DisMouse: Disentangling Information from Mouse Movement Data

Guanhua Zhang
guanhua.zhang@vis.uni-stuttgart.de
University of Stuttgart
Stuttgart, Germany

Zhiming Hu*
zhiming.hu@vis.uni-stuttgart.de
University of Stuttgart
Stuttgart, Germany

Andreas Bulling
andreas.bulling@vis.uni-stuttgart.de
University of Stuttgart
Stuttgart, Germany

ABSTRACT

Mouse movement data contain rich information about users, performed tasks, and user interfaces, but separating the respective components remains challenging and unexplored. As a first step to address this challenge, we propose *DisMouse* – the first method to disentangle user-specific and user-independent information and stochastic variations from mouse movement data. At the core of our method is an autoencoder trained in a semi-supervised fashion, consisting of a self-supervised denoising diffusion process and a supervised contrastive user identification module. Through evaluations on three datasets, we show that DisMouse 1) captures complementary information of mouse input, hence providing an interpretable framework for modelling mouse movements, 2) can be used to produce refined features, thus enabling various applications such as personalised and variable mouse data generation, and 3) generalises across different datasets. Taken together, our results underline the significant potential of disentangled representation learning for explainable, controllable, and generalised mouse behaviour modelling.

KEYWORDS

Classical GUI; Machine Learning; Mouse Movement; Disentangled Representation Learning; Semi-Supervised Learning; Diffusion

ACM Reference Format:
Guanhua Zhang, Zhiming Hu, and Andreas Bulling. 2024. DisMouse: Disentangling Information from Mouse Movement Data. 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, 13 pages. https://doi.org/10.1145/3654777.3676411

1 INTRODUCTION

The mouse remains one of the most important input modalities in human-computer interaction (HCI), serving as a ubiquitous tool for navigating and interacting with a wide range of interactive systems. Mouse movements reflect a complex interplay of various attributes, such as user characteristics (e.g. expertise in using the mouse and familiarity with the specific system being used) [5, 27, 40], interactive goals and tasks that users aim to accomplish [10, 11, 29], the layout and elements of user interfaces (UIs) they interact with [14, 31, 59, 66], and inherent noise within the mouse device itself [64]. As such, separating these different attributes from raw mouse movement data has significant potential but is also profoundly challenging.

Disentangled representation learning has emerged as a promising new paradigm for tackling this challenge. Separating underlying data attributes into multiple complementary representations has gained increasing adoption in various domains, including computer vision [41, 48, 60], language processing [26], and temporal signal processing [34, 50]. The disentangled representations have been shown to offer various benefits, such as producing a set of refined features [50], providing a clear understanding of the isolated factors within the data [56], promoting AI explainablity by imitating how humans obtain semantic meanings [23, 39], or enabling controllable data manipulation by changing a specific representation component [43]. Despite the potential of disentangled representations, no existing work has yet separated different representation components from mouse data.

As a first step to fill this gap, we propose *DisMouse* – the first method to disentangle user-specific information, user-independent information, and stochastic variations within mouse movement data. We prioritise the particular attribute of users because: understanding users is crucial for personalised interactive systems [52], and user labels are pervasively available in existing mouse datasets [40]. Specifically, DisMouse employs a diffusion-based autoencoder architecture and is trained to create near-exact reconstruction of the input data. We first use the diffusion model to separate the semantic representation and stochastic variations. We then particularly design a contrastive user identification module consisting of two branches to further disentangle the semantic representation into user-specific and user-independent components: we split the semantic representation averagely into two halves, and feed each half to each branch. The two branches are linked inversely via a gradient reversal layer (GRL) [16], and the minimisation of their mutual information (MI) [30]. Therefore, this module forces one branch to refine user-specific information while the other focuses on user-independent features. We train DisMouse in a semi-supervised fashion with the diffusion-based reconstruction acting as a self-supervised learning task, while the contrastive user identification module is supervised by the user labels. Cross-dataset evaluations demonstrate that the disentangled representations generalise and capture complementary information about mouse movements. These representations further empower exploring various use cases, such as feature refinement and controllable data generation, including personalised and variable mouse movements, which can then be used for data augmentation.

*Corresponding author

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

ACM ISBN 979-8-4007-0628-8/24/10…$15.00
https://doi.org/10.1145/3654777.3676411

In summary, our contributions are three-fold:

(1) We propose DisMouse[1] – the first computational method to disentangle user-specific, user-independent and stochastic information from mouse movement data. We present a novel semi-supervised training paradigm specifically designed for this goal.
(2) We show that the representations learnt by DisMouse capture complementary information and achieve cross-dataset generalisability.
(3) We demonstrate that these disentangled representations enable various use cases, such as the refinement of mouse features and controllable generation of mouse movements.

2 RELATED WORK

We discuss related work on (1) computational representation learning in HCI, (2) disentangled representation learning, and (3) applications of mouse movement modelling.

2.1 Computational Representation Learning in HCI

A crucial step for building data-driven models in HCI is to learn informative representations of the data. Existing works have primarily focused on learning joint representations that capture various aspects of the input data in a single encoding. For example, Wang et al. presented a Transformer-based autoencoder to learn a semantic representation that considered both individual UI components and their hierarchical structure within the interface [55]. Similarly, Li et al. proposed Screen2Vec to learn a joint representation of user interface layouts, element hierarchy and application descriptions using an image autoencoder and a pre-trained language model [33]. Li et al. introduced a contrastive learning-based method to learn visualisation representations that incorporate visual and structural information [32]. In early work, Borji et al. used hidden Markov models to encode mouse movement locations [2], while recently, Zhang et al. proposed a Transformer-based autoencoder to learn a comprehensive representation of mouse data in both time and frequency domains [67]. In terms of keyboard behaviour, Sun et al. leveraged Gaussian mixture models [52] whereas Zhang et al. applied a natural language encoder, byte pair encoding, to represent both mouse and keystroke action sequences [65].

All of these works have focused on learning joint representations of HCI data. While joint representations have been commonly used in various HCI tasks, they may not fully capture the underlying factors influencing these data. Interactive behaviour, in particular, is a complex interplay of various attributes, including user characteristics, interactive goals and UI designs.

2.2 Disentangled Representation Learning

Disentangled representation learning aims to break down observable data into multiple independent representations that carry complementary aspects within the data. Because of the ability to improve model explainability and controllability [56], disentangled representations have been widely used in various research fields, such as computer vision [41, 48, 60], natural language processing [23], recommender systems [39], and temporal signal processing [34, 50]. For example, Preechakul et al. employed a diffusion probabilistic model to separate a human face image into two distinct parts: a high-level component capturing the overall semantic meaning and a low-level component representing stochastic variations [43]. These disentangled representations enabled applications such as image attribute manipulation, where users could modify specific aspects of an image while preserving others. Wang et al. disentangled an image into its content and style components. Their method explicitly extracted the content information and implicitly learnt the complementary style information via diffusion models [57]. In the realm of time series data, Li et al. proposed a variational autoencoder-based framework to disentangle both individual latent factors and group-level semantic segments [34]. This enabled the creation of refined features and allowed researchers to understand which components captured information about what aspects of the input data. Su et al. focused on human activity signals, including a 3-axis accelerometer, gyroscope, and magnetometer. They proposed a method to disentangle activity patterns from personal styles or environmental noise using an adversarial disentanglement mechanism [50].

Despite the benefits of separating information contained in data, disentangling representations from user interactive behaviour remains unexplored. Our work fills this gap by learning disentangled representations from mouse movement behaviour – a pervasive, informative, and important modality in HCI research and real-world applications.

2.3 Applications of Mouse Movement Modelling

Mouse data have been demonstrated to be crucial for numerous applications, such as user modelling [5, 13, 40], interactive goal recognition and forecasting [11, 29, 31]. Different from prior works that used information theory [19], motor control [8, 18, 28, 38, 54], or statistical methods [35] to model mouse movements in 3D space, we focus on data-driven methods for on-screen mouse data. For example, Chuda et al. extracted features including movement velocity and acceleration to identify users during the process of web browsing [5]. Xu et al. showed that users' visual attention could be predicted from mouse movements [59] while Rheem et al. revealed that the mouse response time and trajectory deviation are closely linked with cognitive load [45]. Elbahi et al. used mouse trajectories in an e-learning interface to fit a hidden Markov model to recognise interactive tasks [10], while Zhang et al. built random forest-based classifiers to predict the next formatting activity that users intended to perform in a text editing scenario based on their mouse movement coordinates [66]. Although mouse data have been used in various data-driven applications like the above, most of them followed the same workflow of learning a joint representation and then building a classifier on top. The entangled underlying attributes within mouse movement data restrict the use cases of mouse behaviour.

In contrast, DisMouse disentangles three mouse representations that contain different, refined information of user-specific, user-independent and stochastic variations. This unlocks new possibilities for HCI applications, including fine-grained control over

[1] https://perceptualui.org/publications/zhang24_uist

the generation process, leading to personalised or variable mouse movement data.

3 DISENTANGLING MOUSE BEHAVIOUR

To decompose different information from mouse movement data, we propose *DisMouse*, a novel semi-supervised method leveraging a diffusion-based autoencoder (see Figure 1 for an overview of our method). DisMouse can disentangle three components, i.e., user-specific representation, user-independent representation and stochastic variations within the data. We prioritised the user factor in this work because understanding user characteristics is key to personalised interactive systems [40, 52], and existing mouse datasets often provide user labels, making incorporating user identity in the analysis and training process feasible.

3.1 DisMouse Architecture

Overall, DisMouse utilised an autoencoder architecture and was trained in a *semi-supervised* manner. We employed a *self-supervised* diffusion process (depicted by the blue and purple arrows in Figure 1) to reconstruct the input data while separating semantic information and the remaining stochastic variations x_T. x_T is essential for near-exact reconstruction and data manipulation [43]. We chose diffusion models because they have achieved state-of-the-art performance in data denoising, reconstruction and generation [22, 25, 60]. To further disentangle the remaining semantic information, we particularly designed a *supervised* module (green arrows in Figure 1), comprised of two classifiers based on multi-layer perceptron (MLP). Both classifiers were trained for user identification, but they were linked inversely through a gradient reversal layer (GRL) [16] and a mutual information loss (L_{MI}) [30]. Our model facilitated the independent learning of individual representation components while enabling their subsequent integration for data generation within a unified diffusion-based framework.

3.1.1 Self-Supervised Diffusion Process. Our approach leveraged a self-supervised diffusion process to learn a joint semantic representation and stochastic variations. We first used a **semantic encoder** to compress the input data into a latent embedding vector (denoted by E_{sem}). This embedding subsequently served as a conditioning signal to guide the denoising process and generate the output within the decoder.

For the **decoder**, we adopted a **conditional DDIM** (denoising diffusion implicit model) [49], which included a forward and a reverse process. During the forward process, Gaussian noise was progressively added to to the input x_0 across discrete time steps t ($t \in [1, T]$) to generate a sequence of increasingly noisy versions x_t:

$$q(x_t|x_0) = N(\sqrt{\alpha_t}x_0, (1-\alpha_t)I),\ \alpha_t = \prod_{i=1}^{t}(1-\beta_i) \tag{1}$$

where β_t is a hyperparameter controlling the noise level at each step. The reverse denoising process $\epsilon_\theta(x_t, t, E_{sem})$ of DDIM is a deterministic process guided by the condition E_{sem}, aiming to recover the clean input x_0. This process is formulated as:

$$p_\theta(x_{0:T}|E_{sem}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, E_{sem}) \tag{2}$$

where

$$p_\theta(x_{t-1}|x_t, E_{sem}) = \begin{cases} N(f_\theta(x_1, 1, E_{sem}), 0) & t = 1 \\ q(x_{t-1}|x_t, f_\theta(x_t, t, E_{sem})) & otherwise \end{cases} \tag{3}$$

In practice, $f_\theta(x_t, t, E_{sem})$ denotes a neural network function implemented with a UNet architecture [46] trained to estimate the denoised version at each step. The specific form of $f_\theta(x_t, t, E_{sem})$ is given by:

$$f_\theta(x_t, t, E_{sem}) = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t, E_{sem})) \tag{4}$$

After being trained, the DDIM can also be used as a **stochastic encoder** to particularly compute a deterministic x_T from the input data (depicted by the purple arrows in Figure 1). This x_T is important for use cases like data reconstruction and manipulation. The computation simply runs the deterministic process backwards [43]:

$$x_{t+1} = \sqrt{\alpha_{t+1}}f_\theta(x_t, t, E_{sem}) + \sqrt{1-\alpha_{t+1}}\epsilon_\theta(x_t, t, E_{sem}) \tag{5}$$

The DDIM employs a UNet composed of 20 residual blocks based on one-dimensional convolutional layers (1DCNN) inspired by [7], where each block incorporates a skip connection to improve the training efficiency [20]. The detailed architecture of the residual block can be found in Figure 2. We opted for 1DCNN given its ability to capture local dependencies of the input signals, good performance in prior diffusion models, and fast computation [53]. The semantic embedding E_{sem} is first input to a two-layer MLP and then fed to the conditional DDIM module. According to Equation 4 and 5, the time step t is required in the diffusion process. Therefore, we also input t to a separate MLP before feeding into the conditional DDIM. The semantic encoder uses a simplified residual block that only has the first three layers (group normalisation [58], SiLU activation [21] and 1DCNN), given that it does not need any conditioning information. The 1DCNN layer had 128 channels and kernels of size 3, a stride of 1 and a padding of 1. As such, the dimension of E_{sem} is 128.

3.1.2 Supervised User Information Disentanglement. To further separate the user-specific and user-independent information within the semantic representation E_{sem}, we particularly designed a supervised learning module (the Contrastive User Identification module in Figure 1). This module leverages a contrastive strategy to encourage the disentanglement of these information components.

We first split E_{sem} into two equal halves, i.e., the dimension of each half is 64. We sent the first half to an MLP-based classifier to perform user identification. This directly encourages the first half to capture user-specific information for accurate user identity classification. On the contrary, the second half is processed through a GRL [16] before being fed to a separate user identification classifier. The GRL essentially inverts the sign of the gradient and thus achieves the reversion of updating direction during backpropagation [9]. Through GRL, DisMouse pushes the second half of the embedding towards user-independent information and discourages user differentiation while prompting capturing general patterns in mouse movements. Both classifiers comprise two fully-connected layers with 256 hidden units each, separated by a ReLU activation function. A Softmax layer at the end predicts user identification labels. Overall, this contrastive training paradigm encourages the

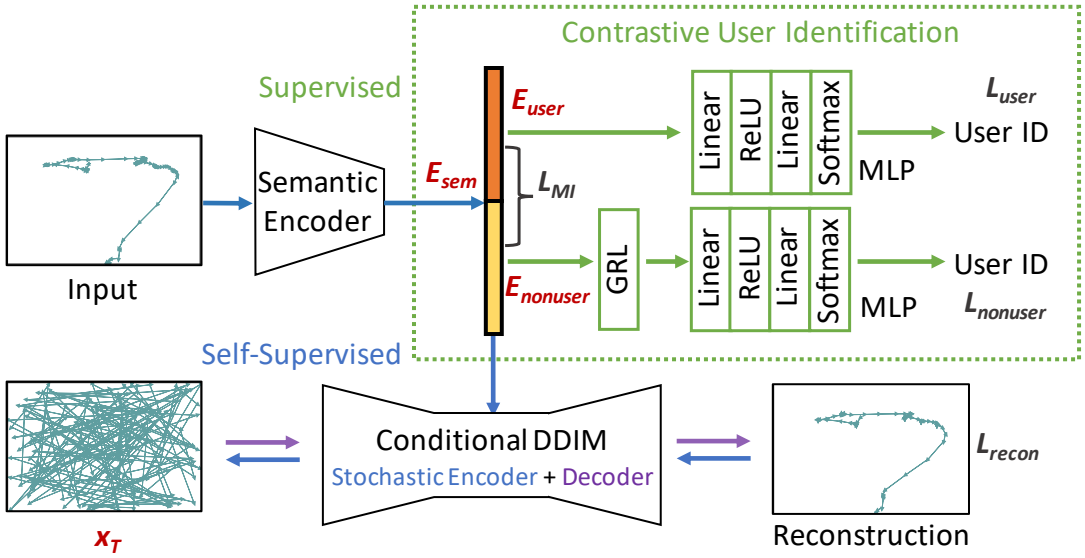

Figure 1: Overview of DisMouse, our proposed semi-supervised model for disentangling information from mouse movement data. Our model employs a self-supervised denoising diffusion process to decompose the input into a semantic embedding E_{sem} and stochastic variations x_T. We then particularly design a supervised contrastive module (in the dashed box) to disentangle user-specific (E_{user}) and user-independent ($E_{nonuser}$) representations. Here, we split E_{sem} equally into two halves and then feed the first half to an MLP-based classifier for user identification and the second half to a second classifier with a similar architecture but an additional gradient reversal layer (GRL). During training, the model leverages a combination of loss functions, including the reconstruction loss for denoising, two classification losses for the contrastive user identification, and a mutual information loss to push E_{user} and $E_{nonuser}$ towards semantic distinction.

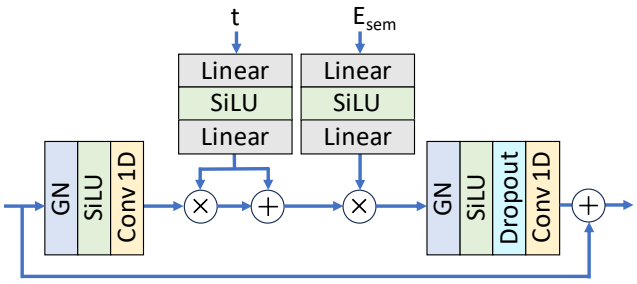

Figure 2: Architecture of the residual block used in the conditional DDIM, conditioned on time step t and the semantic representation E_{sem}.

first half (E_{user}) to encode user-specific information while guiding the second half ($E_{nonuser}$) to represent user-independent patterns.

3.2 Semi-Supervised Training Scheme

3.2.1 Multi-Task Training. To train DisMouse, we proposed a novel semi-supervised training scheme, combining the aforementioned training tasks:

(1) **Diffusion denoising**: We compared the original input data with the reconstruction, as well as the noise ϵ added in the forward process with the predicted noise $\epsilon_\theta(x_t, t, E_{sem})$ in the denoising process. We calculated the mean squared errors (MSEs) from the two comparisons to form the loss function L_{recon} for this task:

$$L_{recon} = L_{noise} + L_{input}$$
$$= \|\epsilon_\theta(x_t, t, E_{sem}) - \epsilon\|_2^2 + \|Reconstruction - Input\|_2^2$$

(2) **Contrastive user identification**: We utilised two separate branches for user identification. Each branch corresponded to a cross-entropy loss function (L_{user} and $L_{nonuser}$) comparing the predicted labels with the ground-truth labels. In addition, we incorporated a mutual information loss L_{MI} between the user-specific representation E_{user} and the user-independent representation $E_{nonuser}$ to encourage extracting discriminative patterns [68, 69].

In summary, the overall training loss L_{total} of DisMouse is defined as:

$$L_{total} = L_{recon} + \beta(L_{user} + L_{nonuser} + L_{MI}) \quad (6)$$

where β is a hyperparameter controlling the weight of the user identification losses and the MI loss. We set β to 0.01 in our experiments. We trained the model for 250 epochs with a batch size of 512 and used the Adam optimiser with a learning rate of 1e-4.

3.2.2 Dataset. We trained DisMouse on the Clarkson dataset [40] given that, to our best knowledge, it is the largest publicly available dataset collected in an unconstrained real-world setting. The dataset contains mouse movement data from 103 participants over a time span of 2.5 years, and consists of 24.7 M samples. The large user population and extended collection period give the Clarkson dataset

the potential to offer diverse and comprehensive representations of mouse behaviour.

3.3 Data Preprocessing

The mouse movement input is a sequence of samples. Each sample is a triplet (x, y, t), where x and y denote the current on-screen coordinates of the cursor, and t is the timestamp. To account for varying screen resolutions, we first rescaled the raw x and y coordinates to the $[0, 1]$ range via a MinMaxScaler. Unlike fixed-rate sampling techniques, mouse data collection produces samples only when the mouse moves. To ensure consistent sampling frequency, we resampled the mouse data at 20 Hz aligning with prior work [42, 67]. Specifically, if a sampling unit (50 ms) lacked data, we duplicated the previous sample to maintain temporal consistency. Then, we divided the mouse data into eight-second segments, with a one-second stride [17, 44]. These segments serve as the input to DisMouse for further processing and analysis.

4 EVALUATION OF DISENTANGLEMENT

We evaluated the effectiveness of DisMouse in disentangling the three representations E_{user}, $E_{nonuser}$, and x_T. We examined if the three representations contained the corresponding expected information: Both E_{user} and $E_{nonuser}$ carried semantic information, whereas x_T primarily captured the inherent stochastic variations within the mouse movement data; between the two semantic representations, E_{user} had more user-specific information while $E_{nonuser}$ contained more general, user-independent information.

To assess the generalisability of our method, we froze the trained DisMouse and transferred it to other datasets without any fine-tuning. We chose two datasets – ACTIVITY [66] and EMAKI [65], given that they are publicly available and cover both lab and out-of-the-lab settings. ACTIVITY comprises data from 16 participants formatting pre-defined text in a controlled laboratory setting. In each trial, participants performed a sequence of formatting activities. There were seven candidate activities: bold, italic, underline, font size, font family, alignment, and indentation. The EMAKI dataset, on the other hand, represents an out-of-the-lab dataset collected in a more natural setting. 39 participants joined an online user study using their own computers to perform three interactive tasks: writing and editing an article, drawing and editing images, and completing questionnaires about demographics and personality traits.

4.1 Semantic vs. Stochastic Representations

We assessed the quality of the learnt representations by evaluating their effectiveness in reconstructing the original input data. Reconstruction quality reflects the ability of these representations to capture essential information about the mouse movement data. We calculated the mean squared error (MSE) between the reconstruction and the original input as the evaluation metric. A lower MSE indicates a better reconstruction. To pinpoint which representation pertained to the core properties of the input mouse data, we conducted an ablation study of the representations. Instead of discarding a representation entirely, we replaced each of them with Gaussian noise during the reconstruction. This is because the diffusion process inherently requires all three components to produce a reconstruction. As shown in Table 1, using all three representations together (DisMouse) yielded the best reconstruction quality. This confirmed that each component captures meaningful aspects of the input mouse movement data and that the information in the three components was complementary to each other. Compared to x_T, replacing the semantic representations E_{user} or $E_{nonuser}$ brought a larger MSE increase, with over 100 times higher MSE on both test datasets. This observation suggests that E_{user} and $E_{nonuser}$ encoded more critical information than x_T. These results are in line with prior findings that x_T captures the remaining stochastic variations and is essential for reconstruction in DDIM [43].

Model	ACTIVITY	EMAKI
DisMouse	**9.86e-5**	**3.24e-5**
$E_{user} \rightarrow$ Noise	1.46e-2	4.30e-3
$E_{nonuser} \rightarrow$ Noise	1.15e-2	9.12e-3
$x_T \rightarrow$ Noise	1.29e-3	5.26e-4

Table 1: Reconstruction MSE achieved by DisMouse and by replacing each of the learnt representations (E_{user}, $E_{nonuser}$, x_T) with Gaussian noise sampled from $N(0, I)$. Replacing any of them results in significantly larger reconstruction errors, indicating that all three components contain meaningful information about the mouse movement data. Compared to the stochastic variations x_T, replacing the semantic representations E_{user} or $E_{nonuser}$ leads to larger MSE increases, showing that these two representations capture more critical information of the mouse data. The lowest MSEs corresponding to the best reconstruction quality are marked in bold.

Furthermore, we plotted human-interpretable reconstruction results achieved by modifying the three representations in Figure 3. We manipulated the representations by adding noise $\lambda n, n \sim N(0, I)$ to them and observed their impact. Starting from the left, the five columns display: the ground truth input, the reconstruction via DisMouse, reconstruction with altered E_{user}, reconstruction with altered $E_{nonuser}$ and reconstruction with altered x_T, respectively. We can see that modifying x_T primarily affects minor details while altering E_{user} or $E_{nonuser}$ significantly impacts the mouse movement trajectory's shape, location and underlying interactive goal. We further validated this quantitatively by calculating MSEs between the original and generated mouse traces and conducting Wilcoxon signed-rank tests. We found significant differences between DisMouse and E_{user}*+Noise*, as well as $E_{nonuser}$*+Noise* ($p < .001$), but no significant difference between DisMouse and x_T*+Noise*. These observations reinforce the notion that x_T conveys finer stochastic variations, while E_{user} and $E_{nonuser}$ carry higher-level semantic information.

4.2 User-Specific Information

To quantify if and how much each component grasped user-specific information, we compared each component's performance on user identification. We first used the frozen semantic encoder to get E_{user}, and then input it to train new classifiers with varying nodes in the last layer for different datasets. We employed five-fold user-dependent cross-validation since both the training and testing sets

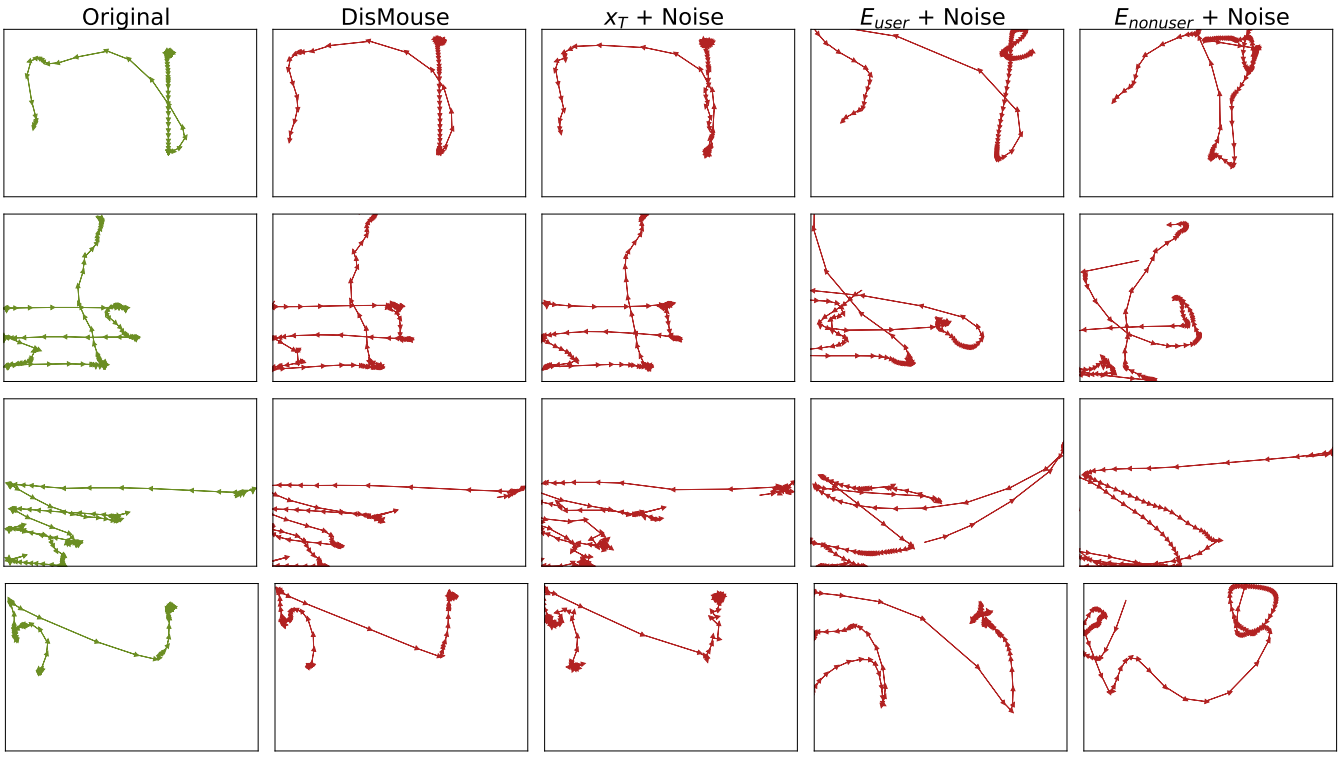

Figure 3: Four examples of input reconstruction using the representations (E_{user}, $E_{nonuser}$ and x_T) learnt by DisMouse, and altering each of them by adding Gaussian noise. Each small arrow indicates a mouse movement data sample (50 ms). The columns (from left to right) display the original mouse trajectories, the reconstruction via DisMouse, as well as the reconstruction with the altered, noisy E_{user}, $E_{nonuser}$ and x_T, respectively. Changing x_T primarily affects minor details, such as the smoothness of the path while preserving the overall patterns like shapes and locations. On the other hand, altering E_{user} or $E_{nonuser}$ significantly impacts the reconstructed mouse movement. These observations indicate that x_T mainly captures stochastic variations, while E_{user} and $E_{nonuser}$ encode core semantic information of mouse behaviour.

Input		ACTIVITY	EMAKI
Handcrafted Features (Full)		36.62±1.98	23.74±1.17
Handcrafted Features (L1)		<u>40.55±1.23</u>	24.22±1.48
Handcrafted Features (Tree)		37.40±1.06	25.88±0.70
VAE μ		35.42±1.10	<u>26.20±0.19</u>
VAE σ		23.86±1.04	20.52±0.33
DisMouse	E_{user}	**51.00±1.39**	**37.60±0.40**
	$E_{nonuser}$	28.35±1.00	13.62±0.23
	x_T	14.23±0.67	11.19±0.29

Table 2: User identification accuracies (mean±standard deviation, in percentage) achieved on the ACTIVITY and EMAKI datasets using the representations learnt by DisMouse (E_{user}, $E_{nonuser}$ and x_T). We compared with handcrafted features (the full set or subsets selected by L1-/tree-based methods) and representations learned by a 1DCNN-based VAE (μ or σ). The highest accuracies are shown in bold, and the second highest are underlined.

must contain data from every user. Specifically, we first randomly divided each participant's data into five sets, and in each fold, we combined the four sets from all users for training. At the same time, the remaining were used for testing. We repeated this process five times and reported their average accuracy in Table 2. To better understand the performance, we also included the results obtained using 75 handcrafted features and a variational autoencoder (VAE) as references. These handcrafted features have been commonly used by a wide range of mouse modelling works and typically capture statistics of on-screen locations, angles, velocities, etc (refer to Appendix A for the complete set). We further added L1-based and tree-based feature selection methods[2] on these features. In addition, we compared with VAE, a well-established method of learning generalised, entangled representations. We implemented the VAE based on 1DCNN, the identical basic block employed in DisMouse. Given that VAEs learnt two latent embeddings representing the mean (μ) and standard deviation(σ), we compared DisMouse representations with both of them.

As shown in Table 2, E_{user} largely outperformed the other two components, the handcrafted features, and VAE representations.

[2]https://scikit-learn.org/stable/modules/feature_selection.html

For example, compared to $E_{nonuser}$, E_{user} obtained a 23.98% (1.76x) higher accuracy on EMAKI and a 22.65% higher accuracy on ACTIVITY. Compared to the handcrafted features and the representations learnt by VAE, E_{user} exhibited improvements of 10.45% and 11.40% on ACTIVITY and EMAKI, respectively. This showed that E_{user} encoded rich user-specific information. On the contrary, x_T consistently achieved the lowest accuracies across datasets. For instance, x_T obtained approximately only half the accuracy compared to $E_{nonuser}$ (14.23% vs. 28.35%) on the ACTIVITY dataset. This observation again indicated that x_T possessed stochastic variations with minimal semantic user information.

4.3 Effectiveness of DisMouse Design

Since we particularly designed a two-branch contrastive module and minimised mutual information between E_{user} and $E_{nonuser}$ to disentangle the semantic information, we conducted an ablation study of these new designs to examine their contributions. We separately removed either of the two branches and its corresponding loss L_{user} or $L_{nonuser}$, as well as the mutual information loss L_{MI}. User identification performance was then evaluated for each configuration. From Table 3, we can observe that removing any of the above designs impacted the representations as expected. User identification accuracy for E_{user} decreased, indicating a reduction in captured user-specific information. Conversely, the accuracy for $E_{nonuser}$ improved, suggesting that it contained more user information in these ablated models. As such, the gap between the two representations was reduced. For example, on the ACTIVITY dataset, removing the branch that weakened user identification ($w/o\ L_{nonuser}$) resulted in a difference of only 2.59% between the accuracies of E_{user} and $E_{nonuser}$. These observations indicated that the ablations led to less disentanglement of the user-related semantic information. Therefore, our design of the contrastive module and the MI loss in DisMouse is effective. They play a crucial role in disentangling user-related semantic information from the overall representation and lead to a clear separation between user-specific, user-independent semantics and stochastic variations.

5 USE CASES ENABLED BY DISMOUSE

We next report three use cases unlocked by the disentangled representations that isolated specific attributes of the mouse movement data: DisMouse can concurrently refine user-specific and user-independent features E_{user} and $E_{nonuser}$, which have the potential to benefit applications requiring corresponding features; manipulating E_{user} can be used to generate personalised mouse movement data; and altering x_T can generate mouse movement variations to augment training data further.

5.1 Refining Mouse Movement Features

Prior research has shown that representations tailored to specific attributes can improve applications that rely heavily on those attributes, e.g., activity-related representations for human activity recognition [50]. DisMouse has the ability to refine different semantic features (E_{user} and $E_{nonuser}$) simultaneously. As demonstrated in Table 2, E_{user}, the refined user-specific features consistently achieved higher user identification accuracy compared to the handcrafted features that contain rich but a mix of user-specific and user-independent information. We further examined whether the refined user-independent features, $E_{nonuser}$, could benefit applications that value patterns that are shared across users rather than those specific to users.

We focused on two example applications, task recognition and next activity prediction, that are essential for adaptive and anticipatory interactive systems [12, 31, 67]. Given that the ACTIVITY dataset provided annotations of seven activities and EMAKI provided labels of three interactive tasks, we conducted next activity prediction on ACTIVITY and task recognition on EMAKI. We performed a five-fold user-independent cross-validation to assess the generalisability of DisMouse across users on the two applications. As such, we randomly split these participants into five sets. In each fold, we trained the classifier using data from four sets of participants and tested it on the remaining set. We repeated this procedure five times and calculated the average accuracy across all folds as the final performance metric.

As Table 4 presents, $E_{nonuser}$ consistently obtained higher accuracies than the handcrafted features and VAE representations. A Wilcoxon signed-rank test confirmed the statistical significance of these improvements. When predicting which one out of seven activities the user would perform the next on the ACTIVITY dataset, $E_{nonuser}$ improved the accuracy by 4.17% ($p < .01$) compared to handcrafted features and VAE embeddings. Similarly, when recognising which one out of the three tasks the users were performing on the EMAKI dataset, $E_{nonuser}$ achieved an accuracy improvement by 3.30% ($p < .01$). These results demonstrated that $E_{nonuser}$ effectively captured information related to common patterns across users. In addition, DisMouse consistently outperformed existing methods that were specifically designed for these classification tasks (see Appendix B). Therefore, DisMouse can concurrently refine distinct mouse features, making them valuable for applications requiring specific information types.

5.2 Generating Personalised Mouse Movement Data

DisMouse's ability to disentangle user-specific information unlocks the possibility of personalised mouse movement data generation. This section explores the potential of using E_{user} to synthesise mouse trajectories that maintain the original user's interaction intent while adopting the movement style of another user. Specifically, given a mouse movement trajectory from user A, we replaced its user-specific representation E^A_{user} with a different user, B's E^B_{user}. The remaining components, $E^A_{nonuser}$ (capturing user-independent movement patterns) and x^A_T (representing stochastic variations), are retrained from user A. This modified representation set (E^B_{user}, $E^A_{nonuser}$, x^A_T), was then fed into the DisMouse diffusion process to generate a new mouse trajectory. The resulting mouse movement trajectory reflects the original user's interaction goal (e.g., similar movement direction and location) but is executed in the movement style characteristic of the target user. To illustrate this concept, we selected two users from the ACTIVITY datasets who exhibit distinct movement styles. We plotted five examples from each of the two users in the first and fourth rows of Figure 4 in green. We can see that user A generated smoother and more flowing movements, while user B tended to make sharper turns and used more straight

Model	ACTIVITY		EMAKI	
	$E_{user} \uparrow$	$E_{nonuser} \downarrow$	$E_{user} \uparrow$	$E_{nonuser} \downarrow$
DisMouse	**51.00±1.39**	**28.35±1.00**	**37.60±0.40**	**13.62±0.23**
w/o L_{user}	40.82±1.53	35.14±0.80	26.52±0.47	19.48±0.30
w/o $L_{nonuser}$	42.07±1.18	39.48±0.75	32.49±0.54	22.89±0.45
w/o L_{MI}	45.46±0.80	33.59±0.91	31.07±0.16	18.76±0.19

Table 3: User identification accuracies (mean±standard deviation, in percentage) on the ACTIVITY and EMAKI datasets using DisMouse and the different ablations of our designs for disentangling mouse representation. These designs include contrastive user identification (L_{user} and $L_{nonuser}$), and minimising the mutual information between E_{user} and $E_{nonuser}$ (L_{MI}). Removing any of these components resulted in a decrease in E_{user} performance and an increase in $E_{nonuser}$ performance, indicating a shift in the information captured by each representation. The gap between E_{user} and $E_{nonuser}$ accuracy also narrowed. These observations support the effectiveness of our method designs in disentangling user-specific and user-independent information from mouse movement data. The highest E_{user} accuracies and the lowest $E_{nonuser}$ accuracies are marked in bold.

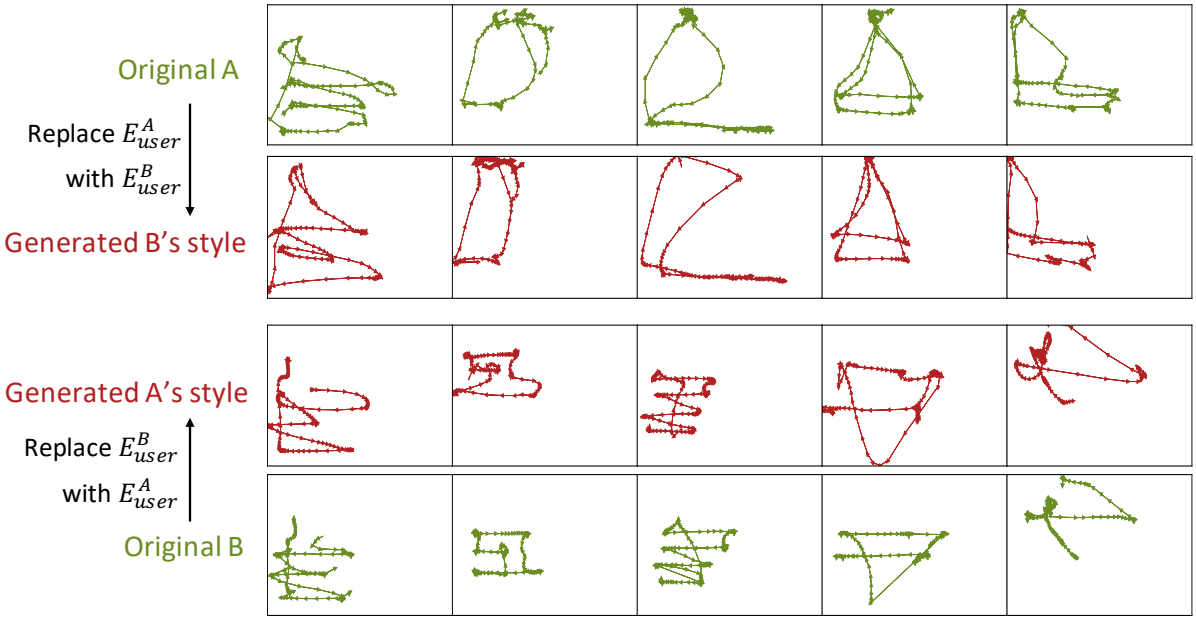

Figure 4: Generating mouse movement data that are personalised to two users from the ACTIVITY dataset. The first and fourth rows (in green) depict five examples of their original trajectories. User A exhibits smoother and flowing movements, whereas B's trajectories involve sharper turns and more straight lines. We swapped their user-specific representation E_{user} to generate trajectories in each other's style (second and third rows, red). The generated data retain the original moving directions and locations but adopt the stylistic characteristics of the target user.

lines. Then we swapped their E_{user} components and plotted the generated new trajectories in red in the second and third rows.

The generated trajectories demonstrate the effectiveness of our approach. These new data maintained the original movement directions and locations (interactive goals) but exhibited the stylistic characteristics of the target user. We observe that the generated data have moving directions and locations similar to the original trajectory but in target users' styles. For example, after replacing the A's E_{user} with B's, the trajectories became sharper and contained more straight lines; whereas replacing B's E_{user} with A's led to trajectories that have fewer abrupt turns but more rounded corners. This initial exploration paves the way for further investigation and potential applications in personalised behaviour modelling, user-centred designs and creating realistic and user-specific test scenarios.

5.3 Generating Mouse Movement Variations

Limited labelled data is a persistent challenge in the HCI community due to the high cost of data collection and annotation [4, 51]. DisMouse offers a solution by leveraging the stochastic variation component x_T to generate mouse movement variations. Figure 3

Input	Next Activity Prediction	Task Recognition
Handcrafted Features (Full)	58.61±2.32	<u>67.94±0.23</u>
Handcrafted Features (L1)	53.52±3.03	67.49±1.33
Handcrafted Features (Tree)	55.68±0.78	66.19±0.74
VAE μ	<u>59.51±0.58</u>	67.59±1.22
VAE σ	57.86±1.81	64.11±2.17
$E_{nonuser}$	**63.68±6.16**	**71.24±1.81**

Table 4: Accuracies (mean±standard deviation, in percentage) achieved on next activity prediction and task recognition, which are two example applications that focus on common (user-independent) patterns and are relevant for intelligent interactive systems. We compared $E_{nonuser}$ with handcrafted features (the full set or subsets selected by L1-/tree-based methods) and representations learned by a 1DCNN-based VAE (μ or σ). The highest accuracies are shown in bold, and the second highest are underlined.

has shown that adding Gaussian noise to x_T can simulate such variations that preserve the core movement intent (shape, location and directions) while differing in details like the straightness between two points. As such, DisMouse can be used to augment training data for data-driven models. We randomly kept 5% of the training data to imitate the scarcity of labelled data [62, 67]. For each training sample, we sampled Gaussian noise and added it to x_T 19 times to generate 19 similar mouse movement trajectories, assigned them the same label as the training sample, and added them to the training set. In this way, we restored the training set to the original size.

We covered all the aforementioned classification applications, i.e., we presented the performance of data augmentation on user identification, next activity prediction and task recognition. We compared our approach with six commonly used data augmentation methods, including duplication, jittering, scaling, rotation, permutation or warping [24, 61]. Every method created 19 augmented samples for each of the samples to restore the data set to the original size. As Table 5 demonstrates, augmenting data using our x_T consistently outperformed other augmentation methods across all the applications and datasets. For example, the accuracy of next activity prediction improved by 4.79% ($p < .001$), task recognition improved by 1.35% ($p < .05$), and user identification improved by 1.70% ($p < .01$) and 1.44% ($p < .05$) on the ACTIVITY and EMAKI datasets, respectively. The Wilcoxon signed-rank test confirmed the statistical significance of these enhancements. Our approach surpassed data duplication because altering x_T injected additional informative content by changing movement details while preserving the essential semantics of the original data. As such, this strategy expanded the training set with meaningful variations.

6 DISCUSSION

6.1 Potential of Disentangling Mouse Behaviour

Disentangled representation learning separates underlying data attributes into different components, thus contributing to explainability and creating the opportunity for various use cases. As the first to disentangle mouse information, we separated the user-specific and user-independent semantics and stochastic variations. For example, we have shown that these disentangled representations enabled controllable, personalised (Section 5.2) or variable (Section 5.3) mouse movement data generation. DisMouse also concurrently produced two refined mouse features E_{user} and $E_{nonuser}$. The refinement was effective for applications that focused on specific information. For example, E_{user} contained refined user-specific features and thus benefited the corresponding application of user identification (see Table 2); whereas $E_{nonuser}$ carried the residual semantic information and thus benefited applications leveraging universal patterns rather than individual user patterns, such as next activity prediction and task recognition (see Table 4). We conducted experiments across different datasets and applications, showing the generalisability of our DisMouse.

Furthermore, given that $E_{nonuser}$ grasped user-independent information, it presents an exciting opportunity for anonymising mouse movement data, i.e., removing user-specific characteristics while preserving common interaction patterns. Investigating privacy-related applications in future research will also be interesting, such as enhancing the security UIs and user behaviour analysis tools.

We have demonstrated the effectiveness and strong potential of disentangling mouse representations in understanding, characterising and generating mouse movement data, which can be used to improve future interactive systems. Our work also provides inspiration of disentangling representations of other interactive behaviour and HCI data.

6.2 Design of DisMouse

We propose the first method of disentangling mouse movement data, using a conditional diffusion-based encoder-decoder architecture. We chose diffusion models because of their recent groundbreaking success in generating language or vision data [37, 63], but they are still under-explored in HCI. The autoencoder architecture first allowed us to learn representations of the input, manipulate any of them, and subsequently integrate them to generate mouse movement data. To disentangle user-specific and user-independent semantic representations, we particularly introduced a two-branch supervised module including the following designs: 1) feeding the first half of the semantic representation to one classifier for user identification to learn user-specific information; 2) inputting the second half to another user identifier but after a GRL to learn user-independent information; 3) minimising the mutual information between the two halves. We illustrated that the disentangled three representations (E_{user}, $E_{nonuser}$ and x_T) conveyed different and complementary information of the input data, via examining the reconstruction both visually (Figure 3) and through MSE metrics (Table 1), as well as via user identification (Table 2). An ablation study showed that all three designs were essential for effective disentanglement (Table 3).

Moreover, we transferred the frozen DisMouse model trained on a large-scale dataset, Clarkson, to two other datasets, ACTIVITY and EMAKI. The three datasets were collected from users under various interactive tasks and settings. For example, Clarkson was collected in a totally unconstrained setting, while ACTIVITY was

Augmentation	ACTIVITY		EMAKI	
	Next Activity Prediction	User Identification	Task Recognition	User Identification
Duplication	<u>53.96±5.62</u>	<u>13.11±3.82</u>	57.85±2.38	6.13±0.44
Jittering	53.18±8.86	10.43±4.28	53.88±3.98	5.05±1.20
Scaling	51.93±8.56	12.11±5.27	57.07±2.64	<u>6.18±0.59</u>
Rotation	53.64±2.46	11.45±2.45	<u>57.92±2.68</u>	5.10±1.04
Permutation	42.20±8.48	10.76±4.03	45.04±3.04	3.92±1.28
Warping	47.15±5.54	11.14±4.83	56.04±1.39	6.15±1.24
x_T + Noise	**58.75±4.74**	**14.81±3.42**	**59.27±2.59**	**7.62±0.76**

Table 5: Accuracies (mean±standard deviation, in percentage) of next activity prediction, user identification and task recognition on ACTIVITY and EMAKI datasets. The original training set size was first reduced to 5% to simulate data scarcity and then augmented back to 100% by altering the stochastic representation x_T and six baselines: duplication, jittering, scaling, rotation, permutation and warping. The highest accuracies are shown in bold while the second-highest ones are underlined.

recorded in a restricted laboratory setting. EMAKI occupied a middle ground – an online study offering participants more freedom than ACTIVITY but still constrained the tasks they had to perform. Our promising results suggested DisMouse's generalisability in capturing mouse movement behaviour patterns. By freezing the model for evaluations and applications, we provided a directly deployable tool and lowered the barrier for interactive behaviour modelling in HCI. Future work can explore using DisMouse on a wider range of datasets. As deep learning continues to evolve, it will also be interesting to investigate innovative techniques to develop more methods to disentangle interactive behaviour.

6.3 Limitations and Future Work

In this work, DisMouse focused on disentangling user-specific information from mouse data due to its importance for personalised interactive systems and prevalence in existing datasets. However, mouse behaviour can also be influenced by other attributes, such as interactive tasks and the design of UIs. Future research can explore disentangling these additional factors, or investigate deeper into user factors to disentangle cognitive state like emotions [3], stress [13] and attention [59]. This is challenging for machine learning in general because the space of possible factors is potentially vast and unknown, but current XAI methods are limited to labels provided with the datasets. Furthermore, although EMAKI offers age, gender and personality trait labels, no prior work has studied if the collected data is linked to these factors. Despite the challenges, entangling fine-grained factors will enable the analysis of user behaviour in a more holistic manner and lead to more adaptive and intelligent interactive systems. Moreover, disentangling UIs or interactive tasks requires mouse trajectory data collected from users performing the same task across different UIs and performing different tasks on the same UI. However, publicly available datasets that offered raw mouse data did not meet the above requirements regarding UIs and tasks [52, 66]. In contrast, those that met the above requirements did not release raw mouse data [6, 36]. Therefore, we call for collecting new datasets that allow the disentanglement of these factors from raw mouse data. Another challenge of disentangling UIs or tasks is to handle more complex mouse data due to real-world tasks, dynamic UIs, or unintentional inputs. Additionally, DisMouse is limited to analysing mouse movement data, thus future research can include more mouse events such as clicks and scrolls, and extend to disentangle multimodal representations such as keyboard [65], gaze [59] and UIs [33, 55]. Multimodal representations would unlock the investigation of the interplay between different modalities, provide a more comprehensive understanding of user behaviour, and potentially lead to richer disentangled representations.

7 CONCLUSION

In this work, we introduced DisMouse, a novel semi-supervised diffusion-based method of, for the first time, learning disentangled representations from mouse movement data. This disentanglement contributes to the explainability of mouse behaviour modelling and helps researchers gain a granular understanding of the underlying factors influencing mouse movements. Extensive experiments demonstrated the effectiveness of DisMouse, with the learnt representations generalising across different datasets. Furthermore, we showed that DisMouse unlocked various applications including feature refinement benefiting classification tasks and controllable generation of personalised or variable mouse movement data. More broadly, our findings pave the way for exciting new avenues in HCI, such as exploring similar techniques to disentangle other types of HCI data, understanding various underlying attributes, and building personalised, explainable and efficient user and behaviour modelling methods.

ACKNOWLEDGMENTS

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting G. Zhang. Z. Hu was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). We would like to thank Mayar Elfares and Yanzhou Chen for their technical support, and thank anonymous reviewers for their helpful feedback.

REFERENCES

[1] Margit Antal and Elöd Egyed-Zsigmond. 2019. Intrusion detection using mouse dynamics. *IET Biometrics* 8, 5 (2019), 285–294.

[2] Ali Borji, Dicky N Sihite, and Laurent Itti. 2012. Probabilistic learning of task-specific visual attention. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 470–477.

[3] Kiran Chhatre, Radek Daněček, Nikos Athanasiou, Giorgio Becherini, Christopher Peters, Michael J Black, and Timo Bolkart. 2023. Emotional Speech-driven 3D Body Animation via Disentangled Latent Diffusion. *arXiv preprint arXiv:2312.04466* (2023).

[4] Jeremy Chu, Dongsheng An, Yan Ma, Wenzhe Cui, Shumin Zhai, Xianfeng David Gu, and Xiaojun Bi. 2023. WordGesture-GAN: Modeling Word-Gesture Movement with Generative Adversarial Network. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg, 1–15.

[5] Daniela Chudá and Peter Krátky. 2014. Usage of computer mouse characteristics for identification in web browsing. In *Proceedings of the 2014 International Conference on Computer Systems and Technologies*. ACM, Ruse, 218–225.

[6] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* 36 (2024).

[7] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[8] Seungwon Do, Minsuk Chang, and Byungjoo Lee. 2021. A simulation model of intermittently controlled point-and-click behaviour. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–17.

[9] Xiaobing Du, Cuixia Ma, Guanhua Zhang, Jinyao Li, Yu-Kun Lai, Guozhen Zhao, Xiaoming Deng, Yong-Jin Liu, and Hongan Wang. 2020. An efficient LSTM network for emotion recognition from multichannel EEG signals. *IEEE Transactions on Affective Computing* 13, 3 (2020), 1528–1540.

[10] Anis Elbahi, Mohamed Ali Mahjoub, and Mohamed Nazih Omri. 2013. Hidden markov model for inferring user task using mouse movement. In *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*. IEEE, Hammamet, 1–7.

[11] Anis Elbahi and Mohamed Nazih Omri. 2015. Web user interact task recognition based on conditional random fields. In *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I 16*. Springer, Valletta, 740–751.

[12] Anis Elbahi, Mohamed Nazih Omri, Mohamed Ali Mahjoub, and Kamel Garrouch. 2016. Mouse movement and probabilistic graphical models based e-learning activity recognition improvement possibilistic model. *Arabian Journal for Science and Engineering* 41, 8 (2016), 2847–2862.

[13] Paul Freihaut, Anja S Göritz, Christoph Rockstroh, and Johannes Blum. 2021. Tracking stress via the computer mouse? Promises and challenges of a potential behavioral stress marker. *Behavior Research Methods* 53, 1 (2021), 1–21.

[14] Eugene Yujun Fu, Tiffany CK Kwok, Erin You Wu, Hong Va Leong, Grace Ngai, and Stephen CF Chan. 2017. Your mouse reveals your next activity: towards predicting user intention from mouse interaction. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, Turin, 869–874.

[15] Hugo Gamboa and Ana Fred. 2004. A behavioral biometric system based on human-computer interaction. In *Biometric Technology for Human Identification*, Vol. 5404. SPIE, Orlando, 381–392.

[16] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[17] Daniel Garabato, Jorge Rodríguez García, Francisco J Novoa, and Carlos Dafonte. 2019. Mouse Behavior Analysis Based on Artificial Intelligence as a Second-Phase Authentication System. *Multidisciplinary Digital Publishing Institute Proceedings* 21, 1 (2019), 29.

[18] Eric J Gonzalez and Sean Follmer. 2023. Sensorimotor Simulation of Redirected Reaching using Stochastic Optimal Feedback Control. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg, 1–17.

[19] Julien Gori and Olivier Rioul. 2020. A feedback information-theoretic transmission scheme (FITTS) for modeling trajectory variability in aimed movements. *Biological cybernetics* 114, 6 (2020), 621–641.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Las Vegas, 770–778.

[21] Dan Hendrycks and Kevin Gimpel. 2023. Gaussian Error Linear Units (GELUs). arXiv:1606.08415 [cs.LG]

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[23] Zhenya Huang, Xin Lin, Hao Wang, Qi Liu, Enhong Chen, Jianhui Ma, Yu Su, and Wei Tong. 2021. Disenqnet: Disentangled representation learning for educational questions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 696–704.

[24] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one* 16, 7 (2021), e0254841.

[25] Chuhan Jiao, Yao Wang, Guanhua Zhang, Mihai Bâce, Zhiming Hu, and Andreas Bulling. 2024. DiffGaze: A Diffusion Model for Continuous Gaze Sequence Generation on 360 {\deg} Images. *arXiv preprint arXiv:2403.17477* (2024).

[26] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339* (2018).

[27] Antti Keurulainen, Isak Rafael Westerlund, Oskar Keurulainen, and Andrew Howes. 2023. Amortised experimental design and parameter estimation for user models of pointing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.

[28] Markus Klar, Florian Fischer, Arthur Fleig, Miroslav Bachinski, and Jörg Müller. 2023. Simulating Interaction Movements via Model Predictive Control. *ACM Transactions on Computer-Human Interaction* 30, 3 (2023), 1–50.

[29] Saskia Koldijk, Mark Van Staalduinen, Mark Neerincx, and Wessel Kraaij. 2012. Real-time task recognition based on knowledge workers' computer activities. In *Proceedings of the 30th European Conference on Cognitive Ergonomics*. ACM, Edinburgh, 152–159.

[30] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical review E* 69, 6 (2004), 066138.

[31] Tiffany CK Kwok, Eugene Yujun Fu, Erin You Wu, Michael Xuelin Huang, Grace Ngai, and Hong-Va Leong. 2018. Every little movement has a meaning of its own: Using past mouse movements to predict the next interaction. In *23rd International Conference on Intelligent User Interfaces*. ACM, Berlin, 397–401.

[32] Haotian Li, Yong Wang, Aoyu Wu, Huan Wei, and Huamin Qu. 2022. Structure-aware visualization retrieval. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans, 1–14.

[33] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021. Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Virtual, 1–15.

[34] Yuening Li, Zhengzhang Chen, Daochen Zha, Mengnan Du, Jingchao Ni, Denghui Zhang, Haifeng Chen, and Xia Hu. 2022. Towards learning disentangled representations for time series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3270–3278.

[35] Yanxi Li, Derek S Young, Julien Gori, and Olivier Rioul. 2024. A novel mixture model for characterizing human aiming performance data. *Statistical Modelling* (2024), 1471082X241234139.

[36] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement Learning on Web Interfaces using Workflow-Guided Exploration. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1802.08802

[37] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. 2024. Latent diffusion for language generation. *Advances in Neural Information Processing Systems* 36 (2024).

[38] Hee-Seung Moon, Seungwon Do, Wonjae Kim, Jiwon Seo, Minsuk Chang, and Byungjoo Lee. 2022. Speeding up inference with user simulators through policy modulation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–21.

[39] Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Siqing Li, and Ji-Rong Wen. 2021. Knowledge-guided disentangled representation learning for recommender systems. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–26.

[40] Christopher Murphy, Jiaju Huang, Daqing Hou, and Stephanie Schuckers. 2017. Shared dataset on natural human-computer interaction to support continuous authentication research. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 525–530.

[41] Lilian Ngweta, Subha Maity, Alex Gittens, Yuekai Sun, and Mikhail Yurochkin. 2023. Simple disentanglement of style and content in visual representations. In *International Conference on Machine Learning*. PMLR, 26063–26086.

[42] Phillip T Pasqual and Jacob O Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto, 743–752.

[43] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. 2022. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10619–10629.

[44] Md Muhaimenur Rahman and Sarnali Basak. 2021. Identifying user authentication and most frequently used region based on mouse movement data: A machine learning approach. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, Virtual, 1245–1250.

[45] Hansol Rheem, Vipin Verma, and D Vaughn Becker. 2018. Use of mouse-tracking method to measure cognitive load. In *Proceedings of the human factors and ergonomics society annual meeting*. SAGE Publications Sage CA: Los Angeles, CA, Los Angeles, 1982–1986.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing*

and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, 234–241.
[47] Sergio Salmeron-Majadas, Ryan S Baker, Olga C Santos, and Jesus G Boticario. 2018. A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios. *IEEE Access* 6 (2018), 39154–39179.
[48] Changhao Shi, Sivan Schwartz, Shahar Levy, Shay Achvat, Maisan Abboud, Amir Ghanayim, Jackie Schiller, and Gal Mishne. 2021. Learning disentangled behavior embeddings. *Advances in neural information processing systems* 34 (2021), 22562–22573.
[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
[50] Jie Su, Zhenyu Wen, Tao Lin, and Yu Guan. 2022. Learning disentangled behaviour patterns for wearable-based human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–19.
[51] Zixiong Su, Shitao Fang, and Jun Rekimoto. 2023. LipLearner: Customizable Silent Speech Interactions on Mobile Devices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg, 1–21.
[52] Yan Sun, Hayreddin Ceker, and Shambhu Upadhyaya. 2016. Shared keystroke dataset for continuous authentication.
[53] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* 34 (2021), 24804–24816.
[54] Emanuel Todorov and Michael I Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature neuroscience* 5, 11 (2002), 1226–1235.
[55] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2words: Automatic mobile UI summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual, 498–510.
[56] Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. 2022. Disentangled representation learning. *arXiv preprint arXiv:2211.11695* (2022).
[57] Zhizhong Wang, Lei Zhao, and Wei Xing. 2023. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7677–7689.
[58] Yuxin Wu and Kaiming He. 2018. Group Normalization. arXiv:1803.08494 [cs.CV]
[59] Pingmei Xu, Yusuke Sugano, and Andreas Bulling. 2016. Spatio-temporal modeling and prediction of visual attention in graphical user interfaces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose, 3299–3310.
[60] Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. 2023. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *arXiv preprint arXiv:2301.13721* (2023).
[61] Zihan Yang, Richard O Sinnott, James Bailey, and Qiuhong Ke. 2023. A survey of automated data augmentation algorithms for deep learning-based image classification tasks. *Knowledge and Information Systems* 65, 7 (2023), 2805–2861.
[62] Zebin You, Yong Zhong, Fan Bao, Jiacheng Sun, Chongxuan Li, and Jun Zhu. 2024. Diffusion models and semi-supervised learners benefit mutually with few labels. *Advances in Neural Information Processing Systems* 36 (2024).
[63] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16010–16021.
[64] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a mobile device into a mouse in the air. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 15–29.
[65] Guanhua Zhang, Matteo Bortoletto, Zhiming Hu, Lei Shi, Mihai Bâce, and Andreas Bulling. 2023. Exploring Natural Language Processing Methods for Interactive Behaviour Modelling. In *The Proceeding of 2023 IFIP TC13 Conference on Human-Computer Interaction (INTERACT)*. IFIP, York, 1–18.
[66] Guanhua Zhang, Susanne Hindennach, Jan Leusmann, Felix Bühler, Benedict Steuerlein, Sven Mayer, Mihai Bâce, and Andreas Bulling. 2022. Predicting Next Actions and Latent Intents during Text Formatting. In *Proceedings of the CHI Workshop Computational Approaches for Understanding, Generating, and Adapting User Interfaces*. ACM, New Orleans, 1–6.
[67] Guanhua Zhang, Zhiming Hu, Mihai Bâce, and Andreas Bulling. 2024. Mouse2Vec: Learning Reusable Semantic Representations of Mouse Behaviour. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. 1–17. https://doi.org/10.1145/3613904.3642141
[68] Wenrui Zhang, Ling Yang, Shijia Geng, and Shenda Hong. 2023. Self-Supervised Time Series Representation Learning via Cross Reconstruction Transformer. *IEEE Transactions on Neural Networks and Learning Systems* 1, 1 (2023), 1–11.
[69] Chen Zhao, Le Wu, Pengyang Shao, Kun Zhang, Richang Hong, and Meng Wang. 2023. Fair Representation Learning for Recommendation: A Mutual Information Perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, Washington DC, 4911–4919.

A HANDCRAFTED FEATURES

Table 6 presents the 75 handcrafted features we extracted from the mouse movement data to serve as a reference for the performance of user identification (see Table 2), as well as next activity prediction and task recognition (see Table 4). We opted for these specific features because they have been widely used in a range of existing data-driven mouse movement models.

B COMPARISON WITH EXISTING TASK-SPECIFIC METHODS

Following [67], we implemented existing methods that were specifically designed for next activity prediction, task recognition and user identification. Table 7 shows that DisMouse representations consistently achieved the best results across datasets and classification tasks. This confirmed the effectiveness of the concurrent feature refinement allowed by DisMouse.

Statistic	Description	References
	X and Y coordinate	[15, 31]
Mean,	Travel distance	[1, 13–15, 47]
Median,	Straight distance	[14]
Maximum,	X, Y, angular and total speed	[1, 13–15, 31, 47]
Minimum,	X, Y and total acceleration	[1, 13–15, 31, 47]
Standard deviation	Angle, angle difference	[1, 13–15, 31, 47]
	Jerk	[1]
	Curvature	[1, 15]

Table 6: Handcrafted mouse movement features that have been commonly used in prior mouse modelling research. In our work, we used them for user identification, next activity prediction and task recognition to compare with DisMouse.

Method		Next Activity Prediction
		ACTIVITY
Features & Activities + SVM		64.15±6.41
Features & Activities + RF		65.03±3.68
Features & Activities + LSTM		73.34±3.29
DisMouse	$E_{nonuser}$	63.68±6.16
	$E_{nonuser}$ & Activities	**83.04±3.60**

Method	Task Recognition
	EMAKI
Raw + HMM	40.87±10.56
Raw + CRF	43.65±7.34
Features + NB	47.74±4.90
Features + KStar	55.72±0.83
Features + DT	63.49±2.49
Features + MLP	67.94±0.23
DisMouse $E_{nonuser}$	**71.24±1.81**

Method	User Identification	
	ACTIVITY	EMAKI
Features + kNN	32.79±1.26	24.04±0.17
DisMouse E_{user}	**51.00±1.39**	**37.60±0.40**

Table 7: Accuracies (mean±standard deviation, in percentage) of next activity prediction, task recognition and user identification. We followed [67] and compared DisMouse embeddings with existing methods that were particularly designed for these tasks [5, 10–12, 14, 29, 31, 66]. The best accuracies are shown in bold.