Demonstrating Multi-Suction Item Picking at Scale via Multi-Modal Learning of Pick Success

Che Wang*, Jeroen van Baar†, Chaitanya Mitash†, Shuai Li*, Dylan Randle†,
Weiyao Wang†, Sumedh Sontakke*, Kostas E. Bekris†, and Kapil Katyal‡
Amazon Robotics
*Seattle, WA; †North Reading, MA; ‡Arlington, VA
Email: (corresponding author) chewwang@amazon.com

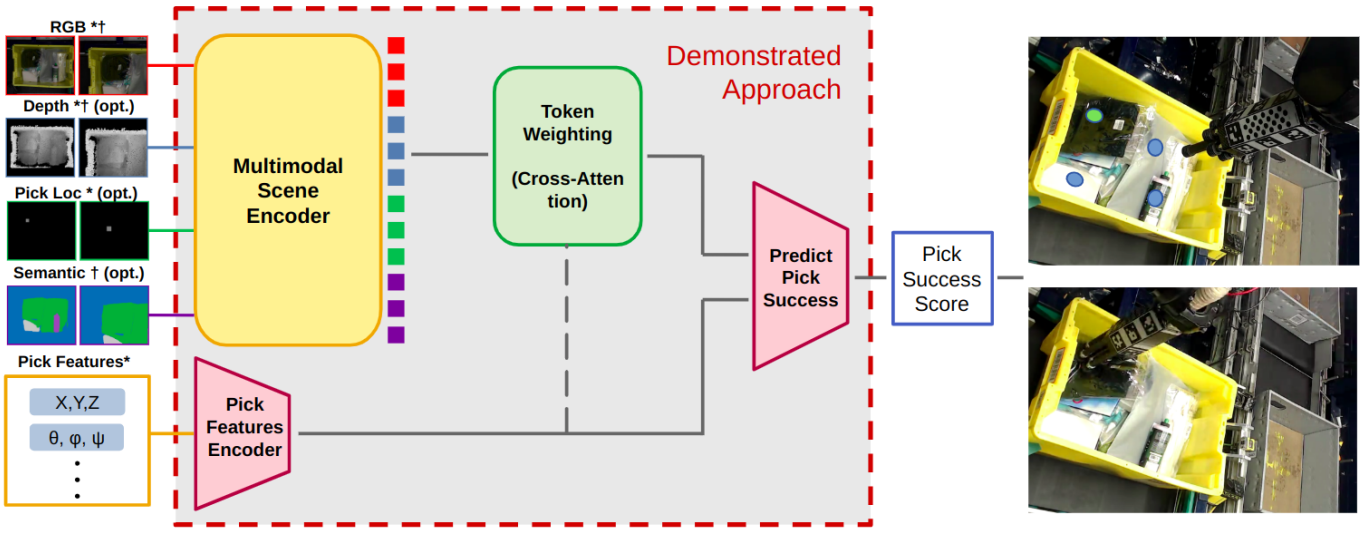


Fig. 1: Given RGB, depth, pick location and semantic data for a picking scene as well as features of candidate, multi-suction picks, this work demonstrates how state-of-the-art multi-modal visual encoders can learn expressive representations. These representations are used to evaluate candidate picks via a cross-attention mechanism and a pick success prediction head. All input image modalities on the left (RGB, Depth, Pick Location, Semantic) can also be cropped to produce a local image (right side of the input images pairs). Operating over local image crops helps boost performance. The modalities marked with * are used in the default approach. The modalities marked with † are utilized during pretraining. The modalities marked with (opt.) are optional during the finetuning/inference but can enable higher performance if deployed. The demonstrated strategy is trained using picks executed on an operating, industrial setup. They are generated by a previously deployed engineered approach. The strategy achieves improved performance relative to the engineered approach and learning-based alternatives.

***Abstract*—This work demonstrates how autonomously learning aspects of robotic operation from sparsely-labeled, real-world data of deployed, engineered solutions at industrial scale can provide with solutions that achieve improved performance. Specifically, it focuses on multi-suction robot picking and performs a comprehensive study on the application of multi-modal visual encoders for predicting the success of candidate robotic picks. Picking diverse items from unstructured piles is an important and challenging task for robot manipulation in real-world settings, such as warehouses. Methods for picking from clutter must work for an open set of items while simultaneously meeting latency constraints to achieve high throughput. The demonstrated approach utilizes multiple input modalities, such as RGB, depth and semantic segmentation, to estimate the quality of candidate multi-suction picks. The strategy is trained from real-world experience, i.e., given examples of successful and failed attempts to pick items. The training picks have been generated by an engineered strategy. A real-world limitation when learning in such live, industrial setups is that only a single or a few picks can be attempted per scene. The learning strategy first pretrains multi-modal visual models in a self-supervised manner to effectively reconstruct the input modalities in the target domain. A downstream model is then trained to evaluate the quality of multi-suction picks given the learned multi-modal embedding, while the multi-modal model is further fine-tuned. The manuscript provides comprehensive experimental evaluation performed over a large item-picking dataset, an item-picking dataset targeted to include partial occlusions, and a package-picking dataset, which focuses on containers, such as boxes and envelopes, instead of unpackaged items. The evaluation measures performance for different item configurations, pick scenes, and object types. Ablations help to understand the effects of in-domain pretraining, the impact of different modalities and the importance of finetuning. These ablations reveal both the importance of training over multiple modalities but also the ability of models to learn during pretraining the relationship between modalities so that during finetuning and inference, only a subset of them can be used as input.**

I. Introduction

Picking items from unstructured piles is an important yet challenging task for robotic manipulation in real-world settings, such as warehouse automation. Methods for picking from clutter have to be robust to an open set of items while simultaneously meeting latency constraints to achieve high throughput. While robot manipulation in clutter has long been approached via model-based reasoning [12, 15, 7, 13], the recent focus has been on data-driven approaches and their potential benefits. In particular, learning-based methods have been introduced to address the large variety of items, for pinch grasping [19, 22, 10], suction-based grasping [21, 3], or both [23, 24]. Suction-based grasping, the modality considered in this work, is popular in real-world settings (e.g., logistics and fulfillment) as it simplifies the attachment of a target object to a robot's end-effector.

This work is inspired by data-driven methods for robot picking as well as the progress in multimodal, multi-task models [34]. It aims to explore the impact of such models in robot picking given the availability of large-scale data obtained from real-world industrial robotic deployments. The accompanying experiments use large real-world picking datasets. In contrast to prior work that is often limited either to a closed set of items or training in simulation, this work aims to address the case of picking an open-set of possible items that can appear in any possible configuration, i.e., all possible products and configurations that arise in a real-world warehouse environment, using a multi-suction end-effector.

Recent work [20] proposed a shallow model for pick success evaluation, which relied on features engineered by human experts. The model was similarly derived from large-scale, in-domain multimodal data for a multi-suction end-effector. It focused, however, on picking packages instead of items from an open-set. The variety in appearance for packages, such as boxes, or envelopes, is considerably smaller compared to an open-set of items, considered here. The prior work demonstrated that the shallow model with engineered features achieved stronger performance than a deep learning model[1].

The current study first confirms the prior finding in the context of open-set item picking, i.e., it is not trivial to develop a deep architecture that learns effective visual representations that outperform a shallow model using features engineered by an expert (see Fig. 3). Motivated, however, by the need to automate and simplify the development and training of picking solutions, this work demonstrates that recent deep visual architectures allow the automated learning of multimodal representations from real-world data.

In particular, we propose a model for pick success prediction (see Fig. 1) that operates over a multimodal visual encoder, such as the Multimodal Multi-Task Masked Autoencoder (MultiMAE) [1] trained on RGB, depth and semantic data. This encoder can be pretrained to correlate information across different modalities, resulting in a representation that captures information about items and their relation in a cluttered pile. Then, the learned multi-modal representation of the scene is combined with pick features via a cross-attention mechanism and is finetuned to predict the quality of the pick. Experiments show that this architecture achieves improved performance over a highly-tuned, shallow model as well as against deep model baselines that utilize a generic, frozen encoder. The demonstrated architecture also consistently outperforms the alternatives when applied to different types of picks as well as to scenes corresponding to package picking. All experiments presented are focused on multi-suction end effectors as they work well for the warehouse setting.

In summary, the contributions of this work are the following:

1. The combination of multimodal pretraining and finetuning of the MultiMAE, properly coupled with an encoding of the pick candidate information, such as position and orientation, replaces the need for engineered features, and outperforms the previously demonstrated shallow model.

2. We present experiments showing the learned multimodal representation works on a standard item picking setting, item picking with increased occlusion and random pick samples, and a setting where we pick packages instead of items.

3. Extensive ablations of the demonstrated architecture reveal which technical components are critical to achieving the best performance and a series of insights in the large-scale robot picking setting. For instance, while many efforts in the literature advocate for frozen visual representations trained on large generic datasets [28, 29, 33], we show that considerable performance gains can be achieved through in-domain pretraining and finetuning. The majority of these gains can be achieved even with small exposure to in-domain multimodal pretraining (e.g., just seeing 1% data of the available data).

In the appendix, we provide additional results, in the supplementary materials, we provide videos of picking items from clutter to illustrate the challenging nature of the task.

II. Related Work

This section first reviews work on robot picking and then discusses learning visual representations for robotic tasks.

A. Robotic Item Picking

Traditional methods for robot grasping [26] involve geometric reasoning, planning and optimization methods. They often calculate the poses and forces for robotic contacts so as to satisfy certain mechanical constraints, such as force or form closure [27, 25]. The requirement, however, for accurate geometric and physical object models often limits the effectiveness of these solutions in unstructured setups involving an open-set of objects, which is the focus of this work.

This motivated the introduction of learning-based approaches for robot grasping more than a decade ago [16]. Such data-driven methods brought the promise of more effective grasping in challenging, cluttered setups with unknown objects. The majority of these works focus on generating grasps for parallel, pinch grippers [10, 9, 34]. Since the Amazon Picking Challenge [6], however, it has been well understood

[1]That model used a convolutional visual encoder using as input an RGB image but without access to multimodal data or engineered features.

that suction-based grippers can be very effective in real-world picking setups by simplifying pick reasoning. While the underlying representation learning tools and insights of this work are not necessarily limited only to such end-effectors, the accompanying experiments have been performed using a multi-suction gripper. Suction-based grippers have been deployed in real-world production environments and allow fast and robust picking of items that can potentially be heavy.

Some of the data-driven solutions for robot picking have been extended to address suction-based grippers, such as the Dex-Net family of solutions [23, 3]. In Dex-Net, a pick candidate is evaluated using an expert-designed evaluation system. In this work, we directly train the model to learn pick success prediction given success labels of past production data, and demonstrate that recent deep architectures have the ability to generate informative representations for this task. This direction minimizes the need for expert knowledge, giving a more effective solution to the picking problem.

In the context of approaches that have been tested on large-scale, real-world data, prior work Li et al. [20] has demonstrated that a shallow model can have good performance in terms of pick success prediction as part of a larger grasp planning pipeline for package picking. Packages tend to have more limited geometries and physical features relative to picking an open set of items, which is the focus of this work. The shallow model in prior work outperformed a deep learning model with a convolutional visual encoder. This work shows that with proper pretraining and finetuning, a deep model with a multimodal visual encoder can in fact outperform the shallow model as well as alternative deep architectures in terms of picking items from an open-set. Thus, a critical objective is to identify appropriate intermediate visual representations that allow the evaluation of robotic picks.

B. Learning Visual Representations

The value of a visual representation for control has been recognized previously. Pretrained vision models for control [29] exploit a visual representation for learning a motor policy. Alternatives focused on manipulation [28] have proposed a representation for video image data for learning a variety manipulation tasks. A Masked Auto-Encoder (MAE) [14] was proposed to be used to learn a control policy [33]. In order to exploit the multimodal nature of the available data, this work adopts Multi-Modal Multi-Task Masked Autoencoder (MultiMAE) [1], the multimodal extension of MAE instead. In addition, the above mentioned approaches use generic robotic data for pretraining and freeze the visual encoder during downstream operation. This work identifies that when large amounts of in-domain data are available, pretraining and finetuning in-domain data provides improved performance.

There are also other recent efforts that investigate using multimodal input for robotic control [17, 34, 18, 30]. We focus on MultiMAE due to:

(1) Multiple visual modalities are often available in robotic workcells: a color camera for RGB data, a depth sensor for depth images, and a segmentation model that provides semantic segmentation information. To generate the robot picking actions, the demonstrated architecture reasons about the 3D surface of the items as well as the segment boundaries of the items. So we expect additional modalities including depth and segment can help the model to better predict pick outcomes. The accompanying ablations show that multimodal pretraining and finetuning (with RGB, depth and semantics), which MultiMAE can leverage [1], bring significant performance improvements compared to using a single modality.

(2) We are interested in a representation that can be pretrained in a self-supervised manner. This allows us to operate over large-scale data without additional curation and labeling. MAEs lend themselves well to this desirable objective. On top of that, they are also simple, robust and efficient [14].

To the best of our knowledge, this is the first application of a multimodal model for prediction of pick success at scale.

III. Setup

Robotic work cells in the industry are instrumented with sensors of various modalities, such as color and depth. The sensors are typically providing a top-down observation of the cluttered items to be picked. An example top-down view is shown in Fig. 1. The visible items in the color images are segmented using previously trained segmentation models, providing semantic information. A set of pick candidate locations is generated across the items, where a pick candidate is defined by a number of "pick features", including X, Y, Z coordinates of the pick location, which pistons of the multi-suction gripper are activated during the pick, orientation of the end-effector, among others. To achieve high throughput, items with a high probability of pick success for a particular pick candidate, should be picked. The goal is thus to train a pick success prediction model, which given the multimodal image data, provides a prediction of success for a given pick candidate.

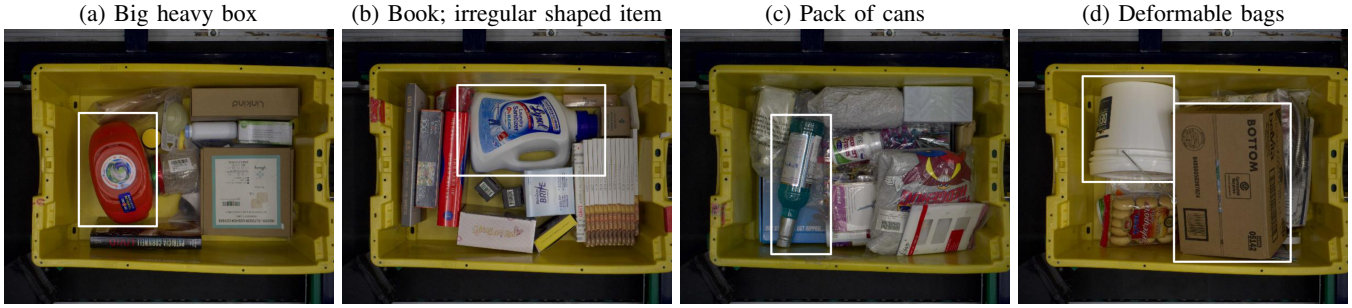
A. Datasets

We have access to two datasets for training models for item pick success prediction: the first dataset, the *standard* dataset, contains 343K multimodal images of items in unstructured clutter for pretraining the MultiMAE. It also has nearly 275K training examples which contains both multimodal images along with pick features for the specific pick candidate of an item, for which the grasp was executed in deployment. For the executed grasp, we also have an annotation of pick success or failure. The second dataset, which we will refer to as the *random* dataset, is a much smaller dataset containing the same type of data as the 275K training examples mentioned above. The difference with the standard dataset is that the items were picked according to randomly selected pick candidates. So this dataset contains more picks of items that are partially occluded by other items. We also present results on a dataset obtained for a different domain involving the picking of packages.

B. Nature of items

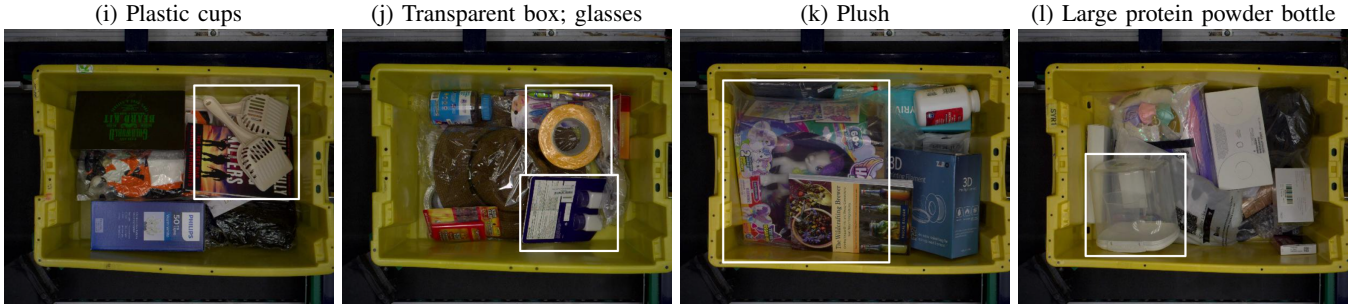
In Figure 2, we further emphasize the challenging nature of our setting with example pick scene images.

(a) Big heavy box (b) Book; irregular shaped item (c) Pack of cans (d) Deformable bags



(e) Washer pods container (f) Detergent (g) Glass bottles (h) Bucket; bag of pastry

(i) Plastic cups (j) Transparent box; glasses (k) Plush (l) Large protein powder bottle



(m) Spatula without packaging (n) Tape; irregular shaped item (o) Transparent toy box (p) Transparent container

Fig. 2: Example pick scenes from the open-set item manipulation task. Some difficult items are highlighted in white boxes.

Items such as the big box in (a) can be hard to manipulate due to its heavy weight; some items are deformable such as the plastic bags in (d), the plush toy in (k); some items are transparent, such as the container in (p); some items have irregular or round shapes, making it hard to find a proper pick point, such as the washer pods containers (e), the detergent bottle (f), glass bottles (g), plastic bucket (h), spatula (m), and tape (n); some items can be difficult to pick for multiple reasons, for example the box in (j) is transparent, has an irregular surface and can potentially be heavy, and the toy box in (o) has both deformable and rigid parts, with some transparent packaging. The protein powder bottle in (l) is both round and heavy. It is also important to note that some items can be damaged if not picked properly, such as the glasses in (j), and the book in (b). Additionally, boxes that contain smaller items could open and plastic packaging can be damaged when picked incorrectly. These examples show that reliably picking from an open set of real world items is a very challenging task.

IV. Methodology

A. Preliminary Evaluation of Visual Encoders

Previously, a shallow model with manually engineered features was proposed, which outperformed a deep model with a CNN-based visual encoder [20]. However, several works in the literature, e.g., [33] have proposed to use pretrained visual

encoders. We first trained a model similar to that shown in Fig. 1. Instead of the MultiMAE encoder, however, we used a pretrained visual encoder whose weights are frozen, along with token mean pooling for the token weighting (Fig. 4). We train the model on the datasets with annotated pick success (as described above), and unimodal input, i.e., RGB only. Fig. 3 shows the performance of the shallow model compared to a visual encoder with randomly initialized weights (ViT no pretrain), and various visual encoders pretrained on generic datasets. Results show that using a frozen generic encoder clearly achieves higher performance over an encoder with randomly initialized weights. Nevertheless, the shallow model with expert-engineered features significantly outperforms the pretrained visual encoder models.

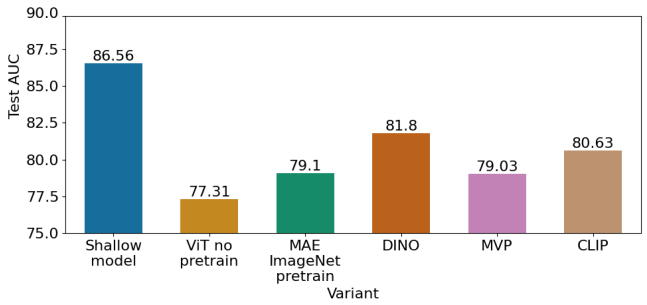


Fig. 3: Performance of a shallow model compared with pretrained visual encoders. From left to right: shallow model (XGBoost) baseline that uses expert features; using a ViT model [8] with random initial weights; ViT model with MAE [14] pretraining on ImageNet; ViT model with DINO [4] pretrain; and ViT model with MVP [33] pretraining.

B. Demonstrated Strategy

To close the performance gap with the shallow model, it is evident that pretraining with generic data by itself is not sufficient. We propose several improvements over these models with pretrained visual encoders. Again referring to Fig. 1), we propose the use of a MultiMAE for visual encoding to exploit the availability of multimodal image input. Each new modality will be processed into additional image tokens with a different input adapter. Furthermore, instead of mean pooling, the pick features are input to a cross-attention layer for the Token Weighting. The cross-attention layer aims to learn to relate the encoded pick features over the tokens. Finally, we propose to perform two stages of training: first we train the MultiMAE in a pretraining stage with pixel reconstruction objectives, then we train the entire model in a second stage with the pick success prediction objective. During the second stage we update the weights of the MultiMAE while training. We refer to this second stage as the finetuning stage. Before presenting results and our ablation study, we will first describe the pretraining and finetuning stages in more detail.

C. Pretraining the MultiModal Autoencoder

We first pretrain the MultiMAE on 343K examples of deployment data in the standard dataset. The nature of the

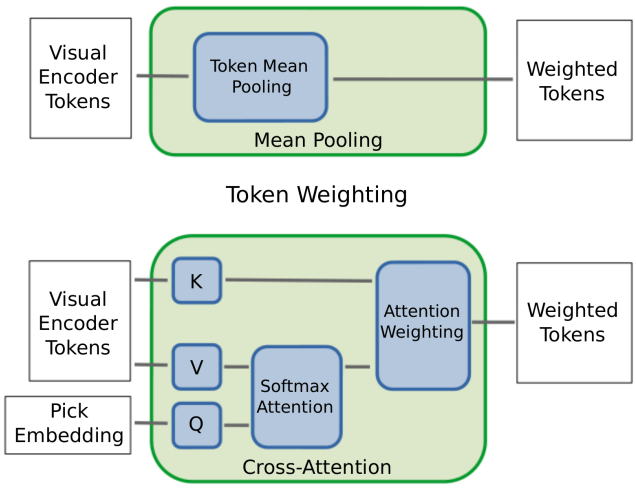


Fig. 4: Given visual encoder tokens outputted by the MultiMAE encoder, we implement two types of token weighting: either using a simple mean pooling of all tokens, or weighting via cross-attention. In the latter case, the encoded pick features are also provided as input to the cross-attention block.

multimodal data is similar to that in the original MultiMAE work [1]. During pretraining, the model receives as input RGB, depth and semantic images of the pick scene. We use a similar schedule as in [1] for randomly masking a portion of the tokens from the three modalities, and the MultiMAE learns to reconstruct the pixels in the missing image patches. When pretraining on the deployment dataset, the MultiMAE is initialized with weights from a publicly available version pretrained on ImageNet, then pretrained for 800 epochs.

At pretraining, the RGB and depth images are resized to 224×224. For the semantic segmentation input images, we first define 9 classes of semantic categories for the items encountered. Instance segmentation images in our dataset, obtained from a previously trained segmentation model, are directly converted into semantic segmentation images. The images are downsampled by a factor of $4\times$ compared to the RGB and depth data to reduce computation cost [1]. Thus, the resulting dimensions are 56×56. At the start of pretraining, we initialize a different class embedding for each semantic class, which is then updated during pretraining. When reconstructing the semantics, MultiMAE will predict a value for each class, i.e., the semantics output has nine channels per pixel. An example of the resulting model's ability to reconstruct all three modalities with randomly masked input is shown in Fig. 5. When we visualize the reconstruction, for each pixel, we show the class with the highest prediction value.

D. Fine-tuning for Pick Success Prediction

We next train the MultiMAE along with the additional network components on 275K training examples to perform pick success prediction. We refer to this stage as *finetuning*. We use 69K validation examples to decide when to stop training. The model is then evaluated on 86K test data, which have not

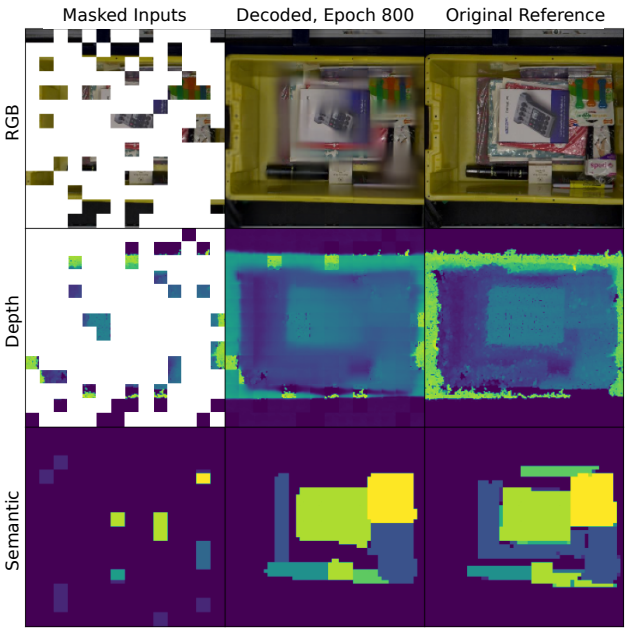


Fig. 5: Example of MultiMAE reconstruction for all three modalities, i.e., RGB, depth and semantic segmentation. The left column shows the ground truth reference images.

been seen during both stages of training. For each split the success to fail ratio is about 11:1. We use a weighted loss to account for this high class imbalance. We also incorporate pick information into the model by using a cross-attention module as shown in Fig. 4 bottom. The cross-attention results in a weighted sum of the image tokens based on the pick features.

For simplicity, we discuss the case where we only consider one attention head. We first project the pick embedding into a query token Q, and project the encoder tokens (we omit the global *CLS* token) into value tokens V and key tokens K. The V, K, and Q projection layers are simply linear layers and are trained with the pick success objective. Following self-attention [8], Q and K are multiplied and then divided by a scale value, followed by a softmax, resulting in an attention weight vector. We then take a weighted sum of the encoder tokens using the attention weight vector. The intuition behind using a cross-attention layer is the fact that we aim to combine information from two entirely different domains: images and pick features. The MultiMAE provides a visual encoding of the multimodal images, which in turn is *queried* with pick features for pick success. In the next section, we will compare cross-attention with the more naive setting of using mean pooling, where each token has the same weight for the attention weight vector. Finally, the weighted sum is then input to a MLP prediction head, together with the projected pick features, to obtain pick success prediction.

Next we will present our experimental results that reflect on the effectiveness of the demonstrated approach.

V. Experimental Results

We use the Area Under the Receiver Operating Characteristic Curve (ROC AUC) as our main metric for evaluating the performance. This metric indicates how well the model performs under different decision thresholds and is more suitable for the case where the data is imbalanced [11]. The expert-engineered features are always provided for the shallow model baseline, but not to the demonstrated MultiMAE model.

A. Experiments

For our main experimental evaluation, we will adopt the best performing model among the variants of the MultiMAE we have experimented with. We will refer to this as the *demonstrated approach*. The settings for the demonstrated approach are:

- We pretrain on in-domain RGB (R), depth (D) and semantic segmentation (S) images (Pretrain R-D-S);
- For the finetuning stage we use RGB and depth images (Finetune R-D), along with an image of the pick location.
- The token weighting is done using cross-attention.
- An additional pick location image is added to further boost performance.
- During finetuning, the MultiMAE encoder weights are updated according to the pick success loss.
- Finally, the input images for finetuning are cropped according to a bounding box with additional padding derived from the target item's segmentation mask. We refer to this as the local crop. The local crops are augmented with random offsets for additional robustness.

Ablation studies that follow will evaluate the importance of the above implementation choices.

1) Largely Unoccluded Item Picks - Standard Dataset: Fig. 6 shows a performance comparison of the demonstrated MultiMAE approach against the previously best performing method to date on this dataset, i.e., the shallow model [20], as well as a baseline version of a (unimodal) mask autoencoder (MAE Base). The demonstrated approach has a significantly higher performance than the MAE Base variant (79.1 vs 90.6). Their differences are studied further in the ablations. The demonstrated approach also outperforms the shallow model by about 4 points without the need to access the expert-engineered features that the shallow model uses.

We also compared against a learn-from-scratch, point-cloud baseline. This baseline adopts the PointTransformerV3 (PTv3) model [32] as point cloud encoder. Each encoder block receives a point cloud as input and employs: 1) a 3D sparse convolution layer [5] to serve as conditional positional embedding, and 2) a self-attention transformer layer on patches created using space filling curves. An action is represented by a set of pick tokens. We use a linear layer to embed a pick feature vector containing the discretized approach angle, wrist rotation, and suction cup activation map to get the feature, and store this for each pick token. The pick token is then used as query in the cross attention and the average location of activated suction cups is used as location for this pick token

when performing 3D rotary positional encoding for the cross attention. A linear projection head is then applied to the output of this decoder layer. This baseline results in a test AUC of 84.6, which is stronger than the MAE baseline, but is lower than both the shallow model and the demonstrated MultiMAE model. This again highlights that multimodal pretraining can provide significant benefits in terms of pick success prediction. Note that this result simply shows the scene encoder in the demonstrated approach can also be a point cloud model. Although this paper is focused on 2D multimodal data, we believe proper 3D pretraining as well as combining 3D and 2D modalities can be beneficial, and we plan to further investigate 3D modalities in future work, as discussed in Section VI.

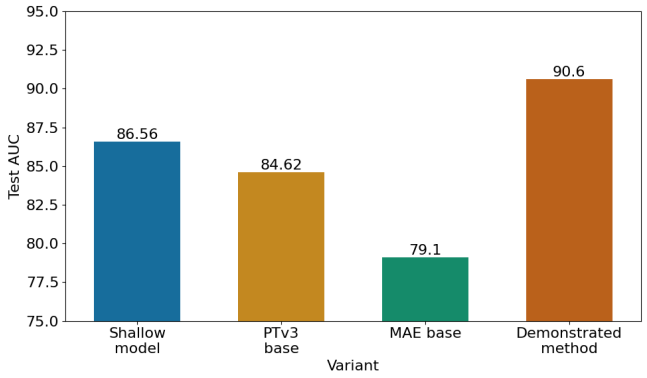


Fig. 6: Performance comparison of the shallow model [20] (relies on expert features) against a PTv3 baseline, a MAE baseline, and the demonstrated method on a dataset consisting of largely unoccluded item picks. The demonstrated approach outperfoms the shallow model by about 5% in test AUC.

2) Partially Occluded Item Picks - Random Dataset: In order to evaluate the robustness of the demonstrated approach and learned representation, we also evaluate the demonstrated approach against the shallow model on a random pick dataset. The item configuration distribution of this dataset, with many picks for items that are partially occluded (shown in Fig 7), is very different from the standard dataset, where we mostly pick unoccluded items. This "random dataset" contains 8,461 training examples, 2,115 validation and 2,644 test examples. The success-failure ratio on each split is approximately 4.4:1.

Table I compares the shallow model and the demonstrated approach in three different settings: (a) Pretrain on the standard dataset, finetune on the random dataset, then test on the random dataset (PT: STD, FT: RND, Test: RND); (b) Pretrain on standard, finetune on standard, and then zero-shot test on the random dataset (PT: STD, FT: STD, Test: RND); and (c) Pretrain on standard, finetune on standard first, and then further finetune on the random dataset, then test on the random dataset (PT: STD, FT: BOTH, Test: RND). Note that the shallow model does not have pretraining, and for case (c), the shallow model is finetuned (trained) on the combination of the standard and random datasets.

The demonstrated approach outperforms the shallow model consistently in all settings. This demonstrates that the representations learned from the demonstrated approach are robust on different item configurations.

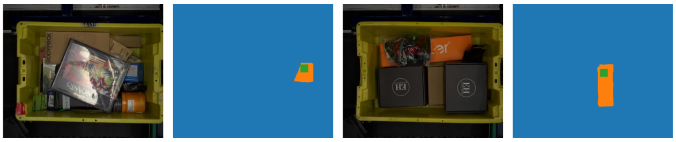


Fig. 7: Examples of picking partially occluded items. Columns 1&2, and columns 3&4 contain an example of an RGB image and corresponding image with the target item mask and pick point location. In both cases, the target item is partially occluded by a larger item at the top of the unstructured pile.

TABLE I: Performance comparison between the shallow model and the demonstrated approach on a different dataset where random items that are partially occluded are picked in different settings. The model is pretrained (PT) on standard dataset (STD). And then can be finetuned (FT) on the random dataset (RND), or first finetuned on the standard dataset then on the random dataset (BOTH). Then the model is tested on the random dataset.

Model	PT	FT	Inference	Performance
Shallow	-	RND	RND	82.92
Demonstrated	STD	RND	RND	**86.28**
Shallow	-	STD	RND	83.58
Demonstrated	STD	STD	RND	**85.87**
Shallow	-	BOTH	RND	85.16
Demonstrated	STD	BOTH	RND	**88.06**

3) Package Picking Dataset: We also investigate whether the demonstrated approach can work on a different pick scene containing packages rather than items. Here we pretrain and finetune on the multimodal data derived from the package manipulation task described in [20]. This is quite different from the item manipulation setting, and the majority of the items to pick are packages such as boxes, bags and envelopes, as shown in Fig. 8. Here we consider a package picking dataset with 100K training, 20K validation and 20K test examples, each with 2:1 success-failure ratio.

Fig. 8: Example pick scenes with packages. There is less variety in object appearance relative to the domain of item picking, but the level of clutter can still be challenging.

Table II shows when we perform multimodal pretraining and finetuning on this package picking dataset, we can also out-

perform the shallow model that relies on expert features, and obtain the best performance. This demonstrates the demonstrated approach also works with a different pick scene with a different item distribution.

TABLE II: Package picking dataset experiment. Results show that the demonstrated approach can also outperform the shallow model for this different setting where the majority of the items to pick are packages such as boxes, bags and envelopes.

Package Picking Dataset Experiment	Performance
Shallow Model	86.50
No Pretrain	84.54
Generic RGB Pretrain, Frozen	86.99
In-domain Multimodal Pretrain, Finetune	**88.28**
Item Picking Data Pretrain	87.40
Item Picking Data Pretrain, Frozen	85.43

B. Ablation Studies

We present a series of ablations to better understand what the most important factors are for the performance of the demonstrated method, and how performance differs with variations of data, input modalities and other settings.

1) Effect of Visual Modalities: Table III shows the effect of having different combinations of the visual modalities, i.e., RGB (R), depth (D) and semantics (S), at the pretraining and finetuning stages. The results show that pretraining with more modalities can bring performance gain even when finetuning with RGB only. When pretrained with all three modalities, having depth and semantics as additional input at the finetuning stage can also further improve performance. When only a single modality is used at finetuning, RGB has the best performance, followed by depth and semantics. Note that for our demonstrated approach, we do not use semantics during finetuning, since further adding semantics will increase latency while providing only minimal performance improvement.

2) Effect of In-Domain Pretraining: Many popular works in the literature advocate for the use of frozen visual representations that are pretrained on large generic datasets [33, 29, 28]. This is reasonable for tasks where only a small amount of in-domain data is available. Table IV shows how pick prediction success performance can be affected heavily by the pretraining dataset. Although pretraining on generic datasets (ImageNet) with either RGB only or all three modalities will improve performance over not pretraining the visual encoder at all, pretraining on in-domain data with all three modalities can further boost performance (row 4 in Table IV).

3) Effect of Local Crop Sizes: For the visual input at the finetuning stage, we can either use the image of the entire pick scene, or we can crop out a portion of the image centered around the target item. The size of the crop is determined by the segment bounding box and a padding value. We note that the model is pretrained on both global and local crop due to random crop augmentation following [1]. A padding of 0 means the crop is tight around the target item segment, larger values for the padding will include more surrounding information. When we have multiple visual modalities, we crop all of them the same way. Fig. 9 shows the global scene image and the same image with different local crop sizes. The performance we get when using them as input is shown directly below each image. A local crop is better than a global image, a padding value of 50 gives the best score.

4) Different Ways to Incorporate Pick Features: The pick success prediction model needs to integrate information from very different domains: visual modalities and encoded pick features, which contain 3D pose and other information relevant to the pick, e.g., activated suction cups of the multi-suction cup end effector. How we combine the images and the pick information can affect the performance. In this ablation we study three different ways to incorporate the pick features: (1) use a cross-attention module for learned token weighting; (2) use a local crop of the input images, centered around the target item, instead of the entire (global) image; (3) mark the pick point on another 2D image, and use it as an additional visual input modality. Table V shows how different ways of

TABLE III: Visual modality ablation. Comparison of performance for pretraining and finetuning with different modalities. The modalities are RGB (R), depth (D), and semantic segmentation (S) images. The effect is measured with respect to the default setting, denoted with **. The default setting is also described in Figure 1. The improvement of using all three modalities for both pretraining and finetuning (last row) is minimal. The best performance is highlighted in bold.

Visual Modality Ablation		Performance	Effect
Pretrain: R	Finetune: R	87.35	-3.24
Pretrain: R-D	Finetune: R	89.11	-1.49
Pretrain: R-S	Finetune: R	89.93	-0.67
Pretrain: R-D-S	Finetune: R	90.05	-0.55
Pretrain: R-D-S	Finetune: D	87.86	-2.74
Pretrain: R-D-S	Finetune: S	85.25	-5.35
Pretrain: R-D-S	Finetune: R-D**	90.60	0.00
Pretrain: R-D-S	Finetune: R-S	90.14	-0.46
Pretrain: R-D-S	Finetune: R-D-S	**90.76**	0.17

TABLE IV: Pretrain domain ablation. Comparison of performance for different pretraining settings. Generic: pretrain on ImageNet, in-domain: pretrain on deployment data. Pretraining with RGB, depth, and semantic segmentation on in-domain data (indicated by **) achieves highest performance.

Pretrain Domain Ablation	Performance	Effect
No pretrain	80.44	-10.16
Pretrain: R generic	81.84	-8.76
Pretrain: R-D-S generic	84.43	-6.17
Pretrain: R-D-S in-domain **	**90.60**	0.00

(a) Global (89.32) (b) Local+0 (89.78) (c) Local+50 (90.60) (d) Local+100 (90.42)

Fig. 9: Global image vs local crop centered around the target item with different padding values. This is also described in Figure 1. Performance (in parenthesis) with local crops is better compared to the global image. A padding of 50 is the best. gives the best performance.

incorporating pick location affect performance.

TABLE V: Pick incorporation ablation. Comparison of performance for different ways to incorporate pick location information. Here "cross-attn" means use cross attention to compute weighted tokens; "pick loc image" means marking the pick point on a 2D image, and using it as an additional visual input modality. The effect is measured with respect to the default setting, denoted with **.

Pick Incorporation Ablation	**Performance**	**Effect**
Global mean pool w/o pick loc image	85.76	-4.84
Global cross-attn w/o pick loc image	**88.55**	-2.05
Local mean pool w/o pick loc image	89.52	-1.08
Local cross-attn w/o pick loc image	**89.76**	-0.84
Local mean pool w/ pick loc image	90.51	-0.09
Local cross-attn w/ pick loc image **	**90.60**	0.00

With global scene multimodal images, without providing a pick location image, and using mean pooling of the encoded image tokens as the weighting (refer to Fig. 4), it is hard for the model to identify the target item when there are multiple items in the scene (row 1 in Table V). Adding the cross attention module allows the model to associate the pick location information of the pick features with the pick point location and target item in the image. Fig. 10 shows evidence that in this case, the model is able to learn (through the pick success prediction loss) which image tokens to pay attention to, which improves the performance (row 2 in Table V.

The right two columns of Fig. 10 show the model initially has random attention, but learns to focus more on the target item and image patches near the pick point. We also visualize the pick point as a small square in the second column, together with the target item segmentation mask. The crosshair in the three columns on the right is only used for visualization purpose in the visualized attention map. Note that in this particular experiment, the target item mask and the pick point square are only provided as a reference and are not available to the model during training or inference.

If we switch to using a local crop image centered around the target item, the model can more easily combine the pick

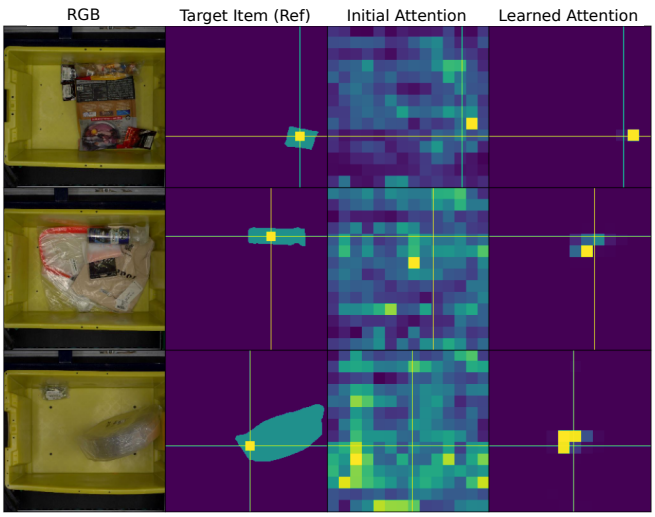


Fig. 10: Different examples of visualization of the learned attention. In this particular case, only the RGB and the pick coordinates are provided to the model. The target item mask and pick point in the second column are only for reference. The model is able to learn to pay attention to regions near the target item and pick point.

features with the encoded images, leading to performance improvement (row 3, Table V). However, notice that the positive effect of cross attention is reduced in the local input setting (compare rows 3 and 4).

Inspired by recent work that shows explicitly marking object location with a single pixel can help manipulation [31], we also tried providing the pick location explicitly as an additional 2D image, and found it can further boost performance. Here the pick location is marked by a square of pixels on a 224×224 single channel image, as shown in Fig. 11. The last two rows in Table V show this can bring performance to 90.60 (Local cross attention + pick loc image). Again notice that the effect of cross attention is weaker when we use a local crop together with the pick location image.

Fig. 11: Left: RGB input image. Middle: pick location and target item mask for our reference. Right: the pick location image which is used as an additional input.

These results show that using cross attention, local crop and pick location image can all improve performance, and their effects are partly overlapping. In practice, the best input setting can depend on the use case, e.g. when evaluating large amounts of pick candidates in a scene, using a global image with cross attention and without pick location image can give

the lowest overhead. So we include all three components in our demonstrated approach for more robustness.

5) Effect of Finetuning the Visual Encoder: We found that finetuning the visual encoder with pick success prediction leads to improved performance compared to freezing it: 87.89 vs 90.6 as shown in Table VI. While it can be good to use a frozen visual encoder in a small data setting, this result shows we should further finetune when a large dataset is available.

TABLE VI: Performance of the demonstrated method with or without encoder finetuning at finetuning stage.

Encoder Finetuning	Performance	Effect
Frozen visual encoder	87.89	-2.71
Finetuned visual encoder	**90.60**	0.00

6) Effect of Data Augmentation: In our experiments we also explored whether using data augmentation during finetuning helps performance. The data augmentation is a random shift of the local crop. The crop is done on images of shape 512×612 (done before they are resized to 224×224). The center of the crop is initially the center of the item segment. When data augmentation is used, the center can be shifted to four directions with a random value (-25 to 25 pixels). We also use a random crop padding of up to 150 pixels (in 50 pixel increments) to change the size of the crop. We then ensure the target item is always in view, so if the crop region is shifted too much that the item is missing in the view, then the crop region is enlarged to include the target item. The same crop is applied to all visual inputs (RGB, depth, semantics, and pick location image). Augmentation is only applied for training examples and not used during validation or testing. Table VII shows that although our dataset is fairly large, data augmentation still can provide marginal improvement.

TABLE VII: Performance of the demonstrated method with or without data augmentation.

Data Augmentation Ablation	Performance	Effect
Without data augmentation	90.40	-0.20
With data augmentation	**90.60**	0.00

7) Effect of Pretraining Epochs: Table VIII shows how much performance changes as we pretrain for a larger number of epochs on in-domain data. We always start in-domain pretraining with generic weights pretrained on ImageNet containing three modalities[1]. On row 1, "0 epochs" means we use the generic pretrained weights without any in-domain pretraining, and then directly go into the finetuning stage. The results show that performance improvement gains are largest in earlier epochs of training, and after pretraining for 200 epochs the performance starts to improve more slowly. Nevertheless, the best performance is obtained with the most epochs, which is 800 in our experiments.

TABLE VIII: Performance of the demonstrated method with different number of pretraining epochs on in-domain data. Numbers after the dashed line indicate higher performance compared to the shallow model.

Pretrain Epoch Ablation	Performance	Effect
0 epochs	84.43	-6.17
20 epochs	87.48	-3.12
100 epochs	90.08	-0.52
200 epochs	90.53	-0.07
400 epochs	90.51	-0.09
800 epochs	**90.60**	0.00

8) Effect of Pretraining Data Ratio: Table IX shows how much performance changes as we pretrain on different ratios of the in-domain dataset. These results show that even when pretrained on 1% of the in-domain data (3.4K), there is already a significant benefit, and the performance can be further improved when we have more in-domain data for pretraining.

TABLE IX: Performance of the demonstrated method when pretrained on different data ratios. By default, we pretrain on 100% of the data (343K). Numbers after the dashed line indicate higher performance than the shallow model. With just 1% of in-domain the model can outperfom the shallow model.

Pretrain Ratio Ablation	Performance	Effect
0%	84.43	-6.17
1% (3.4K)	87.90	-2.70
10% (34K)	89.46	-1.14
25% (86K)	90.14	-0.46
50% (172K)	90.38	-0.22
100% (343K)	90.60	0.00

9) Alternative Representation Learning Methods with In-domain Data: To further understand how other representation learning methods perform with in-domain pretraining, we conducted experiments to compare the performance of MultiMAE, DINO and MOCO-v3 when they are pretrained with in-domain data, then finetuned under the same setting. We use MOCO-v3 instead of CLIP since our dataset does not have text annotation.

For a fair comparison, we first experiment with 4 settings, where we pretrain on in-domain data for 100 epoches using MAE, MultiMAE, DINO, MOCO-v3, respectively. For each setting we then use the exact same finetune set up, with exactly the same hyperparameters and model architecture as the demonstrated method, and using RGB as the only visual input. Since MultiMAE is able to leverage additional visual modalities, we additionally add a setting where we pretrain with MultiMAE for 100 epoches, then finetune with RGB, depth image and pick location image (which is the default setting we used in the demonstrated method). These experiments help us better understand the impact of in-domain pretraining

and multi-modal learning, and whether other representation learning methods with a single modality can achieve the performance of multi-modal pretraining and finetuning.

The results are summarized in Table X. The results show that with in-domain pretraining, MAE, DINO and MOCO-v3 achieve similar results, with MOCO-v3 slightly better than MAE ($85.32 > 84.85$), and DINO slightly better than MOCO-v3 ($85.88 > 85.32$). All three of them are slightly weaker than our shallow baseline (86.56). However, when we use MultiMAE for multi-modality pretraining, even when we only use RGB as the visual modality at finetuning, we see the performance improves to 87.97, outperforming the baseline.

When we add more visual modalities to the finetuning stage (RGB, depth and pick location image, same as the default setting in our demonstrated method), we see performance further improves to 90.08. These results are consistent with the other ablations and show that both in-domain pretrain and multi-modal pretrain + finetuning are critical to the final superior performance.

TABLE X: Comparing different representation learning methods with in-domain pretrain.

Pretrain Method	Finetune Modality	Performance
MultiMAE	RGB, Depth, Pick location	**90.08**
MultiMAE	RGB	87.97
DINO	RGB	85.88
MOCO-v3	RGB	85.32
MAE	RGB	84.85

Takeaways from Ablations: A number of insights are revealed from these ablations:

1) In-domain and multimodal pretraining gives the largest performance boost, the benefit from multimodal pretrain persists even if only RGB is used at finetuning.
2) Using a local crop around the target item with some surrounding information gives best performance.
3) Cross-attention allows the model to focus on the target item, while a pick location image gives further boost.
4) Further finetuning the encoder with pick success prediction loss is better than freezing weights.
5) Data augmentation helps even with large-scale data.
6) Pretraining for longer epochs gives better performance with diminishing returns.
7) Significant gains can be achieved by just performing in-domain multimodal pretraining on 1% of the available training data.

C. Impact of the Demonstrated Approach

1) Confusion Matrix: The confusion matrix in Figure 12 shows that the demonstrated approach gives 9% lower false positive rate and 1% lower false negative rate, compared to the baseline.

It is important to note that each failed pick in the real world can have a cost depending on the item type and the type of failure. For example, if a failed pick caused the item to remain in the tote without damage, then the model can try to pick it up again, and the cost is relatively small.

However, there are other types of pick failure. If the picked item dropped to the ground, it may require human intervention to remove it or it might interfere with other robotic units nearby. If multiple items are picked, it will lead to further error in downstream tasks in the automated system. And if the item is damaged, then it has to be removed and replaced. These failure cases can have a very high cost.

2) Real World Test Deployment: In a real world test deployment of the demonstrated approach, we used the demonstrated approach to perform about 17K pick attempts. Over the course of one week, compared with our baseline method, this approach can potentially reduce multipick by 2%, mispick by 38% and amnesty by 41%, demonstrating its potential for significant cost reduction when deployed at the entire fleet.

In summary, improving the metric by a few percent might not be a big deal in a lab setting, but its impact can be very significant for a large-scale real world system.

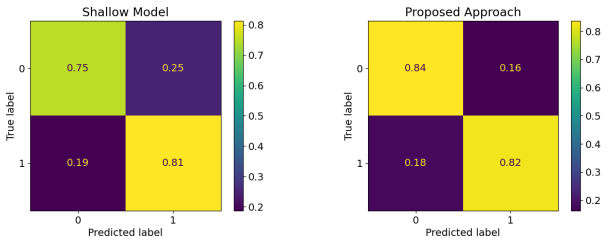


Fig. 12: Confusion matrices on the test set of the standard dataset, for the shallow model and the demonstrated approach.

VI. Conclusion

This paper focuses on the highly challenging, and less explored large-scale real-world robotic item picking setting with multi-suction grippers. It proposes a way to pretrain and finetune a visual encoder to learn useful multimodal visual representations that are better than expert-engineered features. The demonstrated approach can significantly and consistently outperform the previously best performing shallow model on the task of pick success prediction over three datasets with different item configuration, pick scene and object type. It also outperforms a learn-from-scratch point-cloud encoder alternative. Extensive ablations further highlight the critical technical components that lead to this strong performance, and reveal a number of useful insights.

The promising results in this paper open up a number of future directions. One direction is to further incorporate more modalities, such as text, which can allow the learned representations to be better used in other tasks in the same domain [28, 18], such as damage prediction and targeted picking. It is also interesting to further investigate how to effectively use 3D data such as point cloud, or design new pretraining schemes that are tailored towards robotic data and task settings.

VII. Limitations

The limitations of this paper are as follows: (1) The learned success prediction model relies on a heuristic pick generator that outputs candidate picks. Developing a learning-based, robust pick generator is an interesting direction that can build on top of the demonstrated approach. (2) The extensive experiments presented in this paper are focused on the setting of a single robotic arm with a multi-suction end-effector, and do not include other embodiments, such as multi-arm settings or other end-effector types, such as pinch gripper or soft deformable gripper. (3) This paper focuses on the highly-challenging, open-set item picking challenge in a industrial warehouse setting. The results are not tested in other environments, such as a household or hospital setting.

Acknowledgments

We want to thank everyone at Amazon Robotics for providing helpful insights and support in many aspects throughout the research. We want to especially thank Azarakhsh Keipour, Mostafa Hussein for providing critical feedback with an early version of the paper.

References

[1] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal Multi-task Masked Autoencoders. In *European Conference on Computer Vision*, pages 348–367, 2022. doi: 10.1007/978-3-031-19836-6_20. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136970341.pdf.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[3] Hanwen Cao, Hao-Shu Fang, Wenhai Liu, and Cewu Lu. SuctionNet-1Billion: A Large-Scale Benchmark for Suction Grasping. *IEEE Robotics and Automation Letters*, 6 (4):8718–8725, 2021. doi: 10.1109/LRA.2021.3115406. URL https://ieeexplore.ieee.org/document/9547830.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, Oct 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://ieeexplore.ieee.org/document/9709990.

[5] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

[6] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriquez, J. M. Romano, and P. R. Wurman. Analysis and Observations From the First Amazon Picking Challenge. *IEEE Transactions on Automation Science and Engineering*, 15:172–188, January 2018. doi: 10.1109/TASE.2016.2600527. URL https://ieeexplore.ieee.org/document/7583659.

[7] M. Dogar and S. S. Srinivasa. A Planning Framework for Non-Prehensile Manipulation under Clutter and Uncertainty. *Autonomous Robots*, 33:217–236, Jun 2012. doi: 10.1007/s10514-012-9306-z. URL https://link.springer.com/article/10.1007/s10514-012-9306-z.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations (ICLR 2021)*, May 2021. URL https://iclr.cc/virtual/2021/poster/3013.

[9] 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 11441–11450, 2020. doi: 10.1109/CVPR42600.2020.01146. URL https://ieeexplore.ieee.org/document/9156992.

[10] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. AnyGrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023. doi: 10.1109/TRO.2023.3281153. URL https://ieeexplore.ieee.org/document/10167687.

[11] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[12] K. Goldberg and M. Mason. Bayesian Grasping. In *IEEE International Conference on Robotics and Automation*, pages 1264–1269 vol.2, Cincinnati, Ohio, USA, August 1990. doi: 10.1109/ROBOT.1990.126172. URL https://ieeexplore.ieee.org/document/126172.

[13] M. Gupta and G. S. Sukhatme. Using manipulation primitives for brick sorting in clutter. In *IEEE International Conference on Robotics and Automation*, pages 3883–3889, Saint Paul, Minnesota, USA, May 2012. doi: 10.1109/ICRA.2012.6224787. URL https://ieeexplore.ieee.org/document/6224787.

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked Autoencoders Are Scalable Vision Learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, Jun 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://ieeexplore.ieee.org/document/9879206.

[15] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones. Contact-reactive grasping of objects with partial shape information. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235, Taipei, Taiwan, Dec 2010. doi: 10.1109/IROS.2010.5649494. URL https://ieeexplore.ieee.org/document/

5649494.

[16] Ashutosh Saxena Ian Lenz, Honglak Lee. Deep Learning for Detecting Robotic Grasps. In *Proceedings of Robotics: Science and Systems IX (RSS 2013)*, pages 1–8, Berlin, Germany, June 2013. doi: 10.15607/RSS.2013.IX.012. URL https://www.roboticsproceedings.org/rss09/p12.html.

[17] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier J. H'enaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A General Architecture for Structured Inputs & Outputs. In *International Conference on Learning Representations*, April 2022. URL https://iclr.cc/virtual/2022/spotlight/6270.

[18] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Proceedings of Robotics: Science and Systems XIX (RSS 2023)*, pages 1–30, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.032. URL https://www.roboticsproceedings.org/rss19/p032.html.

[19] Sergey Levine, Peter Pastor Sampedro, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018. doi: 10.1177/0278364917710318. URL https://doi.org/10.1177/0278364917710318.

[20] Shuai Li, Azarakhsh Keipour, Kevin Jamieson, Nicolas Hudson, Charles Swan, and Kostas Bekris. Demonstrating Large-Scale Package Manipulation via Learned Metrics of Pick Success. In *Proceedings of Robotics: Science and Systems XIX (RSS 2023)*, pages 1–11, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.023. URL https://www.roboticsproceedings.org/rss19/p023.html.

[21] Jeffrey Mahler and Ken Goldberg. Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 515–524, Nov 2017. URL https://proceedings.mlr.press/v78/mahler17a.html.

[22] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In *Proceedings of Robotics: Science and Systems XIII (RSS 2017)*, pages 1–10, Cambridge, Massachusetts, USA, July 2017. doi: 10.15607/RSS.2017.XIII.058. URL https://www.roboticsproceedings.org/rss13/p58.html.

[23] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5620–5627, Brisbane, QLD, Australia, May 2018. doi: 10.1109/ICRA.2018.8460887. URL https://ieeexplore.ieee.org/document/8460887.

[24] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019. doi: 10.1126/scirobotics.aau4984. URL https://www.science.org/doi/abs/10.1126/scirobotics.aau4984.

[25] Matthew T Mason. *Mechanics of robotic manipulation*. MIT Press, Cambridge, Massachusetts, USA, 2001. ISBN 0262133962. URL https://mitpress.mit.edu/9780262133968/.

[26] A. Miller and P. K. Allen. Graspit! A Versatile Simulator for Robotic Grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, Dec. 2004. doi: 10.1109/MRA.2004.1371616. URL https://ieeexplore.ieee.org/document/1371616.

[27] R.M. Murray, Z. Li, S.S. Sastry, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Taylor & Francis, 1994. ISBN 9781315136370. doi: 10.1201/9781315136370. URL https://www.taylorfrancis.com/books/mono/10.1201/9781315136370.

[28] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of The 6th Conference on Robot Learning*, pages 892–909, Auckland, New Zealand, December 2022. URL https://proceedings.mlr.press/v205/nair23a.html.

[29] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *Proceedings of the 39th International Conference on Machine Learning*, pages 17359–17371, Baltimore, Maryland, USA, July 2022. URL https://proceedings.mlr.press/v162/parisi22a.html.

[30] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj.

[31] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. In *Proceedings of The 7th Conference on Robot Learning*, pages 1–21, Atlanta, Georgia, USA, Nov 2023. URL https://proceedings.mlr.press/v229/stone23a.html.

[32] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024.

[33] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. doi: 10.48550/arXiv.2203.06173. URL https://arxiv.org/abs/2203.06173.

[34] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2T2: Multi-Task Masked Transformer for Object-centric Pick and Place. In *Proceedings of The 7th Conference on Robot Learning*, pages 1–12, Atlanta, Georgia, USA, Nov 2023. URL https://proceedings.mlr.press/v229/yuan23a/yuan23a.pdf.

APPENDIX

Additional experiments and details:

1) Scaling Trend with Finetune Dataset Size: In Figure 13 we show how performance scales with different finetune dataset sizes for the demonstrated approach as well as alternative representation learning methods. For a fair comparison, for each method, we again pretrain for 100 epoches on in-domain data, and keep all finetuning settings and hyperparameters exactly the same. We also use RGB as the only input during finetuning stage, so that MultiMAE does not have an advantage from using multi-modal data at finetune stage.

The result shows that MAE has slightly weaker performance than DINO and MOCOv3, DINO is slightly weaker than MOCOv3 but start to become better at a larger data size, and MultiMAE consistently outperform the other methods at all finetune data ratios (we tried use 1%, 5%, 20%, 50% and 100% of the finetune data).

Extensive discussions on the scaling behavior of MAE compared to other representation learning methods can also be found in He et al. [14] and Bachmann et al. [1]. Note that our approach can work with any visual encoders. So we can easily replace MultiMAE with a different and more performant multi-modal representation learning approach in the future.

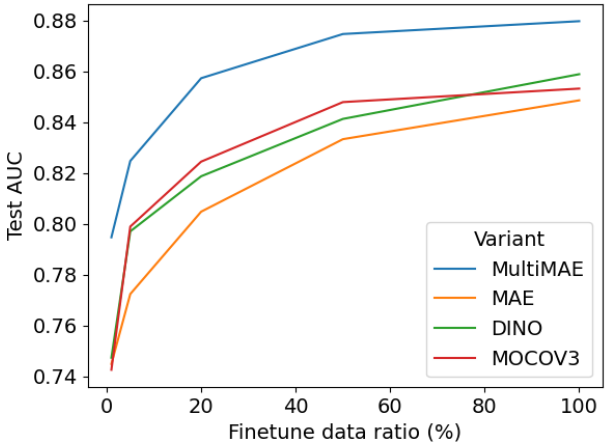


Fig. 13: Scaling trend of different representation learning methods with respect to finetune data sizes.

2) Effect of Stacking Modalities: We also test different representation learning objectives on multi-modal data by channel-wise stacking the R, G, B, D, S inputs.

We conducted experiments where we stack these inputs into 5 channels, and finetune MultiMAE, MAE, DINO and MoCoV3 under the same setting as in Table 10 (in-domain pretraining using standard settings for 100 epochs, then finetune with the stacked image and pick parameters). The results are shown in Table XI. MultiMAE is still the best performing model. It is better than MultiMAE with RGB only ($88.52 > 87.97$), but worse than the recommended MultiMAE with 3 separate visual modalities ($88.52 < 90.8$). This shows stacking visual modalities is better than using a single modality, but weaker than using each modality separately. The stacking does make inference faster (from 687 ms to 194 ms, batch of 128).

TABLE XI: Performance comparison when stacking RGB, depth and semantic inputs.

Objective:	MultiMAE	MAE	DINO	MoCoV3
Test AUC:	88.52	85.53	85.63	84.98

3) Effect of Image Resolution: We now evaluate the impact of different image resolutions. The results are shown in Table XII. We used 224x224 to achieve strong performance with relatively low latency, which is important for industrial systems. Resolution and the number of modalities impact the number of tokens, and, thus, affect latency. Further hyperparameter tuning may improve performance for higher resolutions.

Resolution	96×96	144×144	224×224	304×304	400×400
AUC	89.43	90.18	90.60	90.54	90.13
Time (ms)	152 ms	222 ms	687 ms	1,780 ms	4,320 ms

TABLE XII: Model performance and inference time for different resolutions (batch size = 128)

4) VAE Representations: we also performed experiments testing a BEiT [2] model using VAEs as tokenizers. We downloaded a pretrained checkpoint and finetuned using the same setting. BEiT given RGB only results in 85.61 test AUC, and 87.70 given stacked channels. This is slightly better than MAE pretrained on generic data with stacked channels (85.98), but weaker than MultiMAE pretrained with in-domain data and separate visual modalities (90.60). This supports the paper's finding that the MAE objective is not the most important factor. It is the in-domain pretraining, encoder finetuning and the multi-modality that improve performance. MultiMAE can easily be replaced with other multi-modal encoders and this may eventually result in better performance.

5) Pretrain on Items, Finetune on Packages: We also tried pretrain on item picking data and then run pick success training on package data. This creates a significant distribution shift. We achieve 87.40 test AUC when further finetuning the encoder on package picking data but 85.43 when freezing the encoder. This is a lower score than pretraining on package picking data (88.28) but higher than without any pretraining (84.54). This shows that: (1) the representation learned from multimodal pretraining on a different data distribution can still be useful; and (2) further supports that in-domain pretraining-finetuning improves performance for challenging industrial settings.

6) Latency: On a single NVIDIA A10G GPU with 12 AMD EPYC 7R32 CPU cores for dataloader workers, consider a batch of 128 test data examples (pick candidates) to evaluate, it takes about 197 ms if the input has one visual modality, 409 ms for two, and 687 ms for three modalities. Note that the latency will be lower if there are a smaller number of pick candidates. Using different resolutions also affect latency. We plan to explore optimizing this further in future work.