GrOMP: Grasped Object Manifold Projection for Multimodal Imitation Learning of Manipulation

William van den Bogert[1], Gregory Linkowski[2], and Nima Fazeli[3]

***Abstract*—Imitation Learning (IL) holds great potential for learning repetitive manipulation tasks, such as those in industrial assembly. However, its effectiveness is often limited by insufficient trajectory precision due to compounding errors. In this paper, we introduce Grasped Object Manifold Projection (GrOMP), an interactive method that mitigates these errors by constraining a non-rigidly grasped object to a lower-dimensional manifold. GrOMP assumes a precise task in which a manipulator holds an object that may shift within the grasp in an observable manner and must be mated with a grounded part. Crucially, all GrOMP enhancements are learned from the same expert dataset used to train the base IL policy, and are adjusted with an n-arm bandit-based interactive component. We propose a theoretical basis for GrOMP's improvement upon the well-known compounding error bound in IL literature. We demonstrate the framework on four precise assembly tasks using tactile feedback, and note that the approach remains modality-agnostic. Data and videos are available at williamvdb.github.io/GrOMPsite.**

I. Introduction

Imitation learning (IL) is a powerful tool for generating complex manipulator behavior for repeatable tasks. Diffusion-based behavior cloning has enabled IL to generate multi-modal trajectories, solving a pervasive interpolation problem [1]. However, all IL methods suffer from compounding errors [2]: as the policy rolls out, small errors in the policy compound resulting in deviation from desired behavior. As such, there are no guarantees that learned high-precision assembly tasks can be as successful as the manufacturing industry requires.

Today, assembly automation is highly dependent on specialized fixtures and end effectors to ensure repeatable performance. These highly structured environments are extremely effective, but also incur a high financial cost if designs are upgraded and fixtures are changed to fit modified parts. The more flexible automation approach we consider in this paper involves IL for manipulation with a non-specialized gripper, such that we cannot assume a rigid grasp on an object where in-hand object poses also incur compounding errors.

To address this challenge, we introduce **Grasped Object Manifold Projection (GrOMP)**, an interactive framework which operates on top of an IL policy to constrain a grasped object to a lower dimensional task space, as outlined in Fig 1. Our method learns a task-space manifold from expert demonstrations and projects IL trajectories to this manifold, removing compounding errors orthogonal to its tangent space. We also provide an interactive reinforcement learning method for selecting the manifold based on observed successes. Finally, we demonstrate GrOMP against vanilla IL—implemented as a Diffusion Policy [1]—on real robot experiments.

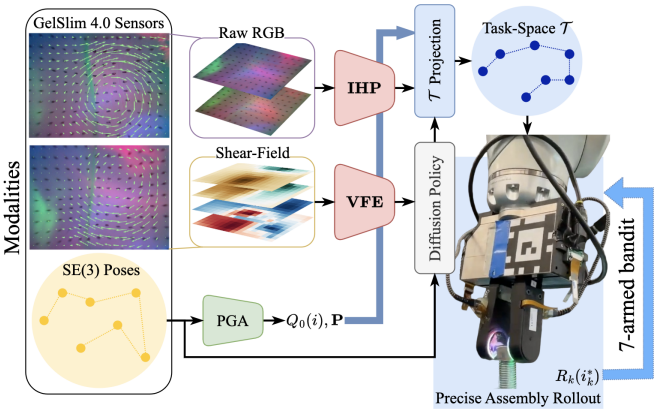

Fig. 1. An overview of Grasped Object Manifold Projection as implemented in this paper. Vision-based tactile sensors (GelSlim 4.0) provide a field of shear displacements and raw RGB images. Shear-fields and proprioception serve as modalities for Diffusion Policy (DP), while RGB images are used for in-hand object pose estimation (**IHP**). DP trajectories and object poses are used to project robot-driven grasped object behavior to the task space $\mathcal{T}$, derived from principal geodesic analysis (PGA) of the expert dataset. A 7-armed bandit adjusts this projection based on rollout rewards.

II. Related Works

Perhaps the most influential interactive behavior cloning algorithms is DAgger [3], which continuously augments the training dataset with the resulting policy rollouts, with the option to query the expert. DAgger demonstrated theoretically and in practice an improvement upon the bound in Eq. 2, and went on to inspire a host of variants. For instance, MEGA-DAgger [4] queries multiple imperfect experts. Human-Gated DAgger (HG-DAgger) [5] allows a human expert to invoke doubt to the novice before contributing to the dataset. These methods are tested on autonomous driving rather than manipulation. Diffusion-Meets-DAgger (DMD) [6] is well-tested on eye-in-hand manipulation tasks and uses diffusion to produce novel task views that become part of the DAgger dataset. DAgger and its variants provide inspiration for this work.

GrOMP does not rely on dataset aggregation as DAgger does. Rather, it assumes an existing, constant, task-specific manipulation dataset, and applies extra understanding of a task manifold to the behavior-cloned policy. This is inspired by several Bayesian and reward-based methods. DropoutDAgger [7] ensures the predicted action is sufficiently close to the

[1] William van den Bogert is with the Mechanical Engineering Department at the University of Michigan, MI, USA willvdb@umich.edu

[1] Gregory Linkowski is with the Robotics & Automation Research Department at the Ford Motor Company, MI, USA glinkows@ford.com

[3] Nima Fazeli is with the Robotics Department at the University of Michigan, MI, USA nfz@umich.edu

expert action and invokes the expert if not. SQIL [8] provides simple rewards for novice behavior that is closer to the expert. CCIL [9] also maintains closeness to the expert by generating corrective labels for training. These methods serve a similar purpose to the projection onto the task manifold learned from expert demonstration as in GrOMP. GrOMP bases its adjustment of this manifold on classical RL, which is used in conjunction with IL in other methods. Inverse Reinforcement Learning (IRL) is a classic method which learns a reward function for RL from demonstration, while Generative Adversarial Imitation Learning (GAIL) [10] extracts a policy directly from demonstrations as if IRL was used.

The state of the art for "vanilla" behavior cloning (imitation learning via learning the state-action map) is Diffusion Policy (DP) [1], where the state-action map is a generative diffusion model. This method has been shown to require >100 demonstrations for manipulation tasks that require medium precision [1]. For GrOMP, we show an improvement upon DP for high precision assembly tasks. In this paper, we demonstrate this with tactile feedback. While DP was initially only tested using visual feedback, it lends itself well to multimodal feedback and has since been tested with tactile feedback, including vision-based tactile feedback. While [11] found DP with tactile feedback to produce zero successful USB insertions, we find some success, with greater success when GrOMP is used.

III. Preliminaries

A. Problem Statement

We assume a manipulator robot is tasked with learning a precise assembly task, wherein one part grasped by its end effector must be mated to another which is fixed in the workspace. We consider a non-rigid grasp. That is to say the transformation $\mathbf{T}_{to}(t) \in \mathrm{SE}(3)$ between the end-effector and the object is not constant, thus we define separately the object state $\mathbf{s}_o \supseteq \mathbf{T}_{so}(t)$ and the robot state $\mathbf{s}_r \supseteq \mathbf{T}_{st}(t)$. In words, both states contain their respective poses, but there may be other elements i.e. forces, contacts, velocity. While the ground truth states are unknown, we assume access to observations (in this paper specifically, robot proprioception and high-resolution tactile sensing at the fingertips).

Under these assumptions, the manipulator must learn to perform this mating from a distribution of initial conditions throughout the demonstrations presented for supervised imitation learning (IL). As with all behavior cloning, we seek the resulting policy $\hat{\pi}$ as a state-action map, where states can consist of multi-modal observations, and actions consist of robot end-effector trajectories of $\mathbf{T}_{st}(t) \in \mathrm{SE}(3)$.

B. Background and Motivation

As presented in the seminal works of Ross et al. [3], [12], an imitation learning policy can be derived using the following supervised learning formulation:

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{\mathbf{s}_r \sim d_{\pi^*}} [\ell(\mathbf{s}_r, \pi)] \tag{1}$$

where Π is the policy class, $d_{\pi^*} = \sum_{t=1}^{T} d_\pi^t$ is the distribution of states under the expert policy π^* after T steps, and ℓ is the observed surrogate loss, which is minimized instead of $C(\mathbf{s}, \mathbf{a})$, the true cost for the particular task. Ross et. al [3] showed a bound for the cost-to-go $J(\pi) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{s}_r \sim d_\pi^t}[C_\pi(\mathbf{s}_r)]$ for the policy π where $C_\pi(\mathbf{s}_r) = \mathbb{E}_{\mathbf{a} \sim \pi(\mathbf{s}_r)}[C(\mathbf{s}_r, \mathbf{a})]$ is the immediate cost of enacting the policy π in the state $\mathbf{s}_r$. Specifically, by assuming $\ell(\mathbf{s}_r, \pi)$ is the expected 0-1 loss of π with respect to π^*, [12] showed the following bound:

$$J(\pi) \leq J(\pi^*) + T^2 \epsilon \tag{2}$$

Where $\epsilon = \mathbb{E}_{\mathbf{s}_r \sim d_{\pi^*}}[\ell(\mathbf{s}_r, \pi)]$. For our problem of learning precise assembly with a non-rigid grasp, we are most interested in controlling the object state $\mathbf{s}_o$ and reformulate the expert policy relying on $\mathbf{s}_o$:

$$\pi^* = \arg\min_{\pi \in \Pi} \mathbb{E}_{\mathbf{s}_o \sim d_\pi} [C_\pi(\mathbf{s}_o)] \tag{3}$$

This reformulation potentially worsens the bound given the uncertainty in the robot commanded actions (our input) and the relative object motion in the grasp (our desired output), particularly due to contacts.

This formulation implies that compounding errors accumulate across all states [12]. However, many tasks, particularly those encountered in manufacturing evolve over lower-dimensional manifolds in the state space. These manifolds are induced by the mechanical constraints of a task (e.g., screw motion for nut and bolt assembly or slide-to-insert for peg-insertion). Inspired by this insight, we propose **Grasped Object Manifold Projection (GrOMP)**, which constrains the object to a lower-dimensional task space manifold $\mathcal{T}$ during IL rollout. Our method results in a combined policy $\hat{\pi}_{\mathbf{s}_o}$ that provides corrections to the base IL policy via projections onto this manifold. Given that the manifold $\mathcal{T}$ is derived from observations of the task-specific expert policy π^*, we assume $\mathcal{T}$ is representative of the task completed by π^*. Thus, we expect the compounding error represented by Eq. 2 are mitigated along directions orthogonal to $\mathcal{T}$.

IV. Method

In this section, we describe our instantiation of GrOMP, first describing the projection which is applied to the IL trajectories, then outlining the selection process of the projection dimensionality from the expert demonstrations, and the n-arm bandit method for adjusting the projection selection from the resulting success rates.

A. Task Manifold Projection

Our goal is to derive a behavior cloned policy using the formulation from Ross et al. (Eq. 1). GrOMP supplements this problem with the mapping $\mathbf{F}_{\mathbf{s}_o} : \mathrm{SE}(3) \to \mathrm{SE}(3)$ which is computed from a lower-dimensional task space manifold $\mathcal{T} \subseteq \mathrm{SE}(3)$ that is representative of the task. This policy seeks the robot pose $\mathbf{T}_{st}$ such that the object pose is projected to the task space $\mathcal{T}$ which is learned from the demonstration dataset using the method described in Section IV-B. We formalize this computation as:

$$\mathbf{T}_{st} = \mathbf{F}_{\mathbf{s}_o}(\hat{\pi}(\mathbf{s}_r)) = \mathbf{F}_{\mathbf{s}_o}(\mathbf{T}_{st}^{\hat{\pi}}) = (\mathrm{proj}_{\mathcal{T}} \mathbf{T}_{so}) \mathbf{T}_{to}^{-1} \tag{4}$$

In words, we first project the object pose w.r.t. the world-frame into the task manifold ($\text{proj}_{\mathcal{T}}\mathbf{T}_{so}$), then compute the goal robot pose from the estimated object pose w.r.t. the robot. We assume that the task demonstrations are sufficiently informative to recover $\mathcal{T}$. To illustrate the importance of formulating the object pose (as opposed to robot pose) constraint on the task manifold $\mathcal{T}$, we imagine a task where the goal is to keep the object stationary w.r.t. the world frame. For such a task, $\dim(\mathcal{T}) = 0$. However, the robot may move in a subset of $\text{SE}(3)$ that will allow the object to be stationary due to the possibility of relative slip. In this case of $\dim(\mathcal{T}) = 0$, $\mathbf{T}_{st} = \mathbf{T}_{to}^{-1}$ so that $\mathbf{T}_{so} = \mathbf{I}$. We assume for our method that $\mathbf{T}_{to}$ is measured via tactile sensors. Alongside the manifold constraint, the IL policy $\hat{\pi}$ would be responsible for manipulating the object within the task space to complete the task.

B. Deriving Task Space from Demonstration

While a task space could be manually selected, in this section we present a method whereby this manifold can be learned from demonstration using principal geodesic analysis (PGA) [13]. To perform PGA, we first project the $\text{SE}(3)$ object poses in the expert demonstration dataset to the tangent space at the mean, using the logarithm map:

$$\boldsymbol{\xi}^* = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} \text{ where } \hat{\boldsymbol{\xi}} = \log((\overline{\mathbf{T}_{\mathbf{so}}^{\boldsymbol{\pi}^*}})^{-1}\mathbf{T}_{\mathbf{so}}^{\boldsymbol{\pi}^*}) = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{v} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} \quad (5)$$

where these $\boldsymbol{\xi}^* \in \mathbb{R}^6$ are the twists and $\overline{\mathbf{T}_{so}^{\pi^*}}$ represents the geodesic mean (calculated as in [13]) on the manifold of $\text{SE}(3)$ of all object poses in the expert dataset with respect to the world frame, and $\hat{\boldsymbol{\omega}}$ represents the skew-symmetric matrix representation of $\boldsymbol{\omega}$ [14].

We then define the normalized twist $\boldsymbol{\xi}_n^*$, such that the rotational and translational components are of similar magnitudes:

$$\boldsymbol{\xi}_n^* = \begin{bmatrix} \boldsymbol{\omega}/\max(||\boldsymbol{\omega}||) \\ \boldsymbol{v}/\max(||\boldsymbol{v}||) \end{bmatrix} \quad (6)$$

We then perform principal component analysis (PCA) via singular value decomposition (SVD) on all normalized twists, represented as the twist matrix $\boldsymbol{\Xi}_{T\times 6}^* = \left[\boldsymbol{\xi}_n^{*,0} \dots \boldsymbol{\xi}_n^{*,T-1}\right]^\top$, where T is the number of data points. We assume the Euclidean mean of all twists is at the origin due to the use of geodesic mean in Eq. 5. This provides the PCA transform $\mathbf{P}_{6\times 6}$:

$$\boldsymbol{\Xi}^* = \mathbf{U}\boldsymbol{\Sigma}\mathbf{P}^\top \text{ where } \mathbf{P} = \left[\boldsymbol{p}_1 \dots \boldsymbol{p}_6\right]$$

We can define a set of $\boldsymbol{\Xi}_P^i$ candidate projected twists by removing the principal degrees of freedom of the least significance, where $i = 0, \dots 6$:

$$\boldsymbol{\Xi}_P^i = \boldsymbol{\Xi}^* \left[\boldsymbol{p}_1 \dots \boldsymbol{p}_i \quad \mathbf{0}_{6\times(6-i)}\right]$$

which implies that $\boldsymbol{\Xi}_P^0 = \mathbf{0}_{T\times 6}$ (a case of no projection). We then map the projected vector back to the original normalized tangent space defined by Eq. 6:

$$\boldsymbol{\Xi}^i = \boldsymbol{\Xi}_P^i\mathbf{P}^\top \text{ where } \boldsymbol{\Xi}^i = \left[\boldsymbol{\xi}_n^{i,0} \dots \boldsymbol{\xi}_n^{*i,T-1}\right]^\top$$

Note that all $\boldsymbol{\xi}_n^0 = \mathbf{0}$, and all $\boldsymbol{\xi}_n^6 = \boldsymbol{\xi}_n^*$. Less degrees of freedom ensures more predictable behavior of the robot, but we emphasize that some degrees of freedom are necessary to complete the task. For this reason, we define a loss $\mathcal{L}_{\text{proj}}^i$ (visualized in Fig 2) for each possible projection $i \in [0 \dots 6]$:

$$\mathcal{L}_{\text{proj}}^i = \frac{\sum_{t=0}^{T-1} \|\boldsymbol{\xi}_n^{*,t} - \boldsymbol{\xi}_n^{i,t}\|}{\sum_{t=0}^{T-1} \|\boldsymbol{\xi}_n^{*,t}\|} + \frac{i}{6} \quad (7)$$

This loss trades off removing degrees of freedom vs accumulating errors. Intuitively, the more constraints the robot has, the less error accumulation at the cost of expressivity. This loss provides a prior for how to best select a projection for a given task; however, it could be sub-optimal due to the dataset or inference procedure. Thus, in the next section, we describe an interactive method to build upon this prior for a given task.

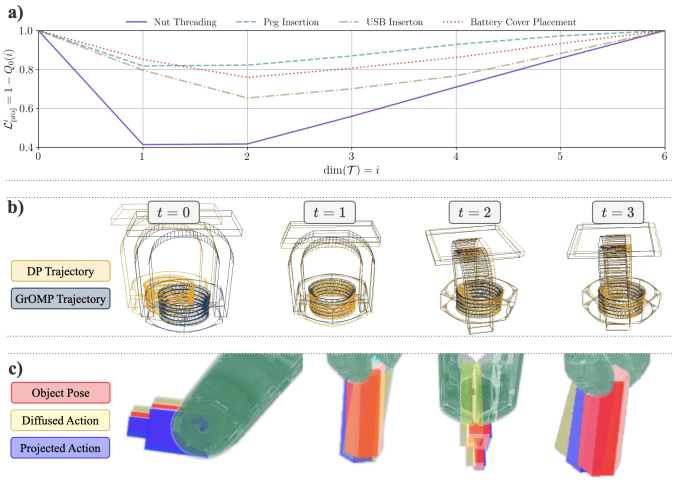

Fig. 2. **a)** Projection loss priors (from Eq. 7) derived from the dataset of each task tested in Section VI. **b)** Projection of the object and robot trajectories (as calculated in Section IV-D) in the expert dataset of nut threading along the manifold determined by Eq. 9, in this case $i_k^* = 2$. **c)** Diffused vs. projected actions visualized for peg insertion and USB insertion alongside the current object pose observation. Here, the action is visualized as the last point in the action trajectory.

C. 7-Arm Bandit Adjustment

As stated in the previous section, Eq. 7 provides a potentially suboptimal prior for how many principal components should describe the task manifold $\mathcal{T}$. Here, we describe an interactive formulation to infer the manifold, starting from the prior given by Eq. 7. To achieve this, we formulate the problem as a nonstationary n-arm bandit [15]. We first initialize the value $Q_k(i)$ of each projection as $Q_0(i) = 1 - \mathcal{L}_{\text{proj}}^i$. The reward of a selected projection $R_k(i_k^*)$ at each iteration k is derived from $\alpha_k(i_k^*)$ successes and $\beta_k(i_k^*)$ failures after $K = \alpha_k(i_k^*) + \beta_k(i_k^*)$ task attempts:

$$R_k(i_k^*) = \frac{\alpha_k(i_k^*)}{\alpha_k(i_k^*) + \beta_k(i_k^*)}$$

Then, we update the value of the selected projection as follows, with step size γ:

$$Q_{k+1}(i_k^*) = Q_k(i_k^*) + \gamma[R_k(i_k^*) - Q_k(i_k^*)] \quad (8)$$

We use an ϵ-greedy method [15] to obtain i_k^* after each K trials, selecting the projection with the highest value:

$$i_k^* = \underset{i \in [0 \dots 6]}{\arg\max}\, Q_k(i) \tag{9}$$

We use Eq. 9 to select i_k^*, except when we make random selection for $i_k^* \in \{0 \dots 6\}$ with a probability of ϵ. We run this iteration over policy rollouts and continuously update our belief over the Q estimates.

D. Trajectory Rollout

In this section, we describe how to obtain $\text{proj}_{\mathcal{T}}\mathbf{T}_{so}$, which is necessary to calculate the robot poses $\mathbf{T}_{st}$ for policy execution (as stated in Eq. 4). In order to do this, we first select the task manifold described by dimensionality i_k^* via the n-armed bandit approach described in the previous section. We then represent the object pose trajectories predicted by the learned policy $\mathbf{T}_{so} = \mathbf{T}_{st}^{\hat{\pi}}\mathbf{T}_{to}$ as a normalized twist matrix $\hat{\boldsymbol{\Xi}}$, following the procedure from Sec. IV-B. Continuing this procedure, we project $\hat{\boldsymbol{\Xi}}$ to the reduced PGA space with i_k^* columns of $\mathbf{P}$, then map back to the original tangent space with $\mathbf{P}^\top$. Finally, we convert the resulting twists back to $\text{SE}(3)$ using the exponential map:

$$\left[\boldsymbol{\xi}_n^{i_k^*,0} \dots \boldsymbol{\xi}_n^{i_k^*,T-1}\right]^\top = \hat{\boldsymbol{\Xi}}\left[\boldsymbol{p}_1 \dots \boldsymbol{p}_{i_k^*} \quad \mathbf{0}_{6\times(6-i_k^*)}\right]\mathbf{P}^\top$$

$$\text{where } \boldsymbol{\xi}^{i_k^*} = \begin{bmatrix}\boldsymbol{\omega}^{i_k^*} \\ \boldsymbol{v}^{i_k^*}\end{bmatrix}, \text{ and}$$

$$\text{proj}_{\mathcal{T}}\mathbf{T}_{so}^{\hat{\pi}} = \overline{\mathbf{T}_{so}^{\pi^*}}\exp(\hat{\boldsymbol{\xi}}^{i_k^*})$$

$$\text{where } \hat{\boldsymbol{\xi}}^{i_k^*} = \begin{bmatrix}\hat{\boldsymbol{\omega}}^{i_k^*}\max(||\boldsymbol{\omega}||) & \boldsymbol{v}^{i_k^*}\max(||\boldsymbol{v}||) \\ \mathbf{0}_{1\times 3} & 0\end{bmatrix}$$

We show a grasped object's trajectory projected along this manifold in Fig. 2.

V. Experimental Setup

In this section, we describe the way in which we tested GrOMP using a tactile and proprioceptive representation of $\mathbf{s}_r$, and a tactile-derived in-hand pose estimator such that $\mathbf{T}_{to}$ is measurable in accordance with our assumptions stated in Section IV-A.

A. Diffusion-Based Behavior Cloning

Our policies are learned with Diffusion Policy (DP) [1], using a cosine noise schedule [16] with K forward steps. For fast rollout of policies, we use a DDIM [17] with $K_{\text{inf}} = K/c$ reverse steps and η parameter. To form a complete data point, an action $\mathbf{A}_t^0$ is coupled with an observation feature $\mathbf{z}_{o,t}$ that is an embedding of multimodal feedback preceding time t. The noise prediction net ϵ_θ is trained using a loss:

$$\mathcal{L}_D = \sum \|\epsilon - \epsilon_\theta(\mathbf{A}_t^k, \mathbf{z}_{o,t}, k)\|$$

ϵ_θ uses a 1D CNN U-Net architecture, as implemented in [1]. Upon rollout of the trained policy, $T_e = 4$ of the actions in $\mathbf{A}_t^0$ are executed.

B. Tactile Perception

In this paper, vision-based tactile sensing from GelSlim 4.0 [18] provides both one of the DP modalities (along with proprioception) and the means with which $\mathbf{T}_{to}$ is measured.

1) Tactile Shear-Field: For DP, we found a shear-field based representation $\boldsymbol{\mathcal{U}}$ to be far more successful for our purposes than the raw GelSlim image representation. To derive the shear-field components (x, y), each a 13×18 matrix, from raw GelSlim RGB images $\mathbf{I}$, we used the **open-cv2** library. Our method uses a function $\mathbf{Flow}$ which calculates the optical flow components from the undeformed frame $\mathbf{I}_0$ to the deformed frame $\mathbf{I}_t$: $(\text{x}, \text{y}) = \mathbf{Flow}(\mathbf{I}_0, \mathbf{I}_t)$. The full input to DP (including encoding) from both L and R fingers on our parallel jaw gripper is $\boldsymbol{\mathcal{U}}_{13\times18\times4} = \{\text{x}^L, \text{y}^L, \text{x}^R, \text{y}^R\}$.

2) Tactile In-Hand Pose Estimation: To recover $\mathbf{T}_{to}$ for GrOMP, we use an in-hand pose estimation module $\mathbf{IHP}$, consisting of a convolutional variational auto-encoder [19] and MLP, with layers as listed in Table I. These are trained simultaneously on reconstruction of the raw RGB GelSlim image (downsampled 16x, 6-channel from two fingers), minimization of KL-divergence of the latent space, and regression to ground truth object poses. This is encompassed in the loss $\mathcal{L}_{\text{ihp}}$:

$$\begin{aligned}\mathcal{L}_{\text{ihp}} = w_{\text{rec}} &\sum \|\{\mathbf{I}_R, \mathbf{I}_L\} - \{\mathbf{I}_R, \mathbf{I}_L\}_{\text{rec}}\| \\ &+ w_{\text{kl}} KL(N(\mu, \Sigma) \| N(0,1)) \\ &+ \sum \|\boldsymbol{x}_{\text{ihp}} - \boldsymbol{x}_{\text{ihp}}^{\text{gt}}\|\end{aligned}$$

Where the in-hand pose $\boldsymbol{x}_{\text{ihp}} = (y_{to}, z_{to}, \theta_{to})$, i.e. we restrict $\mathbf{T}_{to} \in \text{SE}(2)$. We assume in-hand poses outside of this plane are minimal due to our parallel-jaw gripper. Additionally, the latent space $\mathbf{z}_\mathbf{I} \sim N(\mu, \Sigma)$ is derived from the encoding: $(\mu, \Sigma) = \mathbf{IHP}_{\text{Enc}}\{\mathbf{I}_R, \mathbf{I}_L\}$, and is the input for the decoding $\{\mathbf{I}_R, \mathbf{I}_L\}_{\text{rec}} = \mathbf{IHP}_{\text{Dec}}(\mathbf{z}_\mathbf{I})$ and the prediction $\boldsymbol{x}_{\text{ihp}} = \mathbf{IHP}_{\text{MLP}}(\mathbf{z}_\mathbf{I})$. The dataset for training this in-hand pose estimation module is collected via apriltag registration, frequent regrasping, and manual object manipulation, as shown in Fig. 3.

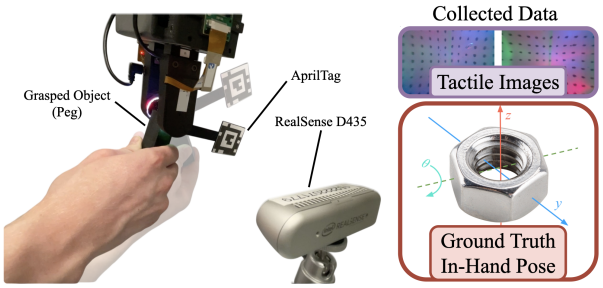

Fig. 3. The tactile and ground-truth $\text{SE}(2)$ object pose data for in-hand pose estimation is collected while the object is grasped between the tactile sensors, and a human manually moves the object in the grasp. The grasp occasionally opens during this collection. Ground-truth object pose data comes from AprilTag registration using a RealSense D435 camera.

C. Observation Encoding

Observations to be encoded are normalized linearly to be in the domain $(-1, 1)$ to work with the diffusion U-Net. The

TABLE I
IHP LAYERS

Segment	Layer	Description
Encoder $\mathbf{IHP}_{\text{Enc}}$	Conv2D	In: 6, Out: 12, 3×3, MaxPool(2)
	Conv2D	In: 12, Out: 32, 3×3, MaxPool(2)
	Flatten	From size $H \times W \times 32$ to size h
	Linear	In: h, Out: 256
	2×Linear	In: 256, Out: $\mu : 256, \Sigma : 256$
	Sample $\sim N(\mu, \Sigma)$	In: $\mu : 256, \Sigma : 256$, Out: 256
Predictor $\mathbf{IHP}_{\text{MLP}}$	Linear	In: 256, Out: 84
	Linear	In: 84, Out: 3
Decoder $\mathbf{IHP}_{\text{Dec}}$	Linear	In: 256, Out: 256
	Linear	In: 256, Out: h
	Reshape	From size h to size $H \times W \times 32$
	ConvTranspose2D	In: 32, Out: 12, 4×4, MaxPool(2)
	ConvTranspose2D	In: 12, Out: 6, 4×4, MaxPool(2)

shear field $\boldsymbol{\mathcal{U}}$ is normalized according to the values found in the dataset. Each component channel $\mathrm{x}^L, \mathrm{y}^L, \mathrm{x}^R, \mathrm{y}^R$ is normalized to $(-1, 1)$. The translational values within the cartesian proprioception $\mathbf{T}_{st}$ are also normalized based on the values in the dataset, with each component normalized to $(-1, 1)$. We convert the rotation matrix within $\mathbf{T}_{st}$ to the 6D representation from [20] which remains un-normalized through the entire process, as suggested for Diffusion Policy [1]. Thus, the entire representation for proprioception is the translation-normalized 9D vector $\boldsymbol{x}$.

We use an observation horizon of just $T_o = 1$. We use a convolutional neural network (CNN), $\mathbf{VFE}$ to encode shear-field observations, the layers of which are described in Table II. $\boldsymbol{x}$ is directly concatenated into $\mathbf{z}_{o,t}$:

$$\mathbf{z}_{o,t} = \mathbf{VFE}(\boldsymbol{\mathcal{U}}_t) \oplus \boldsymbol{x}_t$$

D. Task Demonstrations

We collect a tactile and proprioceptive dataset for each task the robot is to perform; both are important to our implementation of both GrOMP and IL.

1) Demonstration Collection: We collect kinesthetic demonstrations since this methodology is particularly well suited to both precise assembly tasks and tactile sensing. To collect kinesthetic demonstrations, the robot is first placed in a compliant state using impedance control. Then, the human expert manually manipulates the end effector of the robot towards task completion, as shown in Fig. 4.

2) Demonstration Processing: All demonstration episodes for behavior cloning a single task are sampled such that each episode contributes $T_E = 64$ action/observation pairs, where

TABLE II
VFE LAYERS

Layer	Description
Conv2D	In: 4, Out: 16, 4×4, MaxPool(2)
Conv2D	In: 16, Out: 64, 4×4, MaxPool(2)
Flatten	To Size h
Linear	In: h, Out: 128

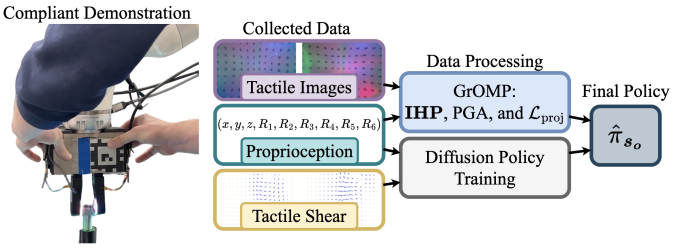

Fig. 4. View of manual demonstration as described in Section V-D1, and the pipeline of the tactile and proprioceptive data toward the generation of the combined policy $\hat{\pi}_{\mathbf{s}_o}$.

$\mathbf{A}_t^0$ consists of $T_a = 8$ cartesian robot poses $\mathbf{T}_{st}$ (converted to 9D pose) within the $\mathrm{SE}(3)$ robot workspace. Using position (as opposed to velocity) as an action space in this way is consistent with [1] and lends itself well to GrOMP. Actions are normalized to $(-1, 1)$ just as observations. We augment the dataset 8 times by adding noise $\sim N(0, 0.1)$ to the normalized tactile shear-field. We use a train-validation split ratio of 80:20. For every 10 episodes, we then have 4096 action-observation pairs to train, and 1024 to validate. The weights of ϵ_θ that produce the lowest $\mathcal{L}_D$ during training are selected for testing.

VI. Experiments and Results

We tested GrOMP as described in Section IV with DP as described in Section V, against pure vanilla DP as a baseline, with four precise assembly tasks: nut threading, peg insertion, USB insertion, and battery cover placement. We treated this as an interactive imitation learning trial where demonstration episodes are added every 10 trials to train new policies. For the DP baseline, we just tested the effect of adding episodes, beginning with 10, then 20, 40, 60, 80, and finally 100. For GrOMP we tested the same, with the effect of the projection to $\mathcal{T}$ (Eq. 4). The initial value $Q_0(i)$ was determined by the first 10 episodes only. We set the value update in Eq. 8 to occur every $K = 1$ trial, with $\gamma = 0.025$. The ϵ-greedy method for selecting the projection dimensionality i_k^* is performed with $\epsilon = 0.1$. Each experiment—a full set of 6 different amounts of training episodes, tested 10 times each—is replicated 4 times. For each of these 4 replications, episodes are introduced in a new random order. As a result, each task is tested with **240 runs for GrOMP and 240 runs for the DP baseline**. Each run is given a maximum policy horizon: $T_E = 64$ action prediction steps to complete the task. Snapshots of the four tasks we test are found in Figs. 5–8.

A. Nut Threading

For nut threading, the object to be assembled is an M20 nut which must be mated with an M20 bolt fixed vertically to the environment. The task is considered successful if the nut cannot be removed from the bolt via a vertical lift (i.e. it must be twisted off). Fig. 5 shows nut threading results.

B. Peg Insertion

For peg insertion, the object to be assembled is a 25 mm wide hexagonal prism peg which must be inserted into a

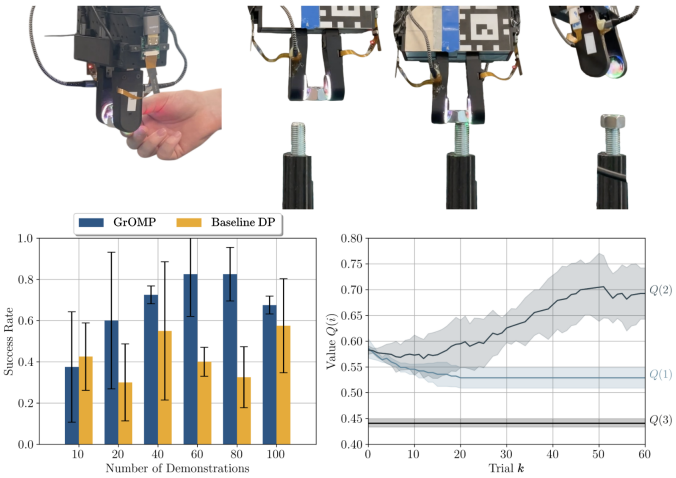

Fig. 5. Nut threading results. **Top:** Handoff, initialization, policy, and success snapshots. **Left:** GrOMP vs DP performance results as demonstrations are added to training. **Right:** Change of highest values in $Q(i)$ from Section IV-C over the 60-trial horizon, averaged over the 4 runs of GrOMP. Filled area surrounding curves represents σ (the initial projection loss in Eq. 7 sometimes yielded $i_k^* = 1$ but $i_k^* = 2$ was the eventual result in all 4 runs of USB insertion).

hexagonal hole with 0.25 mm radial clearance, fixed vertically to the environment. The task is considered successful if the peg falls to the bottom of the hole when released by the gripper. Results for peg insertion can be seen in Fig. 6.

C. USB Insertion

For USB insertion, the object to be assembled is the female end of a USB-A extension cable, which must be inserted into a male USB connector fixed horizontally to the environment. The task is considered successful if a horizontal movement primitive can complete the insertion after the policy horizon has completed. Results for USB insertion can be seen in Fig. 7.

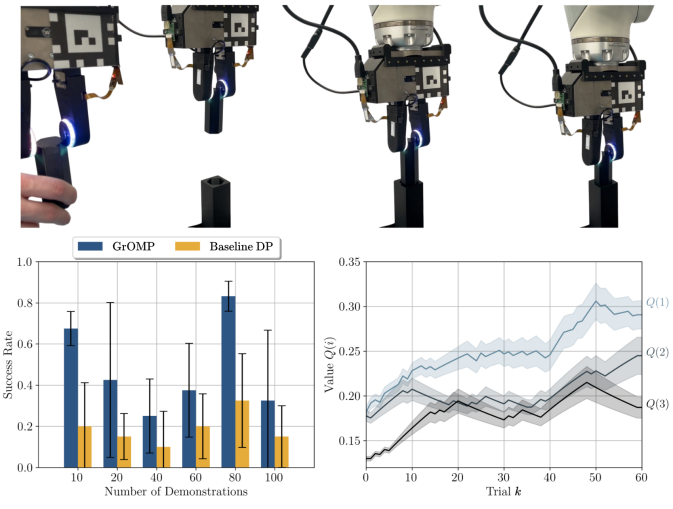

Fig. 6. Peg insertion results. **Top:** Handoff, initialization, policy, and success snapshots. **Left:** GrOMP vs DP performance results as demonstrations are added to training. **Right:** Change of highest values in $Q(i)$ from Section IV-C over the 60-trial horizon, averaged over the 4 runs of GrOMP. Filled area surrounding curves represents $\sigma/8$ (high variance in the result of Eq. 9 occurred between the 4 runs of peg insertion).

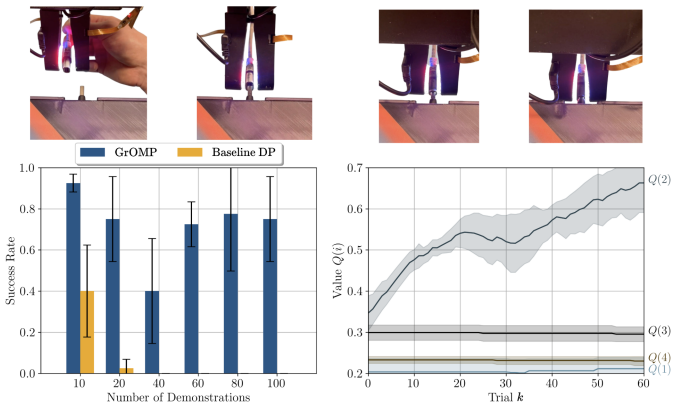

Fig. 7. USB insertion results. **Top:** Handoff, initialization, policy, and success snapshots. **Left:** GrOMP vs DP performance results as demonstrations are added to training. **Right:** Change of highest values in $Q(i)$ from Section IV-C over the 60-trial horizon, averaged over the 4 runs of GrOMP. Filled area surrounding curves represents σ (Eq. 9 always yielded $i_k^* = 2$ in all 4 runs of USB insertion).

D. Battery Cover Placement

For battery cover placement, the object to be assembled is a battery cover for a Roku remote control, which must be mated with the remote control body fixed horizontally to the environment. The task is considered successful if a horizontal movement primitive can complete the mating after the policy horizon has completed. Results for battery cover placement can be seen in Fig. 8.

VII. Discussion

Our results suggest that GrOMP improves task performance over vanilla Diffusion Policy (DP). One key effect of GrOMP is the tendency to avoid unrecoverable states in contrast to

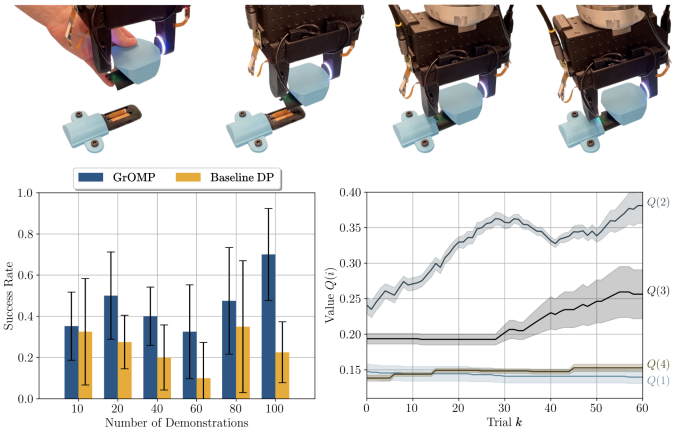

Fig. 8. Battery cover placement results. **Top:** Handoff, initialization, policy, and success snapshots. **Left:** GrOMP vs DP performance results as demonstrations are added to training. **Right:** Change of highest values in $Q(i)$ from Section IV-C over the 60-trial horizon, averaged over the 4 runs of GrOMP. Filled area surrounding curves represents σ (Eq. 9 yielded $i_k^* = 2$ most commonly, with $i_k^* = 3$ being selected near the end of 1 of 4 runs of battery cover placement).

vanilla IL, especially when the object shifts in the grasp. For example, nut threading with DP alone may exhibit behavior where the nut contacts the bolt such that it tilts heavily within the grasp. GrOMP allows the robot to position itself such that the nut remains vertical, even though this robot configuration does not occur in DP's expert dataset.

GrOMP's behavior is beneficially reactive to disturbances. This same kind of tilting can happen in peg insertion, USB insertion, and battery cover placement as well. A secondary recovery strategy requires obstacle avoidance. For instance, our peg insertion task is tested with a hole that does not have an excess of surface surrounding its mouth, instead there is empty space. If the peg is erroneously plunged into this empty space rather than the hole, a re-lift of the peg is required to recover, but this behavior does not appear in the expert dataset, nor is it likely to occur with GrOMP's projection. GrOMP is more likely to prevent this kind of erroneous behavior to begin with, leading to our more successful peg insertion results seen in Fig. 6.

A. Anomalous Baseline Performance

We note that our results do not always show a positive relationship between the number of demonstrations and task success rates which contradicts conventional wisdom. We particularly did not expect the seemingly negative trend between these variables for our USB insertion results, where a policy trained on 10 demonstrations can achieve around a 40% success rate while policies trained on more data all but fail completely.

This type of performance degradation has presented in experimental imitation learning literature before. Specifically, in proposing DAgger [3], Ross et al. compared the interactive behavior cloning algorithm with a supervised learning baseline (Diffusion Policy is our supervised learning baseline). Their results showed first no improvement with an increasing number of demonstrations, and then a drop in success rate when the policy is presented with further training examples. Their explanation for this phenomenon stated that similar demonstrations being introduced to training cannot be expected to improve performance, but did not attempt to explain the drop in performance.

Speaking specifically to the task of USB insertion, one paper by George et al. [11] investigated behavior cloning of this task against multiple sensing modalities (visual, tactile, and both) and multiple behavior cloning methods (Diffusion Policy and Action Chunking Transformers) [11]. They fixed the number of task demonstrations at 100, providing us with a close comparison to one combination of factors from our own experiments. Matching our results, they showed that a tactile-only diffusion policy trained on 100 demonstrations achieved no success. Perhaps 10 demonstrations under their implementation would produce some success.

The reason for this paradoxical outcome is debatable, though we hypothesize some form of overfitting is occurring, possibly due to suboptimal, repetitive, or noisy demonstrations. There are also several hyperparameter and architecture choices when it comes to Diffusion Policy. We do note that testing over this set is prohibitively expensive given the need for real-world rollouts. Our results show that GrOMP successfully improves upon Diffusion Policy for these tasks, independent of the possible existence of suboptimal demonstrations or design choices.

B. Limitations

Future improvements may be inspired by GrOMP's limitations. GrOMP currently does not learn a temporal manifold, meaning projecting to $\mathcal{T}$ will not necessarily progress the task temporally. DP retains full responsibility for task completion, while GrOMP attempts to prevent accumulating errors. GrOMP also requires a reliable in-hand pose estimator. Any improvements in the pose estimation method will serve to improve GrOMP. For instance, we demonstrated a predictable $\mathrm{SE}(2)$ estimator given our assumptions (Sec. V-B). GrOMP can be improved if $\mathbf{IHP}$ were expanded to $\mathrm{SE}(3)$ as some works have attempted [21]. We note that tactile may not be available for all systems. A secondary solution would be to use vision techniques to estimate poses; however, these techniques are known to struggle with occlusion.

Our hope is that the inspiration behind GrOMP leads to the consideration of more geometry-based methods on top of the benefits that imitation learning already provides. Along this research path, the data efficiency and precision required by manipulation in fixtureless industrial assembly can be fully realized.

REFERENCES

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[2] A. Block, A. Jadbabaie, D. Pfrommer, M. Simchowitz, and R. Tedrake, "Provable guarantees for generative behavior cloning: bridging low-level stability and high-level behavior," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, (Red Hook, NY, USA), Curran Associates Inc., 2024.

[3] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 627–635, PMLR, 11–13 Apr 2011.

[4] X. Sun, S. Yang, M. Zhou, K. Liu, and R. Mangharam, "Mega-dagger: Imitation learning with multiple imperfect experts," 2024.

[5] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083, 2019.

[6] X. Zhang, M. Chang, P. Kumar, and S. Gupta, "Diffusion Meets DAgger: Supercharging Eye-in-hand Imitation Learning," in *Proceedings of Robotics: Science and Systems*, (Delft, Netherlands), July 2024.

[7] K. Menda, K. R. Driggs-Campbell, and M. J. Kochenderfer, "Dropoutdagger: A bayesian approach to safe imitation learning," *CoRR*, vol. abs/1709.06166, 2017.

[8] S. Reddy, A. D. Dragan, and S. Levine, "SQIL: imitation learning via regularized behavioral cloning," *CoRR*, vol. abs/1905.11108, 2019.

[9] L. Ke, Y. Zhang, A. Deshpande, S. Srinivasa, and A. Gupta, "Ccil: Continuity-based data augmentation for corrective imitation learning," 2024.

[10] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[11] A. George, S. Gano, P. Katragadda, and A. B. Farimani, "Visuo-tactile pretraining for cable plugging," 2024.

[12] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 661–668, PMLR, 13–15 May 2010.
[13] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 995–1005, 2004.
[14] R. M. Murray, S. Sastry, and Z. Li, "A mathematical introduction to robotic manipulation," 1994.
[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
[16] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," *CoRR*, vol. abs/2102.09672, 2021.
[17] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2022.
[18] A. Sipos, W. van den Bogert, and N. Fazeli, "Gelslim 4.0: Focusing on touch and reproducibility," 2024.
[19] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *CoRR*, vol. abs/1906.02691, 2019.
[20] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
[21] J. Zhao, Y. Ma, L. Wang, and E. H. Adelson, "Transferable tactile transformers for representation learning across diverse sensors and tasks," 2024.