

Learning the Latent Space of Robot Dynamics for Cutting Interaction Inference

Sahand Rezaei-Shoshtari^{1 2}, David Meger^{1 2}, Inna Sharf³

Abstract—Utilization of latent space to capture a lower-dimensional representation of a complex dynamics model is explored in this work. The targeted application is of a robotic manipulator executing a complex environment interaction task, in particular, cutting a wooden object. We train two flavours of Variational Autoencoders—standard and Vector-Quantised—to learn the latent space which is then used to infer certain properties of the cutting operation, such as whether the robot is cutting or not, as well as, material and geometry of the object being cut. The two VAE models are evaluated with reconstruction, prediction and a combined reconstruction/prediction decoders. The results demonstrate the expressiveness of the latent space for robotic interaction inference and the competitive prediction performance against recurrent neural networks.

I. INTRODUCTION

The research presented in this paper is motivated by the need to increase intelligence and automation in timber-harvesting applications. These are operations which involve a heavy-duty multi-degree-of-freedom arm, mounted on a mobile base and equipped with sophisticated end-effectors, which are employed to saw and manipulate wood. The interaction dynamics that occur in these operations, in particular in the wood-cutting tasks, are complex and present significant challenges for model learning approaches. An accurate representation of the dynamics of such interactions can enable more intelligent control of the arm and motion planning for the whole machine.

This paper aims to tackle some of the challenges related to model learning of robotic arms carrying out a cutting operation. To this end, we use the Kinova Jaco arm equipped with a specialized end-effector as a small-scale proxy for a timber-harvesting machine. In particular, we hypothesize that a latent space, an unobserved space inferred from the raw sensory data of a robot, can provide valuable, yet concise information on its interactions, even in a complex cutting task. We further extend this notion to be able to predict its future states to render the model learning framework useful for model-based reinforcement learning and control.

The inherent strength of latent variable models in capturing representations has made them particularly useful for learning complex dynamics models of systems with unobserved states and interactions. Time-lagged Variational Autoencoders [1] include a time-lag element into the architecture in order to compress complex nonlinear systems in a single embedding with high fidelity to the underlying

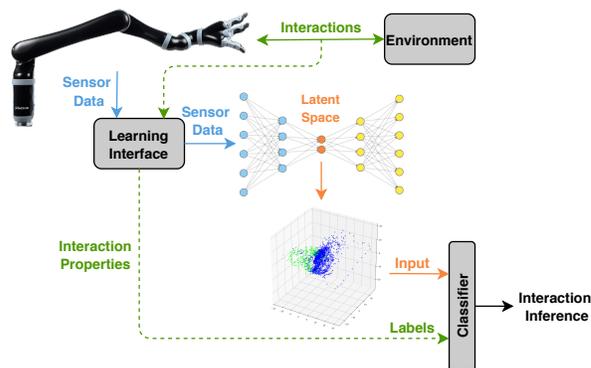


Fig. 1: Overview of the latent space learning and classification framework. First, a latent space is learned from the robot sensor data using Variational Autoencoders and then samples from the latent space are used as the inputs for training a classifier for inferring robot interactions.

dynamics. The time-lag enables the model to act as a standard forward model capable of predicting the future. Neural Relational Inference [2] learns to infer the interactions of a system while simultaneously learning its dynamics from observational data; the latent code represents the interactions of the system in the form of a graph neural network.

Recently, a significant shift towards the use of latent dynamics models has emerged in model-based reinforcement learning [3], [4], [5], [6], [7]. Learning the control policies in the latent dynamics space, rather than the original high-dimensional state space, often results in better generalization [3]. In the same context, deep generative models have been utilized to learn and predict physical interactions of a robotic arm from raw video data [8].

Besides neural networks, Gaussian processes have also received considerable attention in the context of latent variable models [9]. Latent force models [10] have been proposed as a hybrid approach between Gaussian processes and ordinary differential equations; these physically driven kernels are defined based on the underlying interactions of the system and are thus able to better capture them. Stochastic control policies for these models have also been derived for controlling second-order systems with unknown input signals [11].

Closely related to the approach taken in this work, the latent space of dynamics has been substantiated as a powerful tool for capturing physical interactions of a robot, particularly in cutting tasks. DeepMPC [12] proposes an MPC framework for robotic manipulators using recurrent neural networks for learning unobserved properties of food cutting dynamics. Similarly, an MPC formulation for velocity

¹School of Computer Science, McGill University, Montreal, Canada

²Mila Quebec AI Institute ³Department of Mechanical Engineering, McGill University, Montreal, Canada. Correspondence to: Sahand Rezaei-Shoshtari sahand.rezaei-shoshtari@mail.mcgill.ca

controlled robots has been derived based on recurrent neural networks incorporating force/torque measurements of an external sensor [13]. Furthermore, auxiliary tasks, such as thickness prediction of the cutting material, have been shown to help learning an embedding space of the dynamics model from images in the context of vegetable slicing [14].

Differently from previous approaches explored in the context of robotic cutting, our framework uses Variational Autoencoders instead of recurrent models and, furthermore, it only uses proprioceptive measurements of joint states of the robot. With an introduction of a time-lagged element into the decoder alongside the standard reconstruction, we demonstrate the expressiveness of latent space for interaction inference, as well as forward prediction.

We evaluate our proposed method on a series of actual robotic cutting tasks on a variety of wood specimen most commonly found in timber-harvesting operations. To the best of our knowledge, this is the first attempt at learning representations in the context of robotic wood cutting with a saw. In addition, we have released an extensive experimental dataset generated over the course of this work*.

We continue with Section II, which provides a background on Variational Autoencoders; section III describes the proposed framework for learning interaction-rich robot dynamics models, while Section IV evaluates this method by presenting the details of the experiments and the obtained results. Finally, section V summarizes our work.

II. BACKGROUND

A. Variational Autoencoders

Generative latent variable models aim to learn the probability distribution of a dataset by simultaneously learning the distribution of its unobserved lower-dimensional representations. Assuming $x \in \mathcal{X}$ and $z \in \mathcal{Z}$ as the full state and latent space, respectively, samples are then generated from a learned joint distribution of $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$, where vectors \mathbf{z} and \mathbf{x} are respectively sampled from a prior distribution $p_\theta(\mathbf{z})$ and a conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. Imposing a prior distribution on the latent variables ensures the smoothness of the distribution.

Applying Bayes' rule gives the true posterior distribution of the latent variables as $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})/p_\theta(\mathbf{x})$, where the denominator can be calculated by integrating the joint distribution $p(\mathbf{x}, \mathbf{z})$ over the latent space \mathcal{Z} to obtain the marginal likelihood $p_\theta(\mathbf{x})$ as:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}. \quad (1)$$

The underlying task is to maximize likelihood of \mathcal{X} , however, the integration and the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ are intractable. Variational Autoencoders (VAE) [15] tackle this issue by introducing a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ as an approximation to the true and intractable posterior. The recognition model functions as a probabilistic encoder by mapping the inputs into the latent variables, while the generative model $p_\theta(\mathbf{x}|\mathbf{z})$

acts as a probabilistic decoder, by reconstructing the original inputs conditioned on the latent codes.

Rearranging and rewriting the marginal likelihood yields Equation (2) as the loss function in VAEs:

$$\mathcal{L}(\theta, \phi, \mathbf{x}^{(i)}) = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) \quad (2) \\ + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})],$$

where the first term is the KL-divergence between the approximate and true posterior and acts as a regularization term for measuring the amount of information lost by approximating the true posterior. The second term is the reconstruction loss for measuring the expectation of the likelihood of the reconstructed data with respect to the distribution over the latent variables provided by the encoder [16].

In VAEs, both the distribution over the latent variables $p_\theta(\mathbf{z})$ and the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ are assumed to be a multivariate Gaussian. This assumption on the approximate posterior allows for applying the ‘‘reparametrization trick’’ by implementing the posterior as $\log q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \mu^{(i)}, \sigma^{2(i)}\mathbf{I})$ [15].

B. Vector-Quantised Variational Autoencoders

One of the major shortcomings of VAEs is that the prior on the latent space is considered to be a Gaussian; this assumption is maintained throughout the learning process by imposing the KL-divergence term in the loss function as defined in Equation (2). As a result, their performance can decrease when this assumption is not satisfied. On the other hand, Vector-Quantised Variational Autoencoders (VQ-VAE) [17] do not impose the said assumption and learn a discrete latent representation while simultaneously learning the prior.

In VQ-VAEs, the prior and posterior distributions are categorical and the samples drawn from the latent representation index an embedding space $e \in \mathbb{R}^{K \times D}$ where K is the size of the discrete latent space and D is the dimensionality of each latent embedding vector. The input \mathbf{x} is passed to the model and the encoder creates the output $\mathbf{z}_e(\mathbf{x})$; the discrete latent variables \mathbf{z} are in turn calculated using a nearest neighbour look-up in the embedding space e .

In general, discrete latent embeddings can be a potent substitute in cases where the latent variables are discrete by nature. Additionally, from the empirical standpoint, it has been demonstrated that VQ-VAEs do not suffer from the ‘‘posterior collapse’’ which is often observed in VAEs. Posterior collapse happens when the latent variables are ignored by the decoder when it is powerful enough to reconstruct the original inputs, regardless of the latent embeddings [17].

III. METHODOLOGY

Traditionally, for modeling and control of robotic arms with active interactions, the forces have been identified and included in the rigid body dynamics model [18], stated as:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_\varepsilon + \mathbf{J}^\top(\mathbf{q})\mathbf{F}, \quad (3)$$

where $\mathbf{q} \in \mathbb{R}^N$ is the vector of joint positions, $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{N \times N}$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^N$ represents Coriolis and centrifugal forces, $\boldsymbol{\tau}_g(\mathbf{q}) \in \mathbb{R}^N$ denotes gravitational effects, $\boldsymbol{\tau} \in \mathbb{R}^N$ is the vector of joint torques, $\boldsymbol{\tau}_\varepsilon$ is the

*The dataset and its details are available at https://github.com/McGill-AML/aml_robot_cutting_dataset

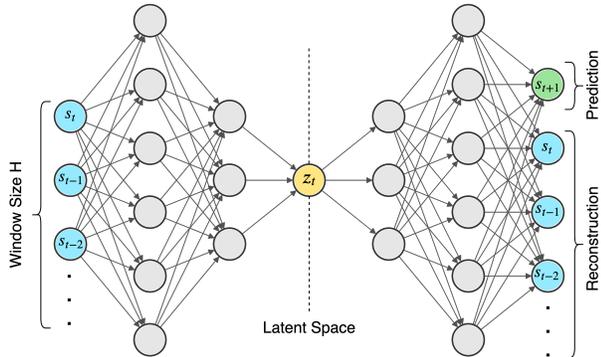


Fig. 2: Variational Autoencoder as a dynamics model. The inputs are the the states over a window of size H .

additional torque due to friction and disturbances, $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix with respect to the end-effector, and \mathbf{F} is the interaction force and moment applied to the end-effector.

In the context of cutting interactions, a common approach in contact dynamics is to approximate the interaction force as a spring-damper system of the form [18]:

$$\mathbf{f} = \mathbf{K}_p(\mathbf{x} - \mathbf{x}_r) + \mathbf{K}_d(\dot{\mathbf{x}} - \dot{\mathbf{x}}_r), \quad (4)$$

where \mathbf{x} and \mathbf{x}_r are respectively the current and reference positions of the blade inside the specimen. The values for \mathbf{K}_p , \mathbf{K}_d , and \mathbf{x}_r are dependant on the properties of the specimen (e.g., material and thickness) or configuration of the cutting tool. However, capturing these unobservable parameters of the environment can be challenging. Instead of utilizing System Identification tools [19], we propose to learn a latent space that encodes these unobservable features alongside a dynamics model in a unified framework for inferring interactions and predicting future states.

A. Latent Space of Robot Dynamics

We utilize Variational Autoencoders to learn the latent space of cutting dynamics. VAEs are usually trained via reconstruction of input vectors using a reconstructive decoder network. However, in the context of dynamics models, reconstruction of current robot joint states, albeit useful for learning expressive representations, does not allow for prediction of future states, which is essential for model-based reinforcement learning and control. Thus, we propose to modify the decoder to predict the next states alongside the standard reconstruction, as shown in Figure 2. During training, the loss function is then extended to impose the prediction error in addition to the reconstruction and KL-divergence losses. The prediction loss uses the representation to capture the transition model of the underlying dynamics, while the reconstruction and KL-divergence losses ensure expressiveness of these representations, without which the performance of interaction inference is decreased significantly as demonstrated in Section IV.

In this work, the input vectors of the dynamics model $\mathbf{x} \in \mathbb{R}^M$ are defined as the manipulator joint states (positions, velocities and torques) aggregated over the last H time-steps. Therefore, for a robot arm with N degrees of freedom, the input is a $\mathbf{x} \in \mathbb{R}^{3 \times H \times N}$ vector. Furthermore, a reconstruction decoder, has $\mathbf{x}' \in \mathbb{R}^{3 \times H \times N}$ output vectors, whereas a

prediction decoder outputs $\mathbf{x}' \in \mathbb{R}^{2 \times N}$ vectors, as it predicts only the joint positions and velocities of the next state.

B. Classification with Deep Generative Models

The approach taken here is closely related to the methods seen in semi-supervised learning frameworks. Latent-feature discriminative models (M1 models) are commonly used in this context and are based on the idea that a learned model capable of providing embeddings or feature representations of the data, can significantly simplify the classification task by clustering the related observations in a latent space [20].

Taking the same approach, we first learn the latent space or latent embeddings of robot dynamics with VAE and VQ-VAE models and consequently learn a classifier for predicting class labels from this latent representation. As discussed in [20], the lower dimensionality of the latent space as well as the fact that the non-linear transformation of the data allows for its higher moments to be captured by the latent space, improve the classification performance. Figure 1 shows an overview of the learning framework; support vector machines (SVM) have been used as the classifier.

IV. EVALUATION

In this section, we evaluate the performance of the framework described earlier for inferring robot interactions and predicting future states in interaction-rich environments.

A. Experimental Setup

Motivated by the timber-harvesting application, the experiments in this research are carried out on a series of real robotic cutting tasks with different wood specimens. We have collected a Robot Cutting Dataset consisting of approximately 1500 seconds, a total of 180 cuts (15 cuts on 12 unique wood specimens) using Kinova Jaco arm retrofitted with a custom end-effector fixture and saw. The wooden rods have been selected from sets of five materials and five thicknesses (see Table I). The test bed, Figure 3, consists of four components:

- **Kinova Jaco 2 Arm:** A light-weight six-DOF robotic arm equipped with joint encoders and joint torque sensors for collecting data at 10 Hz.

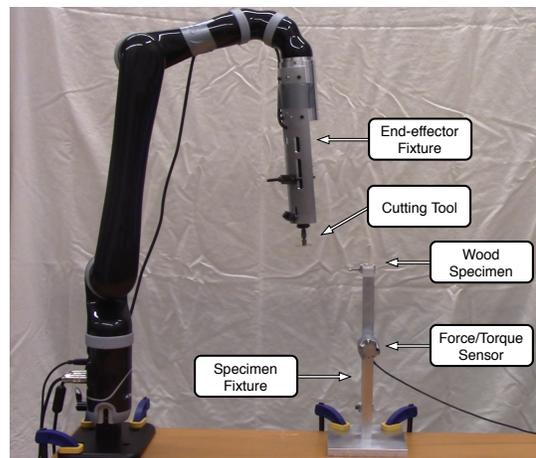


Fig. 3: Robot cutting test bed used for data collection. A custom end-effector fixture replaces the original gripper.

- **Cutting Tool and End-effector Fixture:** Designed as a replacement for the original three-finger gripper, the aluminum end-effector fixture holds the cutting tool. The tool is a Dremel 100 Single Speed Rotary Tool Kit, equipped with a circular saw blade.
- **Specimen Fixture:** This component provides a firm support for holding the wood specimen and measuring the cutting force, using an integrated force/torque sensor in the middle of the fixture. The latter is not used in our work because integration of sensors on a timber-harvesting machine is highly challenging.

During each cut, the robot is moved manually from the start position, Figure 3, towards the wood specimen in Cartesian velocity control. In the case of stiff and thick specimens, a cut is not possible with a single pass and the tool needs to be retracted for a second try. This multimodality imposes further challenges for latent space learning.

The dataset is split into three subsets for training (80%), validation (10%), and test (10%). Furthermore, for labeling

TABLE I: Details of wood specimens used in the dataset.

Material	Thickness (in)				
	3/16	1/4	5/16	3/8	7/16
LVL	✓	✓	✓	✓	✓
Maple	□	✓	□	✓	□
Oak	□	✓	□	✓	□
Birch	□	✓	□	✓	□
Hardwood	□	□	□	✓	□

the dataset and identifying robot’s engagement in a cutting interaction, we have compared the moving mean and standard deviation of joint torques over a window [21]: a sudden increase in the variance can be interpreted as a start or end time of a cutting episode.

The network architectures are multilayer perceptron (MLP) with two hidden layers, each of dimension 40, with standard Rectified Linear Unit (ReLU) [22]. We trained the models using the Adam optimizer [23] with learning rate $5e-4$ and batch size 128. The data has been normalized to have zero mean and standard deviation of 1.

B. Interaction Inference

As an evaluation of the quality and expressiveness of the latent space of robot dynamics, we solve a classification problem for inferring the properties of robot interactions. Namely, we consider three classification tasks: (1) cutting detection for inferring whether the robot is engaged in a cut, (2) material classification, and (3) thickness classification.

Figure 4a shows the classification performance of SVM models trained on the latent representation of VAE models, as a function of latent space dimension for various values of KL-divergence weight in the loss function. Figure 5a presents the classification performance on the latent embeddings learned by VQ-VAEs, as a function of embedding dimension and number. The performance is measured using the standard $F_1 = 2/(p^{-1} + r^{-1})$ score which is the harmonic mean of precision p and recall r . In order to fully investigate the importance of reconstruction and prediction decoders in

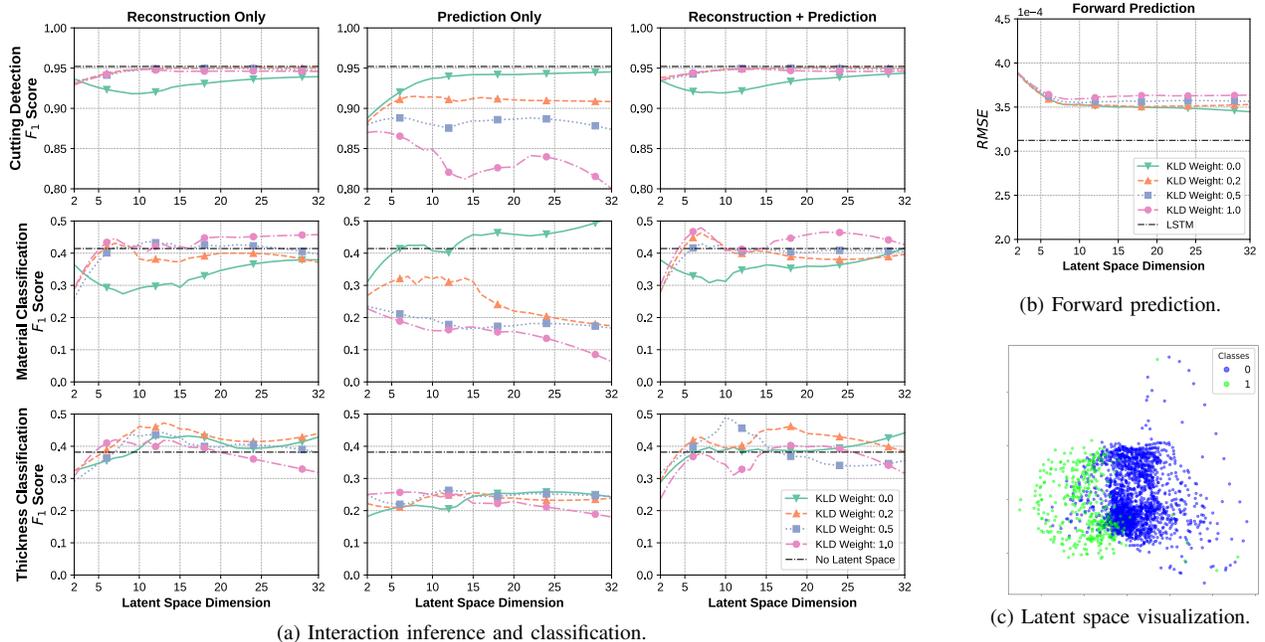


Fig. 4: Performance of VAE models for different dimensions of the latent space, evaluated on the test set. (a) Interaction inference using the latent space of VAE models, separated by the classification task at each row (cutting detection, material classification, and thickness classification) and the setup of the VAE at each column (reconstruction only, prediction only, and both). The baseline method is classification directly on the state space. (b) RMSE of forward prediction of a VAE model. The baseline method is an LSTM network. (c) Visualization of a 4-dimensional latent space of a VAE model. The latent vectors have been projected to 2-dimensional space using PCA; the blue circles indicate free-motion and the green circles represent a cutting-motion.

learning expressive latent representations, each classification task has been tested with three implementations, the results for which are organized in respective columns: (1) reconstructing the current joint states, (2) predicting the next joint states, and (3) simultaneous reconstruction and prediction.

As the interaction inference results for prediction-only models suggest, learning the latent space by solely predicting the next state of the robot does not provide an interpretative representation of the state for robotic interactions. This could be the result of a “mode collapse”: since the output of the predictive decoder is of much lower dimension compared to that of the reconstructive decoder, the latent space is ignored by the predictive decoder as it can predict accurately without relying on the latent space. Combining the two, as suggested in Section III, ensures both the interpretability of the latent space, and the predictive ability of the decoder, thus making it useful for model-based reinforcement learning and control. Moreover, the classification performance is increased slightly compared to reconstruction-only models, as employing a predictive model helps with capturing the time-dependency of the system in the latent space.

In addition, the outperformance of the proposed framework against the baseline method, classification on the state space, emphasizes the fact that the lower dimensionality of the latent space is able to capture higher moments of data and

TABLE II: F_1 scores of the best models selected during validation and evaluated on the test set.

	VAE		VQ-VAE		Baseline*
	Dim.	Score	Dim. \times Num.	Score	
Cutting	11	.9526	2×10	.8979	.9521
Material	4	.4222	6×8	.4623	.4143
Thickness	6	.5107	10×12	.3842	.3817

*The baseline is classification directly on the state space.

simplify the classification task. Figures 4c and 5c present the visualization of the latent space of VAE and VQ-VAE models as an example of the interpretability of the latent space.

Table II compares the F_1 scores obtained on the best latent variable models. The best model in each category is chosen respectively based on the validation loss during training and is thus only a representative of the latent space learning performance. Notwithstanding the effect of KL-divergence weight on the performance of VAE models, they outperform VQ-VAEs in general, despite their simplifying assumptions. The continuous domain of dynamics models can be an explanation for this, as it can be better captured through a continuous latent space. Further optimization of the KL-divergence in VAEs, can also improve their performance.

Finally, it is worth mentioning that the generally low scores on material and thickness classification tasks are due to the inherent difficulty of inferring these values by relying

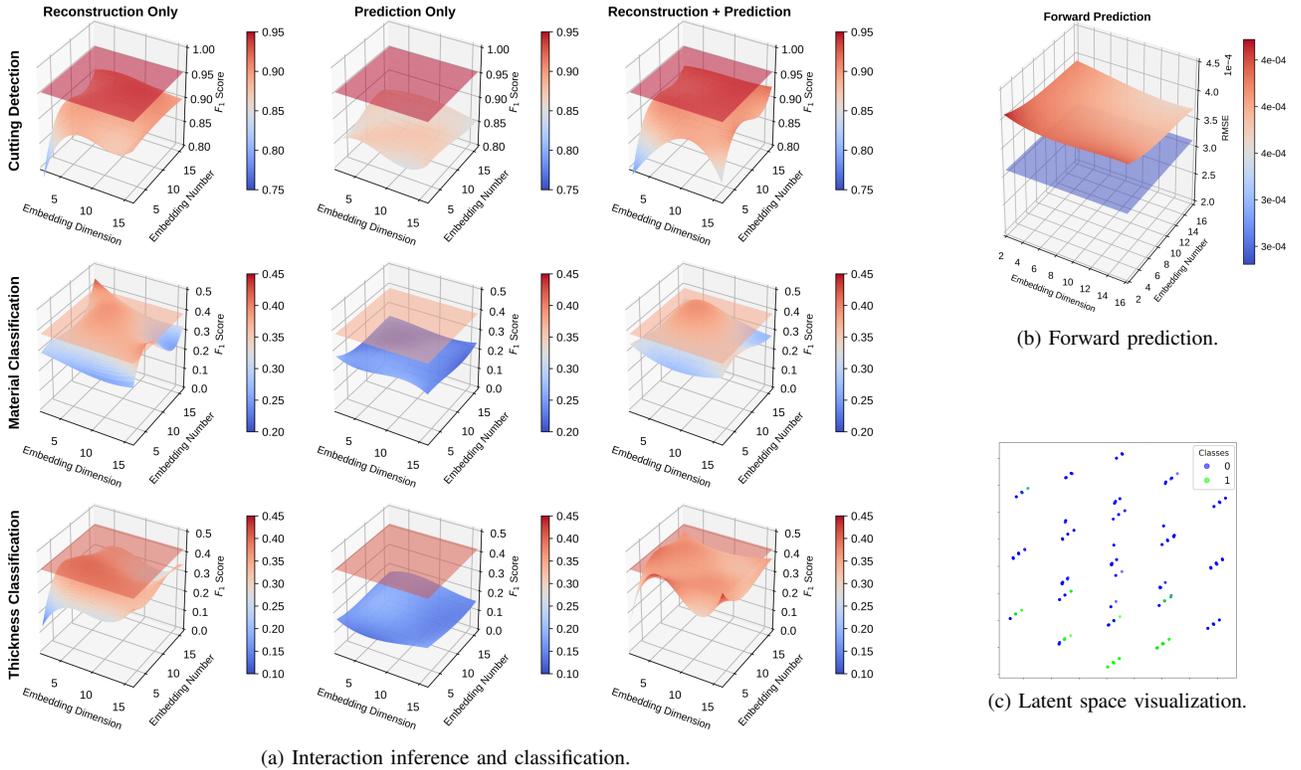


Fig. 5: Performance of VQ-VAE models for different dimensions and numbers of the latent embeddings, evaluated on the test set. (a) Interaction inference using the latent space of VQ-VAE models, separated by the classification task at each row (cutting detection, material classification, and thickness classification) and the setup of the VAE at each column (reconstruction only, prediction only, and both). The baseline method is classification directly on the state space. (b) RMSE of forward prediction of a VQ-VAE model. The baseline method is an LSTM network. (c) Visualization of a 4×4 -dimensional latent space of a VAE model. The latent vectors have been projected to 2-dimensional space using PCA; the blue circles indicate free-motion and the green circles represent a cutting-motion.

TABLE III: RMSE of forward prediction of the best models selected during validation and evaluated on the test set.

	VAE		VQ-VAE		LSTM
	Dim.	RMSE	Dim. \times Num.	RMSE	
Prediction	11	$3.54e-4$	2×10	$3.93e-4$	$3.12e-4$

only on the robot joint states.

C. Prediction

In order to reliably apply a dynamics model for planning and control, the forward prediction errors need to be substantially low. As an evaluation of the prediction performance of the proposed approach, we compare it with recurrent neural networks, namely LSTMs [24]. We have used a sequence of robot states with the same length as the window size H as the inputs to a two-layer stacked LSTM network with a hidden dimension of 40 and ReLU activation units. We conducted an active search for hyperparameter values to tune the network and noticed negligible performance changes.

Figures 4b and 5b show the root-mean-square error of forward predictions for VAE and VQ-VAE models as a function of the latent space size. In Table III, we summarize the RMSE of the best validation models chosen during training, the same models as in the first row of Table II.

As the results show, LSTM models marginally outperform VAE and VQ-VAE models. Despite the fact that latent variable models use a simpler network architecture without any recurrent feedback, they can achieve competitive performance for modeling robot dynamics with unobserved interactions, thus showcasing the importance of latent space learning. In addition, the latent space of these models can be directly used for inferring robot interactions as discussed in Section IV-B; whereas, there is no explicit access to the latent space of recurrent neural networks for such use cases, other than their saved memory.

V. CONCLUSION

In this paper, we explored the notion of latent space of robot dynamics models, particularly in the case of complex and interaction-rich environments. We demonstrated that by modifying a Variational Autoencoder to predict the future states of a robot, alongside the reconstruction of the current states, an expressive latent space can be learned directly from the raw sensory joint states data over a finite window. To evaluate our approach, we collected a dataset from a series of robotic wood cutting tasks with a retrofitted Kinova Jaco arm. We showed that the learned latent representations can encode the information required for drawing inference on robot's interactions. Furthermore, we showed that the prediction capabilities of the proposed model are competitive with recurrent neural networks, while providing more interpretative information through the latent space. Future work may explore recurrent autoencoders and use of predictions in model-predictive control.

ACKNOWLEDGMENTS

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN).

REFERENCES

- [1] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, "Variational encoding of complex dynamics," *Physical Review E*, vol. 97, no. 6, p. 062412, 2018.
- [2] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *arXiv preprint arXiv:1802.04687*, 2018.
- [3] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, 2019, pp. 2555–2565.
- [4] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Advances in Neural Information Processing Systems*, 2015, pp. 2746–2754.
- [5] L. Kaiser, M. Babaieizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al., "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.
- [6] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare, "Deepmdp: Learning continuous latent space models for representation learning," in *International Conference on Machine Learning*, 2019, pp. 2170–2179.
- [7] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine, "Solar: Deep structured representations for model-based reinforcement learning," in *International Conference on Machine Learning*, 2019, pp. 7444–7453.
- [8] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in Neural Information Processing Systems*, 2016, pp. 64–72.
- [9] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in neural information processing systems*, 2004, pp. 329–336.
- [10] M. A. Alvarez, D. Luengo, and N. D. Lawrence, "Linear latent force models using gaussian processes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2693–2705, 2013.
- [11] S. Sarkka, M. A. Alvarez, and N. D. Lawrence, "Gaussian process latent force models for learning and stochastic control of physical systems," *IEEE Transactions on Automatic Control*, 2018.
- [12] I. Lenz, R. A. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control." in *Robotics: Science and Systems*. Rome, Italy, 2015.
- [13] I. Mitsioni, Y. Karayiannidis, J. A. Stork, and D. Kragic, "Data-driven model predictive control for food-cutting," *arXiv preprint arXiv:1903.03831*, 2019.
- [14] M. Sharma, K. Zhang, and O. Kroemer, "Learning semantic embedding spaces for slicing vegetables," *arXiv preprint arXiv:1904.00303*, 2019.
- [15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [16] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
- [17] A. van den Oord, O. Vinyals, et al., "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6306–6315.
- [18] R. Featherstone and D. E. Orin, "Dynamics," in *Springer Handbook of Robotics*. Springer, 2016, pp. 37–66.
- [19] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robotics and computer-integrated manufacturing*, vol. 26, no. 5, pp. 414–419, 2010.
- [20] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.
- [21] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.
- [22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.