Learning to Grasp in Clutter with Interactive Visual Failure Prediction

Michael Murray, Abhishek Gupta, and Maya Cakmak

***Abstract*— Modern warehouses process millions of unique objects which are often stored in densely packed containers. To automate tasks in this environment, a robot must be able to pick diverse objects from highly cluttered scenes. Real-world learning is a promising approach, but executing picks in the real world is time-consuming, can induce costly failures, and often requires extensive human intervention, which causes operational burden and limits the scope of data collection and deployments. In this work, we leverage interactive probes to visually evaluate grasps in clutter without fully executing picks, a capability we refer to as Interactive Visual Failure Prediction (IVFP). This enables autonomous verification of grasps during execution to avoid costly downstream failures as well as autonomous reward assignment, providing supervision to continuously shape and improve grasping behavior as the robot gathers experience in the real world, without constantly requiring human intervention. Through experiments on a Stretch RE1 robot, we study the effect that IVFP has on performance - both in terms of effective data throughput and success rate, and show that this approach leads to grasping policies that outperform policies trained with human supervision alone, while requiring significantly less human intervention. Code, datasets, and videos available at `https://robo-ivfp.github.io`**

I. Introduction

The ability to grasp diverse objects from cluttered environments is central to many robotic applications: from picking items off warehouse shelves to unloading groceries at home. Robots that can reliably grasp objects can automate tasks such as object picking, sorting, and packing. However, developing robust grasping behavior is not trivial, especially in unstructured environments with clutter and large amounts of object diversity. For example, modern warehouses process millions of unique objects from rigid to highly deformable with various shapes and sizes. These objects are often densely packed into highly cluttered containers. The diverse and complex dynamics of such environments make simulating or directly modeling the objects challenging.

Learning from real-world experience is a promising approach that circumvents the challenges of simulation, but typically requires extensive human supervision both in terms of providing labels and in terms of resetting up scenes for autonomous data collection. Additionally, executing picks in the real world is time-consuming, can induce costly failures or object damage, and often requires extensive human intervention. During training, this significantly increases the burden of data collection and limits the scale at which data can be collected. During execution, failures can be irreversible or require difficult recovery, which can disrupt operational efficiency and limit the viability of robot deployments.

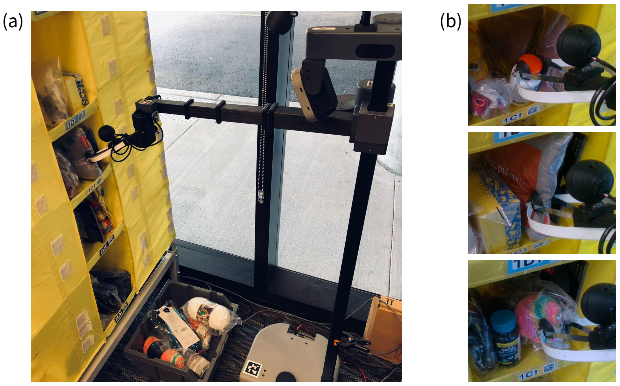

Fig. 1. (a) A Stretch RE1 robot picking objects from densely packed containers in an industrial warehouse setting. (b) The robot's gripper grasping various objects from cluttered bins. The robot must handle a diverse set of objects with various shapes, sizes, and physical properties. To successfully pick an object, the robot must grasp and extract the target object without dislodging the other objects in the container.

Ideally, failures would be detected early in the picking process, without requiring full execution to determine if a pick will succeed. This would enable us to avoid costly failed picks before they happen. Such capability could also be used to autonomously reward picks without disturbing the scene, providing supervision to continuously shape and improve picking behavior as the robot performs picks in the real world, while minimizing human intervention. We observe that picking can be divided into two sub-tasks: grasping and extraction. Grasping success is critical and highly informative of pick success, while irreversible failures typically happen during extraction. This presents an opportunity to avoid costly failures by predicting success before extraction.

However, it's often difficult to determine whether a grasp is successful and stable from passive visual observation alone due to partial observability, an issue that is compounded by the low visibility and high occurrence of occlusions in densely packed bins. Tactile feedback can help, but is insufficient due to being unable to detect certain modes of failure that are common in cluttered scenes, such as multi-picks. To address this challenge, we draw on ideas from *interactive perception*, a broad class of techniques in which the environment is manipulated to create rich sensory signals that would not be present through passive perception alone [1]. By using interaction to probe for information about the stability of a grasp, we can visually detect failures that are not perceivable by passive vision or tactile feedback. In

[1] The authors are with the Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA. `{mmurr, abhgupta, mcakmak}@cs.washington.edu`

This work was supported by an Amazon Science Fellowship.

doing so, we can both improve the training throughput since interventions can be minimized and the detected failures can be used to finetune grasp strategies, and also improve success rates since unstable grasps can be pre-empted and avoided.

While detecting suboptimal grasps using probes is useful for both training throughput and evaluation success rate, it can be challenging to actually perform this detection autonomously. On the other hand, humans possess a remarkable ability to visually judge grasp quality and refine their judgement through visual feedback [2] while only partially executing grasps. We are interested in exploiting this ability by directly leveraging human feedback for learning to perform and evaluate robotic grasping behavior, both in terms of the actual grasping behavior and in terms of preemptive evaluation of unsuccessful grasps. We propose a framework in which a human first demonstrates a potential grasp by teleoperating a robot, then observes the robot using probing motions to reveal information about the object configurations in the cluttered scene and test the stability of the grasp. We find that by observing the robot verify their grasp through interaction, humans are able to quickly and accurately classify grasp success.

This enables us to train an interactive visual grasp classifier capable of evaluating grasps in clutter without executing full picks, a capability we refer to as *Interactive Visual Failure Prediction (IVFP)*. Such a capability can be used to autonomously verify grasps during execution to avoid costly downstream failures, which directly improves success rates. IVFP can also be used to autonomously reward grasps as the robot performs picks in the real world, enabling real world learning to improve grasp success with minimal human intervention. Moreover, during evaluation at test time, expensive failures can be preempted by first performing interactive probing and IVFP, and avoiding risky and unsuccessful grasps. We evaluate our approach in a real-world robot deployment using a Stretch RE1 in an industrial warehouse setting. Our experiments show that IVFP can immediately improve picking success by performing introspective online verification. Moreover, we show that IVFP used as a reward function can help improve grasping policies to outperform policies learned through imitation alone. Finally, we show that data collection with IVFP requires significantly less human intervention than typical data collection pipelines wherein picks are fully executed. This suggests that interactive probing can provide significant gains both in terms of training throughput and in terms of overall system success rate in cluttered warehouse settings.

II. Related Work

A. Learning to Grasp

Recent advancements in machine learning and deep learning have paved the way for the development of data-driven grasp learning techniques. These approaches enable robots to learn grasping strategies based on raw sensory inputs, without any prior knowledge or explicit modeling of the target objects. For a survey on learning based robotic grasping we refer the reader to [3]. The vast majority of prior works focus on learning to select grasp configurations in advance, prior to making contact with the target object [4]–[11]. Most similar to our work, Calandra et al. [12] propose the use of passive visio-tactile feedback to assess grasps after contact. However, we find that passive feedback is not sufficient for predicting failures in highly cluttered scenes. In this work, we show that rich visual signals can be acquired through interaction, and that by using interaction to test our grasps, we can iteratively adjust and improve grasps from vision alone.

B. Interactive Perception

Interactive perception is an active area of research involving agents that perform physical interactions to obtain information about the latent state of a partially observed environment [1]. Similar to our work, many existing approaches use robot interaction to assess the state of a desired task [12]–[15], but prior works typically rely only on tactile or proprioceptive feedback. In contrast, we look to assess the state of our task from the visual feedback produced by interaction. Further, the vast majoriy of prior works use interactive behaviors only during task execution. Most similar to our work, Huang et al. [16] recently propose employing interactive perception behaviors as a reward function for training task policies. However, their approach is limited to controlled environments where ground truth success and failure examples can be synthesized (e.g. simulated environments or mechanically controlled environments). In this work, we propose using human supervision to train interactive behaviors that can serve as reward functions for real-world reinforcement learning in uncontrolled environments. Additionally, while their work uses interaction to retrospectively test for failure *after* task completion, we propose to use interaction *during* the picking task to predict and avoid *future* down-stream failures.

III. Problem Statement

Consider the picking task illustrated in Figure 1. The task is initiated when the robot arrives at a scene of diverse objects densely packed into a cluttered bin. The robot must grasp and extract a given target object without grasping other objects in the bin. The picking task is performed by a manipulation robot with Cartesian motion and a parallel-jaw gripper.

Each *grasp* is defined as a set of variables determining actions of the robot: a 3D point (x, y, z) indicating the grasp point and a pre-grasp gripper width w.

Let $\mathcal{G}$ be the set of all possible grasps, and $\mathcal{S}$ the set of scene states. At each timestep t, the current state $s_t \in \mathcal{S}$ is defined by the bin layout, the poses and states of all objects in the bin, and the pose and state of the robot. The robot does not have access to the state s_t, but only to an observation o_t. An observation $o_t = (I_t, M_t)$ includes an RGB-D camera image I_t and the target object mask M_t. Given the observation o_t, the robot's goal is to generate a grasp action $a_t \in \mathcal{G}$. Once a grasp is generated and executed, the robot performs a fixed extraction motion. The task is considered successful if the target object masked by M_t is extracted from the bin with all other objects remaining in

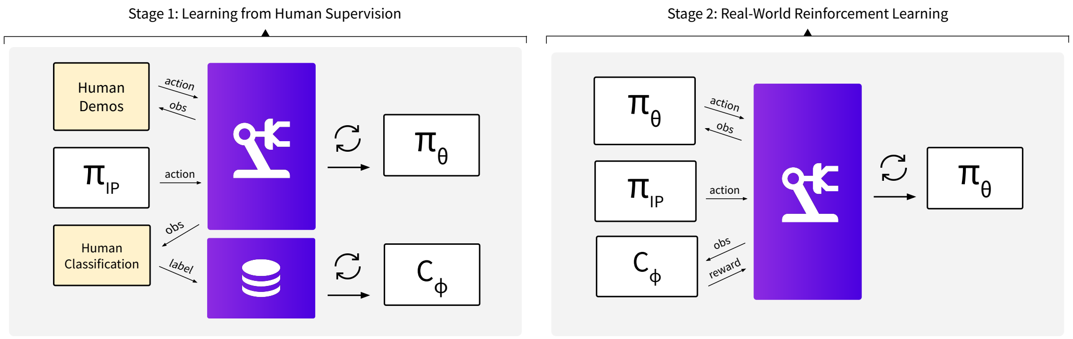

Fig. 2. A diagram of the proposed framework for learning to grasp with Interactive Visual Failure Prediction (IVFP). The framework consists of two stages of learning. Stage 1: A human demonstrates the target task and classifies their success on the task based on the observations produced by an interactive perception policy. The human demonstrations are used to train an initial task policy and the labels are used to train a task classifier. In Stage 2, the robot acts using the latest task policy and autonomously determines task reward using the learned classifier. The policy is updated offline periodically to maximize predicted reward.

the bin. Once the entire pick has been executed, whether successful or not, the process starts over on the next scene, which may be a slightly modified or entirely new scene.

IV. Method

We are interested in developing IVFP capability for two important applications. First, we want to verify potential grasps in order to avoid costly failures downstream. Second, because learning methods are limited by the cost of collecting human supervision, we are interested in using IVFP to autonomously reward grasps and improve them through experience. IVFP provides multiple advantages for these purposes. The interaction produces visual feedback that is highly informative of pick success, supporting accurate grasp classification, and the probe allows us to classify grasps without executing a full pick, enabling execution and training operations with minimal human intervention.

To capture these advantages, the probe design should prioritize (1) reversibility, so as to not disturb the scene, and (2) information gain, to enable accurate classification. We design our probes as a partial execution of the extraction step, where the item is lifted and pulled, but not removed from the bin. In this way, we can gain information about the grasp's impact on extraction before irreversible failures can occur. We also note that by designing the probe as a partial execution of the extraction step, we can simply continue with extraction in the case of success, further facilitating efficient data collection. Since it is challenging to heuristically determine grasp success from probes, we use human supervision to extract the rich information provided by the probe. We note that humans have the ability to both demonstrate potential grasps and perceive when a grasp will fail from interaction, and we utilize human operators for both types of supervision.

We illustrate our framework for learning with IVFP in Figure 2. Our approach consists of a learned grasping task policy π_θ, a learned grasp classifier $\mathcal{C}_\phi$, an interactive perception policy π_{IP}, and two stages of learning. In the first stage, a human *demonstrates* a grasp, observes the interactive perception policy π_{IP} probing their grasp, then subsequently *labels* their grasp based on the observations produced by the probe. The demonstrations are used to train an initial grasping task policy π_θ and a grasp classifier $\mathcal{C}_\phi$ is trained from the labels.

In the second stage, we use the components learned from human supervision as building blocks for learning from experience. Now, the robot autonomously generates potential grasps using the latest task policy π_θ. The learned classifier $\mathcal{C}_\phi$ is used both for avoiding failures and for autonomously determining task reward. Using the reward determined by $\mathcal{C}_\phi$, the policy π_θ is updated offline periodically to maximize predicted reward.

A. Modeling the Grasp Policy

At each timestep t, the input to the grasp policy is the current observation $o_t = (I_t, M_t)$ and the output is a grasp action $a_t = (w_t, x_t, y_t, z_t)$. The grasp policy is responsible for choosing grasps that are most likely to succeed based on the current observation. Note that the grasp policy performs the initial grasp, while interactive probing and grasp success classification are done with a separate partial execution strategy outlined in Section IV-A.

We separate the policy into two action-value modules (Q-functions) that correspond to grasp success: The gripper width module $\mathcal{Q}_{width}$ chooses a pre-grasp gripper width, and conditioned on the chosen gripper width, the grasp point module $\mathcal{Q}_{grasp}$ decides where to grasp. Both modules are implemented as neural networks and their architectures are illustrated in Figure 3a. Note that rather than directly outputting grasp positions and widths, we represent these with implicit functions as noted in prior work [8].

For both modules, the raw observation o_t is first embedded into a pre-trained feature representation space by a vision transformer backbone. This backbone serves as a function that takes the raw observations o_t as input and outputs patch embeddings F_t. The gripper width module $\mathcal{Q}_{width}$ first applies a global average pooling layer to the patch embeddings F_t followed by a linear classifier. This network models an action-value function $\mathcal{Q}_{width}(w_t|F_t)$ that correlates with grasp success which we sample from to obtain the pre-grasp

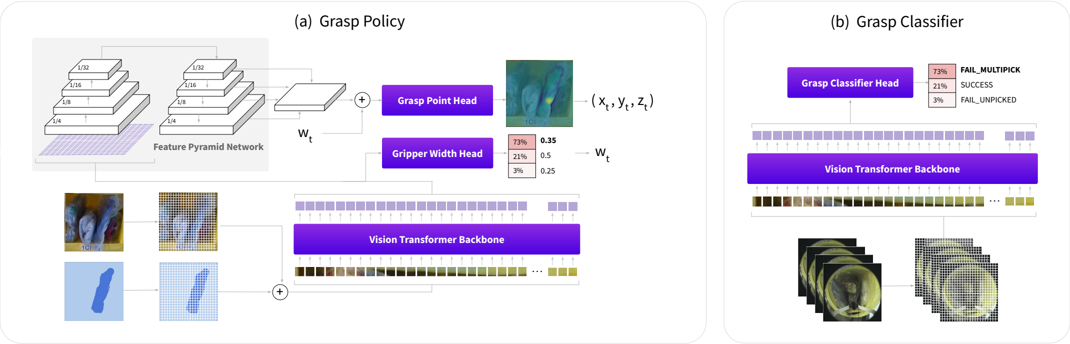

Fig. 3. (a) Architectures of the neural networks used to model the grasp policy π_θ. The policy takes as input the current RGB image I_t and target object mask M_t. The output includes a 3D grasp position (x_t, y_t, z_t) and pre-grasp gripper width w_t. (b) Architecture of the neural network used to model the grasp classifier $\mathcal{C}_\phi$. The input to the classifier is a video of the interactive perception policy π_{IP} testing a grasp. The output is a grasp class prediction c_t.

gripper width w_t:

$$w_t = \underset{w}{\operatorname{argmax}} \mathcal{Q}_{\text{width}} \left(w \mid F_t\right)$$

The grasp point module $\mathcal{Q}_{grasp}$ models a spatial action-value function [17]–[19] taking input $\gamma_t = (F_t, w_t)$ and outputting a dense pixelwise prediction $\mathcal{Q}_{grasp} \in \mathbb{R}^{H \times W}$ of action-values which are used to select a grasp point:

$$(u_t, v_t) = \underset{(u,v)}{\operatorname{argmax}} \mathcal{Q}_{\text{grasp}} \left((u, v) \mid \boldsymbol{\gamma}_t\right)$$

To execute the grasp, we map the selected point (u_t, v_t) from the camera image frame to a 3D grasp location (x_t, y_t, z_t) using the depth channel of the image and the known camera calibration. We base our network $\mathcal{Q}_{grasp}$ on the Upernet [20] architecture for its high efficiency on spatial tasks. The visual feature embeddings F_t are fed through a Feature Pyramid Network [21] and the outputs are fused. We project the pre-grasp gripper width w_t to match the dimensions of the fused feature map, concatenate them together, then finally apply a convolutional layer to produce a dense pixelwise prediction.

Both networks $\mathcal{Q}_{width}$ and $\mathcal{Q}_{grasp}$ are initially trained in a supervised maximum likelihood manner to predict grasp actions that imitate the human demonstrations. The networks are trained separately with $\mathcal{Q}_{width}$ using standard cross entropy loss and $\mathcal{Q}_{grasp}$ using a modified version of the cross entropy loss that incorporates a Gaussian penalty to encourage the model to make predictions that are close to the target point without requiring exact matches.

B. Interactive Perception and Grasp Success Classification

After performing a grasp according to π_θ, we want to predict if the grasp will succeed in order to avoid costly failures during execution and efficiently reward grasps during training. But it is difficult to determine grasp success from passive observation alone, so to better inform grasp classification, the robot verifies the grasp using the interactive perception policy π_{IP}. For this work, we used a fixed interactive perception policy that performs a cyclic lift-and-pull probing motion to test the grasp. The motion is designed to be a reversible partial execution so as to not perturb the scene, while being able to be executed directly if the grasp is predicted to be successful. This motion produces a set of visual observations I_t^{IP}.

Based on these observations, the grasp classifier $\mathcal{C}_\phi$ is responsible for determining whether a continuation of this particular grasp would be successful or not. The classifier is implemented as a neural network that takes I_t^{IP} as input and outputs a grasp class prediction $c_t \in \{\text{SUCCESS}, \text{FAIL}\}$. An illustration of the network architecture can be found in Figure 3b. The network begins with a vision transformer backbone which is pre-trained using a masked auto-encoding scheme [22] on *Something-Something v2* [23], a large-scale dataset with 220,847 videos of humans manipulating objects. The backbone is used to obtain patch embeddings F_t^{IP} followed by a global average pooling layer and finally a linear classifier. This network models the distribution $P(c_t|I_t^{IP})$ from which we sample c_t. The network $\mathcal{C}_\phi$ is trained in a supervised manner using standard cross entropy loss.

We combine the interactive perception policy π_{IP} and the learned classifier $\mathcal{C}_\phi$ to achieve IVFP capability. This capability allows us to both autonomously determine rewards for learning from experience and autonomously verify grasps during execution. In the following sections we describe each of these applications in detail.

C. Using IVFP for Autonomous Reinforcement Learning

By imitating human demonstrated grasps, we can bootstrap our initial grasping task policy π_θ. As the performance of this policy is limited by the cost of human supervision, we want to further improve the policy by learning from experience. For this purpose, the IVFP capability achieved through the combination of π_{IP} and $\mathcal{C}_\phi$ serves as an interactive reward function (IRF) [16]. After each grasp action a_t, the policy π_{IP} is executed to produce I_t^{IP} which is used by $\mathcal{C}_\phi$ to predict the grasp classification c_t. This classification is used to directly determine the reward $\mathcal{R}_t$:

$$\mathcal{R}_t = \begin{cases} 1, & \text{if } c_t = \text{SUCCESS} \\ -1, & \text{if } c_t = \text{FAIL} \end{cases}$$

After accumulating a dataset of action-reward pairs, we fine-tune the grasping task policy π_θ using an off-policy

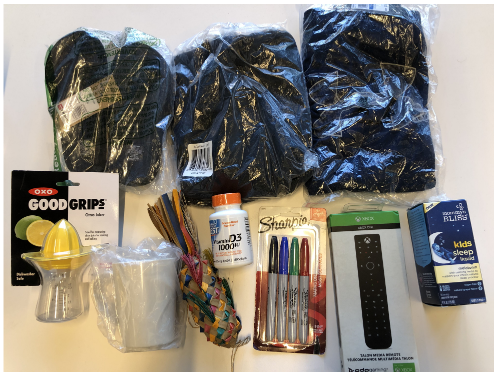

Fig. 4. A subset of the objects used for evaluation. Our item set includes 42 unique objects with a variety of object sizes, shapes, and physical properties. The objects can be rigid or highly deformable.

variant of the REINFORCE algorithm [24] in a contextual bandit setting. Specifically, we update the policy to maximize the expected reward using the policy loss:

$$\mathcal{L} = -\mathbb{E}[\mathcal{R}_t \cdot \nabla_\theta \log(\pi_\theta(a_t|s_t))]$$

D. Using IVFP for Verification in the Loop

In addition to autonomously determining grasp rewards, we want to utilize the IVFP capability to verify grasps and avoid costly failures. To this end, at test-time we sample multiple grasp parameters from our action-value networks $\mathcal{Q}_{width}$ and $\mathcal{Q}_{grasp}$ so that we can iteratively attempt alternative grasps in the case of failure. First we sample multiple gripper widths from $\mathcal{Q}_{width}$ and each candidate gripper width is input into $\mathcal{Q}_{grasp}$ along with the patch embeddings F_t. This results in a set of action-value maps, one for each candidate gripper width. We then sample multiple (w, u, v) combinations across all of the action-value maps weighted by predicted grasp success. We first execute the grasp parameters that are most likely to succeed, verify that grasp using π_{IP}, and evaluate the grasp using $\mathcal{C}_\phi$. When a failed grasp is detected, we move on to the grasp parameters that are the next most likely to succeed. This process repeats until we have either detected a successful grasp or exhausted our set of candidate grasp parameters.

V. Experimental Setup

A. Hardware

To evaluate our approach, we conduct a series of experiments on a Stretch RE1 robot [25]. The robot's mobile base, arm lift, and telescoping arm are moved in conjunction to reach a 3D target grasp point. An Intel RealSense D435i RGB-D camera is mounted to the frame and a 185 degree FOV fisheye camera is mounted to the wrist, providing observations for the grasp policy and the grasp classifier respectively. We deploy the robot in front of a picking work-cell, like those found in industrial fulfillment warehouses, with a shelving unit housing densely packed bins.

B. Item Set

Our item set consists of 42 unique objects with various shapes, sizes, and physical properties, including deformable and bagged objects. 32 of the objects are used during training and 10 are held out for unseen object evaluation. A subset of the objects can be seen in Figure 4.

C. Data Collection

Our dataset consists of 2,143 human teleoperated picks. To teleoperate the robot, participants use a custom web-based interface designed specifically for this task. First, a camera image of the target bin is displayed and the participant is prompted to select a grasp point by clicking on the image. We map the selected (u, v) position from the camera image to a 3D grasp location using the depth image and known camera calibration. Next, the robot moves its end effector to a pre-grasp pose relative to the selected grasp point and the user is prompted to select a pre-grasp gripper width using a slider. Finally, the robot executes the grasp followed by our fixed interactive perception motion policy, effectively testing the participant's chosen grasp parameters. After watching the images produced by the interactive perception motion, the participant chooses to classify the grasp as a success (in which case the robot executes a fixed extraction policy) or as a failure (in which case the robot resets and a new grasp point is chosen). In total we had 12 participants provide demonstrations including one of the researchers and 11 colleagues recruited from our department.

D. Training Details

In the demonstrations, successful picks are more common than failed picks, resulting in an imbalanced dataset. For classification we undersample successes to create a more balanced dataset consisting of 975 successes and 961 failures.

E. Baselines and Experiments

Centroid: A heuristic baseline always grasping from the center of the masked object.

Random: A random baseline sampling points uniformly from within the masked pixels.

Imitation Learning (IL): A learned baseline using the initial grasping policy produced by imitating the behavior of the human demonstrations.

Verification-in-the-Loop (IL+VitL): In this method, IVFP is used to verify grasps and retry failed grasps until success or no candidates grasps remain.

Reinforcement Learning (RL): In this method, the predictions from IVFP are used to fine-tune the grasp policy using reinforcement learning. We report results after training for 20 iterations and 50 iterations. In each iteration, we collect a batch of 64 grasps.

F. Evaluation Metrics

To quantify these approaches we report on the following two evaluation metrics:

Success Rate (SR) is the percentage of picks for which the target object was extracted successfully.

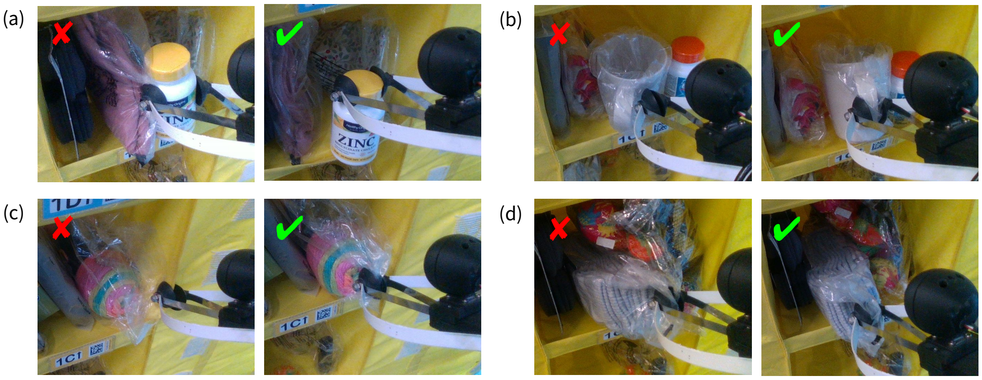

Fig. 5. Qualitative examples of IVFP utilized for Verification in-the-Loop (VitL). Failed grasps (left) are identified by IVFP and iterated upon to produce successful grasps (right). In (a) and (d) the initial grasp configuration resulted in a multi-pick failure. In (c) the initial grasp configuration resulted in collision with the bin and a subsequent missed-pick. In (b) the initial grasp configuration resulted in a missed-pick.

Method	**Seen objects**		**Unseen objects**	
	SR	**UPH**	**SR**	**UPH**
Centroid	44.83%	36.08	-	-
Random	29.61%	25.2	-	-
IL	56.76%	45.92	49.16%	40.18
IL+ViTL	67.33%	43.76	57.51%	37.05
RL @ 20	69.16%	56.58	61.66%	48.8
RL @ 50	73.33%	58.4	62.51%	49.6

TABLE I

EVALUATION RESULTS OF VARIOUS GRASPING METHODS ACROSS TWO METRICS: SUCCESS RATE (SR) AND UNITS PER HOUR (UPH).

Units Per Hour (UPH) indicates how many target objects could be picked per hour, quantifying the speed at which the robot is picking.

G. *Additional Experiments*

To study the effect that interaction has on performance, we perform an ablation study where we compare accuracy of a model with access to the observations produced from interaction against a model with access to only passive observations. To evaluate the data throughput benefits and tradeoffs of our approach, we compare a 1 hour data collection with IVFP to a 1 hour data collection using a more typical collection pipeline wherein the robot fully executes each pick. We report on three metrics: picks collected per hour, human interventions per hour, and collected label accuracy.

VI. RESULTS

All methods described in Section V were evaluated on both seen and unseen object sets. For each method, we evaluate over 10 trials each consisting of 12 picks.

In Table I, we can see that using IVFP for verification in the loop results in significant performance gains. Qualitative examples can be seen in Figure 5. This method can be applied immediately as it requires no additional training. However, it comes at the cost of operation speed as verifying every grasp results in a decrease in UPH. Results of RL from 10 to 50 iterations show that we can improve performance by using IVFP to learn from experience.

The results of our data throughput experiment are summarized in Table II, emphasizing that our approach can significantly reduce the burden of data collection with a minimal impact on collected label accuracy. In Table III, the results of our ablation study on the effect of interaction show that interaction is crucial for classification and illustrate the tradeoff between interaction time and classifier accuracy.

Method	**Picks/Hr**	**Interventions/Hr**	**Label Acc.**
Full picks	82	24	100%
IVFP	158	6	96%

TABLE II

COMPARISON OF DATA THROUGHPUT TRADEOFFS BETWEEN DATA COLLECTION WITH FULL PICKS AND WITH IVFP.

Perception	**Seen objects**			**Unseen objects**		
	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
Passive	0.5	0.56	46%	0.43	0.41	41%
Interactive (0.5s)	0.69	0.72	72%	0.61	0.62	64%
Interactive (1.0s)	0.79	0.81	81%	0.72	0.76	75%
Interactive (1.5s)	0.84	0.86	87%	0.81	0.83	83%
Interactive (2.0s)	0.94	0.93	94%	0.92	0.9	90%

TABLE III

PERFORMANCE OF THE LEARNED CLASSIFIER WHEN INTERACTION IS USED AS COMPARED TO WHEN PASSIVE PERCEPTION IS USED.

VII. CONCLUSION

In this work, we presented an approach to grasping in clutter using Interactive Visual Failure Prediction (IVFP). In our approach, the robot interacts with the environment by performing a cyclic interactive probe designed to inform grasp success. We combine the interactive behavior with a visual classifier learned from human feedback to achieve IVFP. We perform experiments in the context of a real-world robot deployment showing that this approach both improves grasping performance and reduces the burden of data collection. While effective in our domain, our approach utilizes a fixed interaction policy which won't necessarily generalize to other domains. To address this limitation, exploring methods of learning interaction policies as in [16] is an exciting direction. Additionally, our task is performed in a relatively constrained contextual bandits setting and future work should explore how to apply IVFP on longer horizon problems with richer action spaces.

References

[1] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, “Interactive perception: Leveraging action in perception and perception in action,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[2] G. Maiello, M. Schepko, L. K. Klein, V. C. Paulun, and R. W. Fleming, “Humans can visually judge grasp quality and refine their judgments through visual and haptic feedback,” *Frontiers in Neuroscience*, vol. 14, p. 591898, 2021.

[3] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, “A survey on learning-based robotic grasping,” *Current Robotics Reports*, vol. 1, pp. 239–249, 2020.

[4] I. Kamon, T. Flash, and S. Edelman, “Learning to grasp using visual information,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1996, pp. 2470–2476.

[5] 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, 2015.

[6] D. Kappler, J. Bohg, and S. Schaal, “Leveraging big data for grasp planning,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4304–4311.

[7] E. Johns, S. Leutenegger, and A. J. Davison, “Deep learning a grasp function for grasping under gripper pose uncertainty,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4461–4468.

[8] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” 2017.

[9] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, “Volumetric grasping network: Real-time 6 dof grasp detection in clutter,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1602–1611.

[10] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, “Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[11] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, “Dex-nerf: Using a neural radiance field to grasp transparent objects,” *arXiv preprint arXiv:2110.14217*, 2021.

[12] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[13] A. Rodriguez, D. Bourne, M. Mason, G. F. Rossano, and J. Wang, “Failure detection in assembly: Force signature analysis,” in *2010 IEEE International Conference on Automation Science and Engineering*. IEEE, 2010, pp. 210–215.

[14] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, “Skill learning and task outcome prediction for manipulation,” in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3828–3834.

[15] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, “Learning to switch between sensorimotor primitives using multimodal haptic signals,” in *From Animals to Animats 14: 14th International Conference on Simulation of Adaptive Behavior, SAB 2016, Aberystwyth, UK, August 23-26, 2016, Proceedings 14*. Springer, 2016, pp. 170–182.

[16] K. Huang, E. S. Hu, and D. Jayaraman, “Training robots to evaluate robots: Example-based interactive reward functions for policy learning,” in *Conference on Robot Learning*. PMLR, 2023, pp. 11–21.

[17] J. Wu, X. Sun, A. Zeng, S. Song, J. Lee, S. Rusinkiewicz, and T. Funkhouser, “Spatial action maps for mobile manipulation,” *arXiv preprint arXiv:2004.09141*, 2020.

[18] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, “Transporter networks: Rearranging the visual world for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[19] M. Shridhar, L. Manuelli, and D. Fox, “Cliport: What and where pathways for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[20] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified perceptual parsing for scene understanding,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 418–434.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[22] Z. Tong, Y. Song, J. Wang, and L. Wang, “Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” *Advances in neural information processing systems*, vol. 35, pp. 10 078–10 093, 2022.

[23] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, “The” something something” video database for learning and evaluating visual common sense,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.

[24] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” *Advances in neural information processing systems*, vol. 12, 1999.

[25] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, “The design of stretch: A compact, lightweight mobile manipulator for indoor human environments,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3150–3157.