2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Amortized Inference for Efficient Grasp Model Adaptation

Michael Noseworthy[1*], Seiji Shaw[1*], Chad C. Kessens[2], Nicholas Roy[1]

***Abstract*— In robotic applications such as bin-picking or block-stacking, learned predictive models have been developed for manipulation of objects with varying but known dynamic properties (e.g., mass distributions and friction coefficients). When a robot encounters a new object, these properties are often difficult to observe and must be inferred through interaction, which can be expensive in both inference time and number of interactions. We propose an encoder/decoder action-feasibility model to efficiently adapt to new objects by estimating their unobserved properties through interaction. The encoder predicts a distribution over the unobserved parameters while the decoder predicts action feasibility, which can be used in an uncertainty-aware planner. An explicit representation of uncertainty in the encoder enables information-gathering heuristics to minimize adaptation interactions. The amortized distributions are efficient to compute and perform comparably to particle-based distributions in a grasping domain. Finally, we deploy our method on a Panda robot to grasp heavy objects.**

I. Introduction

In automated manipulation, tasks such as bin-picking or block-stacking are often repeated for a large set of objects that have a wide distribution of *geometric* and *dynamics* parameters (e.g., masses, centers of mass, and friction coefficients). Previous work has shown that robots can learn manipulation dynamics (e.g., stacking or throwing) entirely from online data when the dynamics properties of the objects are known [1] or have minimal impact on action outcome. It is unreasonable, however, to expect that a robot should be able to observe the dynamics properties of a novel object. Robots without tactile or force feedback, for example, will not be able to directly observe inertial and frictional properties of objects. A robot may not even know which properties will play a role in the manipulation dynamics.

Our goal is for robots to discover new objects' unobserved dynamics parameters through efficient interaction for reliable prediction and planning. A standard approach to handling unknown parameters is to use a Bayes filter [2] to maintain a probability distribution or belief over the unobserved properties. Every time a new action is executed, the resulting observation (e.g., grasp success or failure) is used to update the belief using an observation model that relates the unknown parameters to observations.

When the observation is a simple function of the belief, such as a linear function applied to a Gaussian distribution, the belief update can be efficient. However, when the observation model is a complicated function, more expressive belief representations, such as sets of particles, are required. Particle-based approaches suffer from the curse of dimensionality: the number of particles needed to represent the belief scales exponentially with the number of unobserved variables. For tasks such as grasping, there are five unknowns: mass, static friction, and three for center of mass. Sampling and updating the belief can therefore be expensive.

Fig. 1: Our method allows a Panda (right) to infer *unobserved properties* of objects based on interaction history. This allows the robot to efficiently grasp objects with non-uniform mass distributions such as weighted blocks (left) or ShapeNet objects (center).

When manipulating objects with unknown properties, a robot will need to perform many belief updates as it interacts and collects observations. Further, while planning, it is common to simulate belief updates to choose informative actions. As such, the cost of the belief update is critical. Particle-based representations quickly become intractable and limit the amount of interaction possible within a fixed time budget.

To develop a dynamics model with an efficient belief representation and update procedure, we propose to *jointly learn* (1) inference networks that are trained to predict posterior distributions over the unobserved parameters given a history of interactions and their outcomes, and (2) an action feasibility classifier that can be used for planning. The model uses an encoder/decoder architecture based on *Neural Processes* [3] for unsupervised learning of an object-level latent space shared between the inference network (encoder) and feasibility classifier (decoder). For our grasping task, the input to the inference network is a set of labeled grasps for a specific object. The posterior over the latent parameters is computed as a *single forward pass* through the inference network/encoder, which only scales linearly with the number of observed grasps irrespective of the number of unobservable parameters. The feasibility classifier/decoder can in turn be used for planning by consuming samples from the posterior and predicting the success probability for candidate grasps.

We show that our model, called a Grasping Neural Process (GNP), can classify stable grasps comparably to particle-

*Equal Contribution, {mnosew,seijis}@mit.edu
[1]CSAIL, MIT [2]DEVCOM Army Research Laboratory (ARL).
This work was supported by ARL and the NSF Graduate Research Fellowship Program under Grants W911NF-23-2-0012 & 2141064. The views & conclusions contained in this document are those of the authors & should not be interpreted as representing the official policies, either expressed or implied, of DEVCOM Army Research Laboratory, the National Science Foundation, or the U.S. Government.

based methods at a fraction of the computational cost. In addition, the distributions produced by the inference network can effectively be used in information-gathering heuristics to adapt with fewer grasps. In simulation, we deploy the GNP in an uncertainty-based planner to robustly grasp objects with the minimal required force. We also demonstrate the GNP, trained in simulation, can be deployed to successfully adapt and grasp weighted blocks on a physical robot (see Fig. 1).

II. Problem Formulation

Given a previously unseen object with unknown dynamics properties, our goal is for a robot to reliably manipulate (e.g., grasp or place) it using a small number of adaptation interactions. To do so, we take a model-based approach where we aim to predict action feasibility in a way that allows the model to rapidly adapt to novel objects.

Formally, we consider a robot interacting with K objects, where each object's state is divided into observed and unobserved properties. The observed state of object k at time t, $x_t^{(k)}$, can be time-varying, whereas the unobserved state, $z^{(k)}$, is assumed to be static. Many time-varying properties (e.g., pose or velocity) can be extracted from a perception system, while static properties often cannot be directly observed (e.g., friction, mass, or coefficients of restitution).

Our objective is to learn a model that predicts the feasibility of an action for a specific object[1] $y_t \in \{0, 1\}$ (e.g., grasp stability). This model can then be integrated with a *task and motion planning* system for long-horizon manipulation tasks. Concretely, the goal is to estimate $p(y_t|x_t, z, a_t)$, where $a_t \in \mathcal{A}$ is the action space (e.g., grasps or pushes).

We assume the robot's operation is divided into three distinct phases: *training*, *adaptation*, and *testing*. In the *training* phase, the set of objects is fixed, and the robot has some finite amount of time to act in the environment without any task descriptions. In this work, we assume we already have a dataset of completed interactions provided to us for the training phase. In the *adaptation* phase, the robot is given a new object and a short amount of time to experiment and adapt by selecting information-gathering interactions. Finally, there is a *testing* phase, in which a robot is given a task to be performed, and interacts with the objects to complete the task. No further adaptation or learning is performed in the *testing* phase.

III. Methods

We primarily focus on the *adaptation phase*. How can a robot *efficiently*, in both computational and sample efficiency, adapt to objects with unknown dynamics properties?

We take a probabilistic approach where we perform posterior estimation of the unknown properties, z, given an increasing amount of observed data, $D_t = \{(x_1, a_1, y_1), \ldots, (x_t, a_t, y_t)\}$. Rather than estimate the posterior distribution through an expensive online computation, we propose to amortize the inference procedure by first training an inference network, q_ϕ, to directly predict the parameters of the posterior distribution (Section III-A). This model is learned jointly with the feasibility model, $p_\theta(y|x, z, a)$. The resulting posterior distribution, $q_\phi(z|D_t)$ (which we will refer to as an amortized distribution), can be used directly for informative action planning (Section III-B) or in conjunction with the feasibility model for uncertainty aware planning (Section III-C). See Fig. 3 for an overview.

[1]We will omit the superscript k when referring to a single object.

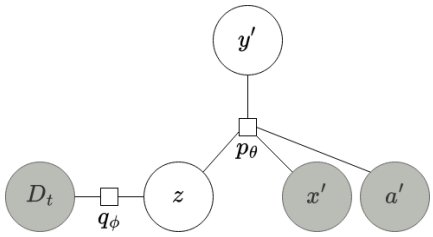

Fig. 2: After executing t actions and observing their outcomes, D_t, the robot can predict a posterior over the unobserved properties, $q_\phi(z|D_t)$. $p_\theta(y'|x', z, a')$ uses this posterior along with unlabeled actions, a', and observed properties, x', to predict feasibility, y'.

A. Amortized Posterior Updates

When interacting with a novel object, the robot must update its posterior after each interaction, which can be a time-intensive operation. To address this issue, we propose to use amortized inference for constant-cost posterior distributions. Instead of performing an expensive online posterior update, we will train a neural network to predict the posterior distribution given the online samples, D_t, observed so far. The inference network, $q_\phi(z|D_t)$, will need to approximate true posterior distributions $p(z|D_t)$ for all possible timesteps, t, and observation realizations. We take $q_\phi(z|D_t)$ to be a multi-dimensional Gaussian distribution.

At any timestep, t, during the adaptation phase, the graphical model in Fig. 2 represents the data generation process. Note that this graphical model represents a single timestep of the adaptation phase where we have already observed the labels for t actions, D_t, but have not yet observed the label, y', for a new action, a', with observed state, x'. In order to learn both a feasibility model, $p_\theta(y'|z, x', a')$ that can be used for planning and an inference network, $q_\phi(z|D_t)$, we take an amortized variational inference approach.

In training, for a single object, we seek to minimize:

$$\mathbb{E}_z \left[\log p_\theta(y'|x', a', z)\right] - D_{KL}\left(q_\phi(z|D_T)||q_\phi(z|D_t)\right), \quad (1)$$

where $D_T = D_t \cup \{(y', a', x')\}$. Note that we do not require labels for the unobserved properties as the latent space is learned (and need not be human-interpretable).

The proposed method is an instantiation of *Neural Processes* [3] for robotic manipulation. The training objective is derived from the evidence lower bound (ELBO) for amortized variational inference over functions (i.e., each sample of the latent space represents a different action-feasibility classifier). The first term of this objective incentivizes the model to make accurate and object-consistent predictions, using the latent space when necessary. The second term is the KL-divergence between the *full posterior* $q_\phi(z|D_T)$ and

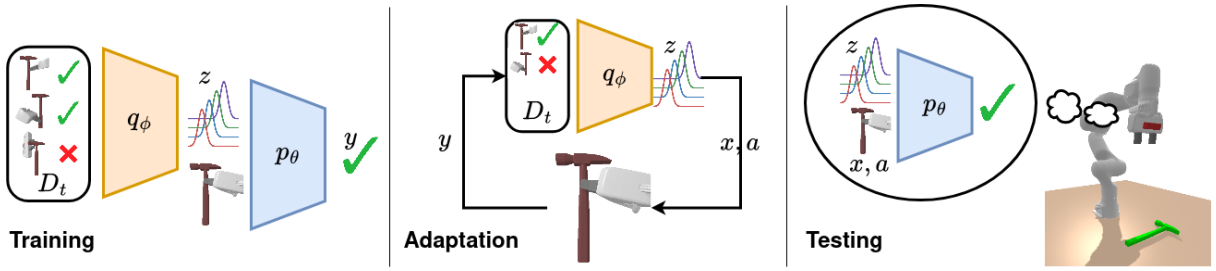

Fig. 3: Grasp Neural Processes (GNPs) use an offline *training phase* (left) to jointly learn an action feasibility model, p_θ, and an inference network, q_ϕ, that predicts a posterior distribution over unobserved properties. During the *adaptation phase* (center), the learned inference network can be used for efficient online posterior updates and action selection. Finally, in the *testing phase* (right), the robot can use the current belief, z, along with the feasibility model to perform manipulation tasks.

the *partial posterior* $q_\phi(z|D_t)$. The partial posterior includes only a subset of the full data and the KL-term encourages it to be as accurate as possible with limited data. This ensures the partial posteriors will be useful even when only a few labeled interactions are available during the adaptation phase.

During training, Eq. 1 is optimized using stochastic gradient descent. Mini-batches are sampled where each element of a batch corresponds to $D_T^{(k)}$ for many objects, k. Each $D_T^{(k)}$ is further divided into partial datasets $D_t^{(k)}$ by uniformly sampling t between 0 and T ($\mathcal{N}(0,1)$ is used for $q_\phi(z|D_t)$ when $t=0$). This ensures the encoder can represent posterior distributions for variable input sizes.

B. Informative Action Selection

We have shown so far how to perform efficient posterior updates at a single timestep: evaluate $q_\phi(z|D_t)$. We are also interested in minimizing the total number of time-consuming interactions used to collect observations. While selecting a *random* action may be quick, many interactions are likely required before reaching a sufficient performance level.

Instead, it is common to take an information-theoretic approach. Actions are selected which maximize the information gain at a single timestep (a greedy, but often good, approximation to optimizing over sequences of actions [4]):

$$\max_{a_{t+1}} H(q_\phi(z|D_t)) - \mathbb{E}_{y_{t+1}}\left[H(q_\phi(z|D_t, x_{t+1}, a_{t+1}, y_{t+1}))\right] \tag{2}$$

One way to solve this maximization is via a simple pool-based approach: first generate a collection of M unlabeled samples, then score each according to this objective. However, computing expected information gain requires two posterior updates, which can be expensive with particle-based methods: $O(MN)$ if N is the number of particles.

Using the inference network within the information gain computation has a much more favorable complexity, scaling linearly with the number of unlabeled samples: $O(M)$. Given a similar computational budget to baselines, more unlabeled samples can be considered, leading to more informative actions and therefore fewer expensive interactions.

C. Uncertainty Aware Planning

After seeing a limited number of adaptation actions, captured by D_t, there will always be some level of remaining uncertainty about the unknown properties. This uncertainty, or model confidence, can be incorporated into task planning to ensure robust behavior.

We consider two task formulations. The first is to simply find an action that is feasible. We use the learned networks, q_ϕ and p_θ, to find the most feasible action:

$$a^* = \text{argmax}_{a\in\mathcal{A}}\, p(y|x,a) \tag{3}$$

We also consider bandit-style tasks specified by a reward function, $R(a)$, where we must trade off maximizing the reward with current uncertainty:

$$a^* = \text{argmax}_{a\in\mathcal{A}}\, \mathbb{E}_{y\sim p(y|x,a)}[R(a)] \tag{4}$$

The predictive posterior used in both task formulations is computed using the learned distributions: $p(y|x,a) = \int p_\theta(y|x,z,a) q_\phi(z|D_t) dz$. This integral is approximated using Monte-Carlo sampling.

IV. Grasping Task

We focus on grasping objects with non-visible dynamics parameters. Grasping is important for many manipulation tasks and success depends on understanding properties such as mass and friction. For example, to grasp a table leg for furniture assembly, the robot must grasp as close to the center of mass as possible to avoid slip. Concretely, we aim to predict grasp stability (i.e., force closure) of candidate grasps given an object's observable properties, x_t, which consist of object geometry and grasp pose. The amortized posterior will need to infer center of mass (CoM), friction, and mass from a sequence of interactions to form an accurate prediction. We refer to our model as a *Grasp Neural Process* (GNP).

A. Dataset Generation

All experiments are run on two synthetic datasets in PyBullet: box primitives (*Boxes*), and a post-processed set of objects derived from 191 ShapeNet classes (see Fig. 1, center) [5].[2] For the training phase, we use 1000 unique object geometries with 5 dynamic property samples each. This leads to datasets with 5000 objects and 50 labeled grasps per object.

Object Generation: The CoM is uniformly sampled within the convex hull of the object, and lateral friction and mass are chosen from a uniform distribution over $[0.1, 0.3]$.

[2]The ShapeNet objects were post-processed to be watertight for simulation and rescaled so they could fit in the robot gripper as in [6].

Grasp Generation: We assume access to object meshes and a floating Panda gripper to check grasp stability. The mesh is first used to sample grasp points that are within a specified antipodal tolerance (30 degrees). We then sample a random gripper orientation around the line connecting the grasp points and ensure there are no collisions with the gripper. Finally a grasp force is uniformly sampled in the range $[5, 20]N$. The stability label is generated by closing the gripper in simulation and applying perturbation forces in random directions (a force closure approximation).

B. Network Architectures

Grasp Neural Processes (GNPs) use an encoder/decoder architecture (representing the inference net and learned feasibility model respectively) with domain-specific structure for grasping data. The encoder accepts an arbitrary number of grasps representing the history of grasps tried so far.

Input Features: Each object is represented by a *global point cloud* of 256 points sampled uniformly from the surface of the mesh. Each grasp is further represented by a local point cloud. The *local point cloud* is in the reference frame of the gripper and only includes points within $3cm$ of either grasp point. The *global point cloud* allows the network to reason about object-level properties like *volume* and *moments of inertia* while the *local point cloud* allows reasoning about local features like *curvature* and *surface normals*. Each grasp is further represented by the grasp points as well as the grasping force. The *encoder* will have access to the grasp's label while the decoder will not.

Auxiliary Networks: We use PointNets [7] to encode both local and global point clouds. A separate PointNet instance is used for each point cloud type and maps the set of points into a single fixed-length embedding.

Encoder: The GNP encoder takes as input a collection of grasps. It outputs the parameters of a d-dimensional diagonal Gaussian distribution, $\mu_z, \sigma_z \in \mathbb{R}^d$. We repurpose the PointNet architecture to operate over a set of grasps.

Decoder: Finally, the GNP decoder takes as input a set of unlabeled grasps, and a latent sample, $z \sim \mathcal{N}(\mu_z, \sigma_z^2)$, to output a grasp stability probability using an MLP.

V. Experiments

In our experiments, we evaluate whether GNPs yield well-performing grasp-detectors with faithful uncertainty representations of the unknown dynamic parameters in the *ShapeNet* and *Boxes* datasets. We first compare the proposed method to an accurate (but expensive) particle-based baseline, achieving comparable performance at a fraction of the cost (Section V-A). Then, we show how the learned posterior can be leveraged for faster information-gathering (Section V-B). Finally, we evaluate the model as part of an uncertainty-aware planner for a grasping task (Sections V-C and V-D), and deploy the model on a real Franka Emika Panda robot.

A. Particle Filter Baseline

To understand the trade-off between efficiency and prediction performance of GNPs, we compare our approach

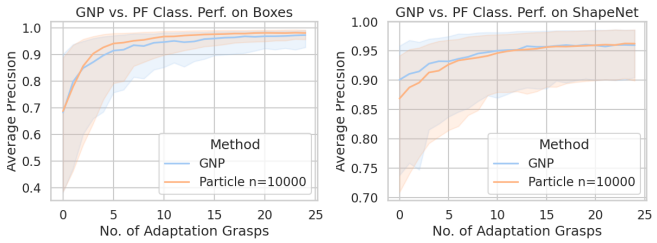

Fig. 4: **GNP vs. PF** The proposed method achieves comparable adaptation performance to an expensive particle filter baseline with many particles. Lines show the median *Average Precision* of the grasp classifier across 500 novel objects. Shaded regions are between the top and bottom quartiles.

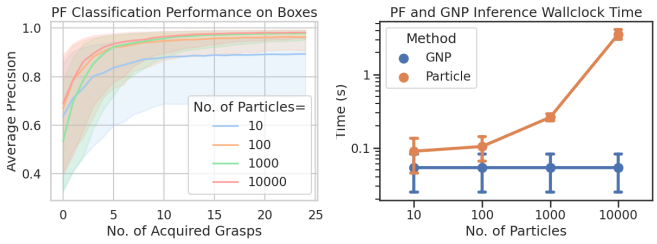

Fig. 5: **Particle Filter Analysis (Boxes)** The particle filter performance is dependent on the number of particles (left) yet the inference time scales poorly with the number of particles (right), compared to the proposed amortized inference method. Timing results represent the mean/std across 2500 inference runs.

to a particle filter. We reuse our decoder architecture and train it to predict grasp stability using the known dynamics parameters with a standard cross-entropy loss. This model is then used during the *fitting* phase as the observation model of a standard particle filter found in Chopin et al. [8]. The filter is initialized with a varying number of particles ($N = 10, 100, 1000, 10000$) sampled from a uniform distribution over the range of valid dynamics parameters.

Fig. 4 shows the performance of our GNP method compared to the particle filter using a large number of particles ($N = 10,000$). We report *average precision*[3] of the grasp stability classifier evaluated over time as the model collects more adaptation grasps. We find that GNPs perform comparably to the particle filter at a fraction of the online inference cost used by the baseline as shown in Fig. 5.

B. Efficient Information Gathering

We evaluate if the amortized posteriors can be used to gather data more efficiently using an information-gain (IG) acquisition function (Section III-B). A useful uncertainty representation should permit the model to reason about what inputs are more informative than others.

Given a trained GNP, we simulate the adaptation phase under the random and information-gain (IG) strategies. At each timestep, we sample 20 unlabeled grasps from the novel object. The *random* strategy chooses a single grasp from this pool. The *IG* strategy computes the expected information-gain metric for each grasp, and greedily chooses the grasp

[3]Average precision focuses on positive grasps (balancing precision and recall), which shows how the classifier would perform in downstream tasks.

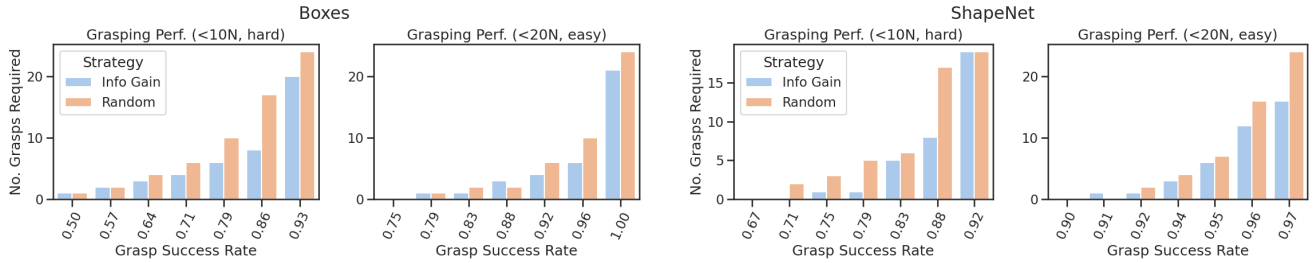

Fig. 6: **GNP Performance as a Grasp Detector** Number of adaptation grasps needed to reach a desired *Grasp Success Rate*. The GNP chooses the most likely grasp considering the current uncertainty arising from limited data. We consider *easy* and *hard* versions of the task that use different force limits. The IG adaptation strategy tends to produce higher success rates with fewer adaptation grasps.

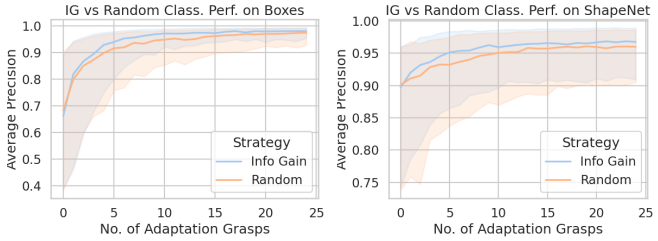

Fig. 7: **Information Gain** GNP encoders can be exploited for faster information-gathering. *Average Precision* per grasp for IG and random adaptation strategies. Solid lines are medians, and error ranges are the upper and lower quartiles across 500 test objects.

with maximal IG. To evaluate performance, we again report the average precision of the grasp classifier (Fig. 7).

On the Boxes dataset, our results show that a greedy IG heuristic produces better performance with less acquisition data. This suggests that amortized posteriors are still useful for probabilistic reasoning. On the ShapeNet dataset, the IG heuristic has a smaller performance improvement over random. We attribute this to the significant influence of object geometry on grasp stability. Some objects have irregular geometries that are difficult to learn. For other objects, grasp stability is governed primarily by their geometry (e.g. an hourglass that is only graspable at the center), rather than their underlying dynamics parameters. Since all boxes have a similar geometry, the stability of a grasp is far more sensitive to the dynamics parameters than those of ShapeNet.

C. Finding Stable Grasps for Novel Objects

To evaluate the utility of GNPs for practical grasping tasks, we integrate the model into the grasp planner discussed in Section III-C. In simulation, we first generate a novel object and perform the adaptation phase using both IG and random strategies. After each observation, we use a sampling-based approximation of Eq. 3 to select the most likely grasp.

In Fig. 6, we report how many adaptation grasps are required to reach a desired *grasp success rate* (how often the chosen grasp is actually successful) on ShapeNet and Boxes datasets (with 500 novel objects per dataset). We consider two task difficulty levels: where the chosen grasp must be either $< 20N$ (easy) or $< 10N$ (hard). Using a smaller grasp force requires better understanding of dynamics properties.

Across both datasets and adaptation strategies, GNPs are able to successfully adapt to novel objects: performance increases with the number of adaptation grasps (requiring as little as 5 grasps to achieve high success rates). The IG strategy shows small improvements for easy grasps ($< 20N$ force; smaller bars are better). For hard grasps ($< 10N$ force), IG shows a larger gain, sometimes needing as few as half the required adaptation grasps for similar performance to random (e.g., to achieve 85% grasp success rate).

D. Uncertainty-Aware Planning

Another task that requires understanding dynamics properties is that of grasping an object with minimal force. Succeeding at this task requires grasping as close to the center of mass as possible to avoid slip. We formalize this task using the reward function:

$$R(G) = \begin{cases} F_{max} - F_G, & \text{grasp is stable,} \\ R_{failure}, & \text{otherwise.} \end{cases}$$

Grasp forces are in the range $[5, 20]N$ and smaller forces lead to larger reward. F_G is the force of the chosen grasp, and we choose a large negative reward ($R_{failure} = -F_{max}$) to prefer robust behaviour where the robot uses larger forces in the face of uncertainty (as opposed to dropping the object).

We perform this task using an uncertainty aware planner (Eq. 4), with sampling-based optimization (using 200 samples). We evaluate task performance with 500 objects after each adaptation grasp. In Fig. 8, we report how often the chosen grasp was successful and the normalized regret achieved for successful grasps.

We compare our approach that uses the uncertainty from the amortized posterior distributions (Monte Carlo) to an ablation that only uses the mean of the posterior distribution when planning (Most Likely). This ablation evaluates the utility of learned posteriors for robust grasp planning under uncertainty. Monte Carlo achieves higher success rates, showing robust behaviour even with few adaptation grasps. This comes at a cost of higher initial regrets, but task performance improves as the model collects more information, which is the desired robust behaviour. We also evaluate the Monte Carlo planner with the *particle filter* baseline (Fig. 9). There is a larger performance gap between particles and GNPs than for the avg. precision evaluation (Fig. 5). Achieving low regret requires grasping close to the classifier's decision boundary where uncertainty is likely to be higher. Thus it is important to have a good uncertainty representation (which requires many particles).

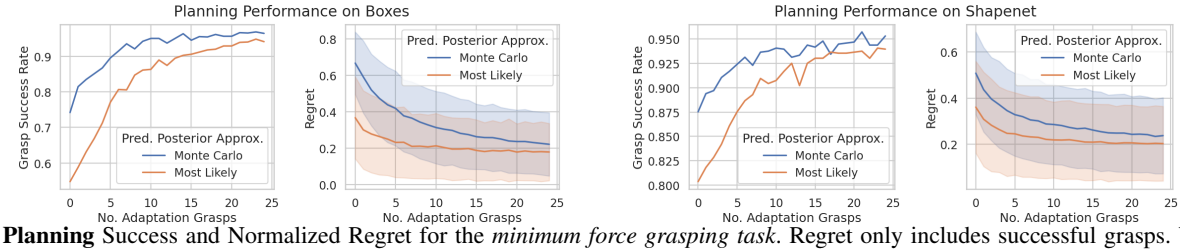

Fig. 8: **Planning** Success and Normalized Regret for the *minimum force grasping task*. Regret only includes successful grasps. We ablate our GNP model to show the impact of the explicit uncertainty representation. Lines are means and shaded regions are a standard deviation.

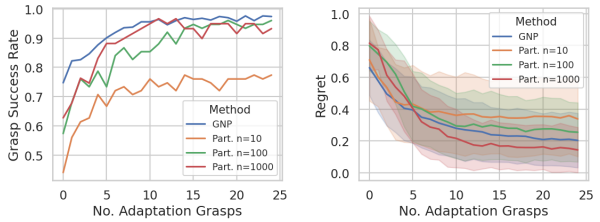

Fig. 9: **Planning** Success and Normalized Regret for the *minimum force grasping task* for the *particle filter* baseline with varying numbers of particles (for the boxes dataset).

	Block 1	Block 2
Mass (g)	236	196
Dims (cm)	6.0/14.0/5.5	5.5/9.0/5.5
COM (cm)	-0.2/3.4/-0.1	-0.2/1.9/-0.15
Succ. (@0)	8 / 20	15 / 20
Succ. (@10)	18 / 20	19 / 20

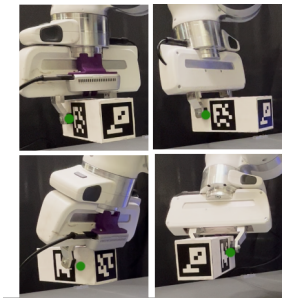

Fig. 10: **Robot Demo** [Left] Grasp success rate for two novel blocks. The baseline method (@0) only uses visual information while the GNP method (@10) uses 10 adaptation interactions. COM is relative to the object center. [Right] After 10 adaptation grasps, the robot consistently grasps the object near the CoM (green dot).

E. Panda Grasping Demo

We deploy the trained model on a Franka Emika Panda robot. The *training* phase is performed solely in simulation using the *Boxes* dataset, but we perform the *adaptation* and *testing* phases on the real robot (Fig. 10) using two heavy blocks with different masses and offsets of center of mass (i.e., only grasps on one side of the object will succeed). In the real world, we only sample grasps of $5N$ that are kinematically feasible, and grasps are labeled based on whether they slip in the gripper after lifting the arm.

We compare to an antipodal grasp sampler that chooses grasps based solely on object geometry. For our method, we allow the robot 10 adaptation grasps chosen using the IG strategy. For both blocks, the GNP model successfully leverages the adaptation phase to achieve better performance than a baseline that only uses visual information. Please see Fig. 10 and the accompanying video for more robot results.

VI. Related Work

Several previous works have introduced Bayesian models for rapid adaptation to novel environment dynamics. For example, [9] and [10] both use latent variables to parameterize a task and Gaussian Processes as the global dynamics. [11] has a similarly structured model, but uses a BNN as the dynamics model. [12] and [13] use both deep ensembles and low-dimensional latent variables within their models. For grasping, [14] developed adaptive grasp classifiers using a similar approach to ours. We extend their approach by considering probabilistic latent spaces that can be used for robust planning and efficient adaptation.

Other work has focused on learning object dynamics with unobservable object properties. In some works, the authors assume the global dynamics are known *a priori*, which can inform the values of object-specific properties [15], [16], [17], [18]. In our work, we desire to jointly infer the dynamics and object properties. Related work which does not assume the dynamics are known often rely on either a fixed dataset at adaption time [19], [20], [21], [22] or a task definition in order to infer the latent properties [23], [9], [24], [10]. Instead, we use information-gain heuristics with respect to model uncertainty, which is applicable even when the task is unknown. Furthermore, our approach can yield zero-shot performance on a new task, because all the experimentation was performed in a task-agnostic adaptation phase.

Disentangling object properties through interaction, termed 'interactive and active perception' [25], is a common strategy. Applications include understanding object kinematics [26], [27] or object geometry [28], [29]. Bandit-style approaches to grasp selection require a large number of grasps to be effective [30], [31]. Grasping-specific solutions have also been found [32], [33]. Unlike our approach, very few methods except for [27], [29] address model uncertainty over dynamic and geometric information, and only [29] leverages uncertainty for efficient information-gathering.

VII. Conclusions

We present Grasp Neural Processes: an action feasibility model with an explicit representation of uncertainty over a novel object's dynamics parameters. We experimentally verify that the amortized posterior distributions are more efficient than particle filters while achieving comparable performance. When integrated with downstream grasp planners, the learned uncertainty representation leads to increased robustness and enables info-gain heuristics. In the future, we plan to extend our models to partially-observed geometry and multiple actions that share the same latent space.

Acknowledgements

We thank Rachel Holladay, Caris Moses, Isaiah Brand, and Sebastian Castro for feedback and implementation support.

REFERENCES

[1] M. Noseworthy, C. Moses, I. Brand, S. Castro, L. Kaelbling, T. Lozano-Pérez, and N. Roy, "Active learning of abstract plan feasibility," in *Proceedings of Robotics Science and Systems*, 2021.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[3] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Y. W. Teh, "Neural Processes," in *Proceedings of ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[4] N. Houlsby, F. Huszar, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," in *Proceedings of NeurIPS Workshop on Bayesian optimization, experimental design and bandits: Theory and applications*, 2011.

[5] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[6] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[8] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002.

[9] F. Doshi-Velez and G. Konidaris, "Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2016.

[10] S. Sæmundsson, K. Hofmann, and M. P. Deisenroth, "Meta reinforcement learning with latent variable gaussian processes," in *Uncertainty in Artificial Intelligence*, 2018.

[11] T. W. Killian, S. Daulton, G. Konidaris, and F. Doshi-Velez, "Robust and Efficient Transfer Learning with Hidden Parameter Markov Decision Processes," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

[12] C. F. Perez, F. P. Such, and T. Karaletsos, "Generalized Hidden Parameter MDPs Transferable Model-based RL in a Handful of Trials," in *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[13] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra, and S. Levine, "Model-Based Meta-Reinforcement Learning for Flight with Suspended Payloads," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.

[14] R. Chen, N. Gao, N. A. Vien, H. Ziesche, and G. Neumann, "Meta-Learning Regrasping Strategies for Physical-Agnostic Objects," in *ICRA Workshop on Scaling Robot Learning*, 2022.

[15] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, 2013.

[16] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning," 2015, p. 9.

[17] J. Wu, J. Lim, H. Zhang, J. Tenenbaum, and W. Freeman, "Physics 101: Learning Physical Object Properties from Unlabeled Videos," in *Procedings of the British Machine Vision Conference 2016*, 2016.

[18] L. Zhang and J. C. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012.

[19] P. Y. Lu, S. Kim, and M. Soljačić, "Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning," *Physical Review X*, vol. 10, no. 3, p. 031056, 2020.

[20] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. Tenenbaum, and S. Levine, "Entity Abstraction in Visual Model-Based Reinforcement Learning," in *Proceedings of the Conference on Robot Learning*, 2019.

[21] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "DensePhysNet: Learning Dense Physical Object Representations via Multi-step Dynamic Interactions," in *Proceedings of Robotics: Science and Systems*, 2019.

[22] D. Zheng, V. Luo, J. Wu, and J. B. Tenenbaum, "Unsupervised Learning of Latent Physical Properties Using Perception-Prediction Networks," in *Uncertainty in Artificial Intelligence*, 2018.

[23] M. Denil, P. Agrawal, T. D. Kulkarni, T. Erez, P. Battaglia, and N. de Freitas, "Learning to Perform Physics Experiments via Deep Reinforcement Learning," in *Proceedings of The International Conference on Learning Representations*, 2017.

[24] K. Rakelly, A. Zhou, D. Quillen, C. Finn, and S. Levine, "Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[25] 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, Dec. 2017.

[26] S. Y. Gadre, K. Ehsani, and S. Song, "Act the part: Learning interaction strategies for articulated object part discovery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[27] R. Martín-Martín and O. Brock, "Coupled recursive estimation for online interactive perception of articulated objects," *The International Journal of Robotics Research*, vol. 41, no. 8, pp. 741–777, July 2022.

[28] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, "Learning of grasp selection based on shape-templates," *Autonomous Robots*, vol. 36, no. 1, pp. 51–65, Jan. 2014.

[29] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2013.

[30] M. Danielczuk, A. Balakrishna, D. Brown, and K. Goldberg, "Exploratory Grasping: Asymptotically Optimal Algorithms for Grasping Challenging Polyhedral Objects," in *Proceedings of the Conference on Robot Learning*, 2021.

[31] L. Fu, M. Danielczuk, A. Balakrishna, D. S. Brown, J. Ichnowski, E. Solowjow, and K. Goldberg, "LEGS: Learning Efficient Grasp Sets for Exploratory Grasping," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2022.

[32] Z. Zhao, X. Li, C. Lu, and Y. Wang, "Center of mass and friction coefficient exploration of unknown object for a robotic grasping manipulation," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, 2018, pp. 2352–2357.

[33] D. Ćehajić, S. Hirche, *et al.*, "Estimating unknown object dynamics in human-robot manipulation tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017, pp. 1730–1737.