TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision

Paul Streli
Meta Reality Labs
Redmond, WA, USA &
Department of Computer Science
ETH Zürich, Switzerland

Mark Richardson
Meta Reality Labs
Seattle, WA, USA

Fadi Botros
Meta Reality Labs
Redmond, WA, USA

Shugao Ma
Meta Reality Labs
Redmond, WA, USA

Robert Wang
Meta Reality Labs
Redmond, WA, USA

Christian Holz
Department of Computer Science
ETH Zürich, Switzerland

Figure 1: We present a novel method for detecting touch input on surfaces from egocentric hand tracking alone. (a) Here, a user is interacting in mixed reality through touch on a desk, which provides haptic feedback. (b) Our method senses input events from egocentric views—a challenging task, as hand self-occlusion causes sensing uncertainty about exact touch locations. Our learning-based method explicitly models these uncertainties and resolves them as part of a probabilistic framework, accounting for user behavior and context to enable rapid and dexterous text input on a virtual surface keyboard (c).

ABSTRACT

While passive surfaces offer numerous benefits for interaction in mixed reality, reliably detecting touch input solely from head-mounted cameras has been a long-standing challenge. Camera specifics, hand self-occlusion, and rapid movements of both head and fingers introduce considerable uncertainty about the exact location of touch events. Existing methods have thus not been capable of achieving the performance needed for robust interaction.

In this paper, we present a real-time pipeline that detects touch input from all ten fingers on any physical surface, purely based on egocentric hand tracking. Our method *TouchInsight* comprises a neural network to predict the moment of a touch event, the finger making contact, and the touch location. TouchInsight represents locations through a bivariate Gaussian distribution to account for uncertainties due to sensing inaccuracies, which we resolve through contextual priors to accurately infer intended user input.

We first evaluated our method offline and found that it locates input events with a mean error of 6.3 mm, and accurately detects touch events ($F_1 = 0.99$) and identifies the finger used ($F_1 = 0.96$). In an online evaluation, we then demonstrate the effectiveness of our approach for a core application of dexterous touch input: two-handed text entry. In our study, participants typed 37.0 words per minute with an uncorrected error rate of 2.9% on average.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Mixed / augmented reality**; **Text input**.

KEYWORDS

Touch detection, uncertainty estimation, Bayesian inference, text entry, language models, mixed reality, egocentric hand tracking

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0628-8/24/10
https://doi.org/10.1145/3654777.3676330

ACM Reference Format:
Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3654777.3676330

1 INTRODUCTION

Direct interaction has become a preferred form of input on mixed reality (MR) headsets [39]. Users can intuitively manipulate virtual objects and interfaces using their hands and ngers, enabled by advancements in real-time egocentric hand tracking [34, 36]. While MR interaction has so far been designed for mid-air interaction, recent research has demonstrated the bene ts of moving these interactions to surrounding passive surfaces, such as for improved input control and performance [42, 100]. Surfaces provide haptic feedback and an opportunity for rest, and thus make the interaction more comfortable and avoid fatigue during prolonged use [13].

Users are well-acquainted with touch interaction on physical objects or surfaces in the real world. Arguably, the success of smartphones and tablets is due to their focus on this input modality, even for intricate tasks such as text entry [98]. Transferring this mode of interaction to MR systems is di cult, however; following their transition to built-in tracking without the need for stationary cameras, touch input needs to be inferred from camera observations.

Accurately inferring touch events from egocentric views is considerably challenging. Challenges include hand self-occlusion that results in uncertainty about the exact locations of ngers, detecting precise moments of physical contact, and obtaining the corresponding touch input locations. Previous research on vision-based contact recognition has thus far required ngers and touch surfaces to be clearly visible [18, 27, 28]. To enable more dexterous and rapid input, as is needed for typing, additional instrumentation of the hands [92, 94, 119] or external motion capture has been required [78] to achieve practical accuracy for downstream tasks.

In this paper, we present TouchInsight, a method for recognizing touch input on any physical surface for all ten ngers from the egocentric views of moving head-mounted cameras. Our method identi es moments of touch events and estimates their input locations. The key novelty is our explicit model of the uncertainty inherent to the sensing pipeline, in addition to user behavior. This allows our method to robustly infer user intentions even for quick touch input and when hands are self-occluded in egocentric vision. Our method has particular implications for two-handed surface typing, which we use as a challenging task in our online evaluation.

Inferring touch from uncertain input observation

Figure 2 illustrates the problem we address in this paper: uncertainty estimation across a series of error sources for egocentric vision-based touch detection. As a user aims to touch a target with their nger (e.g., to activate a command), resulting endpoint nger locations form a Gaussian distribution [60]. This distribution captures the inaccuracy of touch input due to user behavior (user error), re ecting the speed-accuracy trade-o in the human motor system and the absolute precision of nger touch in general [44, 45]. The second source of uncertainty stems from egocentric hand tracking; due to tracking inaccuracy in any hand-pose estimation [36] and reduced visibility due to self-occlusion (see Figure 1), recovered hand poses can signi cantly di er from the actual hand con guration (sensing error). This introduces sensing uncertainty about the tracked nger endpoints. Consequently, using these hand poses as input to infer touch events introduces irreducible aleatoric uncertainty in the estimated touch locations.

Figure 2: For a single touch input event, the o set between the target and the nger endpoint can be considered as user error. Due to inaccuracies in hand tracking, the endpoint of a tracked nger might additionally deviate from the actual nger location. We refer to this as sensing error, which introduces sensing uncertainty about the touch location.

While `user error' as the rst source of uncertainty has received much attention in previous work on touch sensing [6, 44, 60], the signi cance of `sensing error' as the second source has so far been negligible, as touch sensors have high resolution and precision [61]. However, for camera-based touch estimation, the second error source substantially exceeds the rst in magnitude, particularly as the camera sensor is moving and both its e ective resolution and sampling rate are lower when capturing the user's ngers.

The key novelty of our approach is to explicitly model these two sources of uncertainty through bivariate Gaussian distributions. Our method e ectively integrates sensing uncertainties with user uncertainties in a closed-form expression that allows us to reason about potential user intentions in a probabilistic command prediction framework based on touch locations. Implementing our framework, we present a purely vision-based ten- nger text entry system on a surface-aligned virtual keyboard that runs on a standalone mobile MR headset. Our text entry system fuses the probabilistic touch and uncertainty estimates with per-key touch distributions to obtain a likelihood distribution for the intended character over the vocabulary of the keyboard. We re ne text input predictions through the language priors from a 6-gram character language model and resolve decoding errors through beam search to consider additional priors from a trigram word language model.

We evaluated our method in two folds. In an o ine evaluation, we quanti ed TouchInsight's accuracy in detecting touch events, achieving an F_1-score of 0.99 for event detection and 0.96 for correctly identifying the contacting nger, with a mean temporal latency of less than 70 ms. We also assessed our network's accuracy in predicting location, achieving a mean position error of 6.3 mm while providing meaningful uncertainty estimates. To determine the practical value and e cacy of our method, we conducted an online evaluation for a text entry task where 12 participants transcribed sentences from the Twitter dataset [95]. Participants achieved a mean text entry rate of 37.0 words per minute (WPM) with an uncorrected error rate (UER) of 2.9% after ve phrases of training. Our method thereby signi cantly outperformed index nger-based text entry on a mid-air keyboard (19.7 WPM with 8.0% UER) in terms of input performance, task load, and user preference.

Contributions

In summary, our work makes the following contributions:

- a probabilistic command prediction framework to accurately infer intended user input. Our framework incorporates uncertainties from both user behavior and sensing inaccuracies.
- a novel method to detect touch input on physical surfaces based on egocentric hand tracking. Given the environment, our neural network identi es contact events ($_1$=0.99), the speci c nger involved ($_1$=0.96), the timing of the touch event (<3 ms mean error with an intended o set of 2 frames at 30 Hz), and its location (6.3 mm o set). Importantly, TouchInsight simultaneously estimates the inherent uncertainty of the touch location through a bivariate Gaussian distribution.
- a novel text entry system for ten- nger text input on physical surfaces in mixed reality with passive haptic feedback. Our system runs in real-time and solely takes hand images captured by the Quest 3 cameras from an egocentric perspective as input. In an online user study with 12 participants who transcribed sentences from the Twitter dataset [95] using our text entry system, participants achieved 37.0 WPM with an uncorrected error rate of 2.9%, compared to 19.7 WPM with 8.0% UER in mid-air.

2 RELATED WORK

Our work is related to touch sensing, text entry in mixed reality and virtual reality (VR), as well as statistical keyboard decoding.

2.1 Touch interaction in MR/VR

Vision-based hand pose estimation has made signi cant progress over the last decade, driven by the advancements in learning-based methods [72, 73] and the collection of relevant datasets [31, 52, 122]. Performing hand tracking from an egocentric perspective on MR headsets [34, 36] presents unique challenges. The perspective causes a high degree of self-occlusion for numerous hand poses [71, 102]. State-of-the-art solutions primarily operate on multi-view image sequences captured by wide eld-of-view monochrome cameras [34, 36] to obtain scale information and o er a wide tracking volume. They leverage temporal information to reduce uncertainty in the estimated poses and ensure temporal consistency [34, 36].

Their performance and maturity have ultimately led to the availability of hand pose tracking in today's commercial mixed reality headsets (e.g., Quest 3 [88], Apple Vision Pro [16], Magic Leap 2 [64]). This enables direct interaction with virtual objects and interfaces [3, 59, 100], which, so far, has been predominantly con ned to mid-air [59]. However, research has demonstrated the bene ts of leveraging real-world physical objects for inputs as proxies [12, 40, 43, 89], o ering advantages in terms of passive haptic feedback [12] and increased input control [43]. Aligning interfaces with physical surfaces improves accuracy, task performance, and agency while decreasing physical exertion [13].

Sensing touch on surfaces is challenging. External tracking systems have been used for interactive research purposes [12, 78, 108]. For mobile scenarios, prior work has integrated wearable sensors (e.g., acoustic sensors [87], inertial measurement units (IMUs) attached to ngers [31, 88, 94] or wrists [25, 67, 92]). In addition, vision-based solutions have been proposed that employ additional markers [53] or require active illumination to cast shadows [57, 106], sense vibrations [93] or depth [10, 20, 33, 37, 80, 85, 107, 110, 111].

To estimate contact from monocular images, alterations of the ngernail [11, 66] or object deformations [11] during press events provide visual cues. Alternatively, prior work estimates contact by analyzing the intersections between estimated hand meshes and objects [29, 38], constrained by the requirement for millimeter-level accuracy. Changes in object trajectory and required interaction forces [19, 54, 75] o er insights into hand-object interaction but fail with static environment objects like tables and walls.

Grady et al. [27, 28] proposed a neural network to directly estimate contact regions on surfaces from single RGB images, but their method relies on an external static camera and good visibility of the corresponding ngertips. Dupré et al. [18]'s technique using Oculus Quest Pro hand tracking for index nger input implements a deterministic state-machine to handle tracking inaccuracies but is unsuitable for rapid input events (e.g., typing).

In contrast to prior work, our method, TouchInsight, detects rapid input on passive surfaces from all ten ngers, solely based on egocentric hand tracking, and explicitly models uncertainties about touch locations due to tracking inaccuracies.

2.2 Probabilistic command and text prediction

Unlike physical keyboards, soft keyboards add uncertainty regarding the user's intended key selection due to a lack of con rming haptic feedback, occlusion, and `fat  nger` errors [44, 45]. This a ects user interfaces, requiring keys to be large enough to accommodate noisy user input when processing input deterministically.

Probabilistic input frameworks can o er an alternative solution by accounting for these user uncertainties and continuously inferring a distribution over potential user intentions [9, 103 105], which are taken into account throughout the interaction sequence [81, 82]. Bayes' theorem has become a cornerstone for handling and resolving these uncertainties [7] and found wide application in commercial soft keyboards [26, 99], mitigating key size constraints [98]. The likelihood that the user intends to press a key is determined based on the location of a touch point on a Gaussian distribution tted to previous observations for the respective key [6, 26, 60]. For text input, the likelihood can be combined with a language prior to estimate the probability for a character in a respective location [26]. Based on the previous input history, the prior can be obtained through a language model, often implemented as an n-gram with nite context [97, 99, 118] or more recently as a deep neural network [24, 112]. Zhu et al. [120] propose a technique to extend this concept to general point-and-click command input by obtaining a prior based on previous selection history.

Rogers et al. previously estimated the uncertainty of nger movements due to the low sensor resolution of a capacitive array via a particle lter, which they leveraged for interactive purposes [79]. Prior research further indicates that the display of uncertainties leads to appropriately adapted user behavior [51, 105].

In this paper, we propose a framework that extends probabilistic command prediction using Bayes' theorem with the aleatoric uncertainties about inferred touch locations due to sensing inaccuracies. Thus, TouchInsight accounts for both user and sensing errors, and leverages prior context to more accurately infer intended input. The

uncertainty about the touch location guides the probabilistic framework to focus not only on a single point but consider a broader area depending on the level of uncertainty.

2.3 Text entry systems for mixed reality

Prior work has explored alternative text entry interfaces for MR and VR headsets to enable e cient typing without a physical keyboard.

Mid-air typing via tracked index nger using the built-in cameras of MR/VR headsets has been demonstrated as a viable alternative to controller-based text entry [17, 101] with mean text entry rates between 17.75 WPM [16] and 26.1 WPM [115] and is available on commercial devices including Quest [38]. Markussen et al. [65] and Shen et al. [84] have investigated word-gesture keyboards in mid-air for faster input. ATK enables 10- nger typing mid-air but relies on an external LeapMotion sensor [116]. Yet, mid-air interaction has been shown to lead to fatigue when used over extended periods of time [48]. Eye-based text entry is another alternative but so far is limited by the text entry speeds users can achieve (11 WPM [40]).

To o er passive haptic feedback, researchers have leveraged body surfaces. PinchType [21] allows users to select groups of characters by pinching with the thumb against a respective nger, which is tracked using a marker-based optical system. Related e orts have detected nger touches using additional wearable instrumentation of the hand in various con gurations, such as by appropriating one (12 WPM) [56, 114], both ngertips (23 WPM) [113], or all nger segments [58, 109] as touchpads. STAR [49] proposes a two-thumb text entry interface that leverages the hand as a surface for haptic feedback, tracks the thumbs using the cameras of the Hololens, and detects touches using capacitive tape on the ngertips, achieving a mean text entry rate of 22 WPM.

Similarly, at rigid surfaces like tables can be appropriated as touch interfaces. QwertyRing enables one- nger typing on surfaces sensed by a nger-mounted IMU [32]. Grady et al. [27] investigates index nger-based text input on a table-projected keyboard, with touch events estimated by an external static RGB camera.

TelemetRing [94] and TapStrap [1] equip all ngers with accelerometers to identify individual ngers hitting a surface, enabling chord-based character entry. Zhang et al. [119] employ a ne-tuned transformer model to decode sequences of character groups, input by users through individual nger strokes using TapStrap. Similarly, TapType [92] detects individual ngers hitting the surface via a wristband with integrated IMU sensors. To address the greater uncertainty in nger classi cation, the authors propose a Bayesian neural network to predict a distribution over all ten ngers, which is then used in a probabilistic text decoder.

Prior work has demonstrated that users can transfer skills acquired on a physical keyboard to typing on at surfaces [24], with skilled typists showing spatially consistent key press distributions even during eyes-free touch typing [55, 76, 86, 121]. To decode such surface touch typing, Richardson et al. [78] developed a neural network to directly estimate typed text from keyboard-relative spatial hand-tracking features derived from an external marker-based motion capture system. More recently, Richardson et al. [77] enhanced their method to infer typed text from markerless egocentric hand tracking [36], reporting a typing speed of 42.4 WPM and 7.0% UER without the aid of an additional language model.

Our method, TouchInsight, infers geometrically accurate touch locations that support broader surface-touch interactions in MR/VR. We demonstrate the e ectiveness of our method for on-surface text entry. Instead of directly predicting intended keys from tracked hand motions, we resolve uncertainties using our probabilistic framework, which relies on inferred touch locations and priors from both character- and word-level language models. The estimated uncertainties about touch locations aid in controlling beam search decoding [5], broadening the distribution of location-based key likelihoods across the keyboard in cases of higher uncertainty.

3 METHOD

In the following, we introduce our method TouchInsight for recognizing ten- nger touch input on passive surfaces based on egocentric hand tracking from an MR headset. Our method takes the temporal sequence of tracked hand poses as input and predicts the occurrence of a touch event, the identity of the nger making contact, and a probability distribution capturing the uncertainty about the 2D location of the touch point on the surface. We further demonstrate how to incorporate the predicted touch location distribution into a probabilistic command prediction framework, which we implement in the form of a text entry system that enables ten- nger text input in real time on an MR headset.

3.1 Uncertainty-aware touch estimation

3.1.1 Network architecture. For hand tracking, we rely on a ne-tuned end-to-end trained 3D feature extraction and pose regression network based on UmeTrack [36]. The network takes a variable number of cropped hand images as input, depending on the number of MR headset cameras capturing the hand, and runs at 30 Hz on Quest 3. For each hand, the hand-tracking model's output frames consist of a joint angle vector $\theta \in \mathbb{R}^{20}$ representing the local pose of the hand skeleton, a 6 degrees of freedom (DOFs) global root transformation of the hand T, the user's hand scale s, and a hand con dence score q. From the hand mesh generated for each frame based on θ, T, and s through linear blend skinning (LBS), we derive 3D key points located at the pad, edge, and tip of each nger. These key points capture the nger's direction and volume, which is a compact yet comprehensive representation for identifying touch events, and are normalized relative to a surface-anchored world-coordinate system. For example, in the context of text input, the normalization reference point might be the center of the keyboard between the F and G keys. We stack the three 3-dimensional key points for each nger along with the con dence score q of each hand to form a vector $h \in \mathbb{R}^{92}$. Our touch estimation network then receives the sequence $H_C = (h_{C-g})_{g=0}^{N-1}$, based on the last N output frames from the hand-tracking model as input.

The core of our touch estimation network is built upon an 8-layer TransformerXL model [15], which uses a left-context for self-attention of 4 frames and integrates a convolutional layer with a kernel size of 3 before each attention layer, resulting in an overall context window length of 41 frames ($N = 41$). The output embedding e_C for each frame at C is fed to the touch classi cation module consisting of one linear layer that predicts an 11-dimensional softmax output probability distribution $z_C \in \mathbb{R}^{11}$, indicating whether

Figure 3: Overview of our framework enabling text entry on surfaces in MR. Our touch estimation network receives hand-tracking feature sequences, N_C, as input. The features are derived from hand poses tracked from egocentric views via the headset's cameras and are normalized relative to a coordinate system anchored to a surface-aligned virtual keyboard. The network estimates the occurrence of touch events, the identity of the touching nger z_C, and a bivariate Gaussian distribution for the touch location $?^1x_C|N_C^o$. Our probabilistic text decoder fuses this distribution with $?^1x_8|~_8^o$ capturing the distribution of touch points for intended key presses and a context prior based on input history from a language model $?^1~_8|~_1\bullet"""\bullet_8~1^o$.

any of the 10 ngers of the tracked hands is making contact with the surface or whether it is a blank frame, n.

The touch localization module, consisting of two linear layers, estimates a 2D Gaussian distribution for the touch location relative to the surface-aligned x-y coordinate plane of the reference coordinate system,

$$?^1x_C|N_C^o = ?^1x_C|' \ C^o = N\ x_C|-_{N_C}\bullet\ _{N_C}\ \bullet \tag{1}$$

parameterized by a mean $_{N_C}$ and covariance matrix $_{N_C}$ for every input frame C. This captures the likelihood that the user's touch occurred on a speci c location x_C, given the hand-tracking features $f h_{C\ g}g_{g=0}^{\ 1}$. We use z_C to mask the output of the predicted touch location, which we only consider in case the most likely class output of z_C is not the blank frame, i.e., $\text{argmax}_x z_C < n$. Additionally, if the same nger class is predicted for several consecutive frames, we only consider the rst frame of the sequence. By treating the touch location as a probabilistic distribution, our approach enables the touch estimation network to account for the inherent aleatoric uncertainties due to stochasticity and limitations inherent in the input, considering multiple factors such as occlusion, temporal jitter, and biases in the tracked hand poses.

3.1.2 Touch estimation loss. The loss function for our touch estimation network consists of two terms for the touch classi cation L_z and L_e as well as one term for the localization, L_x,

$$L\ = U_2L_2\ ,\ U_4L_4\ ,\ U_GL_G" \tag{2}$$

Classi cation loss. For recognizing touch events, we employ the Connectionist Temporal Classi cation (CTC) loss [30] to address imbalances related to blank frames and accommodate the sequential nature of touch events,

$$L_2 = \ \log\ \tilde{O}\ \ \tilde{O}\ \ \ \ !\ r_{C\bullet\text{€}}\ \bullet \tag{3}$$
$$c\ 2A\ \ C$$

where A is the set of all possible alignments between the target nger sequence and the output sequence of the touch classi cation module, c represents one possible alignment, and c_C is the label at position C in alignment c with $r_{C\bullet\text{€}}$ being the predicted probability for it. This provides the network with increased temporal exibility in deciding the optimal timing for predicting a touch event. In addition, to ensure that the network maintains latency within acceptable bounds, we add a cross-entropy loss term, which encourages the nger classi cation module to predict the correct nger touch event 3 frames following the ground-truth touch event,

$$L_4 = \tilde{O}\ \ \tilde{O}\ \ l_{C\bullet 9}\log\ r_{C\bullet 9}" \tag{4}$$
$$C2Z\ \ 9$$

Here, $l_{C\bullet 9}$ and $r_{C\bullet 9}$ are the 9-th elements of the ground-truth one-hot vector z_C and the predicted softmax output vector $?r_C$, respectively. The set Z includes all frames where a ground-truth contact event is detected, de ned as $Z = f C|\ \text{argmax}_x z_{1C\ 3^o} < ng$.

Localization loss. For the localization module, we minimize the negative log-likelihood (NLL) of the predicted touch distribution $?^1x_C|N_C^o$ with regard to the actual touch location x_C. To reduce the in uence of predictive variance on the gradient calculations and thereby prevent suboptimal model tting, we apply the V-NLL [83], which weights the contribution for each sample in the loss function by its V-exponentiated variance estimates. This loss has been developed for the univariate Gaussian NLL. For adaptation to our scenario involving bivariate Gaussian NLL, we constrain our covariance matrix, $_C$, to be diagonal. This assumption of an uncorrelated bivariate Gaussian distribution allows the loss function to be expressed as a sum of univariate Gaussian NLL terms,

$$L_G = \tilde{O}\ \ \tilde{O}\ \ l_{C\bullet 9}f\ ^{2V}_{N_C\bullet 9}c\ \ \frac{1}{2}\log\ f^{\ 2}_{N_C\bullet 9}\ ,\ \frac{G_{C\bullet 9}\ `_{N_C\bullet 9}}{2f^{\ 2}_{N_C\bullet 9}}\ \bullet \tag{5}$$
$$C2Z\ \ 9=0$$

where $f_{N_C\bullet 0}$ and $f_{N_C\bullet 1}$ represent the diagonal elements of $_{N_C}$; $`_{N_C\bullet 0}$ and $`_{N_C\bullet 1}$, and $G_{C\bullet 0}$ and $G_{C\bullet 1}$ correspond to the elements of $-_{N_C}$ and x_C. The b csymbol denotes the stop-gradient operator.

3.2 Probabilistic command prediction

In the following, we demonstrate how to integrate the uncertain prediction of a touch location into probabilistic frameworks that adopt Bayes' theorem to infer intended touch selection targets [7]

and commands [120]. While this approach, in principle, extends to other touch sensing modalities and command input tasks, we will demonstrate it for touch estimated from tracked hand poses for text input (see Figure 3).

We consider a sequence of intended commands (in our case, text entry keys or characters), denoted as $\mathbf{a} = a_1, a_2, \ldots, a_M$, of length M. These commands correspond to touch points $\mathbf{x} = x_1, x_2, \ldots, x_M$ and are associated with windows of sensor observations $\mathbf{N} = N_1, N_2, \ldots, N_M$ (in our case, tracked hand poses). Our goal is to identify the most likely sequence of intended characters $\hat{\mathbf{a}} = \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_M$ based on the observed sensor data. We formulate this problem as

$$\begin{aligned}\hat{a}_1, \ldots, \hat{a}_M &= \operatorname*{argmax}_{a_1, \ldots, a_M} P(a_1, \ldots, a_M \mid N_1, \ldots, N_M) \\ &= \operatorname*{argmax}_{a_1, \ldots, a_M} P(a_1, \ldots, a_M)\, P(N_1, \ldots, N_M \mid a_1, \ldots, a_M) \\ &= \operatorname*{argmax}_{a_1, \ldots, a_M} \prod_{m=1}^{M} P(a_m \mid a_1, \ldots, a_{m-1})\, P(N_m \mid a_1, \ldots, a_m)\end{aligned} \tag{6}$$

Assuming that tracked hand poses across typed characters are conditionally independent given their corresponding intended key and that the prior on a character is conditionally independent given the previously typed text, we can rewrite Equation (6) as

$$\hat{a}_1, \ldots, \hat{a}_M = \operatorname*{argmax}_{a_1, \ldots, a_M} \prod_{m=1}^{M} P(N_m \mid a_m)\, P(a_m \mid a_1, \ldots, a_{m-1}) \tag{7}$$

Due to the generally greater complexity of the space of possible tracked hand poses, estimating the likelihood of a speci c hand pose sequence given an intended command is challenging. However, we can leverage the law of total probability to rewrite $P(N_m \mid a_m)$ as

$$\begin{aligned}P(N_m \mid a_m) &= \int P(N_m, x_m \mid a_m)\, dx_m \\ &= \int P(N_m \mid x_m, a_m)\, P(x_m \mid a_m)\, dx_m.\end{aligned} \tag{8}$$

We also make the approximation that touch points x_m are uniformly distributed over the touch surface (i.e., $P(x_m) = 1/22 \cdot 22$ R). Additionally, we consider tracked hand pose sequences N_m to be conditionally independent of the intended character a_m once the touch location x_m is known, simplifying to $P(N_m \mid x_m, a_m) := P(N_m \mid x_m)$. This simpli cation ignores the character-speci c nuances across hand pose sequences (i.e., distinct hand motions for di erent characters). These character-speci c gestures can vary signi cantly across individuals and are more complex to model. By adopting this simpli cation, our model aims to better generalize across users.

We rewrite Equation (7) as

$$\begin{aligned}&\operatorname*{argmax}_{a_1, \ldots, a_M} \prod_{m=1}^{M} P(a_m \mid a_1, \ldots, a_{m-1})\, P(N_m \mid x_m)\, P(x_m \mid a_m) \\ =&\operatorname*{argmax}_{a_1, \ldots, a_M} \prod_{m=1}^{M} P(a_m \mid a_1, \ldots, a_{m-1})\, \frac{P(x_m \mid N_m)\, P(N_m)}{P(x_m)}\, P(x_m \mid a_m) \\ =&\operatorname*{argmax}_{a_1, \ldots, a_M} \prod_{m=1}^{M} \underbrace{P(a_m \mid a_1, \ldots, a_{m-1})}_{\text{context prior}}\, \underbrace{P(x_m \mid N_m)}_{\text{sensing}}\, \underbrace{P(x_m \mid a_m)}_{\text{user}}\end{aligned} \tag{9}$$

since following our assumptions $P(N_m)$ and $P(x_m)$ do not a ect the solution. Equation (9) thus allows us to nd the most likely character sequence based on a context prior $P(a_m \mid a_1, \ldots, a_{m-1})$ based on previous input history and the likelihood of observing a touch location given a considered key. For this likelihood, our model accounts for both the variation in touch location for a given key due to user behavior and the uncertainty in the observed touch location due to the sensing pipeline's limitations. The context prior can be estimated by a language model.

Following prior research on ten- nger typing on touch surfaces [22, 86] and key selection on interactive surfaces [6, 26], we posit that for a given key k, the touch points follow a bivariate Gaussian distribution with a mean μ_k and a full covariance matrix Σ_k,

$$P(x_m \mid a_m = k) = \mathcal{N}(x_m \mid \mu_k, \Sigma_k) \tag{10}$$

This distribution characterizes the touch points resulting from a user's attempt to select a key via a touch on a surface-anchored keyboard, capturing the user's mental model of the keyboard and the spatial error distribution relative to the keys' locations. The likelihood $P(x_m \mid a_m)$ thus relates to the uncertainty about the user's intended key given a touch location. The uncertainty of the touch location due to the sensing or inference pipeline $P(x_m \mid N_m)$ weights the likelihood for a given touch location according to the likelihood that a touch occurred at this speci c location. In our case, this term is estimated by our touch estimation network (see Equation (1)).

Given that we integrate over the product of two bivariate Gaussian distributions (see Equation (1) and (10)), there exists a closed form solution [74],

$$\begin{aligned}\int P(x \mid N) P(x \mid \mu, \Sigma)\, dx &= \int \mathcal{N}(x \mid \mu_N, \Sigma_N)\, \mathcal{N}(x \mid \mu, \Sigma)\, dx \\ &= d_2 \int \mathcal{N}(x \mid \mu_2, \Sigma_2)\, dx \\ &= d_2 \cdot\end{aligned} \tag{11}$$

Here,

$$d_2 = \mathcal{N}(\mu \mid \mu_N, \Sigma_N + \Sigma) = \frac{1}{\sqrt{\det 2\pi (\Sigma_N + \Sigma)}} \exp\left(\frac{1}{2} (\mu - \mu_N)^{\top} (\Sigma_N + \Sigma)^{-1} (\mu - \mu_N)\right), \tag{12}$$

$$\Sigma_2 = \left(\Sigma_N^{-1} + \Sigma^{-1}\right)^{-1} \left(\Sigma_N^{-1} \mu_N + \Sigma^{-1} \mu\right), \tag{13}$$

$$\mu_2 = \left(\Sigma_N^{-1} + \Sigma^{-1}\right)^{-1}, \tag{14}$$

Thus, using Equation (12), we can e ciently compute the combined likelihood factor. Moreover, since these operations are fully di erentiable, the resulting likelihood factor d_2 can be incorporated into another loss term. This term would directly penalize the classi cation of a command, although this application is beyond the scope of our current work and presents an opportunity for future research to include additional motion information.

4 IMPLEMENTATION

We now describe the implementation of our interactive ten- nger text entry system that operates in real-time on a Quest 3 headset. To implement our system, we collected an egocentric touch input

dataset. We then designed, trained, and compiled our touch estimation network, which we deployed with a text decoder following our probabilistic command prediction framework. The decoder estimates a distribution over the keyboard keys and combines it with a language prior from an n-gram model.

4.1 Egocentric touch input dataset

To collect data, we recruited participants who had a baseline level of experience with touch typing on QWERTY keyboards. Participants were given a 60-second physical keyboard typing test in which they had to sustain a 45 WPM entry rate. We collected a dataset from 385 participants, 90.3% of whom were right-handed, 47.4% of whom were male, and with ages ranging from 23 to 38 years old.

Apparatus. During data collection, participants' hands were adorned with 3 mm hemisphere retro-re ective markers, su cient to enable high- delity ground-truth marker-based hand tracking with an OptiTrack motion capture system [35]. Participants wore a headset equipped with four monochrome wide- eld-of-view cameras that captured synchronized VGA images at 30 Hz. Participants would tap their ngers on a synchronized Sensel Morph touchpad [47], which was used to record ground-truth contact points. OptiTrack marker trees were attached to both the Sensel Morph touchpad and the headset so that the touchpad, ground-truth hand tracking, and egocentric hand tracking could all be represented in a common coordinate space. The 3 mm markers used for ground-truth hand tracking are small enough not to be visibly obvious in VGA-resolution monochrome cameras mounted on a headset and thus do not interfere with egocentric markerless hand tracking. To motivate data collection, we tasked participants with a text transcription task. We placed a paper print-out of a QWERTY keyboard layout on top of the Sensel Morph touchpad and presented participants with prompts to transcribe on this touchpad keyboard while we recorded ground-truth 2D contact information.

Procedure. During the recording sessions, participants performed 9 blocks of transcription, with each block consisting of 5 minutes of typing phrases drawn from a single corpus. The corpora comprised short phrases from MacKenzie and Soukore 's phrase set [63], various short text sentences sourced from the internet, and randomly generated sequences of characters. Participants were provided with coarse real-time feedback in the form of a cursor illustrating how much of the prompt had been typed.

Data analysis. In post-hoc, we corresponded the motion-capture-based hand-tracking ngertips with the Sensel Morph touchpad contact events to establish the active nger responsible for each contact. Often, there are multiple proximal contacts, which leads to ambiguity in assigning nger correspondences. To help resolve ambiguity, we applied a bipartite graph matching algorithm to establish assignments that would remain consistent over time.

Additionally, we compute the bivariate Gaussian distribution $?^{1}xj\sim^{o} = N^{1}xj- \sim\bullet \ \sim^{o}$ from the distribution of touch points on the keyboard for each target key, and use part of the dataset to ne-tune our hand-tracking network.

4.2 Touch estimation network

We implemented the network in PyTorch. Using the ne-tuned UmeTrack network, which runs in real-time on the Quest 3, we obtain the hand pose sequence from the egocentric camera footage captured by the headset. The touch estimation network then receives the sequence $\mathbf{H}_C$, which comprises the 3D positions of key points at the pad, tip, and edge of each nger, as input. The coordinates of these key points are relative to an origin set at the center of the keyboard between the F and G keys. To minimize unseen and unrelated inputs, we further zero any key points in the input that fall outside of a bounding box surrounding the keyboard region. After training, we compile the model using the JIT QNNPACK to run on the headset. The network also receives and stores the touch distributions for the keyboard keys. Equation 12 allows for e cient computation of the combined likelihood factor α_2 using tensors.

4.3 Probabilistic text decoder

The text decoder combines the output of the touch estimation network with a language prior estimated by a language model based on the sequence of previously entered characters. We implemented the decoder in C++.

Language model. We deployed an n-gram language model using KenLM [41]. To obtain a prior for each entered character, we employed a 6-gram character language model with a vocabulary including all lowercase characters of the Latin alphabet, the digits 0 to 9, and punctuation marks such as comma `,` and period `.`. In addition, we use a trigram word language model with a vocabulary of 100k English words [96]. We trained the language models on Wikipedia after converting everything to lowercase and removing sentences with characters outside the speci ed character vocabulary.

Beam search. Our text decoder combines the probability distribution over the keyboard keys, as determined by the touch estimation network, with the language priors as the sum of log probabilities. It uses beam search to nd the most likely combination of characters that form a word, operating with a beam width of 20 sequences and considering only word pre xes from the word vocabulary, as structured by a trie. For each complete word, the decoder applies a word prior log probability from the word-level language model. The beam search also accounts for omission and insertion errors through respective penalties [99].

Interaction vocabulary. Users initiate the positioning of the keyboard by activating a virtual calibration button. This action a xes the keyboard to the user's left index nger for ve seconds. Subsequently, they place their left hand at on the surface until the countdown elapses. Through this calibration process, users can position the keyboard to support comfortable interaction.

As users type, the per-character decoded output is displayed in a gray font to ensure visual consistency during fast input. After a short delay, the most likely word from the beam search is presented in a white font below. By pressing space, the suggested word is entered. To select the per-character output allowing for the input of words outside the word vocabulary users can pinch their thumb and middle nger together. After deleting a character using backspace, the autocorrection is deactivated for the current word.

Table 1: Comparison of model training objectives across several metrics of interest, including mean and standard deviation of touch position error [mm], negative log-likelihood, precision, recall, and F$_1$-score for identifying the occurrence of a touch event and the speci c nger involved, temporal o set [ms], CoER, as well as ChER using greedy or beam search decoding, factoring in (with uncertainty, w.u.) or ignoring (without uncertainty, wo. u.) estimated touch location uncertainty.

objective	#	mean pos.#	std. pos.#	NLL#	touch classi cation			nger classi cation			temp. o .	#	CoER#	Greedy ChER#		Beam ChER#	
					Prec.":	Rec.":	F1"	Prec.":	Rec.":	F1"				w. u.	wo. u.	w. u.	wo. u.
MSE	-	8.22	9.94	-	.992	.990	.991	.984	.953	.963	-1.11		31.07%	-	20.72%	-	14.84%
pad proj.	-	15.24	8.42	-	.992	.990	.991	.984	.953	.964	5.47		64.01%	-	61.02%	-	69.78%
tip proj.	-	10.60	7.79	-	.991	.991	.991	.984	.953	.963	0.61		46.51%	-	42.82%	-	51.29%
#-NLL	0.0	11.29	10.99	3.96	.992	.989	.991	.984	.954	.964	0.56		41.67%	29.15%	30.04%	16.99%	24.71%
#-NLL	0.5	8.08	11.66	3.22	.992	.990	.991	.985	.953	.963	6.05		29.32%	18.77%	20.35%	10.83%	14.82%
#-NLL	0.8	7.49	11.65	3.13	.991	.990	.991	.984	.952	.963	-0.26		27.51%	16.42%	17.70%	9.67%	12.83%
#-NLL	0.9	6.30	9.24	3.16	.992	.990	.991	.983	.952	.962	2.92		25.12%	14.77%	15.78%	8.49%	11.07%
#-NLL	1.0	6.39	9.93	3.30	.990	.990	.990	.984	.952	.963	-0.28		25.30%	15.29%	8.16%	10.03%	10.79%

5 EVALUATION 1: TOUCH ESTIMATION

Data preparation and network training. We evaluated the accuracy of inferring touch points based on egocentrically tracked hand poses using our touch input dataset. We trained the network on a subset of the dataset consisting of 376 participants for a xed number of 160 epochs, using a batch size of 64 and the Adam optimizer [50]. The training dataset was composed of 5.29 M unique contact events. We set $\omega_2 = 1$, $\omega_4 = 0.01$, and $\omega_G = 0.001$. For the cross-entropy loss term, we set $\delta = 2$ frames. We then evaluated the trained network on a held-out test set with 9 participants, who performed 17 K touches per person on average. The touches were distributed as follows across ngers: 5.12%/5.48% with the left/right pinky, 7.23%/11.69% with the left/right ring nger, 11.51%/12.46% with the left/right middle nger, 16.26% / 16.82% with the left/right index nger, 4.70%/8.74% with the left/right thumb.

Alternative implementations. We evaluated the performance of our touch localization using di erent values for δ. In addition, we experimented with alternative implementations and loss functions. We assessed the accuracy of touch location estimates derived from the surface-projected tip or pad key points of ngers classi ed as touching. Moreover, we compared the NLL with a standard mean squared error loss (MSE).

Metrics. We assessed the performance of our touch localization in terms of the mean and standard deviation of position error. This error is de ned by the average distance o set between the predicted mean of a touch location and the actual touch location, as captured by the Sensel Morph touchpad. To evaluate the calibration of the estimated uncertainty, we determined the negative log-likelihood (NLL) of the predicted distribution $\mathcal{N}(x_C | \mu_C, \Sigma_C)$, which is a lower bound for the evidence [39]. Additionally, we evaluated the accuracy of our touch classi cation. Speci cally, we assessed accuracy across eleven classes: one for each of the ten ngers and a no-touch class to account for missed and ghost touches (i.e., no ground-truth or predicted touch). Because we supervised with the CTC loss function and the model was able to learn a variable emission latency, we rst aligned predicted and actual touch events in a manner similar to how the Levenshtein edit distance aligns text [62] by treating the sequence of nger identities like a string of characters. We considered predictions made more than 5 frames before or 15 frames after the actual touch as incorrect. We reported the precision, recall, and F$_1$-score for determining whether a touch occurred (i.e., di erentiating between no-touch and touch). For correctly detected touches, we also calculated the precision, recall, and F$_1$-score for accurately identifying the nger that makes contact. We also analyzed the temporal o set between a predicted touch event and the actual contact on the synchronized Sensel Morph touchpad. Since the temporal granularity of our touch estimation was bound by the headset's hand-tracking frame rate, we reported the mean frame o set relative to the target o set of 2 frames ($\delta = 2$ in Equation (4)) multiplied by the hand tracking update period (i.e., 1000/30 ms). Furthermore, we provided an assessment of the relative error rate between the ground-truth contact key (i.e., the key whose boundaries encompass the contact point as measured by the Sensel Morph touchpad) and the closest key based on the predicted mean touch location, which we termed the contact key error rate (CoER).

Table 2: Comparison of mean touch position error across di erent training objectives and ngers (LT: left thumb, LI: left index nger, LM: left middle nger, LR: left ring nger, LP: left pinky, RT: right thumb, RI: right index nger, RM: right middle nger, RR: right ring nger, RP: right pinky).

objective	#	LT	LI	LM	LR	LP	RT	RI	RM	RR	RP
MSE	-	9.25	6.15	7.54	11.26	14.54	8.21	6.69	8.49	9.45	10.91
pad proj.	-	13.77	18.05	13.17	11.08	10.18	10.34	19.20	14.24	16.08	23.86
tip proj.	-	6.22	11.45	7.36	9.65	8.95	7.90	13.39	9.52	12.62	20.31
#-NLL	0.9	6.07	4.87	5.98	9.06	11.35	6.40	5.41	6.22	6.70	7.77

Finally, we evaluated our text decoder in an o ine simulation using the touch events captured during the transcription of the MacKenzie and Soukore 's phrase set. Participants entered these phrases at a mean rate of 84.32 WPM on the Sensel Morph touchpad. We calculated the character error rate (ChER[1]) as the Levenshtein distance between the decoded and target text, divided by the length of the target string. We analyzed the performance of our text decoder under two decoding strategies: *greedy* per-character decoding, which used priors from the character-level language model, and *beam* search decoding, which incorporated priors from both the character- and word-level language models. For each decoding strategy, we calculated ChER in two scenarios: one considering the uncertainty about the estimated touch location, if available, and another treating the touch locations as deterministic (speci cally, by setting the covariance matrix Σ_N in Equation (12) to zero).

[1]Previous work uses the abbreviation CER [92, 98, 99], which is easily confused with corrected error rate.

Results. The results are presented in Table 1. The mean position error ranged from 6.30 mm to 15.24 mm, with the lowest error observed for the $\mathcal{V}$-NLL method with $\mathcal{V} = 0.9$. NLL values ranged from 3.13 to 3.96, with the lowest NLL observed for the NLL method with $\mathcal{V} = 0.8$. Across various methods and $\mathcal{V}$ values, touch event, and finger identity classification accuracy remained consistently high, with F_1-scores exceeding 0.99 and 0.96, respectively. The temporal offset, which measures the difference between the target offset of 2 frames for predicted touch events and the actual moment of contact on the touchpad, varied from -1.11 ms (early detection) to 6.05 ms (late detection). These deviations were generally less than the update period for a single frame, showcasing the high temporal precision of our trained touch estimation network.

We also reported the mean position error across different fingers in Table 2. Except for the left pinky finger, the method using $\mathcal{V}$-NLL with $\mathcal{V} = 0.9$ achieved the lowest position error for all fingers compared to alternative methods. The position errors for NLL were the lowest for the index finger (right: 5.41 mm, left: 4.87 mm), followed by the middle (right: 6.22 mm, left: 5.98 mm) and thumb (right: 6.40 mm, left: 6.07 mm), ring (right: 6.70 mm, left: 9.06 mm), and pinky (right: 7.77 mm, left: 11.35 mm) fingers. We observed a similar trend for the MSE objective but generally higher errors compared to $\mathcal{V}$-NLL. The method that directly inferred the touch location based on the surface-projected tip key point of the tracked hand mesh outperformed its equivalent based on the pad key point, but on average resulted in a greater position error than the model that learned to predict the touch location using the NLL objective.

The impact of touch location accuracy is evident for the downstream task of text entry. The contact key error rate was lowest at 25.30% for the $\mathcal{V}$-NLL objective with $\mathcal{V} = 0.9$, and highest at 64.01% for the method utilizing pad key points. With a greedy decoding strategy, ChER ranged from 61.02% to 14.77%, achieving its lowest with the $\mathcal{V}$-NLL objective at $\mathcal{V} = 0.9$ when factoring in location uncertainty, which improved decoding results for all uncertainty-aware models (i.e., using $\mathcal{V}$-NLL). Beam search decoding further reduced the ChER to 8.49% for the NLL method with $\mathcal{V} = 0.9$.

Discussion. The $\mathcal{V}$-NLL objective with $\mathcal{V} = 0.9$ achieved superior performance in terms of position error compared to other objectives, even outperforming the MSE model, while providing uncertainty estimates with comparably low NLL. The model also reliably detected touch events with a minimal deviation (< 3 ms) from the target latency of 2 frames (67 ms) and an F_1-score of 0.991. This touch event accuracy is ultimately critical for the reliable recognition of input events during typing. Additionally, the model accurately classified the touching finger with a macro-averaged F_1-score of 0.963, a recall of 0.952, and a precision of 0.984.

The results further support a learning-based approach for estimating touch location. Naïve touch localization based on specific hand key points likely struggles due to variations in finger angles during key hits and limited hand-tracking accuracy.

The higher position errors for the ring and pinky fingers are likely due to increased joint occlusions from an egocentric view, as shown in Figure 1. Our network trained with $\mathcal{V}$-NLL accounts for this via its estimated uncertainties. For example, for the model with $\mathcal{V} = 0.9$, the touch location covariance matrices Σ_N, associated higher mean uncertainties with touches by the left pinky finger ($\bar{f}_{N=0} = 11.6$) and $\bar{f}_{N=1} = 5.6$ which correlated with a higher mean position error of 9.9 mm in the x-axis and 3.8 mm in the y-axis. In contrast, for touches with the left index finger, uncertainties were $\bar{f}_{N=0} = 4.6$ and $\bar{f}_{N=1} = 4.1$, corresponding to lower mean errors of 3.1 mm on both axes.

Moreover, although the use of a diagonal covariance matrix for $-\log(\mathcal{N}(x; \mu, \Sigma))$ (see Equation (5)) constrains the network's expressiveness regarding the uncertainty about the touch location, we found that the error distribution between the predicted mean and the ground-truth touch location correlated only weakly across axes ($\rho = 0.15$).

Our evaluation also highlighted the challenges of decoding intended text based solely on estimated touch locations. With position errors of around 6 mm only one-third of an average finger's width we observed an error rate exceeding 25% in accurately identifying the ground-truth contact key from the predicted touch location. Moreover, deviations between the intended character and the contact key, due to user errors, further complicate inference.

Our probabilistic framework benefits from incorporating language priors to infer the target phrases. Beam search decoding further reduced character error rates compared to greedy per-character decoding. Our evaluation also showed that the uncertainty about the estimated touch location enhances decoding performance. The objective function with $\mathcal{V}$-NLL thus not only improves the accuracy of the mean touch location likely by avoiding overconfident predictions during periods of higher uncertainty but the uncertainties also effectively guide the text decoder.

6 EVALUATION 2: ONLINE TEXT ENTRY

Users adjust their behavior based on the feedback they receive from an interactive system. For instance, to minimize typing errors, users may choose to type more slowly [3, 5]. This complicates testing changes to our probabilistic text decoder through offline simulation alone. Thus, to demonstrate and evaluate our probabilistic framework end-to-end, we conducted an online text entry experiment in which participants transcribed phrases solely using egocentric hand tracking from the Quest 3. We considered three different conditions: 1) **Midair**: state-of-the-art baseline condition where participants entered text using mid-air poke typing with their index fingers, commonly used in commercial products and research [17, 21], 2) **GreedySurf**: condition leveraging our probabilistic on-surface text entry system using greedy per-character decoding (with priors from the character-level language model), 3) **BeamSurf**: condition using our probabilistic on-surface text entry system with per-word beam search decoding (with priors from the character- and word-level language models). For GreedySurf and BeamSurf, we used the touch estimation network with $\mathcal{V} = 0.9$ trained on our touch input dataset (see Section 4.1).

6.1 Study design

Apparatus. For the study, participants wore a Quest 3 headset. Participants were seated in front of an empty desk. For the study setup (see Figure 4), we implemented a virtual reality environment in Unity, featuring a virtual keyboard and a text entry field with the intended target phrase shown above. For Midair, we implemented the keyboard following the Virtual Keyboard API, which is part of the Meta XR Core SDK [70]. For GreedySurf and BeamSurf, we

Figure 4: The gure shows the apparatus for our online text entry evaluation (a) as well as the evaluation interface for Midair (b), GreedySurf (c), and BeamSurf (d).

implemented an on-surface touch keyboard following a standard QWERTY layout matching the overlay from our touch input dataset capture. While the study was running on the headset, a moderator ensured that participants conformed to the instructions through a screenshare streamed to an external computer.

Participants. We recruited 12 participants (5 female, 7 male, ages 23 37, mean=26.7). On a 7-point Likert scale, all participants rated themselves 4 or higher for I would consider myself a fast typer (mean=5.5), and 4 and higher for I consider myself a uent English speaker (mean=6.3). The self-ratings for I would consider myself a touch typist ranged from 2 to 7 (mean=5.25, median=5.5), with two participants rating themselves below 4. Participants self-reported their prior experience with VR technology on a 5-point Likert scale (from 1 never to 5 almost every day). Participants' ratings ranged between 2 and 4, and their median prior experience was 3.

Task. During each trial, participants' task was to transcribe target phrases as quickly and accurately as possible.

Procedure. The study started with a brief introduction to the task. Participants then completed a questionnaire, providing demographic information and self-rating their English and typing skills. As a reference, they then transcribed 10 phrases on a physical keyboard using the TextTest++ tool, ignoring case [117]. Participants then conducted the evaluation for the three di erent conditions: Midair , GreedySurf , and BeamSurf. The order of the conditions was counterbalanced across participants. For each condition, participants rst transcribed ve phrases as training, also practicing the use of selecting suggestions and deleting erroneous input. They then completed three blocks in which they each transcribed 10 phrases from the TwitterIV phraseset, followed by another block with 10 phrases from the TwitterOOV phraseset [98]. The 100k English word vocabulary of our decoder included all words from the TwitterIV phrase set. Each phrase from the TwitterOOV phraseset included one out-of-vocabulary (OOV) word. The phrases were randomly selected, ensuring that no sentence appeared twice in the study or had been used in the data collection and that they were counterbalanced across conditions. After each condition, participants completed a NASA Task Load Index (NASA-TLX) and rated their subjective text entry speed (I entered text quickly from 1 strongly disagree to 7 strongly agree) and accuracy (I entered text accurately from 1 strongly disagree to 7 strongly agree) on a 7-point Likert scale. Between blocks, participants took 1-minute breaks, and between conditions, they took 5-minute breaks. At the end of the study, participants selected their preferred text entry method and provided qualitative feedback.

6.2 Results

To assess text entry performance, we analyzed text entry rate and accuracy. We reported the text entry rate in words per minute (WPM), determined by the time di erence between the rst and last keystroke for a phrase following the de nition of MacKenzie [61]. For accuracy, we computed the uncorrected error rate (UER) and corrected error rate (CER) following TextTest++ [90, 117], in addition to ChER (see Section 5). ChER is close to UER but ignores corrected characters in its computation.

We reported the mean performance across participants for all conditions and blocks in Table 3. Figure 5 shows the performance in terms of entry speed and error rates (UER, CER, ChER) across participants. As a reference, on the physical keyboard, participants entered sentences with a mean speed of 67.50 WPM (min=43.76, max=95.36, SD=15.88), a UER of 0.78% (min=0.0, max=3.09, SD=1.01), a CER of 6.35% (min=1.08, max=12.38, SD=3.26) and a ChER of 0.8% (min=0.0, max=3.37, SD=1.08).

For signi cance testing, we considered the participant as a random factor, and the text entry condition and block as within-subject factors. We performed a two-factor Aligned Rank Transform (ART) ANOVA for WPM, UER, and ChER as the assumptions for normality and homogeneity were not satis ed. Post-hoc pairwise comparisons were performed using the ART-C algorithm with Bonferroni-adjusted p-values. For CER, we performed a standard two-way repeated measures ANOVA with Greenhouse-Geisser correction as the data was normally distributed (all Shapiro-Wilk $p > .05$) and the assumption on equal variance between groups was satis ed (all Levene's $p > .05$). Post-hoc pairwise comparisons were performed using Bonferroni correction.

6.2.1 Entry rate. We found a main e ect of the condition on WPM, but no signi cant di erence for blocks and no signi cant interaction e ects. Both on-surface conditions, GreedySurf and BeamSurf, allowed participants to enter text signi cantly more quickly than in Midair . For Midair , the mean text entry rate was 19.26 WPM (SD=3.60) in Block 1, 19.36 WPM (SD=4.51) in Block 2, 20.44 WPM (SD=4.17) in Block 3, and 20.92 WPM (SD=5.03) in Block OOV with out-of-vocabulary words. In the BeamSurf condition, participants' mean text entry rate was 36.31 WPM (SD=14.05) in Block 1, 37.32 WPM (SD=14.57) in Block 2, 37.50 WPM (SD=11.07) in Block 3,

Table 3: Results of the online text entry evaluation. We reported the mean text entry rate (WPM) and error rates (UER %, CER %, ChER %) across conditions (Midair , GreedySurf , BeamSurf) and blocks (Block 1 , Block 2 , Block 3 , Block OOV). Signi cant pairs for post-hoc comparisons using Bonferroni correction are shown if ? Ÿ "05: +? Ÿ "05, *? Ÿ "01, **? Ÿ "001, ***? Ÿ "0001

	Block 1				Block 2				Block 3				Block OOV			
Condition	WPM"	UER#	CER#	ChER#	WPM"	UER#	CER#	ChER#	WPM"	UER#	CER#	ChER#	WPM"	UER#	CER#	ChER#
Midair	19"26	8"54	9.86	9"73	19"36	8"01	11.73	9"32	20"44	7"46	11.21	8"55	20"92	7"22	9.49	8"09
GreedySurf	35"92	7"21	16"43	8"53	34"50	7"56	19"66	9"19	37.54	6"44	17"13	7"83	32.74	6"50	20"83	8"09
BeamSurf	36.31	2.65	18"14	3.34	37.32	2.83	18"06	3.62	37"50	3.08	17"42	3.81	28"27	3.50	22"70	4.73

WPM condition[$_{2,132}$=72"78, ?Ÿ "0001]: Midair Ÿ GreedySurf, Midair Ÿ BeamSurf

UER condition[$_{2,132}$=11"48, ?Ÿ "0001]: Midair ¡ BeamSurf, GreedySurf ¡ BeamSurf

CER condition[$_{2,22}$=10"90, ?Ÿ "01, $_{?}^{2}$="25]: GreedySurf ¡ Midair , BeamSurf ¡ Midair ; CER block[$_{3,33}$=3"82, ?Ÿ "05, $_{?}^{2}$="03]: Block OOV ¡ Block 3

ChER condition[$_{2,132}$=12"09, ?Ÿ "0001]: Midair ¡ BeamSurf, GreedySurf ¡ BeamSurf

Figure 5: Boxplot of the mean text entry rate in WPM and error rates (UER, ChER, CER) across participants for Midair , GreedySurf , and BeamSurf , and Block 1 , Block 2 , Block 3 , and Block OOV .

and 28.27 WPM (SD=1.4) in Block OOV, while in the GreedySurf condition mean entry rate was 35.92 WPM (SD=9.73) in Block 1, 34.50 WPM (SD=7.26) in Block 2, 37.54 WPM (SD=7.55) in Block 3, and 32.74 WPM (SD=8.79) in Block OOV. We did not nd any signi cant di erence between GreedySurf and BeamSurf (? ¡ "1).

6.2.2 Error rate. There is a signi cant main e ect of the condition on UER, but no interaction e ects and main e ect for blocks. BeamSurf had signi cantly lower UERs compared to GreedySurf and Midair , with no signi cant di erence between Midair and GreedySurf. We made the same observations for ChER with a signi cant main e ect for condition, and a signi cantly lower ChER for BeamSurf compared to GreedySurf and Midair .

For Midair , participants achieved a mean UER of 8.54% (SD=6.98) and ChER of 9.73% (SD=7.72) in Block 1, 8.01% UER (SD=7.79) and 9.32% ChER (SD=9.10) in Block 2, 7.46% UER (SD=9.95) and 8.55% ChER (SD=11.2) in Block 3, and 7.22% UER (SD=10.76) and 8.09% ChER (SD=11.76) for Block OOV. UER and ChER for GreedySurf were 7.21% (SD=5.23) and 8.53% (SD=5.38) in Block 1, 7.56% (SD=5.95) and 9.19% (SD=5.93) in Block 2, 6.44% (SD=3.91) and 7.83% (SD=4.05) in Block 3, and 6.50% (SD=3.59) and 8.09% (SD=3.82) in Block OOV. The autocorrection in BeamSurf led to signi cantly fewer uncorrected errors with a mean UER of 2.65% (SD=1.27) in Block 1, 2.83% (SD=1.21) in Block 2, 3.08% (SD=1.56) in Block 3, and 3.50% (SD=1.66) in Block OOV. Similarly, mean ChER was 3.34% (SD=1.56) in Block 1, 3.62% (SD=1.50) in Block 2, 3.81% (SD=1.84) in Block 3, and 4.73% (SD=2.26) in Block OOV.

Participants had to backspace to correct mistyped characters, requiring the deletion of correctly entered characters to correct errors at the beginning of a word and causing relatively large CER. We found a main e ect of condition and block on CER, but no interaction e ects. Pairwise comparisons revealed signi cant differences between Block 3 and Block OOV as well as between both GreedySurf and BeamSurf compared to Midair .

Due to the slower entry rate, participants were likely more hesitant to perform and retype mistyped text, leading to a lower mean CER of 9.86% (SD=5.69) in Block 1, 11.73% (SD=4.85) in Block 2, 11.21% (SD=4.90) in Block 3, and 9.49% (SD=4.37) in Block OOV compared to GreedySurf with 16.43% (SD=8.36) in Block 1, 19.66% (SD=9.01) in Block 2, 17.13% (SD=6.90) in Block 3, and 20.83% (SD=8.30) in Block OOV and BeamSurf with 18.14% (SD=7.43) in Block 1, 18.06% (SD=8.11) in Block 2, 17.42% (SD=6.98) in Block 3, and 22.7% (SD=7.05) in Block OOV.

6.3 Perceived performance and workload

We tested for signi cant di erences between conditions regarding perceived speed, accuracy, mental task load, physical task load, temporal task load, performance, e ort, frustration, and overall task load according to the raw NASA-TLX survey (scale from 1 100). After checking for normality and homogeneity of variances, we performed a one-way ANOVA with post-hoc pairwise comparisons using Bonferroni correction for accuracy, mental task load, physical task load, temporal task load, performance, e ort, frustration, and overall task load. For perceived speed, we conducted a one-way

Table 4: Results from subjective ratings on speed and accuracy, and NASA-TLX task load survey for the three conditions (Midair (M), GreedySurf (G), BeamSurf (B)) of the online text entry study. Statistical test details have been omitted for clarity. Signi cances for post-hoc pairwise comparisons using Bonferroni correction are indicated if ? Ÿ "05: +? Ÿ "05, *? Ÿ "01, **? Ÿ "001.

Condition	Speed"	Accuracy"	Mental#	Physical#	Temporal#	Performance#	E ort #	Frustration#	Overall#
Midair	2"17(1.34)	3"58(1.51)	59"92(27.20)	67"06(24.15)	52"78(22.92)	58"33(21.40)	68"25(21.23)	61"11(25.83)	61"24(18.33)
GreedySurf	4"33(0.98)	3"42(1.24)	50"00(23.37)	36"90(20.42)	44"84(20.45)	50"79(16.16)	48"02(17.64)	46"43(22.35)	46"16(14.77)
BeamSurf	4.50 (1.38)	4.50 (1.57)	48.41(23.84)	33.33(21.58)	43.25(19.94)	37.30(20.29)	47.62(22.79)	42.86(20.41)	42.13(15.81)
Signi cant pairs	MŸB, MŸG	n/a	M¡ B, M¡ G	M¡ B, M¡ G	n/a	n/a	M¡ B, M¡ G	n/a	M¡ B, M¡ G

ART ANOVA with post-hoc pairwise comparisons using the ART-C algorithm with Bonferroni-adjusted p-values. The mean and standard deviations across participants are reported in Table 4.

6.3.1 Perceived speed and accuracy. Participants perceived themselves to be signi cantly faster with GreedySurf and BeamSurf compared to Midair . Participants' mean for speed was 4.50 (SD=1.38) for BeamSurf, 4.33 (SD=0.98) for GreedySurf , and 2.17 (SD=1.34) for Midair . We could not nd a signi cant di erence between BeamSurf and GreedySurf , which is in alignment with measured WPM (see Section 6.2.1). Even though we observed signi cantly smaller UER for BeamSurf (see Section 6.2.2), participants did not perceive a signi cant di erence in terms of accuracy between conditions. Participants rated their accuracy on average with 4.50 (SD=1.57) for BeamSurf, 3.42 (SD=1.24) for GreedySurf , and 3.58 (SD=1.51) for Midair .

6.3.2 (Raw) NASA-TLX. Mental task load, physical task load, effort and overall task load were signi cantly lower for BeamSurf compared to Midair . Similarly, mental task load, physical task load, e ort, and overall task load were signi cantly lower for GreedySurf compared to Midair . We did not nd signi cant di erences in temporal task load and frustration, nor in performance, according to the pairwise post-hoc tests between conditions after adjusting p-values using Bonferroni (all ? ¡ 0"05).

6.3.3 alitative feedback. Out of 12 participants, 9 participants had a preference for BeamSurf, 2 for GreedySurf , and 1 for Midair . When asked about their experience, participant 9 stated that they found Midair to be exhausting and they were missing haptic feedback . Participant 3 explained their preference for Midair due to a higher level of control, even though it involved trading o speed. Participant 7 found typing with autocorrection in BeamSurf much easier , and participant 6 speci ed that ten- nger typing on a surface was very intuitive, requiring them not to look at their hands while typing, similar to on a physical keyboard. Participant 1 was further impressed by the speed at which they could enter text on the MR headset. There were also suggestions for improvements. Participant 8 would have preferred the autocorrect for a word to remain active even after pressing backspace.

6.4 Discussion

Our ndings indicate that both on-surface conditions outperform mid-air input in terms of text entry rate, which is also re ected in participants' subjective ratings on input speed. The mean text entry rate of 37 WPM for BeamSurf also outperforms most MR/VR text entry methods from prior work [49], in particular approaches based on head-mounted cameras (e.g., compare Yi. et al.'s 26.1 WPM [115]). This is likely due to the passive haptic feedback, which provides users with more tangible cues compared to visual feedback alone.

The results also show the bene ts of our probabilistic framework, foremost enabling text entry for both on-surface conditions based on tracked hand poses. BeamSurf 's autocorrection leveraged ambiguous input and uncertainties over the entire word, which led to signi cantly smaller uncorrected error rates (< 3%). These rates approach results reported for fast text input on smartphones [99].

Corrected error rates of GreedySurf and BeamSurf indicate room for improvement in terms of accuracy, as well as the need for more e cient correction techniques that do not require the deletion of correctly decoded input.

While we did not nd a signi cant di erence between text entry rates and uncorrected error rates across blocks, the mean entry rates for BeamSurf are notably lower on phrases with OOV words (28.27 WPM), likely because participants started to trust the autocorrection and had to perform costly corrections in case of mispredictions. Future iterations should focus on more versatile language models and the option for overwriting the autocorrection post-hoc.

The overall task load, according to the NASA-TLX survey, was signi cantly smaller for BeamSurf and GreedySurf than for Midair . We attribute this to signi cantly smaller physical demands in both conditions. Even though participants could support their elbows on the desk during mid-air input, additionally resting their wrists on the surface allowed for even more comfortable input, which is in line with previous ndings [13].

The top-performing participant achieved a mean text entry rate of 73.6 WPM with a UER of 1.5% and a CER of 2.6% in block 2 using BeamSurf. This highlights the potential of our text entry system to support fast text input for expert users.

However, we also acknowledge that there remains a gap between reference text entry rates on a physical keyboard compared to BeamSurf . Participants entered text 45% slower on average (67.5 WPM vs. 37.0 WPM), indicating the need for further improvements to fully substitute physical keyboards in MR/VR environments.

7 LIMITATIONS AND FUTURE WORK

While our evaluation provided support for our approach and conrmed the promising performance of ten- nger touch sensing from egocentric vision using our method, several limitations motivate interesting avenues for future work.

Constrained Character Set. Our method was evaluated using a keyboard with a limited set of keys, speci cally excluding capitalization and function keys, such as shift. However, our method

estimates touch locations in a manner that is agnostic to speci c key locations and, thus, in principle, is applicable to the entire keyboard space. Future work should focus on expanding our approach to encompass a broader set of keys, either through the acquisition of more comprehensive touch data or by experimenting with default distributions for unseen command elds.

Stateful touch. Our current touch estimation network accurately predicts the onset of touch events, including time, location, and the responsible nger, but falls short in recognizing sustained contact. Future work could experiment with re ning the network to predict touch states press, hold, and release using additional motion and visual nger features [28].

No eye tracking. During the collection of the touch input dataset (Section 4.1), participants interacted with a visible keyboard overlay. No speci c instructions were provided regarding whether participants should maintain visual contact with their hands while typing. Consequently, the touch distribution encompasses inputs re ecting varying degrees of attentiveness and precision. This includes more accurate input during periods of heightened attention and careful typing as well as less precise input during rapid, eyes-free typing. Future iterations of our probabilistic text decoder could account for this by employing adapted touch distributions. Such adaptations could be personalized and contingent upon the user's current focus of attention, approximated by the direction of gaze (many of today's headsets are already equipped with eye trackers).

Resolving uncertainty through motion cues. Our current framework infers user input based on touch locations relative to command interfaces (e.g., buttons and keys). Our approach thus generalizes across a wider range of devices and input tasks. However, in certain tasks, including text entry, hand motions o er additional cues that could help resolve input uncertainty. For instance, touch typists consistently use speci c nger for certain characters. Future adaptations speci cally designed for text entry could integrate hand motion information to enhance accuracy. This kind of information may vary from one user to another, making it suitable for personalizing a model to each individual user's typing style.

8 CONCLUSION

We have presented TouchInsight, a novel method for detecting touch input on physical surfaces with uncertainty based on egocentric vision from head-worn devices. TouchInsight incorporates a neural network to accurately recognize the moment of touch events, the identity of the touching nger, and the input location on the surface. The key novelty of our method is to explicitly integrate the uncertainties that stem from the two errors involved during detection: user error introduced by potential occlusion, nger softness, and motor inaccuracy and sensing error introduced by egocentric vantage points that lead to self-occlusion in addition to regular sensor noise. Our network accounts for the sensing uncertainties and estimates a bivariate Gaussian distribution for the touch location. In our evaluation, our method inferred locations with a mean position error of 6.3 mm.

Beyond the mere uncertainty scores for estimated touch locations, we demonstrate how these uncertainties provide a key bene t as part of a probabilistic framework to decode rapid surface touch input from all ten ngers during dexterous text entry. To enhance accuracy, our probabilistic framework also incorporates priors from both character- and word-level language models. We evaluated our probabilistic framework as part of an end-to-end text entry system. Participants entered text with a mean entry rate of 37.0 WPM with 2.9% uncorrected error rate, outperforming a mid-air keyboard baseline in performance, task load, and user preference.

Taken together, we believe that our approach enables better interaction in MR, particularly during prolonged productivity tasks that often require e cient text input in addition to direct interaction. Because our framework can in principle generalize to a wide range of command prediction tasks, we believe that our probabilistic and uncertainty-aware approach holds substantial promise to support all on-surface input interactions in MR in the future.

ACKNOWLEDGMENTS

We thank Yangyang Shi, Bradford Snow, Pinhao Guo, and Jingming Dong for helpful discussions and comments, as well as the participants of our user studies.

REFERENCES

[1] 2020. Tap Strap 2: All-In-One Wearable Keyboard, Mouse & Air Gesture Controller. https://www.tapwithus.com/

[2] Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, and Andrew Blake. 2007. High precision multi-touch sensing on surfaces using overhead cameras. In Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '07). IEEE, 197 200.

[3] Nikola Banovic, Tovi Grossman, and George Fitzmaurice. 2013. The e ect of time-based cost of error in target-directed pointing tasks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1373 1382.

[4] Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K Dey, and Jennifer Manko . 2017. Quantifying aversion to costly typing errors in expert mobile text entry. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. 4229 4241.

[5] Nikola Banovic, Ticha Sethapakdi, Yasasvi Hari, Anind K Dey, and Jennifer Manko . 2019. The limits of expert text entry speed on mobile keyboards with autocorrect. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services. 1 12.

[6] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: modeling nger touch with tts' law. In Proceedings of the SIGCHI conference on human factors in computing systems. 1363 1372.

[7] Xiaojun Bi and Shumin Zhai. 2013. Bayesian touch: a statistical criterion of target selection with nger touch. In Proceedings of the 26th annual ACM symposium on User interface software and technology. 51 60.

[8] Doug A Bowman, Sabine Coquillart, Bernd Froehlich, Michitaka Hirose, Yoshifumi Kitamura, Kiyoshi Kiyokawa, and Wolfgang Stuerzlinger. 2008. 3d user interfaces: New directions and perspectives. IEEE computer graphics and applications 28, 6 (2008), 20 36.

[9] Daniel Buschek and Florian Alt. 2015. TouchML: A Machine Learning Toolkit for Modelling Spatial Touch Targeting Behaviour. In Proceedings of the 20th International Conference on Intelligent User Interfaces (Atlanta, Georgia, USA) (IUI '15). Association for Computing Machinery, New York, NY, USA, 110 114. https://doi.org/10.1145/2678025.2701381

[10] Jongeun Cha, Seung-man Kim, Ian Oakley, Jeha Ryu, and Kwan H. Lee. 2005. Haptic Interaction with Depth Video Media. In Advances in Multimedia Information Processing - PCM 2005, Yo-Sung Ho and Hyoung Joong Kim (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 420 430.

[11] Nutan Chen, Göran Westling, Benoni B Edin, and Patrick van der Smagt. 2020. Estimating ngertip forces, torques, and local curvatures from ngernail images. Robotica 38, 7 (2020), 1242 1262.

[12] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D Wilson. 2017. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. 3718 3728.

[13] Yi Fei Cheng, Ti any Luong, Andreas Rene Fender, Paul Streli, and Christian Holz. 2022. ComforTable User Interfaces: Surfaces Reduce Input Error, Time, and Exertion for Tabletop and Mid-air User Interfaces. In 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE.

[14] Wenzhe Cui, Rui Liu, Zhi Li, Yifan Wang, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, IV Ramakrishnan, Fusheng Wang, et al. 2023. GlanceWriter:

Writing Text by Glancing Over Letters with Gaze. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems 1 13.
[15] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a xed-length context. arXiv preprint arXiv:1901.02860 (2019).
[16] John J Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. ACM Transactions on Computer-Human Interaction (TOCHI) 25, 6 (2018), 1 40.
[17] John J Dudley, Jingyao Zheng, Aakar Gupta, Hrvoje Benko, Matt Longest, Robert Wang, and Per Ola Kristensson. 2023. Evaluating the performance of hand-based probabilistic text input methods on a mid-air virtual qwerty keyboard. IEEE Transactions on Visualization and Computer Graphics (2023).
[18] Camille Dupré, Caroline Appert, Stéphanie Rey, Houssem Saidi, and Emmanuel Pietriga. 2024. TriPad: Touch Input in AR on Ordinary Surfaces with Hand Tracking Only. In CHI'24: Proceedings of the 42nd SIGCHI conference on Human Factors in computing systems
[19] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. 2020. Use the force, luke! learning to predict physical forces by simulating e ects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 224 233.
[20] Neil Xu Fan and Robert Xiao. 2022. Reducing the Latency of Touch Tracking on Ad-Hoc Surfaces. Proc. ACM Hum.-Comput. Interact. 6, ISS, Article 577 (nov 2022), 11 pages. https://doi.org/10.1145/3567730
[21] Jacqui Fashimpaur, Kenrick Kin, and Matt Longest. 2020. PinchType: Text Entry for Virtual and Augmented Reality Using Comfortable Thumb to Fingertip Pinches. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1 7. https://doi.org/10.1145/3334480.3382888
[22] Leah Findlater, Jacob O Wobbrock, and Daniel Wigdor. 2011. Typing on at glass: examining ten- nger expert typing patterns on touch surfaces. Proceedings of the SIGCHI conference on Human factors in computing systems 2453 2462.
[23] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. 2018. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In Proceedings of the IEEE conference on computer vision and pattern recognition 409 419.
[24] Shaona Ghosh and Per Ola Kristensson. 2017. Neural networks for text correction and completion in keyboard decoding. arXiv preprint arXiv:1709.06429 (2017).
[25] Jun Gong, Aakar Gupta, and Hrvoje Benko. 2020. Acustico: Surface Tap Detection and Localization Using Wrist-Based Acoustic TDOA Sensing. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (Virtual Event, USA) (UIST '20). Association for Computing Machinery, New York, NY, USA, 406 419. https://doi.org/10.1145/3379337.3415901
[26] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In Proceedings of the 7th international conference on Intelligent user interfaces 194 195.
[27] Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. 2024. PressureVision++: Estimating Fingertip Pressure from Diverse RGB Images. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 8698 8708.
[28] Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. 2022. Pressurevision: Estimating hand pressure from a single rgb image. In European Conference on Computer Vision Springer, 328 345.
[29] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. 2021. Contactopt: Optimizing contact to improve grasps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 1471 1481.
[30] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classi cation: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning 369 376.
[31] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 1059 1070. https://doi.org/10.1145/3332165.3347947
[32] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4, 4, Article 128 (Dec. 2020), 29 pages. https://doi.org/10.1145/3432204
[33] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (Santa Barbara, California, USA) (UIST '11). Association for Computing Machinery, New York, NY, USA, 283 292. https://doi.org/10.1145/2047196.2047233
[34] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Je Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muza er Akbay, Zheng Wang, et al. 2020. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. ACM Transactions on Graphics (TOG) 39, 4 (2020), 87 1.
[35] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D Twigg, and Kenrick Kin. 2018. Online optical marker-based hand tracking with deep labels. Acm transactions on graphics (tog) 37, 4 (2018), 1 10.
[36] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. 2022. UmeTrack: Uni ed multi-view end-to-end hand tracking for VR. In SIGGRAPH Asia 2022 Conference Papers 1 9.
[37] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In Proceedings of the 24th annual ACM symposium on User interface software and technology 441 450.
[38] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 11807 11816.
[39] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. The elements of statistical learnin. Cited on (2009), 33.
[40] Fengming He, Xiyun Hu, Jingyu Shi, Xun Qian, Tianyi Wang, and Karthik Ramani. 2023. Ubi Edge: Authoring Edge-Based Opportunistic Tangible User Interfaces in Augmented Reality. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems 1 14.
[41] Kenneth Hea eld. 2011. KenLM: Faster and Smaller Language Model Queries. In Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, Scotland, 187 197. https://www.aclweb.org/anthology/W11-2123
[42] Steven Henderson and Steven Feiner. 2010. Exploring the bene ts of augmented reality documentation for maintenance and repair. IEEE transactions on visualization and computer graphics 17, 10 (2010), 1355 1368.
[43] Steven Henderson and Steven Feiner. 2010. Opportunistic Tangible User Interfaces for Augmented Reality. IEEE Transactions on Visualization and Computer Graphics 16, 1 (2010), 4 16. https://doi.org/10.1109/TVCG.2009.91
[44] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting ngerprints. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 581 590.
[45] Christian Holz and Patrick Baudisch. 2011. Understanding touch. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 2501 2510. https://doi.org/10.1145/1978942.1979308
[46] Apple Inc. 2024. Apple Vision Pro. https://www.apple.com/apple-vision-pro/
[47] Sensel Inc. 2024. Sensel Morph. https://morph.sensel.com/
[48] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems 3328 3339.
[49] Taejun Kim, Amy Karlson, Aakar Gupta, Tovi Grossman, Jason Wu, Parastoo Abtahi, Christopher Collins, Michael Glueck, and Hemant Bhaskar Surale. 2023. STAR: Smartphone-analogous Typing in Augmented Reality. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology 1 13.
[50] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In International Conference on Learning Representations
[51] Konrad P Körding and Daniel M Wolpert. 2004. Bayesian integration in sensorimotor learning. Nature 427, 6971 (2004), 244 247.
[52] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. 2021. H2o: Two hands manipulating objects for rst person interaction recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision 10138 10148.
[53] Minkyung Lee, Woontack Woo, et al. 2003. ARKB: 3D vision-based Augmented Reality Keyboard.. In ICAT.
[54] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic. 2019. Estimating 3d motion and forces of person-object interactions from monocular video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 8640 8649.
[55] Zhuojun Li, Chun Yu, Yizheng Gu, and Yuanchun Shi. 2023. ResType: Invisible and Adaptive Tablet Keyboard Leveraging Resting Fingers. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 722, 14 pages. https://doi.org/10.1145/3544548.3581055
[56] Chen Liang, Chi Hsia, Chun Yu, Yukang Yan, Yuntao Wang, and Yuanchun Shi. 2023. DRG-Keyboard: Enabling Subtle Gesture Typing on the Fingertip with Dual IMU Rings. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6, 4, Article 170 (jan 2023), 30 pages. https://doi.org/10.1145/3569463

[57] Chen Liang, Xutong Wang, Zisu Li, Chi Hsia, Mingming Fan, Chun Yu, and Yuanchun Shi. 2023. ShadowTouch: Enabling Free-Form Touch-Based Hand-to-Surface Interaction with Wrist-Mounted Illuminant by Shadow Projection. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[58] Zongjian Liu, Jieling He, Jianjiang Feng, and Jie Zhou. 2023. PrinType: Text Entry via Fingerprint Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–31.

[59] Tiffany Luong, Yi Fei Cheng, Max Möbus, Andreas Fender, and Christian Holz. 2023. Controllers or Bare Hands? A Controlled Evaluation of Input Techniques on Interaction Performance and Exertion in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[60] Yan Ma, Shumin Zhai, IV Ramakrishnan, and Xiaojun Bi. 2021. Modeling Touch Point Distribution with Rotational Dual Gaussian Model. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 1197–1209. https://doi.org/10.1145/3472749.3474816

[61] I. Scott MacKenzie. 2015. A Note on Calculating Text Entry Speed. https://www.yorku.ca/mack/RN-TextEntrySpeed.html#:~:text=Words%20Per%20Minute&text=The%20transformation%20requires%20multiplying%20the,using%20actual%20words%20per%20minute.

[62] I. Scott MacKenzie and R. William Soukoreff. 2002. A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction* (Aarhus, Denmark) *(NordiCHI '02)*. Association for Computing Machinery, New York, NY, USA, 243–246. https://doi.org/10.1145/572020.572056

[63] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. 754–755.

[64] Inc. Magic Leap. 2024. *Magic Leap 2*. https://www.magicleap.com/magic-leap-2

[65] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1073–1082.

[66] Stephen A Mascaro and H Harry Asada. 2004. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *IEEE Transactions on Robotics and Automation* 20, 1 (2004), 26–35.

[67] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 519–528.

[68] Meta. 2024. *Meta Quest 3: Expand your world with Meta Quest 3*. https://www.oculus.com/quest-3/

[69] Meta. 2024. *Use Your Fingers (Not Controllers) to Swipe Through the VR Interface on Meta Quest*. https://about.fb.com/news/2023/02/meta-quest-direct-touch-use-your-fingers-in-vr/

[70] Meta. 2024. *Virtual Keyboard Overview*. https://developer.oculus.com/documentation/unity/VK-unity-overview/

[71] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. 2019. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (ToG)* 38, 4 (2019), 1–13.

[72] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*. 1154–1163.

[73] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing Hands in 3D with Transformers. In *CVPR*.

[74] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. 2008. The matrix cookbook. *Technical University of Denmark* 7, 15 (2008), 510.

[75] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. 2017. Hand-object contact force estimation from markerless visual tracking. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2883–2896.

[76] Daniel R Rashid and Noah A Smith. 2008. Relative keyboard input system. In *Proceedings of the 13th international conference on Intelligent user interfaces*. 397–400.

[77] Mark Richardson, Fadi Botros, Yangyang Shi, Bradford Snow, Pinhao Guo, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. 2024. StegoType: Surface Typing from Egocentric Cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*.

[78] Mark Richardson, Matt Durasoff, and Robert Wang. 2020. Decoding surface touch typing from hand-tracking. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*. 686–696.

[79] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2010. FingerCloud: uncertainty and autonomy handover incapacitive sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 577–580.

[80] Elliot N. Saba, Eric C. Larson, and Shwetak N. Patel. 2012. Dante vision: In-air and touch gesture sensing for natural surface interaction with combined depth and thermal cameras. In *2012 IEEE International Conference on Emerging Signal Processing Applications*. 167–170. https://doi.org/10.1109/ESPA.2012.6152472

[81] Julia Schwarz, Scott Hudson, Jennifer Mankoff, and Andrew D Wilson. 2010. A framework for robust and flexible handling of inputs with uncertainty. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 47–56.

[82] Julia Schwarz, Jennifer Mankoff, and Scott Hudson. 2011. Monte carlo methods for managing interactive state, action and feedback under uncertainty. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 235–244.

[83] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. 2022. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=aPOpXlnV1T

[84] Junxiao Shen, John Dudley, and Per Ola Kristensson. 2023. Fast and Robust Mid-Air Gesture Typing for AR Headsets using 3D Trajectory Decoding. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[85] Vivian Shen, James Spann, and Chris Harrison. 2021. FarOut Touch: Extending the Range of Ad Hoc Touch Sensing with Depth Cameras. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 5, 12 pages. https://doi.org/10.1145/3485279.3485281

[86] Weinan Shi, Chun Yu, Xin Yi, Zhen Li, and Yuanchun Shi. 2018. TOAST: Ten-finger eyes-free typing on touchable surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–23.

[87] Yilei Shi, Haimo Zhang, Jiashuo Cao, and Suranga Nanayakkara. 2020. VersaTouch: A Versatile Plug-and-Play System That Enables Touch Interactions on Everyday Passive Surfaces. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 26, 12 pages. https://doi.org/10.1145/3384657.3384778

[88] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 59 (jun 2020), 25 pages. https://doi.org/10.1145/3397309

[89] Adalberto L Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3307–3316.

[90] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/642611.642632

[91] Paul Streli and Christian Holz. 2021. Capcontact: Super-resolution contact areas from capacitive touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[92] Paul Streli, Jiaxi Jiang, Andreas Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–16 pages. https://doi.org/10.1145/3491102.3501878

[93] Paul Streli, Jiaxi Jiang, Juliete Rossie, and Christian Holz. 2023. Structured Light Speckle: Joint Ego-Centric Depth Estimation and Low-Latency Contact Detection via Remote Vibrometry. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–12.

[94] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. 2020. *TelemetRing: A Batteryless and Wireless Ring-Shaped Keyboard Using Passive Inductive Telemetry*. Association for Computing Machinery, New York, NY, USA, 1161–1168. https://doi.org/10.1145/3379337.3415873

[95] Keith Vertanen. [n. d.]. Challenging Twitter phrase set. Sentences from twitter designed to be challenging to recognize. 213 out-of-vocabulary phrases, 194 in-vocabulary phrases. https://keithv.com/data/twitter-phrases.zip

[96] Keith Vertanen. 2019. Vocab 100K. English word vocabulary of 100K words. https://keithv.com/data/vocab_100k.txt

[97] Keith Vertanen, Crystal Fletcher, Dylan Gaines, Jacob Gould, and Per Ola Kristensson. 2018. *The Impact of Word, Multiple Word, and Sentence Input on Virtual Keyboard Decoding Performance*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174200

[98] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M Stanage, Robbie Watling, and Per Ola Kristensson. 2019. VelociWatch: Designing and evaluating a virtual keyboard for the input of challenging text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[99] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual*