2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Learn to Grasp with Less Supervision: A Data-Efficient Maximum Likelihood Grasp Sampling Loss

Xinghao Zhu[1], Yefan Zhou[1], Yongxiang Fan[2], Lingfeng Sun[1], Jianyu Chen[1], and Masayoshi Tomizuka[1]

***Abstract*— Robotic grasping for a diverse set of objects is essential in many robot manipulation tasks. One promising approach is to learn deep grasping models from large training datasets of object images and grasp labels. However, empirical grasping datasets are typically sparsely labeled (i.e., a small number of successful grasp labels* in each image). The data sparsity issue can lead to insufficient supervision and false-negative labels, and thus results in poor learning results. This paper proposes a Maximum Likelihood Grasp Sampling Loss (MLGSL) to tackle the data sparsity issue. The proposed method supposes that successful grasps are stochastically sampled from the predicted grasp distribution and maximizes the observing likelihood. MLGSL is utilized for training a fully convolutional network that generates thousands of grasps simultaneously. Training results suggest that models based on MLGSL can learn to grasp with datasets composing of 2 labels per image. Compared to previous works, which require training datasets of 16 labels per image, MLGSL is 8× more data-efficient. Meanwhile, physical robot experiments demonstrate an equivalent performance at a 90.7% grasp success rate on household objects. Codes and videos are available at [1].**

I. Introduction

Robotic grasping in unstructured environments can benefit applications from warehouse automation to home servicing. Supervised machine learning approaches have demonstrated promising results in planning grasps under various uncertainties. One kind of approach is to sample grasp candidates and evaluate [2]–[11]. Such two-step methods, however, might be time-consuming at execution.

An alternative is to train grasp planning models in end-to-end manners [12]–[21]. These approaches directly generate grasps and have a shorter planning time. While end-to-end models typically require densely labeled ground-truth samples in training, existing empirical datasets, generated by physical execution or human labeling, only include scarce labels for each image [22], [23]. Though many other good grasps exist, they are unfortunately not labeled. The sparsity issue can lead to a lot of false negatives in training end-to-end models. Previous works [14]–[18] assume that unlabeled points are not valid and treat them as failed grasps. However, robust grasps may still exist in these areas but are mistakenly labeled as negatives. These wrong labels can harm the learning results.

This paper proposes a maximum likelihood grasp sampling loss (MLGSL) to improve data efficiency in training grasp planners with single-view depth images. The main difference between MLGSL and other works is the assumption toward unlabeled regions. Previous works [14]–[18] regard unlabeled areas as failures and estimate the success rate for each pixel. In contrast, we leave unlabeled pixels intact and solely apply supervisions with labeled grasps. Specifically, we propose a stochastic grasp selection process to estimate the likelihood for each pixel to be the best grasp point and maximize such likelihood for labeled grasps. Since no labels are generated for unlabeled areas, MLGSL can reduce the false-negative problem in training grasp planners. Training results demonstrate that models with MLGSL can learn to grasp with fewer labels compared to previous works [15]–[17], [24], while physical experiments show a similar grasp success rate at 90.7% (Fig. 1). Moreover, this paper demonstrates that attention mechanisms do not contribute to dense grasp plannings. Furthermore, a dataset is constructed with multiple-object scenes and collision-free grasp labels to improve the performance in clutter.

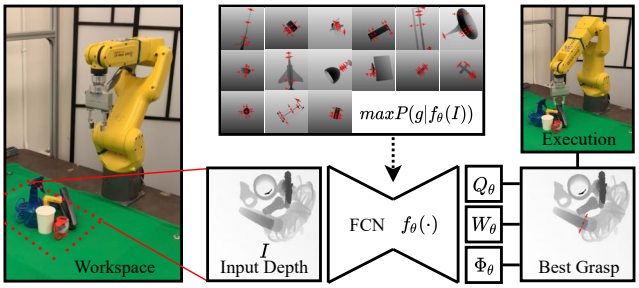


Fig. 1. Grasp planning and execution pipeline. When an object is presented in the workspace, a stereo camera captures a depth image; a trained generative model $f_\theta(\cdot)$ rapidly computes grasp configuration maps Q_θ, W_θ, and Φ_θ. The best grasp is generated based on configuration maps and executed with the robot manipulator. The grasp model is trained offline with empirical datasets and the proposed loss function.

Related works are introduced in Section II. Section III presents the proposed approach. Training and experiments are presented in Section IV and V. Section VI concludes the paper and suggests the future work.

II. Related Works

A. Grasp Planning and Datasets

Analytic and machine learning methods have been studied to plan grasps across various objects. Existing analytic methods [25]–[27] can be used to search for optimal grasps. These techniques, however, are less robust in practice due to perceptional limitations and unseen geometries.

[1]Mechanical Systems Control Lab, UC Berkeley, CA USA. {zhuxh, yefan0726, lingfengsun, jianyuchen, tomizuka}@berkeley.edu
[2]FANUC Advanced Research Lab, FANUC America Corporation, CA USA. Yongxiang.Fan@fanucamerica.com
*Labels refer to marking the image to indicate a successful robotic grasp.

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811685

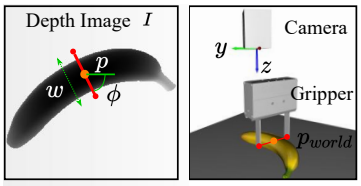
Machine learning is an alternative approach to plan grasps. Current methods show that it is preferable to learn grasp quality functions and optimize them at the runtime [2], [4]–[11]. However, sampling or optimization makes the algorithm time-consuming and requires pre-defined heuristics. Other end-to-end approaches propose to infer grasp poses from the raw input directly. These approaches can be generative or discriminative. Generative models [12]–[16], [28]–[31] perform grasp pose regression and grasp quality assessment simultaneously. Discriminative approaches [17], [19] use fully convolutional networks to evaluate thousands of grasps simultaneously without direct sampling. Reinforcement learning has also been introduced in grasp planning [24], [32]. Q learning is leveraged to estimate the state-action qualities with iterative updates.

Machine learning approaches typically require large datasets consisting of sensor readings and ground-truth grasp labels. Synthetic datasets can be rapidly generated using analytic quality metrics and simulated sensors [2], [3], [13]. However, synthetic data can cause robustness issues due to the simulation-to-reality (sim-to-real) gap. Specifically, analytic metrics might disagree with physical grasp results in complex circumstances. Alternatively, empirical methods collect data from human labeling or grasp execution with correlations of physical success [4], [5], [22], [23]. Nevertheless, empirical data can be expensive to acquire and are typically sparsely labeled (i.e., inadequate grasp labels in each image). This paper chooses to use empirical datasets to avoid the sim-to-real gap.

B. Dense Grasp Planning

Recent works leverage dense pixel-wise evaluation and regression for grasp planning. These approaches utilize Fully Convolutional Networks (FCNs) to evaluate millions of grasps in parallel. FCNs in [15], [16] predict the grasp success rate and generate grasp configurations for each pixel. In [17], [19], FCNs are trained to predict the grasp success rate for pre-defined grasp primitive actions.

During training, FCNs require sufficient labels to perform the image-to-image learning [17]. However, existing datasets typically include scarce grasp labels for each image [22], [23]. Some works have been proposed to resolve the sparsity issue. A label generation method is proposed in [15], [16], which assumes that grasps close to successful labels are robust and treats unlabeled grasps as failures. However, this approach introduces both false negatives and false positives. On the one hand, robust grasps might be away from existing labels and are wrongly labeled as negatives. On the other hand, generated grasps that are close to existing successful labels might be unstable. In [19], [21], networks are converted from discriminative evaluators according to the injective mapping between convolutional and fully connected layers. A grasp-push policy represented by FCNs is trained in [24] with Q learnings. Although these methods eliminate the need for dense labels, they still require large datasets or rollouts. This paper proposes a loss function capable of training the FCN from scratch with inadequate labels.



(a) (b)

Fig. 2. Grasp Representation $g = (p, \phi, w)$. The planar pose (p, ϕ, w) in (a) represents the grasp's center position, orientation, and width in the image. Grasp g in (a) has quality $q = 1$ since it is a successful grasp. g is executed perpendicular to the image plane at point p_{world} in the Cartesian frame as shown in (b), where p_{world} is p in the world frame. The gripper further moves ϵ cm below p_{world} in the direction of the camera's z-axis.

C. Attention Mechanism

Attention mechanisms have achieved promising results in computer vision [33]–[35]. Previous works have introduced spatial attention mechanism (SAM) to robotics for interest-region extractions and feature reductions. In [36], [37], hierarchical SAM is employed to constrain sampling in reinforcement learning (RL) and demonstrated improved efficiency in action-space sampling. SAM is used in [38], [39] to detect grasps in clutter, achieving improved grasping success rate with RL. However, experiments in this paper demonstrate that attention mechanisms do not contribute to dense grasp planning with FCNs.

III. Grasp Planning with Maximum Likelihood Grasp Sampling Loss

This section first introduces notations of the problem. The proposed loss function is then illustrated and compared with previous works. Finally, the network architectures and datasets are presented.

A. Notations

The grasp planning problem is defined as detecting a grasp that allows the robot to pick up objects. Moreover, no explicit knowledge of the object is given beyond camera readings.

1) Grasp: Let $I \in \mathbb{R}^{\mathcal{H} \times \mathcal{W}}$ define a given depth image with height $\mathcal{H}$ and width $\mathcal{W}$. The i-th grasp is defined in the image I and denoted by $g_i = (p_i, \phi_i, w_i)$, where p_i is a pixel in the image representing the grasp center. $\phi_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the gripper's rotation, and $w_i \in [0, 150]$ pixels is the gripper's width in the image (Fig. 2). Each grasp has a quality measurement $q_i \in [0, 1]$ that indicates the grasp success rate.

2) Grasp Labels: Empirical datasets contain k success grasp labels $\tilde{g}$ for each image, where $\tilde{g} = \{\tilde{g_i}\}$ for $i \in [1, ..., k]$ and $k \ll \mathcal{H} \times \mathcal{W}$. Note $\tilde{q_i} = 1$ since they are guaranteed to succeed.

3) Grasp Configuration Maps: Similar to [15], we refer to the set of grasps in the image as the grasp configuration map $G = (Q, \Phi, W) \in \mathbb{R}^{3 \times \mathcal{H} \times \mathcal{W}}$, where $Q, \Phi, W \in \mathbb{R}^{\mathcal{H} \times \mathcal{W}}$ contain values of q_i, ϕ_i, w_i at each pixel of I. In practice, we use two components $\Phi_s = \sin(2\Phi), \Phi_c = \cos(2\Phi)$ for Φ to resolve the symmetricity of antipodal grasps.

4) Grasp Planning Models: This paper uses a grasp neural network $f_\theta(\cdot)$ to approximate the dense grasp configuration maps, $\hat{G} = (\hat{Q}, \hat{\Phi}, \hat{W}) = f_\theta(I)$. Predicted grasps are $\hat{g} = \{\hat{g_i}\}$ for $i \in [1, ..., \mathcal{H} \times \mathcal{W}]$ at each pixel.

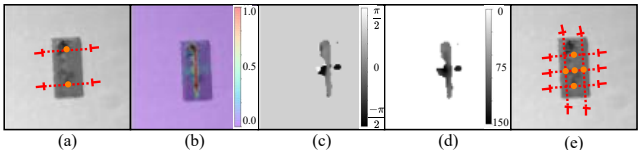


Fig. 3. (a) shows the input depth image I with $k = 2$ success grasp labels $\tilde{g}$ (red). (b-d) show predicted grasp configuration maps $\hat{G} = (\hat{Q}, \hat{\Phi}, \hat{W})$ respectively. Different colors represent different values as in the color bar. (e) shows 5 selected grasps based on grasp configuration maps $\hat{G}$.

B. Maximum Likelihood Grasp Sampling Loss

Following definitions in Section III-A, this paper uses two grasp notations for each image I: the predicted grasp configuration maps $\hat{G}$ and the grasp labels $\tilde{g}$. The former maps are densely predicted by the network, which have values at each pixel. In contrast, the latter $\tilde{g}$ only has k successful grasp labels, as shown in Fig. 3. The objective of the grasp model is to estimate grasp configuration maps by $\hat{G} = f_\theta(I)$, which yield the highest grasp success rate.

At execution, grasps are selected based on the predictions $\hat{G}$. First, a grasp pixel p is drawn from the image conditioned on the quality map $\hat{Q}$. Categorical distribution is used to describe the sampling process. Each pixel is a category with a probability proportional to the predicted quality $\hat{q}_i$. This suggests that for each grasp pixel, the higher the grasp quality is, the more chance it can be selected to execute. Then, corresponding rotation $\hat{\phi}$ and width $\hat{w}$ are selected from dense prediction maps $\hat{\Phi}$ and $\hat{W}$ at pixel p.

This paper assumes that success grasp labels $\tilde{g}$ are chosen from the prediction maps $\hat{G}$ based on the above selection procedure. To optimize the learning performance, we propose to maximize the chance of observing $\tilde{g}$ from $\hat{G}$ using a maximum likelihood estimation (MLE).

The grasp selection process is first modeled mathematically. The grasp configuration $\boldsymbol{p}, \boldsymbol{\phi}, \boldsymbol{w}$ are regarded as random variables. The pixel location $\boldsymbol{p}$ is discrete and takes on a value in the pixel indexes, i.e., $p \in [1, ..., \mathcal{H} \times \mathcal{W}]$. The probability mass function of $\boldsymbol{p}$ is a Categorical distribution defined as $P(p|\hat{Q})$, representing the chance that the grasp at pixel p is selected for execution. The rotation $\boldsymbol{\phi}$ and the width $\boldsymbol{w}$ are real values in the configuration space, whose density functions are Gaussians as $P(\phi|p, \hat{\Phi})$ and $P(w|p, \hat{W})$.

Based on the above model, the probability of observing a grasp label $\tilde{g}_i$ from the prediction map $\hat{G}$ is $P(\tilde{g}_i|\hat{G})$. The weights of the network θ are trained to maximize such a probability:

$$\begin{aligned}
&\max_\theta && P(\tilde{g}_i|\hat{G}) \\
=&\max_\theta && P(\tilde{p}_i, \tilde{\phi}_i, \tilde{w}_i|\hat{G}) && \text{(1a)} \\
=&\max_\theta && P(\tilde{p}_i|\hat{G}) \cdot P(\tilde{\phi}_i|\tilde{p}_i, \hat{G}) \cdot P(\tilde{w}_i|\tilde{\phi}_i, \tilde{p}_i, \hat{G}) && \text{(1b)} \\
=&\max_\theta && P(\tilde{p}_i|\hat{Q}) \cdot P(\tilde{\phi}_i|\tilde{p}_i, \hat{\Phi}) \cdot P(\tilde{w}_i|\tilde{p}_i, \hat{W}) && \text{(1c)}
\end{aligned}$$

In (1a), $\tilde{g}_i$ is replaced with grasp configurations as in III-A. Chain rules are applied to obtain (1b). Equation (1c) is modified based on the grasp selection model. The first term maximizes the chance to select a successful grasp at pixel $\tilde{p}_i$ for execution. The second and third terms maximize the chance of observing $\tilde{\phi}_i$ and $\tilde{w}_i$ at pixel $\tilde{p}_i$ in $\hat{\Phi}$ and $\hat{W}$.

Equation (1c) is then jointly optimized for all grasp labels $\tilde{g}$. This paper further assumes that each label is independent:

$$\begin{aligned}
&\max_\theta && \prod_{i=1}^{k} P(\tilde{g}_i|\hat{G}) && \text{(2a)} \\
\approx&\max_\theta && \prod_{i=1}^{k} P(\tilde{p}_i|\hat{Q}) \cdot P(\tilde{\phi}_i|\tilde{p}_i, \hat{\Phi}) \cdot P(\tilde{w}_i|\tilde{p}_i, \hat{W}) && \text{(2b)} \\
\propto&\max_\theta && \sum_{i=1}^{k} \log P(\tilde{p}_i|\hat{Q}) + \log P(\tilde{\phi}_i|\tilde{p}_i, \hat{\Phi}) && \text{(2c)} \\
& && \quad + \log P(\tilde{w}_i|\tilde{p}_i, \hat{W}) \\
=&\max_\theta && \sum_{i=1}^{k} \log P(\tilde{p}_i|\hat{Q}) - \text{MSE}(\hat{\phi}_i, \tilde{\phi}_i) && \text{(2d)} \\
& && \quad - \text{MSE}(\hat{w}_i, \tilde{w}_i)
\end{aligned}$$

Three terms are maximized in (2d). The first term maximizes the chance of selecting robust grasps to execute. The second and third terms are modified from (2c) to minimize the mean square error (MSE) between predictions and labels, as [40] suggests. Such modifications stabilize the training process without loss of accuracy.

From all above, the grasp model $f_\theta(\cdot)$ is trained as:

$$\theta = \underset{\theta}{\text{argmin}}\ \mathcal{L}(\hat{G}, \tilde{g}) \tag{3}$$

where

$$\begin{aligned}
\hat{G} &= (\hat{Q}, \hat{\Phi}, \hat{W}) = f_\theta(I) \\
\mathcal{L}(\hat{G}, \tilde{g}) &= \sum_{i=1}^{k} -\log P(\tilde{p}_i|\hat{Q}) + \text{MSE}(\hat{\phi}_i, \tilde{\phi}_i) \\
&\quad + \text{MSE}(\hat{w}_i, \tilde{w}_i)
\end{aligned}$$

The maximum likelihood grasp sampling loss (MLGSL) in (3) minimizes a pixel selection loss and two pixel-wise regression losses. The main difference between MLGSL and others is MLGSL only uses existing labels. Previous works [15]–[17], [19], [24] regard unlabeled pixels as negatives and use regression or spatial cross-entropy losses to estimate the quality for the entire image. In contrast, MLGSL only computes losses for pixels that have successful grasp labels (i.e., only at $\tilde{p}_i$), and it directly predicts whether a pixel is the best grasp point. Specifically, MLGSL estimates the likelihood that a pixel is the most robust grasp with the grasp selection model and maximizes the likelihood for successful grasp labels. As suggested in [41], [42], estimating grasp qualities for each pixel is challenging due to only scarce label exists and the false-negative problem. By contrast, MLGSL directly optimizes the likelihood for each grasp and is more reliable and easier to converge [42].

C. Model Architectures

The grasp planning model is used to predict dense grasp configuration maps $\hat{G}$, consisting of $\hat{Q}, \hat{\Phi}_c, \hat{\Phi}_s, \hat{W}$. Note that angle maps Φ is calculated by $\Phi = \frac{1}{2}\tan^{-1}\frac{\Phi_s}{\Phi_c}$. The model

uses a fully convolutional topology, similar to [16]. The architecture includes four downsampling layers, two dilated layers, and two upsampling layers. Downsampling layers use kernel size of $[11, 5, 5, 5]$ respectively, activated by ReLU and max-pooling. Two dilated layers apply $[5, 5]$ kernels with dilation $[2, 4]$. Upsampling layers employ transpose convolutional kernels with size 3 and striding 2.

SAM blocks are also added as described in [34]. SAM utilizes both max-pooling and average-pooling along the channel axis and forwards them to a convolution layer. Outputs are then integrated into input features.

D. Dataset

The network is trained with a single object dataset and a cluttered object dataset.

This paper directly adopt Jacquard [22] as the *single object dataset*. The dataset contains more than 50k images of 11k objects and 1 million success grasp labels. We apply random rotation and zoom to images and resize them to 300×300.

The *cluttered object dataset* is generated based on Jacquard. A few images are randomly selected from the single object dataset and fused into a cluttered sample. Before fusing, each single object image is segmented, rotated, zoomed, and translated in the image plane. Success grasp labels are then merged and pruned according to collision constraints. Since data in Jacquard includes images from different viewpoints, such operation can reflect the geometry of cluttered scenes.

IV. Training Results

We trained a series of models to test the proposed approach. The goals of the experiments are three-fold: 1) to demonstrate that the proposed loss function can increase the grasp performance with fewer labels and samples, 2) to determine whether attention modules help in learning dense grasp configurations, and 3) to inspect the collision-avoidance ability in cluttered scenes.

A. Evaluation Metrics

Three metrics are utilized to evaluate models' performance: predictions' success rate and predictions' accuracy and recall. For prediction success rate, a predicted grasp $\hat{g}_i$ is considered success if

$$\exists \tilde{g}_j \in \tilde{g},\ \ni \text{IoU}(\tilde{g}_j, \hat{g}_i) \geq 25\% \text{ and } |\tilde{\phi}_j, \hat{\phi}_i| \leq 30^\circ$$

where $\text{IoU}(\cdot)$ represents the intersection over union ratio between two grasps. $\tilde{g}$ is the set of success grasp labels. This paper selects top one (Top-1) and top five (Top-5) grasps to measure the success rate as in [15], [16], [22].

Besides the grasp success rate, we measure predictions' accuracy and recall. A grasp quality discriminator [2] was pre-trained to evaluate the robustness of grasps. For each validation data, 100 predicted grasps $\hat{g}_i$ are first uniformly sampled with predicted quality $\hat{q}_i$. Then, the discriminator evaluates quality for $\hat{g}_i$, obtaining ground-truth quality label q_i. Prediction accuracy and recall are measured based on q_i and $\hat{q}_i$ for $i \in [1, ..., 100]$.

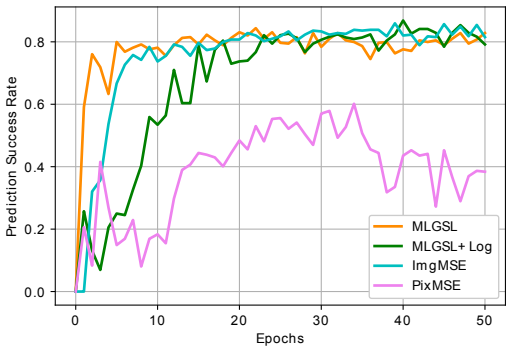


Fig. 4. Comparing the Top-1 prediction success rate of MLGSL with baseline methods. Models are trained with densely-labeled datasets (16 labels per image).

TABLE I
Training performance of MLGSL and baselines (Mean %)

Method	Top-1	Top-5	Accuracy	Recall
MLGSL	82.8	91.0	80.2	75.7
MLGSL (2 labels)	81.6	90.3	77.3	73.4
ImgMSE [16]	82.5	89.7	85.2	92.3
ImgMSE [16] (2 labels)	42.4	45.2	41.9	17.4

B. Baseline Methods

We compare the training performance of MLGSL to the following baseline approaches:

1) Image-wise MSE (ImgMSE): ImgMSE is introduced in [15] that use the same prediction maps $\hat{G}$ as ours. This baseline augments successful grasp labels to $\bar{G}$ as in Section II-B. The loss used to train the model is

$$\mathcal{L}_{\text{ImgMSE}} = \text{MSE}(\hat{Q}, \bar{Q}) + \text{MSE}(\hat{\Phi}, \bar{\Phi}) + \text{MSE}(\hat{W}, \bar{W})$$

2) Maximum Likelihood Sampling with LogMSE (MLGSL+Log): MLGSL+Log can be derived from (2c), which uses $\text{MSE}(\cdot)$ to replace $P(\cdot)$, i.e.

$$\mathcal{L}_{\text{MLGSL+Log}} = \sum_{i=1}^{k} -\log P(\tilde{p_i}|\hat{Q}) + \log \text{MSE}(\hat{\phi}_i, \tilde{\phi}_i) + \log \text{MSE}(\hat{w}_i, \tilde{w}_i)$$

3) Pixel-wise MSE (PixMSE): PixMSE applies supervisions solely on labeled pixels with MSE loss, i.e.

$$\mathcal{L}_{\text{PixMSE}} = \sum_{i=1}^{k} \text{MSE}(\hat{q}_i, \tilde{q}_i) + \text{MSE}(\hat{\phi}_i, \tilde{\phi}_i) + \text{MSE}(\hat{w}_i, \tilde{w}_i)$$

C. Results

For comparisons, we train variant models with different loss function designs and architectures. Each model is trained with different seeds for 50 epochs to select the best one.

1) Baseline Comparisons: Our first experiment compares MLGSL to three baseline methods with a single object dataset, in which each training sample includes 16 success grasp labels. Top-1 and Top-5 prediction success rates are shown in Fig. 4 and Table I. We see that MLGSL has similar performances compared to previous ImgMSE, while MLGSL shows a higher convergency rate at first a few epochs. MLGSL with logarithm converges to a similar point as the

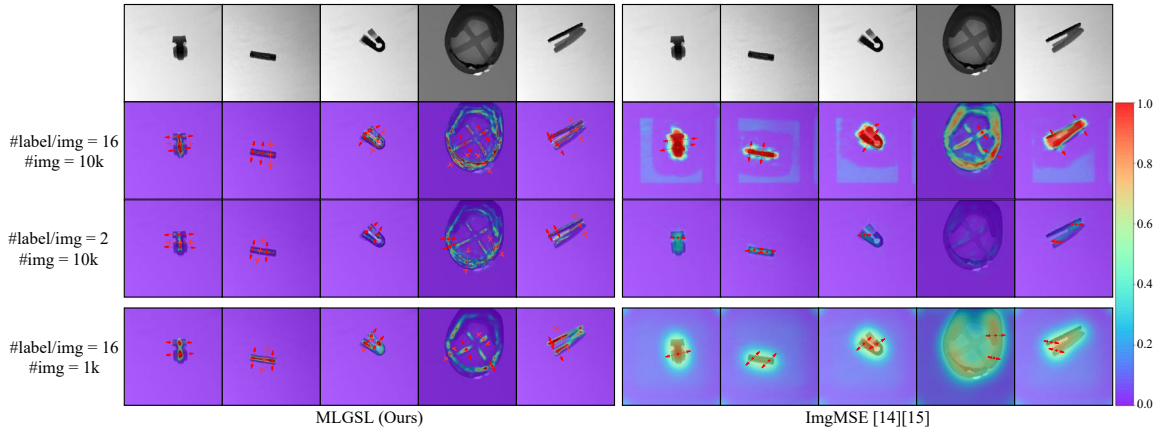


Fig. 5. Predicted grasp distributions with variant models. Predicted grasp qualities are painted as heatmaps with color listed in the right-sidebar. Detected grasps are labeled with red lines in each image. (First row) input depth images, (Left) results from models trained with MLGSL, (Right) results from models trained with ImgMSE, (Second row) results from datasets consisting of 16 labels per image and 10k training images, (Third row) results from datasets consisting of 2 labels per image and 10k training images, (Bottom) results from datasets consisting of 16 labels per image and 1k training images.

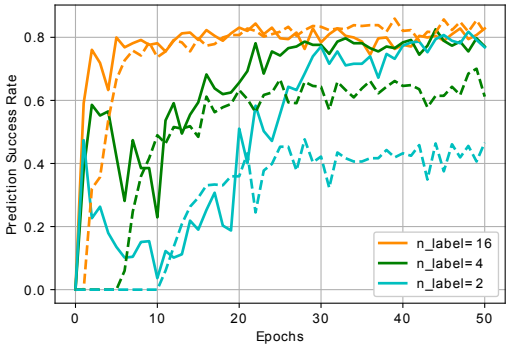


Fig. 6. Comparing the Top-1 prediction success rate of MLGSL to ImgMSE with different numbers of labels (n_label). Success grasp labels are down-sampled to $[2, 4]$ for each training image. Solid lines indicate models' performance trained with MLGSL, and dashed lines indicate that trained with ImgMSE.

previous two methods but with a slower rate, which might occur because the $\log(\cdot)$ operation lowers the gradient in each training step. PixMSE performs the worst among the four approaches. This likely due to it only applies supervisions on specific pixels, resulting in unbounded other areas.

We also compare MLGSL to ImgMSE on prediction accuracy and recall. Results are shown in Table I. It is interesting to observe that models trained with MLGSL have lower accuracy and recall. To seek reasons for such phenomenon, we plot several predicted $\hat{G}$ in the second rows of Fig. 5. As can be seen, models trained with MLGSL have a conservative estimation of the grasp quality and prefer to grasp each object's center. This behavior leads to that only a small area has high quality (red area in Fig. 5). When measuring accuracy and recall, grasps are uniformly sampled in the image (Section IV-A), making many of them out of the high-quality zone. This fact then produces false-negative predictions and low accuracy and recall.

2) Less Training Labels per Sample: We then investigate whether our method can learn grasping with fewer labels. For this study, we down-sample success grasp labels to $[2, 4]$ in each training data and still use all labels for validation. It is a more difficult setting; the grasp planning model learns to effect change only through inadequate demonstrations. We report results in Fig. 6 and Table I. In the figure, models trained with MLGSL are evaluated with the Top-1 prediction success rate, indicated by solid lines. Dashed lines indicate performances of models trained with ImgMSE.

From these results, we see that MLGSL is capable of learning to grasp with 2 labels per image, achieving prediction success rates at 81.6% for Top-1 and 90.3% for Top-5, which is similar to models trained with 16 labels. The third row in Fig. 5 shows predicted $\hat{G}$ by models trained with 2 labels per image. We also notice that ImgMSE under-performs MLGSL in such settings. This is attributed that the label generation method used by ImgMSE generate false-negatives. Padded $\bar{G}$ may mistakenly label high-quality grasps to negatives since they are not close to existing success labels. For MLGSL, unlabeled pixels are regulated indirectly with the probabilistic objective. This procedure minimizes assumptions toward unlabeled areas, thus does not suffer from false-negatives.

3) Less Training Samples: We next train models with a smaller dataset (1k data) using MLGSL and ImgMSE. The former makes models converge to 69.8%, while the latter converges at 60.2%. The results suggest that less training data makes it harder to learn grasping strategies for both methods. However, we still observe our MLGSL outperforms ImgMSE by about 10%. We visualize prediction results at the bottom in Fig. 5. Compare to ImgMSE, MLGSL predicts more reasonable grasp distributions with less training data. The less accurate predictions from ImgMSE might due to false training labels. Models may require more data to compensate for wrong labels; thus, we observe better prediction results in previous experiments with more training data.

4) Attention Module Integration: Besides loss designs, we compare the effectiveness of attention modules by adding SAM to downsampling layers, upsampling layers, all layers,

(a) (b)

Fig. 7. (a) shows experimental setups. (b) shows objects used for single/cluttered grasping experiments.

TABLE II

REAL-WORLD PERFORMANCE OF MLGSL AND BASELINES

Method	Required Data	SR (%)	CTPG (ms)
GGCNN [15]	10k × 16	83.3 (30/36)	21
GGCNN2 [16]	10k × 16	88.9 (48/54)	21
GQCNN [2]	7m × 1	91.7 (33/36)	570
FC-GQCNN [19]	7m × 1	90.7 (49/54)	29
CGPN [13]	30k × 20	85.2 (46/54)	410
MLGSL (Ours)	10k × 2	90.7 (98/108)	21

Require data represents the size of training datasets (number of images × number of labels per image); SR and CTPG represent grasp success rate and computation time per grasp respectively.

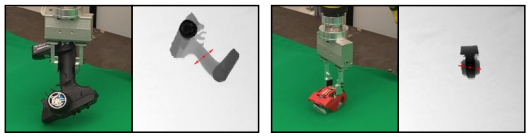
and no layers. Interestingly, results suggest that the attention module is not a contributor to the dense grasp planning. All architectures converge to $81.6 \pm 1.2\%$. This could be because the backbone FCN is simple, which do not allow SAMs to take effect. Furthermore, as suggested in [36], [37], attentions might have a similar effect to quality map $\hat{Q}$ in our models.

5) Collision-Free Datasets: Networks in [15], [16] are trained with single object datasets and directly deployed to the clutter. However, we observe collisions during grasp execution in practice. To improve this, we train and test models with collision-free cluttered datasets. The results show that the collision-free ratio improves to 84.2% when training with the collision-free dataset compared to 67.3% with the single object dataset. Although we observe a 5% down on prediction success rate, models trained with proposed datasets improve collision detection ability by 25%.

V. REAL-WORLD EXPERIMENTS

Trained models were run on a laptop with GTX1060 GPU and 2.5GHz CPU. Experimental setups are shown in Fig. 7(a). An Ensenso N35 camera was used to capture depth images. Invalid depth values were inpainted using OpenCV [43]. Robots executed the grasp with an offset $\epsilon = 1$cm from the selected grasp point along the camera's z-axis.

17 household and 1 adversarial objects were selected to test the models (Fig. 7(b)). Household objects contain items of varying sizes and shapes. Most of the items (staple, tape, cube, robots, sprayers, glue stick) appear in previous works. We used several additional objects that are deformable (cable) and perceptually challenging (thin edges on the cup, helmet, board eraser, scissor, and reflective zinc container). We also added an adversarial object [2] to verify models' robustness with a complex geometry.



(a) (b)

Fig. 8. Two failure cases. (a) shows the object slippage when the robot grasps heavy objects, and (b) shows models mistakenly generate grasps toward deformable thin covers.

We performed physical grasping experiments with two arrangements: 1) isolated single objects and 2) cluttered objects. For single object grasping, we trained a model with MLGSL, and a dataset consists of 2 labels per image. We compared our approach with five baselines [2], [13], [15], [16], [19]. Results are reported in Table II. Each object was grasped for the same time. Note that all baselines require larger well-labeled datasets to achieve listed performances.

For cluttered scenarios, objects were randomly placed inside the workspace. The robot attempted one grasp each time, and the grasped object was removed from the scene. The picking order is greedily determined to maximize the grasp success rate of overall objects and avoid collisions. This procedure continues until all objects are removed or consecutively failed five times. We ran this experiment 10 times to measure performances. Models trained with MLGSL and cluttered datasets achieved an object removal rate of 90%, compared to 70% in models trained with single datasets, mainly due to undetected collisions. Comparing to [16], we observed similar results that models trained with cluttered datasets outperform that with single datasets.

Figure 8 displays two common failures of MLGSL. One failure mode occurs when the object is heavy. The current method assumes a fixed contact force, and heavy objects can slip without a grasp force controller. The second type of failure occurs when a thin deformable layer is on top of the object's main body. It is challenging to distinguish thin layers from solid cubes in depth images. Such ambiguity tricks the model into generating unstable grasps.

VI. DISCUSSION AND CONCLUSION

This paper tackles the data sparsity issue in grasp planning with several key contributions. First, we propose a stochastic process to select grasps with dense grasp planners. Second, we present the MLGSL to train FCNs with a small empirical dataset; we show that it can match the performance of the state-of-the-art methods with fewer training data. Third, we show that the attention mechanisms are not contributing to the dense grasp planning. Lastly, we provide a grasping dataset to improve the performance of networks in clutter.

The present work also has limitations. This paper only shows experiments on low-DoF grasping tasks. However, the proposed MLGSL does not constrain grasp dimensions and thus can be generalized to high-DoF grasps. Validating MLGSL in high-DoF tasks will be our future works. Moreover, collision-free is not guaranteed with the proposed dataset; a collision-pruning module is still required in practice.

REFERENCES

[1] Webstite for paper Learn to Grasp with Less Supervision using MLGSL, https://rolandzhu.github.io/MLGSL/.

[2] J. Mahler and et al., "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *Robotics: Science and Systems (RSS)*, 2017.

[3] J. Mahler and et al., "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," *2018 ICRA*, pp. 1–8, 2018.

[4] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *International Conference on Computer Vision (ICCV)*, 2019.

[5] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6232–6238, 2020.

[6] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[7] Y. Fan, X. Zhu, and M. Tomizuka, "optimization model for planning precision grasps with multi-fingered hands," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1548–1554.

[8] Y. Fan, T. Tang, H.-C. Lin, and M. Tomizuka, "Real-time grasp planning for multi-fingered hands by finger splitting," in *2018 IROS*. IEEE, 2018, pp. 4045–4052.

[9] Y. Fan, H.-C. Lin, T. Tang, and M. Tomizuka, "Grasp planning for customized grippers by iterative surface fitting," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 28–34.

[10] H. Liang and et al., "Pointnetgpd: Detecting grasp configurations from point sets," *2019 ICRA*.

[11] Q. Lu and T. Hermans, "Modeling grasp type improves learning-based grasp planning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 784–791, 2019.

[12] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, p. 2286–2293.

[13] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka, "6-dof contrastive grasp proposal network," *2021 IEEE International Conference on Robotics and Automation (ICRA)*.

[14] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds," *2020 ICRA*, pp. 3619–3625, 2020.

[15] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," *Robotics: Science and Systems (RSS)*, 2018.

[16] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, no. 2-3.

[17] A. Zeng and et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 ICRA*, 2018, pp. 3750–3757.

[18] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[19] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, 2019.

[20] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 ICRA*, 2015, pp. 1316–1322.

[21] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IROS*, 2015, pp. 4415–4420.

[22] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IROS*, 2018, pp. 3511–3516.

[23] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724.

[24] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IROS*, 2018.

[25] M. Roa and R. Suarez, "Grasp quality measures: Review and performance," *Autonomous Robots*, vol. 38, pp. 65–88, 07 2014.

[26] N. Shafii, S. H. Kasaei, and L. S. Lopes, "Learning to grasp familiar objects using object view recognition and template matching," in *2016 IROS*, 2016, pp. 2895–2900.

[27] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.

[28] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng, "Roi-based robotic grasp detection for object overlapping scenes," in *2019 IROS*, 2019, pp. 4768–4775.

[29] S. Jiang, X. Zhao, Z. Cai, K. Xiang, and Z. Ju, "Single-grasp detection based on rotational region cnn," in *Advances in Computational Intelligence Systems*. Cham: Springer International Publishing, 2020, pp. 131–141.

[30] Z. Luo, B. Tang, S. Jiang, M. Pang, and K. Xiang, "Grasp detection based on faster region cnn," in *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2020, pp. 323–328.

[31] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[32] C. Bodnar, A. Li, K. Hausman, P. Pastor, and M. Kalakrishnan, "Quantile qt-opt for risk-aware vision-based robotic grasping," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020.

[33] L. C. et al., "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2017, pp. 6298–6306.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *European Conference on Computer Vision (ECCV)*, September 2018.

[35] O. Ulutan, A. S. M. Iftekhar, and B. S. Manjunath, "Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[36] M. Gualtieri and R. Platt, "Learning manipulation skills via hierarchical spatial attention," *IEEE Transactions on Robotics*, vol. 36, no. 4, p. 1067–1078.

[37] M. Gualtieri and R. Platt, "Learning 6-dof grasping and pick-place using attention focus," in *Conference on Robot Learning*, vol. 87, 2018, pp. 477–486.

[38] B. Wu, I. Akinola, and P. K. Allen, "Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes," in *IROS*, 2019.

[39] B. Wu, I. Akinola, A. Gupta, F. Xu, J. Varley, D. Watkins-Valls, and P. Allen, "Generative attention learning: a "general" framework for high-performance multi-fingered grasping in clutter," *Autonomous Robots*, vol. 44, 07 2020.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.

[41] J. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.

[42] C. Szepesvari, *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010.

[43] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.