A Novel Approach for Virtual Locomotion Gesture Classification: Self-Teaching Vision Transformer for a Carpet-Type Tactile Sensor

Sung-Ha Lee [†*] Ho-Taek Joo[†*] Insik Chung[†] Donghyeok Park[‡] Yunho Choi[§] Kyung-Joong Kim[§]
Gwangju Institute of Science and Technology

Figure 1: Example figure of the user using the system. The user navigates the VR environment by gesturing in-place on the carpet-type tactile sensor. Our proposed Self-Teaching Vision Transformer (STViT) algorithm infers the locomotion action performed by the user. This inferred action is then reflected in the avatar's movement within the VR environment. We designed the carpet-type tactile sensor interface for action recognition and proposed a STViT model that can effectively process high-resolution sensor values to enhance the user's VR experience.

ABSTRACT

Locomotion gesture classification in virtual reality (VR) is the process of analyzing and identifying specific user movements in the real world to navigate virtual environments. However, existing methods often necessitate the use of wearable sensors, which present limitations. To address this, we utilize a high-resolution carpet-type tactile sensor as a foot action recognition interface, which was previously unexplored in the context of locomotion gesture classification. This interface can capture the user's foot pressure data in detail to distinguish similar actions. In this paper, to efficiently process captured user's foot tactile data and classify nuanced actions, we utilize a Vision Transformer (ViT) architecture and propose a novel Self-Teaching Vision Transformer (STViT) model integrating elements of the Shifted window Vision Transformer (SwinViT) and Data-efficient image Transformer (DeiT). However, unlike DeiT, our model uses itself from N-steps prior as the teacher model, which is continuously updated. Therefore, improving the ability to classify actions by referencing its own knowledge from previous training stages progressively refines its understanding of similar action gestures. Also, we used the base architecture of SwinViT to utilize patch merging, which improves the ability to differentiate between variations in similar actions by capturing information at different scales. We evaluated seven vision-based methods, demonstrating promising results. Not only did our model outperform ResNet by 19.6%, but it also outperformed each Deit and SwinViT by 3.3% and 2.9%, achieving 92.7% accuracy. To validate our model's real-world applicability, we conducted user preference tests and in-game performance evaluations with 18 participants. As a result, the participants preferred our model to SwinViT and DeiT, backing up the computational results. The video demonstrating the VR testing for STViT can be found in `https://youtu.be/NJslvanRn18`

Index Terms: Human-centered computing—Human–computer interaction (HCI)—Interaction techniques—Gestural input; Human-centered computing—Human–computer interaction (HCI)—Interaction paradigms—Virtual reality

1 INTRODUCTION

Locomotion in virtual reality (VR) refers to the simulation of movement that enables users to navigate through a virtual environment. Virtual locomotion is a fundamental component that significantly contributes to the immersive and interactive nature of the VR environment [15, 4]. Various studies have focused on exploring this field, introducing diverse techniques [35, 45]. However, despite the extensive research on locomotion methods [3], natural virtual locomotion continues to be a challenging domain [32, 33]. One of the primary difficulties arises due to the physical limitations of real-world space, which is unlike its virtual counterpart [34].

In an attempt to address the space constraints of limited real-world space, various in-place locomotion methods have been introduced [23, 24, 17]. These methods involve users performing in-place gestures, such as walking in place, jumping, or sliding, and translating these physical movements into a virtual world using various sensors [54, 13].

In the field of classifying in-place gestures, Shi *et al.* [39] provided a promising foundation for navigating VR environments. Their research was centered on distinguishing between locomotion actions using smart insoles equipped with three pressure sensors and a microchip, proposing dual-check till consensus (DCTC) to improve upon long short-term memory (LSTM). Recently, the in-place locomotion method proposed by Zhao *et al.* [55] implemented a point cloud transformer fused with unsupervised domain adaptation in image learning. Zhao *et al.* [54] introduced the long-term memory

*Sung-Ha Lee and Ho-Taek Joo contributed equally to this work.
†e-mail:{shlee0414,hotaek87,ischung1184}@gm.gist.ac.kr
‡e-mail: skypia0906@gmail.com
§e-mail: {cyh,kjkim}@gist.ac.kr

augmented network to recognize in-place gestures accurately and quickly. Both studies used VIVE trackers attached to the user's thigh to classify locomotion gestures. However, the use of such wearable sensors can lead to discomfort as they disturb smooth movement [8].

Therefore, to recognize the user's actions without the need for wearable equipment, we utilized the carpet-type tactile pressure sensor developed by Luo *et al.* [30] as a foot action recognition interface, as illustrated in Figure 1. This approach, which is novel in the context of gesture classification, was implemented in VR by Choi *et al.* [9] to detect the speed and angle of the user's movements. This interface was validated as providing a comfortable and natural user experience compared to traditional methods, such as treadmills or inertial measurement unit (IMU) sensors, which require users to wear additional devices. This carpet-type sensor provides high-resolution readings, offering a promising direction for user-friendly and efficient action recognition like [53] in VR environments. Furthermore, its capacity to capture detailed foot pressure data holds the potential for discerning even subtle differences in similar user actions, thereby paving the way for a more nuanced understanding and recognition of user movements in virtual environments.

To process tactile sensing data, we opted to use a vision transformer [12]. Unlike recurrent networks such as LSTM, which were utilized in previous locomotion gesture classification studies [39], transformers tend not to experience memory loss over time. Moreover, compared to convolution neural networks (CNNs), which have been employed in multiple tactile perception studies [28], they have larger receptive fields, enabling a better understanding of the global context when extracting features [16].

To effectively classify in-place locomotion actions using tactile sensors, we propose the self-teaching vision transformer model (STViT), which is a novel architecture designed to utilize the benefits of both the DeiT [43] and SwinViT [29]. Our model incorporates a unique self-distillation technique that promotes the incremental accumulation of knowledge, leading to a stable learning process. This is achieved by using the training model itself from N-steps prior as the 'teacher' model, instead of using the separate pre-trained model as the teacher. Another noteworthy characteristic of our proposed model is the incorporation of patch merging, inspired by SwinViT, to enhance the ability to distinguish similar actions. This approach allows the model to change the size and focus of the areas it pays the most attention to, which helps it to better recognize variations in similar actions by capturing information at different scales.

We compared six different algorithms, and our self-teaching transformer model showed promising results. Our model outperformed ResNet [20] by approximately 19.6%, DeiT by 3.3%, SwinViT by 2.9%, Sequencer [42] by 3.4%, GcViT [19] by 10.5%, and ViViT [2] by 6.1%, achieving an accuracy of 92.7%. To validate our proposed method in the real world, we conducted a user study with 18 participants. The results indicate that participants preferred our method over both SwinViT and DeiT ($p < 0.01$). Also, participants were better able to move as they intended compared to SwinViT and DeiT ($p < 0.05$), thereby validating the real-world performance of our proposed method.

The contributions of this work are as follows:

- Our primary contribution is the proposal of a self-teaching vision transformer model, which is a novel architecture that integrates the benefits of the DeiT and SwinViT. This model features a novel self-distillation technique that promotes incremental knowledge accumulation, leading to a more stable learning process.
- We utilized a carpet-type tactile sensor interface for foot action recognition, a novel application in the field of locomotion gesture classification. This high-resolution sensor captures detailed user foot pressure data, facilitating the differentiation of similar actions without the need to attach a sensor to the user's body.
- Our proposed model was evaluated against six different algorithms, demonstrating the best classification accuracy among them at 92.7%. Furthermore, through real-world user testing, our method was proven to be preferred by users ($p < 0.01$) over SwinViT/DeiT, and it showed a lower error.

2 RELATED WORK

2.1 Vision Transformer

Over recent years, the vision transformer (ViT) [12] has distinguished itself as a notable architecture in the realm of image classification, surpassing traditional CNNs across a range of tasks. A distinctive feature of the ViT architecture lies in its approach to processing images as sequences of tokens, a concept reminiscent of how transformers handle textual data within the scope of natural language processing tasks.

In their operation, ViTs partition an image into fixed-sized patches without overlap, and each patch is linearly embedded into a one-dimensional vector. These vectors, in conjunction with positional embeddings, are subsequently incorporated into a prototypical transformer architecture. By exploiting the self-attention mechanism, ViTs are capable of encapsulating long-range dependencies as well as the global context within images. This capability empowers the model to decipher intricate patterns and high-level features.

In the wake of ViT's introduction, there have been numerous significant extensions and enhancements of the architecture. One such advancement is the data-efficient image transformer (DeiT) [43], which focuses on improving the efficiency of training data through distillation-via-attention. Inspired by the ViT model, DeiT exhibits commendable performance even with a limited volume of data. The application of knowledge distillation and data augmentation techniques facilitates this achievement. Despite utilizing fewer parameters, DeiT demonstrates a performance comparable to that of ViT, proving its effectiveness in scenarios where data availability is constrained in the field of computer vision. In pursuit of increased image processing efficiency, the Swin transformer [29] introduces a hierarchical structure. In addition to utilizing window shuffling and split-attention mechanisms to manage larger images, the Swin transformer outperforms ViT in terms of performance. Notably, the Swin transformer maintains a superior performance even when its model size is smaller than that of ViT.

Here, we propose an approach that demonstrates superior performance through the integration of previously studied architectures, namely SwinViT and DeiT. The method is implemented based on the SwinViT and incorporates knowledge distillation through the utilization of self-teaching techniques.

2.2 Foot-based VR Locomotion

Various approaches have been studied to implement natural locomotion within space-constrained VR environments [1]. In particular, foot-based VR locomotion techniques [37, 38, 50] have been widely researched due to their ability to provide an immersive and intuitive navigation experience within the virtual environment. Moreover, by using the lower body for locomotion, foot-based locomotion methods enable efficient multitasking, freeing up the user's hands for interactions while maintaining continuous movement [9].

One of the most natural ways to navigate VR space on foot is through real walking, where users traverse the VR space in the same manner as in a real space [1, 46, 7]. However, real walking often encounters considerable challenges due to the limitations of physical space. To overcome these spatial constraints, alternative methods, such as redirected walking and walking-in-place, have been proposed and explored [36, 40]. Redirected walking subtly manipulates the user's virtual route through rotation and translation,

enabling navigation within larger virtual spaces despite a confined physical area [36, 41]. Walking-in-place allows users to navigate larger virtual space by simulating movement in place [40].

To implement these locomotion strategies, various foot-based devices have been explored to capture the movement speed, direction, and gesture of users. Darken *et al.* [11] studied the omni-directional treadmill, a device that enables users to walk freely in any direction [34]. Shi *et al.* [39] utilized smart insoles equipped with pressure sensors and microchips to classify foot gestures for locomotion. Willich *et al.* [48] introduced a foot-based teleportation technique using a sole that detects the 3D position and pressure of the user's feet. While most foot-based locomotion methods necessitate wearing or attaching sensors to the body, Choi *et al.* [9] recently proposed a high-resolution carpet-type tactile sensor system, which can detect the continuous movement speed and direction of users without the need for any sensors attached to the body. Inspired by this, we adapted the tactile sensor for our foot-based gesture recognition interface.

2.3 Deep Learning for VR Locomotion

Deep learning has been utilized in VR Locomotion for accurate motion recognition. Hanson *et al.* [18] utilized a CNN to distinguish between standing and walking in real time using accelerometer data from a head-mounted display (HMD). Shi *et al.* [39] presented the dual-check till consensus (DCTC), an LSTM-based model that classifies seven types of foot movements. DCTC efficiently minimizes latency by adjusting the sequence length of time series pressor data in accordance with classification probability.

Zhao *et al.* [55] introduced an end-to-end gesture classification framework using point cloud data collected through the HMD and two trackers. In the proposed framework, the point cloud learning model classifies the motion, and the unsupervised domain adaptation module minimizes the accuracy difference caused by various body characteristics of users. Their next study [54] proposed the long-term memory augmented network (LMAN) architecture, a combination of the memory argument network and LSTM for real-time in-place gesture classification. LMAN stores long-term features, abundant with information, in a memory queue during the learning phase. During inference, it extracts short-term features and retrieves the most similar long-term feature from the memory queue. This approach effectively reduces the number of input sequences required by LSTM, thereby shortening inference time while maintaining high accuracy. These methods essentially utilize LSTM to classify more actions in VR locomotion.

3 METHOD

3.1 Interface for Gesture Recognition

For the gesture recognition interface, we propose the use of carpet-type tactile pressure sensors developed by Luo *et al.* [30] to acquire the foot pressure data of users. This is the first time the carpet-type tactile sensor interface is used for VR locomotion gesture recognition. Each tile within the sensing carpet was constructed by positioning electrodes orthogonally on both surfaces of piezoresistive films. Each intersection of these orthogonally positioned electrodes acts as a pressure-sensing point. In this paper, we employed a thin copper film as the electrode. The sensing resolution of a single sensor tile is 32×32 (1024 nodes), and the size is $60cm\times60cm$. We arranged four sensor tiles in a square shape for the experiment, resulting in a total sensor area of $120cm\times120cm$ and resolution of 64×64 (4096 nodes), which extends to $64\times64\times8$ when considering the window size (M).

The selection of this sensor was motivated by its capacity to enhance user comfort as it doesn't necessitate wearing any sensor devices other than an HMD [9]. Moreover, the high spatial resolution provided by the tactile sensor, which captures the human footprints in detail, allows for the recognition of similar locomotion actions.

3.2 Self-Teaching Vision Transformer (STViT)

3.2.1 Overview

The Vision Transformer (ViT) has recently emerged as a strong competitor in image classification, surpassing conventional Convolutional Neural Networks (CNNs). Unlike CNNs, which operate with local receptive fields, which refer to the specific regions in the input data that a neuron or filter is sensitive to and can detect features from, that expand progressively, ViT employs a global receptive field. This global perspective stems from the ViT's self-attention mechanism, allowing every image segment to reference all others, independent of their spatial proximity. This facilitates a more comprehensive context understanding during feature extraction [16].

This paper introduces the Self-Teaching Vision Transformer (STViT), a further development of ViT. STViT is designed to amplify Virtual Reality (VR) experiences by discerning subtle action differences, such as distinguishing between marching and walking. Precise recognition of these subtle distinctions is pivotal in emulating the intricate dynamics of real-world movement, thereby heightening the authenticity and engagement of VR experiences.

To intensify VR immersion, the STViT model incorporates the distinct advantages of both SwinViTs and DeiT. Initially, STViT incorporates dynamic patch partitioning derived from SwinViT. Instead of commencing with static image patches, STViT utilizes a tiered structure that processes patches of varying sizes at different network depths. This adaptive approach aids in the nuanced detection of features across scales, sharpening its ability to perceive subtle action shifts. In addition to this, STViT integrates knowledge distillation [21], drawing inspiration from the DeiT's teacher-student approach. However, STViT diverges from DeiT's approach where a CNN-based "teacher" model is trained and its knowledge is then distilled to a ViT-based "student" model. Instead, STViT introduces a unique self-distillation method.

While STViT retains the traditional teacher-student architecture, what sets it apart is its progression. In the STViT setting, the teacher model is the student model trained before N-steps. This implies that the previously trained model distills its knowledge to the currently succeeding model and the STViT's self-teaching mechanism capitalizes on iterative knowledge reinforcement. It promotes an environment where the model constantly refines its feature understanding of similar actions. In this setup, the teacher model is initially trained with labeled data. After this phase, the student model learns from the teacher model's logits as well as its internal hidden states and attention values. Once the training of the student model is complete, it becomes the new teacher model. This new teacher model then undergoes a retraining process from labeled data. By using this self-reinforcing training paradigm, STViT achieves a more streamlined and efficient training process, eliminating the need for external teacher models. This technique supports continuous knowledge retention and progressive knowledge enhancement, paving the way for a more stable learning curve.

3.2.2 STViT's Base Model (SwinViT)

Figure 2 illustrates the architecture of STViT, which is fundamentally built upon SwinViT and integrates the student-teacher distillation methodology. SwinViT [29] enhances the principles of ViT [12] by introducing specific improvements for superior performance.

Vision Transformer: Typically, the ViT [12] architecture incorporates multi-head self-attention (MSA) blocks, which extended forms of self-attention (SA). In the SA mechanism, three key projections are derived, namely: query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices. These projections are dimensioned as $\in \mathbb{R}^{E\times kD}$, where E represents the token count, D signifies the patch embedding dimensions, and k corresponds to the number of heads. To compute the self-attention mechanism for each position in the sequence, a weighted combination of the value vectors is used. The weights or "attentions", denoted by $\mathbf{A}$, are based on the pairwise similarities

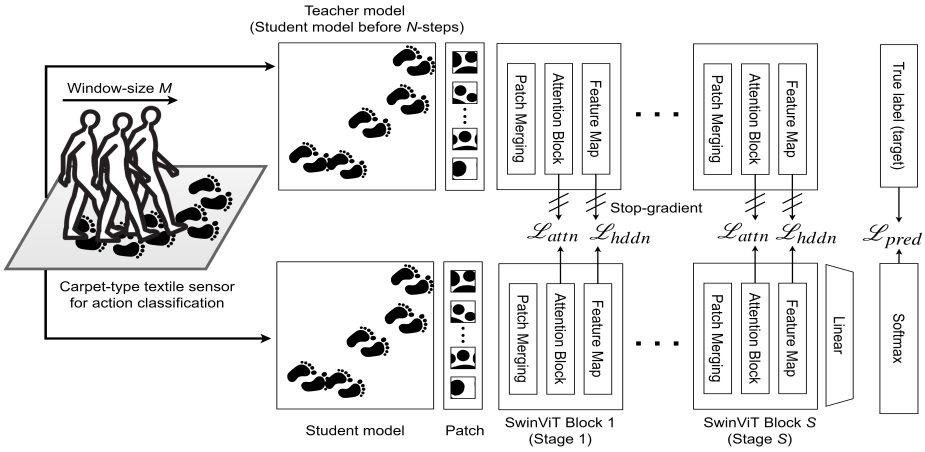


Figure 2: Architecture of our self-teaching vision transformer (STViT). This architecture represents a proposed approach for distinguishing locomotion actions. A vital component, the carpet-type tactile pressure sensors, includes a pressure sensor; the design of this pressure sensor allows for action differentiation, considering both the location of the foot and the varying pressure exerted by the foot. When a participant in a VR experience performs actions on the carpet-type tactile sensors for a duration of "window-size M," the model trained via the proposed method should be capable of distinguishing the action. The proposed model consists of a teacher model and a student model, with the teacher model being the student model from N-steps prior. The proposed method not only considers the cross-entropy loss between the target and prediction but also adds attention loss and hidden loss from the teacher model.

among sequence elements. This operation can be defined by the following equation:

$$\mathbf{h} = \mathbf{AV}, \quad \mathbf{A} = softmax\left(\frac{\mathbf{QK}^T}{\sqrt{D}}\right) \tag{1}$$

Here, the $softmax(\cdot)$ is applied to each k^{th} head, followed by the concatenation of all head outputs through a fully connected layer. Furthermore, the ViT's head output $\mathbf{H}$, formed by combining outputs from all attention heads, enhances feature analysis. Also, the value vector $\mathbf{V}$ captures the diverse elements of the input data for efficient feature integration.

Swin Vision Transformer: As depicted in Figure 2, the SwinViT's hierarchical structure comprises successive "stages", reminiscent of hierarchical patterns in some CNNs. Each stage sees the SwinViT execute a "patch merging" operation, amalgamating neighboring patches into larger ones, effectively halving the resolution. This approach adeptly amalgamates diverse image sections' information. In the elevated stages, consequent to patch merging, a broader image expanse is enveloped, enhancing overall image context comprehension. Conversely, the preliminary stages focus on narrower segments, aiding the detection of intricate patterns. Thus, the SwinViT's tiered approach capitalizes on patch merging at every stage, refining its grasp on the image's diverse intricacies.

The SwinViT excels in differentiating akin actions via tactile sensors, as it can evaluate action characteristics across varied patch sizes. This capability allows for analysis of the entire foot expanse or specific foot-sole pressure points during action differentiation, considering elements like stride and foot pressure.

3.2.3 Objective Function on STViT

In this paper, we propose a novel approach that leverages the distillation technique to bolster the capability of detecting intricate patterns throughout various stages of SwinViT. The proposed method incorporates distilling the attention block and hidden state from a teacher model into a student model within the SwinViT framework. This suggests that the distillation of the attention block and hidden state from the teacher model to the student model afford a more nuanced update of the patch merging process. In our method, the teacher model is the student model from N-steps prior. In particular, our method entails a more nuanced execution of patch merging procedures across several stages of SwinViT, thereby markedly boosting the model's representational capacity. This strategy effectively navigates the student model's learning process by harnessing the knowledge contained within the teacher model as a previous iteration of the student model.

Distillation of Self-Attention on Transformer: Recent literature has demonstrated the efficacy of employing attention maps within transformer layers to guide the training process of student models, as evidenced in works such as [6, 26, 27, 52]. Drawing inspiration from [49], we chose to apply cross-entropy (CE) losses on the relationships among queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$) in the MSA mechanism. More specifically, our initial step involves the concatenation of matrices across all heads. For instance, we define $\mathbf{Q} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_k] \in \mathbb{R}^{E \times kD}$, and $\mathbf{K}$, $\mathbf{V} \in \mathbb{R}^{E \times kD}$ are defined in a similar manner. To simplify the notation, we employ $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{C}_3$ to denote $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively. Following this, we generate nine distinct relation matrices defined as $\mathbf{R}_{ij} = \text{softmax}(\mathbf{C}_i\mathbf{C}_j^T/\sqrt{kD})$. The self-attention distillation loss can be articulated as follows:

$$\mathscr{L}_{attn} = \frac{1}{9E}\sum_{n=1}^{E}\sum_{\substack{i\in\\ \{1,2,3\}}}\sum_{\substack{j\in\\ \{1,2,3\}}} CE(\mathbf{R}^s_{ij,n}, \mathbf{R}^t_{ij,n}) \tag{2}$$

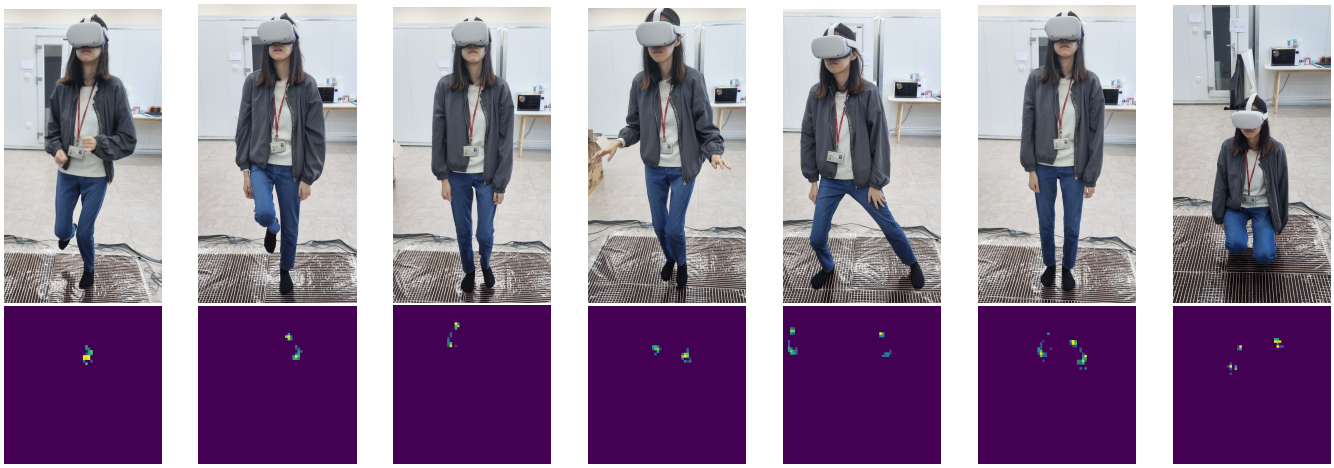
where $\mathbf{R}^s_{ij,n}$ and $\mathbf{R}^t_{ij,n}$ represent the n^{th} rows of $\mathbf{R}_{ij}$ in the student and teacher models, respectively.

Distillation of Hidden States In this research, we define hidden states as each value between SwinViT Blocks. More specifically, the feature maps between these SwinViT stages are referred to as hidden states. The proposed method involves a process of distilling these hidden states from the teacher model $\mathbf{H}^t$ to the student model $\mathbf{H}^s$. At the end of each S stage, we utilize the CE loss function as defined by the following equation:

$$\mathscr{L}_{hddn} = \frac{1}{E}\sum_{n=1}^{E} CE(\mathbf{H}^s_n, \mathbf{H}^t_n) \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{E \times D}$.

Prediction of Label-Logits The final component of the loss function is the CE loss between the true label and the prediction. This is effectively the loss incurred when images of window size M are



(a) running (b) marching (c) walking (d) sneaking (e) sliding (f) standing (g) crouching

Figure 3: Locomotion actions and pressure images for each action

input into the STViT model, comparing the difference between the model's prediction and the true label of the data.

$$\mathscr{L}_{pred} = CE(\mathbf{z}_s, \mathbf{y}) \tag{4}$$

In this equation, $\mathbf{z}_s$ and $\mathbf{y}$ represent the logits inferred by the student model and true label (target), respectively. Finally, we add the CE loss between the predicted and true labels. Consequently, we calculate the final objective function as

$$\mathscr{L}_{train} = \alpha\mathscr{L}_{attn} \times S + \beta\mathscr{L}_{hddn} \times S + \gamma\mathscr{L}_{pred} \tag{5}$$

In this formulation, the term S denotes the number of stages, whereas α, β, and γ are representative of hyperparameters, all of which are assigned a default value of 1.

4 Experiments

4.1 Implementation Details

Data were collected from a total of 10 participants, from which we designated the data from 8 randomly selected participants as our training set and the remaining 2 for the test set. During the training, rotation augmentation is performed to recognize gestures regardless of their orientation. The inference time was approximately 0.013 seconds on an NVIDIA RTX 3070 GPU.

Our model was trained for 40 epochs five times, with a batch size of 8. The learning rate was set to 0.00005, the window size (M) was 10, and the weight decay was set to 0.00001. The teacher model was a student model from 200 steps (N) prior. The patch size and embed dimension, parameters used in the ViT, were set to 2 and 192, respectively. Decreasing patch size improves results but increases model complexity [44]. We opted for a smaller patch size than is generally used, taking into account that pressure images are smaller than typical photographic images, which are commonly the target for processing in a ViT. This consideration is aimed at retaining more detailed information from the pressure images for better gesture recognition. More detailed hyper-parameters can be found in Appendix B.

4.2 Action Selection

We selected a total of seven gestures: running, marching, walking, sneaking, sliding, standing, and crouching as shown in Figure 3. Five of these gestures (walking, sliding, standing, marching, and running) were chosen based on existing VR locomotion gesture recognition research [39, 54, 22]. Here, marching is an action akin to walking, but it involves raising the knees to a higher level. The sneaking was included because of the potential for immersive experiences [10], while crouching was selected due to its usages in exergames [14,51]. Notably, the inclusion of sneaking, marching, and crouching aimed to test our system's ability to distinguish subtle differences in pressure patterns by involving similar actions. For example, marching produces similar foot pressure distributions to walking than running, as can be seen in Figure 3, making marching/walking more challenging to classify. This selection of gestures is intended to validate our model's ability to classify similar actions.

4.3 Data Collection

We collected foot pressure data from 10 participants using carpet-type pressure sensors. 8 participants were male, and 2 were female. The age of the participants in our sample ranged from 20 to 31 years (Mean= 27, SD= 3.24). The height range was 163 to 186 *cm* (Mean=172.5, SD= 6.82), and the weight range was 52 to 85 kg (Mean= 64.5, SD= 9.43). The foot size range was 255 to 280 *mm* (Mean= 265, SD= 11.65).

The participants were asked to perform 7 tasks: walking, sneaking, marching, running, sliding, crouching, and standing. Each task was performed for two minutes on the carpet-type pressure sensor, except for sliding, which was performed for four minutes - two minutes of left sliding and two minutes of right sliding. The total frames of the dataset were approximately 74,654.

4.4 Model Evaluation

Table 1: Accuracy comparison of different models (%)

Model	Model Type	Mean	Standard Deviation
ResNet18 [20]	CNN	73.1	2.1
Sequencer [42]	CNN + LSTM	89.3	0.7
GcViT [19]	ViT	82.2	1.4
DeiT [43]	ViT	89.4	3.2
SwinViT [29]	ViT	89.8	1.3
ViViT [2]	ViT	86.6	2.2
STViT (Ours)	ViT	**92.7**	1.1

4.4.1 Comparisons with Other Methods

In this section, we present a comparative analysis of various vision-based AI models, ranking them based on their accuracy, from highest to lowest. Each model was trained for 40 epochs, and the accuracy is averaged over five runs.

On the lower side of the accuracy scale, we find ResNet18, a model based on the CNN framework, posting an accuracy of 73.1%.

Despite its position at the bottom in terms of accuracy among the models discussed, ResNet18 is a well-established model and sets a benchmark for newer models to compare against. The GcViT model, designed around the Vision Transformer (ViT) architecture, marks an accuracy of 82.2%, demonstrating the effectiveness of transformer architectures in visual tasks. Following this, the ViViT model, built on the ViT architecture to process video, exhibits an accuracy of 86.6%. Closely following is the Sequencer model, which combines a CNN with Long Short-Term Memory (LSTM) structures, and achieves an accuracy of 89.3%. DeiT, a model that uses knowledge distillation, records an accuracy of 89.4%. This underscores how using a teacher model as a teaching guide can result in competitive outcomes. The SwinViT model, another implementation based on the ViT framework, registers an accuracy of 89.8%. Finally, topping the list is the proposed STViT model, standing out with the highest accuracy of 92.7%. This model blends the strong points of DeiT and SwinViT, highlighting the strength of our approach among a range of frequently used models.

Table 2: Accuracy of locomotion actions for the top three models

Model	runn-ing	march-ing	walk-ing	sneak-ing	slide-ing	stand-ing	crouch-ing
DeiT	99.9	98.5	**68.6**	**71.3**	**88.9**	99.9	99.5
SwinViT	99.5	**88.8**	**65.5**	**77.3**	99.8	99.9	99.9
STViT	97.9	95.9	**76.4**	**81.7**	98.2	99.5	99.6
Average	99.1	93.1	**70.2**	**76.8**	95.6	99.8	99.7

Table 2 presents a detailed analysis of the top three models in Table 1—DeiT, SwinViT, and STViT—across various locomotion actions, highlighting their strengths and areas for improvement.

These models excel at categorizing actions like running, standing, and crouching, with average accuracies over 99%. Conversely, performance declines for 'walking' and 'sneaking' actions, with average accuracies of 70.2% and 76.8% respectively. However, STViT stands out, achieving superior accuracy scores of 76.4% for walking and 81.7% for sneaking than other models.

Low average accuracy for 'walking' reveals frequent misidentification of 'walking' as 'marching' and vice versa, signifying the challenge in differentiating these actions, which are 'walking' and 'sneaking,' due to similarities in foot pressure distribution. Likewise, confusion arises for the 'sneaking' category, with the SwinViT model often mislabeling 'sneaking' as 'marching', and the DeiT model mistaking 'sneaking' for 'running'.

The SwinViT model's tendency to confuse 'sneaking' and 'marching' could arise from its reliance on local-spatial dependencies, which can overlook subtle differences in actions, such as the height of foot lift-off or the degree of body tilt in 'sneaking'. This overemphasis on local-spatial patterns can result in such errors.

For the DeiT model, its confusion between 'sneaking' and 'running' could be linked to its use of a distillation token to learn global image information. While this mechanism captures rough image features, it might miss nuanced differences in movement speed, fluidity, and stride patterns, all crucial to distinguish 'sneaking' from 'running'. In Figure 2 (a) running and (d) sneaking, the participant acts using only the forefoot.

While both SwinViT and DeiT have specific weaknesses—with SwinViT struggling to distinguish marching, and DeiT underperforming in identifying sliding actions—these weaknesses can be attributed to their emphasis on local-spatial dependencies and global image representations dependencies, respectively. This suggests the models' trade-off between capturing broader patterns and identifying action details.

In summary, these unique challenges encountered by the SwinViT and DeiT models in recognizing certain types of actions underscore the need for models to balance their focus between global and localized features. Finally, Our model (STViT) could potentially benefit from hybrid approaches that efficiently leverage both global and local contexts, enhancing their ability to recognize and classify a wider range of human actions accurately.

4.4.2 Confusion Matrix for STViT

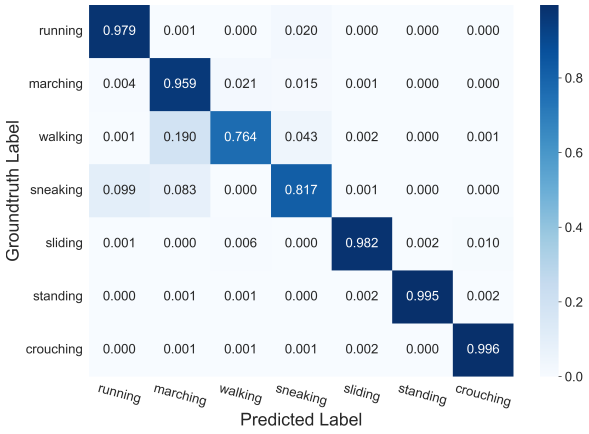


Figure 4: Normalized confusion matrix for STViT

Figure 4 shows the normalized confusion matrix for the concrete information of STViT (Ours) in Table 2. In the confusion matrix, the horizontal axis represents the predicted class, the vertical axis denotes the true label class, and the diagonal reflects accuracy. Compared to the overall accuracy (92.7%) shown in Table 1, walking (81.7%) and sneaking (76.4%) showed much lower accuracy. In particular, the primary cause was the misclassification between walking and marching, sneaking and marching, as well as sneaking and running. Excluding these two motions, the classification accuracy of the remaining motions is 98.2%. The misclassification of walking can be attributed to its feature similarity to marching, and similarly, sneaking is similar to both marching and running, as shown in the feature map of Figure 5.

One thing to note is that both the training and test data for this study were collected from participants without wearing shoes. Therefore, the accuracy of our results might diminish in scenarios where shoes are worn. However, this can be addressed by collecting additional training data with participants wearing shoes.

4.4.3 t-SNE and Analysis of Failure Cases for STViT

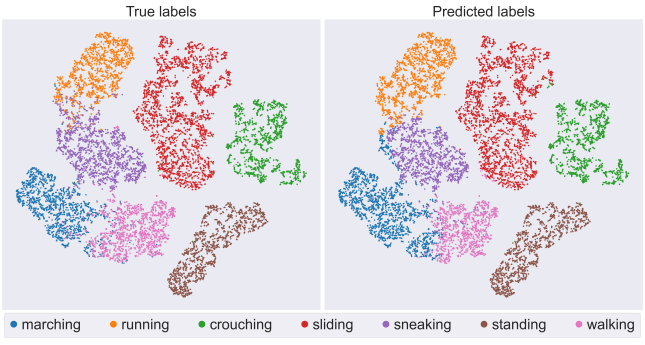


Figure 5: t-SNE visualization for locomotion actions on STViT

By obtaining feature maps before the linear layer on the trained STViT model, we conducted an analysis using t-distributed stochastic neighbor embedding (t-SNE) [47] (Figure 5), which is a machine

learning algorithm that embeds high-dimensional data into lower dimensions. This method allowed us to visualize data patterns.

We input a collected test set into the t-SNE to examine whether the trained model consistently generates a feature map for the same action. The left graph depicts the data output from the t-SNE as labeled data, while the right graph represents the prediction data of the trained model (STViT).

From Figure 5, STViT is generally classifying various locomotion actions well. However, the STViT struggled to differentiate actions such as sneaking and running, marching and walking, and marching and sneaking in Figure 5. In particular, the extent of the cluster by marching (indicated by blue dots) is larger in Figure 5 (Predicted labels) compared to Figure 5 (True labels). This implies the model mistakenly predicted sneaking as marching or walking as marching.

Despite these misclassifications, the proposed model appears to have sufficient potential for performance improvement. This is because sneaking misclassified as marching (blue dots) is close to the sneaking cluster (purple dots), and walking misclassified as marching (blue dots) is close to the walking cluster (pink dots). In other words, Even with these misclassifications, the clusters for sneaking and walking are tightly grouped. Thus, with further refinement of our model, such as fine-tuning the hyperparameters or employing additional training techniques, we can enhance its performance in distinguishing between such closely related actions.

5 User Study

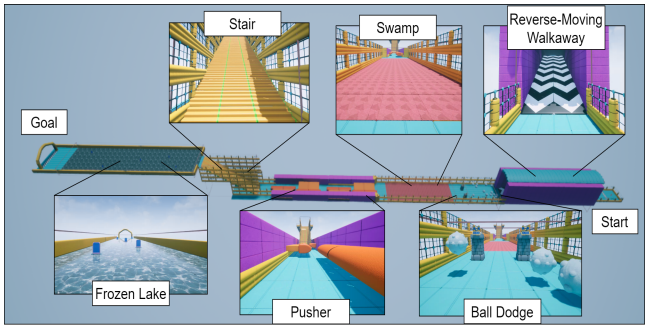


Figure 6: Environmental schematic for the user study

5.1 VR User Test Environment

The test environment, designed as an exergame, was developed using Unreal Engine 4. The objective for the participants was to navigate from the start to the finish line. As shown in Figure 6, the environment was composed of six distinct sections. To successfully pass each section, participants must perform specific locomotion actions. The order and specifications of the sections are as follows:

- A reverse-moving walkway section where participants must run to avoid being pushed backward.
- A ball dodge section, where participants are required to change their direction of movement to evade the balls.
- A swamp section where the participants sink under if they perform non-sneaking actions three times.
- A pusher section, where participants must dodge the approaching pusher by crouching.
- A stair section, where the participants should march, i.e., raise their knees high to climb stairs. If a participant performs an action other than marching, they are unable to move forward.
- A frozen lake section where the participants should slide on ice to move forward. If a participant performs an action other than sliding, they are unable to move forward.

5.2 Participants

We recruited 18 participants. 12 were male, and 6 were female. The age range was from 22 to 30 (Mean=26, SD=2.18), and the height range was from 160 to 186 *cm* (Mean=169.5 *cm*, SD=6.46 *cm*). The weight range was from 47 to 95 *kg* (Mean=66.5 *kg*, SD=11.5 *kg*), and foot size was 230 to 295 *mm* (Mean=260 *mm*, SD=18.3 *mm*). We also evaluated the VR familiarity of participants based on user familiarity through a VR application questionnaire from Shi *et al.* [39]. 4 participants heard about the concept of VR but never used a VR application. 8 participants had hands-on VR experience less than 2 times, and 5 participants had hands-on experience more than 2 times. One participant was an experienced user of VR applications. One participant overlapped with the participants in training data collection.

5.3 Experimental Setup

Experimental Device We used four carpet-type high-resolution pressure sensors for the interface, with a total area of 120 $cm \times 120$ *cm* and a total resolution of 64×64. For the virtual reality setup, we used the Oculus Quest 2 HMD.

Experiment Overview In our locomotion system, the process is as follows: First, as the user performs locomotion actions on a pressure-sensing carpet, foot pressure data are processed by the model. The inferred action value is sent to the VR test environment that we developed, making the VR character move in accordance with the received action. As for controlling the user's direction of movement, we utilized the user's HMD direction, allowing the user to alter their movement direction by turning their head.

The objective of this experiment was to validate our hybrid model in a real-world setting and compare its performance with that of existing models. To achieve this, we used three different action recognition models for comparison: STViT (ours), SwinViT, and DeiT which use trained SwinViT as the teacher model.

Each locomotion action is set to a different movement speed: both crouching and standing have a speed of 0. Sneaking has a speed of 0.3, while walking and marching have a speed of 0.5. sliding has a speed of 0.7, and running is the fastest with a speed of 1. The participant's action type was inferred by the model at every 0.5 s. Each section in the test environment requires a different locomotion action, corresponding to the specific characteristics of these actions.

For the quantitative evaluation, we recorded game log data. We recorded the time taken to navigate through the reverse-moving walkway, ball dodge, and pusher sections. For the swamp, stairs, and frozen lake sections, we counted the number of invalid actions performed (e.g., number of times running in the swamp section). We used different metrics for each section because, in the reverse-moving walkway, ball dodge, and pusher section, participants were not limited to a single action for completion and could perform a variety of actions. Therefore, an invalid action was difficult to define.

Upon completion of the experiment, we conducted a brief interview. First, we requested that participants rank the order of the models they experienced according to their preference and explain their reasons for their rankings. Then, we asked the participants whether they consider that detailed classification of similar actions could enhance their VR experience and requested reasons for their responses.

5.4 Study Procedure

The sequence in which participants experienced the models was determined by a Latin Square design to ensure an equal number of participants experienced the same order of the models. The study procedure was as follows:

- Upon the participants' arrival, we briefed the participants on the purpose of the experiment, overall procedural flow, and locomotion actions required to pass each section. After, we

Table 3: Questions to assess users' subjective experiences. Users rated their responses on a 5-point Likert scale.

Questions
Q 1. I enjoyed it.
Q 2. I felt immersed.
Q 3. I felt a sense of unity with the character.
Q 4. I did not feel any delay.
Q 5. I did not feel any discomfort in control.
Q 6. I was able to move as intended.

demonstrated each locomotion action. Once the briefing concluded, we presented a sample video of a user playing the game using pressure sensors from start to finish.

- The participants wore the HMD and adjusted their equipment. Then, the participants played the game, which took approximately 3 minutes. Throughout the experiment, in-game data were automatically logged. After experiencing a model, participants were given a 2-minute rest period before they tested the next model.
- Upon completion of the experiment, we conducted an interview with the participants.

5.5 Result

Table 4: Results of relative preference model for 18 users. Based on user preference, higher scores correspond to higher preference.

Model	Mean	Standard Deviation	Total
SwinViT	1.78	0.71	32/54
DeiT	1.67	0.82	30/54
STViT	**2.44**	**0.76**	**44/54**

In Table 4, a preference survey was conducted and scores were assigned according to participants' model preferences. In detail, scores of 3, 2, and 1 were assigned in order of preference, and in cases where two or more models had the same preference, the assigned scores were reduced. Statistical analysis was performed using Kruskal-Wallis H [25] and Mann-Whitney U tests [31]. These non-parametric tests were utilized as they allow for robust statistical inference, even when the assumptions of normality are not met. Table 4 presents the results. As assessed by the Kruskal-Wallis H test, there was a significant difference ($p = 0.01$) in the user preferences survey. Additionally, the Mann-Whitney U test revealed a significantly stronger preference for STViT over SwinViT ($p < 0.01$) and DeiT ($p < 0.01$) among users.

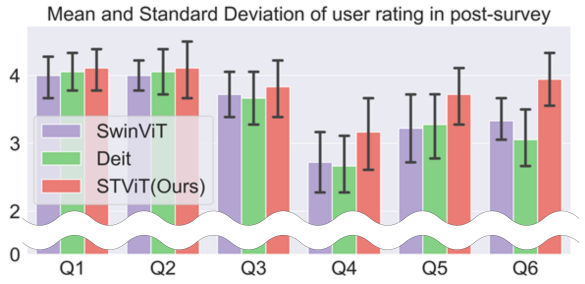


Figure 7: Results of post-survey in Table 3. The graph presents the mean values and standard deviations for each question.

Figure 7 shows the results from the user questionnaire in Table 3. The results show the mean and standard deviation of the user-rated scores on a scale of 1 (strongly disagree) to 5 (strongly agree). In all questions, STViT received the highest scores. Specifically, for Q6 ('I was able to move as intended.'), there was a significant difference between STViT and both SwinViT ($p < 0.05$) and DeiT ($p < 0.01$), as determined by t-test.

Additionally, quantitative evaluation was conducted using game log data as detailed in Section 5.3. The three sections with repeated action tasks (Swamp, Stairs, and Frozen Lake) were evaluated based on the number of invalid actions performed. On the other hand, the three sections (Reverse-moving walkway, Ball Dodge, and Pusher) were evaluated based on the clear time.

The results of the quantitative evaluation were validated using ANOVA and t-tests. The ANOVA results indicated a statistically significant difference in the counts of invalid actions among the three sections - Swamp ($p < 0.05$), Stair ($p < 0.05$), and Frozen Lake ($p < 0.05$). We additionally performed Levene's test [5] to ensure that the homogeneity of variance criterion for ANOVA was met. The results were as follows: Swamp ($p > 0.05$), Stairs ($p > 0.05$), and Frozen Lake ($p > 0.05$). Thus, the homogeneity of variance assumption was satisfied.

We also conducted an ANOVA test related to gender group to determine if there was a difference in performance between male and female participants. The results indicated no significant differences regarding gender across all sections.

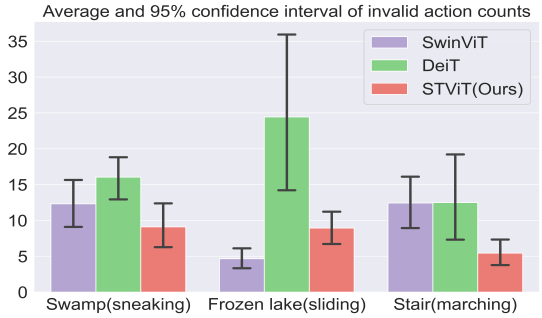


Figure 8: Results of log data collected from each user. The graph shows the average and 95% confidence intervals of the frequency of invalid actions in the section. The text in parentheses next to each section indicates the action to be performed within that section.

Figure 8 shows how each section's invalid action counts are performed. For the Stairs section, the STViT (Mean= 5.4, SD= 3.8) recorded the lowest misclassifications. This was significantly lower than SwinViT (Mean= 12.4, SD= 8.1) and DeiT (Mean= 12.5, SD= 13.5), showing a difference of 7.0 ($t = 2.1, p < 0.05$) and 7.1 ($t = 3.3, p < 0.01$), respectively. In the Swamp section, STViT (Mean= 9.1, SD= 6.3) had 7.0 fewer invalid actions than DeiT (Mean= 16.1, SD= 6.6), a significant difference ($t = 3.2, p < 0.01$). Moreover, STViT had 3.2 fewer actions compared to SwinViT (Mean= 12.3, SD= 7.2). Yet, in the Frozen lake section, SwinViT (Mean= 4.7, SD= 3.0) had the least invalid actions, 4.2 and 19.7 fewer than STViT (Mean= 8.9, SD= 5.4), and Deit (Mean= 24.4, SD= 25.7), respectively. These results suggest that STViT is more accurate in distinguishing hard-to-differentiate actions (e.g., sneaking, marching) than SwinViT and DeiT.

Figure 9 shows the average and 95% confidence intervals of the clear times for each model in the reverse-moving walkaway, ball dodge, and pusher sections. In the reverse walkway and ball dodge section, all models had a similar average clear time, with a less than one-second difference. However, in the pusher section, the STViT exhibited an average clear time of 10 seconds faster than the other models. The Pusher section requires participants to rapidly switch from one action to another. Consequently, the ability to effectively differentiate actions within a shorter time window is important in the Pusher section. Thus, a shorter clear time can be interpreted as

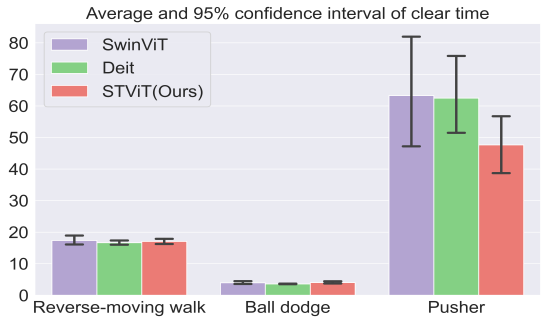


Figure 9: Average and 95% confidence interval of the clear time in the section. The unit of clear time is seconds.

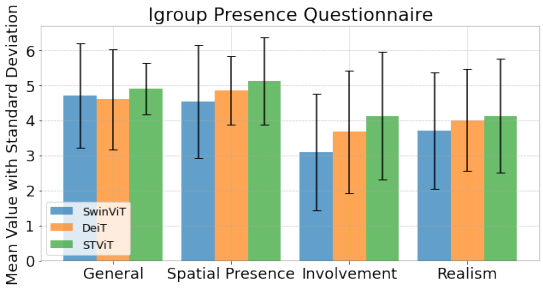


Figure 11: Results of the IPQ. The graph presents the mean values and standard deviations for each question.

STViT being more adept at discerning actions in quick succession and responding more quickly, which is also coherent with the high score of STViT on Q4 ('I did not feel any delay') in Figure 7.

One thing to note is that even though the overall accuracy between DeiT and SwinViT was only 3.3% and 2.9% lower than STViT in section 4.4 (Table 1), in real-world validation, the users showed a clear preference for STViT. This is because while the overall accuracy was similar, in specific actions STViT performed substantially better. For example, in walking, STViT showed 7.8% higher accuracy than DeiT and 10.9% higher accuracy than SwinViT (Table 2). Thus, the users found STViT to be much more reliable.

At the end of the experiment, we interviewed participants about the influence of distinguishing between similar actions on their VR experience. All 18 participants (100%) reported that the ability to discriminate between similar actions enhanced their VR experience. Specifically, 7 participants cited increased immersion as the main reason (e.g., *"I was able to replicate movements that closely resemble my real-life actions, and this has greatly enhanced the sense of immersion in my VR experience."*). 5 participants cited a sense of unity as the main reason (e.g., *"I felt like I'm actually active in the game world. I have a strong sense of unity with my character."*). The remaining reasons were 'increased fun' and 'variety of actions'.

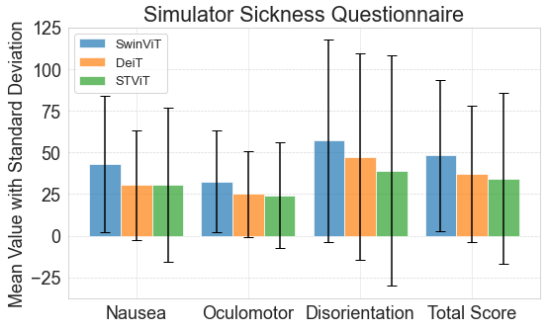


Figure 10: Results of the SSQ. The graph presents the mean values and standard deviations for each question.

We performed the additional experiment with 12 participants. 7 were male, and 5 were female. The age range was from 19 to 29 (Mean=22.3, SD=2.96), and the height range was from 150 to 180 *cm* (Mean=170.7 *cm*, SD=7.67 *cm*). The weight range was from 47 to 95 *kg* (Mean=64.1 *kg*, SD=11.2 *kg*), and foot size was 230 to 295 *mm* (Mean=258 *mm*, SD=17.7 *mm*). We also evaluated the VR familiarity of participants based on user familiarity through a VR application questionnaire from Shi *et al.*. 2 participants heard about the concept of VR but never used a VR application. 6 participants had hands-on VR experience less than 2 times, and 4 participants had hands-on experience more than 2 times.

We conducted ANOVA tests on both SSQ and IPQ. Although there was no significant difference observed between the three models in terms of SSQ scores (Figure 10), as the few outliers significantly increased the value of the standard deviation, STViT recorded the lowest scores in all categories except for Nausea. Regarding the IPQ (Figure 11), STViT scored the highest in all four categories (General Presence, Spatial Presence, Involvement, and Realism). In particular, a significant difference was observed between STViT and SwinViT in terms of Involvement ($p < 0.05$).

In summary, the STViT (Ours) model demonstrated superior accuracy in motion recognition within a VR environment compared to the SwinViT and DeiT models. In particular, the performance improvement was shown in sections requiring differentiation between similar actions. Furthermore, the improved action recognition capabilities of the STViT enhanced the user's VR experience.

6 Conclusions and Future Works

In this paper, we presented a novel method for VR locomotion gesture recognition, leveraging a carpet-type tactile sensor as an interface and proposing a novel STViT algorithm. This interface eliminates the need for wearable equipment, thereby enhancing user comfort and providing a more natural virtual experience. To process tactile data efficiently, we utilized a Vision Transformer and introduced the STViT algorithm. Our model integrates the advantages of both DeiT and SwinViT architectures and incorporates a self-distillation technique. Our model demonstrated improved performance in classifying locomotion actions, as validated by comparisons with six different algorithms and through real-world user testing.

One limitation of our work is that while we developed and tested our model based on a relatively small dataset, vision transformers are generally known to benefit from a large amount of data [12]. Thus, collecting a more diverse or larger quantity of data in the future could prove beneficial in further improving the model's performance.

In the future, we aim to improve the generalization performance of our model other than simply increasing the quantity of training data by implementing a range of data augmentation techniques in addition to the rotation, such as resizing or noise addition. This approach will allow us to increase our model's generalization performance across individuals not included in the training set. Additionally, we plan to integrate headset position data with actions such as crouching, to develop a more reliable system for detecting locomotion actions.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) funded by the MSIT under Grant 2021R1A4A1030075. This work was supported by the GIST-MIT Research Collaboration grant funded by the GIST in 2023. This research was supported by 'Project for Science and Technology Opens the Future of the Region' program through the INNOPOLIS FOUNDATION funded by Ministry of Science and ICT(Project Number: 2022-DD-UP-0312)

REFERENCES

[1] M. Al Zayer, P. MacNeilage, and E. Folmer. Virtual locomotion: a survey. *IEEE transactions on visualization and computer graphics*, 26(6):2315–2334, 2018. 2

[2] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021. 2, 5

[3] I. Bishop and M. R. Abid. Survey of locomotion systems in virtual reality. In *Proceedings of the 2nd International Conference on Information System and Data Mining*, pp. 151–154, 2018. 1

[4] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 annual symposium on computer-human interaction in play*, pp. 205–216, 2016. 1

[5] M. B. Brown and A. B. Forsythe. Robust tests for the equality of variances. *Journal of the American statistical association*, 69(346):364–367, 1974. 8

[6] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. 4

[7] S. S. Chance, F. Gaunet, A. C. Beall, and J. M. Loomis. Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. *Presence*, 7(2):168–178, 1998. 2

[8] H. Cherni, N. Métayer, and N. Souliman. Literature review of locomotion techniques in virtual reality. *International Journal of Virtual Reality*, 2020. 2

[9] Y. Choi, D.-H. Park, S. Lee, I. Han, E. Akan, H.-C. Jeon, Y. Luo, S. Kim, W. Matusik, D. Rus, et al. Seamless-walk: natural and comfortable virtual reality locomotion method with a high-resolution tactile sensor. *Virtual Reality*, pp. 1–15, 2023. 2, 3

[10] S. Cmentowski, A. Krekhov, A. Zenner, D. Kucharski, and J. Krüger. Towards sneaking as a playful input modality for virtual environments. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 473–482. IEEE, 2021. 5

[11] R. P. Darken, W. R. Cockayne, and D. Carmein. The omni-directional treadmill: a locomotion device for virtual worlds. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pp. 213–221, 1997. 3

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. 2, 3, 9

[13] J. P. Freiwald, O. Ariza, O. Janeh, and F. Steinicke. Walking by cycling: A novel in-place locomotion user interface for seated virtual reality experiences. In *CHI*, pp. 1–12, 2020. 1

[14] R. Guo and J. Quarles. Converting sedentary games to exergames: A case study with a car racing game. In *2013 5th International conference on games and virtual worlds for serious applications (vs-games)*, pp. 1–8. IEEE, 2013. 5

[15] K. S. Hale and K. M. Stanney. *Handbook of virtual environments: Design, implementation, and applications*. CRC Press, 2014. 1

[16] Y. Han, R. Batra, N. Boyd, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao. Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer. *arXiv preprint arXiv:2112.06374*, 2021. 2, 3

[17] S. Hanson, R. A. Paris, H. A. Adams, and B. Bodenheimer. Improving walking in place methods with individualization and deep networks. In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*, pp. 367–376. IEEE, 2019. 1

[18] S. Hanson, R. A. Paris, H. A. Adams, and B. Bodenheimer. Improving walking in place methods with individualization and deep networks. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 367–376, 2019. doi: 10.1109/VR.2019.8797751 3

[19] A. Hatamizadeh, H. Yin, J. Kautz, and P. Molchanov. Global context vision transformers. *arXiv preprint arXiv:2206.09959*, 2022. 2, 5

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 2, 5

[21] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[22] P. Ke and K. Zhu. Larger step faster speed: Investigating gesture-amplitude-based locomotion in place with different virtual walking speed in virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 438–447. IEEE, 2021. 5

[23] W. Kim and S. Xiong. User-defined walking-in-place gestures for vr locomotion. *International Journal of Human-Computer Studies*, 152:102648, 2021. 1

[24] J. Kreimeier and T. Götzelmann. First steps towards walk-in-place locomotion and haptic feedback in virtual reality for visually impaired. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–6, 2019. 1

[25] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. 8

[26] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3041–3050, 2023. 4

[27] S. Lin, H. Xie, B. Wang, K. Yu, X. Chang, X. Liang, and G. Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10915–10924, 2022. 4

[28] H. Liu, D. Guo, F. Sun, W. Yang, S. Furber, and T. Sun. Embodied tactile perception and learning. *Brain Science Advances*, 6(2):132–158, 2020. 2

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021. 2, 3, 5

[30] Y. Luo, Y. Li, M. Foshey, W. Shou, P. Sharma, T. Palacios, A. Torralba, and W. Matusik. Intelligent carpet: Inferring 3d human pose from tactile signals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11255–11265, 2021. 2, 3

[31] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947. 8

[32] E. S. Martinez, A. S. Wu, and R. P. McMahan. Research trends in virtual reality locomotion techniques. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 270–280. IEEE, 2022. 1

[33] C. Mousas, D. Kao, A. Koilias, and B. Rekabdar. Evaluating virtual reality locomotion interfaces on collision avoidance task with a virtual character. *The Visual Computer*, 37:2823–2839, 2021. 1

[34] N. C. Nilsson, S. Serafin, F. Steinicke, and R. Nordahl. Natural walking in virtual reality: A review. *Computers in Entertainment (CIE)*, 16(2):1–22, 2018. 1, 3

[35] P. Punpongsanon, E. Guy, D. Iwai, K. Sato, and T. Boubekeur. Extended lazynav: Virtual 3d ground navigation for large displays and head-mounted displays. *IEEE transactions on visualization and computer graphics*, 23(8):1952–1963, 2016. 1

[36] S. Razzaque. *Redirected walking*. The University of North Carolina at Chapel Hill, 2005. 2, 3

[37] M. Schwaiger, T. Thuimmel, and H. Ulbrich. Cyberwalk: An advanced prototype of a belt array platform. In *2007 IEEE International Workshop on Haptic, Audio and Visual Environments and Games*, pp. 50–55. IEEE, 2007. 2

[38] M. C. Schwaiger, T. Thummel, and H. Ulbrich. A 2d-motion platform: The cybercarpet. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pp. 415–420. IEEE, 2007. 2

[39] X. Shi, J. Pan, Z. Hu, J. Lin, S. Guo, M. Liao, Y. Pan, and L. Liu. Accurate and fast classification of foot gestures for virtual locomotion. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 178–189. IEEE, 2019. 1, 2, 3, 5, 7

[40] M. Slater, M. Usoh, and A. Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):201–219, 1995. 2, 3

[41] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas. A taxonomy for deploying redirection techniques in immersive virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 43–46. IEEE, 2012. 3

[42] Y. Tatsunami and M. Taki. Sequencer: Deep lstm for image classification. *Advances in Neural Information Processing Systems*, 2022. 2, 5

[43] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021. 2, 5

[44] H. Touvron, M. Cord, A. El-Nouby, J. Verbeek, and H. Jégou. Three things everyone should know about vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 497–515. Springer, 2022. 5

[45] S. Tregillus, M. Al Zayer, and E. Folmer. Handsfree omnidirectional vr navigation using head tilt. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4063–4068, 2017. 1

[46] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking¿ walking-in-place¿ flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 359–364, 1999. 2

[47] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6

[48] J. von Willich, M. Schmitz, F. Müller, D. Schmitt, and M. Mühlhäuser. Podoportation: Foot-based locomotion in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020. 3

[49] W. Wang, H. Bao, S. Huang, L. Dong, and F. Wei. Minilmv2: Multihead self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2140–2151, 2021. 4

[50] Z. Wang, C. Liu, J. Chen, Y. Yao, D. Fang, Z. Shi, R. Yan, Y. Wang, K. Zhang, H. Wang, et al. Strolling in room-scale vr: Hex-core-mk1 omnidirectional treadmill. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2

[51] D. Watson, R. L. Mandryk, and K. G. Stanley. The design and evaluation of a classroom exergame. In *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, pp. 34–41, 2013. 5

[52] S. Yun, H. Lee, J. Kim, and J. Shin. Patch-level representation learning for self-supervised vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8354–8363, June 2022. 4

[53] J. Zhao, M. Shao, Y. Wang, and R. Xu. Real-time recognition of in-place body actions and head gestures using only a head-mounted display. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 105–114. IEEE, 2023. 2

[54] L. Zhao, X. Lu, Q. Bao, and M. Wang. In-place gestures classification via long-term memory augmented network. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 224–233, 2022. doi: 10.1109/ISMAR55827.2022.00037 1, 3, 5

[55] L. Zhao, X. Lu, M. Zhao, and M. Wang. Classifying in-place gestures with end-to-end point cloud learning. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 229–238, 2021. doi: 10.1109/ISMAR52148.2021.00038 1, 3