LaVA-Man: Learning Visual Action Representations for Robot Manipulation

Chaoran Zhu[1]   Hengyi Wang[2]   Yik Lung Pang[1]   Changjae Oh[1]

[1] Queen Mary University of London   [2] University College London
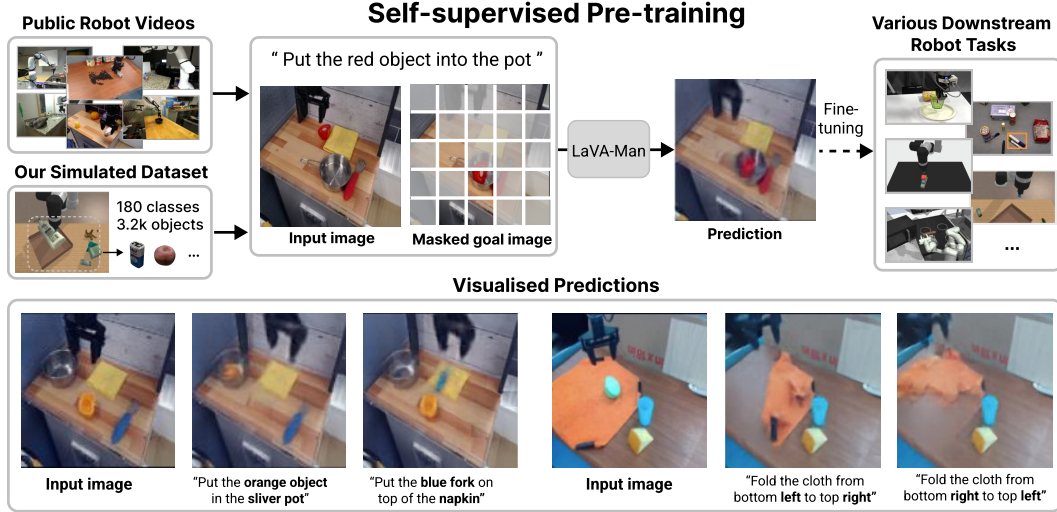
https://qm-ipalab.github.io/LaVA-Man

Figure 1: We introduce **LaVA-Man**, a self-supervised framework for learning visual-action representations for robot manipulation via goal image prediction. We also introduce the Omni-Object Pick-and-Place dataset to ensure the model learns a diverse, open-vocabulary-based object prior. The learned representations can be adapted to various downstream robotic perception and manipulation tasks. The samples of the goal image prediction illustrate that the learned representation can capture the underlying causality of visual state transitions in language-guided manipulation.

Abstract: Visual-textual understanding is essential for language-guided robot manipulation. Recent works leverage pre-trained vision-language models to measure the similarity between encoded visual observations and textual instructions, and then train a model to map this similarity to robot actions. However, this two-step approach limits the model to capture the relationship between visual observations and textual instructions, leading to reduced precision in manipulation tasks. We propose to learn visual-textual associations through a self-supervised pretext task: reconstructing a masked goal image conditioned on an input image and textual instructions. This formulation allows the model to learn visual-action representations without robot action supervision. The learned representations can then be fine-tuned for manipulation tasks with only a few demonstrations. We also introduce the *Omni-Object Pick-and-Place* dataset, which consists of annotated robot tabletop manipulation episodes, including 180 object classes and 3,200 instances with corresponding textual instructions. This dataset enables the model to acquire diverse object priors and allows for a more comprehensive evaluation of its generalisation capability across object instances. Experimental results on the five benchmarks, including both simulated and real-robot validations, demonstrate that our method outperforms prior art.

Keywords: Robot manipulation, self-supervised representation learning.

1 Introduction

Language-guided robot manipulation is a fundamental task in robotics, enabling embodied agents to interpret human instructions and interact with complex environments. This task requires learning a robust representation that effectively associates visual observations with textual instructions, and can be readily mapped to the corresponding robot actions.

The key challenge is to learn such representations in a scalable manner without heavily relying on robot action annotations, such as ground-truth affordance or joint angles. Several works [1, 2, 3, 4] leverage pre-trained vision-language foundation models, such as CLIP [5], which encode images and text into a unified embedding space to serve as the visual-textual representations. These methods compute the cosine similarity between image and text embeddings and learn to map this similarity to robot actions. However, their representations lack causal grounding as they do not capture how the input visual state and textual instructions lead to the resulting visual state after the robot executes an action. Namely, they do not learn true language-guided visual-action representations necessary for manipulation.

To address this limitation, we propose **LaVA-Man**, a self-supervised learning approach for learning **La**nguage-guided **V**isual-**A**ction representations for robot **Man**ipulation. Given textual instructions and visual observations before and after a manipulation, we mask the goal image and train the model to predict its masked content, conditioned on the input image and the language instruction, with minimal guidance from the unmasked regions. Unlike prior approaches that adopt pretext tasks for general-purpose vision (e.g., masked image reconstruction [6, 7]), our goal-image prediction objective captures the underlying causality of manipulation: it enables the model to implicitly learn the association between visual dynamics and action semantics, which is critical for robotic reasoning.

To learn representations that capture diverse and open-vocabulary textual instructions, it is important to train on sequences involving various object instances. However, existing manipulation datasets, such as Ravens [8] and VIMA [9], suffer from limited object diversity. To this end, we introduce the *Omni-Object Pick-and-Place* (*OOPP*) dataset, a simulated tabletop manipulation dataset that consists of 180 object classes and 3,200 unique instances. Built on high-quality real-scanned meshes with language annotations from OmniObject3D [10], OOPP includes curated, scaled objects suitable for manipulation, with scene sequences automatically simulated using PyBullet Gym [11]. For a comprehensive evaluation, we hold out 20 object classes as unseen for inter-class generalisation evaluation, and reserve a subset of instances from 20 seen categories for intra-class evaluation.

We train LaVA-Man on a mixture of synthetic data from the proposed OOPP dataset and real-world robot videos from Bridge [12] and DROID [13]. Once pre-trained, our model can be efficiently fine-tuned with only a few demonstrations for various downstream robotic perception and manipulation tasks. Our contributions are summarised as:

- We propose a self-supervised approach for learning a robust, versatile visual-action representation that can be efficiently fine-tuned on various robotic tasks with a few demonstrations.
- We introduce a new dataset based on the existing pick-and-place benchmarks [1, 8, 9]. Our dataset features 3,200 unique, real-scanned objects from 180 categories.
- We validate our approach on five downstream robotic tasks, including simulated and real-world environments, and establish state-of-the-art performance.

2 Related work

Vision-based robot manipulation. Classic vision-based manipulation methods usually rely on a two-stage pipeline, where vision-based perception [14, 15, 16, 17] is utilised first, followed by a control algorithm [18]. Recent methods have shifted towards end-to-end learning frameworks [19, 20, 21, 22]. However, these methods often rely on limited training labels and demonstrate limited generalisation ability. CLIPort [1] and its variants [2, 4] show enhanced generalisation to open-set problems by leveraging the semantic understanding provided by CLIP [5]. However, their

CLIP-based representations simply provide visual-textual similarity without causal grounding and limit their capabilities in capturing the underlying causality of the language-guided manipulation. Besides CLIP, other recent methods [23, 24, 25] utilise large-scale vision-language models (VLMs) to achieve generalisation. But the reliance on large-scale models may introduce practical limitations.

Self-supervised visual pre-training. Self-supervised visual pre-training has become a fundamental approach for learning generalisable visual representations from large-scale, unlabeled data through various pretext tasks such as solving jigsaw puzzles [26], image colourisation [27], rotation estimation [28], inpainting [29], and instance discrimination [30, 31]. Inspired by masked token prediction in BERT [32], Masked-autoencoders (MAE) [33] mask portions of input images and train the model to reconstruct the missing patches. This method shows great success in learning generalisable representations and leads to various follow-up works on learning temporal correspondence [34, 35] and spatial information [36]. We propose goal-image prediction as a pretext task that implicitly learns the causality in language-guided manipulation, enabling our model to learn visual-action representations that associate visual states with robot actions conditioned on instructions.

Visual pre-training for robotics. Visual pre-training can be applied to robotic manipulation to enhance the generalisation ability [7, 37]. They rely on various pre-training methods, such as contrastive learning [38, 39], MAE [7, 40, 41], and other perception tasks [42, 43]. For instance, the approach in [37] pre-trains a visual backbone using classic tasks such as image classification and object detection. VIP [44], R3M [45], MVP [40, 41], and MCR [46] aim to learn implicit features for robotic tasks by training on video data, while Thiea [47] and SUSIE [48] try to distil knowledge from pre-trained vision foundation models. SUGAR [43] and 3D-MVP [49] extend the pre-training tasks to the 3D domain. To further improve the semantic understanding, Voltron [7] and MPI [42] combine video-based tasks with text conditioning, enabling generalisation to diverse object manipulation tasks. However, video data often contains temporal redundancy, which may allow models to exploit smooth temporal transitions rather than learning meaningful representations [50]. On the other hand, we design a new pretext paradigm to directly predict goal image with textual instructions via asymmetric masking [34, 36].

3 Method

Given a visual observation $\mathbf{o}_s$ and associated language instruction $\mathbf{l}_{s \to f}$, our goal is to learn a policy that predicts a robot action $\mathbf{a}_{s \to f}$ that leads to a goal observation $\mathbf{o}_f$. We formulate this problem into two stages: *1) Visual-action representation learning*: We first train a model f_θ that outputs the predicted goal image, $\hat{\mathbf{o}}_f$, from $\mathbf{o}_s$ and $\mathbf{l}_{s \to f}$. This self-supervised task encourages the model to align visual and textual modalities, capturing how the instruction transforms the scene. *2) Robot action prediction*: We then fine-tune the model by attaching an action prediction head that maps the learned representation to a robot action $\hat{\mathbf{a}}_{s \to f}$. This action may take various forms depending on the task setting, e.g., $\hat{\mathbf{a}}_{s \to f} \in SE(2)$ for tabletop manipulation or $\hat{\mathbf{a}}_{s \to f} \in \mathbb{R}^9$ for joint angle control.

3.1 Learning visual-action representations

We propose a self-supervised approach of asymmetric masking to learn visual-action representations using the pretext task, which aims to reconstruct a partially masked goal image given $\mathbf{o}_s$ and $\mathbf{l}_{s \to f}$. Fig. 2 shows LaVA-Man's architecture and the proposed pretext task of goal image prediction.

Vision encoder. Let $\mathbf{o}_s$ and $\mathbf{o}_f$ be two images that capture the scene before and after the robot manipulation. Both images are firstly divided into N non-overlapping patches, where some patches in $\mathbf{o}_f$ are randomly masked, transforming $\mathbf{o}_f$ into $\tilde{\mathbf{o}}_f$. We design a pretext task that reconstructs $\mathbf{o}_f$ based on $\mathbf{o}_s$, $\tilde{\mathbf{o}}_f$, and a text embedding $\mathbf{e}_{s \to f}$ obtained from $\mathbf{l}_{s \to f}$:

$$
\begin{aligned}
\mathbf{h}_{s \to f} &= \Phi(\mathbf{o}_s, \tilde{\mathbf{o}}_f, \mathbf{e}_{s \to f}), \\
\hat{\mathbf{o}}_f &= \Psi_p(\mathbf{h}_{s \to f}).
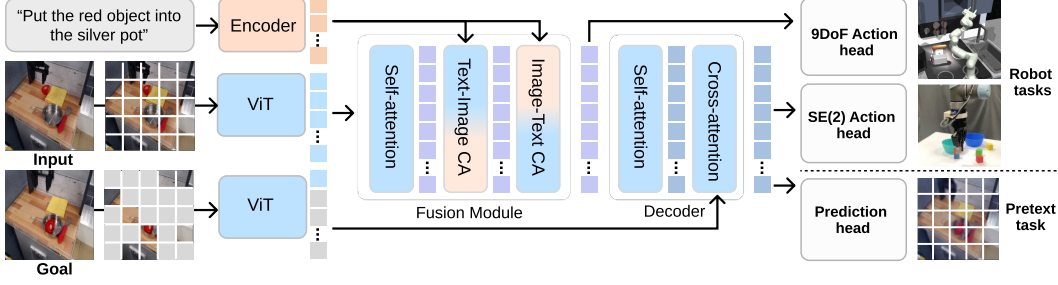\end{aligned}
\tag{1}
$$

Figure 2: **Network structure.** We use a fixed backbone while adapting different output heads for pretext and downstream robot tasks. We use a Siamese ViT encoder with asymmetric masking applied to the goal image only. Visual features from the input are first fused with text features and then integrated with those extracted from the masked goal image in the decoder. During inference, the goal image is fully masked as it is unknown. Additional output heads can be incorporated to support a wider range of downstream tasks. KEY – CA: cross-attention.

Here Φ consists of a siamese Vision Transformer (ViT) encoder that encodes the patches in $\mathbf{o}_s$ and $\mathbf{o}_f$ into the features $\mathbf{v}_s, \mathbf{v}_f$ and a lightweight decoder that decodes $\mathbf{v}_s$, $\mathbf{v}_f$, and $\mathbf{e}_{s \to f}$ to a feature vector $\mathbf{h}_{s \to f}$. Ψ_p is the goal-image prediction head that outputs the reconstruction $\hat{\mathbf{o}}_f$ given $\mathbf{h}_{s \to f}$. We use the CLIP text encoder to encode the text embeddings $\mathbf{e}_{s \to f}$ from $\mathbf{l}_{s \to f}$.

Visual-textual fusion. Inspired by GLIP [51], we perform a multi-stage cross-attention to fuse information from different modalities. First, we use text-to-image and image-to-text attention: $\mathrm{cross\_attn}(\mathbf{v}_s, \mathbf{e}_{s \to f})$ and $\mathrm{cross\_attn}(\mathbf{e}_{s \to f}, \mathbf{v}_s)$, to fuse the initial visual state and language. Then, $\mathbf{v}_f$ queries this fused feature via $\mathrm{cross\_attn}(\mathbf{v}_s, \mathbf{v}_f)$ to produce $\mathbf{h}_{s \to f}$. This enables the model to condition on both the initial image and the language instruction while grounding the goal image.

Goal-image prediction head. The per-patch feature $\mathbf{h}_{s \to f}$ is passed to a lightweight MLP head Ψ_p to generate the per-pixel RGB values for each patch in the predicted goal image $\hat{\mathbf{o}}_f$. We supervise the model using an $\mathcal{L}_2$ loss between $\hat{\mathbf{o}}_f$ and the ground-truth $\mathbf{o}_f$.

3.2 Robot action prediction

Following the previous stage, we fine-tune our model for downstream robot manipulation tasks with an additional lightweight head Ψ_a that predicts the robot action $\mathbf{a}_{s \to f}$:

$$\mathbf{a}_{s \to f} = \Psi_a(\mathbf{h}_{s \to f}, \mathbf{o}_s). \tag{2}$$

At test time, the model does not have $\mathbf{o}_f$ and $\tilde{\mathbf{o}}_f$. However, our pretext task is designed to learn from $\tilde{\mathbf{o}}_f$ with a high mask ratio, and hence our model can still predict $\mathbf{h}_{s \to f}$ as in Eq. 1 while using fully masked $\tilde{\mathbf{o}}_f$. We therefore include $\hat{\mathbf{o}}_f$ as additional input to the action head Ψ_a and rewrite Eq. 2 as:

$$\mathbf{a}_{s \to f} = \Psi_a(\mathbf{h}_{s \to f}, \mathbf{o}_s, \hat{\mathbf{o}}_f). \tag{3}$$

For tabletop manipulation tasks, we define the output action as $\mathbf{a}_{s \to f} = (\mathcal{T}_s, \mathcal{T}_f)$, where $\mathcal{T}_s, \mathcal{T}_f \in \mathbf{SE}(2)$ denote the pick and place poses of the end-effector. The model predicts an affordance map as an intermediate representation following [1, 8], from which $\mathcal{T}_s$ and $\mathcal{T}_f$ are extracted via a softmax operation. For the task of predicting joint angles on specific robot arms, we drop the decoder as [7, 42] and represent each action as a 9-DoF vector $\mathbf{a}_{s \to f} \in \mathbb{R}^9$, comprising seven joint angles and two indicators for grasp status.

4 The Omni-Object Pick-and-Place dataset

Overview. We introduce the OOPP dataset, a tabletop simulation benchmark consisting of 3,200 unique real-scanned objects across 180 distinct categories. As shown in Fig. 3 (a), existing tabletop simulation datasets such as Ravens [8] and VIMA [9] typically exhibit limited object diversity

Dataset	Num of classes	Num of instances	Inter-class variation	Intra-class variation	Real scanned objects
RLBench[†] [52]	100	100	✔	✘	✔
LIBERO[†] [53]	100	100	✔	✘	✔
Ravens [1]	52	52	✔	✘	✔
VIMA [9]	20	1800*	✔	✔	✘
OOPP (ours)	180	3200	✔	✔	✔

[†]RLBench and LIBERO focus on complex manipulation actions rather than diverse object interactions. We approximate the object number by the number of tasks as there is little intra-class object variation.

*VIMA provides 90 different textures per object. We treat each textured variant as a distinct instance here.

(a) Quantitative comparison   (b) Visualised examples
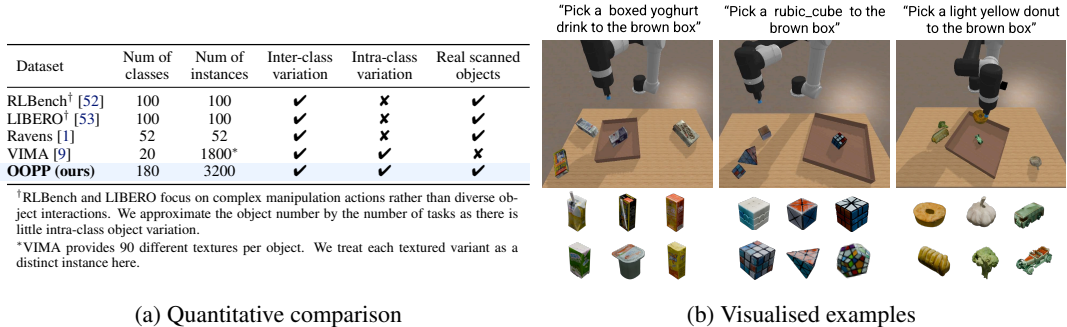
Figure 3: We show (a) Quantitative statistics of the proposed OOPP dataset compared to several existing datasets. (b) Visualised examples of manipulation sequences in OOPP datasets. The first row shows objects seen by the models and the second row shows unseen objects.

and rely on template-based language instructions. This restricts trained models to fixed, narrow vocabularies and prevents them from generalising to open-vocabulary settings. In contrast, OOPP consists of diverse object instances paired with rich language descriptions, enabling models to learn a broad range of object priors. OOPP also allows a comprehensive evaluation of a model's ability to ground diverse, open-vocabulary instructions to visual observations involving both inter-class and intra-class variations.

Data annotation. OOPP is built upon the previous benchmarks [8, 9] in the Pybullet Gym environment [11]. We adopt the real-scanned objects from the OmniObject3D dataset [10], filtering out the objects unsuitable for tabletop manipulation by retaining only objects with diameters between 4cm and 40cm. During simulation, we allow the robot to perform automatic manipulation with a pre-defined placeholder for collecting action annotations. These placeholders are then replaced with real objects. To ensure spatial diversity and prevent overcrowding, we adopt a KDTree-based spatial partitioning strategy inspired by [1], subdividing scenes into feasible regions for object placement. Based on the rich language description for each object provided by the original OmniObjects3D dataset [10], we generate diverse language instructions that are beyond just using object names, offering more natural and varied descriptions of object appearances as in Fig. 3.

Evaluation. To evaluate both intra- and inter-class generalisation, we partition the dataset into three mutually exclusive subsets. We use 160 object classes for training. For intra-class generalisation, we hold out a subset of instances from 20 categories in the training set for testing. For inter-class generalisation, we reserve 20 object categories that are entirely unseen during training. Unseen classes are sampled from four high-level semantic groups—*Food*, *Daily-use & Tools*, *Entertainment*, and *Others*—with each group contributing 3–8 classes. We define two manipulation tasks: 1) *packing-objects-group*, where identical objects are packed simultaneously, and 2) *packing-objects-sequence*, where different objects are packed sequentially. Together, these variants yield six different tasks. Please refer to the supplementary material for more details.

5 Experiments

5.1 Setup

Pre-training datasets. For the pretext task, we train our model using a total of 120k samples from our synthetic OOPP dataset and real-world robot video episodes from Bridge [12] and DROID [13]. For all robot video episodes, we extract only the first and last frames, paired with language instructions. The pre-training phase takes 24 hours with 3×A100 GPUs.

Baselines. We compare our method against two types of methods: 1) Foundation model-based methods such as CLIPort [1], which leverage web-scale vision-language models for pick-and-place

Table 1: **Results on the Ravens benchmark [8].** We report the results of multi-task performance trained on 1,000 demonstrations.

Tasks	Seen objects		Unseen objects		Coloured geometries						Average
	Packing obj-seq	Packing obj-grp	Packing obj-seq	Packing obj-grp	Put block in bowl	Stack block pyramid	Towers of hanoi	Packing boxes pairs	Assembling kits	Separating piles	
Pick-and-place methods											
Transporter [8]	0.49	0.59	0.25	0.45	0.23	0.02	0.1	0.43	0.20	0.44	0.32
CLIP only [5]	0.57	0.72	0.49	0.63	0.43	0.11	0.2	0.5	0.36	0.47	0.45
RN50-BERT [1]	0.46	0.64	0.41	0.59	0.45	0.02	0.21	0.47	0.27	0.5	0.40
CLIPort* [1]	0.80	**0.89**	0.55	0.74	0.59	0.38	0.69	0.80	0.60	0.71	0.68
CLIPort [1]	0.78	0.83	0.57	0.77	0.84	**0.69**	**0.82**	0.83	0.54	0.58	0.73
Pre-training methods											
No pre-training	0.59	0.72	0.40	0.56	0.33	0.22	0.18	0.64	0.55	0.46	0.47
Voltron [7]	0.58	0.7	0.43	0.57	0.80	0.36	0.24	0.70	0.60	0.43	0.54
MPI‡ [42]	0.56	0.75	0.41	0.66	0.67	0.22	0.42	0.42	0.46	0.46	0.50
Ours	**0.83**	0.84	**0.77**	**0.83**	**0.98**	0.57	0.67	**0.94**	**0.75**	**0.93**	**0.81**

*indicates that the model was tested using checkpoints provided by the authors, while others are trained on the same downstream data. MPI‡ is our reimplementation of MPI, trained in a self-supervised manner. KEY – obj: objects, seq: sequence, grp: group.

Table 2: **Results on the OOPP dataset.** We report the results trained on 1,000 demonstrations and tested on 100 demonstrations in the test set.

Method	Packing obj-seq	Packing obj-intra-seq	Packing obj-inter-seq	Packing obj-grp	Packing obj-intra-grp	Packing obj-inter-grp	Average
CLIPort [1]	0.62	0.60	0.48	0.65	0.57	0.60	58.7
Voltron [7]	0.74	0.68	0.54	0.80	0.71	0.72	69.8
MPI‡ [42]	0.69	0.59	0.58	0.73	0.65	0.67	65.1
Ours	**0.84**	**0.74**	**0.73**	**0.84**	**0.86**	**0.77**	**79.6**

MPI‡ is our reimplementation of MPI, trained in a self-supervised manner. KEY – obj: objects, seq: sequence, intra: intra-class, inter: inter-class, grp: group.

tasks; and 2) Self/weakly supervised methods, including Voltron [7] and MPI [42], which learns representations from human/robot manipulation episodes for grounding language-guided manipulation. Since MPI pre-training also relies on supervision from object bounding boxes, we remove the detection head and make the pre-training in a fully self-supervised manner, denoted as MPI‡. Unless specified, we re-train every baseline using our data for fair comparison.

Downstream tasks and benchmarks. We evaluate our model on various downstream tasks, including both simulation and real-world environments (see supplementary for details):

- **Ravens** [8]: This dataset provides ten different tabletop rearrangement tasks. For fair comparison, we re-train all pre-training methods and fine-tune them using the same set of demonstrations.
- **OOPP**: We also evaluate our models on our proposed dataset with diverse object classes and instances for intra- and inter-class evaluation. The training procedure follows that of Ravens.
- **Referring expression grounding** [7]: This benchmark evaluates target object localisation in cluttered scenes based on language instructions, serving as a prerequisite for visuomotor control.
- **Franka Kitchen** [54]: This popular benchmark provides comparisons with other state-of-the-art methods on five visuomotor control tasks in a simulated kitchen environment.
- **Real-robot experiments**: We further evaluated our model on ten different manipulation tasks using UR5 robot arms to evaluate real-world generalisation. We capture the current observation from the top-down view and generate the affordance map for action execution accordingly.

5.2 Evaluation

In *Ravens and OOPP*, the model outputs SE(2) tabletop actions. In Tab. 1, our method outperforms other pre-training-based models and even the method designed for pick-and-place. Our method shows strong capability in interpreting actions with real-scanned objects, as demonstrated in the seen and unseen categories in Tab. 1 and all intra- and inter-class generalisation tasks in Tab. 2. Our model shows limited performance in coloured geometries, possibly because they are less common in real-world scenarios in our pre-training data.

Table 3: **Results on Referring Expression Grounding [7] and Franka Kitchen [54].** Our method outperforms self-supervised methods and is comparable to leading supervised methods. The results of existing methods are quoted from [42] and [7]. Minimum, Medium and Maximum in sub-table (a) denote the scenarios categorised according to the level of clutter.

(a) Referring Expression Grounding

Method	Minimum	Medium	Maximum	Total
Supervised pre-training methods				
MPI [42]	0.94	**0.98**	0.95	0.96
SUGAR [43]	0.98	0.97	**0.96**	**0.97**
Self-supervised pre-training methods				
R-R3M [45]	0.64	0.68	0.55	0.63
MVP [40]	0.51	0.52	0.40	0.49
CLIP [5]	0.67	0.77	0.60	0.68
Voltron [7]	0.88	**0.97**	0.90	0.91
Ours	**0.99**	0.96	**0.96**	**0.97**

(b) Franka Kitchen

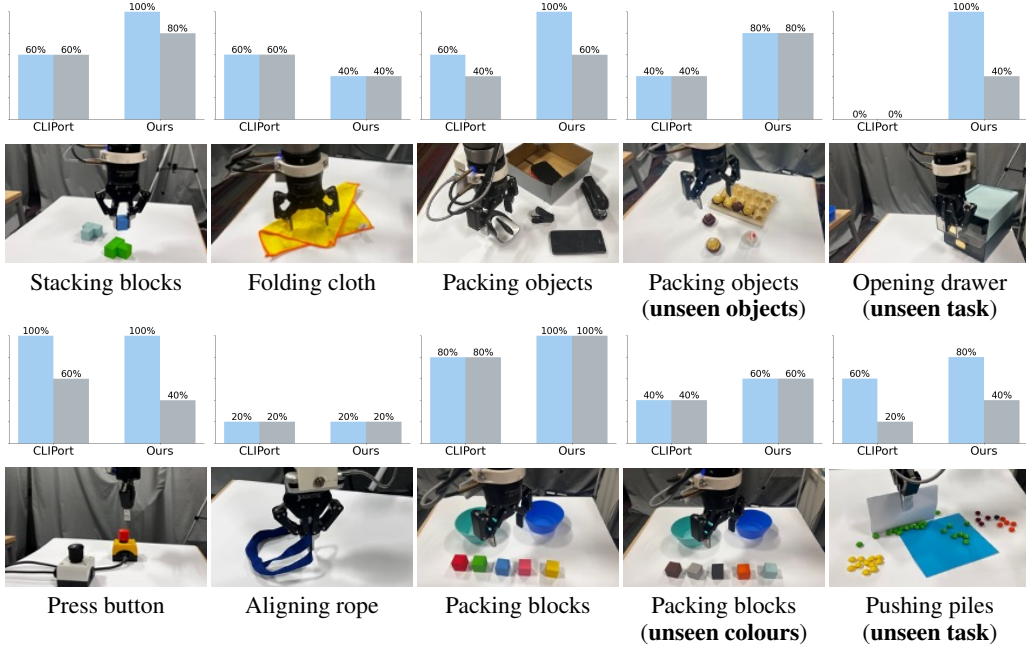| Method	Trun knob	Open door	Flip switch	Open microwave	Slide door	Average
Supervised pre-training methods						
MPI [42]	89.0	**57.7**	93.7	54.0	**100.0**	78.9
Self-supervised pre-training methods						
INSUP [55]	28.0	18.0	50.0	26.7	75.7	39.7
CLIP [5]	26.3	13.0	41.7	24.7	86.3	38.4
R3M [45]	53.3	50.7	86.3	59.3	97.0	69.5
MVP [40]	79.0	48.0	90.7	41.0	**100.0**	71.7
Voltron [7]	76.0	45.3	91.0	41.0	99.3	70.5
Ours	**90.0**	50.7	**94.0**	**61.3**	**100.0**	**79.2**

Figure 4: **Real-robot experiments.** We report the ▇ perceptual score and ▇ physical score, which indicate the success rates for affordance perception and physical robotic tasks, respectively. The tasks of *Opening drawer* and *Pushing piles* are not included in the training demonstrations.

In *Franka Kitchen*, as in Tab. 3 (b), our method achieves state-of-the-art results even compared with supervised pre-training methods [42, 43]. Unlike Ravens and OOPP, the model in *Franka Kitchen* is required to predict the next timestep's joint angles of the robot arm. We hence employ the encoder and fusion module in our model as a frozen backbone and discard the decoder, consistent with other compared baselines.

Referring expression grounding results in Tab. 3 (a) demonstrate that our method achieves better or comparable results, even compared with supervised methods [42, 43]. This validates that the goal-image prediction pretext task can learn the association between the target object and the corresponding language instructions, resulting in accurate object localisation.

In *real robot experiments*, we evaluate our model on a real robot across ten distinct tasks, as shown in Fig. 4. Each task contains 5 different scenarios, and we calculate the overall success rates. The performance is consistent with the results observed in the simulation. Notably, it demonstrated strong generalisation to unseen colours, unseen objects, and even previously unseen tasks. In Fig. 4 we report both perceptual score and physical score, to account for the case where the model correctly outputs the affordance but fails during physical execution due to inaccurate control and object movements, showing the perception-to-physical gap in real-world scenarios (see Sec. 7 for details).
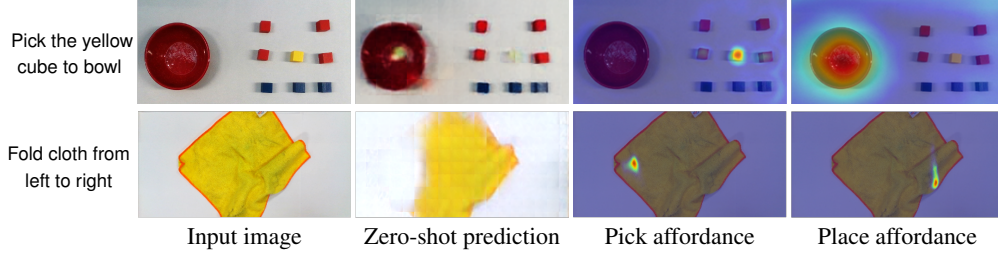
Figure 5: **Qualitative examples from real-robot manipulation.** We show qualitative examples in our real-robot environment. Our goal-image predictions are blurry as in other MAE-based methods [56, 36, 34], but still show plausible object placement and the results are aligned with the input text instructions, which leads to accurate manipulation. (Only affordances for translation are shown)

Method	Rotation tasks	Translation tasks	Avg.
w/o fusion module	82	66	72
w/o OOPP data	87	65	74
w/o Asym. mask	91	68	77
Ours	**96**	**71**	**81**
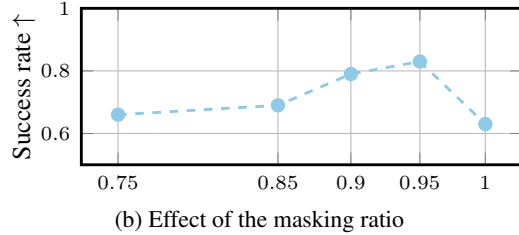
(a) Effect of pre-training strategy

(b) Effect of the masking ratio

Figure 6: **Ablation analysis.** We analyse the effect of pre-training strategies and masking ratios on the Franka Kitchen benchmark and Ravens benchmark, respectively.

5.3 Analysis

In Fig. 6 (a), we present a set of ablation studies on the Franka Kitchen benchmark to evaluate the contribution of several key components of our approach. We evaluate the effect of removing the fusion module, which injects the language embedding directly into the decoder via cross-attention. The suboptimal results are consistent with other recent works [57, 51], showing the importance of integrating visual and textual modalities before decoding. We also examine the impact of training with the proposed OOPP dataset, which introduces a wide variety of object classes. Results show that the downstream performance can be significantly boosted, indicating the model benefits from broader object priors in the OOPP dataset to learn generalisable visual-action representations.

Finally, we evaluate the effect of the asymmetric masking strategy and the optimal masking ratio. Without asymmetric masking, the model may not be able to fully capture the underlying causality in language-guided manipulation, leading to reduced accuracy in downstream tasks, as shown in Fig. 6 (a). In Fig. 6 (b), we further study the impact of different masking ratios by training models under four commonly used settings [33, 36, 58]. A 95% masking ratio performs the best. Lower ratios leak too much information, making learning less effective, while a 100% ratio, where prediction relies entirely on input, leads to convergence issues due to the future state's ambiguity.

6 Conclusion

We presented a visual-action representation learning framework for robot manipulation that leverages goal-image prediction as a pretext task to capture the underlying causality in language-guided robot manipulation. To facilitate learning from diverse object priors, we also proposed the OOPP dataset, which provides a rich collection of tabletop manipulation episodes. Through extensive evaluations in both simulated and real-world settings, our method demonstrates superior performance over existing baselines, exhibiting strong generalisation and effective knowledge transfer across tasks. Future work includes scaling training to diverse video datasets to learn a robust and generalisable representation and extend our learned representations for more complex robotic tasks.

7 Limitations

Manipulating precision. Our model exhibits limited performance when dealing with small objects or those with fine-grained structures, such as in *Towers-of-Hanoi* and *Stack-Block-Pyramid* (Tab. 1). This is attributed to the use of ViTs, which process images as patches and may overlook fine details that are important for accurate manipulation, as shown in Fig 7. Recent methods that enhance the spatial resolution by learning to upsample ViT features, such as DPT [59] or FeatUp [60], could potentially mitigate this issue and are promising directions for future exploration.
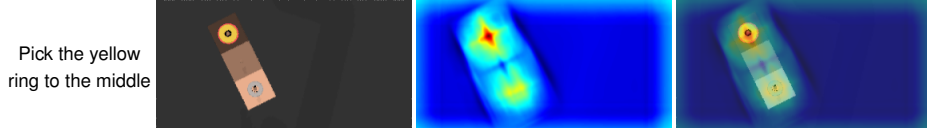

Figure 7: Visualised affordance of *Towers-of-Hanoi* task from the top-down view in Ravens.

Pseudo affordance. On the benchmark of Ravens [8] and OOPP, our method predicts a pseudo-affordance map where the pixel with the highest value is treated as the manipulation point (See Fig. 7). While this approach simplifies fine-tuning, it can be problematic for objects with complex geometry. In such cases, the manipulation can fail due to subtle physical interactions even if the predicted pick-and-place poses are geometrically correct. This leads to a discrepancy between the perceptual score and the actual physical success rate, as illustrated in Fig. 4. Incorporating self-supervised learning that accounts for human or robot interaction dynamics could help mitigate this issue by enabling the model to learn priors over common manipulable regions.

3D awareness. Our model lacks explicit 3D understanding and relies solely on 2D visual cues. As a result, it can struggle to distinguish between objects with similar textures or appearances but differing 3D shapes or structures. Incorporating self-supervised learning methods that promote geometric awareness may help the model learn 3D priors and improve robustness in such cases.

Articulated objects. In real-world scenarios, some objects are composed of multiple rigid parts and interactable. However, our visual pre-training primarily focuses on rigid object manipulation and does not explicitly model object-part interactions. Including articulated objects with several new tasks in our OOPP dataset can potentially help capture these complex interactions.

Blurry prediction results. MAE-based methods are known to produce blurry reconstructions lacking high-frequency details [34, 36, 56]. Our goal image prediction also suffers from the blurry issue, as shown in Fig 1. However, our objective is not to generate high-quality images, but rather to use goal image prediction as a pretext task for learning semantic representations. In addition, blurriness reflects the missing of high-frequency components in the image, which often correspond to perceptual details rather than semantic contents. In fact, despite the perceptual blur, the preserved semantic structure in our predictions demonstrates that the model captures semantic information effectively. The lack of perceptual fidelity may limit interpretability for humans, and we consider this as future work.

8 Acknowledgement

References

[1] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the Conference on Robot Learning*, 2021.

[2] R. Wang, J. Mao, J. Hsu, H. Zhao, J. Wu, and Y. Gao. Programmatically grounded, compositionally generalizable robotic manipulation. In *Proceedings of the International Conference on Learning Representations*, 2023.

[3] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the Conference on Robot Learning*, 2023.

[4] M. Jia, H. Huang, Z. Zhang, C. Wang, L. Zhao, D. Wang, J. X. Liu, R. Walters, R. Platt, and S. Tellex. Learning efficient and robust language-conditioned manipulation using textual-visual relevancy and equivariant language mapping. *IEEE Robotics and Automation Letters*, 2025.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

[6] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In *Proceedings of the Advances in Neural Information Processing Systems*, 2022.

[7] S. Karamcheti, S. Nair, A. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Proceedings of Robotics: Science and Systems*, 2023.

[8] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In *Proceedings of the Conference on Robot Learning*, 2021.

[9] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. VIMA: General robot manipulation with multimodal prompts. In *Proceedings of the International Conference on Machine Learning*, 2023.

[10] T. Wu, J. Zhang, X. Fu, Y. Wang, L. P. Jiawei Ren, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[11] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[12] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. BridgeData V2: A dataset for robot learning at scale. In *Proceedings of the Conference on Robot Learning*, 2023.

[13] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024.

[14] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Proceedings of Robotics: Science and Systems*, 2017.

[15] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis. Single image 3d object detection and pose estimation for grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014.

[16] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017.

[17] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox. Self-supervised 6d object pose estimation for robot manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

[18] G. Du, K. Wang, S. Lian, and K. Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, 54(3):1677–1734, 2021.

[19] K. Zakka, A. Zeng, J. Lee, and S. Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

[20] S. Song, A. Zeng, J. Lee, and T. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3): 4978–4985, 2020.

[21] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel. Learning to manipulate deformable objects without demonstrations. *Proceedings of Robotics: Science and Systems*, 2020.

[22] H. Wu, J. Ye, X. Meng, C. Paxton, and G. S. Chirikjian. Transporters with visual foresight for solving unseen rearrangement tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[23] S. Huang, H. Chang, Y. Liu, Y. Zhu, H. Dong, P. Gao, A. Boularias, and H. Li. A3vlm: Actionable articulation-aware vision language model. In *Proceedings of the Conference on Robot Learning*, 2024.

[24] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[25] X. Li, M. Zhang, Y. Geng, H. Geng, Y. Long, Y. Shen, R. Zhang, J. Liu, and H. Dong. Mani-pllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[26] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015.

[27] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*, 2016.

[28] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations*, 2018.

[29] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[32] J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, 2019.

[33] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[34] A. Gupta, J. Wu, J. Deng, and F.-F. Li. Siamese masked autoencoders. In *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[35] T. Kim, D. Han, B. Heo, J. Park, and S. Yun. Token bottleneck: One token to remember dynamics. *arXiv preprint arXiv:2507.06543*, 2025.

[36] P. Weinzaepfel, V. Leroy, T. Lucas, R. Brégier, Y. Cabon, V. Arora, L. Antsfeld, B. Chidlovskii, G. Csurka, and J. Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In *Proceedings of the Advances in Neural Information Processing Systems*, 2022.

[37] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020.

[38] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *Proceedings of the International Conference on Learning Representations*, 2020.

[39] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. In *Proceedings of the Advances in Neural Information Processing Systems*, 2020.

[40] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of the Conference on Robot Learning*, 2023.

[41] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[42] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo, et al. Learning manipulation by predicting interaction. *Proceedings of Robotics: Science and Systems*, 2024.

[43] S. Chen, R. Garcia, I. Laptev, and C. Schmid. SUGAR: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[44] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *Proceedings of the International Conference on Learning Representations*, 2022.

[45] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *Proceedings of the Conference on Robot Learning*, 2023.

[46] G. Jiang, Y. Sun, T. Huang, H. Li, Y. Liang, and H. Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. In *Proceedings of the International Conference on Learning Representations*, 2025.

[47] J. Shang, K. Schmeckpeper, B. B. May, M. V. Minniti, T. Kelestemur, D. Watkins, and L. Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Proceedings of the Conference on Robot Learning*, 2025.

[48] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *Proceedings of the International Conference on Learning Representations*, 2024.

[49] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal. 3d-mvp: 3d multiview pre-training for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[50] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, L. Yuan, and Y.-G. Jiang. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[51] L. H. Li, P. Zhang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[52] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[53] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[54] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning*, 2020.

[55] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[56] S. He, T. Guo, T. Dai, R. Qiao, C. Wu, X. Shu, and B. Ren. VLMAE: Vision-language masked autoencoder. *arXiv preprint arXiv:2208.09374*, 2022.

[57] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision*, 2024.

[58] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Proceedings of the Advances in Neural Information Processing Systems*, 2022.

[59] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[60] S. Fu, M. Hamilton, L. Brandt, A. Feldman, Z. Zhang, and W. T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *Proceedings of the International Conference on Learning Representations*, 2024.

[61] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2021.

[62] K.-J. Wang, Y.-H. Liu, H.-T. Su, J.-W. Wang, Y.-S. Wang, W. Hsu, and W.-C. Chen. Ocid-ref: A 3d robotic dataset with embodied language for clutter scene grounding. In *Association for Computational Linguistics (ACL)*, 2021.

Appendix

1 Introduction

We provide additional material that supports our paper.

- We invite readers to visit our website about the proposed pretext task (goal-image prediction), Omni-Object Pick-and-Place (OOPP) dataset, and real and simulated robot manipulation.
- In Sec. 2, we provide more details about our method, including our backbone and different output heads utilised in different downstream tasks.
- In Sec. 3, we describe more details of each downstream task, including the real-robot experiment.
- In Sec. 4, we provide additional examples of the proposed pretext task, goal-image prediction, and our OOPP dataset.

2 Method details

Backbone architecture. We adopt ViT-Base [61] with a patch size of 16 as our backbone. The original Vit-Base architecture consists of 12 attention blocks with an embedding dimension of 768, and its the decoder typically comprises of 8 attention blocks with an embedding dimension of 512. For a fair comparison with other methods using ViT-Base as a backbone, we include 6 attention blocks and 6 bi-directional attention blocks in our encoder. In the pre-training stage, our goal-image prediction head is a one-layer fully connected network.

Affordance head. Our affordance head for generating $\mathbf{SE}(2)$ robot actions (the 2D location and 1D rotation) for tabletop manipulation tasks (*Ravens*, *OOPP*, and our real robot experiments) consists of 4 convolutional layers with skip connection to convert the output from the transformer backbone to the affordance map. To generate a rotation angle, we expand a single affordance map into 36 instances, where each instance represents a 10-degree step. We apply the softmax function to the expanded affordance map, which identifies the position and corresponding rotation angle. Since other pre-training methods have not been previously evaluated on this benchmark, we reimplemented them and used a unified action head across all pre-training baselines. For other Pick-and-Place methods, we retain their original action heads which are structurally similar but typically deeper due to their use of ResNet backbones. In addition, since our pre-training methods are designed to predict the goal image, we found that concatenating the goal image into the convolutional layers facilitates improved affordance generation, as the two tasks are closely correlated.

Action head and detection head. Following the baseline established by Voltron [7] and MPI [42], we adopt the same shallow MLP policy network to predict joint velocities of $\mathbb{R}^9$ (7 degree of freedom and two for grasp status) for robot actions in the Franka Kitchen benchmark, and to regress bounding boxes in the Referring Expression Grounding benchmark. We use features after the fusion module for two main reasons: First, for fair comparison: previous methods, including R3M [45], MVP [40] and Voltron [7], only evaluate frozen representations dropping the decoder part. Since we directly report their results on these benchmarks, we adopt a consistent setting. Second, based on task requirements, both benchmarks rely on understanding the current state or the next state, while features after the decoder in our model represent the goal image and correspond to the final state. Therefore, using features before the decoder is more suitable for these two benchmarks.

3 Benchmark details

3.1 Ravens

Overview. Ravens [8] is a simulated benchmark to evaluate tabletop pick-and-place robot manipulation tasks. We use PyBullet OpenAI Gym [11] based on the configuration described in CLIPort [1].

We chose 8 language-conditioned tasks for the experiment, as shown in Fig. 8, including *Packing seen or unseen Google objects sequence*, *Packing seen or unseen Google objects group*, *Put block in the bowl*, *Stack blocking pyramid*, *Towers of Hanoi*, *Packing boxes pairs*, *Assembling kits*, and *Separating piles*. For details of each task, including the train and test split of objects, and the language instruction template, please refer to [1, 8]. Note that we did not split the tasks according to seen or unseen colours, as all the methods have already "seen" all the colours in their pre-train models or the unsupervised pre-training phase. Therefore, we combine both "seen" and "unseen" splits of colours into a single task and the scores reflect the model's perception ability on all the colours.

Evaluation details. We evaluated the capability of the proposed methods on multi-task experiments in the benchmark. Specifically, we trained the model using 1,000 demonstrations drawn from all task categories and assessed performance on another 100 test demonstrations per task. Since prior pre-training methods [7, 42] have not been evaluated on this benchmark, we reimplemented their models using the original codebases and pre-trained them on the same pre-training data as our approach. When adapting to the downstream tasks, these pre-training methods were also equipped with the affordance head described in Sec. 2. Subsequently, all baselines, including the Pick-and-Place baselines, were fully fine-tuned on the same set of downstream demonstrations, following the evaluation protocol outlined in [1], to ensure a fair comparison.

3.2 Franka Kitchen

Overview. Franka Kitchen [54] is a well-established benchmark for evaluating the efficacy of visual representations in facilitating the learning of visuomotor control policies from limited demonstrations. This benchmark comprises five distinct visuomotor control tasks, as shown in Fig. 9, each captured from two camera viewpoints.

Evaluation details. We use the action head described in Sec. 2 for predicting joint velocities. As prior works [5, 7, 40, 42, 45], which leverage supervised or unsupervised pre-training for robot manipulation, commonly adopt this benchmark, we directly report the results stated in their original papers and compare our method against these approaches. Following the evaluation protocols widely adopted in these works, we trained the action head with the backbone frozen using 25 demonstrations, and report average success rates across five tasks, two viewpoints, and three random seeds.
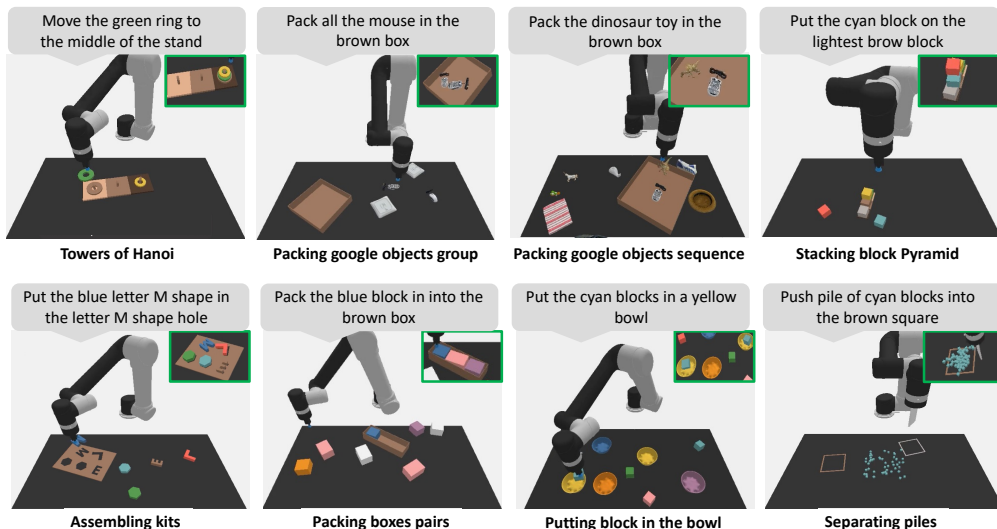

Figure 8: **Examples of eight robot manipulation tasks in the simulator.** The language instructions are on the top of each image and the final states are shown in the green box.

| Open the microwave | Turn on the light | Slide the right door | Turn the stove top knob | Open the left door |

Figure 9: **Tasks in the Franka Kitchen benchmark.**

3.3 Referring Expression Grounding

Overview. The goal of this task is to predict a bounding box of an object in a cluttered scene based on the nature language expression. This task offers the evaluation of language-conditioned scene understanding and object recognising ability, which serves as an important prerequisite for language-based robot manipulation. The benchmark is based on OCID-Ref Dataset [62], which provides representatives scenes in robotics settings. The benchmark also provides splits based on the clutter level.

Evaluation details. We use a shallow MLP as detection head to regress the bounding box directly. We use the evaluation codebase provided by [7]. Similar to Franka Kitchen, we report the results stated in the paper for each baseline [7, 40, 42, 43, 45]. The evaluation metrics are the average precision at 0.25 IoU under each clutter level.

3.4 Real robot experiments

Overview. We validate the applicability of our method in real-world scenarios. We validate our model on 10 manipulation tasks: *Stacking blocks*, *Folding cloth*, *Packing objects*, *Opening drawer*, *Pressing button*, *Aligning rope*, *Packing blocks*, and *Pushing piles*. Each task contains 5 different scenarios that differ in either objects or locations. We manually collected 200 training demos that contain robot image pairs, language descriptions, and annotations for real-world fine-tuning. Five colored blocks and five unseen objects were excluded from the training demonstrations, and no demonstrations from the *Opening drawer* or *Pushing piles* tasks were included, although similar tasks are present in the images used in the self-supervised pre-training phase. This design aims to evaluate whether the model can effectively generalised from the self-supervised pre-training, rather than relying on downstream demonstrations.

Evaluation details. Fig. 11 shows our real-robot environment and objects. We train both our model and CLIPort [1] on our manually collected training demos. We utilise a 6-degree-of-freedom (6-DoF) UR5 robotic arm, Robotiq 2F-85 two-finger gripper, and an Intel RealSense RGB-D camera for

| The timer on the rear left of the pudding_box | The blue shampoo on the rear left of the cylinder food_can | The cuboid red cereal_box on the rear left of the sphere red apple. |

Figure 10: **Examples of the task of Referring Expression Grounding.** This task offers varied language instructions involving a wide range of objects and scenes.

Table 4: **Real robot tasks settings.** We present the language template, variable factors and the success condition for each real robot task

Task name	Language template	Variable factors	Success condition
Stacking blocks	Stack the {color} block on the {color} blocks	Block color and position (pick/place)	Correct block stacked on target block
Folding cloth	Fold the cloth from {direction} to the {direction}	Cloth color, initial orientation	Grasp and fold directions match the template
Packing objects	Pick the {object} into the {object}	Object type, distractor objects	Target object placed into correct container
Opening drawer	Pull out the drawer	Drawer position	Drawer fully opened
Pressing button	Press the {color} button	Button color, button location	Correct button touched
Aligning rope	Align the rope from {direction} to the {direction}	Rope position, direction variation	Rope aligned from start to target direction
Packing blocks	Put the {color} block into the {color} bowl	Block color, bowl color	Correct block placed into matching bowl
Pushing piles	Push the pile of {objects} to the {color} area	Object color, target area color	Pile reaches target area within five attempts

Real robot setup   Seen objects and distractors   Unseen objects

Figure 11: **Real robot experiment.** The figure shows our physical robot setup along with both seen and unseen objects. Seen and unseen colored blocks are also included.

our real-world experiments. We capture the top-down RGB observation which covers the workspace of 60 cm $\times$ 30 cm, and the image from the camera is resized to 320×160 pixels.

Task details. Here we show the language template, variable factors and the success condition for each real robot task in Tab. 4. Images for each task are presented in the main paper.

4 Additional details

Goal-image prediction. We provide more qualitative examples for the goal-image prediction and the results during the training of masked auto-encoders, as shown in Fig. 12 and Fig. 13. In Fig. 12 we show examples of the same input image but with different language instructions. The results show that our model can effectively predict different goal states given different language-based instructions and the initial observation. Namely, these results show that our model can interpret the input instructions, factorize different objects that need to be manipulated, and further understand the spatial location or direction in the scene. This indicates the learned visual-action representations after the self-supervised learning with the pretext task effectively associate visual states with action. In Fig. 13, we present example results in our pre-training phase. Our predicted goal images are blurry as in other MAE-based methods [34, 36, 56]. These results demonstrate that our approach successfully learns to predict the goal image.

Dataset details. We build our Omni-Object Pick-and-Place (OOPP) dataset upon the previous benchmarsk [1, 9] in the PyBullet Gym enviornment [11]. We manually selected 180 real-scanned object classes from the OmniObject3D dataset [10], focusing on those suitable for robotic manipulation, resulting in a total of 3,200 distinct instances. For each object, we reduced the mesh resolution to 20K faces to enable efficient rendering in simulation. Additionally, we filtered out objects that were either too large or too small (i.e., with dimensions less than 4cm or greater than 40cm). Our dataset includes full robot manipulation episodes, comprising of annotated robot actions, language instructions, and simulation-generated rewards. We use 160 object classes for training. For evaluating intra-class generalisation, we hold out a subset of instances from 20 categories included in the training set. For inter-class generalisation, we reserve 20 object categories that are entirely unseen
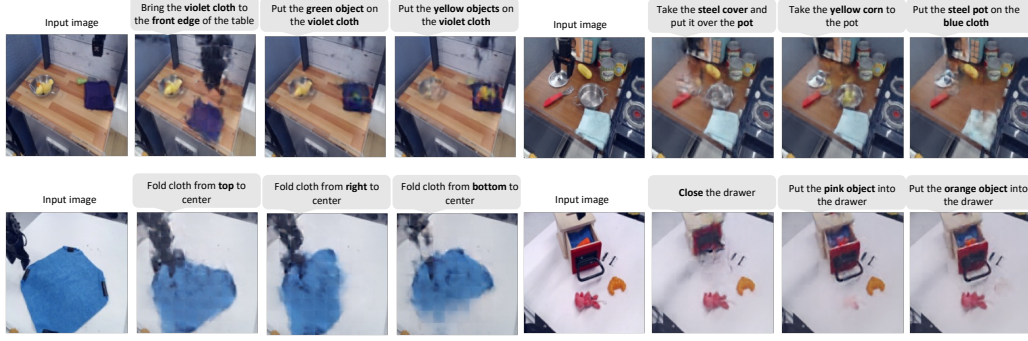
Figure 12: **Examples of goal prediction images.** Given the same input image with a different language, the model effectively understands the diverse language instructions and predicts goal images aligning with the semantics. Notably, all images belong to the test set and are novel to the model.
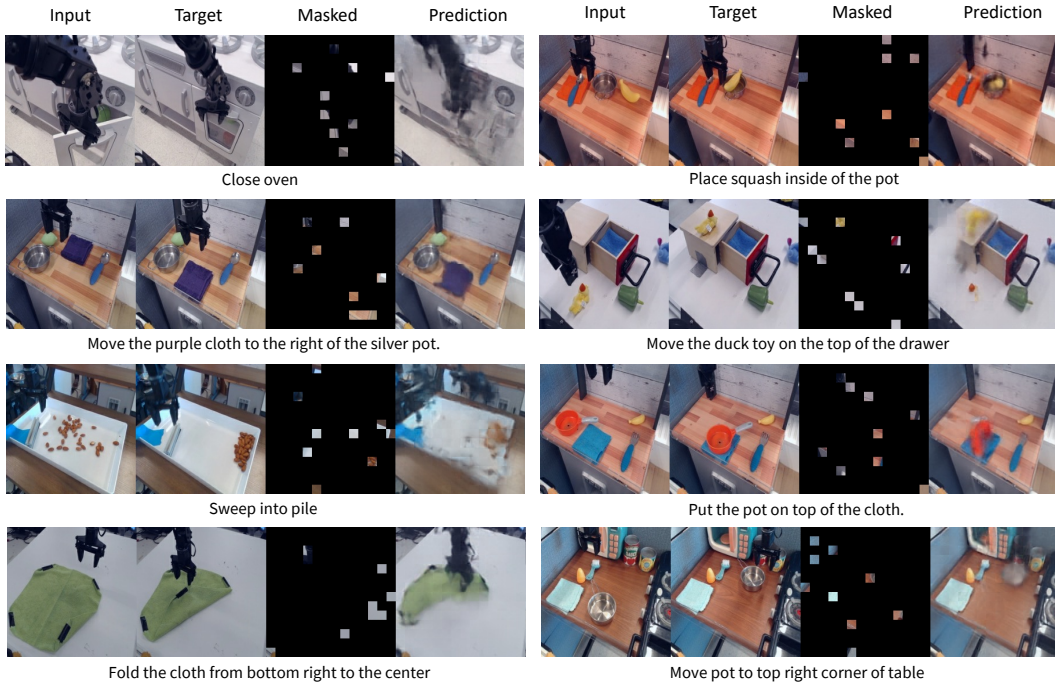

Figure 13: **Examples of masked image prediction.** We show the qualitative examples of our masked auto-encoders on the validation set. The text descriptions are shown below each sample.

during training. Fig. 14 illustrates the first 50 examples across different object classes, while Fig. 15 highlights examples of intra-class variation. The held-out categories for inter-class generalisation are sampled from four high-level semantic groups, as detailed in Table 5.
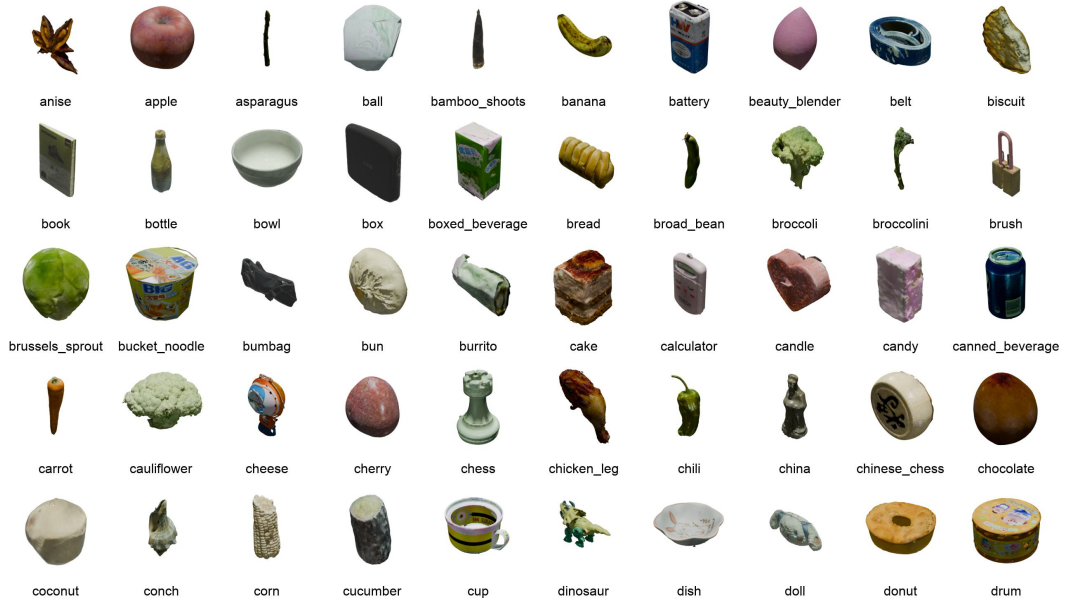
Figure 14: **Examples of objects from different classes.**

Figure 15: **Examples of intra-class variance.**

Table 5: **Semantic group division of seen and unseen classes**. We divided all objects into four semantic groups. We selected unseen classes from each group in proportion to the number of object classes it contains, ensuring an even distribution of unseen categories.

Semantic Group	Seen Classes	Unseen Classes
Food	red_jujube, corn, strawberry, anise, pizza, longan, loquat, chocolate, brussels_sprout, haw_thorn, green_bean_cake, cucumber, litchi, cake, dumpling, mooncake, rice_cake, puff, water_chestnut, mushroom, broccoli, pastry, egg_tart, kiwifruit, fig, cheese, chili, tomato, lemon, oyster, steamed_bun, carrot, mangosteen, bread, ginger, waffle, bun, peach, apple, pear, potato, zongzi, pomegranate, onion, egg, banana, chicken_leg, sausage, coconut, broccolini, hami_melon, durian, asparagus, walnut, mango, loquat, bucket_noodle	orange, biscuit, shrimp, garlic, donut, sweet_potato, candy, cherry, pancake
Daily-use	thimble, beauty_blender, battery, candle, calculator, plug, watch, nipple, power_strip, bottle, medicine_bottle, tissue, belt, dish, flash_light, canned_beverage, fork, cup, teapot, book, glasses_case, bowl, tape_measure, speaker, laundry_detergent, glasses, wallet, insole, bumbag, fan, knife, umbrella, kettle, light, picnic_basket, hammer, shoe, hat, laptop, vase, ornaments, spanner	soap, mouse, scissors, teapot, shampoo, toothpaste
Entertainment	toy_boat, toy_plant, toy_car, toy_plane, timer, whistle, doll, table_tennis_bat, toy_motorcycle, drum, remote_control, garage_kit, china, chess, Chinese_chess, rubik_cube, dinosaur	toy_bus, teddy_bear, toy_animals
Others	hairpin, lotus_root, house (model), plant, dumbbell, package, bamboo_shoots, brush, flute, ornaments, conch, magnet, box	flower_pot, red_wine_glass