JumpMod: Haptic Backpack that Modifies Users' Perceived Jump

Romain Nith
University of Chicago
rnith@uchicago.edu

Jacob Serfaty
University of Chicago
jserfaty@uchicago.edu

Samuel G Shatzkin
University of Chicago
sshatzkin@uchicago.edu

Alan Shen
University of Chicago
alans@uchicago.edu

Pedro Lopes
University of Chicago
pedrolopes@uchicago.edu

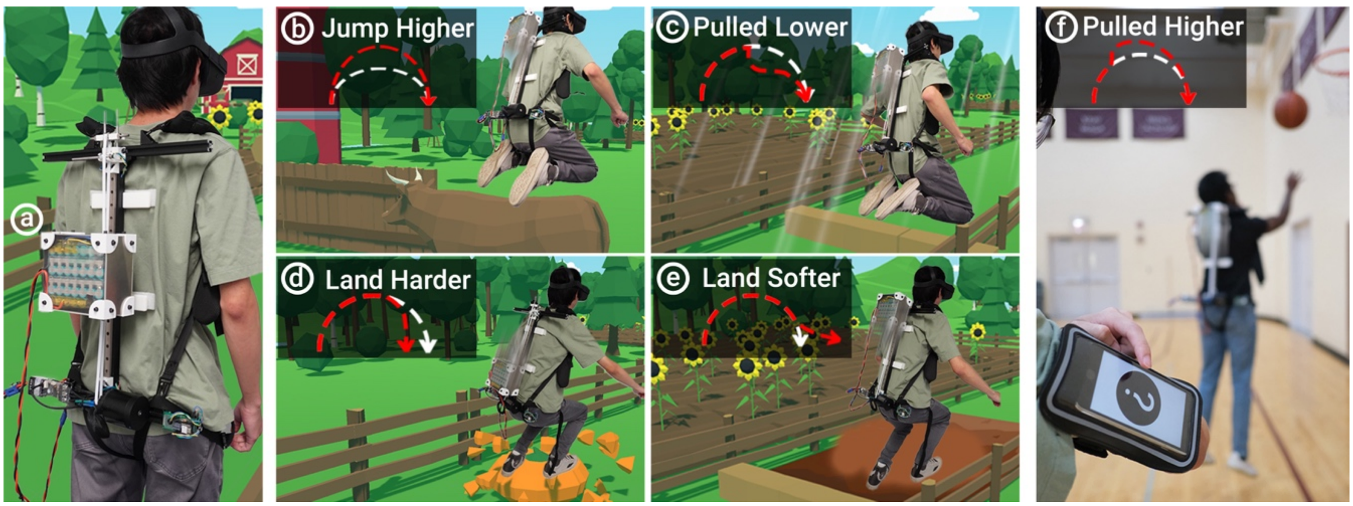

Figure 1: (a) We propose JumpMod, a wearable device that modifies the user's *perceived jump*. It achieves this by moving a weight along the user's back, causing an inertial force that modifies one's sense of vertical momentum. We found, in our second user study, that JumpMod creates five distinct sensations: a sense of (b) jumping higher; (c) being pulled lower; (d) landing harder or (e) softer; and (f) being pulled higher—demonstrated in our applications, including VR and interactive sports such as (f) interactive basketball.

ABSTRACT

Vertical force-feedback is extremely rare in mainstream interactive experiences. This happens because existing haptic devices capable of sufficiently strong forces that would modify a user's jump require grounding (e.g., motion platforms or pulleys) or cumbersome actuators (e.g., large propellers attached or held by the user). To enable interactive experiences to feature jump-based haptics without sacrificing wearability, we propose JumpMod, an untethered backpack that modifies one's sense of jumping. JumpMod achieves this by moving a weight up/down along the user's back, which modifies perceived jump momentum—creating accelerated & decelerated jump sensations. In our second study, we empirically found that our device can render five effects: jump higher, land harder/softer, pulled higher/lower. Based on these, we designed four jumping experiences for VR & sports. Finally, in our third study, we found that participants preferred wearing our device in an interactive context, such as one of our jump-based VR applications.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3580764

KEYWORDS

jumping, full-body, haptics, virtual reality, wearable, backpack

ACM Reference Format:
Romain Nith, Jacob Serfaty, Samuel G Shatzkin, Alan Shen, and Pedro Lopes. 2023. JumpMod: Haptic Backpack that Modifies Users' Perceived Jump. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3544548.3580764

1 INTRODUCTION

A decade ago, users saw a fundamental change in the way they interacted with digital experiences in their homes. With the introduction of motion sensing (e.g., *Kinect*), the user's full body could now serve as an input, enabling a more bodily way to interface with computers. Today, we witness many interactive experiences taking advantage of this, from interactive sports (e.g., laser tag or interactive bouldering) to physical VR experiences (e.g., *BeatSaber* or VR fitness).

However, enabling interactive devices to move the user's *entire body* (i.e., body-scale force-feedback) has proven challenging. This is primarily due to the large forces required to move the user's body mass. Moreover, it is arguably easier to achieve force-feedback in the horizontal plane since a haptic device needs only to overcome the inertia and friction, which might explain the wider proliferation of interactive treadmills [9, 19, 42, 50] or walkable floors [20, 48] when compared to vertical force-feedback devices. A vertical force-feedback device needs to, additionally, overcome gravity (i.e., push against the user's whole-body weight).

When researchers and practitioners (e.g., theme-parks or VR arcades designers) need to vertically actuate the user's *entire* body, they usually resort to installing large actuators in the environment, such as motion platforms [3, 4, 11, 45], mechanically-actuated pulleys [10, 12, 23, 27, 57, 58], wind tunnels [6, 28, 35] or attaching powerful propellers to the user's body [47, 52]. While these are effective since their output force is large, they *completely limit users' mobility*—these devices are at odds with users' home setups (e.g., free-walking VR experiences [17, 33, 44]) and are at odds with experiences where users are highly mobile (e.g., interactive sports [1, 5, 29, 36, 40]). Instead, to enable more mobile contexts, a vertical body-scale actuator that is *untethered and wearable* would prove ideal.

To tackle this and expand the portability of vertical force-feedback, we propose *JumpMod*, an untethered backpack that modifies the user's perceived jump. Our device, depicted in Figure 1, achieves vertical effects by moving a weight along the user's back, even while in mid-air. The key behind *JumpMod* is that its actuator does not physically lift the user—as this would require motors and infrastructure 10x larger/heavier, preventing our small form-factor—instead, *JumpMod* creates only a *sufficiently* strong inertial force that users can feel, which *modifies* their *perceived* jump—this force generated by our device is above Weber's fraction for the noticeable perception of a force [13] (refer to *Study#2* and *Technical Evaluation* for details).

As such, interactive experiences can make use of JumpMod at *different phases* of a user's jump (i.e., launch, ascension, descension, landing) to create *five distinct effects*. We confirmed these findings via three user studies: (1) We found an accuracy of 94% in our jump detection, while participants were not instructed to jump in any controlled manner (i.e., they jumped as they saw fit, including jumps designed to trigger false positives). (2) We empirically found that JumpMod evoked five unique jump effects: jump higher, land harder, land softer, pulled higher, and pulled lower—to robustly measure that these effects were created by our device's moving weight, we conducted this study in the absence of any visual or sound stimuli and included a spoofing condition, with additional vibrations and weak motor movements. (3) We found that JumpMod added realism to VR jump sensations, even compared to existing VR techniques that visually exaggerate the avatar's height [17, 56]. Finally, to demonstrate the applicability of our device, we designed four experiences that utilize these jump effects—two interactive sports and two VR-based applications.

2 WALKTHROUGH

To illustrate the applicability of our device, we demonstrate it with an example of a jumping-based experience in VR. The haptic effects for this experience were designed based on empirical results from our perceptual experiment (see *User Study#2)*. In this walkthrough, our VR user is wearing our untethered backpack and a wireless VR headset. This VR experience is set in a farm scene shown in Figure 2 (b) and the game is played by running and jumping in place to avoid obstacles and collect points by smashing pumpkins.

Jumping higher. Our user is running in place and attempting to jump over obstacles, however, when they attempt to jump over a cow, they fail—they lose speed and must try again. On the second attempt, depicted in Figure 2 (a), they collect a jump boost power-up. As the user jumps again, our device actuates its weight ***up*** in sync with the user's ***launch*** from the ground (refer to *Technical Evaluation* for details; in short, our device takes only 20ms to detect any jump phase with high accuracy). This extra inertial force from the moving weight creates a sense of ***jumping higher*** (see *User Study#2* for the validation of all perceived effects). As a result, not

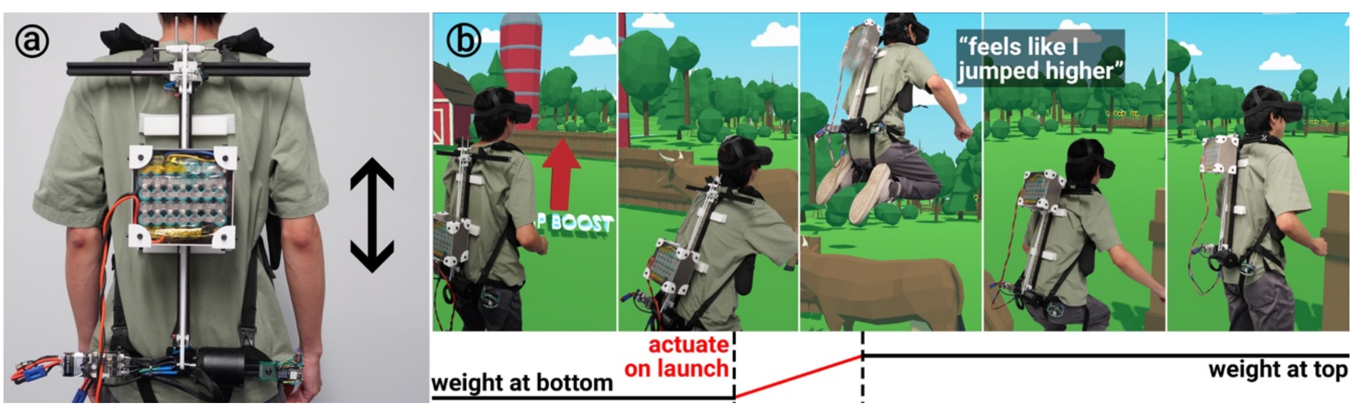

Figure 2: (a) *JumpMod* creates vertical haptic feedback by physically moving a weight up/down. It can create (b) the experience of jumping higher (during a power-up) by moving the weight up in sync with the *launch* phase of a jump as depicted by the plot.

only do they *see* their avatar jumping over the cow (as typical in jumping-based VR experiences, we also scale up the avatar's jump), they *feel* this matches the *physical sensation* of jumping higher—in fact, in our third study, we found that the combination of our device with the visual exaggerated jump was more realistic than only the traditional visual jump effects.

Pulling down. Figure 3 (a) illustrates our user jumping over the next obstacle, a haystack. This time when the user is still mid-jump and about to land, the wind blows harder and pulls the user down (the user sees and hears the wind effects too). To generate this "pulling" sensation, our device moves its weight ***down*** as the user is ***landing***.

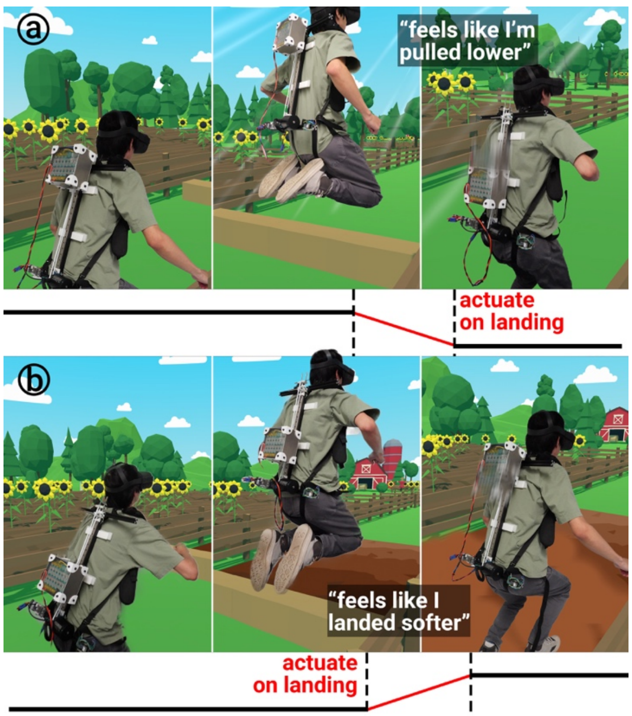

Figure 3: (a) *Pulled down* is rendered when the wind is blowing the user. (b). *Land softer* is rendered when the user lands in the mud.

Landing softer. Figure 3 (b) also illustrates our user landing on mud rather than on solid ground as before. To render this "softer" landing from the mud, our device moves its weight ***up*** when the user is ***landing***.

Landing harder. Next, our user spots a pumpkin in Figure 4 (a), which they need to smash for points. They collect a force power-up that enables smashing pumpkins. To render this sensation of "landing harder" on the pumpkin, our device moves the weight ***down*** when the user is ***descending*** (past the apex, the highest point of their jump).

Resetting the weight's position. So far, our VR experience was designed with all effects in a sequence that left the device's weight in the place where the next effect was required. But this is not necessary; there are several ways that a designer can use to reset the weight's position, ranging from options that conserve realism to those that provide speed: (1) planning the order of the effects so no reset is needed, (2) slowly driving the weight so that user cannot perceive it, (3) associating the reset with other in-game events (e.g., an earthquake that triggers back-and-forth vibrations, but also move the weight to the intended position), and (4) moving it rapidly before the next event. In this particular experience, the designer chose to look ahead to the next obstacle and, if needed, always slowly reset the weight to the required start position (inspired by *HapticTurk*'s lookahead for the next haptic events [8]). As the next event requires the weight to start at the top but is currently at the bottom, the VR application sends a "5s reset up" instruction to our device, which brings the weight up in 5s. To make sure the weight is at the correct position, our device implements a positional PID controller (see *Implementation* details).

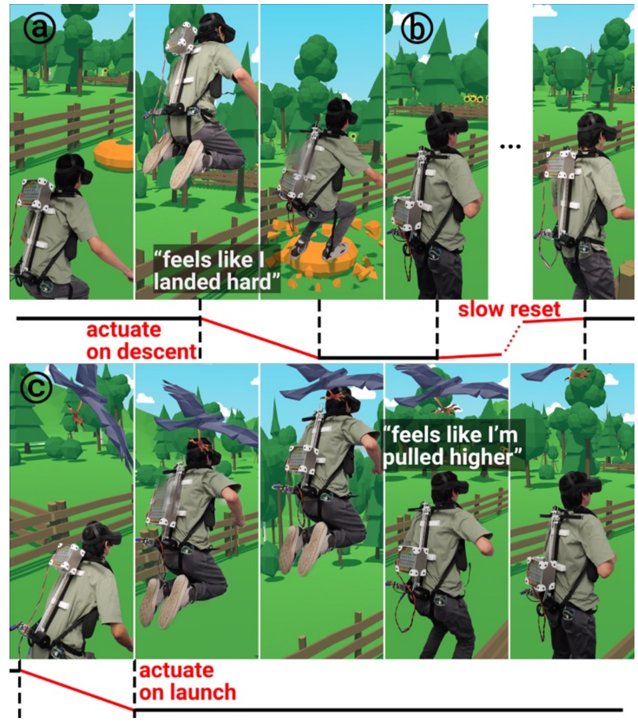

Figure 4: (a) *Landing harder* is rendered when the user lands on a pumpkin to smash it. (b) The weight slowly moves up to reset its position in preparation for the next effect. (c). *Pulled higher* is rendered when the user is assisted by a bird during a jump.

Pulled higher: Finally, Figure 4 (c) depicts the user having collected all the pumpkins and is about to jump over a final obstacle—they can already see the finish line. As they jump, a bird flies next to the user and pulls the user up to help them to the finish line. Our device renders this "pulled higher" by moving the weight ***down*** at the user's ***launch***.

3 RELATED WORK

Our work builds on body-scale haptics devices that render vertical force-feedback, i.e., devices capable of actuating the user's body upwards or downwards. As such, we take inspiration from grounded haptic devices that explored these effects, but also, from novel approaches that modulate the user's perception to create haptic effects at portable form factors.

3.1 Body-scale *grounded* haptics devices that render vertical force-feedback

The canonical way to realize very large forces, such as those capable of actuating the user's body, is to push from the ground against the user's body—as such, these types of haptic devices (which typically employ motorized pulleys or arms) are denoted as *grounded* devices. These devices excel in the amount of output force and are arguably some of the most mainstream haptic devices as they are featured prominently in theme parks, amusement rides, and selected theaters equipped with motion seats.

Vertical motion platforms. The most popular way to realize vertical haptic effects is to physically actuate a platform in which the user is *standing or sitting* in (typically with strong electrical, hydraulic, or pneumatic actuators). For instance, the *Stewart* motion platform generates accelerations (typically in all directions, including vertically), while the user is sitting [3, 4, 11, 34] or laying down [45]. Other variations of this actuation mechanism include stepping on individual platforms (one for each foot, to simulate walking on various terrain heights) [21, 49], walking on platforms that change the user's height [20, 48], or even combining a staircase and motorized lift to create infinite stairs in VR [7].

Pulley-based vertical actuation. Another method to render vertical haptic effects is to pull the user's body weight via pulleys/cables. For instance, *Pull-Ups* [58] can pull/push the user's body to the ceiling by altering the length of a pulley that the user is grabbing onto. Instead of requiring users to grasp, other devices suspend the user's body in midair with pulleys to simulate the skydiving [12] by catching the user with cables and counterweights after jumping off a platform. By motorizing the pulleys, the sense of gravity can also be manipulated with the help of closed-loop systems where jumping physics can be altered such as jumping on the moon [23, 27], or zero-gravity to train astronauts [10].

Replacing actuators with passive props or human-actuators. Two more recent alternative approaches to realizing vertical haptic effects include: (1) replacing mechanical actuators with "human actuators" [8], and (2) replacing actuators with passive props, such as trampolines [54] or physical stair props [38, 39]. While these are promising ways to circumvent some of the limitations of large grounded vertical actuators, these do not inherit the same interactive properties as the latter, i.e., fast response speed, force, and, most importantly, most of these are strictly bounded to the use of virtual reality to not break the immersion.

Limitations. Thus, typical vertical actuation is still based on motion platforms and pulleys. However, these devices are comprised of large actuators, which consume a lot of energy and are grounded to the infrastructure. As a result, existing devices capable of vertical effects are not compatible with mobile experiences (e.g., sports or free-walking VR).

3.2 Body-scale *Ungrounded* haptics devices that render vertical force-feedback

To tackle the limitations of grounded vertical force-feedback, researchers explored alternative wearable actuators.

Changing center of gravity. One way to approximate an ungrounded rendition of the steward platform is to move weights that shift the user's center of gravity. Maekawa et al. proposed a wearable robotic tail that swings as a counterweight, preventing the user from falling [32]. Similarly, *Arque* [37] uses a robotic tail mechanism to also provide haptic feedback for forces in VR. *Inclination Manipulator* [51] creates the sensation of inclination in VR by moving a weight horizontally away from the body. These devices mostly render inclination, not vertical momentum. On the other hand, ergonomic backpacks [18, 46] help reduce strain from heavy loads by reducing vertical momentum but do not provide force-feedback.

Actuating with Air. One way to actuate the user's body vertically without the need for grounding the user to an actuator is by ejecting air from a device worn by the user [25, 52] or from a device that the user is grasping [47]. For instance, *Augmented jump* [52] attached sixteen ducted fans to the user via a frame. By powering all fans, the device is able to lift 50kg of weight off the ground, simulating a 75% reduced gravity when the user jumps. However, because the device requires a lot of power, users need to *carry 24kg of batteries* in addition to the user and the hardware's weight and its fans produce a lot of noise. Similarly, *Weighted Walking* [25] attaches two ducted fans per leg to emulate walking on different viscosity or simulate different gravity (each fan produces 22.4N, but also a lot of noise). Another approach, *Virtual Super-Leaping* [47], is a handheld device with eight mounted propellers. By jumping with the device *held in their hands*, users experience force applied to their hands, while in mid-air. While the force is not applied directly to the torso but to the hands, by coupling this effect with VR visuals, users can perceive bigger jumps when the propellers are thrusting up (27.8N, not enough to lift the user as opposed to *Augmented jump*). This device only pushes air in one direction and, as such, it is limited in sensations (e.g., jumping higher, gliding, or landing harder by turning off the propellers). Being a propeller-based actuator, the device inevitably produces a lot of noise—measured at 95.3dB.

Limitations. To sum up, while these techniques are promising, they have four key limitations. First, these devices demonstrate only how to increase the user's jump, i.e., jumping higher, but miss adding more expressivity in vertical effects, such as modifying one's landing to feel both harder or softer, or even rendering a sense of being pulled up/down in the middle of a jump. Second, propeller-based actuators require tremendous amounts of power, making these actuators only available for short experiences. Third, some of these propeller approaches (e.g., *Virtual Super-Leaping* [47]) require users to hold onto the propellers, which prevents their hands from being used in the interactive experience. Finally, propellers come with additional limitations, such as loudness and ejecting air to the body/face, which creates unwanted haptics. Instead, we take inspiration from these approaches but produce jump sensations using a new type of actuator: our backpack moves a weight along

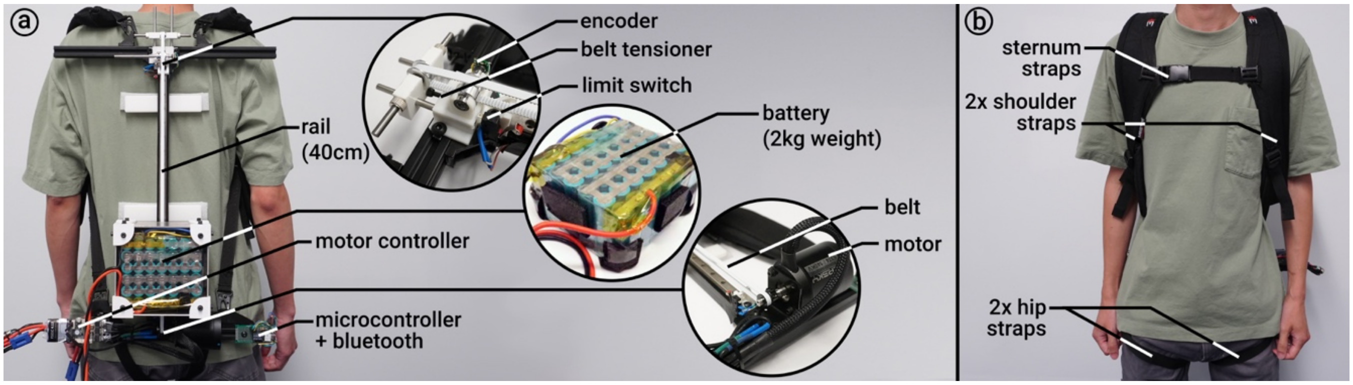

Figure 5: (a) JumpMod is untethered and wearable; it features: a custom battery that doubles as a 2kg weight, a driving belt system with an encoder, and a microcontroller with Bluetooth. (b) Four anchor points and a sternum strap secure the backpack to the user.

the user's back, which we found to produce new jump sensations at a wearable form factor.

3.3 Modifying *virtual* jumps using VR illusions

Finally, one alternative approach that is only applicable to VR experiences is to manipulate the virtual representation of the user's jump to modify their sense of jump – this is often denoted as *Redirected Jumping*. In these VR techniques, the avatar's jump trajectory is changed from the user's actual jump. For example, *PseudoJumpOn* [43] exaggerates the virtual jumping trajectory and allows users to feel as if they jumped higher. To make this illusion believable, many researchers have conducted extensive studies to evaluate the virtual jump trajectories [16, 17, 22, 30, 31] to optimize the physical space for jumping in VR [16]. *Malleable embodiment* [24] uses another approach; it visually depicts the user's past or (predicted) future movements to induce a sensation of jumping heavier or lighter. However, this method not only works solely for VR; it also demands that user always sees a reflected view of their avatar (using VR mirrors, and so forth).

While these techniques are useful in VR, they are not applicable beyond virtual interactions. Moreover, as we will demonstrate in our third user study, even in the case of VR, our backpack produced a more realistic jump sensation than the existing approach of only-visually modifying the avatar's jump.

4 IMPLEMENTATION

JumpMod is implemented as a self-contained haptic backpack (i.e., works on battery and is wireless)—to accelerate researchers interested in reproducing this device, all designs and code will be available and open-source[1].

For sensing jumps, our device couples with existing tracking systems. In all our examples we used HTC VIVE's positional tracking for its accuracy and low latency; however, other tracking methods are possible. For instance, in earlier phases of our research, we explored using a 6-DOF inertial measuring unit (IMU) in the backpack—we found that the drift of the IMU was problematic when coupled with our simple jump detection algorithm and would not suffice for applications requiring high-accuracy in jumping detection (>90%), yet might suffice for applications that do not require this high-bar of accuracy. Future researchers building on our work might indeed explore an IMU-only detection but will need to depart from our simple detection approach towards a more sophisticated algorithm that handles IMU drift.

4.1 Hardware

Our device's hardware can be broken down into four main categories: (1) structure, (2) battery as moving weight and source of power, (3) drive system, and (4) control.

Structure. The structure of the device is built around three 2020 aluminum extruded profiles, shaped to form the letter "I" as depicted in Figure 5 (a). We made this structure into a backpack and improved its ergonomics by adding padded straps (Meister Backpack Straps) on the four corners as well as hard-foam padded supports along the aluminum profiles to spread the contact surface evenly on the user's back.

Battery as moving weight *and* as power. Typical weight displacement devices opt for weights made from dense materials (e.g., lead), which allows the weight to be as small as possible, yet heavy. However, these devices still need to attach a power source—the larger the weight the heavier batteries become. While this is not an issue for grounded devices, this is critical for comfort in wearables. Instead, in JumpMod, our battery has a double functionality: as weight *and* as a power source. To create a battery that satisfied both requirements perfectly, i.e., one that weighed approximately ~2kg and provided fast current discharge for our motor's operation, we created a custom battery by manually packing cylindrical Li-ion cells. Our resulting custom 24V 7Ah battery is made from 42 cylindrical Li-ion cells (Anhui Ruikema Li-ion ICR18650-20P5C, 3.7V, and 2000mAh) in a 6S7P configuration, weighing 2kg. It is enclosed in a box we engineered from five 2mm waterjet-cut aluminum plates and a 3 mm acrylic plate, connected via 3D printed PLA corner brackets.

Battery life. We measured an average power consumption of ~3.8A in a 707ms motor run. With a 7Ah battery capacity, our

[1] http://lab.plopes.org/#JumpMod

device provides about two hours of non-stop operational time—*this is without ever stopping the motor* (i.e., "constantly jumping"). In a more applicable scenario, our backpack would last, for instance, ~**12 hours**, in a high-paced experience where participants would jump every 5s.

Drive system. JumpMod is actuated by a brushless DC motor (Flipsky Outrunner 6374, 140kv, 980g, and 3.5kW max power output) controlled by an electronic speed controller (Flipsky FSESC V4.12 50A). This motor was chosen for its torque output (max 8Nm) in a small and lightweight form factor (980g). Our motor drives the weight using a timing belt pulley (6mm steel core reinforced 2GT) along a linear rail guide (MGN12H 400mm). Since this weight is moved relatively fast (707ms to actuate the weight from end-to-end), the steel core belt is essential to maintain the tension as well as avoid degradation over time (especially compared to the more commonly found nylon core belts). A custom 3D-printed PLA tensioner resides at the other end of the belt allowing us to ensure high tension to maximize the power transfer from the motor to moving the weight while avoiding slipping.

Encoder. Our haptic effects are based on the inertia generated by moving the weight and not by any vibrations generated by the weight hitting the ends of the rail. As such, to move and stop the weight accurately, we implemented an encoder that determines the belt's movement. We employ a rotary magnetic encoder (Pololu Magnetic Encoder Pair Kit for Micro Metal Gearmotors) attached to an idle pulley, which keeps track of the position of the weight.

Controlling the weight's position. This encoder allows closing the loop and controlling the weight position precisely by ensuring the weight reaches target positions (rather than just "actuate for 100ms"). To control this accurately, we implemented a PID controller that maximizes the speed and prevents parasitic movements (overshoots or oscillations).

Microcontroller. The logic in our device is implemented by a Seeeduino XIAO nRF52840 BLE Sense. This was chosen because of its embedded sensors and, more importantly, its built-in Bluetooth. The microcontroller is powered by the electronic speed controller's 5V line and communicates to it via UART, using the VESC library [55]. While we used external trackers from a VR system (HTC VIVE) in our studies for more accurate jump detections, users have the possibility to use its embedded IMU for applications where external trackers are not possible at the cost of lowered detection accuracy.

Safety. Finally, two limit switches are located at the ends of the rail. If these are triggered, the microcontroller stops the motor. Moreover, since microcontrollers can also be affected by faults, we added metallic physical stoppers at the ends of the rail, which ensure that even if the limit switches did not trigger, the weight stays inside the rail.

4.2 Jumping detection algorithm

We detect when users jump using a simple algorithm, based on absolute head position via an external tracking system. In our experiments, we utilized the HTC VIVE Lighthouse V2 tracking system and its headset; yet, as mentioned, others expanding on JumpMod can explore different external tracking systems or even its built-in IMU for jump detection.

Since our tracking is based on HTC VIVE, we implemented it in Unity3D and it communicates wirelessly to our backpack via Bluetooth (communication latency 68.5ms, see *Technical Evaluation* for details). At the start of tracking, our jump detection algorithm records the user's height as a reference. In runtime, whenever the user's position is below their height by 7.5cm and there is no jump in progress, the algorithm determines that the user started crouching in preparation for the jump and their position is currently decreasing. Now, our algorithm is ready to track four jump phases: *launch*, *ascent*, *descent,* and *landing*. First, our algorithm triggers the *launch* phase when the user's head position starts increasing again (extending after crouching). Second, our algorithm triggers the *ascent* phase when the user's height surpasses their reference height (their feet start to lift from the floor). Third, our algorithm triggers the *descent* phase when the user's height stops increasing and starts decreasing (they pass the apex of the jump). Finally, our algorithm triggers the *landing* phase when the user's height crosses their reference height (the user's feet are about to touch the floor). Our algorithm is able to differentiate a real jump (user's feet left the ground) from a fake jump (user intended to jump but decided to abort keeping their feet on the ground) with 94% accuracy (see *User Study#1* for details).

5 TECHNICAL EVALUATIONS: DEVICE AND ACTUATION CHARACTERIZATIONS

We conducted two technical characterizations: **(1) device characterization** (i.e., end-to-end latency, noise, belt slip, and weight), and **(2) actuation characterization** (i.e., load weight consideration, kinematics, force, and backlash).

5.1 Device characterization

End-to-end latency. We measured an end-to-end latency of 105.2ms using a 240fps camera. This latency can be further broken into: (1) 20.0ms for our detection algorithm, (2) 68.5ms of Bluetooth communication from an application to our backpack, and, finally, (3) 16.7ms for the microcontroller to start moving the weight. We minimized our Bluetooth latency by preemptively queuing the next effect and triggering it when needed with a single-byte BLE characteristic.

Operating noise. We measured the sound generated by our backpack as it moves the weight along the rail using a decibel meter (Smart Sensor AS824) placed at the height of a user's ear (quiet environment, three repetitions per driving direction). We measured an average peak intensity of 78.1dB. In comparison, propeller-based vertical force feedback devices tend to be louder. For instance, *Virtual Super-Leaping* [47] produces 95.3dB at full power. Moreover, it is worth noting that propeller-based systems actuate for several seconds (3-10s or more), while our system drives the motor from end-to-end in ~707ms. Not only is our device quieter, but it also produces shorter sounds. One option to reduce the sound of our device even further is to enclose it in a box. Note that this is only possible in our device, but not in propeller-based devices (need to be exposed to the air around the user, to push against the air). By enclosing our device in a generic plastic storage bin (82.5 x 45 x 16.5cm), we measured an average peak sound intensity of 70.3dB

—an 8.8dB reduction from a simple plastic storage, we are confident this can be further reduced with a well-engineered acoustic enclosure.

Force of the belt before slippage (at stall). We found a maximum output force of 45N before our belt system started slipping during stall. This was measured by pulling the motor against a solid nylon string connected to a load cell (DYMH-103 10kg amplified by an HX711). This setup measured the maximum force before one of the components failed. In our case, the maximum was reached when the belt started slipping on the motor shaft. During the measurements, the backpack was laid flat so that no gravitational force was acting on the moving weight. Finally, this depicts an upper limit as it was measured horizontally and with the motor outputting instant torque, i.e., when held upright against gravity, the belt is likely to slip if the stall force exceeds 35N.

Device weight. The total weight of the device is 5.4kg (measured with a Camry Digital Scale, 50g resolution), including its 2kg load, which also doubled as our battery to minimize the total weight of the device (see details in *Load consideration)*. For comparison, *Augmented jump* weights a total of 60kg, including 24kg of batteries [49]. *Virtual Super-Leaping* [47] weighs 1.4kg including the batteries and actuator; while this device is lighter than ours, it requires the user to hold onto it with their hands, likely to cause muscle fatigue faster, compared to JumpMod which is worn on the back.

5.2 Actuation characterization

Load considerations. We considered two key factors when choosing the weight of our load: (1) *force* created by the load as it accelerates—this creates more momentum (i.e., desired haptic sensations); and (2) *backlash-on-landing* created by the user's impact on the ground as they land, even when the device is not being actuated—this force is then transferred to the weight and displaces it, creating an unwanted recoil-like haptic sensation with vibrations and noise. Our device mitigates this by using the motor's brake-mechanism when the weight is not being actuated on the rail—this helps minimize slipping on impact. The challenge is that these two factors are *at odds* with each other—we strived to maximize *force* (implies using a heavier weight) but, simultaneously, strived to *minimize backlash-on-landing* (implies using a lighter weight). Figure 6 depicts averages over three repeated measurements: (a) *backlash-on-landing*—measured as the weight's slip distance relative to the total rail length (in %), after a jump while the motor was set to *brake mode*; and (b) *force*—measured in Newtons (via F=m·a), using the acceleration data from a 6-DOF IMU (MPU6050) attached directly to the moving weight as the backpack was affixed to a stand (e.g., no jumping and gravity force not subtracted from readings). Both measurements were taken across three different weights: 1kg, 2kg, and 3kg—weights larger than 4kg are not possible since, as previously mentioned, the belt will slip if the motor is stalled around 35N of force (or ~3.5kg weight). We found, as depicted in Figure 6 (a, b), that the 2kg weight struck a better balance between peak force (56.9N) with only 2% of backlash-on-landing, compared to a ~3x increase in backlash-on-landing with the 3kg weight. Finally, since our battery doubled as our weight, our final load was constructed by incrementally adding 3.7V LiPo cells until they reached a total weight of 2kg.

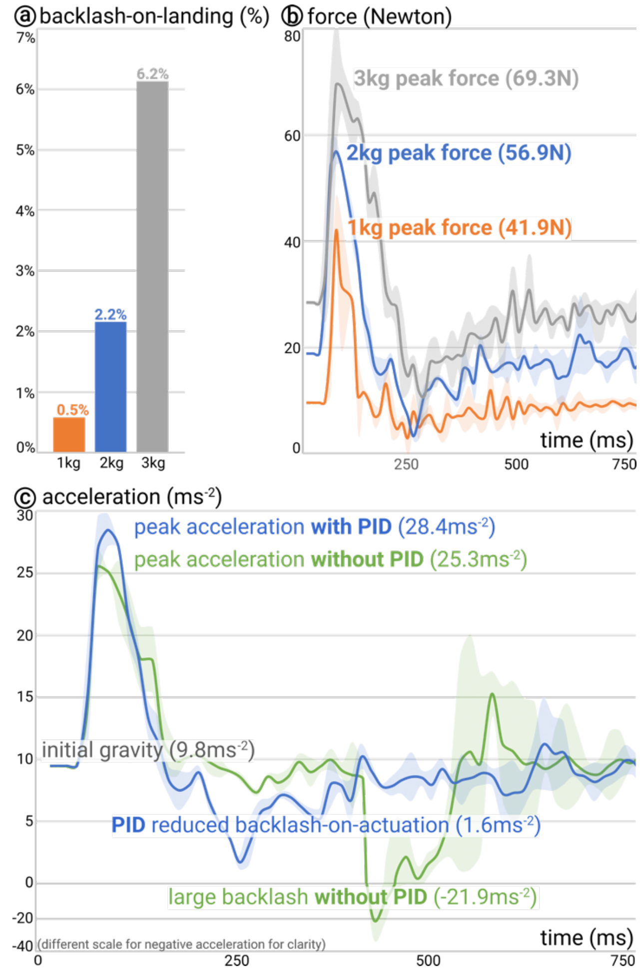

Figure 6: (a, b) Comparison of trade-offs between different loads (1kg, 2kg, and 3kg) with regards to (a) *backlash-on-impact* and (b) *peak force.* (c) Kinematics graph for our device moving its 2kg weight from bottom to top, *with* and *without PID.*

Kinematics. Now, having chosen the 2kg weight, Figure 6 (c) depicts the kinematics of its acceleration from bottom to top, *with* and *without our PID controller*—measured at peaks of 28.4ms^{-2} and 25.3ms^{-2}, respectively. Thus, we found a relatively similar peak acceleration across conditions, yet, as we will analyze later, the sudden deceleration is substantially larger without PID. For each condition, we measured the kinematics using the same procedure as in our aforementioned force measurements in *Load considerations* with three repetitions per condition (e.g., device fixed to a stand and gravity force not subtracted from readings). From this peak acceleration with PID, we determined a 56.9N force generated by our weight (via $F=m\cdot a$). This force was perceived by participants even in the extreme conditions of our *Study#2* (i.e., no visuals, no sounds, and spoofed motor action).

Minimizing backlash-on-actuation. When our device actuates, it rapidly accelerates its weight. This driving mechanism left alone would result in the weight hitting the ends of the rail and generating *backlash-on-actuation*—an unwanted haptic effect, including an additional force in the opposite direction, noise, and vibrations. To minimize this, our device employs a PID controller. As depicted in Figure 6 (c), we found that the addition of our PID controller significantly reduces the backlash-on-actuation by ~4x: with the PID we found an average peak deceleration of $1.6ms^{-2}$; conversely, without PID, we found an average peak deceleration of $-21.9ms^{-2}$.

Limitations. The characterization presented in this section was conducted using our hardware. As such, we urge the readers to not generalize these findings beyond our specific implementation, but instead, to use these as a reference to compare future variations (e.g., with different motors, rail systems, and so forth).

6 STUDY OVERVIEW

We conducted three studies to characterize and understand the perceived effects of our backpack on participants.

In our first study, we characterized **our jump detection accuracy** with participants that were not instructed to jump in any controlled manner (they jumped as they saw fit). We found our system detected all four phases of participants' jumps with 94% accuracy, including jumps designed to induce false positives (quick crouches, no jump).

In our second study, we studied what **types of sensations JumpMod evoked**. We were especially interested in the sensations created by our device alone, in the absence of any visual or sound stimuli. As such, we actuated our device in all possible configurations (up or down, for all four jump phases, which we call an *actuation mode*) and measured participants' feedback as they described their jump sensations. We found that our device, even in the absence of any visual/audio effects, created five unique jumping sensations with strong consensus among participants: jump higher, land harder, land softer, pulled higher, and pulled lower.

Finally, in our third study, we evaluated whether **JumpMod added realism to jump sensations**, even compared to existing VR techniques (e.g., visually exaggerating the avatar's height). We found that participants rated the experience and the jumping realism higher while wearing our device, compared to without (baseline).

7 USER STUDY#1: JUMP DETECTION ACCURACY

We conducted a technical evaluation with 12 participants to assess the accuracy of our simple jump detection algorithm (based on tracking head position). This study was approved by our ethics committee (IRB22-0064).

7.1 Study design

Apparatus. Participants wore an HMD to track their head position (HTC VIVE Pro Eye and Base Station 2.0).

Participants. We recruited 12 participants (five identified as female and seven as male; average age=23.8 years old; SD=3.8). Furthermore, participants' body mass index (BMI) averaged 22.15 (SD=4.11). All participants were healthy with no motor impairment. Participants were compensated with 10 USD for their time.

Tasks. We asked participants to perform two tasks (each 20 jumps): **(1) jump-as-you-like-when-you-like** and **(2) jump-as-fast-as-possible-but-only-on-green**—this latter task was designed to see if participants crouching but then deciding not to jump could trigger false positives. In this task (inspired by the standard "go/no-go" psychophysics design [15]) participants were presented with a VR traffic light (red, yellow, green). First, the yellow light would show up, indicating "get ready and crouch". After a random 1-2s wait, participants monitored the next light and jumped as fast as they could in response to seeing it. They were instructed with the following: if the next light is green, jump; if it is red, rise to stand as fast as possible but without jumping. The "go" vs. "no-go" order was randomized (10 of each).

7.2 Results

Jump detection. For a total of 232 jumps (discarding jumps with corrupted data from the tracking system, e.g., 4-meter-high jumps, incomplete jump data points, etc.), in which participants jumped as they saw fit, we found that our detector was able to recognize 100% of these, including all four phases for each.

Go-no-go detection. For a total of 228 jumps (discarding erroneous data from the tracking system, e.g., 4-meter-high jumps, incomplete jump data points, etc.), we found 100% detection on the "go" jumps (all four phases) and 94.0% detection on the "no-go" jumps (100% *launch*, 86.2% *ascent*, 91.4% *descent*, and 98.3% *landing*).

Study limitations. We acknowledge that our high recognition rate is likely due to the straightforward nature of jump detection using an absolute tracking system—i.e., jumps obey the laws of physics, they all require a crouching phase, and they all require participants to push upwards and to overcome their original height soon after they leave the ground. Thus, it is straightforward to use this physics-knowledge and the participant's height prior to detecting jumps very accurately. Finally, while we asked participants in the first task to jump-as-they-like, we acknowledge other users (e.g., different body abilities) can exhibit a jump style not detectable with this approach.

8 USER STUDY#2: MEASURING THE PERCEIVED VERTICAL EFFECTS

The objective of our second study was two-fold: **(1) validate that it is the movement of the weight that modify a participant's jump experience** (i.e., does the effect originate from movement of the weight or parasitical weak vibrations or placebo effects?); and, **(2) measure the perceived quality of the induced haptic effects** (i.e., does driving the weight up/down at different phases of a jump modify the jumping experience?). This study was approved by our ethics committee (IRB22-0064).

8.1 Study procedure.

Study design. To best validate that it was the driving of the weight that modified the user's perception of their jump, we utilized a standard "oddball" study design [14]. In our oddball study, participants jumped in a series of four jumps. However, at random, in only *one* out of these jumps (at random) our device's weight was actuated significantly (the oddball). In all the other (non-oddball) jumps, we spoofed the activity of our device to the best of our capabilities so that participants could not rely on any other physical cues. We added vibrations, sounds, and even weak motor movements to create a placebo condition—all of this makes it very hard to identify which jump was the oddball, and certainly very hard to identify what was happening in the device. As a result, this study design controlled all the key variables (e.g., extra physical cues like vibration), except for the oddball jump in which our weight is actuated.

Placebo trials. We spoofed the activity of our device, except its inertia, using five techniques: **(1) Sound spoofing**: participants wore earplugs *and* noise-canceling headphones that also emitted white noise. **(2) Vibration spoofing**: we attached a 600W subwoofer (Kuerl 10 Inch 600W Sub-Woofer) that played sinewaves at 30Hz, 280Hz, and 980Hz (the main frequencies that our motor vibrates at, obtained by analyzing the vibrations of our motor using both a Cold Gold Audio piezo microphone and an MPU6050 accelerometer); as such, in all jumps, participants feel large vibrations from the subwoofer that masked the (weaker) vibrations from our motor. **(3) Constant weight/pressure**: in all trials, participants wore the backpack, and our jump sequence was short enough so that its weight/pressure would be equalized fairly. **(4) No distinct visual feedback**: participants faced a tall white wall for the whole study, which did not provide any distinct visual features to track the height of their jumps. **(5) Motor always moved**: in every jump, we always actuated the motor at 20% of its power (weak, acted as placebo) and, only at the oddball trials run at full power—this 20% actuation, which ran in the same direction and for the same duration as the oddball, created additional vibrations and sounds that further improved our previous spoofing methods. Putting these five methods together provided a robust study design, in which participants could not rely on any additional cues and were left *only* with the effect of the *large acceleration* of the weight across their back.

Confirming the spoofing. To gather confidence on our spoofing method, we conducted a pilot test with four pilot participants (average age of 27.5 years old, SD=5.92; average BMI of 22.3; SD=1.60), which were not recruited for the actual study. In these pilots, we confirmed this method for spoofing was robust as pilot participants could not accurately identify if the motor was moving (i.e., they guessed at around 50% accuracy, i.e., chance level).

Apparatus. We utilized our backpack (shown in *Implementation*) but with a medium-density fiberboard (MDF) backplate rather than our hollow frame—this helped secure the subwoofer for adequate spoofing.

Participants. We recruited eight participants (three identified as female and five as male; average age was 23.5 years old; SD=2.1). Participants' body mass index (BMI) averaged at 22.68 (SD=2.49). All participants were healthy with no motor impairments. Participants were compensated with 10 USD for their time.

8.2 Actuation modes

We actuated our haptic device in eight actuation modes: two movement directions (up or down the user's back) × four jump phases (*launch*, *ascent*, *descent*, *landing*). During the study, the presentation order of the modes was randomized.

Table 1: Accuracy detection results per actuation mode.

Accuracy	Moving weight up	Moving weight down
Launch	88%	91%
Ascent	75%	66%
Descent	81%	94%
Landing	79%	86%

8.3 Trial design & metrics

Warmup. Prior to the start of the trials, we gave participants a chance to wear our backpack and jump with it until they felt comfortable; this further allowed participants to get accustomed to any variation in their own jumps.

Trial. A trial was composed of a sequence of four jumps, with one being an oddball, chosen at random. Per actuation mode, participants performed four trials—a total of 128 jumps per participant (2 directions × 4 phases × 4 jumps).

Metrics. After a single trial (four consecutive jumps with one of these being an oddball), the participants were asked to verbally answer two questions: (1) identify the oddball (i.e., which jump felt distinct from all others) and (2) explain the quality of that jump using any descriptors associated to jump sensations (i.e., "jumped higher," "jumped lower," "landed harder," "landed softer," "longer airtime," "shorter airtime," "pulled higher," and "pulled lower"). Participants were asked to select which descriptor(s) best represented the sensation they felt in this trial. If participants selected more than one descriptor (up to four), we asked them to rank these in descending order of importance. Additionally, participants could also describe their perceived jump using their own descriptors. Finally, for data analysis, we discarded the descriptors of any incorrectly identified oddball jumps.

8.4 Results

Feeling the oddball jump. Table 1 depicts our findings regarding the participants' accuracy in perceiving the oddballs correctly. We found the following average detection rates across all participants: **descent-down** (driving the weight upwards at the apex of their jump, 94% accuracy), **launch-down** (driving the weight downwards in sync with participants' jump, 91% accuracy), **launch-up** (driving the weight upwards in sync with participants' jump, 88% accuracy), **descent-down** (driving the weight upwards as participants touched the ground, 86% accuracy), **descent-up** (driving the weight downwards at the apex of their jump, 81%), **landing-down** (driving the weight downwards as participants touched the ground, 79%), **ascent-up** (driving the weight upwards as participants' feet lifted from the ground, 75% accuracy), and **ascent-down** (66% driving the weight downwards as participants' feet lifted from the ground, 75% accuracy). Finally, we observed an overall average accuracy score of 81% (SD=8.6%). Our results suggest that in many of these actuation modes participants perceived jump modulations from our backpack, **even in the extreme spoofing conditions we designed for this study**—this also suggests that in these conditions the **modified jumping sensation was most likely caused by moving the weight along the participants' backs**.

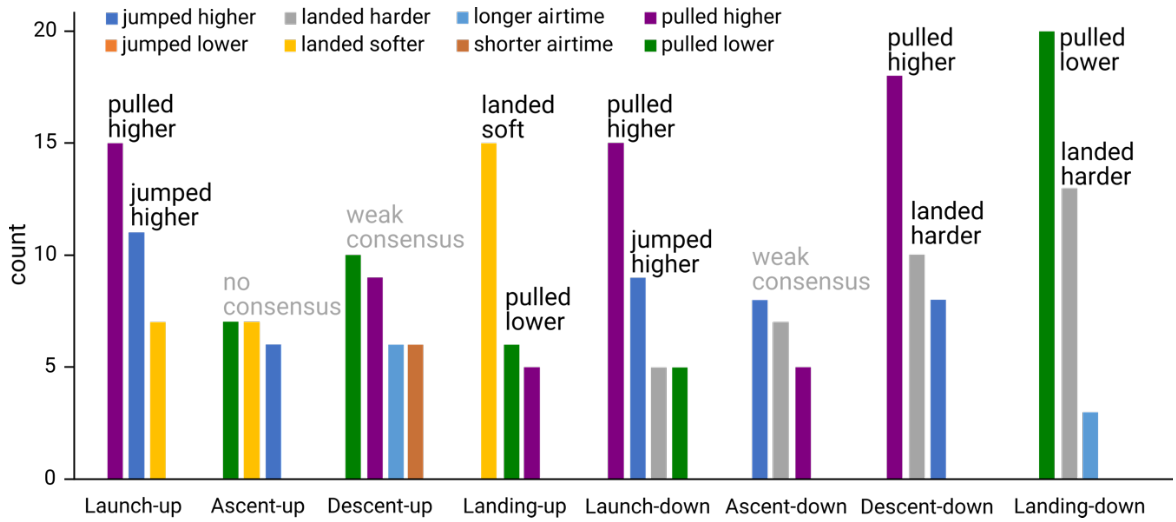

Figure 7: Top-three dominant sensations reported for all eight actuation modes (2 weight up/down movements x 4 jump phases).

Quality of perceived sensation. We now present the individual descriptors to analyze which actuation mode provided uniform agreement across participants—**this is our main finding** since the oddballs were purposefully made hard to recognize via our spoofing. In Figure 7, for the sake of visual clarity, we depict the top-three dominant sensations per actuation mode (unless there were ties, which in case all ties are shown). Furthermore, we annotated cases where there was no consensus across participants (i.e., dominant sensations were tied) or weak consensus (i.e., dominant sensations were close to a tie).

We detailed our results per actuation mode (weight moving up/down and in the different jump phases):

Launch-up. We found that the two most agreed descriptors were *pulled higher*, followed by *jumped higher*.

Ascent-up. We found that there was no consensus between the top descriptors, with participants choosing mostly, with an equal number of votes, between *pulled lower*, *landed softer,* and *longer airtime.*

Descent-up. We found that there was a weak consensus between the top descriptors, with participants choosing, with a fairly equal number of votes, between *pulled lower*, *pulled higher*, *longer airtime,* and *shorter airtime.* This revealed that this actuation mode shows a contradiction between perceived *longer airtime* and *shorter airtime.*

Landing-up. We found that the most agreed descriptor was *landed softer*, followed by a weak consensus with *pulled lower* and *pulled higher.*

Launch-down. We found that the two most agreed descriptors were *pulled higher*, followed by *jumped higher.*

Ascent-up. We found that there was a weak consensus between the top descriptors, with participants choosing mostly, with an equal number of votes, between *jumped higher*, *landed harder,* and *pulled higher.*

Descent-down. We found that most agreed descriptors were *pulled higher*, followed (not closely) by *landed harder.*

Landing-down. We found that the two most agreed descriptors were: *pulled lower*, followed by *landed harder.*

8.5 Discussion

Confirming our findings with psychophysics. We observed, that, even in the extreme spoofing conditions we designed for this study, participants were able to perceive the force added by our device as it accelerated its weight. This is in line with prior work in psychophysics that suggests that participants can notice a 7% increase (Weber's fraction) in force when interacting with a background weight of 5.4kg [13]—this is the total weight of our device. During operation, our device produces a peak force of 56.9N (see *Technical Evaluation*). This force is ~15x larger than Weber's fraction for a perceivable force, according to the aforementioned 7% threshold by [13], and thus noticeable.

Summarizing the results. Figure 8 summarizes the five most voted jump effects with their actuation mode.

9 CONSOLIDATING STUDY#2 RESULTS INTO DESIGN RECOMMENDATIONS FOR JUMP EFFECTS

Prospective designers may have different goals when applying our findings. As such, a prospective designer can traverse the resulting jump effects, which we depicted in Figure 8, differently, depending on their design goals.

1. Maximizing unique effects. A prospective designer may be interested in exploring our device by maximizing the range of jump effects it can induce, allowing for more interactive possibilities. By traversing this table with the goal of *maximizing* effects, one picks an actuation mode for each effect to yield the most unique effects—i.e., not picking the actuation mode that most strongly created each jump effect but an actuation mode that allows one to maximize the number of effects. This yields five possible effects as shown in blue in Figure 9, namely: (1) a sense of jumping higher, (2) a sense of being pulled higher or (3) lower, (4) a sense of landing harder, or (5) softer.

Jump effects	Jump higher	Pulled higher	Pulled lower	Land harder	Land softer
Recommended actuation modes	1. Launch-up 2. Launch-down	1. Descent-down 2. Launch-down 3. Launch-up	1. Landing-down	1. Landing-down 2. Descent-down	1. Landing-up

Figure 8: Summary of the five most voted jump effects from Study#2's findings, with their respective actuation modes.

Jump effects	Jump higher	Pulled higher	Pulled lower	Land harder	Land softer
Recommended actuation modes	**1. Launch-up** ~~2. Launch-down~~	~~1. Descent-down~~ **2. Launch-down** ~~3. Launch-up~~	**1. Landing-down**	~~1. Landing-down~~ **2. Descent-down**	**1. Landing-up**

Figure 9: Recommendation for maximizing unique jump effects, yields up to five sensations.

Jump effects	Jump higher	Pulled higher	Pulled lower	~~Land harder~~	Land softer
Recommended actuation modes	**1. Launch-up** ~~2. Launch-down~~	**1. Descent-down** ~~2. Launch-down~~ ~~3. Launch-up~~	**1. Landing-down**	~~1. Landing-down~~ ~~2. Descent-down~~	**1. Landing-up**

Figure 10: Recommendation for optimizing strongest jump effects, yields up to four sensations.

2. Optimizing for strongest effects. A prospective designer might also be interested in optimizing for different goals, e.g., optimizing for the strongest effects rather than for the number of effects. In this case, they can traverse the table considering only the actuation mode that most strongly created this jump effect. This yields four possible effects as shown in blue in Figure 10, namely: a sense of (1) jumping higher, (2) being pulled higher or (3) lower, and (4) landing softer. This type of design is useful for applications in which the designer wants to elicit only a strong subset of experiences in an environment that might find participants distracted or overwhelmed with additional cues or tasks.

Even more sensations using multimodal combinations. Again, we emphasize that we focused on the sensations that our device *itself* can create, *even in the absence of additional audiovisual cues.* As it is often the case in multimodal interfaces, the addition of visuals (e.g., VR effects that render a sense of movement), or audio (e.g., sounds that suggest landing harder/softer, etc.) effects are likely to become even more unique and diverse (i.e., yielding potentially a larger set of effects). However, we purposefully explored the haptics-only route since it provides a larger application scope, i.e., these results are applicable to VR as well as to interfaces where rendering additional modalities is simply not possible, as we will demonstrate in the example of our two interactive-sports applications.

Using these recommendations for our VR experience. Finally, we depict the usage of these recommendations by detailing how we used this table to design the VR jumping experience used in our third study. We wanted to increase the player's immersion as they: (1) jumped over obstacles with a power-up, (2) were slowed down by strong winds even during their jumps, (3) were slowed down by landing on mud terrain, and (4) smashed pumpkins to collect points. Ideally, these effects would, respectively, yield sensations of: (1) a bigger jump than usual due to the power-up, (2) a sense of being pushed down by the wind, (3) a feel of a softer return to the ground due the mud, and (4) a sense of impact to indicate they successfully crushed a pumpkin. We then traversed the recommendations table to find which actuation modes render these sensations. This yielded, respectively, the following commands that VR experience sends to the backpack: **(1) launch-up** command for jumping higher (boost), **(2) land-down** command for being pulled lower (wind), **(3) land-up** command for landing softer (mud), and **(4) descent-down** command for landing harder (pumpkin).

10 USER STUDY#3: INTERACTIVE USE OF OUR JUMP EFFECTS IN A VR APPLICATION

In our third and final study, we turned to measuring the impact of our jump effects in an interactive context. This study was approved by our ethics committee (IRB22-0064).

Apparatus and VR experience. Participants wore the JumpMod backpack (detailed in *Implementation*) and an HTC VIVE VR headset. We immersed participants in a modified version of the VR jumping experience (shown *Walkthrough*), that asked participants to collect cookies (similar to collecting coins in popular games by walking into it) and smash pumpkins for four minutes (used to make both conditions comparable). Participants experienced four jump effects, namely: jumping higher (power-up), being pulled lower (wind), landing softer (mud), and landing harder (pumpkin)—the design rationale for these effects is described in *Using these recommendations for our VR experience*; in short, we rendered four effects encompassing both midair haptic effects (e.g., jumping higher &

pulled lower), as well as landing haptic effects (e.g., landing harder & landing softer). For fairness with the baseline condition, any jump effect was rendered in all modalities afforded by the VR headset, i.e., we added jumping sounds as well as a 4x multiplier for the avatar's jump height when the jump powerup was active. Finally, to fully test our device in an interactive context we also chose to test a key limitation, the situation in which the backpack's weight needs to be reset prior to the next jump. As such, our modified VR experience featured an earthquake effect (VR view shaking and sound of an earthquake) accompanied by a back-and-forth rocking motion of the weight—this was used anytime the VR determined that the weight needed to be moved to the opposite endpoint prior to the next jump (see *Walkthrough* for the three ways a designer can choose to obfuscate the preparation of the weight). While we could have designed the experience for the easiest method (imperceptibly slow-moving weight) we opted to, instead, explore a faster weight reset that has less impact on the timing of fast-paced experiences such as this VR experience.

Participants. We recruited 12 participants (four identified as female and eight as male; average age was 26.8 years old; SD=7.83). Participants reported an average body mass index (BMI) of 21.96 (SD=3.91). All participants were healthy with no motor impairment. Participants were compensated with 10 USD for their time.

Conditions. Participants played our VR experience twice, one for each interface condition: with **JumpMod** and **without** (baseline)—condition order counterbalanced across participants. In both conditions the VR experience was completely identical: all same visuals, sound effects, and all power-up jumps scaled up by 4x.

Metrics. At the end of each trial, participants rated, using a 7-point Likert scale, their **perceived immersion** (1=not immersed, 7=fully immersed) and **perceived jumping realism** (1=not realistic, 7= indistinguishable from reality). Moreover, after experiencing both conditions, participants were asked to choose their **preferred condition** and explain why. Finally, we asked them to optionally provide any comments on the experience.

10.1 Results

Perceived immersion & jumping realism. To analyze any potential difference between wearing our device against the baseline, with regards to perceived immersion and jumping realism, we utilized a two-tail paired t-test (independent Likert data, following a normal distribution, with a small sample size). We found a statistically significant difference ($p<0.005$) between *immersion* when comparing JumpMod (M=5.58 and SD=0.67) and the baseline (M=4.25 and SD=0.87) as depicted in Figure 11 (a). Moreover, we found a statistically significant difference ($p<0.001$) between *jumping realism* when comparing JumpMod (M=4.67 and SD=0.98) and the baseline (M=3.42 and SD=0.67).

Preferred interface condition. In addition to jumping realism and immersion, we measured participants' preferred interface condition and found an overwhelming majority preferring JumpMod (11 out of 12 participants) as depicted in Figure 11 (b).

Participants' commentaries. 8 (out of 12) participants stated that JumpMod added immersion (P1, P2, P3, P4, P7, P8, P10, P12). For instance, P10 stated, "backpack adds greater significance to the game, makes it more interesting." P2, who started with JumpMod, compared the two conditions by stating "I missed having the backpack on the [baseline]." As expected, three participants (P1, P6, P11) commented that wearing our backpack is more restrictive than wearing nothing. For instance, P11 stated "[it] affected jumps" but added, unprompted, that "[I am] willing to lose comfort but gain more immersion." Surprisingly, one participant (P3) reported, in the final interview, that they experienced cybersickness in the baseline condition, stating "feels like [the VR system] is scaling my jump." However, P3 reported no cybersickness in the JumpMod condition—while this could be due to the added feedback, our study was not designed to test cybersickness and we do not suggest interpreting from a single data point that JumpMod can alleviate that. Finally, no participant commented negatively on the reset of the weight—they were not aware that the earthquake effect was used to reset the weight when needed. In fact, we only observed positive comments about this effect (P3, P7, P12), in line with comments for all other in-game experiences, such as "[it] feels really real when the ground starts cracking!" (P3).

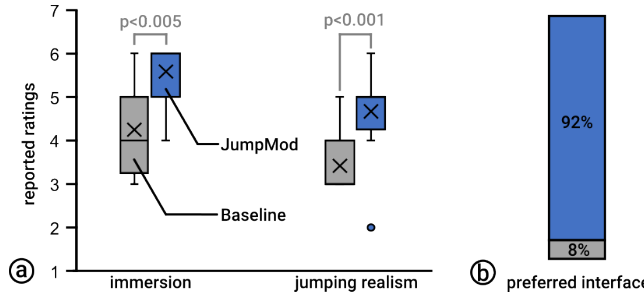

Figure 11: (a) Perceived immersion & jumping realism reported on a Likert scale; (b) Participants' preferred interface.

Study interpretation & limitations. Our results suggest that JumpMod enhanced the immersion of a VR experience that makes use of jumping input, as well as improved the realism of the key interactions in this experience, i.e., jumping. Moreover, we found that, overwhelmingly, participants preferred our device, even if they still felt its added weight, was a limitation. We emphasize that our study is not without limitations and warrants against generalizations beyond this. First, no singular experience can capture the richness of all potential interactive experiences that could use jumping as a key interactive mechanism. Secondly, while our study population exhibits some variance in participants' BMI, it was not exhaustive, and participants with different body abilities were not recruited. Finally, we only measured the immersion and realism reported by participants. We acknowledge that other haptic dimensions were not measured, such as *Harmony, Expressivity,* or *Autotelics (from* [26]).

11 DEMONSTRATING JUMPMOD IN FREE-WALKING VR AND INTERACTIVE-SPORTS

We implemented four applications to demonstrate JumpMod: (1) an in-place VR *farm jumper* (presented in the *Walkthrough*), (2) a free-walking VR *escape-room*, (3) a *jump-rope* trainer, and (4) an *interactive-basketball* game.

11.1 Free-walking & controller-based VR escape-room

Our first application demonstrates a VR *escape room*, in which users solve puzzles by walking and jumping. Figure 12 depicts the interaction in the first of four rooms we designed: (a) the key to solving the puzzle is out of reach even if the user jumps; (b) they find a trampoline, which they grab using the VR handheld controllers; (c) by jumping off the trampoline, (d) they jump higher and grab the key—not only they see a higher jump, but our device creates the feeling of jumping higher. Note that our device is able to render these vertical sensations without encumbering the user's walking (e.g., motion platforms [3], pulleys [27]) or their hands (e.g., handheld propellers [47]).

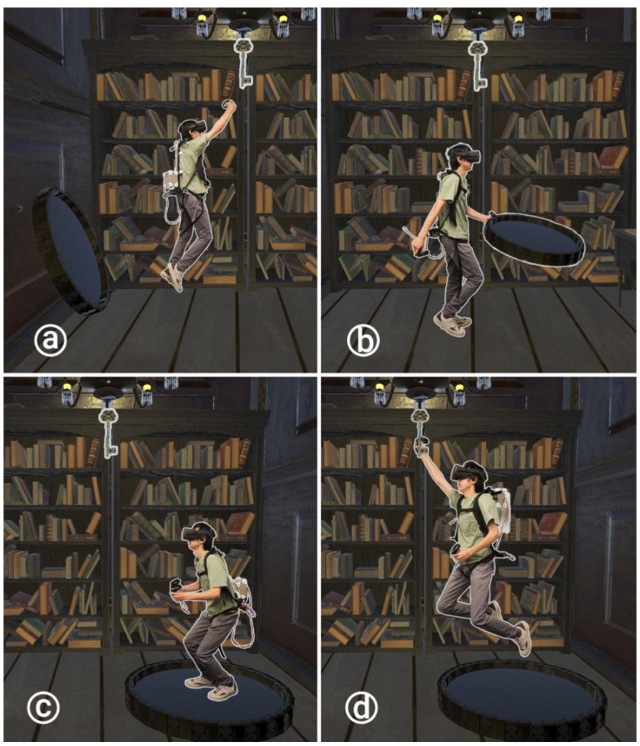

Figure 12: (a) The key is out of reach even if the user jumps to their fullest. (b) They find and move a trampoline, so that (c) by jumping on it (d), they can jump higher and grab the key—they *see* themselves jumping higher and also *feel* it with our device.

11.2 Jump-rope trainer

In this application, we demonstrate JumpMod as the key component in the user's interactive experience, without additional UIs. Figure 13 (a) depicts a user, practicing with a smart jump rope trainer that tracks the rope's position. The user attempts to jump rope but messes up the timing and gets tangled with the rope. By wearing JumpMod as shown in Figure 13 (b), the user *feels* the rope's position rendered via JumpMod (as the rope moves up, the weight moves up, after the rope reaches its apex, the weight, conversely, moves down). This haptic assistance is possible because (unlike propeller-based effects) JumpMod can actuate in two directions (up & down). To enable this experience, we also engineered the rope tracker, shown in Figure 13 (d), which is modeled after existing devices [53]. It detects the revolution of the rope using a magnetic encoder attached to the rope's free-spinning bearing and communicates via Bluetooth.

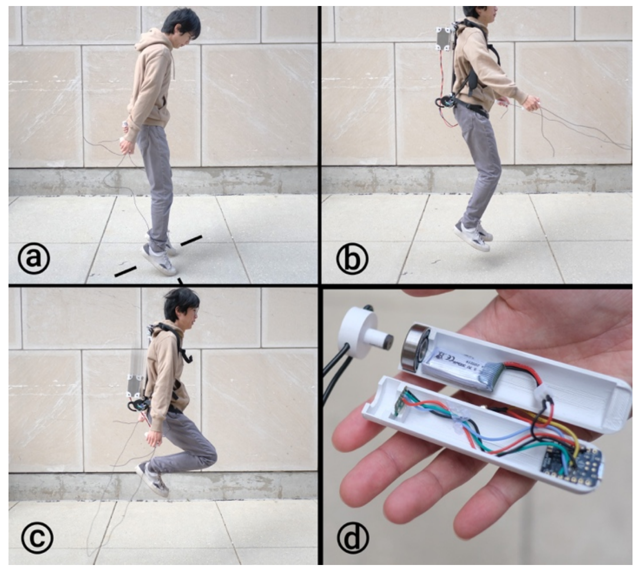

Figure 13: (a) A user attempts to jump rope but messes up the timing and gets tangled. (b) JumpMod helps the user by (c) moving the weight in sync with the rope's position. (d) Our device enables smart jump-ropes to *also* provide users with *output*.

11.3 Interactive basketball game

In our final application, we explored our device in a sports experience where users move around. In this example, two users are playing an interactive basketball game in which they take turns wearing our device and using it as the interactive component of their experience. This experience is made possible by a simple smartphone application we implemented that can send jump commands to our device (implemented using Android SDK [2] and Nordic BLE framework [41]).

Figure 14 depicts the interaction: (a) the opponent, not wearing our backpack, tries to block the player from shooting the ball; (b) seeing they missed the opportunity to block, they tap on a smartphone in their armband, to send the player a *debuff* (a defensive measure to affect the other player), which causes our backpack to render the sensation of pulled lower—negatively affecting their jump when shooting for the hoop; (c) later, the opponent misses another block opportunity, but this time they have exhausted their allowed number of debuffs, so they send a *wildcard* (a riskier move, since it could either be a debuff or a power-up)—unfortunately, this backfires and favors the player wearing JumpMod by rendering a sense of jumping higher—positively affecting their jump when shooting for the hoop.

Figure 14: (a) Interactive basketball with two players. Players take turns wearing the backpack or the smartphone on their arm to send "attacks" to each other, affecting the way their opponent jumps as they shoot the ball to the hoop to score points such as (b) *pulled lower* as a debuff or (c) *jump higher* as a power-up.

12 CONCLUSIONS AND FUTURE WORK

We proposed, implemented, and validated a wearable device, which we call *JumpMod*, that modulates the user's perceived jump. JumpMod achieves this by moving a weight along the user's back, which causes an inertial force that modulates the user's perception of their own vertical momentum. Unlike existing approaches (e.g., wearing or holding large propellers to gain vertical momentum, or sitting or standing on grounded motion platforms), JumpMod modifies the user's perceived jump using its small form factor, small enough that it is worn as a backpack—it is completely untethered, users can walk around freely and do not need to hold any handles to experience its haptic effects.

Throughout three user studies and a technical evaluation, we characterized JumpMod. Most importantly, we found, in our second user study, that JumpMod creates five distinct sensations: (1) a sense of jumping higher, (2) sense of being pulled lower, (3) a sense of landing harder, (4) a sense of landing softer, and (5) a sense of pulled higher. We demonstrated these effects in four applications, including virtual reality and interactive sports.

Finally, as future work, we believe the haptic mechanism behind JumpMod can be extended to create further sensations and be combined with different types of actuators or onboard sensing systems. To accelerate these explorations, we will provide the complete source code and hardware designs as open source[1].

REFERENCES

[1] [1]Altimira, D., Mueller, F. "Floyd," Clarke, J., Lee, G., Billinghurst, M. and Bartneck, C. 2016. Digitally Augmenting Sports: An Opportunity for Exploring and Understanding Novel Balancing Techniques. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, May 2016), 1681–1691.

[2] [2]Android BLE Library: 2022. https://github.com/NordicSemiconductor/Android-BLE-Library. Accessed: 2022-09-14.

[3] [3]Aoki, H. 2001. A study of orientation in a zero gravity environment by means of virtual reality simulation. *AIP Conference Proceedings* (Albuquerque, New Mexico, 2001), 29–34.

[4] [4]Berger, D.R., Schulte-Pelkum, J. and Bülthoff, H.H. 2010. Simulating believable forward accelerations on a stewart motion platform. *ACM Transactions on Applied Perception*. 7, 1 (Jan. 2010), 1–27. DOI:https://doi.org/10.1145/1658349.1658354.

[5] [5]Buckers, T., Gong, B., Eisemann, E. and Lukosch, S. 2018. VRabl: stimulating physical activities through a multiplayer augmented reality sports game. *Proceedings of the First Superhuman Sports Design Challenge: First International Symposium on Amplifying Capabilities and Competing in Mixed Realities* (2018), 1–5.

[6] [6]Cardin, S., Thalmann, D. and Vexo, F. eds. 2007. Head Mounted Wind. *proceeding of the 20th annual conference on Computer Animation and Social Agents (CASA2007)*. (2007).

[7] [7]Cheng, J.-H., Chen, Y., Chang, T.-Y., Lin, H.-E., Wang, P.-Y.C. and Cheng, L.-P. 2021. Impossible Staircase: Vertically Real Walking in an Infinite Virtual Tower. *2021 IEEE Virtual Reality and 3D User Interfaces (VR)* (Mar. 2021), 50–56.

[8] [8]Cheng, L.-P., Lühne, P., Lopes, P., Sterz, C. and Baudisch, P. 2014. Haptic turk: a motion platform based on people. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2014), 3463–3472.

[9] [9]Christensen, R.R., Hollerbach, J.M., Xu, Y. and Meek, S.G. 1998. Inertial Force Feedback For A Locomotion Interface.

[10] [10]Cunningham, T. 2010. System requirements document for the active response gravity offload system (argos). *NASA Engineering directorate document AR&SD-08007*.

[11] [11]Dongsu, W. and Hongbin, G. 2007. Adaptive Sliding Control of Six-DOF Flight Simulator Motion Platform. *Chinese Journal of Aeronautics*. 20, 5 (Oct. 2007), 425–433. DOI:https://doi.org/10.1016/S1000-9361(07)60064-8.

[12] [12]Eidenberger, H. and Mossel, A. 2015. Indoor skydiving in immersive virtual reality with embedded storytelling. *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology* (Beijing China, Nov. 2015), 9–12.

[13] [13]Examining human perception of weight during loaded standing and walking: https://journals.sagepub.com/doi/epdf/10.1177/1071181321651015. Accessed: 2022-12-06.

[14] [14]García-Larrea, L., Lukaszewicz, A.-C. and Mauguiére, F. 1992. Revisiting the oddball paradigm. Non-target vs neutral stimuli and the evaluation of ERP attentional effects. *Neuropsychologia*. 30, 8 (Aug. 1992), 723–741. DOI:https://doi.org/10.1016/0028-3932(92)90042-K.

[15] [15]Gomez, P., Ratcliff, R. and Perea, M. 2007. A Model of the Go/No-Go Task. *Journal of experimental psychology. General*. 136, 3 (Aug. 2007), 389–413. DOI:https://doi.org/10.1037/0096-3445.136.3.389.

[16] [16]Havlík, T., Hayashi, D., Fujita, K., Takashima, K., Lindeman, R.W. and Kitamura, Y. 2019. JumpinVR: Enhancing Jump Experience in a Limited Physical Space. *SIGGRAPH Asia 2019 XR* (New York, NY, USA, Nov. 2019), 19–20.

[17] [17]Hayashi, D., Fujita, K., Takashima, K., Lindeman, R.W. and Kitamura, Y. 2019. Redirected Jumping: Imperceptibly Manipulating Jump Motions in Virtual Reality. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (Mar. 2019), 386–394.

[18] [18]He, L., Xiong, C., Zhang, Q., Chen, W., Fu, C. and Lee, K.-M. 2020. A Backpack Minimizing the Vertical Acceleration of the Load Improves the Economy of Human Walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 28, 9 (Sep. 2020), 1994–2004. DOI:https://doi.org/10.1109/TNSRE.2020.3011974.

[19] [19]Hollerbach, J.M., Mills, R., Tristano, D., Christensen, R.R., Thompson, W.B. and Xu, Y. 2001. Torso Force Feedback Realistically Simulates Slope on Treadmill-Style Locomotion Interfaces. *The International Journal of Robotics Research*. 20, 12 (Dec. 2001), 939–952. DOI:https://doi.org/10.1177/02783640122068209.

[20] [20]Iwata, H., Yano, H., Fukushima, H. and Noma, H. 2005. CirculaFloor. *IEEE Computer Graphics and Applications*. 25, (Jan. 2005), 64–67. DOI:https://doi.org/

10.1109/MCG.2005.5.

[21] [21]Iwata, H., Yano, H. and Nakaizumi, F. 2001. Gait Master: a versatile locomotion interface for uneven virtual terrain. *Proceedings IEEE Virtual Reality 2001* (Mar. 2001), 131–137.

[22] [22]Jung, S., Borst, C.W., Hoermann, S. and Lindeman, R.W. 2019. Redirected Jumping: Perceptual Detection Rates for Curvature Gains. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, Oct. 2019), 1085–1092.

[23] [23]Kang, H.Y., Lee, G., Kang, D.S., Kwon, O., Cho, J.Y., Choi, H.-J. and Han, J.H. 2019. Jumping Further: Forward Jumps in a Gravity-reduced Immersive Virtual Environment. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (Mar. 2019), 699–707.

[24] [24]Kasahara, S., Konno, K., Owaki, R., Nishi, T., Takeshita, A., Ito, T., Kasuga, S. and Ushiba, J. 2017. Malleable Embodiment: Changing Sense of Embodiment by Spatial-Temporal Deformation of Virtual Human Body. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver Colorado USA, May 2017), 6438–6448.

[25] [25]Ke, P., Cai, S., Xu, L. and Zhu, K. 2021. Weighted Walking: Propeller-based On-leg Force Simulation of Walking in Fluid Materials in VR. *SIGGRAPH Asia 2021 Emerging Technologies* (New York, NY, USA, Dec. 2021), 1–2.

[26] [26]Kim, E. and Schneider, O. 2020. Defining Haptic Experience: Foundations for Understanding, Communicating, and Evaluating HX. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2020), 1–13.

[27] [27]Kim, M., Cho, S., Tran, T.Q., Kim, S.-P., Kwon, O. and Han, J. 2017. Scaled Jump in Gravity-Reduced Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*. 23, 4 (Apr. 2017), 1360–1368. DOI:https://doi.org/10.1109/TVCG.2017.2657139.

[28] [28]Kulkarni, S., Fisher, C., Pardyjak, E., Minor, M. and Hollerbach, J. 2009. Wind display device for locomotion interface in a virtual environment. *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (Mar. 2009), 184–189.

[29] [29]Kunze, K., Minamizawa, K., Lukosch, S., Inami, M. and Rekimoto, J. 2017. Superhuman Sports: Applying Human Augmentation to Physical Exercise. *IEEE Pervasive Computing*. 16, 2 (Apr. 2017), 14–17. DOI:https://doi.org/10.1109/MPRV.2017.35.

[30] [30]Li, Y.-J., Jin, D.-R., Wang, M., Chen, J.-L., Steinicke, F., Hu, S.-M. and Zhao, Q. 2021. Detection Thresholds with Joint Horizontal and Vertical Gains in Redirected Jumping. *2021 IEEE Virtual Reality and 3D User Interfaces (VR)* (Mar. 2021), 95–102.

[31] [31]Li, Y.-J., Wang, M., Jin, D.-R., Steinicke, F., Hu, S.-M. and Zhao, Q. 2021. Effects of Virtual Environments and Self-representations on Redirected Jumping. *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (Mar. 2021), 464–465.

[32] [32]Maekawa, A., Kawamura, K. and Inami, M. 2020. Dynamic Assistance for Human Balancing with Inertia of a Wearable Robotic Appendage. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct. 2020), 4077–4082.

[33] [33]Matsumoto, K., Ban, Y., Narumi, T., Yanase, Y., Tanikawa, T. and Hirose, M. 2016. Unlimited corridor: redirected walking techniques using visuo haptic interaction. *ACM SIGGRAPH 2016 Emerging Technologies* (Anaheim California, Jul. 2016), 1–2.

[34] [34]Miermeister, P., Lächele, M., Boss, R., Masone, C., Schenk, C., Tesch, J., Kerger, M., Teufel, H., Pott, A. and Bülthoff, H.H. 2016. The CableRobot simulator large scale motion platform based on cable robot technology. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct. 2016), 3024–3029.

[35] [35]Moon, T. and Kim, G.J. 2004. Design and evaluation of a wind display for virtual reality. *Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, Nov. 2004), 122–128.

[36] [36]Mueller, F., Agamanolis, S. and Picard, R. 2003. Exertion interfaces: sports over a distance for social bonding and fun. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2003), 561–568.

[37] [37]Nabeshima, J., Saraiji, M.Y. and Minamizawa, K. 2019. Arque: artificial biomimicry-inspired tail for extending innate body functions. *ACM SIGGRAPH 2019 Posters* (Los Angeles California, Jul. 2019), 1–2.

[38] [38]Nagao, R., Matsumoto, K., Narumi, T., Tanikawa, T. and Hirose, M. 2018. Ascending and Descending in Virtual Reality: Simple and Safe System Using Passive Haptics. *IEEE Transactions on Visualization and Computer Graphics*. 24, 4 (Apr. 2018), 1584–1593. DOI:https://doi.org/10.1109/TVCG.2018.2793038.

[39] [39]Nagao, R., Matsumoto, K., Narumi, T., Tanikawa, T. and Hirose, M. 2017. Infinite stairs: simulating stairs in virtual reality based on visuo-haptic interaction. *ACM SIGGRAPH 2017 Emerging Technologies* (Los Angeles California, Jul. 2017), 1–2.

[40] [40]Nitta, K., Higuchi, K. and Rekimoto, J. 2014. HoverBall: augmented sports with a flying ball. *Proceedings of the 5th Augmented Human International Conference* (New York, NY, USA, Mar. 2014), 1–4.

[41] [41]Nordic Semiconductor | Specialists in Low Power Wireless: https://www.nordicsemi.com/. Accessed: 2022-09-14.

[42] [42]Oddsson, L.I., Karlsson, R., Konrad, J., Ince, S., Williams, S.R. and Zemkova, E. 2007. A rehabilitation tool for functional balance using altered gravity and virtual reality. *Journal of NeuroEngineering and Rehabilitation*. 4, 1 (Jul. 2007), 25. DOI:https://doi.org/10.1186/1743-0003-4-25.

[43] [43]Ogawa, K., Fujita, K., Takashima, K. and Kitamura, Y. 2022. PseudoJumpOn: Jumping onto Steps in Virtual Reality. *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (Mar. 2022), 635–643.

[44] [44]Razzaque, S., Swapp, D., Slater, M., Whitton, M.C. and Steed, A. Redirected Walking in Place. 8.

[45] [45]Rheiner, M. 2014. Birdly an attempt to fly. *ACM SIGGRAPH 2014 Emerging Technologies* (New York, NY, USA, Jul. 2014), 1.

[46] [46]Rome, L.C., Flynn, L. and Yoo, T.D. 2006. Rubber bands reduce the cost of carrying loads. *Nature*. 444, 7122 (Dec. 2006), 1023–1024. DOI:https://doi.org/10.1038/4441023a.

[47] [47]Sasaki, T., Liu, K.-H., Hasegawa, T., Hiyama, A. and Inami, M. 2019. Virtual Super-Leaping: Immersive Extreme Jumping in VR. *Proceedings of the 10th Augmented Human International Conference 2019* (New York, NY, USA, Mar. 2019), 1–8.

[48] [48]Schmidt, D., Kovacs, R., Mehta, V., Umapathi, U., Köhler, S., Cheng, L.-P. and Baudisch, P. 2015. Level-Ups: Motorized Stilts that Simulate Stair Steps in Virtual Reality. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul Republic of Korea, Apr. 2015), 2157–2160.

[49] [49]Schmidt, H., Hesse, S., Bernhardt, R. and Krüger, J. 2005. HapticWalker—a novel haptic foot device. *ACM Transactions on Applied Perception*. 2, 2 (Apr. 2005), 166–180. DOI:https://doi.org/10.1145/1060581.1060589.

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

[51] [51]Sugamoto, N., Ueta, K., Ujitoko, Y., Sakurai, S., Nojima, T. and Hirota, K. 2019. Inclination Manipulator. *SIGGRAPH Asia 2019 Emerging Technologies* (Brisbane QLD Australia, Nov. 2019), 23–24.

[52] [52]Takahashi, T., Shiro, K., Matsuda, A., Komiyama, R., Nishioka, H., Hori, K., Ishiguro, Y., Miyaki, T. and Rekimoto, J. 2018. Augmented jump: a backpack multirotor system for jumping ability augmentation. *Proceedings of the 2018 ACM International Symposium on Wearable Computers* (Singapore Singapore, Oct. 2018), 230–231.

[53] [53]Tangram Factory Smart Rope ROOKIE Jump Rope - Black/Orange: https://www.apple.com/shop/product/HMXW2ZM/A/tangram-factory-smart-rope-rookie-jump-rope. Accessed: 2022-09-14.

[54] [54]Tiator, M., Köse, O., Wiche, R., Geiger, C. and Dorn, F. 2018. Trampoline Jumping with a Head-Mounted Display in Virtual Reality Entertainment. *Intelligent Technologies for Interactive Entertainment* (Cham, 2018), 105–119.

[55] [55]VescUart: 2022. https://github.com/SolidGeek/VescUart. Accessed: 2022-09-14.

[56] [56]Wolf, D., Rogers, K., Kunder, C. and Rukzio, E. 2020. JumpVR: Jump-Based Locomotion Augmentation for Virtual Reality. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu HI USA, Apr. 2020), 1–12.

[57] [57]Xiu, W., Ruble, K. and Ma, O. 2014. A reduced-gravity simulator for physically simulating human walking in microgravity or reduced-gravity environment. *2014 IEEE International Conference on Robotics and Automation (ICRA)* (May 2014), 4837–4843.

[58] [58]Ye, Y.-S., Chen, H.-Y. and Chan, L. 2019. Pull-Ups: Enhancing Suspension Activities in Virtual Reality with Body-Scale Kinesthetic Force Feedback. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans LA USA, Oct. 2019), 791–801.