun, Y.: Joint object and state recognition using language knowledge. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 3352–3356 (2019)

[21] Spisak, J., Kerzel, M., Wermter, S.: Clarifying the half full or half empty question: Multimodal container classification. In: International Conference on Artificial Neural Networks, pp. 444–456 (2023). Springer

[22] Gouidis, F., Patkos, T., Argyros, A., Plexousakis, D.: Detecting object states vs detecting objects: A new dataset and a quantitative experimental study. In: Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), vol. 5, pp. 590–600 (2022)

[23] Li, X., Huang, S., Yu, Q., Jiang, Z., Hao, C., Zhu, Y., Li, H., Gao, P., Lu, C.: SKT: integrating state-aware keypoint trajectories with vision-language models for robotic garment manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2025, pp. 18828–18833. IEEE, Hangzhou, China (2025)

[24] Nguyen, N., Bi, J., Vosoughi, A., Tian, Y., Fazli, P., Xu, C.: OSCaR: Object state captioning and state change representation. In: Duh, K., Gomez, H., Bethard, S. (eds.) Findings of the Association for Computational Linguistics: NAACL 2024, pp. 3565–3576. Association for Computational Linguistics, Mexico (2024)

[25] Liu, Z., Freeman, W.T., Tenenbaum, J.B., Wu, J.: Physical primitive decomposition. In: Proceedings of the European Conference on Computer Vision (ECCV), Part XII, pp. 3–20 (2018)

[26] Gibson, J.J.: The theory of affordances. In: Shaw, R., Bransford, J. (eds.) Perceiving, Acting, and Knowing: Toward an Ecological Psychology, pp. 67–82. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey (1977)

[27] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., *et al.*: Ego4D: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18995–19012 (2022)

[28] Yang, Y., Yu, H., Lou, X., Liu, Y., Choi, C.: Attribute-based robotic grasping with data-efficient adaptation. IEEE Transactions on Robotics **40**, 1566–1579 (2024)

[29] Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: International Conference on Machine Learning, pp. 23318–23340 (2022). PMLR

[30] Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: TransVG: End-to-end visual grounding with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1769–1779 (2021)

[31] Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: UNITER: Universal image-text representation learning. In: European Conference on Computer Vision, pp. 104–120 (2020). Springer

[32] Gan, Z., Chen, Y.-C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. Advances in Neural Information Processing Systems **33**, 6616–6628 (2020)

[33] Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., Wang, L.: UniTAB: Unifying text and box outputs for grounded vision-language modeling. In: European Conference on Computer Vision, pp. 521–539 (2022). Springer

[34] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., *et al.*: Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. In: European Conference on Computer Vision, pp. 38–55 (2024). Springer

[35] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering vol. 236. Springer, Berlin and Heidelberg, Germany (2012)

[36] Norman, D.A.: The Psychology of Everyday Things. Basic books, New York (1988)

[37] Yang, Y., Lou, X., Choi, C.: Interactive robotic grasping with attribute-guided disambiguation. In: 2022 International Conference on Robotics and Automation

(ICRA), pp. 8914–8920 (2022). IEEE

[38] Yu, S., Lin, K., Xiao, A., Duan, J., Soh, H.: Octopi: Object property reasoning with large tactile-language models. In: Robotics: Science and Systems, Delft, Netherlands (2024)

[39] Huang, S., Chang, H., Liu, Y., Zhu, Y., Dong, H., Boularias, A., Gao, P., Li, H.: A3VLM: Actionable articulation-aware vision language model. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning, vol. 270, pp. 1675–1690. PMLR, Munich, Germany (2025)

[40] Li, X., Zhang, M., Geng, Y., Geng, H., Long, Y., Shen, Y., Zhang, R., Liu, J., Dong, H.: ManipLLM: Embodied multimodal large language model for object-centric robotic manipulation. In: Computer Vision and Pattern Recognition, pp. 18061–18070 (2024)

[41] Huang, S., Ponomarenko, I., Jiang, Z., Li, X., Hu, X., Gao, P., Li, H., Dong, H.: ManipVQA: Injecting robotic affordance and physically grounded information into multimodal large language models. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7580–7587 (2024). IEEE

[42] Guo, D., Xiang, Y., Zhao, S., Zhu, X., Tomizuka, M., Ding, M., Zhan, W.: PhyGrasp: Generalizing robotic grasping with physics-informed large multimodal models. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 14915–14922 (2025)

[43] Booch, G., Maksimchuk, R.A., Engle, M.W., Young, B.J., Connallen, J., Houston, K.A.: Object-oriented analysis and design with applications. ACM SIGSOFT Software Engineering Notes **33**(5), 29–29 (2008)

[44] Manousaki, V., Bacharidis, K., Gouidis, F., Papoutsakis, K., Plexousakis, D., Argyros, A.: Anticipating object state changes. arXiv preprint arXiv:2405.12789 (2024)

[45] Sun, X., Zhao, X., Lee, J.H., Lu, W., Kerzel, M., Wermter, S.: Details make a difference: Object state-sensitive neurorobotic task planning. In: International Conference on Artificial Neural Networks, pp. 261–275 (2024). Springer

[46] Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2641–2649 (2015)

[47] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10674–10685. IEEE

Computer Society, Los Alamitos, CA, USA (2022)

[48] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)

[49] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Chen, C., Li, H., Zhao, W., *et al.*: Efficient gpt-4v level multimodal large language model for deployment on edge devices. Nature Communications **16**(1), 5509 (2025)

[50] Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11975–11986 (2023)

[51] Team, Q., et al.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)

[52] Brooks, J.P., Dulá, J.H.: The L1-norm best-fit hyperplane problem. Applied Mathematics Letters **26**(1), 51–55 (2013)

[53] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229 (2020). Springer

[54] Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)

[55] Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: MAttNet: Modular attention network for referring expression comprehension. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1307–1315 (2018)

Appendix A VLMs Configurations and Performance

We summarize the configurations of VLMs capable of performing the REC task in Table A1, including their architectures, names, sizes, training resources, and training stages. Table A2 further presents the performance of these VLMs on adapted REC benchmarks, including RefCOCO, RefCOCO+, and RefCOCOg. Due to differences in the amount of training resources, training strategies, and data diversity, the performance of these VLMs varies significantly across benchmarks. Our goal is not to achieve state-of-the-art performance on these benchmarks, but rather to verify the effectiveness of the auxiliary regression loss in improving VLM performance on numerical tasks, particularly object localization.

Table A1: Summary of VLM configurations capable of performing the REC task (%). * refers to different resources to be used in the second stage. × states that this stage is not included. Pix2Seq and Pix2Emb stand for pixel-to-sequence and pixel-to-embedding.

VLMs Architecture	VLMs Name	VLMs Size	Training GPUs	Training Stage 1	Training Stage 2
Pix2Seq	Shikra [9]	~13B	8×A100	~100 hours	~20 hours
	Ferret [14]		8×A100	~125 hours	×
	Ferret (v2) [15]		8×A100	N/A	N/A
	SPHINX-1K [8]		32×A100	~125 hours	~38 hours*
	SPHINX-2K [8]		32×A100	~250 hours	~38 hours*
	VistaLLM [13]		32×A100	~72 hours	~30 hours
	CogVLM-G [16]		256×A100	~120000 steps	~60000 steps
	Shikra [9]	~7B	8×A100	~100 hours	~20 hours
	MiniGPT (v2) [10]		8×A100	~90 hours	~20 hours*
	Qwen-VL [11]		N/A	~50000 steps	~19000 steps
	LLaVA-G [12]		N/A	~10000 steps	~8000 steps
	VistaLLM [13]		32×A100	~48 hours	~22 hours
	Ferret [14]		8×A100	~60 hours	×
	Ferret (v2) [15]		8×A100	N/A	N/A
Pix2Emb	NExT-Chat [18]	~7B	8×A100	~59 hours	~10 hours

Table A2: StateVLM: Comparative performance comparison in REC (%). The evaluation metric is Acc@0.5. * refers to the specialist or fine-tuned methods.

VLMs Type	Methods	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val	test
non-VLM	MAttNet* [55]	76.4	80.4	69.3	64.9	70.3	56.0	66.7	67.0
	OFA-L [29]	80.0	83.7	76.4	68.3	76.0	61.8	67.6	67.6
	TransVG* [30]	81.0	82.7	78.4	64.8	70.7	56.9	68.7	67.7
	UNITER* [31]	81.4	87.0	74.2	75.9	81.5	66.7	74.0	68.7
	VILLA* [32]	82.4	87.5	74.8	76.2	81.5	66.8	76.2	76.7
	UniTAB* [33]	86.3	88.8	80.6	78.7	83.2	69.5	80.0	80.0
	G-DINO-L* [34]	**90.6**	**93.2**	**88.2**	**82.8**	**89.0**	**75.9**	**86.1**	**87.0**
Pix2Seq-13B	Shikra [9]	87.8	91.1	81.8	82.9	87.8	74.4	82.6	83.2
	Ferret [14]	89.5	92.4	84.4	82.8	88.1	75.2	85.8	86.3
	Ferret (v2) [15]	92.6	**95.0**	88.9	87.4	92.1	81.4	89.4	90.0
	SPHINX [8]	89.2	91.4	85.1	82.8	87.3	76.9	84.9	83.7
	SPHINX-1K [8]	91.1	92.7	86.7	86.6	91.1	80.4	88.2	88.4
	SPHINX-2k [8]	91.1	92.9	87.1	85.5	90.6	80.5	88.1	88.7
	VistaLLM [13]	89.9	92.5	85.0	84.1	90.3	75.8	86.0	86.4
	CogVLM-G [16]	**92.8**	94.8	**89.0**	**88.7**	**93.0**	**83.4**	**89.8**	**90.8**
Pix2Seq-7B	Shikra [9]	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2
	MiniGPT (v2) [10]	88.1	91.3	84.3	79.6	85.5	73.3	84.2	84.31
	Qwen-VL [11]	88.6	92.3	84.5	82.8	88.6	76.8	86.0	86.3
	LLaVA-G [12]	89.2	—	—	81.7	—	—	84.8	—
	VistaLLM [13]	88.1	91.5	83.0	82.9	89.8	74.8	83.6	84.4
	Ferret [14]	87.5	91.4	82.5	80.9	87.4	73.1	83.9	84.8
	Ferret (v2) [15]	**92.8**	**94.7**	**88.7**	**87.4**	**92.8**	**79.3**	**89.4**	**89.3**
Pix2Emb-7B	NExT-Chat [18]	85.5	90.0	77.9	77.2	84.5	68.0	80.1	79.8
Pix2Seq-7B	Baseline (MiniCPM-V)	16.2	19.6	13.4	12.4	15.7	9.9	7.2	6.7
	StateVLM ($\mathcal{L}_{\text{CLM}}$)	85.1	90.0	77.9	80.2	86.3	70.1	78.9	79.4
	StateVLM ($\mathcal{L}_{\text{CLM+ARL}}$)	86.6	90.9	79.7	81.7	87.2	72.9	80.7	81.1
	Improvements (Avg 1.6)	+1.5	+0.9	+1.8	+1.5	+0.9	+2.8	+1.8	+1.7