frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 13 April 2021
doi: 10.3389/frobt.2021.645290

Characterizing Continuous Manipulation Families for Dexterous Soft Robot Hands

*Jiatian Sun, Jonathan P. King and Nancy S. Pollard**

Foam Robotics Lab, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, United States

There has been an explosion of ideas in soft robotics over the past decade, resulting in unprecedented opportunities for end effector design. Soft robot hands offer benefits of low-cost, compliance, and customized design, with the promise of dexterity and robustness. The space of opportunities is vast and exciting. However, new tools are needed to understand the capabilities of such manipulators and to facilitate manipulation planning with soft manipulators that exhibit free-form deformations. To address this challenge, we introduce a sampling based approach to discover and model continuous families of manipulations for soft robot hands. We give an overview of the soft foam robots in production in our lab and describe novel algorithms developed to characterize manipulation families for such robots. Our approach consists of sampling a space of manipulation actions, constructing Gaussian Mixture Model representations covering successful regions, and refining the results to create continuous successful regions representing the manipulation family. The space of manipulation actions is very high dimensional; we consider models with and without dimensionality reduction and provide a rigorous approach to compare models across different dimensions by comparing coverage of an unbiased test dataset in the full dimensional parameter space. Results show that some dimensionality reduction is typically useful in populating the models, but without our technique, the amount of dimensionality reduction to use is difficult to predict ahead of time and can depend on the hand and task. The models we produce can be used to plan and carry out successful, robust manipulation actions and to compare competing robot hand designs.

Keywords: robot hand, manipulation, dexterity, manipulation planning, robot hand design, Gaussian Mixture Model, soft robot, manipulation families

OPEN ACCESS

Edited by:
Giuseppe Averta, University of Pisa, Italy

Reviewed by:
Yufei Hao, École Polytechnique Fédérale de Lausanne, Switzerland
Atsushi Kakogawa, Ritsumeikan University, Japan

****Correspondence:***
Nancy S. Pollard
nsp@cs.cmu.edu

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *22 December 2020*
Accepted: *01 March 2021*
Published: *13 April 2021*

Citation:
Sun J, King JP and Pollard NS (2021) Characterizing Continuous Manipulation Families for Dexterous Soft Robot Hands. Front. Robot. AI 8:645290. doi: 10.3389/frobt.2021.645290

1. INTRODUCTION

In industrial applications, robotic systems have been successful for decades for certain assembly line tasks such as spot welding (Wang and Guu, 1989). The motivation for automating these tasks comes from the high cost of skilled human labor and the serious risk of injury present in many manufacturing processes. Today, the goals of modern robotics extend beyond the rigid structure of the factory environment, seeking to safely and robustly perform in the relative chaos that is everyday life.

An emerging class of robots designed to address this problem are soft robots made from *intrinsically* soft materials (Deimel and Brock, 2013; Tang et al., 2019; Zhang et al., 2019; Zhou et al., 2019; Li et al., 2020). Apart from being safe, the softness and compliance realized by these robots can

be exploited to reduce the complexity of environmental interactions. For example, the compliance of soft robots allows them to adapt to geometric variations without the need for complex low-level control, a feature shared by robot hands which contain rigid skeletons but may have compliant joints or actuation (Dollar and Howe, 2009; Odhner and Dollar, 2011; Xu and Todorov, 2016; Della Santina et al., 2018; Homberg et al., 2019). Such an exploitation of compliance can be observed in many biological organisms and is therefore a promising characteristic (Majidi, 2014). Soft robots have shown great ability to conform to object surfaces, produce stable grasps, and handle fragile objects gently (Brown et al., 2010; Galloway et al., 2016; Stuart et al., 2017). However, achieving human-level dexterity, including dexterous in-hand manipulation, remains a challenge. Enabling such capabilities requires considerable innovation in hardware, modeling, and control (Marchese and Rus, 2016; Zhou et al., 2018; Abondance et al., 2020).

One very promising approach is to design low-cost, easily manufacturable hand designs which can be customized for specific tasks. For example, King et al. (2018) present a class of novel fully-compliant, tendon-driven soft hands made from off-the-shelf components (**Figure 1**). The primary structure of this type of robot consists of a flexible foam core. The softness and flexibility of the foam hand have been shown to be of great advantage for stable and secure grasping, and robust in-hand manipulation (King et al., 2018). However, these hands do have trade-offs. The advantages gained by softness come at the cost of necessarily more complex control and planning strategies. This is due to the uncertainties created by complex contact mechanics, deformable materials, and the lack of a compact Forward and Inverse Kinematics representations (i.e., solve FEA instead of using rigid body transformations with joint angles). Another trade-off is that soft robots have a lower upper-bound on the velocities and forces they can operate at due to their low stiffness, thus may not be suitable for high speed or high force industrial processes. However there are very complex manipulations that do not require high speeds or high loads, such as tasks done day-to-day by human hands. At these kinds of speeds we have not observed performance issues with the kinds of soft hands shown in this paper.

The goal of this research is to provide an intermediate representation of soft hand capabilities to facilitate control and planning, as well as to discover the inherent competence or dexterity of a given hand design. Specifically, we present an approach to identify and model continuous families of manipulation actions for these robot hands. The models described in this paper can be used directly for planning and control, as successful actions can be sampled from the model or combined in a planner. A planner that has access to a large family of strategies to achieve a manipulation goal can be robust to different environments: if it cannot achieve its goal with its primary strategy, rather than being stuck with failure, it can explore secondary planning strategies and beyond. We note that some manipulation actions are multimodal; in developing our models, we identify discrete different ways in which target manipulations can be accomplished. Identifying and separating multiple modes of performing manipulations provides a valuable tool for both planning and local control. A planner, for example, can attempt to formulate a plan using each potential mode and select the option with greatest utility for the current scenario.

These models presented in this paper also provide a representation of a robot hand's capabilities. This representation

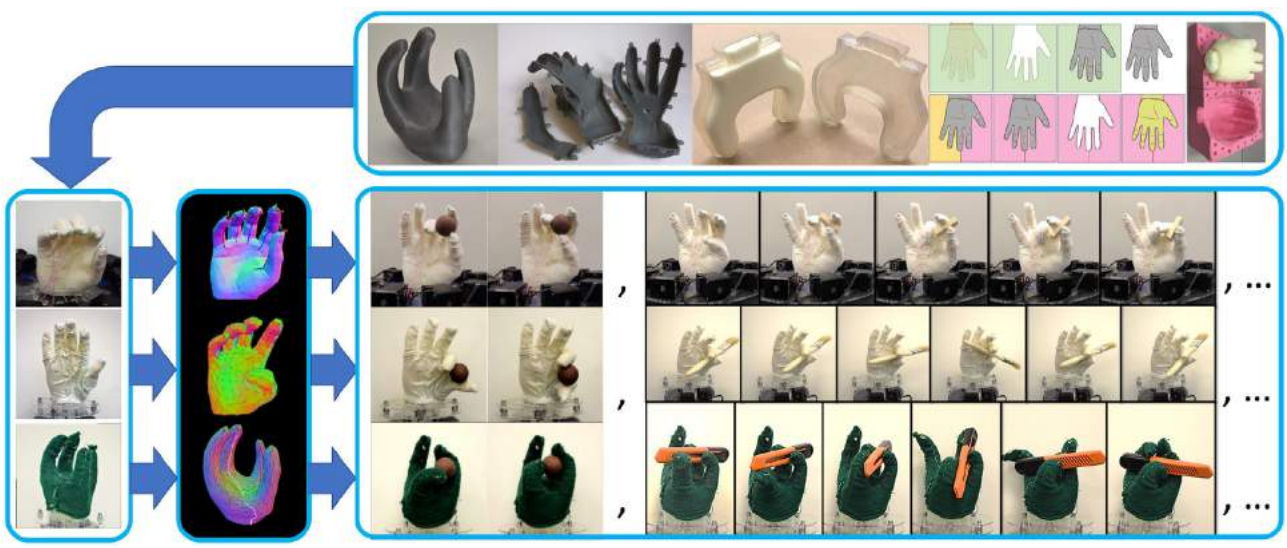

FIGURE 1 | From the top in the direction of the arrows we show: (1) Highly diverse, highly complex hand morphologies can be rapidly and in-expensively fabricated using simple 3D-printing and casting methods. (2) The hands are driven using embedded tendons integrated with the foam via textile skins. (3) Customized, interactive soft-body simulators allow for exploration of designs, actuator configurations, and planning strategies. (4) A variety of complex and dexterous manipulations can be performed across a wide range of hand designs, demonstrating the capabilities, and tremendous potential of soft foam manipulators.

is in terms of the space of actions that produce successful manipulations. A large action space indicates substantial flexibility in performing a manipulation action—that manipulation can be performed by the robot hand in many varied ways. If a hand has a large, continuous successful action space for a manipulation, it may be fair to classify that manipulation task as "easy" for that hand design. As such, these models can be used to compare hands and refine designs. Hand design parameters can be tuned and optimized, or discrete design elements can be selected by measuring their effect on the size of the action spaces for manipulations that are of particular interest for the robot hand under development.

Finally, we present a pure intellectual curiosity behind this work—given a new hand design, we are driven to see just what that design may be able to accomplish. Our approach provides an empirical means to begin to achieve that goal and quantify the results.

2. RELATED WORK

The continuously deformable nature of soft robots makes controlling them a challenging problem. Their ability to accomplish motions such as buckling, extension, or bending, results in soft robots having virtually infinite degrees of freedom. Additionally, George Thuruthel et al. (2018) mention nonlinear material effects such as compliance, visco-elastic material behaviors, and hysteresis, as well as the wide range of design and actuation techniques that account for the non-trivial nature of this problem. Previous works have particularly studied the problem of inverse kinematics (IK) which is concerned with finding a mapping between actuator configuration and desired hand configuration (i.e., pose) (Rolf and Steil, 2013; George Thuruthel et al., 2016; Jiang et al., 2017; Schlagenhauf et al., 2018; Bauer et al., 2020). Existing control approaches can be classified into three main categories: model-based or model-free controllers, as well as combinations of both. The model we refer to here follows the definition of model by Sutton and Barto (2018), that is 'a model of environment, something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave'.

Model-based controllers rely on the establishment of a kinematic mapping from which the actuation can be directly inferred for the desired configuration. A popular approach is to use lumped parameter models and pseudo rigid-body models (Saunders et al., 2010), model caterpillar-like soft robots as a series of extensible linkages. For tentacle-shaped soft robots, Marchese et al. (2014), Marchese and Rus (2016), and Chen et al. (2013) use piecewise constant curvature models to model the robot. For soft robots with arbitrary shapes, Duriez (2013) presents a real-time solution using a finite element method (FEM).

Model-free approaches offer a wide variety of data driven techniques to control soft robots. Neural networks have successfully been used to learn inverse kinematics on a cable driven soft tentacle manipulator with 2 degrees of freedom (Giorelli et al., 2015). Rolf and Steil (2013) have proposed an exploration algorithm for creating task space samples for IK learning. Model free approaches have been successfully used to learn grasping and manipulation tasks for soft hands from demonstration (e.g., Gupta et al., 2016; Della Santina et al., 2019).

Our approach is model-free in the sense that while we sample data from simulation, we do not reason explicitly about the system configuration. Instead we observe the effect of an action toward achieving an intended manipulation (e.g., reorientation of an object within the robot hand).

The challenge of working with data of this sort is three fold. First, the sample space is usually large in dimension, so we need efficient sampling methods. Second, we must determine which degrees-of-freedom, or dimensions, are significant. And third, we require appropriate models for representing planning strategies.

We have identified solutions to these problems by using well known methods like stratified sampling (Mitchell, 1996; Dellaert et al., 1999), principal component analysis (Jolliffe and Cadima, 2016), and Gaussian Mixture Models (GMM) (Reynolds, 2009), to tackle the problems of sampling, significant dimension determination, and modeling approach, respectively.

Prior art has utilized similar building blocks. For example, Khansari-Zadeh and Billard (2011) use Gaussian Mixtures to ensure local asymptotic stability of dynamical systems, Nguyen-Tuong et al. (2008) use Gaussian Process Regression to achieve real-time online model learning, Calandra et al. (2016) use Manifold Gaussian Processes on non-smooth complex functions such as contact mechanics, Bidan Huang et al. (2013) use GMMs to generate grasps given a starting object-hand configuration, Lawrence (2005) use Dual probabilistic PCA and Gaussian Processes for dimensionality reduction, and Engel et al. (2005) use Gaussian Processes with Temporal Differences to learn a controller for a soft octopus tentacle. What distinguishes our approach is that we utilize GMMs to form *global models* of *successful action spaces*. Action spaces can be modeled in many different ways, and we give results from two concrete examples in this manuscript. However, for any model, we aspire to capture the entire valid space of possible manipulations as one or more continuous action regions.

3. METHOD

We propose a sampling based method to identify and model families of manipulations for soft robot hands. Given a manipulation goal for a soft hand, our approach aims to discover and model large continuous action spaces that allow the manipulation goal to be successfully accomplished.

We elaborate on the design of our method in the sections below. In section 3.1, we explain how we represent the physical environment and the soft hand's interaction with that environment, including defining manipulation goals and action spaces. In section 3.2, we unfold our three-step method for generating models of continuous successful action spaces to achieve manipulation goals. Experiments used to test the validity of our method are discussed in section 4.

TABLE 1 | Variable Table.

Variable	Description
$\boldsymbol{\theta}$	Single parameter sample, defining an open loop action for the foam hand
action space	Continuous region of parameters $\boldsymbol{\theta}$ that may achieve some goal (e.g., result in performing a manipulation task successfully)
Θ	Set of parameter samples $\boldsymbol{\theta}$
Ω	Parameter space from which $\boldsymbol{\theta}$ can be sampled for a foam hand
d	Dimension of Ω
$\phi(\boldsymbol{\theta}, t)$	Controller function, which produces an action from parameter $\boldsymbol{\theta}$ and time t
$\boldsymbol{a}_t$	Action performed at time t, defined as an open loop control actuation command
d_a	Dimension of an action; degrees-of-freedom in the action space for a single time t
$\boldsymbol{s}_t$	State of the observed physical system (including manipulated object) at time t
$\boldsymbol{s}_{goal}$	Goal state of the observed physical system.
d_s	Dimension of the state vector; observed degrees-of-freedom in the physical system
score	Weighted difference between observed state and goal state in a manipulation
b	Coefficient vector weighting state differences for computing a manipulation score
$c_{minScore}$	Minimum score required to consider one parameter $\boldsymbol{\theta}$ as successful.
$n_{initial}$	Target number of successful samples to collect in the initial sampling process
n_{sample}	Number of samples to test in one iteration of the initial sampling process
s	Scale factor $s > 1$ controls exploration beyond observed successes in initial sampling
m	Number of PCA dimensions for a proposed GMM, $m \leq d$
k	Number of Gaussians for a proposed GMM
$p(\boldsymbol{\theta})$ or $GMM_{m,k}$	Gaussian Mixture Model built over Θ with hyperparameters m, k
$p'(\boldsymbol{\theta})$	GMM whose Gaussians have normalized inverse weights of $p(\boldsymbol{\theta})$
ϵ_{w_std}	Threshold to achieve sufficiently uniform GMM weights
ϵ_{w_diff}	Threshold to indicate progress is not being made on equalizing GMM weights
ϵ_{succ}	Threshold to indicate a match for desired GMM success rate
$\boldsymbol{w}_{old}$	Old weight of Gaussians constructed in the model building process
$\boldsymbol{w}_{new}$	New weight of Gaussians constructed in the model building process
c_{succ_rate}	Target (required) success rate for each proposed GMM

3.1. Problem Definition

In this section, we formally define the soft robot hand's actions, states, and manipulation goals. **Table 1** contains a listing of variables and their descriptions. Specific instantiations of these variables are given for two robots in sections 4.1 and 4.2.

We start with the definition of actions of a soft robot hand. We assume that the soft robot hand is driven by its controller. The controller takes a parameter vector $\boldsymbol{\theta}$ as input. $\boldsymbol{\theta}$ corresponds to attributes that could be used to generate a control signal, things like positions, joint angles, forces, torques, as well as their combinations. This parameter vector is drawn from a parameter space Ω which is established ahead of time by the user. The specific definition of the parameter space depends both on the robot hand design and the choice of control. For example, one possible set of parameters is a list of actuator commands and associated timestamps, which could be used to actuate the hand with an open-loop PD controller.

Given a parameter vector $\boldsymbol{\theta}$, the controller generates a sequence of open-loop actuation commands called actions, $\boldsymbol{a}_{t=1...T} \in \mathbb{R}^{d_a}$, where d_a represents the number of controlled degrees-of freedom. The controller actuates the foam hand by executing the action generated by $\boldsymbol{a}_t = \phi(\boldsymbol{\theta}, t)$ for all T time steps.

As the soft foam hand's controller executes its actuation commands, its surrounding physical environment would be affected by its behavior. For example, the action may result in manipulation of a grasped object. To represent this change in the system, we define the state of the physical system containing the soft hand for any time step t as a vector, $\boldsymbol{s}_{t=1...T} \in \mathbb{R}^{d_s}$, where d_s represents the number of observed degrees-of-freedom. The physical system always starts with a rest state $\boldsymbol{s}_0$ and ends with a final state $\boldsymbol{s}_T$. The state vector contains numbers which correspond to measurable attributes of the scene at a given time step such as positions, velocities, forces, or even raw sensor values.

The objective of the controller is for the final state to be $\boldsymbol{s}_{goal}$. At each time step t, the foam hand controller can interact with the system by executing $\boldsymbol{a}_t$, which transforms the system according to the state update function, $\boldsymbol{s}_{t+1} = \texttt{simulator}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. After the controller finishes executing the actions, we compare the final state of the system $\boldsymbol{s}_T$ against the goal $\boldsymbol{s}_{goal}$ to evaluate $\boldsymbol{\theta}$'s effectiveness in manipulating the soft hand. Specifically, we generate a score, $score = \boldsymbol{b} \cdot \|\boldsymbol{s}_T - \boldsymbol{s}_{goal}\|$, taking the inner-product of a linear coefficient vector, $\boldsymbol{b}$, and the normed error vector, $\|\boldsymbol{s}_T - \boldsymbol{s}_{goal}\|$.

To classify the parameter vector $\boldsymbol{\theta}$ used by the controller to generate actions, we compare *score* against a minimum score threshold, $c_{minScore}$, where *successful* or *failure case* occurs if *score* is above or below the minimum threshold, respectively. In practice, this is evaluated using the conditional indicator function:

$$\mathbb{1}(score > c_{minScore}) = \begin{cases} 1 & \text{if } \texttt{True} \\ 0 & \text{if } \texttt{False} \end{cases}$$

Therefore, in the later section of this paper, when we mention *successful* parameters, we refers to those that satisfies the indicator function $\mathbb{1}(score > c_{minScore})$. On the contrary, *failed* parameters have scores below the threshold $c_{minScore}$.

The goal of the research described in this paper is to identify large continuous regions of parameters $\boldsymbol{\theta}$ that result in achieving the user specified goal $\boldsymbol{s}_{goal}$. In other words, we wish to find large continuous regions of $\boldsymbol{\theta}$ that are classified as successful. We informally call these regions action spaces as they represent

Algorithm 1: GMM Model Construction Pipeline

Input:
$\Omega, n_{initial}, n_{sample}, s, \epsilon_{w\_std}, \epsilon_{w\_diff}, \epsilon_{succ}, c_{minScore}, c_{succ\_rate}, \boldsymbol{b}$;
Let $d = \dim \Omega$;

// ____Step 1: Collect initial samples____;
Stratified sample a parameter set Θ from the parameter space Ω;
Evaluate Θ and collect the successful ones as $\Theta_{initial}$.;
while $size(\Theta_{initial}) < n_{initial}$ **do**
  Build a Gaussian model $\mathcal{N}(\mu, \Sigma)$ to cover the parameter set $\Theta_{initial}$;
  Sample n_{sample} samples from $\mathcal{N}(\mu, s\Sigma)$;
  Evaluate n_{sample} samples and collect the successful ones as Θ_{new}.;
  $\Theta_{initial} = \Theta_{new} \cup \Theta_{initial}$;
end

// ____Step 2: Build candidate models with different k and m____;
Sample a integer set M of size n_m from $\{1 \dots d\}$;
Sample a integer set K of size n_k from $\{1 \dots d\}$;
$\Theta_{all} = \Theta_{initial}$;
$\Theta_{new} = \Theta_{initial}$;
$\Theta_{old} = \{\}$;
for $m \in M$ **do**
  for $k \in K$ **do**
    // ____Refine model and fill in sparse regions____;
    $\Theta_{new}, \Theta_{old}, GMM_{m,k} =$ BuildModelOnPCA($\{\}, \Theta_{initial}, k, m, c_{minScore}, \boldsymbol{b}$);
    $\boldsymbol{w}_{old} = \{\infty, \infty, \dots, \infty\} \in \mathbb{R}^k$;
    $\boldsymbol{w}_{new} = GMM_{m,k}.weights$;
    while $\max_{i=1}^{k} |w_{new,i} - \frac{\sum_{i=1}^{k} w_{new,i}}{k}| >= \epsilon_{w\_std}$ *and* $sum(\boldsymbol{w}_{old} - \boldsymbol{w}_{new}) > \epsilon_{w\_diff}$ **do**
      $\Theta_{new}, \Theta_{old}, GMM_{m,k} =$ BuildModelOnPCA($\Theta_{old}, \Theta_{new}, k, m, c_{minScore}, \boldsymbol{b}$);
      $\Theta_{all} = \Theta_{all} \cup \Theta_{new}$;
      $\boldsymbol{w}_{old} = \boldsymbol{w}_{new}$;
      $\boldsymbol{w}_{new} = GMM_{m,k}.weights$;
    end
    // ____Trim model to desired success rate____;
    Binary search in between (lo, hi) for r such that $|success\_rate(GMM(\mu, r\Sigma)) - c_{succ\_rate}| < \epsilon_{succ}$;
    Let $GMM_{m,k} = GMM(\mu, r\Sigma)$;
  end
end

// ____Step 3: Select best model for coverage of all successful samples____;
$\Theta_{test} = downsample(\Theta_{all})$;
bestModel = Select model with smallest Mahalonobis distance to Θ_{test};
return bestModel;

Algorithm 2: BuildModelOnPCA($\Theta_{old}, \Theta_{new}, k, m, c_{minScore}, b$)

$\Theta_{newRed} = downsample(\Theta_{new})$;
$\Theta = \Theta_{old} \cup \Theta_{newRed}$;
$\Theta_{pca} = pca(\Theta, m)$;
Build Gaussian Mixture Model(GMM)
$p(\boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i), \boldsymbol{\theta} \in \Theta_{pca}$;
Build GMM $p(\boldsymbol{\theta})' = \sum_{i=1}^{k} \frac{\frac{1}{w_i}}{\sum_{i=1}^{k} \frac{1}{w_i}} \mathcal{N}(\mu_i, \Sigma_i)$;
Sample $\Theta_{lowProb} \sim p(\boldsymbol{\theta})'$;
for $\boldsymbol{\theta} \in \Theta_{lowProb}$ **do**
  for $i = 1, 2, \dots, T$ **do**
    $\boldsymbol{a}_t = \phi(\boldsymbol{\theta}, t)$;
    $\boldsymbol{s}_t$ = simulator $(\boldsymbol{s}_t, \boldsymbol{a}_t)$;
  end
  Calculate $score = \boldsymbol{b} \| \boldsymbol{s}_T - \boldsymbol{s}_{goal} \|$.;
  if $score > c_{minScore}$ **then**
    $\Theta_{succ} = \Theta_{succ} \cup \{\boldsymbol{\theta}\}$;
  end
end
$\Theta_{new} = \Theta_{succ}$;
$\Theta_{old} = \Theta$;
Return $(\Theta_{new}, \Theta_{old}, p(\boldsymbol{\theta}))$

continuous families of actions that the soft robot hand can take in order to achieve the desired result.

3.2. Approach Overview

Our method is designed to identify and model continuous action spaces that allow a soft robotic hand to achieve a manipulation goal successfully. The resulting models are intended for use in manipulation control, manipulation planning, and robot hand design/evaluation.

At high level, our method can be considered as black box that takes in all the physical system information, for example, goal state $\boldsymbol{s_{goal}}$ and initial state $\boldsymbol{s_{initial}}$, and soft hand action information, such as parameter space Ω and controller ϕ, then builds a model to cover the *successful* parameter distribution in Ω, eventually outputting a model to generate parameters for soft hand controllers to achieve their objective of changing their surrounding environment's state to $\boldsymbol{s_{goal}}$.

Specifically, we employ a sampling based approach consisted of three steps, which are shown in **Figure 2**: (1) collect a large number of successful samples of performing the desired manipulation, (2) construct a collection of alternative models, refining each model to "fill-in" sparsely sampled regions and achieve a target success rate, and (3) compare all models for best coverage of an unbiased test dataset formed by selectively downsampling all successful samples ever seen. The approach terminates by selecting and returning the best model. Pseudocode for the full pipeline can be found in Algorithm 1. Each of the three steps will be discussed in detail in sections 3.3–3.5, respectively.

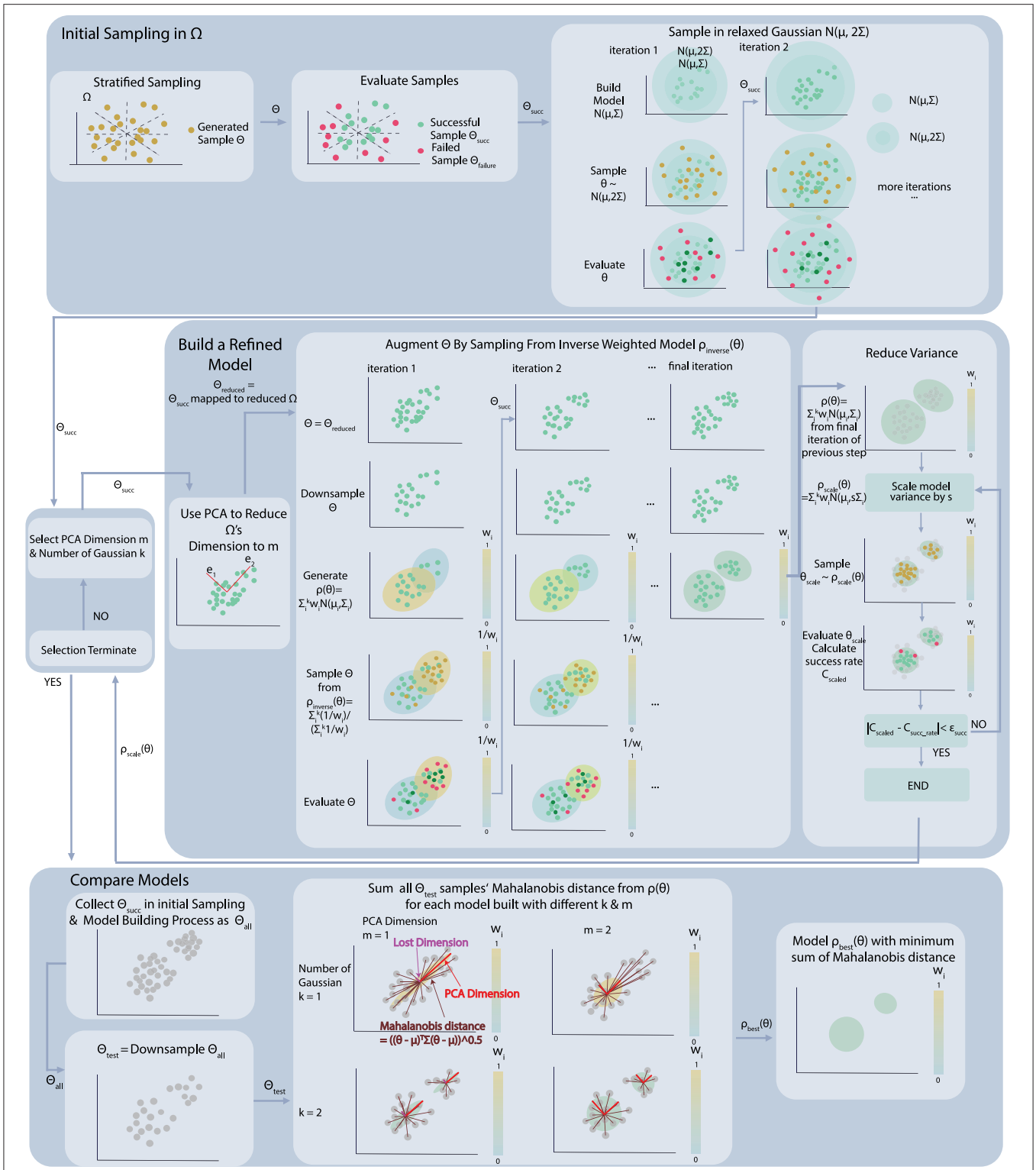

FIGURE 2 | Algorithm flow chart.

3.3. Collect Data in the Full Dimensional Parameter Space

The first step of the algorithm is to collect a large set of successful manipulations. To achieve this goal, we use a two-step process, first collecting data by sampling within the full action space using stratified sampling, and then performing a more focused exploration in the area "near" successful samples that have already been identified.

An initial exploration of the action space is performed using stratified sampling (Mitchell, 1996). The stratification is done by dividing the parameter space $\Omega \in \mathbb{R}^n$ into small sub-regions by subdividing each of the n dimensions. Samples are drawn randomly from within each subregion to ensure some coverage of the entire space. This approach helps induce improved sampling uniformity even with a relatively small sample size.

Once an initial collection of successful samples have been drawn, the search is focused to an area where successful samples are more likely to occur, i.e., we switch to a process of sampling with constraint. The constrained space is constructed by building a Gaussian Model from the existing collection of successful samples, $M_{initial} = \mathcal{N}(\mu, \Sigma) \sim \Theta$, where $\mathcal{N}(\mu, \Sigma)$ is a standard Gaussian where parameters Θ represent successful samples.

To facilitate some exploration within the constrained sampling process, We create a new model $\mathcal{N}(\mu, s\Sigma)$ whose covariance matrix $s\Sigma$ is equivalent to $M_{initial}$'s covariance matrix scaled by a scalar $s > 1$. By sampling from this relaxed Gaussian model $\mathcal{N}(\mu, s\Sigma)$, we increase the probability of collecting successful parameters compared to the original parameter space Ω, without limiting the new sampling domain to only the successful cases we have encountered thus far. The examples in this paper use $s = 2$. We iteratively construct relaxed Gaussian models on the sample set Θ collected *so far* and sample from them until we pass $n_{initial}$, the minimum number of samples required during the initial sampling process.

3.4. GMM Construction

After collecting a suitable number of samples, we move to the next step of the algorithm, the central box in **Figure 2**. The goal of this step is to generate a candidate set of models of the continuous successful action space. The models are Gaussian Mixture Models, and they vary in two parameters: the number of Gaussians selected (k) and the dimension of the space in which they reside (m).

The motivation for testing different numbers of Gaussians (k) is fairly obvious. We wish a representation of our action space that is both compact and comprehensive. As such, we would like to identify the smallest number of Gaussians that covers the continuous successful regions of the action space well (i.e., without missing large successful regions). It is difficult or impossible to predict this number ahead of time, and so we test different values.

The motivation for testing spaces of different dimension (m) is perhaps less obvious. It has been shown repeatedly that many actions that appear complex are in fact highly coordinated (e.g., Safonova et al., 2004; Ranganath et al., 2019). It seems likely that some reduced dimensional space would be adequate for representing the entire successful action family. However, how many dimensions may be needed is again difficult to predict. Here again, we test different values.

The output of this step is a collection of models having different k and m values and an extended set of successful samples, all of which are used to determine a final best model in the third and final step. The next section details the model building process for an individual k and m value, assuming that these values have already been selected.

3.4.1. Dimension Reduction With PCA

We now walk through the process of building and refining a single model (the central block in **Figure 2**). The first step is to identify a suitable reduced dimensional space of size m. For this, we use PCA (Jolliffe and Cadima, 2016), keeping the first m eigenvectors of the projection P as the basis of the space R with reduced dimensions. There are many ways we could take a lower-dimensional projection of our space. PCA is a favorable choice because, by definition, it is an orthogonal linear transformation which maximizes the variance of the projected data. Essentially, all of the *significant* structure is maintained while reducing its span to a volume more tractable for sampling and computation.

$$\begin{aligned} R &= Q\Omega^* \\ C_\Omega &= \frac{1}{n}\Omega^{*T}\Omega^* \\ &= Q\Lambda Q^T \end{aligned}$$

Note that to extract the principal components with regard to the covariance of Ω, we first shift Ω to be a "0-centered" space Ω^*, taking the origin to be its mean. Then, we make an eigen-decomposition on the covariance of the centered parameter space Ω^*. The principle components of Ω^* are what we call the eigenvectors of the covariance matrixC_Ω and it is used to transform the basis of the original space Ω to the reduced space R (Jolliffe and Cadima, 2016).

3.4.2. Bias Reduction: Downsampling and Inverse Model

Even with PCA, if we build a model directly on all of the data accumulated in Θ, it would create bias. This bias arises due to the accumulated data containing disproportionate numbers of successes drawn from the initial stratified sampling process. This mean-offset would result in an advantage for the "first-come" data that decreases the likelihood of discovering disjoint strategies. To eliminate this, we downsample Θ. Specifically, we greedily select a subset $\Theta_{sub} = \text{argmax}_{\Theta_{sub} \subseteq \Theta, |\Theta_{sub}| = n_{sub}} std(\Theta_{sub})$, that has the largest subset among Θ's subsets of same size n_{sub}. In this way, we do not favor frequently sampled data, instead encouraging a narrowing of the success frequency-gap.

To further reduce bias in collected samples and allow sparsely populated regions to be "filled-in," we draw new samples from an "inverse GMM model" built on the downsampled data. Specifically, we construct the original GMM model with the usual expectation maximization algorithm similar to the k-clustering process. Then we create a "inverse model" whose weight w_i' for its ith Gaussian is $\frac{1}{w_i}$, which means that the Gaussian of

highest weight would be the one with lowest weight in the "inverse model." This step is based on the assumption that all of the successful samples/actions should be equally good in terms of performing the target manipulation tasks. Thus, each cluster in the space Ω should be considered as equally important. Correspondingly, Gaussian components of our Gaussian Mixture Model representation, corresponding to clusters in the parameter space, should have evenly distributed weight w_i. We used the method of reducing the samples for high weight Gaussians and increasing the samples for low weight Gaussians to balance out the unevenness in the Gaussian weights by encouraging exploration within Gaussian regions that have low probability in the original model.

We continue this iterative process of reducing bias, by repeatedly reducing dimension of downsampled data with PCA, building a model on the data with reduced dimension, creating an inverse model and sampling new data from the inverse model, downsampling the results and adding them to the downsampled set of data accumulated so far. This process is illustrated in **Figure 2**, which shows graphically how an undersampled region covered by a low-probability Gaussian can be populated through the process and brought to nearly equal probability to the Gaussian containing the larger portion of "first-come" data. A similar phenomenon illustrated with actual data is shown in rows 2 and 3 of **Figure 3**, where a sparsely populated secondary region becomes evenly populated after the inverse model sampling process.

The iterative process terminates when the difference between Gaussian Components' weight of is small enough that $\max_{i=1}^{k} |w_i - \frac{\sum_{i=1}^{k}(w_i)}{k}| < \epsilon_{w\_std}$. Occasionally, if the number of Gaussians is not a good match for the geometry of the successful parameter space, this process may be slow to terminate. In that case, we watch for small changes in weights and finally terminate at a maximum total number of iterations (see Algorithm 1).

3.4.3. Reduce Variance to Elevate Success Rate

So far, models have been constructed based on successful samples only. However, the GMM that results will also contain failures. We observe that these failures are typically on the boundaries of the modeled regions. To reduce failures and bring success rate to a desired level, we propose a simple strategy of variance reduction.

Specifically, to increase the final success rate of the model, we linearly scale the covariance matrix Σ for each component Gaussian of the GMM. We use binary search to search for a scale rate r such that model $GMM_{m,k} = GMM(\mu, r\Sigma)$ would have success rate of approximately c_{succ_rate} when the samples drawn from it for planning are tested either on the physical soft hand or in simulation. Models which cannot achieve the desired success rate after a variance reduction of 0.2 are discarded. Over many experiments, only one model had to be discarded due to this criterion, indicating that the assumption of failures occurring on the boundaries of the Gaussian regions is effective in practice.

3.5. Compare and Select Models

In section 3.4, we explained how an individual model with a fixed m and k value is being created. In this section, we want to elaborate on how m and k would be tuned to create the model that best fit the space.

Ideally, we would perform a global search over all plausible parameters m and k. There are multiple ways of doing such a search of optimal parameters. However, empirically, what we found to work well was to do a line search over PCA dimension m, followed by a line search over k. After extensive informal experimentation, we found that this process was efficient and tended to produce a nearly optimum, if not the optimum model.

3.5.1. Squared Distance Between Θ_{test} and the GMM

The final step of the process (bottom section of **Figure 2**) is to compare models and select the bst one. We compare models based on their coverage of all of the successful test data seen so far. To evaluate the parameter space coverage of different models, we create a test data set Θ_{test} by downsampling all successful data. The downsampling method is the same one used in section 3.4.2. The model that fits this data set the best should be the model that covers the greatest amount of variance we have explored so far in the parameter space Ω.

However, it is challenging to directly compare our models' coverage of the test dataset, since they all have different PCA dimension m. To account for the difference in models' dimension, We need a variance comparison practice that encourages models to preserve components of the test data that are within their dimension-reduced space, while penalizing low-dimensional models for reducing too much variance simply as a result of the projection step involved in dimensionality reduction.

Thus, correspondingly, we measure the model's variance coverage by combining "in-model distance" and "out-of-model" distance from each data in the test set. To check the fitness of a model, we would project each test data point in the data set into the model's dimension reduced space of size m and then calculate its Mahalanobis Distance from each Gaussian of the model. Its minimum Mahalanobis Distance from the Gaussian would be considered as its "in-model distance" from the model. On the other hand, to measure the lost variance of a dimension-reduced model, we construct a Gaussian that tightly covers the test data in the full parameter space and calculate the test data's Mahalanobis Distance from this Gaussian in the "lost" dimensions only. This distance forms the "out-of-model" distance. The equations describing the distance metric we use are as follows, where $\odot$ is the element-wise matrix product.

$$Q_{red} = [q_1 \mid q_2 \mid \cdots \mid q_m]$$
$$Q_{lost} = [q_{m+1} \mid q_{m+2} \mid \cdots \mid q_n]$$

where q_i is the ith column vector of Q
and m is the number of PCA dimensions that
we reserve in the model

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} - \boldsymbol{\mu}_{\Omega^*}$$

where $\boldsymbol{\mu}_{\Omega^*}$ is the shift we used to translate space Ω^* to space Ω

$$f(\boldsymbol{\theta}^*) = Q_{red}^T \boldsymbol{\theta}^*$$
$$g(\boldsymbol{\theta}^*) = Q_{lost}^T \boldsymbol{\theta}^*$$

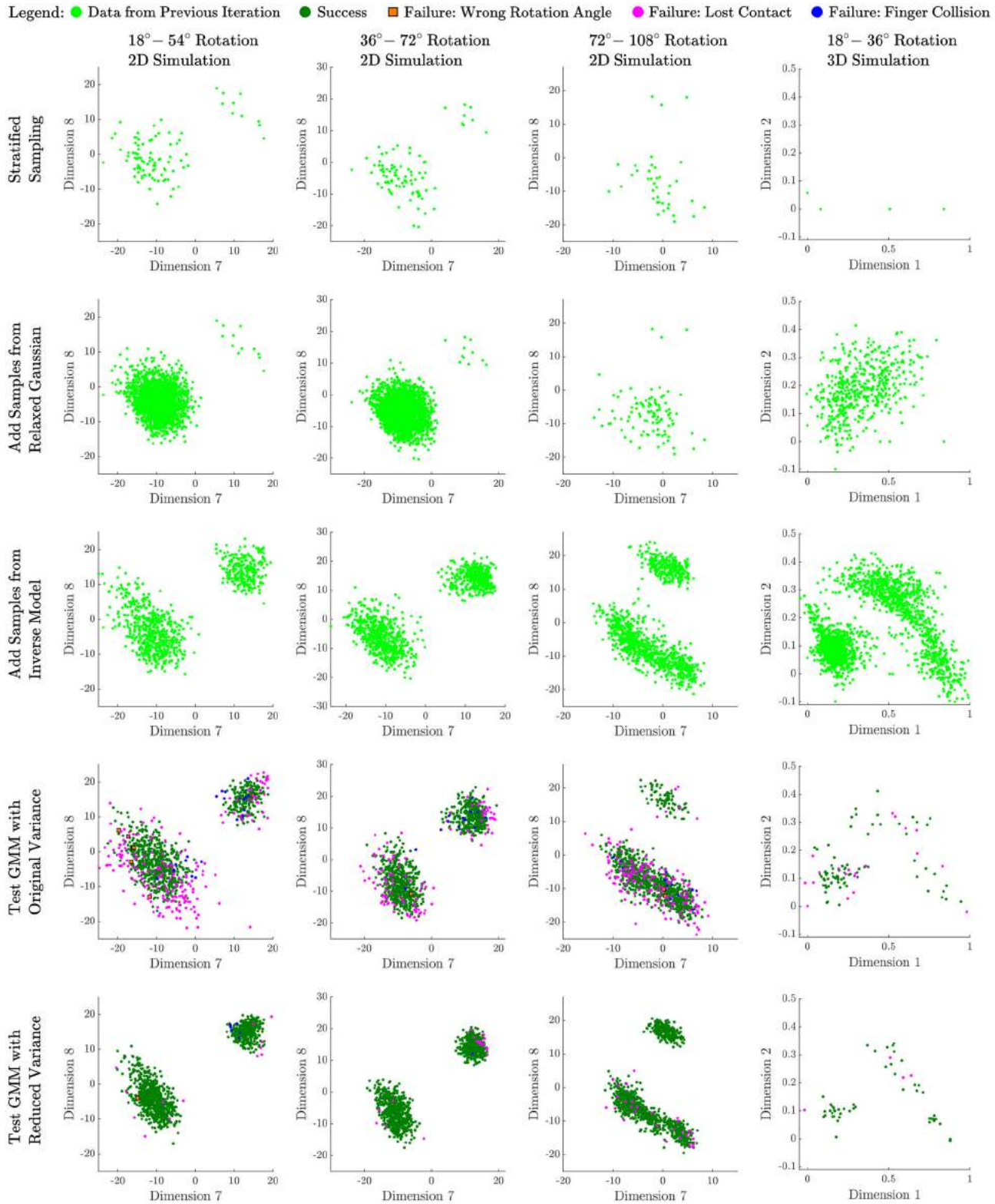

FIGURE 3 | Successful experiments from left to right: Data evolution for our best models using a 2D simulation with angle rotation ranges of 18–54, 36–72, and 72–108 degrees, and a 3D simulation with angle rotation range of 18–36 degrees.

$$\bar{\mu}(\Theta_{test}) = \frac{\sum_{\boldsymbol{\theta}^* \in \Theta_{test}} g(\boldsymbol{\theta}^*)}{|\Theta_{test}|}$$

$$\bar{\sigma}(\Theta_{test}) = \sqrt{\frac{\sum_{\boldsymbol{\theta}^* \in \Theta_{test}} (g(\boldsymbol{\theta}^*) - \bar{\mu}(\Theta_{test})) \odot (g(\boldsymbol{\theta}^*) - \bar{\mu}(\Theta_{test}))}{|\Theta_{test}|}}$$

$$\Sigma_{lost} = \mathtt{diag}(\bar{\sigma}(\Theta_{test}))$$

$$SS\left(\sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)\right) = \sum_{\boldsymbol{\theta}^* \in \Theta_{test}} \left(\min_{i=1}^{k} \left((f(\boldsymbol{\theta}^*) - \mu_i)^T \Sigma_i^{-1} (f(\boldsymbol{\theta}^*) - \mu_i) \right) + \left(g(\boldsymbol{\theta}^*) - \bar{\mu}(\Theta_{test})\right)^T \Sigma_{lost}^{-1} (g(\boldsymbol{\theta}^*) - \bar{\mu}(\Theta_{test})) \right)$$

The sum of squares of each test data's "in-model distance" and "out-of-model distance" forms the total squared distance between a GMM and the test data set. The larger is this total distance, the weaker the model is in covering the parameter space Ω. Therefore, the model with the minimum total distance has the optimal k and m value that results in the best coverage of variance of the original parameter space.

4. RESULTS

4.1. 2D Simulation of a Soft Hand

To test the effectiveness of our method, we design a 2D simulation of a three-finger soft hand. The manipulation goal for the soft hand is 2D rotation of a rigid-body object. Thereby, to fit the manipulation goal, we set up an environment that emulates the general situation that a human hand tries to rotate a thin and elongated object, for example, a pen, in mid air. The setup for the 2D hand is illustrated in **Figure 4** (top).

The state of the system at time step t is represented by the vector $\boldsymbol{s}_t = [x_{l,t}, y_{l,t}, x_{r,t}, y_{r,t}, x_{obj,t}, y_{obj,t}, v_{obj_x,t}, v_{obj_y,t}, \beta_{obj,t}]$, which records the position of left finger tip $\boldsymbol{u}_{l,t} = [x_{l,t}, y_{l,t}]$, the position of the right finger tip $\boldsymbol{u}_{r,t} = [x_{r,t}, y_{r,t}]$, the position and the velocity of the center of mass of the rotated object, $\boldsymbol{u}_{obj,t} = [x_{obj,t}, y_{obj,t}]$ and $\boldsymbol{v}_{obj,t} = [v_{obj_x,t}, v_{obj_y,t}]$, and the rotated angle of the rigid object $\beta_{obj,t}$.

The action command at time step t is defined as the concatenation of the two active fingers' positions $\boldsymbol{a}_t = [u_{l,t}, u_{r,t}]$. The action sequence for two fingers is generated by two cubic Bezier splines, each of which takes four control points as its input. Thus, the action is parameterized by $\boldsymbol{\theta} = [\boldsymbol{p}_{l,1}, \boldsymbol{p}_{l,2}, \boldsymbol{p}_{l,3}, \boldsymbol{p}_{l,4}, \boldsymbol{p}_{r,1}, \boldsymbol{p}_{r,2}, \boldsymbol{p}_{r,3}, \boldsymbol{p}_{r,4}] \in \mathbb{R}^{16}$, which is a vector containing all eight control points $\boldsymbol{p}_{f=\{l,r\},i=\{1,\ldots,4\}}$ to create two cubic Bezier curves for two active fingers. The corresponding action generation function is $\boldsymbol{a}_t = [Bezier([\boldsymbol{p}_{l,1}, \boldsymbol{p}_{l,2}, \boldsymbol{p}_{l,3}, \boldsymbol{p}_{l,4}], t), Bezier([\boldsymbol{p}_{r,1}, \boldsymbol{p}_{r,2}, \boldsymbol{p}_{r,3}, \boldsymbol{p}_{r,4}], t))]$.

$$Bezier([\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4], t) = (1-t)^3 \boldsymbol{p}_1 + 3(1-t)^2 t \boldsymbol{p}_2 + 3(1-t) t^2 \boldsymbol{p}_3 + t^3 \boldsymbol{p}_4, \quad 0 \le t \le 1$$

Eventually, in the 2D simulation, we want to find families of successful rotation trajectories generated from continuous families of parameter $\boldsymbol{\theta}$. In this experiment, we explore the parameter space using the sampling based approach outlined in section 3, Algorithm 1, and **Figure 2**. Based on this setup of 2D simulation, if we can observe a variety of successful rotation actions being generated via our method, than we achieve our goal of creating diverse manipulation strategies for soft hands.

To examine our methods' ability to capture families of strategies for a soft foam hand, we set up three experiments to enable the simulated 2D hand to rotate a slender cuboid shaped object over different ranges of angles around an axis parallel to our line of sight. Those ranges are A:(18–54), B:(36–72), and C:(72–108), with all angles measured in degrees.

Criteria for success specifically were (1) all fingers must be in contact with the rod at the end of the action sequence, (2) the final rotation angle falls in the designated range, and (3) there are no collisions between fingers in the action sequence. A filmstrip illustrating a successful manipulation is shown in **Figure 5**.

During initial sampling, which included both stratified sampling and collecting samples from the relaxed Gaussian, 1,990, 2,436, and 102 successful samples were collected for tests A, B, and C, respectively. The number of successful samples for test C is much lower in the initial steps due to the relative difficulty of the task. The final number of successful samples that were collected as a result of all of the model building processes was 31,250, 27,172, and 28,078 for tests A, B, and C, respectively. Running times were approximately 5.5 h for collecting initial samples and 1.6 h for building, refining, and reducing variance for a single choice of model with a single choice of m and k on a 1.4 GHz Quad-Core Intel Core i5.

Results for three different 2D simulation tests are shown in **Table 2**. Each section of the table gives results for models having different numbers of PCA dimensions m and Gaussians k. The table shows the amount by which variance had to be scaled in order to reach the required success rate. It also shows the Mahalonobis distance of the resulting model for fitting the test dataset.

As can be seen from the table, a number of models having PCA dimensions from 7 to 13 and number of Gaussians ranging from 5 to 11 are seen to perform fairly well. However, there are some clear winners: the 11-dimensional model for test A and the 9-dimensional model for tests B and C, all with 7 Gaussians. Note that models with very small PCA dimension or very low numbers of Gaussians tend to perform poorly.

4.2. 3D Simulation

3D soft hand simulation was also created to test this method. The setup for the 3D simulation is shown in **Figure 4** (bottom). The 3D simulation is implemented with the SOFA stdlib library, a simulation library for deformable objects (Coevoet et al., 2017). We design a 3D soft hand geometry shown in **Figure 4**, which has three cylinder shape fingers being planted on a roof shaped palm. The three fingers are actuated by 9 vertical tendons planted on their surfaces. We set the goal of our 3D hand to be rotating an object to a certain degree with respect to the y axis. Then if our method can capture the parameter space of 3D soft hand's

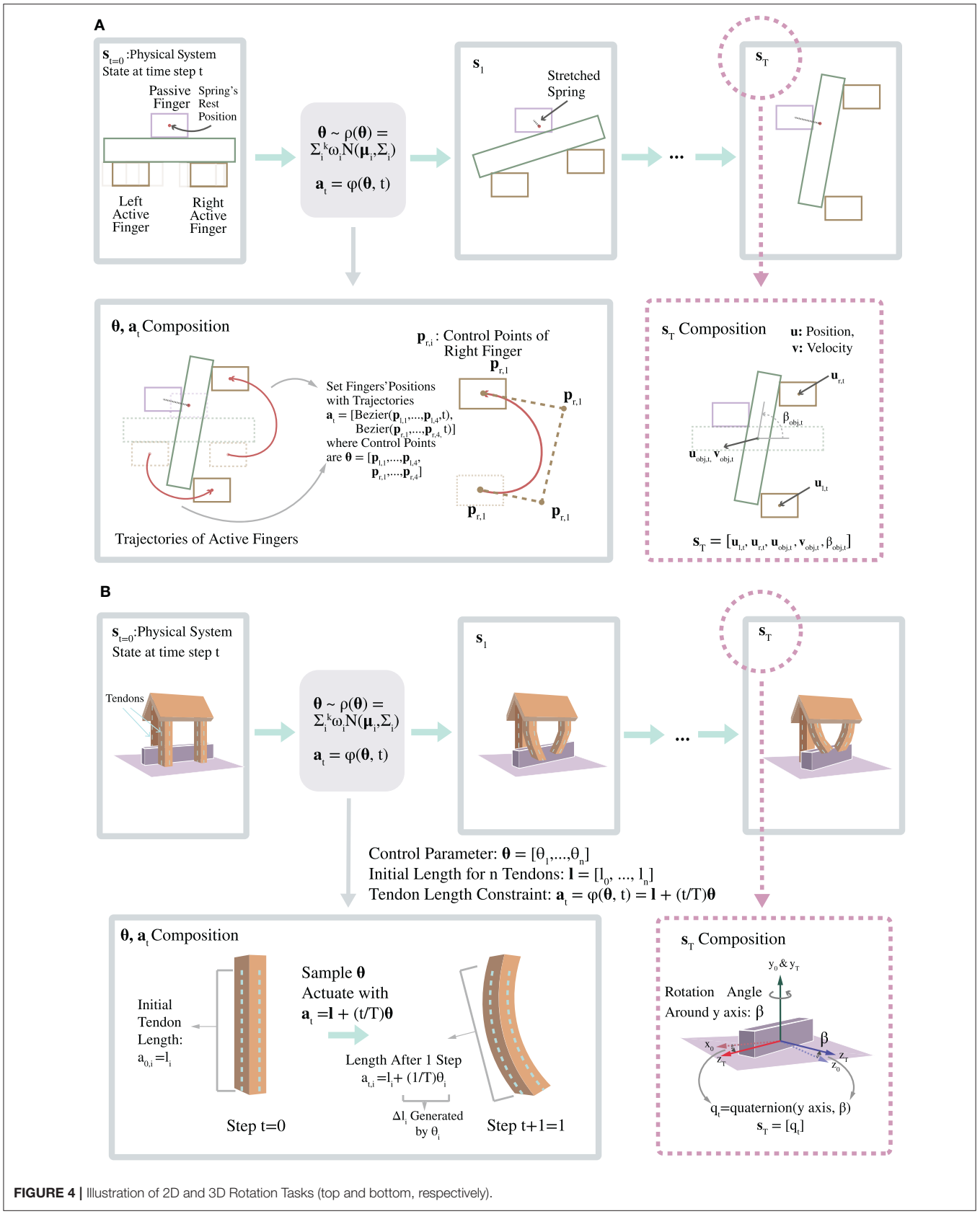

FIGURE 4 | Illustration of 2D and 3D Rotation Tasks (top and bottom, respectively).

action accurately, it should generate actions that can successfully rotate an object to a fixed range of orientations in a variety of ways.

We record the state of the 3D simulation as a vector $\boldsymbol{s}_t = \boldsymbol{q}_t$ at time t, where $\boldsymbol{q}_t \in \mathbb{H}$ is the quaternion of the rigid object being rotated.

The action command at time step t is $\boldsymbol{a}_t = \left[a_{t,0}, \ldots, a_{t,8}\right]$, where and $a_{t,i}$ is the constraint length of the ith tendon at time step t. The action command in our setup is parameterized by the final actuation of all nine tendons $\boldsymbol{\theta} = [\theta_0, \ldots, \theta_8] \in \mathbb{R}^9$, where θ_i is the final actuation of the ith tendon. We generate the action command sequence by gradually shortening the tendons toward their final length. Assuming that ith tendon begins at an actuation of zero and length of l_i, the action at time step t can be rewritten in terms of $\boldsymbol{\theta}$ and $\boldsymbol{l} = \left[l_0, \ldots, l_8\right]$ as $\boldsymbol{a}_t = \boldsymbol{l} + \frac{t}{T}\boldsymbol{\theta}$.

To investigate the action space of this simulated 3D soft hand, we model the distribution of parameter $\boldsymbol{\theta}$ with our method. Therefore, if our model accurately captures the behavior of the 3D soft hand, the samples drawn from our model should result in different families of actions successful in achieving the rotation task.

We set up experiments to rotate a long and thin cuboid placed on top of a platform by 30 degree about the y axis. Criteria for success for this simulation were (1) the rotation angle is within the designated range (18–36 degrees), and (2) the rod rotates about an axis with magnitude difference from vertical less than a given threshold (0.2 in our experiment). **Figure 5** shows a filmstrip of one successful manipulation.

In this case, 1,170 successful samples were collected during initial sampling, and the final number of successful samples that were collected as a result of all of the model building processes was 10,575. Running times were approximately 4 h for collecting initial samples and 7 h for building, refining, and reducing variance for a single choice of model with a single choice of m and k on a 3.6 GHz Intel(R) Core(TM) i7-4790 CPU.

Results for models built with different k and m and 90% success rate are shown in the bottom section of **Table 2**. Here, the 5-dimensional models performed well, with the model having 5 Gaussians achieving the best results.

4.3. Visualization of Processes and Results

To provide insight into the sampling process and results, we present data collected for all four experiments at four stages of our method. **Figure 3** records the change of sampled data distribution and size for all four rotation tests we set up above. Each column of the figure shows the data collected by one test, while each row corresponds to five different stages of our method.

To facilitate the visualization of the parameter space, for 2D simulation we always choose to show the same 2D slice of data. We found dimensions 7 and 8 to be an informative slice for the 2D simulations, showing two disconnected modes in all three cases. This slice compares the horizontal endpoint of the left finger to the vertical endpoint of the left finger. The region to the top right is a region where the fingers swap places, while the larger region toward the bottom left contains more conventional manipulations, often with much less finger movement.

The first row of the plot records the initial data we collect from the stratified sampling process. The second row shows additional samples drawn from the relaxed Gaussian sampling stage (**Figure 2**, second row). The third row shows the samples that have been collected after downsampling and then refining each model (third row in **Figure 2**). Note how the sparsely populated secondary region (representing swapping locations of the fingers) has filled out in the 2D cases. The fourth row shows

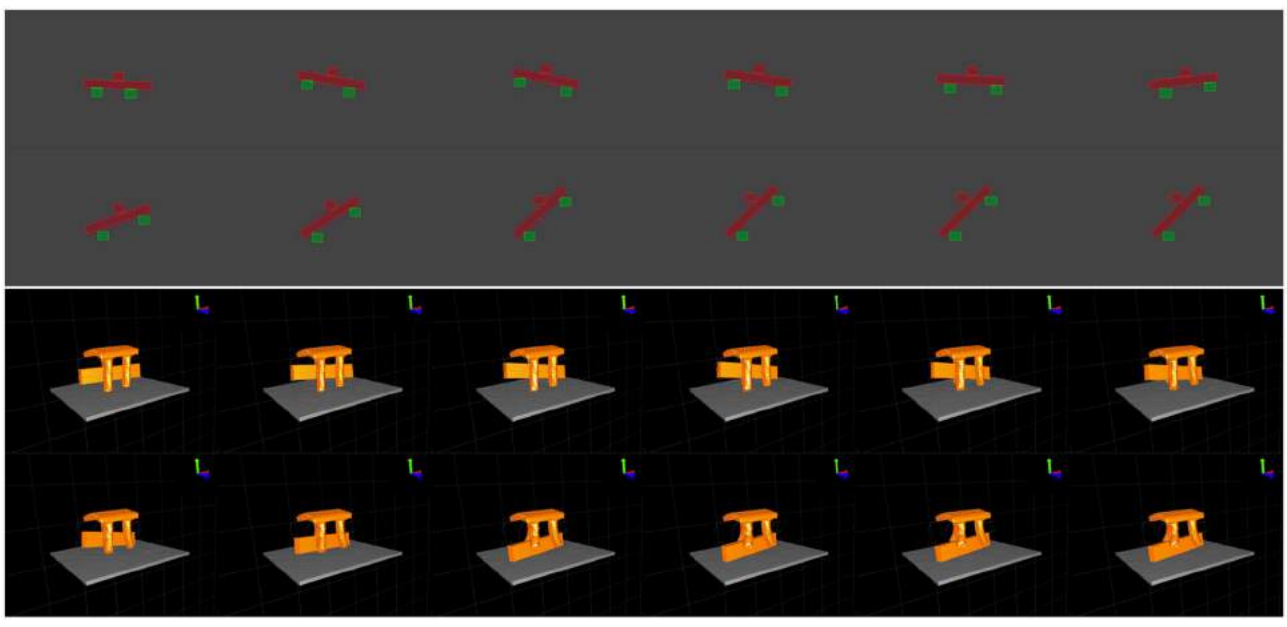

FIGURE 5 | From left to right and top to bottom, frames captured from 2D and 3D simulated manipulations.

TABLE 2 | Simulation results identifying the best models.

PCA Dim. *m*	Num. Gaussians *k*	Variance Reduction Scale *r*	Squared_Distance
2D simulation A: 18–54 degree rotation in 2D with 97% success rate			
3	7	1.0	572.9
5	7	0.49	219.7
7	7	0.39	141.9
9	7	0.4	124.0
11	**7**	**0.4**	**103.9**
13	7	0.3	141.8
11	5	0.29	137.1
11	9	0.4	261.3
11	11	0.35	147.8
2D simulation B: 36–72 degree rotation in 2D with 97% success rate			
3	7	1.0	646.3
5	7	0.6	305.5
7	7	0.62	179.7
9	**7**	**0.51**	**118.0**
11	7	0.501	158.5
13	7	0.495	168.0
9	2	0.51	317.1
9	5	0.5	150.7
9	11	0.55	145.2
2D simulation C: 72–108 degree rotation in 2D with 97% success rate			
3	7	1.0	480.4
5	7	0.449	282.5
7	7	0.255	149.7
9	**7**	**0.425**	**138.2**
11	7	1.0	262.37
9	3	0.229	204.8
3D simulation: 18–36 degree rotation in 3D with 90% success rate			
3	5	0.5	0.203
5	**5**	**0.4**	**0.059**
5	3	0.7	0.089

The bold rows are the models with best performance for the four different experiments in Section 4.

1,000 new samples taken from the final GMM, labeled with their successes and types of failure (100 samples for 3D). The fifth row shows 1,000 new samples drawn after variance reduction (last row of **Figure 2**), also labeled as successes and types of failure (100 samples for 3D). Note how few failures remain in row 5, because the desired success rate has been achieved through variance reduction.

5. DISCUSSION

This section discusses the results presented in section 4. We want to demonstrate first that the dimension reduction of Ω helps us to simplify the object manipulation problem and create models that better fit the space. Afterwards, we continue the analysis of the results by analyzing the visualization of our manipulation control planning strategies sampled from our models in section 4. Eventually, we discuss the method's assumptions and its limitations in solving manipulation problems with higher-degree parameter spaces.

5.1. Benefits of Reducing Dimension of Ω

The benefits of dimension reduction are evident in our results of 2D simulation experiments. We compare our best models against the almost full-dimensional model built with parameters $k = 7, m = 13$. The best models, which have minimum distance from the test data, always have lower dimension than the 13-dimensional model, in all three 2D tests (**Table 2**). Reducing the dimension of parameter space Ω gives us better models than those built on the full-dimensional space because it helps us to limit the sampling space in the iterative data refinement process. The lower dimensional space allows us to focus on exploring the variance of Ω in a few major dimensions, while ignoring the variance in these less significant dimensions of the space. On the other hand, working in the full dimensional space requires us to explore variance of all dimensions, distracting us from fully exploring some major directions. Given limited sampling time, eliminating the less significant dimensions in order to refine the model resulted in more complete exploration and coverage.

5.2. Advantages and Disadvantages of Multiple Gaussians

We dedicate this section to analyze the visualization of our planning strategies, in other words, the process that the simulated soft hand follows our strategies to rotates a rod, in the **Supplementary Videos**.

The visualization shows the advantage that different Gaussians of our model can generate different families of motions. For example, for the 72–108 degree counter-clockwise rotation in 2D, our best model has in total 7 Gaussians, 4 of which initiate motions that keep the left finger at its original position and let the right finger push upward, another Gaussian makes the two fingers exchange their positions and the last two Gaussians force the left finger to leave its original position until the very last moment while letting the right finger to primarily control the rotation of the rod in this process. From the animation, we can tell that there are not only different clusters forming in the parameter space Ω as shown in section 4.3, but also different families of motions shown in the visualization of the strategies sampled from our model in parameter space Ω. The visualization demonstrates the variety of motions could be generated from our planning. Moreover, it demonstrates that different Gaussians of our model can generate different families of motions. Thus, people can use our model to find different manipulation planning strategies and select certain strategies for the hand by picking corresponding Gaussians of our model.

However this "strategy selection by Gaussian" process is slightly inefficient, since each individual Gaussian would not always give rise to motions significantly different from that of another Gaussian. Thus, to determine the Gaussians that are generating similar motions, we need to manually check the animation of their samples. This process could be accelerated by a metric that automatically evaluates the difference between the Gaussians of our model so that it can help us to decide

if two Gaussians generate similar strategies or motions. To design this metric, we can test our method in a lot more different simulation and physical context, so that more data about the correlation between soft hand motions and the parameter space could be gathered to help us understand the relationship between these two space. However, this also means that the metric might only be able to work with certain dynamics model being assigned to either the simulation or the physical hand. The environment/dynamics model assumed by the metric would reduce the application scope of our method and defies our approach's model-free nature. Therefore, for now, the manual process is still beneficial since it keeps our method model-free and applicable to a wider range of soft hand planning scenarios.

5.3. Assumptions and Limitations

Our assumptions about the problem limit our method to only controllers whose parameter space is continuous. On top of that, our method is incapable of discovering successful parameter $\boldsymbol{\theta}$ that are far from the successful data Θ_{succ} collected in the initial full-space sampling process. This is because later steps of constructing the GMM assume that we have already fully explored the boundary of the parameter space in the initial full-space sampling process and proceed to augment the successful data set by filling in the space around existing successful samples. Thus, after we finish the initial full-space sampling process, little variance would be added to the data set later in our method.

Because the coverage of our model is limited to the variance of successful data we discovered in the full-space sampling process, our method would perform less efficiently for very high dimensional parameter spaces. High-dimensional parameter spaces make it harder to hit some successful data in the full-space sampling process, and limit the variance of motions that our final GMM can generate. Therefore, to make our method successful for a high dimensional parameter space, we need to gather more samples from stratified sampling, increasing the running time required for our method.

5.4. Future Directions

There are several future directions for us. One is to further validate the robustness of our method by setting up the real world scenario that our simulation tries to emulate. Testing this method on manufactured soft robots will enable us to evaluate the method's effectiveness for real-world applications.

Another direction is testing out more manipulation tasks for our method. In our current experiments, we only test our method's performance in rotating objects. However, there are many other useful tasks could be considered as our manipulation goals, e.g., grasping, throw, and catch. With more manipulation tasks' models being learned, we can combine these unit manipulation tasks into some more complex behavior.

Also, our method's application can be extended to optimizing design of soft robotic hands. The model we constructed can help evaluate the effectiveness of soft robotic hands' design in achieving certain manipulation goals.

One concrete example of future work is inspired by the results shown in **Figure 6**. The 10 degree-of-freedom soft hand in this figure was specifically designed by Coulson (2020) to accomplish in-hand manipulation tasks. Some examples are shown in the figure. These manipulations were programmed in an open loop manner. Investigation of the entire suite of manipulations showed that the majority could be performed well on the real robot with two keyframes, operating in a reduced-dimensional space of only three dimensions. The set of behavior families corresponding to these manipulations could thus be created using a six-dimensional parameter space Ω and explored in simulation, on the robot, or using a mix of the two modalities, in order to move this collection of pre-programmed capabilities toward a suite of action families that can robustly and flexibly accomplish these manipulations. Apart from the expected value of improving the robustness and flexibility of the interactions that were previously programmed open-loop, competing hand designs can be compared and the hand refined based on the resulting volumes of action spaces generated.

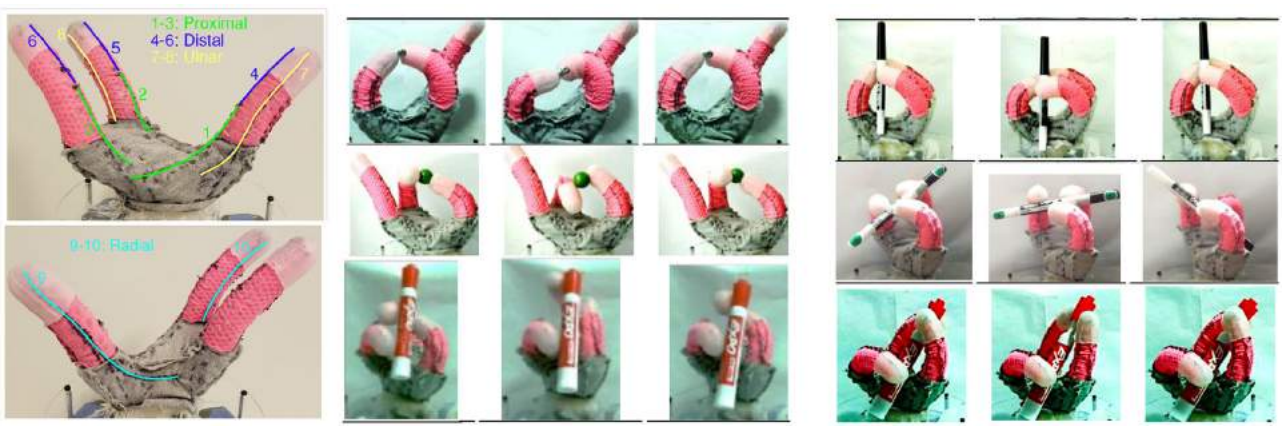

FIGURE 6 | Ten degree-of-freedom dexterous soft foam robot hand developed by Coulson (2020), along with some of manipulations it can perform. The technique described in this paper can be used to identify continuous families of manipulations centered around the actions shown in the future.

We look forward to continuing this line of research and building on these efforts. We believe there is substantial value in attempting to represent a global space of hand capabilities and look forward to creating such representations for soft robot hands such as those represented in this paper. We note, however, that the approach is fully general and can be applied to any robot where it is possible to sample and evaluate an action space.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

JS was responsible for developing the research idea and for producing the majority of the results. JK was responsible for supervising the aspects of the research related to the use of soft robotic hands, as well as for assisting with data analysis. NP was the faculty advisor for the research project. All authors contributed to the article and approved the submitted version.

FUNDING

This research was partially supported by the National Science Foundation awards IIS-1637853 and CMMI-1925130, and in part by a NASA Space Technology Research Fellowship (award 80NSSC17K0140).

ACKNOWLEDGMENTS

The authors would like to acknowledge Dominik Bauer for his assistance with using the SOFA Simulation Library.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.645290/full#supplementary-material

A video supplement demonstrating the results can be found at: https://youtu.be/ZwRDwymXH6Q.

REFERENCES

Abondance, S., Teeple, C. B., and Wood, R. J. (2020). A dexterous soft robotic hand for delicate in-hand manipulation. *IEEE Robot. Autom. Lett.* 5, 5502–5509. doi: 10.1109/LRA.2020.3007411

Bauer, D., Bauer, C., King, J. P., Moro, D., Chang, K.-H., Coros, S., et al. (2020). Design and control of foam hands for dexterous manipulation. *Int. J. Hum. Robot.* 17:1950033. doi: 10.1142/S0219843619500336

Bidan Huang, El-Khoury, S., Miao Li, Bryson, J. J., and Billard, A. (2013). "Learning a real time grasping strategy," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 593–600. doi: 10.1109/ICRA.2013.6630634

Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., et al. (2010). Universal robotic gripper based on the jamming of granular material. *Proc. Natl. Acad. Sci. U.S.A.* 107, 18809–18814. doi: 10.1073/pnas.1003250107

Calandra, R., Peters, J., Rasmussen, C. E., and Deisenroth, M. P. (2016). "Manifold Gaussian processes for regression," in *2016 International Joint Conference on Neural Networks (IJCNN)* (Vancouver, BC), 3338–3345. doi: 10.1109/IJCNN.2016.7727626

Chen, F., Dirven, S., Xu, W., and Li, X. (2013). Soft actuator mimicking human esophageal peristalsis for a swallowing robot. *IEEE/ASME Trans. Mechatron.* 19, 1300–1308. doi: 10.1109/TMECH.2013.2280119

Coevoet, E., Morales-Bieze, T., Largilliere, F., Zhang, Z., Thieffry, M., Sanz-Lopez, M., et al. (2017). Software toolkit for modeling, simulation and control of soft robots. *Adv. Robot.* 31, 1208–1224. doi: 10.1080/01691864.2017.1395362

Coulson, R. (2020). *Soft materials architectures for robot manipulation* (Master's thesis). Carnegie Mellon University, Pittsburgh, PA, United States.

Deimel, R., and Brock, O. (2013). "A compliant hand based on a novel pneumatic actuator," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 2047–2053. doi: 10.1109/ICRA.2013.6630851

Della Santina, C., Arapi, V., Averta, G., Damiani, F., Fiore, G., Settimi, A., et al. (2019). Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands. *IEEE Robot. Autom. Lett.* 4, 1533–1540. doi: 10.1109/LRA.2019.2896485

Della Santina, C., Piazza, C., Grioli, G., Catalano, M. G., and Bicchi, A. (2018). Toward dexterous manipulation with augmented adaptive synergies: the PISA/IIT softhand 2. *IEEE Trans. Robot.* 34, 1141–1156. doi: 10.1109/TRO.2018.2830407

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). "Monte Carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)* (Detroit, MI), *Vol. 2*, 1322–1328. doi: 10.1109/ROBOT.1999.772544

Dollar, A. M., and Howe, R. D. (2009). "The SDM hand: a highly adaptive compliant grasper for unstructured environments," in *Experimental Robotics*, eds O. Khatib, V. Kumar, and G. J. Pappas (Berlin; Heidelberg: Springer), 3–11. doi: 10.1007/978-3-642-00196-3_2

Duriez, C. (2013). "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 3982–3987. doi: 10.1109/ICRA.2013.6631138

Engel, Y., Szabó, P., and Volkinshtein, D. (2005). "Learning to control an octopus arm with Gaussian process temporal difference methods," in *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, *Vol. 18* (Vancouver, BC).

Galloway, K. C., Becker, K. P., Phillips, B., Kirby, J., Licht, S., Tchernov, D., et al. (2016). Soft robotic grippers for biological sampling on deep reefs. *Soft Robot.* 3, 23–33. doi: 10.1089/soro.2015.0019

George Thuruthel, T., Ansari, Y., Falotico, E., and Laschi, C. (2018). Control strategies for soft robotic manipulators: a survey. *Soft Robot.* 5, 149–163. doi: 10.1089/soro.2017.0007

George Thuruthel, T., Falotico, E., Cianchetti, M., and Laschi, C. (2016). "Learning global inverse kinematics solutions for a continuum robot," in *Symposium on Robot Design, Dynamics and Control* (Udine: Springer), 47–54. doi: 10.1007/978-3-319-33714-2_6

Giorelli, M., Renda, F., Calisti, M., Arienti, A., Ferri, G., and Laschi, C. (2015). Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Trans. Robot.* 31, 823–834. doi: 10.1109/TRO.2015.2428511

Gupta, A., Eppner, C., Levine, S., and Abbeel, P. (2016). "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 3786–3793. doi: 10.1109/IROS.2016.7759557

Homberg, B. S., Katzschmann, R. K., Dogar, M. R., and Rus, D. (2019). Robust proprioceptive grasping with a soft robot hand. *Auton. Robots* 43, 681–696. doi: 10.1007/s10514-018-9754-1

Jiang, H., Wang, Z., Liu, X., Chen, X., Jin, Y., You, X., et al. (2017). "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in *2017 IEEE International Conference*

on Robotics and Automation (ICRA) (Marina Bay Sands), 6127–6133. doi: 10.1109/ICRA.2017.7989727

Jolliffe, I. T., and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 374:20150202. doi: 10.1098/rsta.2015.0202

Khansari-Zadeh, S. M., and Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Trans. Robot.* 27, 943–957. doi: 10.1109/TRO.2011.2159412

King, J. P., Bauer, D., Schlagenhauf, C., Chang, K.-H., Moro, D., Pollard, N., et al. (2018). "Design. fabrication, and evaluation of tendon-driven multi-fingered foam hands," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (Beijing), 1–9. doi: 10.1109/HUMANOIDS.2018.8624997

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Mach. Learn. Res.* 6, 1783–1816. doi: 10.1007/978-3-540-78155-4_12

Li, M., Zhuo, Y., Chen, J., He, B., Xu, G., Xie, J., et al. (2020). Design and performance characterization of a soft robot hand with fingertip haptic feedback for teleoperation. *Adv. Robot.* 34, 1491–1505. doi: 10.1080/01691864.2020.1822913

Majidi, C. (2014). Soft robotics: a perspective-current trends and prospects for the future. *Soft Robot.* 1, 5–11. doi: 10.1089/soro.2013.0001

Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D. (2014). "Design and control of a soft and continuously deformable 2d robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hongkong), 2189–2196. doi: 10.1109/ICRA.2014.6907161

Marchese, A. D., and Rus, D. (2016). Design, kinematics, and control of a soft spatial fluidic elastomer manipulator. *Int. J. Robot. Res.* 35, 840–869. doi: 10.1177/0278364915587925

Mitchell, D. P. (1996). "Consequences of stratified sampling in graphics," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New Orleans, LA: 96 SIGGRAPH), 277–280. doi: 10.1145/237170.237265

Nguyen-Tuong, D., Peters, J., and Seeger, M. (2008). Local Gaussian process regression for real time online model learning. *Adv. Neural Inform. Process. Syst.* 21, 1193–1200.

Odhner, L. U., and Dollar, A. M. (2011). "Dexterous manipulation with underactuated elastic hands," in *2011 IEEE International Conference on Robotics and Automation* (Shanghai), 5254–5260. doi: 10.1109/ICRA.2011.5980263

Ranganath, A., Xu, P., Karamouzas, I., and Zordan, V. (2019). "Low dimensional motor skill learning using coactivation," in *Motion, Interaction and Games*, eds H. P. H. Shum, E. S. L. Ho, M. P. Cani, T. Popa, D. Holden, and H. Wang (Newcastle upon Tyne: Association for Computing Machinery), 1–10. doi: 10.1145/3359566.3360071

Reynolds, D. A. (2009). Gaussian mixture models. *Encyclop. Biometr.* 741, 659–663. doi: 10.1007/978-0-387-73003-5_196

Rolf, M., and Steil, J. J. (2013). Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 1147–1160. doi: 10.1109/TNNLS.2013.2287890

Safonova, A., Hodgins, J. K., and Pollard, N. S. (2004). Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.* 23, 514–521. doi: 10.1145/1015706.1015754

Saunders, F., Trimmer, B. A., and Rife, J. (2010). Modeling locomotion of a soft-bodied arthropod using inverse dynamics. *Bioinspir. Biomimet.* 6:016001. doi: 10.1088/1748-3182/6/1/016001

Schlagenhauf, C., Bauer, D., Chang, K.-H., King, J. P., Moro, D., Coros, S., et al. (2018). "Control of tendon-driven soft foam robot hands," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (Beijing), 1–7. IEEE. doi: 10.1109/HUMANOIDS.2018.8624937

Stuart, H., Wang, S., Khatib, O., and Cutkosky, M. R. (2017). The ocean one hands: an adaptive design for robust marine manipulation. *Int. J. Robot. Res.* 36, 150–166. doi: 10.1177/0278364917694723

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction, 2nd Edn*. The MIT Press.

Tang, Z. Q., Heung, H. L., Tong, K. Y., and Li, Z. (2019). Model-based online learning and adaptive control for a "human-wearable soft robot" integrated system. *Int. J. Robot. Res.* Cambridge, MA: A Bradford Book, 1. doi: 10.1177/0278364919873379

Wang, T.-C., and Guu, C.-S. (1989). *Machine Vision Seam Tracking Method and Apparatus for Welding Robots*. US Patent 4,812,614.

Xu, Z., and Todorov, E. (2016). "Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockhom), 3485–3492.

Zhang, J., Wang, B., Zhang, C., Xiao, Y., and Wang, M. Y. (2019). An EEG/EMG/EOG-based multimodal human-machine interface to real-time control of a soft robot hand. *Front. Neurorobot.* 13:7. doi: 10.3389/fnbot.2019.00007

Zhou, J., Chen, X., Chang, U., Lu, J.-T., Leung, C. C. Y., Chen, Y., et al. (2019). A soft-robotic approach to anthropomorphic robotic hand dexterity. *IEEE Access* 7, 101483–101495. doi: 10.1109/ACCESS.2019.2929690

Zhou, J., Yi, J., Chen, X., Liu, Z., and Wang, Z. (2018). BCL-13: a 13-DOF soft robotic hand for dexterous grasping and in-hand manipulation. *IEEE Robot. Autom. Lett.* 3, 3379–3386. doi: 10.1109/LRA.2018.2851360

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.