A Sequential Group VAE for Robot Learning of Haptic Representations

Benjamin A. Richardson
Haptic Intelligence Department
Max Planck Institute for Intelligent Systems
Heisenbergstr. 3, 70569 Stuttgart, Germany
richardson@is.mpg.de

Katherine J. Kuchenbecker
Haptic Intelligence Department
Max Planck Institute for Intelligent Systems
Heisenbergstr. 3, 70569 Stuttgart, Germany
kjk@is.mpg.de

Georg Martius
Autonomous Learning Group
Max Planck Institute for Intelligent Systems
Max-Planck-Ring 4, 72076 Tübingen, Germany
georg.martius@tuebingen.mpg.de

Abstract: Haptic representation learning is a difficult task in robotics because information can be gathered only by actively exploring the environment over time, and because different actions elicit different object properties. We propose a Sequential Group VAE that leverages object persistence to learn and update latent general representations of multimodal haptic data. As a robot performs sequences of exploratory procedures on an object, the model accumulates data and learns to distinguish between general object properties, such as size and mass, and trial-to-trial variations, such as initial object position. We demonstrate that after very few observations, the general latent representations are sufficiently refined to accurately encode many haptic object properties.

Keywords: Haptic sensing, robotic exploratory procedures, representation learning

1 Introduction

When people physically interact with unknown items, they very quickly form a perceptual representation of each object [1]. This representation is typically developed by integrating prior knowledge with new information that is gathered by performing exploratory procedures (EPs) [2, 3]. Each characteristic EP elicits information about certain object properties that are both implicit and explicit, but no one EP alone can provide a complete picture. For example, by pressing into an object, we can determine its stiffness but not necessarily its mass, shape, or size. By enclosing an object we can learn its shape and size, whereas shaking an object provides information about its dynamic properties and whether there are loose fillings inside. Thus, we might think that two objects are very similar after one EP, but we can quickly disambiguate them by accumulating information from additional exploration. To operate in and act upon the real world, autonomous robots will need to have a similar ability to interact physically with objects in their environment, accumulate information across sequential exploratory procedures, and form haptic representations that are useful for real-world tasks.

One challenging aspect of this goal is how to accumulate information over a sequence of interactions with an object. Information accumulation has been investigated in haptic exploration for a variety of tasks including surface classification [4, 5], haptic property identification [6], and contour following [7]. While high performance was obtained in each of these tasks, they rely on task-specific supervised learning instead of learning general representations applicable to many tasks.

Besides simply perceiving object properties, learning robots need to acquire and update useful representations of physical interactions with the world. It is unlikely that a robot can be given a priori

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

all the knowledge or broad representative features it will need throughout its existence. Thus, it should have the ability to learn descriptive factors for a wide range of physical tasks. Unsupervised learning has been demonstrated to discover expressive representations of tactile and haptic data that can perform well across a variety of tasks [8, 9, 10, 11, 12]. However, these representations are learned on huge amounts of data and do not include explicit mechanisms for being updated.

We consider the task of learning comprehensive latent representations of haptic properties from sequences of interactions with objects. Performing a variety of EPs on objects generates multiple observations. No single EP can elicit information about every object property, and thus several observations must be accumulated to build a comprehensive latent representation. We propose a Sequential Group Variational Autoencoder (VAE) based on the Multi-Level VAE [13], which iteratively updates probabilistic latent representations of objects as new information is acquired during sequential EPs. Our method learns generic representations that can be used to infer properties and that contain uncertainty about the inference when the representation is learned on insufficient EPs. After exploring our method on a synthetic MNIST variant with multiple sequential crops, we apply it to and evaluate it on real data from a robot arm that uses four exploratory procedures to interact with 74 objects that vary across multiple haptic property dimensions.

Our work contributes a novel, recursive Sequential Group-VAE architecture that (i) accumulates information from sequential EPs to iteratively update and learn generic representations of haptic properties and objects, (ii) uses those learned representations to infer observations from unseen EPs, and (iii) contains uncertainty when the EPs are insufficient to elicit information about certain haptic properties.

2 Related Work

Haptic Representation Learning Machine-learning techniques have become more popular in haptics in recent years, being used to classify objects/surfaces [8, 4, 14, 15, 10] and semantic properties [6, 16, 9, 11] from both tactile and proprioceptive data. A common approach has been to define features informed by the sensing capabilities of human mechanoreceptors or by human haptic perception (e.g. vibration spectral centroid is correlated with hardness perception [17]), extract those features from raw haptic data captured by a tool or robot, and classify a set of objects from those features [4, 15]. Others used these types of predefined features to learn semantic attributes of objects that can be applied to new, unseen examples [18, 6, 16, 19]. One additional approach has been to use unsupervised learning or compression techniques to develop latent representations of haptic interactions [20, 21, 9, 8, 12, 10, 22, 11], which typically outperform hand-crafted features when used to learn downstream tasks. Bag-of-words models [20] and dictionary learning [21, 22, 11] have demonstrated good generalization across many haptic property identification tasks. In particular, Tatiya et al. [12] adopt a β-VAE to learn latent representations of objects by encoding data from one action and decoding it to predict data from a different action.

Haptic Information Accumulation Information accumulation can occur on multiple timescales during haptic interactions. During single EPs, data can be processed with recurrent models (e.g. LSTMs) that learn to estimate and predict instantaneous state, learning representations of very short moments in time. These methods have demonstrated great effectiveness for a variety of tasks, including hardness detection [23, 24] and clothing material perception [25]. Alternatively, features can be learned on small segments and concatenated to form large feature vectors that represent full EPs [26, 21, 22]. However, this is a different task from accumulating information over multiple exploratory procedures, where relevant information is processed at a different timescale. One method of capturing information across multiple EPs is simply to learn a separate representation of the data captured from each EP and then concatenate those representations [27], but this is limited if any downstream model has a fixed number of inputs. A more flexible approach is to update a representation as more information is gathered. Fishel and Loeb [4] use Bayesian inference to improve texture classification over sequences of parameterized sliding EPs. Gemici and Saxena [6] also use Bayesian inference, but they update beliefs over sets of haptic properties as they perform more EPs. To perform active contour and shape-feature following, Lepora et al. [7] use recursive Bayesian inference to modify the control of a tactile sensor tip while it traces shape features like edges and corners. Each of these cases uses supervised learning to adjust relatively simple models to perform specific tasks. This rigid structure limits their ability to learn representations that generalize to new tasks. On the other hand, Dallaire et al. [5] perform unsupervised clustering of surfaces using Pitman-Yor process mixture models. This approach assumes that observations come from a discrete set of underlying distributions and is powerful for data that is sampled from multinomial or categorical distributions.

Learning Group Representations As a robot explores an object over time, it should not assume that the individual observations are independent. If we assume that a robot understands object permanence (that objects continue to exist even when not immediately perceived) [28], at least during the course of a sequence of interactions, we can leverage that knowledge to build grouped representations. Bouchacourt et al. [13] propose the Multi-Level Variational Autoencoder (ML-VAE) for learning disentangled representations. The ML-VAE splits the latent representation space into two components and forces all samples from a single group to share a single latent vector in one of those components. This approach can learn general representations of classes on multiple datasets. Sato et al. [29] use ML-VAE to perform few-shot anomaly detection of images, grouping samples by domain instead of class.

3 Methods

Rather than assuming that all data from a sequence of interactions is processed together, our method handles shorter observations from discrete EPs to build up object representations over time, with high flexibility that matches the diverse ways in which robots can interact with objects in the world. Specifically, given observations from a sequence of EPs of an object, our method uses a β-VAE to model a probability distribution of the latent variable representation of the observations; this distribution is updated iteratively as each observation in the sequence is processed. Each progressive representation is conditioned on the most recent observation and the previous latent representation, which we call the context. Finally, at each update step, random samples are drawn from the latent distribution and fed into a shared decoder network to try to reconstruct each observation in the sequence. We will first introduce the VAE and then a context-aware VAE method called Multi-Level VAE [13]. Finally, we will describe our method.

3.1 Background

Variational Autoencoder (VAE) In the standard VAE framework [30], we assume a dataset $\mathbf{X} = \{x_1, ..., x_n\}$ composed of independent and identically distributed (i.i.d.) observations generated by a stochastic process from an underlying random variable z, with $z \sim p_\theta(z)$ and $x_i \sim p_\theta(x \mid z)$. The goal is to learn a variational approximation $q_\phi(z \mid x)$ of the true (but typically intractable) posterior over the latent variable $p_\theta(z \mid x)$. Typically, both the distributions $q_\phi(z \mid x)$ and $p_\theta(x \mid z)$ are modeled by encoder and decoder neural networks parameterizing a Gaussian (or normal) distribution, although other choices are possible. Parameters ϕ and θ are learned simultaneously by minimizing the evidence lower bound $\mathcal{L}$ of the marginal log-likelihood of the data, $\log p_\theta(\mathbf{X})$ (more on this in Section 3.2).

Multi-Level Variational Autoencoder (ML-VAE) The ML-VAE [13] does not assume i.i.d. observations, but instead that there are disjoint subsets of observations that come from distinct groups $g \in \mathcal{G}$. Groups are independent from each other, but samples within a single group are not independent. Observations are assumed to be generated from two sets of latent variables: the content C and the style S. Each observation $x_i \in \mathbf{X}_g$ from a group $g \in \mathcal{G}$ is generated from the same latent content variable $\mathbf{c}_g$ and a unique style variable s_i. That is, the likelihood is given by $p_\theta(x_i \mid \mathbf{c}_g, s_i) | \forall x_i \in \mathbf{X}_g$. Because the content and style are assumed to be independent, the variational approximation for a sample $q_\phi(c_i, s_i \mid x_i)$ can be decomposed into the product of $q_{\phi_c}(c_i \mid x_i)$ and $q_{\phi_s}(s_i \mid x_i)$. In this case, q_{ϕ_c} and q_{ϕ_s} are chosen to be normal. A group content variable is approximated by multiplying together all the approximate individual content variables $q_{\phi_c}(\mathbf{c}_g \mid \mathbf{X}_g) \propto \prod_{i \in g} q_{\phi_c}(c_i \mid x_i)$.

3.2 Iterative Latent Update via Group VAE

For our method, we assume a group setting where each independent group is an object $o \in \mathcal{O}$. Let the set of observations from object o be denoted $\mathbf{X}_o$. We observe sequences $\mathbf{x}_o \subseteq \mathbf{X}_o$ of observations, where $\mathbf{x}_o = \{x^1_{a^1}, ..., x^t_{a^t}, ..., x^n_{a^n}\}$, $a^t \in A = \{a_1, ..., a_n\}$ indicates which of the n actions was performed to generate that observation, and t indicates the order in the sequence. Like in ML-VAE, we assume that the observations are generated from two independent underlying latent random variables, the content C and style S, in our case also conditioned on the action. Since our robot will have access to only a particular sequence of actions on the same object, and object identities do not have to be known to the robot between trials, we assume that content is shared across a single sequence, such that every element $x^t_{a^t}$ of the sequence $\mathbf{x}_o$ shares the same latent content variable $\mathbf{c}_o$[1]. The style formulation remains unchanged, where each element x^t has its own style variable s^t, and $\mathbf{s}_o = \{s^t \forall x^t \in \mathbf{x}_o\}$. Thus, within a sequence $\mathbf{x}_o$, individual observations $x^t_{a_i}$ are assumed to be generated from the latent variables according to $x^t_{a_i} \sim p_\theta(x \,|\, \mathbf{c}_o, s^t, a_i)$. Again like the ML-VAE, the variational approximation of a sequence $q_\phi(\mathbf{c}_o, \mathbf{s}_o \,|\, \mathbf{x}_o)$ decomposes into the product of $q_{\phi_c}(\mathbf{c}_o \,|\, \mathbf{x}_o)$ and $q_{\phi_s}(\mathbf{s}_o \,|\, \mathbf{x}_o)$. We also assume these distributions are normal, with

$$q_{\phi_c}(\mathbf{c}_o \,|\, \mathbf{x}_o) = \mathcal{N}(\mathbf{c}_o \,|\, \mu(\mathbf{x}_o, \phi_c), \Sigma(\mathbf{x}_o, \phi_c)), \tag{1}$$

$$q_{\phi_s}(\mathbf{s}_o \,|\, \mathbf{x}_o) = \mathcal{N}(\mathbf{s}_o \,|\, \mu(\mathbf{x}_o, \phi_s), \Sigma(\mathbf{x}_o, \phi_s)) \tag{2}$$

A key feature of our approach is that the variational approximation is updated iteratively, and at each step the approximation is conditioned on the parameters of the previous variational content approximation, which we call the context. Specifically, given an observation $x^t \in \mathbf{x}_o$ at iteration t, the variational approximation is given by

$$q^{(t)}_\phi\left(\mathbf{c}^{(t)}_o, s^t \,|\, x^t, q^{(t-1)}_{\phi_c}\right), \text{ where } q^{(0)} = \mathcal{N}(\mathbf{0}, I) \tag{3}$$

To encourage the content latent variable to capture general descriptions of the object from which a sequence is sampled, our method performs inference for every observation of the sequence at each update step. At each iteration t for sequence $\mathbf{x}_o$, the marginal log-likelihood (or evidence) can be written as the sum of the evidence lower bound (ELBO), denoted $\mathcal{L}$, and the Kullback-Leibler divergence between the true posterior and the variational approximation:

$$\begin{aligned}\log p_\theta(\mathbf{x}_o; t) = &\,\mathrm{KL}\left(q_\phi\left(\mathbf{c}^{(t)}_o, \mathbf{s}^{(t)}_o \,|\, x^t, q^{(t-1)}_{\phi_c}\right) \,||\, p_\theta\left(\mathbf{c}^{(t)}_o, \mathbf{s}^{(t)}_o \,|\, x^t, q^{(t-1)}_{\phi_c}\right)\right) \\ &+ \mathcal{L}^{(t)}(\mathbf{x}_o; \theta, \phi_c, \phi_s)\end{aligned} \tag{4}$$

$$\text{where } \mathbf{s}^{(t)}_o = \{s^u; x^u \in \mathbf{x}_o\} \text{ and } q_{\phi_s}\left(\mathbf{s}^{(t)}_o \,|\, \cdot\right) = \prod q_{\phi_s}(s^u \,|\, \cdot),$$

$$\text{with } q_{\phi_s}\left(s^u \,|\, x^u, q^{(u-1)}_{\phi_c}\right) = \begin{cases} q_{\phi_s}\left(s^u \,|\, x^u, q^{(u-1)}_{\phi_c}\right) & \text{if } u \leq t \\ \mathcal{N}(s^u \,|\, \mathbf{0}, I) & \text{if } u > t \end{cases}.$$

It should be noted that before an observation has been seen by the network, its style is sampled from a standard normal distribution. Because the KL divergence is always non-negative, the ELBO is a lower bound on the marginal log-likelihood. The ELBO for sequence $\mathbf{x}_o$ at step t can itself be written as the negative of the sum of the negative log-likelihood ($\mathcal{L}_{\text{NLL}}$) and the KL divergences between the variational approximations and their corresponding priors:

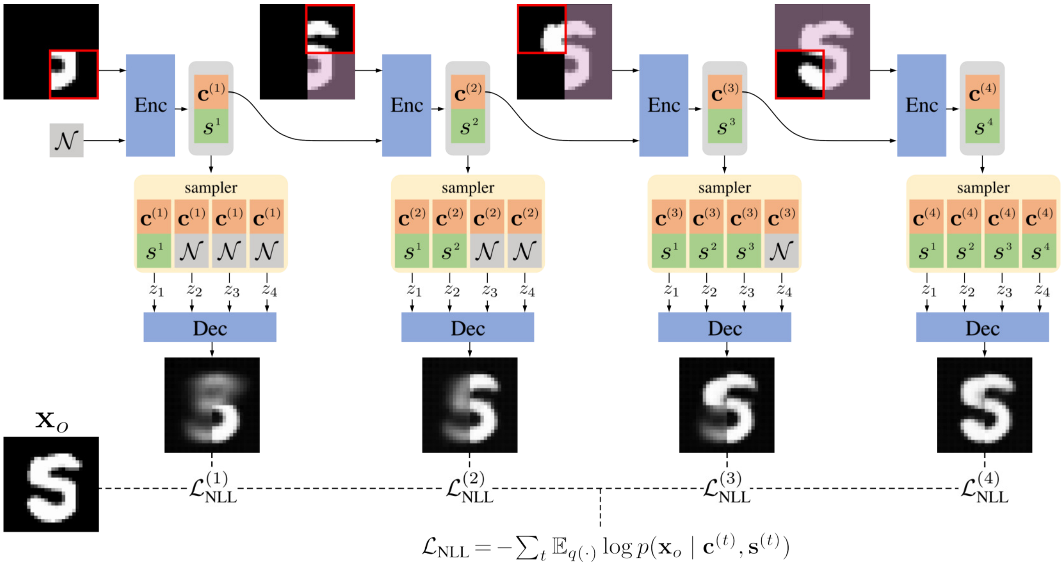
$$\begin{aligned}\mathcal{L}^{(t)}(\mathbf{x}_o; \theta, \phi_c, \phi_s) = &\,\mathbb{E}_{q^{(t)}_{\phi_c}\left(\mathbf{c}^{(t)}_o | x^t, q^{(t-1)}_{\phi_c}\right)} \mathbb{E}_{q^{(t)}_{\phi_s}\left(\mathbf{s}^{(t)}_o | x^t, q^{(t-1)}_{\phi_c}\right)} \log p_\theta\left(\mathbf{x}_o \,|\, \mathbf{c}^{(t)}_o, \mathbf{s}^{(t)}_o, \mathbf{a}_o\right) \\ &- \mathrm{KL}\left(q^{(t)}_{\phi_c}(\mathbf{c}^{(t)}_o \,|\, x^t, q^{(t-1)}_{\phi_c}) \,||\, p_\theta(\mathbf{c}_o)\right) - \mathrm{KL}\left(q^{(t)}_{\phi_s}(\mathbf{s}^{(t)}_o \,|\, x^t, q^{(t-1)}_{\phi_c}) \,||\, p_\theta(\mathbf{s}_o)\right),\end{aligned} \tag{5}$$

where $\mathbf{a}_o$ is the vector of actions that corresponds to the observations in $\mathbf{x}_o$. In practice, we use the β-NLL formulation [31] of the negative log-likelihood. This ELBO loss $\mathcal{L}^{(t)}$ can be summed over all iterations. A detailed algorithm of our modeling procedure can be found in Section S1 of the supplementary materials.

3.3 Demonstration on MNIST

To illustrate our method on a simple example, we first apply it to the MNIST dataset [32]. Here, the set of objects $\mathcal{O}$ is comprised of the ten digits $d = \{0, ..., 9\}$. We define four actions, each of which crops a

[1] A sequence could potentially include every observation from a single object (e.g. over a lifetime of observations), in which case $\mathbf{c}_o$ would be shared across the full object group.



Enc
$\mathbf{c}^{(1)}$
s^1
$\mathcal{N}$
sampler
z_1
z_2
z_3
z_4
Dec
$\mathbf{x}_o$
$\mathcal{L}_{\text{NLL}}^{(1)}$
$\mathcal{L}_{\text{NLL}}^{(2)}$
$\mathcal{L}_{\text{NLL}}^{(3)}$
$\mathcal{L}_{\text{NLL}}^{(4)}$
$\mathcal{L}_{\text{NLL}} = -\sum_t \mathbb{E}_{q(\cdot)} \log p(\mathbf{x}_o \mid \mathbf{c}^{(t)}, \mathbf{s}^{(t)})$

Figure 1: Model architecture on an example MNIST digit. At each step t a single crop x^t is fed into the encoder, which outputs probabilistic content and style encodings $\mathbf{c}^{(t)}$ and s^t. The content encoding $q\big(\mathbf{c}^{(t)} \,|\, x^t, \mathbf{c}^{(t-1)}\big)$ is passed as context into the encoder for the next update. The decoding step attempts to reconstruct all four crops of the MNIST digit. For each crop, the content latent representation $q\big(\mathbf{c}^{(t)} \,|\, x^t, \mathbf{c}^{(t-1)}\big)$ is sampled. If a crop x^u has already been seen, then the style latent representation $q\big(s^u \,|\, x^u, \mathbf{c}^{(t-1)}\big)$ is sampled. If a crop hasn't been seen, then $\mathcal{N} = \mathcal{N}(0,1)$ is sampled. The samples from the content and style representations for each action are then fed into the shared decoder. As more crops are observed, the representation improves and becomes more certain.

different quadrant (top right, . . . , bottom left) of a traditional MNIST image sample; like one haptic EP, one crop usually does not fully identify the object. For model training, we select a digit d and perform one of each cropping action in random order, generating a sequence four crops long. Then we follow the iterative training procedure described above. Fig. 1 shows an overview of the full approximation and inference procedure on an example MNIST digit. As individual crops are observed, the content representation is refined, and the style for each crop is inferred.

We demonstrate the effectiveness and behavior of our method by qualitatively evaluating the performance on this dataset in Section S3. With evidence supporting our method's behavior and performance, we now apply it to the problem of learning haptic representations from sequential EPs by a haptically sensitive robot. This cropped MNIST setting has similarities with our real-world data, as only partial information is perceived with every action. A difference is that in the synthetic example, we are interested only in digit identity, whereas in the haptic application we care about different object properties.

4 Experimental Setup

Hardware We use a six-degree-of-freedom robot arm (Universal Robots UR5) with a wrist-mounted force-torque sensor (Weiss KMS 40) and a modified three-fingered gripper (Right Hand Labs Reflex Takktile 2), as shown in Fig. 2. The UR5 is position controlled and records position, velocity, and effort at 125 Hz for each of its six joints. The KMS 40 records three-axis force and three-axis torque at 500 Hz. The modified Reflex Takktile 2 gripper has three under-actuated fingers each with a customized inertial measurement unit (IMU) for estimating the distal joint angles, and one of the fingers has 14 tactile pressure sensors. The proximal joint positions are measured by encoders. Two of the fingers are coupled and can be rotated in opposite directions (called the preshape). Finally, we record the angular position and load from all four actuating motors in the hand. All hand data is sampled at 25 Hz. All control and data collection were run via ROS in Python and C++, building on the existing libraries for the UR5, KMS 40, and ReflexTakktile 2.

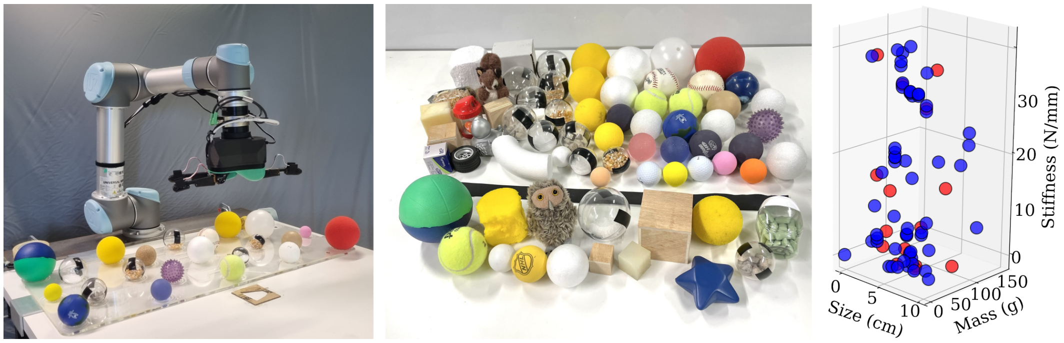


Figure 2: Robot with UR5 arm, customized Reflex Takktile 2 hand, and wrist-mounted KMS 40 six-axis force-torque sensor (left). The set of 74 objects (including empty space) used for the study with the test set below the black divider (center). The distribution of manually measured object properties: size, mass, and stiffness, with the test set in red (right).

Exploratory Procedures (EPs) The robot performs four pre-programmed EPs, which are explained in greater detail in Section S2:

1. *Drag*: The object is grasped, gently pressed into the table, and dragged across the surface at 3 cm/s.
2. *Press*: The robot positions the tips of the two connected fingers above the object and moves down at 1 cm/s until a 10 N force threshold is reached on the KMS 40 sensor.
3. *Shake*: The robot grasps and lifts the object to a fixed position. It then rapidly moves the object up and down four times.
4. *Squeeze*: The hand widens the preshape joint and closes the fingers around the object at 0.2 rad/s until the finger motor loads reach a certain threshold, after which the fingers open.

The EPs squeeze and press are standard human EPs [3], whereas shake has been used as an EP for robots [33]. Because our robot does not have highly sensitive fingertips, we decided to drag the objects across the table instead of sliding a finger across the objects. To diversify the robot's perceptual experiences, it drops the objects into a central cardboard well between successive EPs. The objects can randomly roll or shift within the well, creating some randomness in their initial positions.

Signal Processing To prepare our data for the convolutional model architecture, all signals were reduced to 25 Hz. The UR5 data were downsampled using the scipy `decimate` function. To capture both the transient, high-frequency vibrations and the signal magnitude, force and torque data were separated into AC and DC components using the scipy `spectrogram` (window size $=40$ points, overlap $=20$ points) and `decimate` functions. Each interaction in the dataset was then either cut or padded to the same duration; the standard duration of 400 points (16 seconds) was determined by subtracting the earliest moment of contact across all *presses* from the latest moment of contact across all *presses*. All *presses* and *squeezes* were cropped to these time points, and the shorter *shakes* and *slides* were padded on the end by repeating the values of the last recorded point. Thus, each trial is represented as a 195×400 feature array (see Section S2 for more details).

Objects We designed a set of 74 objects (Fig. 2) to test our approach on learning comprehensive latent representations of haptic properties from sequences of interactions. All objects needed to be graspable and liftable by our selected robot platform. To minimize the influence of object orientation on this first investigation, we decided to use primarily spherical objects along with some other less symmetric objects. Most of the spheres are purchased sports balls or are either hollow or filled shells. The more complex objects include dense wood, a plastic gum container, soft stuffed animals, and a variety of dense, hollow, or filled cubes. The 73 selected items have a range of physical properties, differing significantly in size, stiffness, mass, and filling. The final object is empty space, giving a full set of 74 things the robot can drag, press, shake, and squeeze.

5 Experiments

Our experiments were designed to evaluate the behavior and performance of our method. Specifically, how does the latent content representation change over iterations? How well are the haptic properties of size (in two dimensions), stiffness, mass, and filling encoded in the latent content space? And how is information distributed between the content and style latent spaces? We compare the representations learned by our method to those learned by a standard sequential VAE that does not split its latent representation into content and style, with additional experiments and results shown in Section S3.

5.1 Training Procedure

In total, the robot performed each EP 20 times on each object, for a total of 5 920 trials. We trained and validated our models on 80% of the objects and tested on the remaining 20% of objects (shown separately in Fig. 2). Our encoder and decoder networks are composed of four 1D convolutional layers and two dense layers with ten dimensions for both the content and style vectors. When training the models, we generate new random sequences of four EPs from the same object (as in the MNIST example) at the beginning of every epoch. To have a comparison that demonstrates the utility of the Group VAE, we also train standard VAEs with a latent space of 20 dimensions. More details about the model architecture and training procedure can be found in Section S2.

5.2 Evaluation

To determine how the content representation improves over iterations and how well the properties and objects are represented in the latent representation, we train regression models on content and style representations to predict the haptic properties of mass, press size (width), squeeze size (height), and stiffness (all normalized to be from zero to one), and we train a classification model to predict whether an object has loose filling. We generate random four-EP sequences from the original training dataset and use the content and style representations from *only the final-iteration* as our training and validation data for the models. We similarly generate random four-EP sequences from the test dataset and evaluate the performance on content and style *after each iteration* through the Sequential Group VAE. For each of the labels, we train a total of 25 models. To determine how generalization is impacted by splitting the latent space into content and style, we also train models in the same way on the standard sequential VAE models, using generated representations in the single latent space to train. Finally, to provide a performance baseline, we train models directly from the raw data. Additional architecture and training details can be found in Section S2.

6 Results

The distributions of results for the regression are shown in Fig. 3 with the means marked by black dashes. The low mean squared error (MSE) achieved on the content regressions indicate that general properties of the objects are indeed encoded in content. Conversely, the style contains less content information and performs worse (higher MSE) in most cases; this result is expected because style is designed to capture only trial-specific variations. The standard VAE performs noticeably worse for every property except mass.

We evaluated the regression models (trained on the fourth and final iteration as described above) at each iteration of the test set sequencing. The performance after the first iteration is labeled with “Test 1,” after the second with “Test 2,” and so on. To our surprise, the MSE on the test set content gently increases as a function of sequence iteration, indicating that information about different properties in the test set is not becoming more precise over sequences of EPs. The style MSE demonstrates similar behavior but with generally worse average scores. It is somewhat surprising that there is no improvement over a small number of observations, as this was found to be the case for the ML-VAE performing MNIST classification [13]. We believe this surprising finding could be caused by the limited number of objects (45) in each training set.

Results not shown here revealed that we could not classify whether an object is filled with loose items; binary classification performed no better than random on this task. We believe the system’s poor

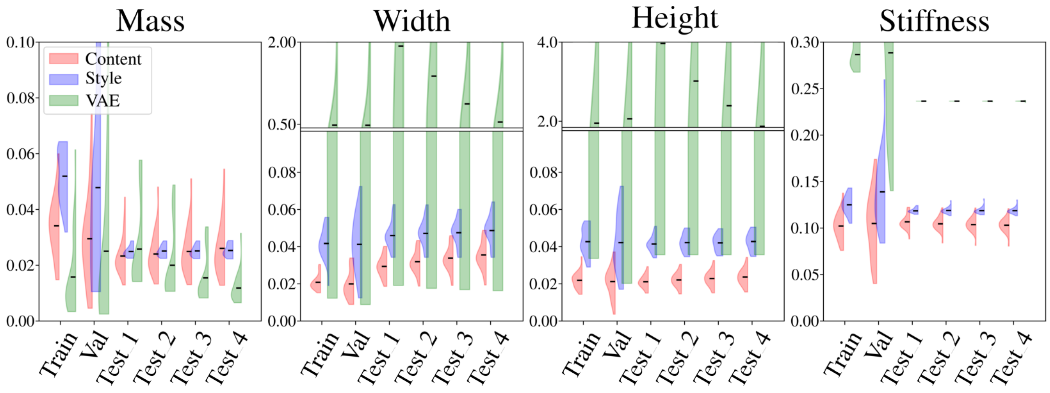


Figure 3: Distributions of regression MSE (lower is better) for individual properties using the Group VAE content (red), Group VAE style (blue), and standard VAE (green) representations. As expected, object properties are more strongly represented (somewhat smaller errors) in the content latent space compared to the style latent space. In most cases the standard VAE does much worse (particularly for width and height, where the axes are re-scaled in the upper portion).

performance on filling classification is primarily caused by lack of high frequency sensing in the robot hand and a large diversity of different fillings across many differently shaped objects.

7 Conclusion

We propose a Sequential Group VAE for learning and updating latent representations of haptic data as a robot sequentially explores and physically interacts with objects. After validating our method on a modified MNIST dataset, we use a robot arm interacting with 74 objects objects to demonstrate that our method accumulates general group information as it iterates through sequences of exploratory procedures (EPs).

We investigate the learned representations and find that they are predictive of relevant object properties, including size, stiffness, and mass. However, we do not clearly observe the expected behavior of significantly improving performance over iterations; furthermore, we expected the performance gap between content and style regression to be larger. Despite these mixed results, we believe our method can become a valuable addition to unsupervised robot learning because it finds representations that are important for future downstream tasks from sequential multimodal haptic sensations. To allow others to build on this approach, we plan to make our data and code publicly available.

8 Limitations

Though promising, the presented work has some limitations. The set of 74 objects is small but differs along many physical and semantic properties. Additionally, the robot's set of four EPs should be expanded by randomly parameterizing the EPs or by allowing the robot to optimize across parameterizations of the EPs; although the learned models are likely not robust to any changes in EP parameterization, the method can easily be adjusted to accommodate more complex actions and would likely be more general. Finally, the robot's sensing capabilities were limited because only one of its fingers includes functional tactile sensors. Furthermore, none of the robot's current fingertip sensors can capture the rich high-frequency vibrations that are likely to be important for perceiving object texture during *drag* actions or loose filling during *shake*.

Although our method could be extended to longer sequences with repeated EPs, we trained only on sequences of length four with one of each EP. We did not test how training repeatedly on the same action influences the learned model; we expect repetition should reduce uncertainty in relevant object properties. We also did not evaluate how each EP influences the update of the latent content representation. These directions should be explored in future work to build on and test the limits of the initial results reported here.

The current formulation of our experiments makes it difficult to assess what information is being learned by the content and the style. Future experiments could systematically vary object position and orientation to see if this trial-dependent information is captured more strongly by style than content.

Acknowledgments

This work was funded by the German Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ 01IS18039B). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Benjamin Richardson.

References

[1] R. L. Klatzky, S. J. Lederman, and V. A. Metzger. Identifying objects by touch: An "expert system". *Perception & Psychophysics*, 37(4):299–302, 1985.

[2] J. J. Gibson. Observations on active touch. *Psychological Review*, 69(6):477, 1962.

[3] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, 1987.

[4] J. Fishel and G. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics*, 6, 2012.

[5] P. Dallaire, P. Giguère, D. Émond, and B. Chaib-Draa. Autonomous tactile perception: A combined improved sensing and bayesian nonparametric approach. *Robotics and Autonomous Systems*, 62 (4):422–435, 2014.

[6] M. C. Gemici and A. Saxena. Learning haptic representation for manipulating deformable food objects. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 638–645, 2014.

[7] N. F. Lepora, K. Aquilina, and L. Cramphorn. Exploratory tactile servoing with active touch. *IEEE Robotics and Automation Letters*, 2(2):1156–1163, 2017.

[8] S. E. Navarro, N. Gorges, H. Wörn, J. Schill, T. Asfour, and R. Dillmann. Haptic object recognition for multi-fingered robot hands. In *Proceedings of the IEEE Haptics Symposium*, pages 497–502, 2012.

[9] H. Liu, F. Sun, D. Guo, B. Fang, and Z. Peng. Structured output-associated dictionary learning for haptic understanding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7): 1564–1574, 2017.

[10] M. Kerzel, E. Strahl, C. Gaede, E. Gasanov, and S. Wermter. Neuro-robotic haptic object classification by active exploration on a novel dataset. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[11] B. A. Richardson and K. J. Kuchenbecker. Learning to predict perceptual distributions of haptic adjectives. *Frontiers in Neurorobotics*, 13:116, 2020.

[12] G. Tatiya, R. Hosseini, M. C. Hughes, and J. Sinapov. A framework for sensorimotor cross-perception and cross-behavior knowledge transfer for object categorization. *Frontiers in Robotics and AI*, 7, 2020. doi:10.3389/frobt.2020.522141.

[13] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[14] A. J. Spiers, M. V. Liarokapis, B. Calli, and A. M. Dollar. Single-grasp object classification and feature extraction with simple robot hands and tactile sensors. *IEEE Transactions on Haptics*, 9 (2):207–220, 2016.

[15] M. Strese, C. Schuwerk, A. Iepure, and E. Steinbach. Multimodal feature-based surface material classification. *IEEE Transactions on Haptics*, 10(2):226–239, 2016.

[16] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, and K. J. Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems*, 63:279–292, 2015.

[17] R. H. LaMotte. Softness discrimination with a tool. *Journal of Neurophysiology*, 83(4):1777–1786, 2000.

[18] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, et al. Using robotic exploratory procedures to learn the meaning of haptic adjectives. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3048–3055, 2013.

[19] Z. Abderrahmane, G. Ganesh, A. Crosnier, and A. Cherubini. Haptic zero-shot learning: Recognition of objects never touched before. *Robotics and Autonomous Systems*, 105:11–25, 2018.

[20] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 243–248, 2009.

[21] M. Madry, L. Bo, D. Kragic, and D. Fox. ST-HMP: Unsupervised spatio-temporal feature learning for tactile data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, 2014. doi:10.1109/ICRA.2014.6907172.

[22] B. A. Richardson and K. J. Kuchenbecker. Improving haptic adjective recognition with unsupervised feature learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3804–3810, 2019.

[23] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson. Shape-independent hardness estimation using deep learning and a gelsight tactile sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 951–958, 2017. doi:10.1109/ICRA.2017.7989116.

[24] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp. Inferring object properties with a tactile-sensing array given varying joint stiffness and velocity. *International Journal of Humanoid Robotics*, 15 (01):1750024, 2018.

[25] W. Yuan, Y. Mo, S. Wang, and E. H. Adelson. Active clothing material perception using tactile sensing and deep learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849, 2018. doi:10.1109/ICRA.2018.8461164.

[26] L. Cao, R. Kotagiri, F. Sun, H. Li, W. Huang, and Z. M. M. Aye. Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[27] H. Soh and Y. Demiris. Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition. *IEEE Transactions on Haptics*, 7(4):512–525, 2014. doi:10.1109/TOH.2014.2326159.

[28] J. Piaget, H. Gruber, and J. Vonèche. The essential Piaget. *Educational Researcher*, 8, 12 1977.

[29] K. Sato, S. Nakata, T. Matsubara, and K. Uehara. Few-shot anomaly detection using deep generative models for grouped data. *IEICE Transactions on Information and Systems*, 105:436–440, 2022.

[30] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[31] M. Seitzer, A. Tavakoli, D. Antic, and G. Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2022.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] J. Sinapov, M. Wiemer, and A. Stoytchev. Interactive learning of the acoustic properties of household objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2524, 2009.

Supplemental Materials: A Sequential Group VAE for Robot Learning of Haptic Representations

S1 Method Algorithm

Algorithm 1: Sequential Group VAE training algorithm.

```
Given $\mathbf{X} = \{(x^i, a^i, o^i)\}$ with $a^i \in A = \{a_1, ..., a_m\}$ and $o^i \in \mathcal{O}$
for each epoch do
    // Create random sequences for training
    $\mathbf{x}_s \leftarrow \{\}$
    for $(x, a, o) \in \mathbf{X}$ do
        $\mathbf{a}_o = \mathrm{permutation}(A \setminus \{a\})$
        $\mathbf{x}_o = \{x$, random set of inputs from object $o$ with actions $\mathbf{a}_o\}$
        $\mathbf{X}_s \leftarrow \{\mathbf{X}_s \cup \mathbf{x}_o\}$
    end
    // Train model
    while All sequences not seen do
        $\mathbf{X}_{s,b} \leftarrow$ Sample batch of sequences $\mathbf{X}_s$
        for $\mathbf{x} \in \mathbf{X}_{s,b}$ do
            $q(c) \leftarrow \mathcal{N}(0,1)$
            $q(\mathbf{s}) \leftarrow \mathcal{N}(0,1)^{|\mathbf{x}|}$
            for $t = 1 ... |\mathbf{x}|$ do
                $x^t \leftarrow \mathbf{x}[t]$
                $q(c) \leftarrow q_{\phi_c}\left(\mathbf{c}^{(t)} \mid x^t, q_{\phi_c}^{(t-1)}\right)$          // Encoding from Eq. 3
                $q(\mathbf{s})[t] \leftarrow q_{\phi_s}\left(s^t \mid x^t, q_{\phi_c}^{(t-1)}\right)$          // Update style for action $t$
                // Reconstruct for all actions
                for $u = 1 ... |\mathbf{x}|$ do
                    Sample $c^u \sim q(c)$
                    Sample $s^u \sim q(\mathbf{s})[u]$
                    $p(x^t) \leftarrow p_\theta(x^t \mid c^u, s^u, a^u)$          // Decode $c^u, s^u$
                end
                Compute $\mathcal{L}^{(t)}(\mathbf{x}, p(\mathbf{x}); \theta, \phi_c, \phi_s)$          // From Eq. 5
            end
            $\mathcal{L}_{\mathbf{x}} = \sum_t \mathcal{L}^{(t)}$
        end
        Update parameters $\theta$, $\phi_c$, $\phi_s$ by back-propagating gradient $\nabla_{\theta, \phi_c, \phi_s} \sum_{\mathbf{x}} \mathcal{L}_{\mathbf{x}}$.
    end
end
```

S2 Experimental Details

S2.1 Exploratory Procedures

The robot is able to perform four exploratory procedures (EPs) that are briefly described in the full paper. We describe them in greater detail here and show image sequences in Fig. S1. All EPs had a high success rate (object remained in the grasp for the duration of the EP) for every object in our dataset. The object always starts by being dropped into a well that consists of a circle cut out of a sheet of cardboard that is adhered to the horizontal table surface.

- *Drag* (Fig. S1a): The robot moves directly above the well where the object is held at one of three heights selected based on the size of the object. It widens the preshape joint to 0.7 radians and then closes the fingers at 0.5 radians per second until each finger reaches a motor load of 150 (arbitrary digital units on a ten-bit scale), after which the finger position is fixed. The object is moved to a fixed location above the table and lowered at 1 cm/s until the force-torque sensor measures an increase of 1 N in the z-axis. The robot then begins recording data, and the object is dragged 5 cm horizontally across the surface at 3 cm/s in the direction of the single finger, such that the two preshape fingers are behind the object. The grasp parameters were chosen such that all objects could be stably grasped throughout the EP. The arm control parameters were chosen so that the robot fingers never pressed into the table when lowering the objects and so that the objects with high friction were not pulled out of the grasp.

- *Press* (Fig. S1b): To perform press, the robot moves into a fixed position above the well containing the object such that the two preshape fingers can be positioned directly over the center of the well. These two fingers are moved to predefined proximal angles, and the preshape is reduced such that the fingertips just touch. The robot begins recording data, and the gripper is lowered at 1 cm/s until the force-torque sensor measures an increase of 10 N in the z-axis, at which point the arm returns to the starting position at 2 cm/s. The predefined proximal pressing angles are determined by a calibration procedure where the robot moves the hand to a fixed position above the table and slowly closes each parallel finger until its motor load reaches 50. The parameters were chosen so that the fingers would always make contact with the objects and to reduce the amount of noise in the force control.
- *Shake* (Fig. S1c): The robot moves to a fixed position over the object and widens the preshape joint to 0.6 radians. It then slowly closes the fingers at 0.5 radians per second until each finger reaches a motor load of 120, after which the finger position is fixed. The robot then lifts the object to a fixed position such that the two preshape fingers are under the object. The robot begins recording data, and it then rapidly shakes the object four times at approximately 2 Hz by actuating the elbow and first wrist joints with 10 rad/s^2 acceleration. The arm control parameters were chosen so that loose contents in objects were thrown out of contact with the objects. The grasp parameters were chosen to maintain stable grasps on all the objects.
- *Squeeze* (Fig. S1d): The robot moves to a fixed position over the object, widens the preshape joint to 0.7 radians, starts recording data, and then slowly closes the fingers at 0.2 radians per second. Each finger closes until it reaches a motor load threshold of 150 and then maintains that motor load. After every finger has reached the threshold, the fingers open at 0.5 radians per second. The fingers were closed at this speed to prevent rapid increases in motor load and enable more accurate force control. The preshape ensured that all objects were contained in the squeeze.

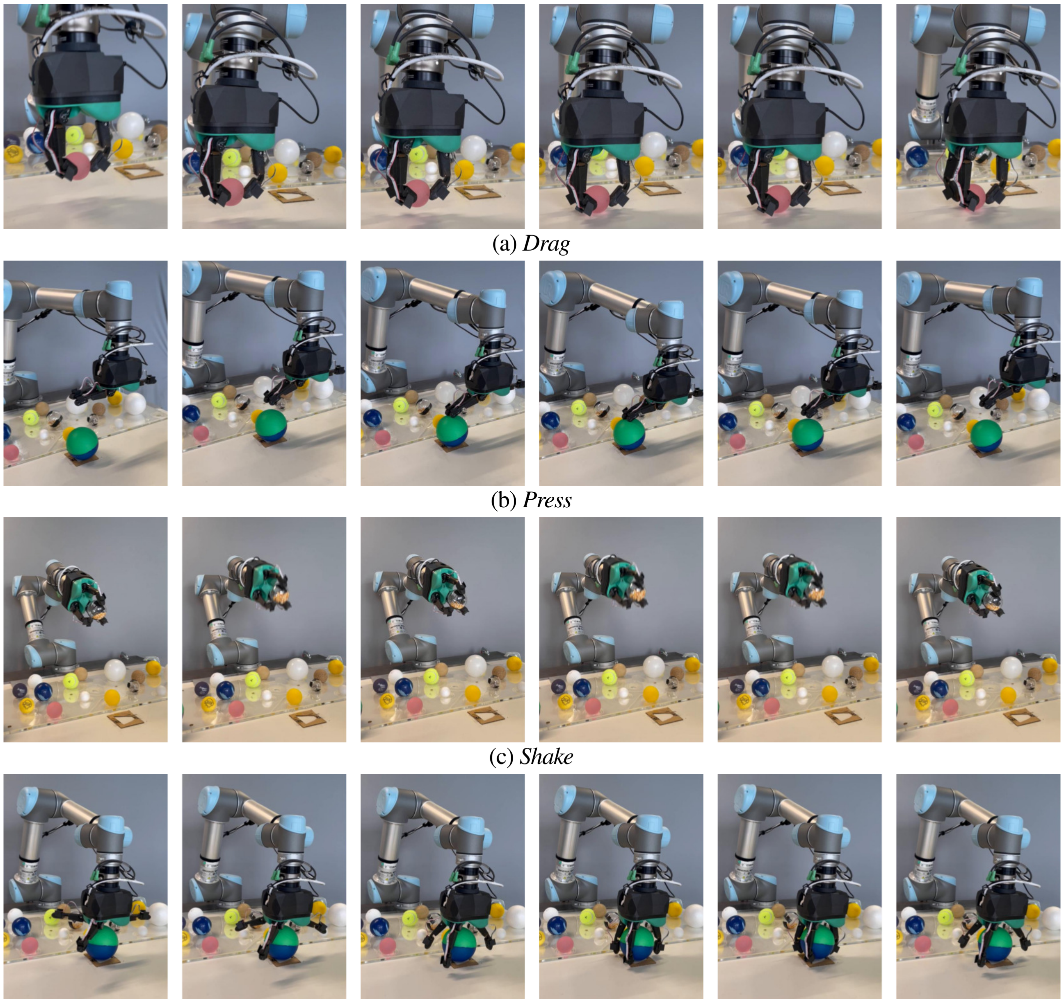
For all force thresholding performed on the data from the KMS 40 force-torque sensor, the threshold was applied to a low-pass-filtered version of the measured force to reduce the influence of high-frequency noise. Specifically, this filtered force was calculated by averaging the most recent 10 measurements, which were collected at 500 Hz. Before each EP, the tactile sensors were all recalibrated by subtracting their present readings, so that they all begin at zero. Furthermore, the IMU sensors were all recalibrated by multiplying by the inverse of their present quaternions.

S2.2 Sequential Group VAE Implementation

The Group and standard VAEs were implemented using the PyTorch library. The models were all trained using the Adam optimizer [S1] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate $= 0.001$.

Instead of using the traditional VAE for our architecture, we use the β-VAE, which introduces the regularization parameter β into Eq. (5) that allows for tuning the trade-off between the negative loglikelihood and the KL-divergence [S2]. We implement separate regularization parameters β_C and β_S for content and style for the Group VAE. The Group VAE was trained with the following parameters:

- MNIST: batch size$= 256$, $\beta_C = 0.01$, $\beta_S = 0.01$, 5 latent dimensions each for content and style.
- Robot Data: batch size$= 32$, $\beta_C = 0.0033$, $\beta_S = 0.0165$, 10 latent dimensions each for content and style, β-NLL$= 0.5$.



(a) *Drag*

(b) *Press*

(c) *Shake*

(d) *Squeeze*

Figure S1: Sequences of still images captured during each of the robot's four EPs.

The standard (no latent split) VAE was trained with the following parameters:

- Robot Data: batch size $=32$, $\beta=0.0033$, 20 latent dimensions, β-NLL $=0.5$.

Structure of Model Inputs The input to a single iteration of the model is an array of size 195×400, where each row represents a different feature over time. There are 18 total features from the UR5 (three values – position, effort, and velocity – from each of the six joints), 45 total features from the Reflex Takktile 2 hand (22 features from one finger, eight from the other two fingers, and seven from the palm), and 132 total features from the KMS 40 force-torque sensor (six DC features and 21 rows from each of the six spectrograms). The length of 400 comes from processing the data and making every action the same duration (described in Section 4).

Robot Data The encoder network consists of four 1D convolutional layers followed by two dense layers, and it outputs the content and style mean and variance, which parameterize $q_{\phi_c}\left(\mathbf{c}^{(t)} \mid x^t\right)$ and $q_{\phi_s}(s^t \mid x^t)$. The four convolutional layers have kernel sizes and strides of $\{6,6,5,4\}$ and $\{4,4,2,2\}$, with $\{128,64,64,32\}$ filters. Each layer is followed by batch normalization and leaky rectified linear unit (Leaky ReLU) activation functions. The output of the convolution layers is flattened and concatenated with the previous content mean and variance. This concatenated vector of size e_d is the input to the first dense layer, which has size $e_d \times 100$ and is followed by the Leaky ReLU activation function. The output of this layer is fed into four separate dense layers, one for each of the content mean and variance and style mean and variance. The outputs of these four layers parameterize $q_{\phi_c}\left(\mathbf{c}^{(t)} \,|\, x^t, q_{\phi_c}^{(t-1)}\right)$ and $q_{\phi_s}\left(s^t \,|\, x^t, q_{\phi_c}^{(t-1)}\right)$.

We then sample a content vector c and a style vector s from these distributions and concatenate them with a one-hot encoding a of the corresponding EP. This vector is then fed into the decoder (shared across all actions). The decoder is the reverse of the encoder except for the slightly larger input to the first layer (to accommodate a) and the final deconvolutional layer, which is split into two equivalent layers whose outputs parameterize the mean and variance of the normal distribution $p_\theta(x \,|\, c, s, a)$.

For the standard VAE we just ignore the style. The encoder doesn't output separate latent spaces, and the decoder only samples from the unified latent space.

Cropped MNIST [S3] The encoder and decoder architectures for modeling the cropped MNIST data are very similar to those used for the robot data. The differences are that the convolutions are 2D, all four convolutional layers use 32 filters with a kernel size of 3 and stride of 2, and the dense layers have 50 hidden nodes instead of 100.

Additional Implementation Notes We slightly alter the loss function from Eq. (5). Instead of computing the style-KL term for all the style vectors $\mathbf{s}^{(t)}$, we compute it only for the most recent style vector s^t. Additionally, the loss is back-propagated through s^t only at iteration t. For all future iterations it is used as a constant, detached from the back-propagation graph, to condition the generative model $p_\theta(x \,|\, \mathbf{c}, s^t, a)$

S2.3 Regression and Binary Classifier Implementation

We implemented relatively small neural networks to perform regression and classification of the object properties. We used networks with one hidden layer with 50 nodes and a Leaky ReLU activation function. The models were all trained with a batch size of 32 using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $= 0.0001$, and weight decay $= 1$. On the other hand, the baseline models share the convolutional architecture of the encoder but then compress the output via two linear layers with 100 hidden nodes followed by a Leaky ReLU activation function. They were trained with the same parameters except for a learning rate $= 0.00003$.

S3 Additional Results

Performance on Synthetic MNIST. As described in Section 3.3, we illustrated our methodology using a synthetic example of the MNIST dataset [S3]. After training many models, we can qualitatively evaluate our model behavior by performing inference over iterative context updates under multiple conditions: using context updates and style, using context updates but ignoring style during inference, and using content and style for inference but bypassing context update. Full reconstructions of test digits are shown in Fig. S2a. Moving from left to right, as more of the digits are seen, the content representation improves, and the reconstructions of both the unseen and seen crops improve.

Selected style-independent and context-independent reconstructions are shown in Fig. S2b. With no style information, the inference relies solely on the content. As more of each digit is seen, the content typically remains stable or improves and more accurately captures general shapes of the digits (middle block of Fig. S2b). Conversely, when the content is not updated, the reconstructions are much messier, and the general digit representation is unstable (lower block).

Latent Representation Variance An interesting result of the evidence accumulation strategy of the Multi-Level VAE is that as more evidence is accumulated, the variance of the group content distribution decreases [S4]. While that variance reduction is not analytically derived from our formulation, our method seems to nonetheless demonstrate a similar property. We generated multiple random sequences for each trial and measured the variances of the latent content and style representations for each of those encoded sequences at each iteration. Fig. S3 shows the resulting distributions over thousands of sequences of the average standard deviations across all ten content and style dimensions as a function of sequence iteration.

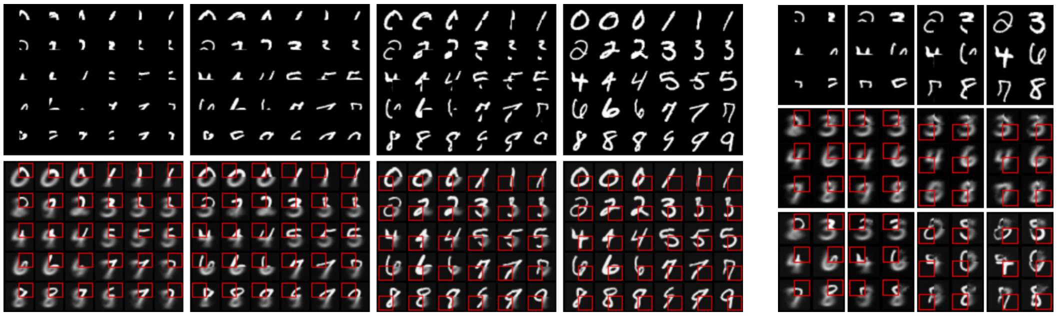


Figure S2: (a) Available crops (top blocks) and corresponding reconstructions (bottom) over four progressive iterations for thirty sample MNIST digits (left blocks). The digits at (row,col) = (2,1), (2,2), (4,3) and (5,3) show the content refinement most clearly over time. (b) Progressive crops for six sample digits (top right blocks) with their style-independent (middle) and context-independent (bottom) reconstructions.

For the content representations, the average standard deviation decreases as a function of sequence iteration for the training and validation, but not for the testing sets. There is less of any trend in standard deviation in the style representations. These findings reinforce our conclusion that the content representations are capturing meaningful information about the essential properties of the objects being touched, and that this information is being accumulated over multiple interactions. Additionally, the style representations encode mainly trial-to-trial variability and do not accumulate information over interactions.

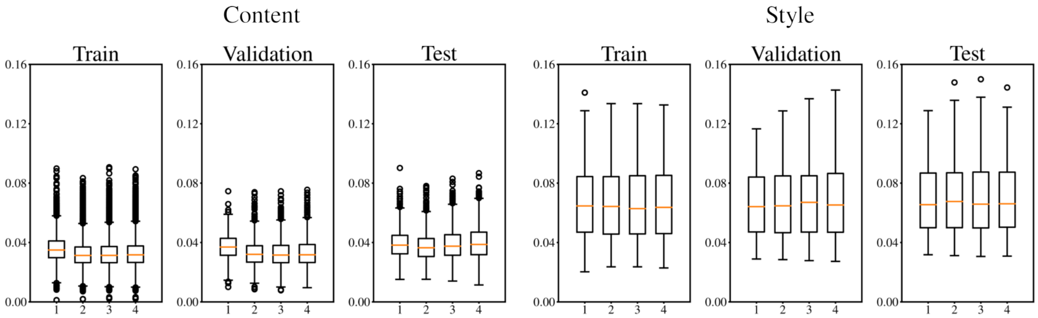


Figure S3: Distributions of the average standard deviations across the ten dimensions of content and style latent representations as a function of sequence iteration. Standard deviation (and therefore variance) in the content representation decreases as a function of sequence iteration. Style variance doesn't change at all.

Qualitative Haptic Property Representation To determine how well haptic properties are captured in the latent content representation, we generated random sequences of four EPs for every object, passed them through the full iterative modeling process, and then compressed the ten-dimensional content space into two dimensions using t-distributed stochastic neighbor embedding (t-SNE). We can then visualize the distribution of each haptic property in the compressed latent space.

Fig. S4 shows the results of this qualitative latent embedding. For visualization purposes, we plotted only 10 random embeddings per object. There is clearly structure encoded in the latent space for all haptic properties. Size and mass are highly separable in just these two dimensions. Interestingly, most of the filled objects are clustered together but are indistinguishable.

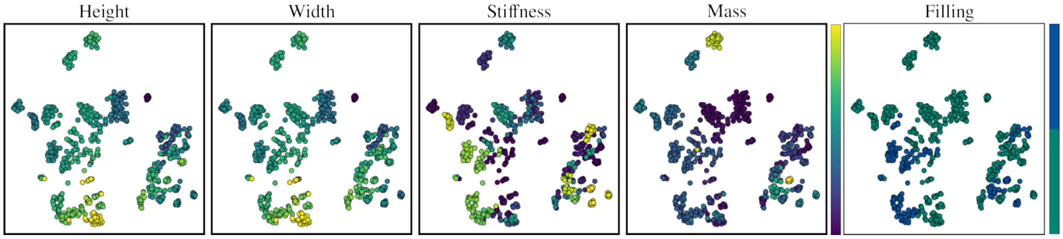


Figure S4: Final-step content latent-space embeddings of 10 sequences per object, as visualized in 2D using t-SNE. The points in each plot are colored to show the respective property value: press size (height), squeeze size (width), stiffness, mass, and whether the object has a filling. For the four continuous properties, higher values are indicated by the colors at the top of the color bar.

References

[S1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. URL http://arxiv.org/abs/1412.6980.

[S2] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[S3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[S4] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.