



HAL
open science

SUGAR: Pre-training 3D Visual Representations for Robotics

Shizhe Chen, Ricardo Garcia, Ivan Laptev, Cordelia Schmid

► **To cite this version:**

Shizhe Chen, Ricardo Garcia, Ivan Laptev, Cordelia Schmid. SUGAR: Pre-training 3D Visual Representations for Robotics. CVPR 2024 - IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2024, Seattle WA, United States. pp.18049-18060, 10.1109/CVPR52733.2024.01709 . hal-04721493

HAL Id: hal-04721493

<https://hal.science/hal-04721493v1>

Submitted on 4 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SUGAR : Pre-training 3D Visual Representations for Robotics

Shizhe Chen[†] Ricardo Garcia[†] Ivan Laptev^{*} Cordelia Schmid[†]

[†] Inria, École normale supérieure, CNRS, PSL Research University

^{*} Mohamed bin Zayed University of Artificial Intelligence

https://cshizhe.github.io/projects/robot_sugar.html

Abstract

Learning generalizable visual representations from Internet data has yielded promising results for robotics. Yet, prevailing approaches focus on pre-training 2D representations, being sub-optimal to deal with occlusions and accurately localize objects in complex 3D scenes. Meanwhile, 3D representation learning has been limited to single-object understanding. To address these limitations, we introduce a novel 3D pre-training framework for robotics named **SUGAR** that captures semantic, geometric and affordance properties of objects through 3D point clouds. We underscore the importance of cluttered scenes in 3D representation learning, and automatically construct a multi-object dataset benefiting from cost-free supervision in simulation. SUGAR employs a versatile transformer-based model to jointly address five pre-training tasks, namely cross-modal knowledge distillation for semantic learning, masked point modeling to understand geometry structures, grasping pose synthesis for object affordance, 3D instance segmentation and referring expression grounding to analyze cluttered scenes. We evaluate our learned representation on three robotic-related tasks, namely, zero-shot 3D object recognition, referring expression grounding, and language-driven robotic manipulation. Experimental results show that SUGAR’s 3D representation outperforms state-of-the-art 2D and 3D representations.

1. Introduction

Visual perception plays an essential role in robotics and enables autonomous agents to understand and interact with their physical environment. Nevertheless, learning generalizable visual representations for robotics is challenging due to the scarcity of real robot data and the large variety of real-world scenes. Despite substantial efforts in robot data accumulation [2, 76, 77] and augmentation [39, 85], it remains prohibitively expensive to collect large-scale datasets comprising a broad range of robotic tasks.

To alleviate the burden of data collection, recent endeavors [36, 37, 48, 49, 51, 62] have sought to leverage large-scale internet data to pre-train 2D visual representations for

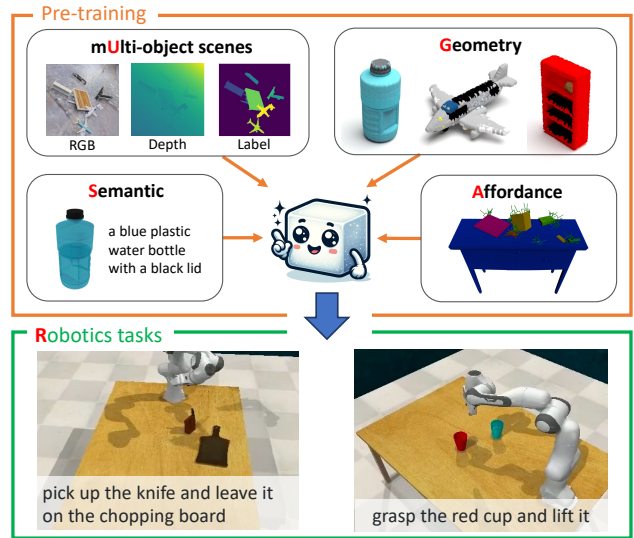



Figure 1. We introduce SUGAR , a pre-training framework for robotic-related tasks, which learns semantic, geometry and affordance on both single- and multi-object scenes.

robotics. For example, MVP [62], VIP [48] and VC-1 [49] use self-supervised learning on image or video datasets, while EmbCLIP [37], R3M [51] and Voltron [36] further perform cross-modal pre-training based on videos with aligned language descriptions. While these 2D representations have demonstrated promising performance, they still fall short in addressing occlusions in complex cluttered scenes [79] and accurately predicting robotic actions [7] in the 3D world.

Recently, growing research attention has been paid to 3D visual representations for robotics. A majority of approaches train 3D-based models from scratch [7, 43, 69], potentially losing the generalization ability. Several recent works lift pre-trained 2D features to the 3D space [20, 34, 56, 89], which compromise efficiency due to processing multi-view images and do not fully take advantage of 3D data. To improve 3D representation learning, prior endeavours propose self-supervised pre-training of 3D models [44, 53, 59, 86]. Pre-training in existing work, however, is typically limited to single objects and complete point clouds, hence, ignoring

self-occlusions and clutter in 3D images of real scenes. In addition, these 3D representations mainly focus on learning geometry and semantic meaning of objects and ignore object affordances which is important for robotics manipulation.

To enhance the capability of 3D representation in robotics, we propose SUGAR - a novel pre-training framework that learns semantics, geometry and affordance properties of objects in 3D point clouds, as illustrated in Figure 1. We automatically construct cluttered scenes with multiple objects using large-scale 3D datasets for pre-training, where the object semantics, locations, and grasping poses can be obtained for free in simulation. To jointly train multiple properties, we propose a versatile transformer-based model comprising a point cloud encoder and a prompt-based decoder. We use masked point modeling and cross-modal knowledge distillation tasks to train representations for geometry and semantic understanding respectively. In order to better understand objects and their spatial relations in cluttered scenes, we apply 3D instance segmentation and referring expression grounding tasks in pre-training. Furthermore, a grasping pose synthesis task is proposed to enable the learning of object affordance in cluttered scenes, which is high-relevant to robotic manipulation. We adopt curriculum learning to progressively train SUGAR on single- and multi-object scenes.

We address three downstream tasks for a comprehensive robotic-related evaluation of SUGAR. The first task is zero-shot 3D object recognition [44], a benchmark task for 3D shape understanding; the second task is referring expression grounding [46, 79] in cluttered scenes, which serves as a precursor for interactive robot grasping [46]; the last but the most important evaluation is language-guided robotic manipulation [31] that aims to learn a unified policy for multiple robotic tasks (see bottom of Figure 1). Experimental results show that SUGAR significantly outperforms models trained from scratch and previous pre-trained models [36, 51, 61], demonstrating the importance of 3D pre-training in cluttered scenes and learning object affordances for robotics. In summary, the contributions of our work are three-fold:

- We present SUGAR - a framework with versatile transformer architecture for 3D point cloud representation learning on cluttered scenes.
- We pre-train SUGAR on five tasks, namely masked point modeling, cross-modal learning, grasping pose synthesis, instance segmentation and object grounding, enabling to learn semantics, geometry, and affordance of objects.
- We experimentally demonstrate that SUGAR outperforms the state of the art on three robotic-related tasks including zero-shot object recognition, referring expression grounding and language-guided robotic manipulation.

2. Related Work

Visual representation learning for robotics is a fundamental yet challenging problem. To learn the representa-

tion, many approaches [2, 4, 33, 63] rely on in-domain data from target robotic tasks such as real robot demonstrations. While significant efforts have been made to collect more real robot data [2, 76, 77], the scalability of such data is still constrained by its cost. To alleviate the data scarcity issue, various techniques have been explored including data augmentation [2, 39, 40, 85], self-supervised learning [41, 54, 63], and the integration of task-specific information [35, 87]. More recently, thanks to the advancement in pre-training with large-scale internet data [27, 61, 72], a number of works [9, 36, 37, 48, 49, 51, 55, 62, 67, 82] have showcased the benefits of applying pre-trained visual representations to the robotic domain such as features learned on ImageNet [11], videos of human performing everyday tasks [21, 22], and CLIP [61] features. In particular, Voltron [36] demonstrates the advantage of incorporating aligned language descriptions to the visual representation learning for language-driven robotic tasks. Nevertheless, these approaches predominantly pre-train 2D visual representations, which can be sub-optimal for robotics in the 3D world. In this work, we focus on pre-training a 3D representation for robotic-related tasks.

3D point cloud understanding has received increased attentions with the rapid development of 3D sensors [25]. A variety of neural network architectures have been developed to effectively process 3D point cloud data such as voxel-based networks [65], Graph CNNs [80] and PointNet series [57, 58, 60]. As self-attention in transformers [75] is fundamentally a set operator, transformer-based models [24, 90] have gained popularity for unordered points and achieved superior performance in 3D object recognition [44], scene segmentation [66] and so on. The transformer architecture also makes it easy to pre-train by self-supervised masked point modeling [53, 86] or cross-modal learning from images and texts [14, 44, 83, 84, 88]. ReCon [59] further proposes a new transformer architecture to benefit from multiple pre-training tasks for point cloud understanding. However, these works only pre-train 3D representation on complete point clouds of single objects, limiting the generalization ability to more complex scenes. Ponder [28] is a pioneer work to pre-train point clouds of indoor scenes using neural rendering, but it ignores the semantic information in representation learning. Furthermore, none of the above works considers affordance learning for 3D objects which has particular importance for robotic manipulation. Our proposed approach SUGAR is designed to address these limitations by jointly pre-training on five diverse tasks in multi-object settings.

Robotic manipulation refers to the control of environment through selective contacts by the agent [50]. This task requires rich perceptual and common-sense understanding of objects around the robot. Our work concentrates on two important manipulation problems: object grasping - a fundamental skill for robotic manipulation [38, 52], and

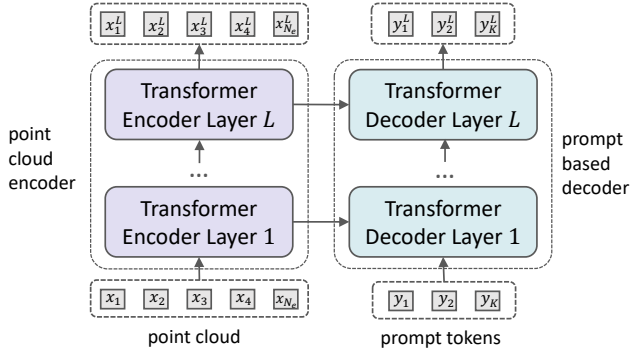


Figure 2. Network architecture of SUGAR. It consists of a point cloud encoder to generate point embeddings and a prompt-based decoder that takes task-specific prompt tokens and layer-wise connections to point embeddings to obtain prompt embeddings.

language-guided robotic manipulation [3, 47, 70, 92] which enables convenient human-robot interaction and skill generalization across tasks. Many deep learning based grasping methods [42, 73] are based on 3D point clouds due to its superiority to address visual occlusions and good sim-to-real transfer performance. Training such models has been facilitated by large-scale grasping datasets, for example, GraspNet-1Billion [18] contains real RGB-D scans and simulated grasp poses and ACRONYM [16] is composed of synthetic scenes and grasps verified in physical simulation. For language-guided robotic manipulation, though 3D visual representations have recently been explored [7, 32, 69], most existing methods still rely on 2D visual representations [4, 15, 20, 64, 68] in order to benefit from pre-trained vision-and-language models. To the best of our knowledge, our work is the first to explore large-scale 3D pre-training techniques to improve robotic manipulation.

3. SUGAR: 3D Pre-training for Robotics

We propose SUGAR, a pre-training framework for 3D point clouds which learns semantic, geometric and affordance properties of objects during pre-training. We first introduce the network architecture in Section 3.1 and then describe the self-supervised pre-training tasks in Section 3.2. In Section 3.3, we present the construction of pre-training datasets, followed by implementation details in Section 3.4.

3.1. Network Architecture

Figure 2 illustrates our network architecture, which is a versatile framework to solve multiple tasks given the input point cloud and task-specific prompts. Suppose $X = \{x_i\}_{i=1}^N$ is a point cloud where $x_i \in \mathbb{R}^6$ is composed of 3D coordinates and RGB colors and N is the number of points, and $Y = \{y_i\}_{i=1}^K$ is a sequence of prompt tokens (explained in Section 3.2). The model consists of a point cloud encoder $\mathcal{E}(X)$ to generate point embeddings and a prompt-based de-

coder $\mathcal{D}(\mathcal{E}(X), Y)$ to obtain task-specific embeddings, both of which utilizes transformer blocks [75]. We present the details of the two modules below.

Point cloud encoder. Given X , we first use farthest point sampling to select N_e key points and group S_e nearest points for each key point as a local point cloud. We normalize the local point cloud using the corresponding key point as the center and employ a shared PointNet [57] to encode each local point cloud into a token embedding $x_i^0 \in \mathbb{R}^d$ where d is the dimensionality of the feature. The position of the local point cloud is set as the 3D coordinates of its key point, and a feed-forward network (FFN) is utilized to obtain the position embedding $x_i^p \in \mathbb{R}^d$. We use a standard transformer model with L layers to encode the point cloud tokens. The computation at the l -th layer is:

$$\{x_i^l\}_{i=1}^{N_e} = \text{FFN}(\text{SA}(\{x_i^{l-1} + x_i^p\}_{i=1}^{N_e})), \quad (1)$$

where SA is the self-attention operator. We omit the residual and layer normalization for simplicity. Please refer to the transformer paper [75] for details.

Prompt-based decoder. Given the task-specific prompt tokens Y , we use a linear layer to project them into token embeddings $\{y_i^0\}_{i=1}^K, y_i^0 \in \mathbb{R}^d$ where the linear layer is different for each task. The decoder consists of the same number of blocks of self-attention (SA) and cross-attention (CA) layers as the encoder to layer-wisely query the encoded point embeddings and update the prompt tokens:

$$\{\hat{y}_i^l\}_{i=1}^K = \text{FFN}(\text{SA}(\{y_i^{l-1}\}_{i=1}^K)), \quad (2)$$

$$\{y_i^l\}_{i=1}^K = \text{FFN}(\text{CA}(\{\hat{y}_i^l\}_{i=1}^K, \{x_i^l\}_{i=1}^{N_e})). \quad (3)$$

The output embeddings $\{y_i^L\}_{i=1}^K$ can then be used for each specific task described below.

3.2. Pre-training Tasks

In this section, we describe the five pre-training tasks that empowers SUGAR to recognize, segment and grasp objects in 3D point clouds and, hence, to be a state-of-the-art 3D representation for robotic manipulation. Figure 3 (left) illustrates all the pre-training tasks.

Masked point modeling (MPM). Masked modeling is a general self-supervised task in many different domains such as texts [13], images [27], videos [72] and 3D point clouds [53, 59, 86]. It enables learning geometry structures by masking a fraction of points and reconstructing the original point cloud. Specifically, we randomly mask out 60% of local point cloud tokens following the best practice in [59] and only feed the unmasked tokens to the point cloud encoder \mathcal{E} . We then reconstruct the masked tokens via a light-weight point cloud decoder which is a 4-layer plain transformer. The decoder is fed with unmasked point embeddings $\{x_i^L\}$ and a special [mask] embedding added to the corresponding position embedding for each masked token. The output

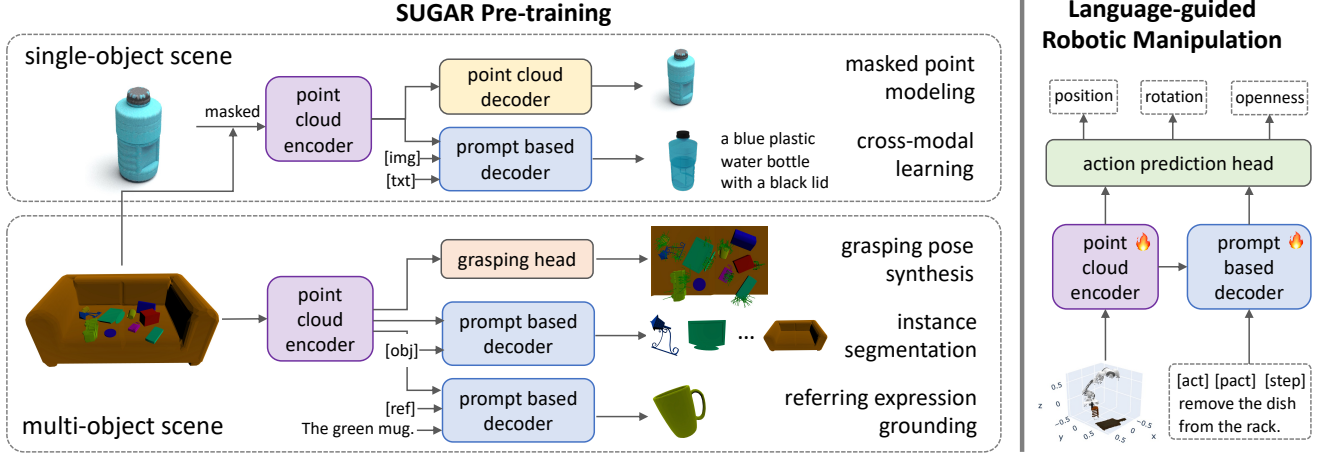


Figure 3. Left: Five pre-training tasks for SUGAR using single- and multi-object scenes. The modules of the same color are shared. Right: The pre-trained point cloud encoder and prompt-based decoder are finetuned on the downstream task of robotic manipulation.

embeddings of the masked tokens are utilized to predict 3D coordinates of each point in the local point cloud. We further include color prediction for each point to enhance the learning of textures. Assume $\hat{X}_j \in \mathbb{R}^{S_e \times 6}$ and $X_j \in \mathbb{R}^{S_e \times 6}$ are the predicted and groundtruth local point cloud respectively, we extend the original reconstruction loss l_2 Chamfer Distance [17] to also consider colors:

$$L_{mpm} = \frac{1}{N_e S_e} \sum_{j=1}^{N_e} \left(\sum_{(\hat{x}, x) \in A_j} \|\hat{x} - x\|_2^2 + \sum_{(x, \hat{x}) \in B_j} \|x - \hat{x}\|_2^2 \right)$$

where

$$A_j = \{(\hat{x}, x) | \hat{x} \in \hat{X}_j, x \in X_j, x = \arg \min_{x \in X_j} \|\hat{x}_{:3} - x_{:3}\|_2^2\}, \quad (4)$$

$$B_j = \{(x, \hat{x}) | x \in X_j, \hat{x} \in \hat{X}_j, \hat{x} = \arg \min_{\hat{x} \in \hat{X}_j} \|x_{:3} - \hat{x}_{:3}\|_2^2\}.$$

The $x_{:3}$ denotes the first three dimension of the vector x , which is the 3D coordinates of the point.

Cross-modal learning (CML). As pre-trained image and text models have achieved great success in representation learning, it is beneficial to distill knowledge from these models to train the 3D visual representation [44, 59, 83, 84]. Suppose we have R^I images and R^T text descriptions aligned with a point cloud X , for instance, by projecting X from 3D to 2D using different camera poses and captioning the 2D images. We use pre-trained image and text models to extract image and text features as $\{f_r^I\}_{r=1}^{R^I}$ and $\{f_r^T\}_{r=1}^{R^T}$ respectively. Our goal is to extract point cloud features from X that are close to the aligned image or text features. For this purpose, we leverage two prompt tokens `[img]` and `[txt]` as input to the decoder \mathcal{D} and then project the two output embeddings via a linear layer to the same space as the pre-trained image or text features, denoted as \hat{f}^I and \hat{f}^T . A smoothed l_1 loss L_{cml} is used to minimize the distance between \hat{f}^I and each f_r^I , as well as between \hat{f}^T and each f_r^T for cross-modal knowledge distillation.

Grasping pose synthesis (GPS). Object grasping is a precursor for many robotic applications and thus we treat it as a fundamental skill to learn for the visual representation. It is however challenging to predict diverse grasping poses through a deterministic model. To simply the problem, we make an assumption that there exists at most one optimal grasping pose for each point in X , where $g_i^v \in \{0, 1\}$ denotes if the point has a valid grasping pose and $g_i^m \in \mathbb{R}^{4 \times 4}$ is the optimal grasping pose for the point if $g_i^v = 1$. Given point embeddings $\{x_i^L\}_{i=1}^{N_e}$ from \mathcal{E} , we first use tricubic interpolation to upsample features for each point $\{x_i^L\}_{i=1}^N$. Then we use a linear layer to get the binary prediction $\hat{g}_i^v \in \mathbb{R}$ given x_i^L . For valid points, we predict the relative position of the optimal grasping pose to the point as $\hat{g}_i^p \in \mathbb{R}^3$ and the 6D representation $\hat{g}_i^r \in \mathbb{R}^6$ for 3D rotation [91], which can form grasping pose $\hat{g}_i^m \in \mathbb{R}^{4 \times 4}$ using Gram-Schmidt process. The training objective consists of a binary cross-entropy loss (BCE) and a l_2 distance for the grasping pose:

$$L_{gps} = \frac{1}{N} \sum_{i=1}^N \text{BCE}(\hat{g}_i^v, g_i^v) + g_i^v \|\hat{g}_i^m - g_i^m\|_2^2. \quad (5)$$

Instance segmentation (INS). Segmenting 3D objects is a key ability to understand cluttered scenes. To address the task, we use a set of object tokens (`[obj1]`, \dots , `[objK]`) as the prompt tokens to the decoder \mathcal{D}_θ and obtain output embeddings $\{y_i^L\}_{i=1}^K$. For each object token y_i , we measure its similarity with the upsampled point embeddings $s(y_i, x_j) = y_i^L \cdot x_j^L$ and thus generate the instance segmentation mask $m_i \in \mathbb{R}^N$, $m_{ij} = \sigma(s(y_i, x_j)) > 0.5$ where σ is the sigmoid function. In addition, we use linear layers to predict an objectiveness score given y_i^L and image and text features similar to that in CML task. To train the model, we take the approach in DETR [5] to use Hungarian matching for groundtruth assignment, and then given the best matches we compute a combined loss including a BCE loss and DICE

loss [71] for instance segmentation, a BCE loss for objectiveness score prediction and a smoothed l_1 loss for cross-modal learning on the object level.

Referring expression grounding (REG). The task aims to segment an object described by a natural language sentence in cluttered scenes. We use a prompt token `[ref]` and a sequence of encoded text tokens by a pre-trained language model to \mathcal{D} , and predict the object mask using the output embedding of `[ref]` similar to the INS task. We combine the BCE loss and DICE loss for training.

3.3. Pre-training Data

We first introduce the single-object datasets and then describe our automatic constructed multi-object dataset.

Single-object dataset. ShapeNet [6] is a commonly used dataset for 3D point cloud pre-training [53, 59, 86], containing 55 categories and about 52K instances. To scale up the pre-training dataset, we follow OpenShape [44] to ensemble ShapeNet, 3D-FUTURE [19] (16.5K instances), ABO [8] (8.0K instances) and Objaverse [10] (no LVIS split, 752.2K instances). We use the same point clouds, aligned images and text descriptions as [44]. Specifically, the point cloud of an object is obtained by evenly sampling 10K points from the mesh surface of 3D object asset and interpolating the point colors based on the mesh texture. The aligned images are generated by rendering the object from 12 fixed views around the object, and the text descriptions are built by captioning the rendered images via existing image captioning systems, retrieving descriptions of similar images, or the metadata of the object name.

Multi-object dataset. We use Blenderproc [12] for automatic multi-object scene construction. First, we randomly select 2 to 10 objects from single-object datasets and sample their locations in a 3D space. Then, we enable the physical simulation to fall objects onto a plane with random textures to generate a realistic scene. Finally, we randomly sample camera position near the scene and let the camera point to the center of the scene for RGB-D image and segmentation label rendering. Figure 1 top left shows an example of the generated data. The point cloud and per-point labels can be obtained by projecting the 2D image to 3D given the camera parameters. In total, we generate 48.9K multi-object scenes using objects from ShapeNet, and 62.8K scenes using objects from Objaverse (no LVIS) dataset.

Grasping pose generation. It is expensive to automatically generate high-quality grasping poses requiring pose sampling and verification in physical simulation [16]. However, recent works [18, 78] have shown that having diverse grasp poses per objects is more important than scale to generalize to various objects. Therefore, we re-use the existing grasping dataset ACRONYM [16] which contains 2K physically verified grasping poses per object for around 8K objects in ShapeNet. We generate 62.7K multi-object scenes follow-

ing [16] and filter invalid grasp poses with collisions in the scene. For each point, we select the nearest grasping pose to the point as the optimal grasping pose; there is no optimal pose if the nearest distance is above a certain threshold.

3.4. Pre-training Details

Training strategy. We adopt a curriculum learning scheme to pre-train the 3D representations as it is more challenging to understand the multi-object scenes than single objects. We first pre-train the model on the single-object datasets with the MPM and CML tasks as show in the top row of Figure 3 (left); then we joint train on both single- and multi-object datasets using all the five pre-training tasks.

Implementation details. We set the number of points $N = 4096$, the number of key points $N_e = 256$ and the group size $S_e = 32$. Due to the computational cost, we only adopt a small model size with $d = 384$, $L = 12$. In the CML task, OpenCLIP ViT-bigG-14 [30] is used to extract image and text features. For the multi-object dataset, the plane background is kept in the point cloud 50% of the time during training. We pre-train two sets of models according to the pre-training data: ‘SN’ uses objects only in ShapeNet, and ‘Ens’ uses the ensembled four datasets. More training details are presented in the supplementary material.

4. Evaluation on Robotic-related Tasks

To thoroughly evaluate the pre-trained representation, we resort to three robotic-related tasks including zero-shot object recognition, referring expression grounding, and language-guided robotic manipulation. We present datasets, downstream adaptation and quantitative results for each task in the following three sections.

4.1. Zero-shot Object Recognition

The task aims to classify unseen 3D objects without training on those specific categories. It evaluates the generalization ability of visual representations for semantic understanding.

Datasets. We use three 3D object recognition benchmarks including ModelNet40 [81], ScanObjectNN [74] and Objaverse-LVIS [10]. ModelNet40 contains 40 categories and 2,468 objects in the test split. The objects are synthetic 3D models without colors. ScanObjectNN is one of the most challenging 3D datasets, consisting of 15 common categories and 587 real-world 3D scans in the test split. There are three evaluation setups with increasing levels of difficulty: 1) OBJ_ONLY which only includes ground truth segmented objects; 2) OBJ_BG where objects are with background data; 3) PB_T50_RS with 2,935 testing examples where perturbations are applied to the ground truth object bounding boxes to extract the point cloud. The scanned objects contain colors. Objaverse-LVIS [10] is an annotated subset of the large-scale Objaverse dataset and comprises 46,205 shapes among 1,156

Table 1. Zero-shot object recognition performance on three benchmarks. The Top1 accuracy is reported if not specified otherwise. The blue colored results in brackets on the ScanObjectNN dataset are obtained using colored point clouds.

Pretrain data	Method	ModelNet40	ScanObjectNN			Objaverse-LVIS		
			OBJ_ONLY	OBJ_BG	PB_T50_RS	Top1	Top3	Top5
ShapeNet	ReCon [59]	61.2	39.6	38.0	29.5	1.1	2.7	3.7
	CLIP2Point [29]	49.5	35.5	30.5	23.3	2.7	5.8	7.9
	ULIP-PointBERT [83]	60.4	49.9	-	-	6.2	13.6	17.9
	OpenShape-PointBERT [44] ¹	70.3	51.8 (52.0)	41.9 (42.4)	28.5 (28.6)	10.8	20.2	25.0
	SUGAR (single)	71.2	48.6 (52.8)	46.3 (50.3)	33.4 (35.4)	13.3	22.6	27.3
	SUGAR (multi)	66.5	53.5 (55.0)	47.9 (50.8)	34.2 (36.5)	12.1	20.1	25.1
Ensembled (no LVIS)	ULIP-PointBERT [83]	71.4	46.0	-	-	21.4	38.1	46.0
	OpenShape-PointBERT [44] ¹	85.3	50.8 (52.0)	51.4 (52.6)	39.4 (40.3)	39.1	60.8	68.9
	SUGAR (single)	84.3	49.6 (65.3)	56.2 (68.0)	41.8 (49.3)	42.1	64.6	72.1
	SUGAR (multi)	83.8	53.2 (63.5)	53.2 (65.6)	38.2 (48.7)	39.4	61.6	69.3

LVIS [26] categories. The objects are crawled from Internet, containing both synthetic 3D models and real-world 3D scans. Colors are included in the point cloud.

Evaluation metrics. We use Top1 classification accuracy as the main evaluation metric, and also report Top3 and Top5 for the Objaverse dataset to compare with previous work.

Downstream adaptation. We randomly sample 4,096 points for each object and set RGB values as -0.2 (gray color) if there is no color in the point cloud. As in the CML pre-training task, we use `[img]` and `[txt]` prompt tokens to extract point cloud features that are in the same space of the pre-trained image and text features. The two predicted features are fused to obtain the final representation, which is used to perform KNN classification with the pre-trained text features of object categories.

Results. Table 1 presents the performance on the test split of the three benchmarks. In the top block, all the compared methods and SUGAR (single) are trained on single objects in ShapeNet, while SUGAR (multi) is further trained on multi-object scenes constructed from ShapeNet objects. Our SUGAR (single) achieves comparable performance with the state-of-the-art method OpenShape [44] on ModelNet40 and ScanObjectNN without colors, and significantly outperforms OpenShape on Objaverse-LVIS and ScanObjectNN with colors, *e.g.*, +7.9% on OBJ_BG split. As OpenShape randomly discards all point cloud colors during pre-training, it sacrifices the ability of texture recognition but improves the results for uncolored point clouds. Our model however learns to reconstruct both geometry structures and colors in masked point modeling, taking advantage of both geometric and texture information for semantic recognition. SUGAR (multi) deteriorates the performance on ModelNet40 and Objaverse-LVIS datasets where the point clouds are complete, but performs better on the ScanObjectNN dataset with real scanned

objects. This is because the multi-object dataset captures more realistic scenes with occlusions. Furthermore, scaling up the pre-training data significantly improves the zero-shot recognition performance as shown in the bottom block of Table 1. Since the Objaverse dataset contains both synthetic 3D objects and real scans, the performance on ScanObjectNN is boosted by a large margin. SUGAR (single) achieves 68.0% Top1 accuracy on OBJ_BG split of ScanObjectNN, outperforming state of the art by 15.4%. However, we observe around 3% decrease in SUGAR (multi) compared to SUGAR (single). We hypothesize two reasons for the performance drop. First, we only use a small transformer model which may not have sufficient capacity to jointly solve the five pre-training tasks when the pre-training data increases. Second, there are multi-object scenes in the crawled Objaverse dataset such as multi-level buildings, hence, construction of new scenes using 3D assets in Objaverse may require additional treatment, which we leave to future work. For completeness of comparison with prior work [44], we include results of pre-training on Ensembled with LVIS dataset in the supplementary material, which show similar trend.

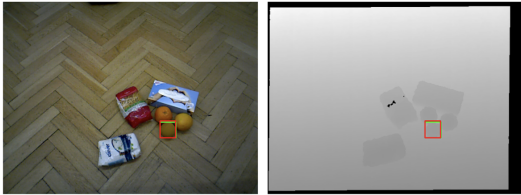
4.2. Referring Expression Grounding

Given a natural language description of an object, the task is to segment the target object in the 3D point cloud of cluttered scenes. Solutions to this task require semantic scene understanding and spatial relation reasoning.

Datasets. We use the OCID-Ref [79] and RoboRefit [46] datasets, which are representative of robotic manipulation scenes. OCID-Ref is collected in clean lab environments and consists of 58 object categories, 2,298 RGB-D images and 259,839 referring expressions for training. It has 18,342 and 27,513 referring expressions in validation and test splits respectively. However, all the validation and test images appear in the training split. This makes it impossible to evaluate the generalization ability for unseen scenes and

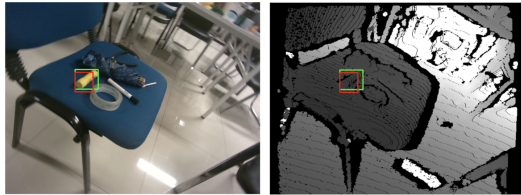
¹We re-run OpenShape on the ScanObjectNN as the reported number in OpenShape [44] is on the whole dataset of the OBJ_ONLY setup.

Sentence: the ball on the front right.



(a) An example in the test split of OCID-Ref dataset.

Sentence: will you please pass me the glue stick



(b) An example in the testB split of RoboRefit dataset.

Figure 4. Referring expression examples on the OCID-Ref and RoboRefit dataset. The green bounding box is the groundtruth annotation, and the red bounding box is predicted by our SUGAR model. RoboRefit contains natural scenes and noisy depth observations.

Table 2. The results of Acc@0.25 for referring expression detection on the testing split of OCIF-Ref dataset.

Method	Total	Clutter level		
		Min	Med	Max
R3M [51]	63.30	63.87	68.34	55.33
MVP [62]	49.58	50.98	53.83	41.94
CLIP [61]	68.35	67.01	76.61	60.33
V-Cond [36]	90.77	87.56	96.58	90.17
SUGAR (Ens_m)	97.74	98.52	97.75	96.68

objects. In contrast, RoboRefit contains more natural and diverse scenes captured by noisy RGB-D cameras. There are 7,929 RGB-D images and 36,915 referring expressions in the training split. Two test splits are used for evaluation: testA shares similar scenes to the training split with 1,859 scenes and 8,523 sentences; scenes and objects of testB with 1,083 RGB-D images and 5,320 sentences are different from training. Figure 4 shows examples of the two datasets.

Evaluation metrics. Previous works [36, 46, 79] all evaluate in the 2D domain. For fair comparison, we first predict a 3D segmentation mask given the point cloud and then project it in 2D. The major metric in the OCID-Ref dataset is Acc@0.25 which is the ratio of predicted 2D bounding boxes that have IoU larger than 0.25 with the groundtruth. RoboRefit uses Acc@0.5 for 2D object detection and mIoU for segmentation which is the mean IoU between the predicted and groundtruth segmentation masks.

Downstream adaptation. We finetune SUGAR similar to the pre-training REG task for 3D segmentation mask prediction, except that we increase the number of key points N_e to improve precision. We set $N_e = 512$ for OCID-Ref and 1536 for RoboRefit if not stated otherwise. As the depth sensor in RoboRefit is noisy, we automatically remove outliers [1] to clean 3D object masks in training and evaluation.

Experimental results. Table 2 presents the results for the OCID-Ref dataset. For fair comparison with previous work [36], we fix the visual encoder and only finetune the decoder in SUGAR pre-trained on the ensembled multi-object dataset (Ens_m). Results show that SUGAR outperforms the state-of-the-art 2D visual representations.

RoboRefit is a more challenging and realistic dataset. We

Table 3. Performance of referring expression detection (evaluated by Acc@0.5) and referring expression segmentation (evaluated by mIoU) on the RoboRefit dataset. We use $N_e = 1536$ for our SUGAR models if not stated otherwise.

Method	testA		testB	
	Acc@0.5	mIoU	Acc@0.5	mIoU
RefTR (r50) [46]	84.22	81.16	54.12	52.98
RefTR (r101) [46]	81.19	78.07	45.68	49.75
SUGAR (no pre-train, $N_e=768$)	87.30	79.02	50.64	51.49
SUGAR (no pre-train)	87.56	81.31	55.62	57.02
SUGAR (SN_s)	88.02	81.84	52.90	56.80
SUGAR (SN_m w/o grasp)	88.66	81.42	59.70	59.76
SUGAR (SN_m)	89.05	81.75	61.85	60.53
SUGAR (Ens_m)	89.47	82.11	65.04	62.80

compare with state-of-the-art transformer models proposed in [46] based on RGB-D images for fair comparison with SUGAR using colored point cloud input. The results are presented in Table 3. First, we can see that the number of point cloud tokens matters a lot from the first two rows in the SUGAR variants. The task requires high resolution point cloud embeddings. Our SUGAR architecture trained from scratch achieves comparable or better performance than the state-of-the-art methods. Pre-training in single-object datasets - SUGAR (SN_s), does not benefit the object grounding in cluttered scene on the more difficult testB split. SUGAR (SN_m) pre-trained on ShapeNet multi-object dataset improves the performance of SUGAR w/o pre-training by around 6% on testB Acc@0.5. Interestingly, we find that learning the object affordances is beneficial for referring expression grounding, see comparison to (SN_m w/o grasping). SUGAR (Ens_m) pre-trained on a larger dataset achieves the best performance and outperforms previous best Transformer (r50) model in [46] by more than 10% on the challenging testB split on Acc@0.5, demonstrating the generalization ability of our SUGAR representation.

4.3. Language-guided Robotic Manipulation

This task aims to train a policy that can follow natural language instruction to perform manipulation tasks. Our evaluation in this section is focused on the performance of multi-

Table 4. Success rates of multi-task policies on 10 tasks of RLbench simulator.

Method	Pre-train	Avg.	Pick & Lift	Pick-Up Cup	Push Button	Put Knife	Put Money	Reach Target	Slide Block	Stack Wine	Take Money	Take Umbrella
Auto- λ [23]	-	69.3	87	78	95	31	62	100	77	19	64	80
Hiveformer [23]	-	83.3	88.9	92.9	100	75.3	58.2	100	78.7	71.2	79.1	89.2
Hiveformer	R3M [51]	88.5	89.6	87.0	79.2	82.6	87.0	100	91.6	90.2	81.8	95.6
Hiveformer	CLIP [61]	87.2	87.2	96.0	68.8	68.8	94.8	100	92.4	93.0	73.2	97.6
PolarNet [7]	ShapeNetPart	89.8	97.8	86.0	99.6	80.5	94.1	100	93.4	80.5	68.1	97.8
SUGAR	-	85.9	77.7	92.7	91.7	69.4	87.7	99.7	94.3	83.1	66.8	95.7
	SN_s	88.1	95.4	96.2	97.6	61.6	81.2	100	92.0	94.0	66.0	97.0
	SN_m w/o grasp	91.9	94.9	94.1	90.9	83.9	92.5	100	97.0	96.2	72.3	97.0
	SN_m	93.0	93.1	94.5	98.9	85.4	97.8	100	97.9	94.5	70.0	98.4
	Ens_m w/o grasp	92.0	93.1	93.7	98.8	85.5	92.3	99.9	97.3	93.7	68.8	97.2
	Ens_m	93.0	95.8	95.7	96.1	86.5	94.2	100	97.0	93.5	72.0	98.8

task policies and sample-efficient learning.

Datasets. We evaluate models on the 10-task benchmark in the RLbench [31] simulator following previous work [7, 23, 45]. The task names are listed in Table 4. For each task, we use 100 demonstrations for behavior cloning. Each demonstration consists of a sequence of keysteps of RGB-D image observations from three cameras and a 7-DoF action denoting the position, rotation and openness state of the gripper. The policy is required to predict keystone actions which are then executed by a motion planner. More details are provided in the supplementary material.

Evaluation metric. The policy is evaluated by success rate. An episode achieves a score of 0 or 1 without any partial credits. We perform three runs and report the average [7, 23].

Downstream adaptation. We feed different prompt tokens into the decoder of SUGAR for action prediction, namely a masked current action token [act], a previous action token [pact], a step id [step] and the language textual tokens. The output embeddings together with point embeddings from the encoder are used to predict actions via fully connected layers. More details are in the supplementary material.

Experimental results. Table 4 shows the results of multi-task policies on the 10 tasks. We compare with the state-of-the-art methods and also improve the 2D-based method Hiveformer [23] with pre-trained image backbones including R3M [51] and CLIP [61]. The image backbone is also fine-tuned end-to-end for action prediction otherwise the model performs poorly due to the large visual domain gap. Our model based on pre-trained SUGAR outperforms the 2D pre-trained models, PolarNet, and 3D models without pre-training. We also observe performance improvement with grasping prediction and multi-object scene in pre-training. As there are sufficient training data for the policy, we do not observe further improvement using SUGAR (Ens_m). However, when we reduce the training data to 10 demonstrations per task, there is a clear advantage of Ens_m representation compared to the SN_m as shown in Figure 5. SUGAR

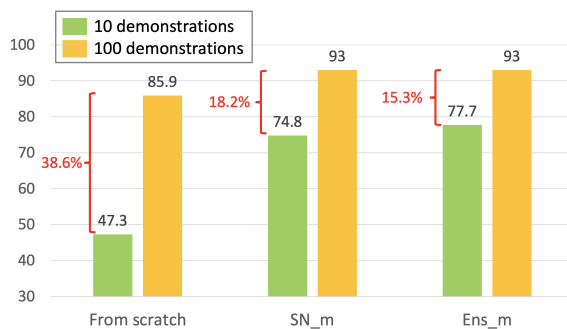


Figure 5. Performance of training with 10 demonstrations.

(Ens_m) significantly boosts the performance of the model trained from scratch with over 30% improvement. We further provide results on a real robot in the supplementary material and show the improvement from the pre-trained 3D representation for robot learning in real world.

5. Conclusion

This work presents SUGAR, a novel 3D pre-training framework for robotics. It employs a versatile transformer-based architecture that jointly supports five pre-training tasks to learn semantic, geometric and affordances properties of objects in cluttered scenes. Experimental results demonstrate the excellent performance when using SUGAR for three robotic-related tasks, namely, zero-shot 3D object recognition, referring expression grounding, and language-driven robotic manipulation. Our work emphasizes the importance of cluttered scenes and object affordances when pretraining 3D representations for robotic applications.

Acknowledgements. This work was partially supported by the HPC resources from GENCI-IDRIS (Grant 20XX-AD011012122). It was funded in part by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute), the ANR project VideoPredict (ANR-21-FAI1-0002-01) and by Louis Vuitton ENS Chair on Artificial Intelligence.

References

- [1] Open3d point cloud outlier removal. http://www.open3d.org/docs/latest/tutorial/Advanced/pointcloud_outlier_removal.html, 2024. 7
- [2] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023. 1, 2
- [3] brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In *CoLR*, 2022. 3
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2, 3
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4
- [6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5
- [7] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023. 1, 3, 8, 13, 15
- [8] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21126–21136, 2022. 5
- [9] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. *arXiv preprint arXiv:2310.09289*, 2023. 2
- [10] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 5
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2
- [12] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 5
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3
- [14] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022. 2
- [15] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 3
- [16] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021. 3, 5
- [17] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 4
- [18] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020. 3, 5
- [19] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021. 5
- [20] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023. 1, 3
- [21] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The” something something” video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 2
- [22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, 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*, pages 18995–19012, 2022. [2](#)
- [23] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR, 2023. [8](#), [13](#)
- [24] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021. [2](#)
- [25] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020. [2](#)
- [26] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. [6](#)
- [27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. [2](#), [3](#)
- [28] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16089–16098, 2023. [2](#)
- [29] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023. [6](#)
- [30] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. [5](#)
- [31] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. [2](#), [8](#), [13](#)
- [32] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *CVPR*, 2022. [3](#)
- [33] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Fredrik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. [2](#)
- [34] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. [1](#)
- [35] Rico Jonschkowski and Oliver Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39:407–428, 2015. [2](#)
- [36] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023. [1](#), [2](#), [7](#)
- [37] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022. [1](#), [2](#)
- [38] Kilian Kleeburger, Richard Bormann, Werner Kraus, and Marco F Huber. A survey on learning-based robotic grasping. *Current Robotics Reports*, 1:239–249, 2020. [2](#)
- [39] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020. [1](#), [2](#)
- [40] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020. [2](#)
- [41] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *ICML*, pages 5639–5650. PMLR, 2020. [2](#)
- [42] Yiming Li, Tao Kong, Ruihang Chu, Yifeng Li, Peng Wang, and Lei Li. Simultaneous semantic and collision learning for 6-dof grasp pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3571–3578. IEEE, 2021. [3](#)
- [43] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022. [1](#)
- [44] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yin hao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *arXiv preprint arXiv:2305.10764*, 2023. [1](#), [2](#), [4](#), [5](#), [6](#), [13](#), [14](#)
- [45] Shikun Liu, Stephen James, Andrew J Davison, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *arXiv preprint arXiv:2202.03091*, 2022. [8](#), [13](#)
- [46] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. VI-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. *arXiv preprint arXiv:2308.00640*, 2023. [2](#), [6](#), [7](#)
- [47] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. In *NeurIPS, 5th Robot Learning Workshop: Trustworthy Robotics*, 2022. [3](#)
- [48] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. [1](#), [2](#)
- [49] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan

- Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023. 1, 2
- [50] Matthew T Mason. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1: 1–28, 2018. 2
- [51] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2022. 1, 2, 7, 8
- [52] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, et al. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 2023. 2
- [53] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 1, 2, 3, 5
- [54] Jyothishh Pari, Nur Muhammad Shafiqullah, Sridhar Pandian Arunachalam, and Lrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 2
- [55] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022. 2
- [56] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 1
- [57] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3
- [58] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2
- [59] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. *arXiv preprint arXiv:2302.02318*, 2023. 1, 2, 3, 4, 5, 6
- [60] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 2
- [61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 7, 8
- [62] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2022. 1, 2, 7
- [63] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. *arXiv preprint arXiv:2306.10007*, 2023. 2
- [64] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maroon, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *TMLR*, 2022. 3
- [65] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2
- [66] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 2
- [67] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021. 2
- [68] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2021. 3
- [69] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2022. 1, 3
- [70] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020. 3
- [71] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017. 5
- [72] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019. 2, 3
- [73] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. 3

- [74] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 5
- [75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3
- [76] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023. 1, 2
- [77] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, et al. Bridgedata v2: A dataset for robot learning at scale. *arXiv preprint arXiv:2308.12952*, 2023. 1, 2
- [78] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023. 5
- [79] Ke-Jyun Wang, Yun-Hsuan Liu, Hung-Ting Su, Jen-Wei Wang, Yu-Siang Wang, Winston Hsu, and Wen-Chin Chen. Occl-ref: A 3d robotic dataset with embodied language for clutter scene grounding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5333–5338, 2021. 1, 2, 6, 7
- [80] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2
- [81] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5
- [82] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 2
- [83] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1179–1189, 2023. 2, 4, 6
- [84] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023. 2, 4
- [85] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspier Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. Scaling robot learning with semantically imagined experience. In *arXiv preprint arXiv:2302.11550*, 2023. 1, 2
- [86] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 1, 2, 3, 5
- [87] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020. 2
- [88] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023. 2
- [89] Tong Zhang, Yingdong Hu, Hanchen Cui, Hang Zhao, and Yang Gao. A universal semantic-geometric representation for robotic manipulation. *arXiv preprint arXiv:2306.10474*, 2023. 1
- [90] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2
- [91] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 4
- [92] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. VIOLA: Object-centric imitation learning for vision-based robot manipulation. In *CoRL*, 2022. 3

In Section A, we provide more implementation details for SUGAR pre-training and downstream adaptation. Then in Section B we present additional quantitative results. We further perform real robot experiments in Section C to demonstrate the effectiveness of SUGAR pre-training for robotic manipulation in the real world. Finally, we discuss limitations and future work in Section D.

A. Implementation Details

A.1. Pre-training

Network details. We set the number of points $N = 4096$, the number of key points $N_e = 256$ and the group size $S_e = 32$ to obtain the point cloud input tokens. The SUGAR encoder and decoder contains $L = 12$ transformer blocks with hidden size $d = 384$ and 6 attention heads per block.

Training details. We pre-train two sets of models according to the pre-training data: ‘SN’ uses objects only in ShapeNet, and ‘Ens’ uses the ensembled four datasets. For the ‘SN’ model, we train 100K iterations on the single-object dataset with learning rate $1e-4$ and 100K iterations on the multi-object dataset with learning rate $1e-5$ and batch size 128. For the ‘Ens’ model, we train 300K iterations on the single-object dataset and 200K iterations on the multi-object dataset using the same learning rate and batch size as in ‘SN’ models. The pre-training is performed on one NVIDIA-A100 GPU, taking 50 hours for the ‘SN’ model and 130 hours for the ‘Ens’ model.

A.2. Referring expression grounding

For the OCID-Ref dataset, we fix the point cloud encoder and only finetune the prompt-based decoder. We finetune the model with a batch size of 64 and learning rate of $1e-4$ for 20 epochs. For the RoboRefit dataset, we finetune the full model with a batch size of 16 and learning rate of $4e-5$ for 50 epochs. We use the AdamW optimizer with cosine learning rate scheduler.

A.3. Language-guided robotic manipulation

Experimental setup. Our experimental setup on RL-Bench [31] 10 tasks is the same as previous works [7, 23]. Specifically, we use three cameras located on the left shoulder, right shoulder and wrist of the robot with known camera intrinsics and extrinsics. Each camera produces an RGB-D image with image resolution of 128×128 at every step. A merged point cloud can be obtained given the camera parameters. Following [7], we only keep points inside the robot’s workspace by using a fixed bounding box around the table. We use voxel downsampling to uniformly downsample the point cloud with 0.5cm grid size. For robotic control, we use keysteps [7, 23, 45] - key turning points in action trajectories where the gripper changes its openness state or velocities

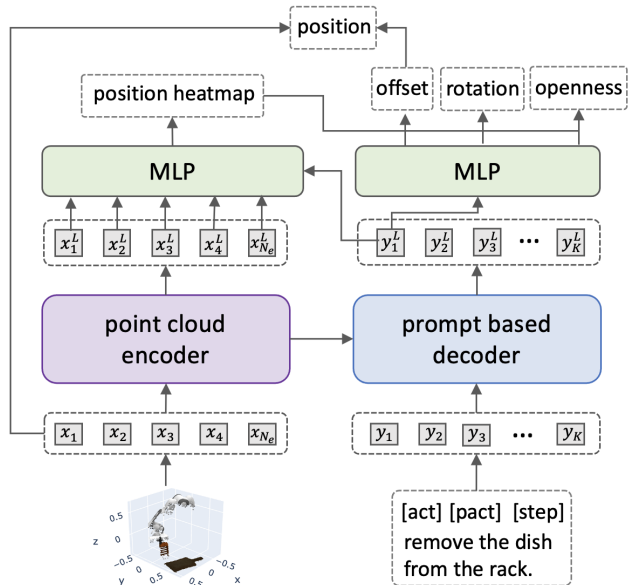


Figure 6. Network architecture for language-guided robotic manipulation. The point cloud encoder and prompt based decoder can be finetuned from SUGAR pre-training. We use two multi-layer perceptrons modules (MLP) as the action prediction head.

of joints are close to zero. The control policy should predict a position (3D), rotation (4D represented by quaternion) and openness state (1D) of the gripper for the next keystone. The default motion planner in RL-Bench is used to find a trajectory between two keysteps.

Model details. Figure 6 illustrates the policy network in detail. We first combine the action prompt embedding y_1^L and point embeddings $\{x_i^L\}_{i=1}^{N_e}$ to compute a heatmap over all the key points, which denotes the importance of the key points for action prediction. We then average the point embeddings and position of key points respectively using the heatmap. The averaged point embedding is concatenated with y_1^L to regress the position offset relative to the averaged key point position, a rotation vector and an openness state. The policy is trained by behavior cloning, with MSE loss for position and rotation, and BCE loss for openness state.

Training details. We use a batch size of 8 to train the model for 200K iterations for the 10 RL-Bench tasks. We adopt a learning rate of $5e-5$ for the model trained from scratch, while a lower learning rate of $2e-5$ for the model initialized from SUGAR pre-training.

B. Additional Results

Zero-shot object recognition. Though we consider the Ensembled w/o LVIS setup to be better suited for evaluating the generalization ability of models, we include results with LVIS training in Table 5 for complete comparison with prior work [44]. Training with LVIS split improves perfor-

Table 5. Zero-shot object recognition performance with models trained on Ensembled w/ LVIS dataset.

Method	ModelNet40	ScanObjectNN			Objaverse-LVIS		
		OBJ_ONLY	OBJ_BG	PB_T50_RS	Top1	Top3	Top5
OpenShape [44]	84.4	54.0	59.1	43.6	46.8	69.1	77.0
SUGAR (single)	84.6	65.3	67.6	49.8	49.5	72.2	78.8
SUGAR (multi)	84.5	64.9	66.8	48.3	46.8	69.7	76.6

Table 6. Performance of referring expression detection (evaluated by Acc@0.5) and referring expression segmentation (evaluated by mIoU) on the RoboRefit dataset.

Method	testA		testB	
	Acc@0.5	mIoU	Acc@0.5	mIoU
SUGAR (no pre-train)	87.56	81.31	55.62	57.02
SUGAR (Ens_s)	88.11	81.71	52.59	56.57
SUGAR (Ens_m)	89.47	82.11	65.04	62.80

mance on the LVIS dataset but does not impact much on the other two datasets. Our model still outperforms the SoTA method [44] under this setup.

Referring expression grounding. We provide an additional variant SUGAR (Ens_s) in Table 6, which is pre-trained on single objects of the ensembled dataset. To be noted, we only initialize the point cloud encoder for SUGAR variants pre-trained on single objects as we find initializing both encoder and decoder deteriorates the performance. As the decoder in single-object pre-training focuses on the overall scene for cross-modal learning, we hypothesize that the learned cross-modal attentions can suffer from recognition of local objects. As shown in Table 6, the single object pre-training on the ensembled dataset does not benefit the generalization on unseen cluttered scenes in testB split, demonstrating the importance of pre-training on multi-object scenes.

Language-guided robotic manipulation. In Table 7, we include both the averaged success rate and standard deviations for the RL Bench 10-task experiment. As the 10 RL Bench tasks use objects with simple shapes like cups and cubes, pre-training on ShapeNet can be sufficient and thus we do not observe further performance improvement from pre-training on Ens_m. Compared to PolarNet, our model performs slightly worse on Pick & Lift and Push Button though it achieves better performance on average. To be noted, PolarNet employs additional normal and height features in the point cloud, while our method omits those for generalizability in pre-training. As shown in PolarNet, normal and height features benefit some tasks like “Push Button” where the main failure cases are that the gripper does not push down enough to the button. We also notice relatively large variations on individual tasks, and thus we consider the averaged performance is more stable for comparison.

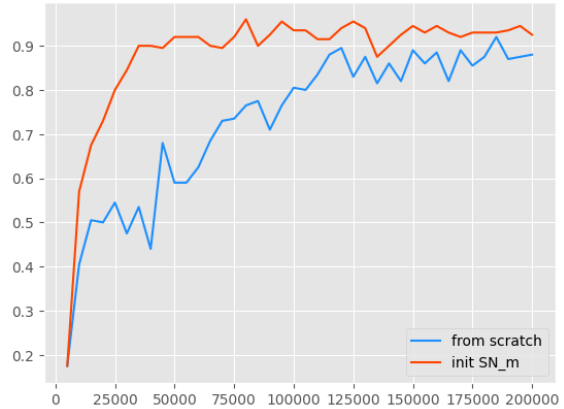


Figure 7. Success rate on RL Bench validation split in different training iterations. We compare the policy trained from scratch and the model initialized from SUGAR pre-training.

In Figure 7, we present the performance on the RL Bench validation split for policies trained from scratch and initialized from SUGAR pre-training. We can see that the policy can converge much faster and achieve better performance with the pre-training.

C. Real-world Robotic Manipulation

To evaluate the effectiveness of SUGAR pre-training for real robots, we further perform real world experiments for language-guided robotic manipulation.

To be specific, we use a UR5 robotic arm equipped with a RG6 gripper and set two Intel RealSense D435 RGB-D cameras on the front and lateral sides of the robot’s workspace. We adopt 5 real-world tasks including *stack cup*, *put fruit in box*, *open drawer*, *put item in cabinet* and *hang mug* as illustrated in Figure 8. For each task, we collect 20 real-robot demonstrations, where each demonstration consists of RGB-D images and proprioceptive information of the gripper at keysteps (typically 3-6 keysteps).

We train a multi-task policy using the collected real-robot data, and evaluate 10 episodes for each task where the object locations and distractor objects are different from the training data. Table 8 presents results of a model trained from scratch on the real robot data and a model initialized from SUGAR

Table 7. Averaged success rate of three runs for multi-task policies on 10 tasks of RL Bench simulator.

Method	Pre-train	Avg.	Pick & Lift	Pick-Up Cup	Push Button	Put Knife	Put Money	Reach Target	Slide Block	Stack Wine	Take Money	Take Umbrella
PolarNet [7]	ShapeNetPart	89.8 \pm 1.5	97.8 \pm 1.4	86.0 \pm 2.1	99.6 \pm 0.4	80.5 \pm 1.1	94.1 \pm 0.8	100 \pm 0.0	93.4 \pm 0.9	80.5 \pm 3.6	68.1 \pm 4.3	97.8 \pm 0.2
SUGAR	-	85.9 \pm 3.9	77.7 \pm 4.9	92.7 \pm 4.2	91.7 \pm 0.9	69.4 \pm 8.0	87.7 \pm 1.2	99.7 \pm 0.4	94.3 \pm 0.4	83.1 \pm 7.8	66.8 \pm 9.2	95.7 \pm 1.6
	SN_m	93.0 \pm 1.0	93.1 \pm 1.3	94.5 \pm 1.0	98.9 \pm 0.8	85.4 \pm 1.4	97.8 \pm 1.3	100 \pm 0.0	97.9 \pm 0.8	94.5 \pm 1.5	70.0 \pm 1.6	98.4 \pm 0.2
	Ens_m w/o grasp	92.0 \pm 1.6	93.1 \pm 1.3	93.7 \pm 1.3	98.8 \pm 1.1	85.5 \pm 0.1	92.3 \pm 5.3	99.9 \pm 0.1	97.3 \pm 1.4	93.7 \pm 0.6	68.8 \pm 4.2	97.2 \pm 0.9
	Ens_m	93.0 \pm 1.7	95.8 \pm 1.3	95.7 \pm 1.6	96.1 \pm 5.1	86.5 \pm 2.7	94.2 \pm 1.6	100 \pm 0.0	97.0 \pm 0.5	93.5 \pm 0.6	72.0 \pm 2.9	98.8 \pm 0.9

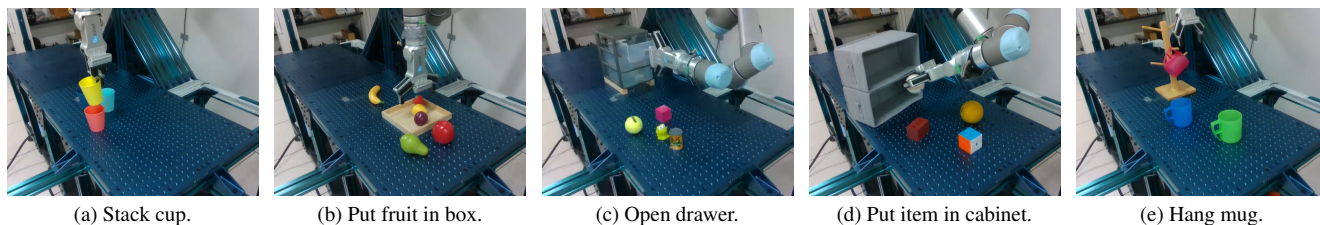


Figure 8. Illustration of the adopted five real robot tasks.

Table 8. Success rate of multi-task policies on 5 real-world tasks. We evaluate 10 episodes for each task.

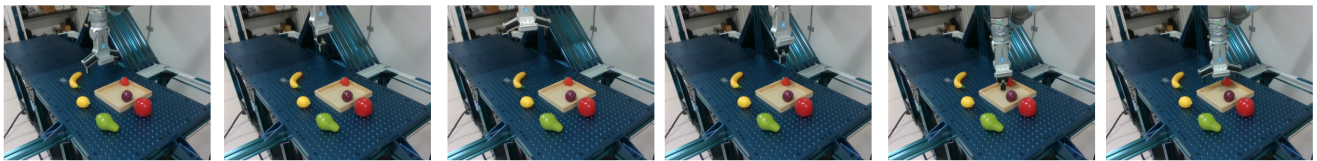
	no pretrain	SUGAR (Ens_m)
Stack cup	0/10	10/10
Put fruit in box	0/10	4/10
Open drawer	0/10	3/10
Put item in cabinet	0/10	9/10
Hang mug	0/10	6/10

pre-training. The model trained from scratch overfits on the limited training data and totally fails in evaluation. As shown in Figure 9a, the model trained from scratch has serious problems of localizing the target object. Our SUGAR pre-training significantly improves the performance for language-guided manipulation in the real world, leading to an average of 64% success rate over the five tasks. Figure 9b presents a successful case of putting lemon in the box. However, we also notice that the model initialized from SUGAR pre-training still has problems in precise object localization in Figure 9c. The problems can result from the sub-optimal network design that largely downsamples the point cloud, the regression action prediction head that is more unstable compared to classification, and the noisy depth sensors. We will investigate more on the policy networks to improve the robotic manipulation performance.

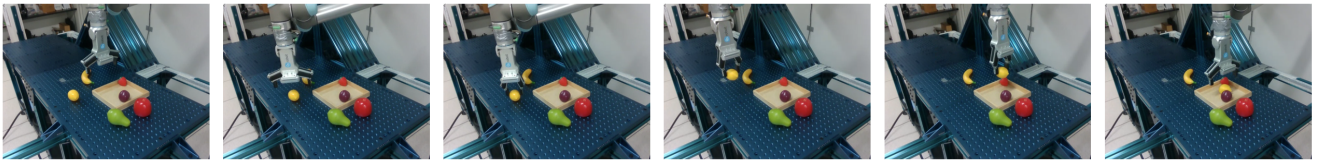
D. Limitations and Future Work

This work only adopt a plain transformer architecture for point cloud encoding, which is computationally expensive. For example, compared to the SoTA method PolarNet [7], our model consists of 4.5x more parameters (65M vs. 14M)

and runs 1.3x slower (18h vs. 14h in training on one V100 GPU). This is because PolarNet is based on a UNet backbone which is more efficient. Our vanilla transformer-based backbone alone does not show clear advantage over the UNet backbone for robotic manipulation as seen in Table 7, although the proposed pre-training significantly boosts the performance. We believe that the proposed pre-training can benefit other architectures and plan to explore more efficient 3D backbones in our future work.



(a) A failure case of the multi-task policy trained from scratch.



(b) A successful case of the multi-task policy initialized from SUGAR pre-training.



(c) A failure case of the multi-task policy initialized from SUGAR pre-training.

Figure 9. Examples of real world execution on the *Put fruit in box* task for different policies.