Predicting Hand-Object Interaction for Improved Haptic Feedback in Mixed Reality

M. Salvato, Negin Heravi, Allison M. Okamura, *Fellow, IEEE*, and Jeannette Bohg

***Abstract*—Accurately detecting when a user begins interaction with virtual objects is necessary for compelling multi-sensory experiences in mixed reality. To address inherent sensing, computation, display, and actuation latency, we propose to predict when a user will begin touch interaction with a virtual object before it occurs. We hypothesize that the sequence of hand poses when approaching an object, combined with object pose, contain sufficient information to predict when the user will begin contact. By leveraging this information, we could reduce or eliminate latency in providing haptic feedback during virtual object interaction. We focus on small time horizons, on the order of 100 ms, to overcome sense-to-actuation latency for haptic feedback in mixed reality systems. We use a time series of tracked hand poses, along with virtual object geometry to perform our prediction. By calculating minimum hand-object distance and feeding those along with hand poses to a self-attention-based network, we achieve approximately 52.8 ms of timing error for a 100 ms prediction horizon. Additionally, we test our system against different levels of tracking and hand-object alignment noise, finding minimal change in timing error. By contrast, when only extrapolating joint and hand velocities, we find that timing error consistently exceeds the prediction horizon.**

***Index Terms*—Grasping, haptics and haptic interfaces, human detection and tracking, human and humanoid motion analysis and synthesis.**

I. Introduction

COMPELLING virtual and augmented reality experiences with haptic feedback require that virtual object interactions occur smoothly and with minimal latency [1]–[3]. In virtual reality, humans can detect errors in haptic stimulation timing that are more than 15 ms before or 50 ms after visual stimulation [4]. Additionally, [5] found 45 ms to be the required temporal resolution for optimal realism in visuo-haptic interfaces. While state-of-the-art hand tracking can have latency as low as 13ms [6], hand pose reconstruction errors [7], especially when exacerbated by errors such as blurring [8] or occlusion, result in reduced tracking accuracy [7] and consequently timing errors for human-object interaction. This reduces realism and can create force-feedback instability for haptics. Furthermore, haptic device latency due to hardware limitations introduces additional challenges. For example, low speed, encountered-type haptic devices have visual haptic latency as high as 200 ms [9]. A recent haptic latency testbed had approximately 100 ms latency between a sensed tap and haptic feedback [4]. These issues highlight the need to consider novel approaches to augment hand tracking for haptic interaction, in order to accurately determine touch interaction timing and movement with maximal accuracy. Specifically, by predicting contact before it happens, we can circumvent such latencies by sending the actuation signal to the haptic device early, resulting in more precise haptic feedback timing.

To address this challenge, we propose an interaction-expectation model for hands interacting with virtual objects and demonstrate its ability to predict contact timing in advance (Fig. 1). We hypothesize that changes in hand pose over time encode information about when the human expects object interaction to begin, and develop an interaction-expectation model based on this hypothesis. The interaction-expectation model augments existing hand tracking systems by providing predicted future hand-object interaction timing. These predictions can be used to reduce object interaction latency resulting in improved haptic interaction timing and increasing realism in mixed reality systems. Our model uses a recorded history of hand points, as well as virtual object pose, as input to predict upcoming hand-object interaction before it begins. The output of our model is the probability of contact for a fixed time horizon into the future. To achieve this, our model uses an architecture based on fully connected layers with skip connections, combined with self-attention [11], trained on the publicly available GRAB dataset [12]. We compare the performance of our model when using as input either the full mesh representation of the hand, or only the joint vertices, with different levels of measurement noise. As an evaluation metric, we use the accuracy of first detected contact time by the model.

In sum, our paper presents a predictive hand-object interaction-expectation model that is learned from human grasping data. The main contributions are:

1) By using a history of hand pose information, our model predicts contact before it occurs with an average timing error of 52.8 ms for a prediction horizon of 100 ms. By contrast, extrapolating joint and hand velocities into the

Manuscript received September 9, 2021; accepted January 10, 2022. Date of publication February 7, 2022; date of current version February 16, 2022. This letter was recommended for publication by Associate Editor Hiroyuki Kajimoto and Editor Jee-Hwan Ryu upon evaluation of the reviewers' comments. The work of Negin Heravi was supported by the NSF Graduate Research Fellowship.This work was supported by National Science Foundation under Grant 1812966, a Link Foundation Modeling, Simulation, and Training Fellowship, Stanford HAI and the HAI-Google Cloud Credits Program. *(Corresponding author: M. Salvato.)*

M. Salvato, Negin Heravi, and Allison M. Okamura are with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: msalvato@cs.stanford.edu; nheravi@stanford.edu; aokamura@stanford.edu).

Jeannette Bohg is with the Department of Computer Science, Stanford University, 94305 USA (e-mail: bohg@stanford.edu).

Digital Object Identifier 10.1109/LRA.2022.3148458

Fig. 1. The interaction-expectation model predicts contact with a virtual object before it occurs. As a demonstrative example, the user begins reaching for the virtual apple. We predict the interaction before it occurs. This information can be communicated to a haptic device (picture from [10]), so it can provide haptic feedback at the correct time, unhindered by sense-to-actuation latency.

future and checking mesh to mesh contact yields an error of 129.7 ms.

2) We study the effectiveness of using different hand representations, specifically full hand meshes or joint vertices only.
3) We show that our proposed model is robust to noise. For per point Gaussian noise with a mean error of 30 mm, we find a 1.4 ms increase in timing error. An additional 5 mm of translational noise across the whole hand results in an additional 2.2 ms of error.
4) We demonstrate that our model generalizes to grasping behavior of both new users and new objects.

II. Related Work

The increased availability of mixed reality headsets has demonstrated the importance of visual hand tracking. For example, the Oculus Quest [13], uses a tracking pipeline that detects a hand, determines the joint pose, and creates a mesh of it. Concurrently, a number of wearable technologies for haptic feedback in virtual reality have been developed to give rich mixed reality experiences to users in a variety of forms, such as wrist-worn devices that provide haptic feedback not collocated with the stimulus [10], fingertip devices designed to give cutaneous feedback to the skin on the fingertip as a user interacts with virtual objects [14], encountered-type haptic devices such as robotics shape displays where a robot moves a tangible surface to the point of interaction [9], [15], as well as devices capable of applying forces across finger joints, allowing the device to stop the motion of the user's fingers as they grab an object [16]. Unfortunately, the hand tracking methods popular for virtual reality headsets [13] are insufficient for many of these haptic devices. This is because these methodologies are either too imprecise or too slow for compelling haptic experiences. Delays in the haptic signal can affect the perception of the mechanical properties of virtual objects such as compliance [1] and cause instability [2], [3]. To mitigate these issues, haptics researchers use other methods for tracking, such as electromagnetic trackers [14], which are expensive and cumbersome.

Haptic devices introduce their own latency caused by hardware limitations. To mitigate these issues, [9] proposed illusions such as using dynamic redirection for encountered-type haptic devices to direct the user's motion to a device reachable state. However, these methods have a low threshold at which the user starts noticing the illusion [17].

Prior studies have looked into full human body pose prediction [18]–[20] with potential delay reduction applications, but predictive hand tracking requires finer-scale tracking and prediction due to the fast and precise movement of the hands. Researchers have studied predictive hand tracking for gesture prediction [21] and for detection of the user's intended reach target using Long Short-Term Memory networks [22] and temporal trajectory template matching [23], but they do not predict the timing of contacts.

III. Interaction-Expectation Model

To address interaction timing errors caused by latency in both the tracking system and haptics hardware, we propose an interaction-expectation model. In this model, based on a user's hand movements, we predict when the user expects to begin interacting with an object via touch. We consider prediction horizons of 50 ms, 100 ms, and 200 ms. Depending on the system, these horizons provide sufficient time to prepare the hardware for rendering an upcoming contact before it has occurred.

A. Input and Output

At regularly spaced time intervals, the proposed model takes as input H points in 3D space representing the hand, as well as O points representing an object of known geometry to be interacted with. At timestep t, t such values have been generated. The models then maps

$$\mathbb{R}^{(H+O)\times 3\times t} \rightarrow \{0, 1\}$$

The binary output value represents whether the system believes contact will be made at timestep $t + D$, where $D \in \mathbb{Z}$ is a hyperparameter indicating the prediction horizon for object contact.

B. Assumptions

Our approach assumes: (1) 3D points for hands passed in are from the same person and represent a continuous movement, (2) across timesteps, the point at a given index represents the same anatomical location, and (3) hands are sampled at regular intervals.

IV. Training Network

Our interaction-expectation model prediction is performed using a network consisting of fully-connected layers with skip connections feeding into a self-attention layer. The code for our model is available: https://github.com/charm-lab/interaction-expectation

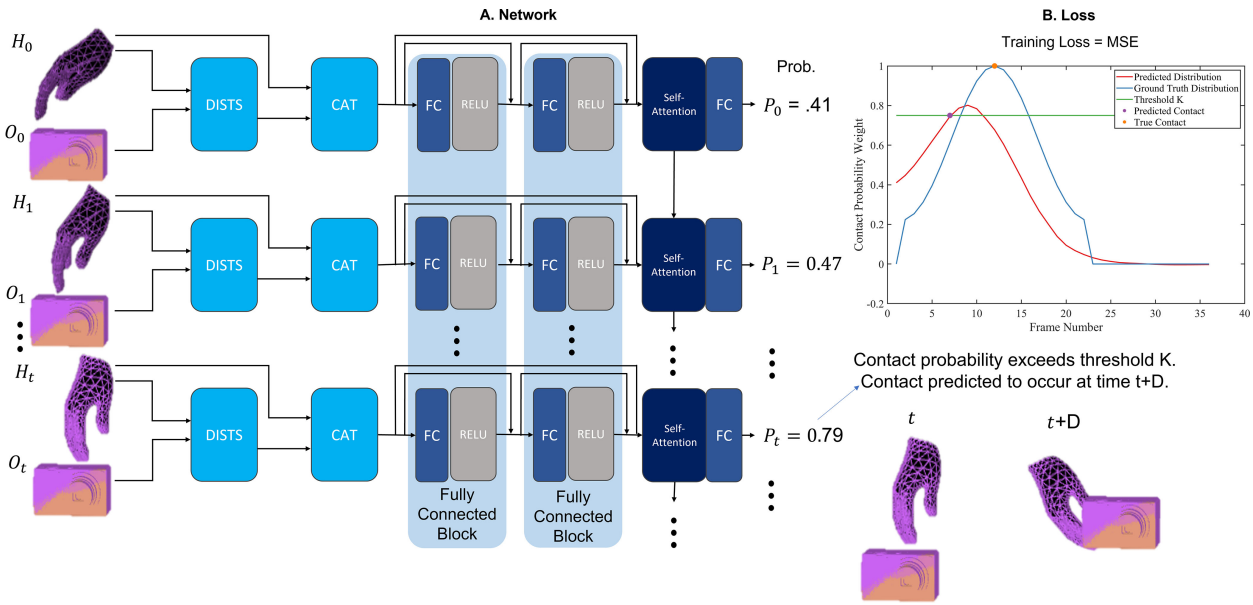

Fig. 2. Network architecture. (A) At each timestep, the network takes as input the hand and object mesh. The distance between each hand vertex and the nearest object vertex is calculated. This vector is concatenated with the hand vertices and passed through two fully connected blocks, each consisting of a fully connected layer and rectified linear unit, with skip connections between them. This is used as an encoding to a self-attention and fully connected layer. The self-attention mechanism is masked to only use prior data. (B) Our network outputs contact probability at time $D + t$ for each t. This produces a distribution, and loss is computed as the mean squared error of the predicted distribution and ground truth distribution. To determine the single contact timing of the network, we set K, the threshold for the predicted probability of contact. In this example we use a time horizon of $D = 12$ timesteps (100 ms at 120 Hz). DISTS: Minimum hand-object distances, CAT: Concatenate, FC: Fully Connected, BN: BatchNorm.

A. *Hand-Object Distances*

To compute the input to our model, we assume a known hand and object mesh. As the first step, we compute the minimum distance from each hand vertex to the nearest object vertex at each timestep in the movement sequence. We concatenate these hand-object distances with the positions of each hand vertex. Therefore, the input to the network is $\mathbb{R}^{H\times 4}$, where H is the number of hand vertices. This reduces the dimensionality of our input space dramatically compared to inputting objects directly, while still maintaining key information about the nature of the approach of the hand to the object. As will be described in Section IV-B, we can apply this method to a full hand mesh or using only the joint vertices. We find that on average, this operation takes 9.1 ms for full hand meshes and 0.3 ms for joints only with a Intel Xeon E5-1650 v4 CPU at 3.60 GHz.

B. *Neural Network*

To predict future hand-object contact, we leverage the change in hand configuration and hand-object distance over time by incorporating a self-attention mechanism into our model. We found that fully connected layers with skip connection are effective at encoding information from the hand and hand-object distances to pass into the self-attention layer. Fig. 2 A shows the full model architecture. A fully connected layer converts the output of the self-attention to the appropriate dimensionality for our loss function. The self-attention layers are masked to only use prior data.

C. *Latency*

We ran our distance computation and neural network using an Intel Xeon E5-1650 v4 CPU and NVIDIA Quadro P5000 GPU. We find that on average, the hand-object distance operation takes 9.1 ms for full hand meshes and 0.3 ms for joints.We found the estimated average time for a single timestep to pass through the neural network is 0.14 ms for all mesh vertices, and 0.06 ms for joints only. This yields a total inference time of 9.24 ms for full mesh vertices, and 0.36 ms for joints only. This is much less than required for a 100 ms horizon.

D. *Loss*

For input data at time t, we wish to predict whether contact will occur at time $t + D$. For each time t, our network outputs a probability of contact $t + D$. This results in an array P of size T with each such prediction. Fig. 2 B shows an example network prediction.

In practice, it is difficult to train a model that predicts only binary contact. Instead, we provide a blurred array of the expected output, P^*, as seen in Fig. 2 B. We blur the binary contact data with a Gaussian with a standard deviation of 5 timesteps (4.2 ms), cut off at ± 10 timesteps (8.3 ms) from the true value. Our loss is the mean squared error between the predicted and ground truth distributions, $MSE(P, P^*)$.

E. *Initial Contact Timing Error Metric*

To determine the single initial contact timing of the network, we set K, the threshold for the predicted probability of contact. We define P_t network output at time t. We determine initial

contact is made at

$$t' = \underset{t \in T}{\operatorname{argmin}}(t | P_t \geq K) + D$$

If t^* is the true contact time, our metric, the timing error, for that sequence is $|t' - t^*|$.

F. Training Procedure

For all experiments we partition our data into train-validation-test splits. We trained each model for 100 epochs of our data. We found all experiments reached their minimum loss, and began to diverge, prior to this number of epochs. To determine the test model, we used the model with the minimum loss over all validation sequences.

During training, we test $K = [0, 0.01, 0.02, \ldots, 1]$. We select the K for which the mean initial contact timing error time across all sequences is minimized. During test time, we choose the K which was optimal on our validation set.

For a given sequence, if there does not exist a time t for which $P_t \geq K$, no contact is detected. In practice, this drives the model to low K and high error. Thus, during training we determine K for a given model by selecting the K for which timing error is minimized, excluding up to 5% of the sequences for which no contact was detected. We find as the model trains successfully; even on the test set, the results are typically below the 5% 'no detection' rate.

We randomize the start time of sequences during training to provide generalization. The sequences can start any time from the initialization of data, to $2 * D$ before contact.

V. Experiments

The primary goal of our experiments was to evaluate the performance of our interaction-expectation model to predict the timing of initial hand-object interaction before it has occurred. Our experiments were driven by four main questions:

1) Does a human's hand motion contain information about an upcoming contact, and can our model use this information to predict contact at a given prediction horizon for new users?
2) Does our model benefit from using a reconstructed hand mesh instead of joint locations?
3) How robust is our model to different levels of hand tracking error and noise?
4) Does our model generalize across objects grasped?

A. GRAB Dataset

We used the GRAB dataset [12], [24] for this project. This dataset consists of sequences of hand meshes over time, as well as tracked object meshes. The person is instructed to pick up an object that is stationary on a table, and then interact with the object in a specified way. The dataset has 10 subjects acting on 51 objects in different ways, including (i) lifting, (ii) handing over, (iii) passing between hands, and (iv) utilizing the item for its purposes. The data is recorded using a Vicon motion tracking system with markers on the hand and on 3D-printed objects. The use of 3D-printed objects as opposed to virtual equivalents results in natural haptic feedback being provided to the user and consequently results in a model based on natural human behavior, which is ideal for our purposes. The hand meshes provided in this dataset are created by fitting Vicon data to the SMPL-X [25] format using a modified MoSh++ [26] algorithm. Each node in the hand mesh consists of a 3-dimensional feature vector x, where $x \in \mathbb{R}^3$ is the 3D location of that point of the hand. The data is provided at 120 Hz. For simplicity, we limit our data to right-handed grasps.

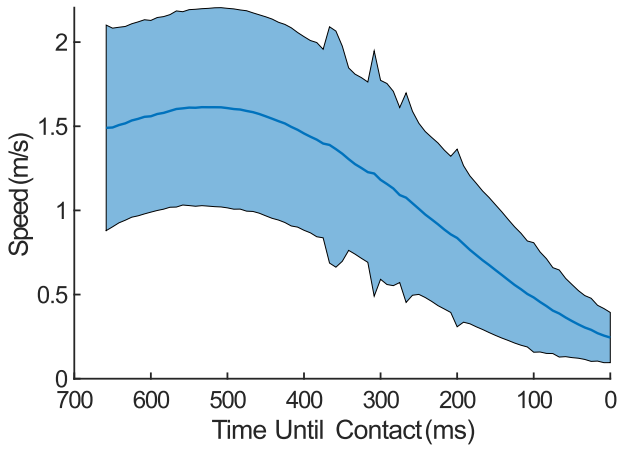

Fig. 3. Average hand speed prior to contact. We calculate hand speed as the speed of the center of mass at the hand. At t=0, the hand is in contact. We show the speed for the 80 timesteps (667 ms) prior to and including contact at 120 Hz. The boundary of the shaded region represents standard deviation.

Fig. 3 shows the average speed of hands in the GRAB dataset prior to contact. There is a large standard deviation of the speed across sequences, and the hand speed slows down approaching contact.

We split the 10 GRAB dataset subjects into five 6-2-2 train-validation-test splits for our by-subject experiments (Sections V-D, V-C). We split the 51 objects into a single 31-10-10 split for our by-object experiments (Section V-E).

B. Baseline

As a baseline, we consider a simple extrapolation of full hand meshes into the future and determine contact time. At time t, for each joint angle j, we extrapolate to j_{t+D} based on j_t and j_{t-1}. To do this, we use the SLERP [27] algorithm for extrapolation. While SLERP is typically for interpolation, the Tensorflow [28] implementation extends to extrapolation.

After determining the new joint angles, we use a SMPL-X [25] model to convert from joints to mesh. After determining the new hand mesh, we extrapolate the hand mesh based on linear velocity as well.

The contact metric for the GRAB dataset is not easily reimplemented (we chose not to, at explicit suggestion of the authors [12]). Therefore for our baseline, we decided contact occurred based on the minimum vertex-vertex distance between the hand and object meshes. We used a threshold of 4.5 mm, as in [12]. We found that when no extrapolation was performed, the average error between our methods was 0.033 timesteps (0.275 mm) – sufficiently small as to not impact our analysis.

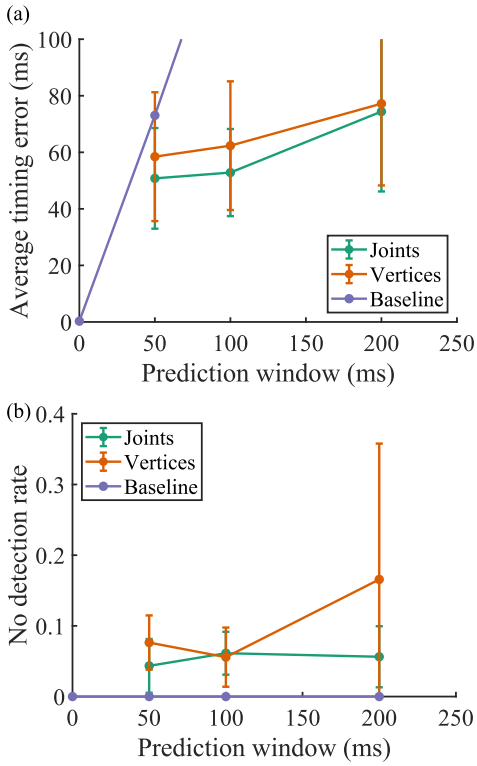

Fig. 4. Error vs. predictive horizon. A. Average timing error across all test sequences when trained on joints or full mesh vertices. Each of the 3 prediction horizons is trained on five different splits of the data, with standard deviation shown on the error bars. We also compare with our baseline method. B. Similarly, the rate at which no contact is detected for each method.

C. Noise-Free Hand-Object Interaction-Prediction Over Multiple Time Horizons

We designed our first series of experiments to answer questions 1) and 2) – whether it is possible to predict contact over future time horizons, and if full hand mesh data offers an advantage over hand joints. We performed the training described in Section IV-F on our network model.

We evaluate on 3 time horizons: 50, 100, and 200 ms (6, 12, 24 timesteps at 120 Hz), and over both full hand meshes and only joint vertices, each with 5 splits of the data. The average prediction error is shown in Fig. 4. We see that the joints-only data does not perform worse – indicating we have no need to incur the computational cost of predicting over the full mesh. The GRAB dataset uses SMPL-X to generate the mesh model using a linear network with the joint locations as input. While the SMPL-X model encodes information across subjects, it is possible this information is fundamentally no more useful than the joint information itself. Therefore, while it is not certain joints only would be better, for our work the GRAB network encoded information is not necessary.

We note that the 'no detection' rate is not monotonic with the length of the prediction window. This is to be expected – if the system is predicting contact time in advance, we cannot guarantee that the system will detect contact before it is made. Therefore, such a prediction has a fundamental tradeoff between accuracy of contact timing and whether it detects a particular contact at all. In practice, the prediction threshold should be tuned based on the particular use case.

We also apply our baseline to each predictive horizon. We see the baseline does worse than our method at each horizon.

D. Effect of Noise

The GRAB dataset was collected using a Vicon motion capture system, leading to high-quality tracking and reconstruction. To simulate a more realistic setting, we artificially induced noise into the dataset and trained with it. We limited these experiments to joints data only based on the success of joint-only data for the noise-free experiments. We use 100 ms as an exemplar time horizon. We add two different types of noise to the system.

First, we added per-joint 3D Gaussian noise. This simulates hand tracking and reconstruction error. We apply Gaussian noise such that the average error would be 10, 30, or 100 mm. We chose these noise values based on the tracking performance of Oculus Quest [13] and the HANDS'19 challenge [7], and added 100 mm as a high noise level for comparison. We calculate the appropriate associated Gaussian standard deviation via the mean of the Chi distribution – for a desired mean error of e, we use a standard deviation of

$$\sigma = e\sqrt{\frac{\pi}{8}}$$

This noise is injected freshly for each joint for each timestep. These results are seen in the solid lines of Figs. 5 and 6.

Second, we added hand-object relative translation noise. To simulate room tracking error of an augmented reality headset, we inject relative translation between the hand and object. We apply Gaussian noise with an average of 5 mm translation error. In contrast to the previous noise, this noise is held constant for all points within a given sequence. This is because we expect room tracking noise error to vary little within the timeframe of a grasp, and to affect all points similarly. We chose this translation based on the mean perceived holographic drift in a Microsoft HoloLens [29]. These results are seen in the dashed lines of Fig. 5 and Fig. 6.

Fig. 5 shows the results when trained and tested on separate subjects, as in the noise-free experiment. Each experiment was run on one split, due to limited available computation. We see only minimal impact of both joint and translation noise on the timing error and no-contact-detection rates. While we do not guarantee our injected noise profile is identical to measured tracking noise, as ours is homoscedastic, we show that detection time can be robust to noise.

At an approach speed of approximately 0.5 m/s at 100 ms prior to contact (Fig. 3), 100 ms represents 5 cm of movement. The noise levels we injected for testing are based on measured noise from the Oculus Quest [13], HANDS'19 challenge [7], and Microsoft HoloLens [29]. The measured noise levels are consistently less than 5 cm, so the small effect of injected noise is expected.

E. Generalization Across Objects

When interaction-expectation is used in a mixed reality system, it needs to extrapolate to subjects for whom the system

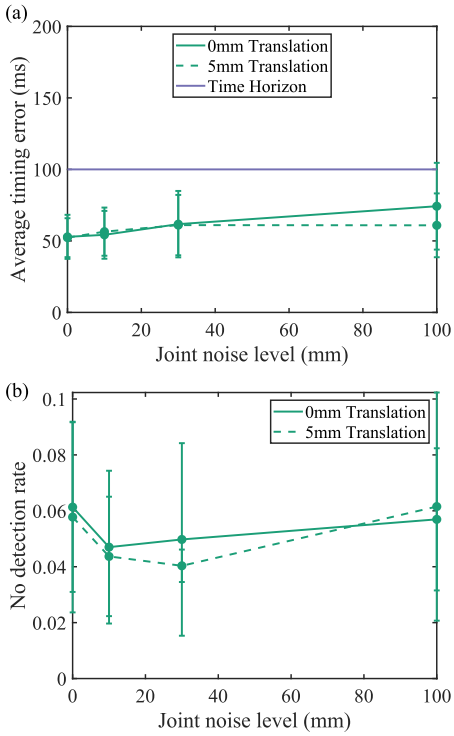

Fig. 5. Error vs. Noise for 5 splits of the data across subjects with 100 ms time horizon. A. Average timing error across all test sequences when trained with various levels of per joint point Gaussian noise. The noise level is the average distance error over all points. These experiments are performed with no translational noise (solid lines) or 5 mm translational noise (dashed lines). Data is for 5 splits across subjects on joint data. B. Rate at which no contact is detected for each noise level is shown.

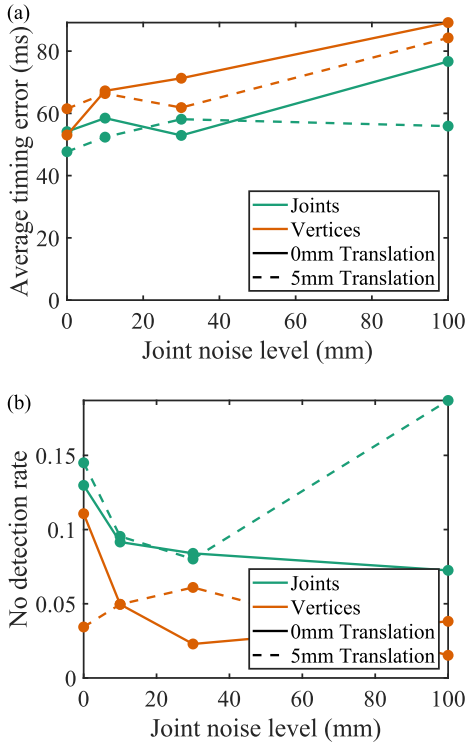

Fig. 6. Error vs. Noise for Objects A. Average timing error across all test sequences when trained with various levels of per joint point Gaussian noise for a split of the data across objects. The noise level is the average distance error over all points. These experiments are performed with no translational noise (solid lines) or 5 mm translational noise (dashed lines). All examples are over a 100 ms time horizon. B. Rate at which no contact is detected for each method.

was not trained. In the previous experiments, we show that error on unseen subjects is low. In addition, it is possible that a user would want to program novel objects into the system. To test the feasibility of this, we ran experiments on a 31-10-10 train-validation-test split over the 51 GRAB objects. The results are shown in Figs. 6 and 7. While the error and 'no detection' rate are higher for untrained objects, the system still extrapolates to novel objects.

When generalizing across objects, neither joints nor vertices are strictly better. The joints have a lower average error, but the 'no detection' rate for vertices is lower. Given the relative difficulty of acquiring full mesh information compared to joint information, for many uses joint information alone would be preferable. In cases where it is critical that any prediction exceeds the contact detection threshold, vertices could be preferable.

Fig. 7 shows the error on the 10 test objects. We see that while there is variation between objects it is low relative to the baselines.

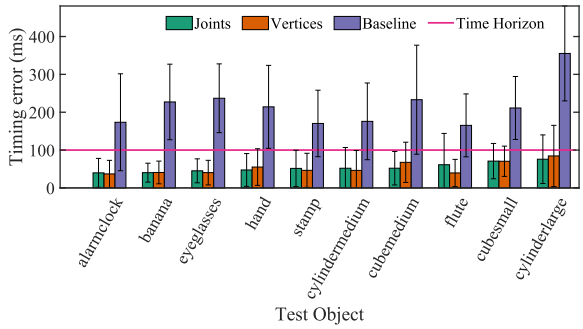

Fig. 7. Results for generalization to new objects. Our model outperformed the baseline consistently. We see some variance across objects. Joint and vertex results are similar.

VI. Conclusions

In this paper, we show that it is possible to predict the timing of future hand-object interaction based on a history of hand poses. Our model takes as input a history of hand and object poses, and predicts upcoming contact for a horizon of 100 ms with an average timing error of 52.8 ms. We demonstrate that our method is robust to noise and can generalize across both subjects and objects. With a latency of 0.36 ms per timestep, our inference time is significantly shorter than our horizon. Our system can be integrated with existing hand tracking methods, to provide predictions on contact timing given a known object. In future work, this method will be evaluated for its ability to provide accurate haptic feedback via a wearable device when integrated into a mixed reality system. We also seek to apply similar methods to predict future hand poses.

ACKNOWLEDGMENT

We thank Yu Sun and Tianze Chen from the University of South Florida for fruitful discussions in early stages of this project. We also thank Omid Taheri from the Max Planck Institute for Intelligent Systems for advice on calculating mesh-mesh intersection timing.

REFERENCES

[1] M. Di Luca *et al.*, "Effects of visual–haptic asynchronies and loading–unloading movements on compliance perception," *Brain Res. Bull.*, vol. 85, no. 5, pp. 245–259, 2011.

[2] W. R. Ferrell, "Delayed force feedback," *Hum. Factors*, vol. 8, no. 5, pp. 449–455, 1966, pMID: 5966936.

[3] D. Wang, K. Tuer, M. Rossi, L. Ni, and J. Shu, "The effect of time delays on tele-haptics," in *Proc. 2nd IEEE Int. Workshop Haptic, Audio Vis. Environ. Their Appl. Proc*., 2003, pp. 7–12.

[4] M. D. Luca and A. Mahnan, "Perceptual limits of visual-haptic simultaneity in virtual reality interactions," in *Proc. IEEE World Haptics Conf.*, 2019, pp. 67–72.

[5] I. M. L. C. Vogels, "Detection of temporal delays in visual-haptic interfaces," *Hum. Factors*, vol. 46, no. 1, pp. 118–134, 2004, pMID: 15151159.

[6] F. Mueller *et al.*, "Ganerated hands for real-time 3D hand tracking from monocular RGB," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 49–59.

[7] A. Armagan *et al.*, "Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction," in *Proc. Eur. Conf. Comput. Vis.*, Cham: Springer, 2020, pp. 85–101.

[8] A. Ng *et al.*, "Designing for low-latency direct-touch input," in *Proc. 25th Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, 2012, pp. 453–464.

[9] E. J. Gonzalez *et al.*, "Reach : Extending the reachability of encountered-type haptics devices through dynamic redirection in VR," *Proc. 33rd Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 236–248.

[10] E. Pezent *et al.*, "Tasbi: Multisensory squeeze and vibrotactile wrist haptics for augmented and virtual reality," in *Proc. IEEE World Haptics Conf.*, 2019, pp. 1–6.

[11] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[12] O. Taheri *et al.*, "GRAB: A dataset of whole-body human grasping of objects," in *Proc. Eur. Conf. Comput. Vis.*, Cham: Springer, 2020, pp. 581–600.

[13] S. Han *et al.*, "MEgATrack: Monochrome egocentric articulated hand-tracking for virtual reality," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 87:1–87:13, Jul. 2020.

[14] S. B. Schorr and A. M. Okamura, "Fingertip tactile devices for virtual object manipulation and exploration," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2017, pp. 3115–3119.

[15] W. McNeely, "Robotic graphics: A new approach to force feedback for virtual reality," in *Proc. IEEE Virtual Reality Annu. Int. Symp.*, 1993, pp. 336–341.

[16] " Haptx: Haptic gloves for virtual reality and robotics," Accessed: Sep. 03, 2021. [Online]. Available: https://haptx.com/

[17] A. Zenner and A. Krüger, "Estimating detection thresholds for desktop-scale hand redirection in virtual reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2019, pp. 47–55.

[18] Y. Horiuchi *et al.*, "Computational foresight: Forecasting human body motion in real-time for reducing delays in interactive system," in *Proc. ACM Int. Conf. Interactive Surfaces Spaces*, New York, NY, USA, 2017, pp. 312–317.

[19] E. Wu and H. Koike, "Futurepose - mixed reality martial arts training using real-time 3D human pose forecasting with a RGB camera," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1384–1392.

[20] M. Wei *et al.*, "History repeats itself: Human motion prediction via motion attention," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 474–489.

[21] T. H. Vu *et al.*, "Smartwatch-based early gesture detection 8 trajectory tracking for interactive gesture-driven applications," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–27, Mar. 2018, pp. 1–27, Art. no. 39.

[22] A. Clarence, J. Knibbe, M. Cordeil, and M. Wybrow, "Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality," in *Proc. IEEE Virtual Reality 3D User Interfaces*, 2021, pp. 150–159.

[23] M. Hahn *et al.*, "3D action recognition and long-term prediction of human motion," in *Computer Vision Systems*, A. Gasteratos *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 23–32.

[24] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "ContactDB: Analyzing and predicting grasp contact via thermal imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8701–8711.

[25] G. Pavlakos *et al.*, "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10967–10977.

[26] N. Mahmood *et al.*, "Amass: Archive of motion capture as surface shapes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5442–5451.

[27] K. Shoemake, "Animating rotation with quaternion curves," in *Proc. 12th Annu. Conf. Comput. Graph. Interactive Techn.*, 1985, pp. 245–254.

[28] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, *software available from tensorflow.org*.

[29] T. Frantz *et al.*, "Augmenting Microsoft's Hololens with vuforia tracking for neuronavigation," *Healthcare Technol. Lett.*, vol. 5, no. 5, pp. 221–225, 2018.