2025 International Conference On Rehabilitation Robotics (ICORR)
May 12-16, 2025. Chicago, IL, USA

Comparative Analysis of Temporal Difference Learning Methods to Learn General Value Functions of Lower-Limb Signals

Sonny T. Jones, Grange M. Simpson, Wyatt M. J. Young,
Kylee North, Patrick M. Pilarski, Ashley N. Dalrymple

Abstract— **Millions of people in the United States suffer from paralysis, resulting in significant deficits in motor function. Restricted mobility due to these deficits and the lack of adaptive rehabilitative solutions make traversing complex and challenging terrains unsafe. Exoskeletons offer a promising solution, but their effectiveness could be greatly enhanced by incorporating reinforcement learning algorithms for real-time adaptation to changing environments and the user's unique gait biomechanics. This study explored different temporal difference learning methods for predicting signals recorded from sensors on the lower-limbs, including muscle activation from electromyography, underfoot pressure, and joint angles from goniometers. Specifically, the performance of the temporal difference learning methods TD(λ), TOTD, and SwiftTD to quickly and accurately predict these signals was examined. From initial findings, SwiftTD generally converged faster, while TOTD typically achieved lower convergence errors. These outcomes varied depending on the specific signal that was being predicted, highlighting the need for careful consideration of algorithm choice depending on the signal, accuracy, and speed. The results, therefore, support the informed selection of specific algorithms for providing predictive knowledge to adaptive, machine learning-controlled assistive rehabilitative technologies. These findings will enable the selection of appropriate predictive algorithms, leading to the development of better exoskeletons and other assistive devices to enhance the mobility and quality of life of individuals with motor paralysis.**

I. Introduction

There are an estimated 5.4 million individuals currently in the United States who suffer from motor paralysis, with stroke and spinal cord injury as the leading causes [1]. People with impaired motor function often suffer from muscle stiffness, muscle spasms, and altered muscle activation, including weakness, impaired coordination, and reduced endurance [2]. Many individuals adapt to their restricted mobility, but experience difficulties while walking on more complex terrains such as slopes, curbs, stairs, uneven surfaces, and while turning. There is a critical need to provide personalized adaptive solutions to navigate changing and challenging terrains. The lack of adaptive solutions for rehabilitative devices to navigate changes in terrain transitions limits safe, reliable ambulation in complex environments. Developing responsive rehabilitative solutions can improve overall autonomy and mobility in people with motor impairments.

Current state-of-the-art control methods rely on static approaches such as state machines or kinematic models to define control parameters, which are not predictive and cannot anticipate changes of terrain in dynamic environments [3], [4]. Reinforcement Learning (RL) is a type of machine learning that can both learn and adapt sensorimotor predictions over time [5]. Because of this property, RL-based controllers for exoskeleton control promise to enable adaptation to individual users and varying environmental conditions [6], [7]. RL utilizes a trial-and-error learning paradigm that chooses the best actions and behavior based on learned expectations about future success (i.e., reward) [5]. This allows behavior to beneficially change before future events occur. RL also facilitates adaptation to unforeseen events via mechanisms of exploration and meta-learning [8]. Therefore, RL holds promise for more adaptable exoskeleton control, especially in dynamic environments [7]. Adaptability is particularly important for community ambulation, where individuals encounter various terrains and obstacles that require real-time adjustments to maintain stability and safety.

For adaptable RL-based controllers of exoskeletons, accurate and real-time signal prediction is essential. Temporal difference (TD) learning—the main learning process underpinning RL in both animals and machines [5]—has been shown to learn reliable predictions of various gait-related signals, including electromyography (EMG), underfoot pressure, and joint angles, for applications in prosthesis and exoskeleton control, as well as spinal cord stimulation [9]–[11]. However, there remains a gap in identifying the most effective TD algorithms for approximating these gait-relevant sensor signals. Closing this gap is necessary for enhancing the accuracy and functionality of RL-based controllers that rely on these predictions for real-time decision-making.

This present study explores and compares different TD learning algorithms for the prediction of walking kinetic and kinematic signals relevant to exoskeleton control, including muscle activation from EMG, underfoot pressure, and joint angles. The approach aims to predict these complex biomechanical signals while people walk across different terrains. By providing methods to determine the most appropriate algorithm for this task, groundwork can be laid for future applications that leverage the predictive capabilities of RL to anticipate transitions between different types of terrains dur-

We thank the Departments of Biomedical Engineering and Physical Medicine and Rehabilitation for funding this study. PMP was supported by NSERC, Amii, and the Canada CIFAR AI Chairs program. We also thank our participants who volunteered their time.

S. T. Jones (corresponding author e-mail sonny.jones@utah.edu), G. M. Simpson, W. M. J. Young, and K. North are with the Department of Biomedical Engineering, University of Utah, Salt Lake City, UT, USA.; P. M. Pilarski is with the Alberta Machine Intelligence Institute (Amii), and the Department of Medicine and Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada; A. N. Dalrymple (corresponding author e-mail ashley.dalrymple@utah.edu) is with the Department of Biomedical Engineering and the Department of Physical Medicine and Rehabilitation, University of Utah, Salt Lake City, UT, USA.

2025 International Conference On Rehabilitation Robotics (ICORR) | 979-8-3503-8068-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICORR66766.2025.11063114

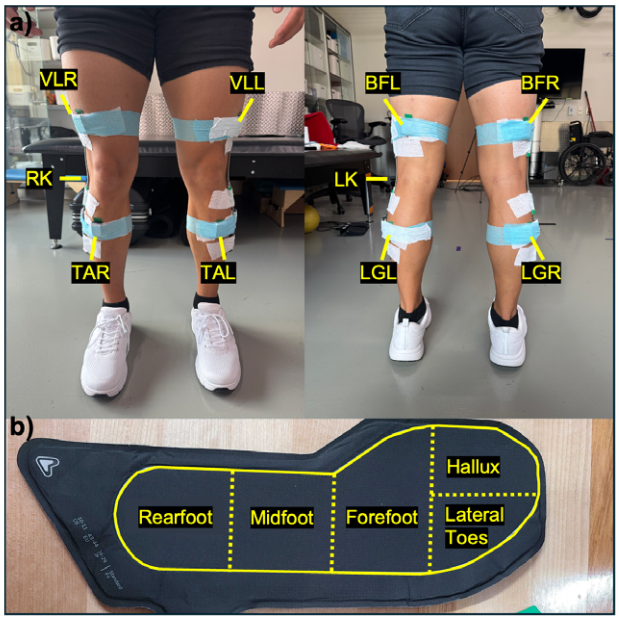

Fig. 1. Experimental setup showing a study participant equipped with the sensor suite. a) EMG sensors are placed on the TAL, TAR, LGL, LGR, VLL, VLR, BFL, and BFR to capture muscle activations during walking. Goniometers are placed on the LK and RK to capture joint angles. b) Pressure insoles with the divided bins.

ing walking. The goal is to create adaptive control strategies for lower-limb exoskeletons that adapt to the user's unique biomechanics and facilitate safe gait over variable terrains.

II. Methods

A. Participants

Nine healthy participants (3 female, aged 19 - 31 years) with no known neuromuscular deficits were recruited for this study. All participants provided written informed consent following the University of Utah Institutional Review Board (IRB00171076).

B. Data Acquisition and Processing

Participants were fitted with 8 wireless EMG sensors, 2 wireless goniometers (Trigno, Delsys, Natick, MA, USA), and 2 wireless pressure-sensing insoles (XSensor, Calgary, AB, Canada). EMG sensors were placed bilaterally on the tibialis anterior (TAL/TAR), lateral gastrocnemius (LGL/LGR), vastus lateralis (VLL/VLR), and biceps femoris (BFL/BFR), with 'L' or 'R' at the end of each abbreviation indicating the left or right leg, respectively. Before sensor placement, each muscle belly was cleaned with abrasive gel and alcohol (LemonPrep, Mavidon, Flat Rock, NC) and treated with conductive electrode gel (Signagel, Parker, Fairfield, NJ). Sensors were secured with double-sided Delsys adhesive and Medipore medical tape (3M, Saint Paul, MN). Each EMG sensor sampled data at 1926 Hz. Goniometers were placed bilaterally on the lateral knee joint (LK and RK) with Medipore tape and sampled joint angle data at 519 Hz. Pressure insoles were placed in the participant's shoes and sampled underfoot pressure at 33 Hz. The sensor setup is illustrated in Figure 1a. Hardware and data collection were controlled using a custom graphical user interface (GUI) built in Python (version 3.8.1, Python Software Foundation, Wilmington, DE) running on a computer with an Intel Core i7 CPU and 32 GB of RAM. Data were collected online and saved for offline analysis.

EMG data were processed using a 4th-order Butterworth bandpass filter (10 Hz high pass, 450 Hz low pass, the function *scipy.signal.filtfilt* doubles filter order). The filtered signal was rectified, and a moving average (window size of 0.1 times the EMG sampling rate) was applied to obtain the signal envelope. The resulting envelope was down-sampled to 33 Hz to match the insole sampling rate. EMG data were normalized to each muscle's maximum voluntary contraction (MVC), where participants maximally contracted the muscle isometrically [12]. Pressure data were normalized to the participant's weight. The pressure insoles contain 235 total sensors. The foot was divided into 5 different regions, and the average value of each region was taken at each time step (Figure 1b) [13]. Pressure signals were smoothed using an exponential moving average with a filter size equal to 11 samples. Goniometer data were down-sampled to 33 Hz and were normalized to [0, 1].

C. Machine Learning Approach

1) Selective Kanerva Coding: Selective Kanerva Coding (SKC) is a function approximation method to reduce a high-dimensional continuous state space into a binary feature vector, shown to be useful with stochastic sensor data in rehabilitation settings [10], [11], [14]. Thirty sensor signals formed the state space: 8 EMG signals, 20 underfoot pressure signals (regions and moving averages), and 2 goniometer signals [10]. Sensor signals were normalized to [0, 1].

SKC was initialized by randomly distributing $K = 2500$ prototypes across the state space. Prototype positions were held constant after initialization. The c closest prototypes to the current state were found according to the Euclidean distance between the current state and the prototypes, sorted with Hoare's quickselect [14]. 3-c values were utilized, such that the resulting binary feature vector had a length equal to $len(c)*K$ long with $sum(c)$ active features, for more flexible representations and more granularity of the state space [10].

2) General Value Functions: Traditionally, RL allows an agent to explore an environment, learning to assign value to actions and states according to the expected, or predicted, future reward. [5]. The value function approximates the expected return, the future cumulative discounted sum of rewards an agent can expect from a particular state. Value functions can also be used to estimate the future values of any arbitrary signal of interest, called the cumulant (Z) [15], [16]. These functions are called General Value Functions (GVFs; [16]). GVFs have previously been used to quickly predict walking-related signals, including EMG, ground reaction forces, and angular velocity [9], [10].

3) Temporal Difference (λ) Learning: Temporal Difference (TD) learning is a fundamental RL algorithm where an agent learns from experience without relying on a model

of the environment [17]. TD learning is the key method used to update predictions of GVFs [16]; it provides good performance and low computational complexity [5], [18]. TD learning adapts the predicted future rewards for a specific state. It uses the observed reward for the current state and the predicted value of the next state to adjust the current prediction, making it closer to the observed reward plus the discounted value of the next state. The core component of this update is the TD error (δ), which calculates the difference between the current state estimate (V) and an estimate of the future state (V'). TD learning uses a discount factor (γ, $0 \leq \gamma \leq 1$) that determines whether the algorithm estimated more immediate or long-term reward.

TD(λ) extends basic TD learning by introducing an additional decaying parameter (λ, $0 \leq \lambda \leq 1$) that determines whether the algorithm prioritizes previous learning or current value estimations to update the predictions [5]. This decaying parameter is combined with an eligibility trace (e) to assign credit to previous states for the current reward or error. The TD(λ) algorithm is outlined in black in Algorithm 1. While TD(λ) is powerful, it can be sensitive to divergence in the online setting from the mathematically optimal solution in settings where function approximation is used, especially with larger step sizes. [18]. TD(λ) uses an accumulated trace to update e, which can lead to unbounded updates. The combination of unbounded updates and large step sizes causes the learning algorithm to become unstable, leading to divergence and poor prediction performance.

4) True Online Temporal Difference Learning: True Online Temporal Difference (TOTD) learning solves the divergence problem of TD(λ) learning by adding extra terms to the update rule for the eligibility trace and weight vectors [18]. The additional components introduced by TOTD are shown in red in Algorithm 1. Specifically, TOTD uses a Dutch trace to update the eligibility trace, which bounds the eligibility update. Additionally, TOTD introduces a TD error correction to account for the change in the estimates of the predictions over time. This correction helps to maintain consistency with the online mathematical update, leading to more accurate predictions without divergence. These improvements enable more stable learning, particularly with larger step sizes. TOTD is also computationally more efficient than TD(λ), especially with larger state spaces [18].

5) SwiftTD: SwiftTD aims to improve TD(λ) learning by introducing adaptive step sizes with maximum bounds that decay over time [19]. This approach solves the problem of large step size sensitivity in TD methods, allowing faster learning while maintaining stability. Instead of updating the weight vector with a single step size, this algorithm uses a custom step size to adjust each weight. SwiftTD dynamically adjusts each step size based on the magnitude of previous updates, helping to overcome slow learning from small updates. A meta-step size (θ) is employed to update the step sizes of the weights. A step size bound (η) and decay (ϵ) are introduced to combat divergence from overly large updates. The additional components added by SwiftTD are highlighted in blue in Algorithm 1.

Algorithm 1 Learning GVFs with TD Methods

Indicators: TD(λ), TOTD, SwiftTD
Input: $\lambda, \gamma, \alpha, \theta, \epsilon, \eta, \alpha_{init}$
Initialize: $w, x, V_{old}, S, e, \bar{e}, p, h \leftarrow 0, \beta \leftarrow \alpha_{init}$
Repeat every time-step:
Generate next state S' and cumulant Z'
$x' \leftarrow SKC(S')$
$V \leftarrow w^T x$, $V' \leftarrow w^T x'$
$\delta \leftarrow Z + \gamma V' - V$ {TD Error}
$T \leftarrow e^T x$
$E \leftarrow max(\eta, \alpha^T x^2)$
$V_{diff} \leftarrow V - V_{old}$
$e \leftarrow \gamma\lambda e + x - \alpha\gamma\lambda(e^T x)x$ {TD(λ), TOTD}
$w \leftarrow w + \alpha(\delta + V_{diff})e - \alpha V_{diff} x$ {TD(λ), TOTD}
for i = 1, 2, ..., n **do**
  $\alpha_i \leftarrow e^{\beta_i}$
  $e_i \leftarrow \gamma\lambda e_i + \frac{\eta}{E}\alpha_i x_i - \alpha_i\gamma\lambda T x_i$
  $w_i \leftarrow w_i + \delta e_i - \alpha_i x_i V_{diff}$
  $p_i \leftarrow \gamma\lambda p_i + x_i h_i$
  $\beta_i \leftarrow \beta_i + \frac{\theta}{\alpha_i + e^{-8}}\delta h_i$
  $\bar{e}_i \leftarrow \gamma\lambda\bar{e}_i + \alpha_i x_i[1 - \gamma\lambda T - \gamma\lambda x_i \bar{e}_i]$
  $k \leftarrow h_i$
  $h_i \leftarrow h_i[1 - \alpha_i x_i^2] - h_i^{old} x_i[e_i - x_i\alpha_i] + \delta\bar{e}_i - x_i\alpha_i V_{diff}$
  $h_i^{old} \leftarrow k$
  $if(E > \eta \; and \; x_i \neq 0) \; : \; \beta_i = \beta_i - ln(\epsilon)$
end for
$V_{old} \leftarrow V', x \leftarrow x'$

TABLE I
LEARNING PARAMETERS FOR TD ALGORITHMS

Parameter	TD(λ)	TOTD	SwiftTD
λ	0.5	0.5	0.9
α	0.015	0.015	NA
α_{init}	NA	NA	0
θ	NA	NA	-2
ϵ	NA	NA	0.9
η	NA	NA	0.2

6) Learning Parameter Selection: The learning parameters used for each algorithm were selected using a grid search on a subset of data to minimize prediction error, comparing the predicted GVF and the actual return on a subset of the data (Table 1).

D. Experimental Protocol

Participants performed MVCs for each muscle group for EMG signal normalization. Participants walked across designated tracks with various terrains, including even ground, uneven ground, up and down ramps, upstairs, downstairs, and left and right turns (Figure 2). The experiment was comprised of a total of 10 trials. Participants first walked across Track 1 four times, then three times across Track 2, and concluded by walking across Track 3 three times. Continuous data were collected from all sensors while participants transversed the terrains during the walking trials. Transitions between terrains were synchronously marked when the participant first stepped onto a new terrain surface.

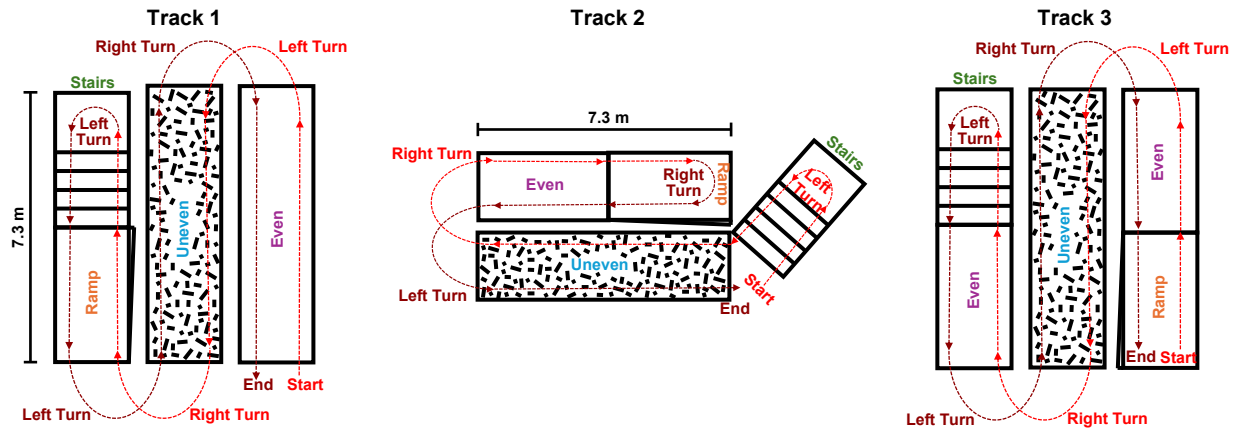

Fig. 2. Participants begin walking on the track from the 'Start' and follow the arrows until the 'End' mark. The straight portion of the terrain track was 7.3 m long. Terrain tracks include even ground, uneven ground, up and down ramps, upstairs, downstairs, and left and right turns. Participants first began walking on track 1 four times, followed by three times on track 3, and finished with three times on track 3.

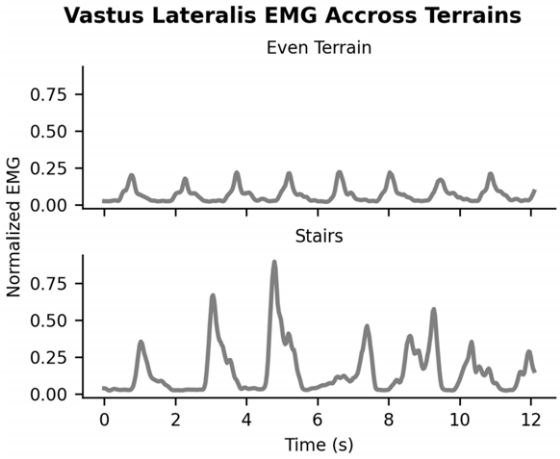

Fig. 3. Representative EMG captured from the VLL muscle from P3 walking on different terrains, showing larger VLL activation on stairs compared to even ground.

E. Analyses and Statistics

Each TD learning method was used to obtain GVFs for EMG, pressure, and goniometer signals as cumulants at multiple time scales: 0.25, 0.5, 1.0, and 2.0 seconds. In total, 360 GVFs were learned in parallel [16]. The γ values were calculated using $\gamma = 1 - \frac{1}{T}$ where $T = fs * timescale$ and were 0.879, 0.940, 0.970, and 0.985, respectively [10]. These time scales capture the immediate and longer-term dynamics of the sensor signals.

The performance of each algorithm was evaluated by comparing the convergence time and the root mean squared error (RMSE) between the expected and predicted returns. [15], [20]. This was done by fitting the learning curve with a single breakpoint piecewise regression, where the second line was fitted to the flat, stabilized error of the learning curve. The x-value, corresponding to timesteps, at 10% along the second fitted line was taken as the convergence time, and the y-intercept was taken as the convergence error. This enabled the assessment of the speed and accuracy of learning GVFs using each TD learning method.

Normality was assessed using the Shapiro-Wilk Test. Data were found to be non-normally distributed; therefore, the Kruskal-Wallis test was used to compare the convergence times and errors between the TD learning methods. Metrics were compared using Dunn's Test with a Holms-Sidak correction to correct for multiple comparisons.

III. Results and Discussion

The time to featurize the data and obtain predictions using GVFs was 9.8 $\pm$ 0.8 ms for TD(λ), 13.9 $\pm$ 0.9 ms for TOTD, and 22.3 $\pm$ 1.6 ms for SwiftTD. Therefore, the computation times allow for real-time implementation within our 33 Hz (30.3 ms) loop time.

A. Sensor Signals Differ Between Different Walking Terrains

As participants walked across the terrains, the sensor signals exhibited distinct differences in shape and amplitude. For example, during stair ascent, larger activations of the VL muscle were required to lift the individual up the stairs (Figure 3). Larger VL activation was additionally associated with an increase in underfoot pressure towards the forefoot and hallux as they pushed off the ball of their foot during stair ascent. Although this was the most noticeable example, there were differences in all signals between terrains.

B. SwiftTD Outperforms TOTD in Convergence Time But Can Have Higher Convergence Error When Learning GVFs

The expected return was compared to the predictions of the raw signal by the TD methods (Figs. 4 and 5). Figure 5 shows the raw signal, expected return, and predicted return from the TD learning algorithms. The expected return represents a temporal abstraction of the cumulative discounted signal, which TD learning methods try to predict [15]. Generally, the expected return anticipated the rise and fall of the raw signal, demonstrating successful signal prediction.

Convergence error and time were derived from RMSE curves, comparing the predicted signal to the expected return for each sensor signal. The predictions for TD(λ) and TOTD were nearly identical (Figure 5), and there was no statistical difference in convergence time and error between both methods; therefore, TD(λ) was excluded from further

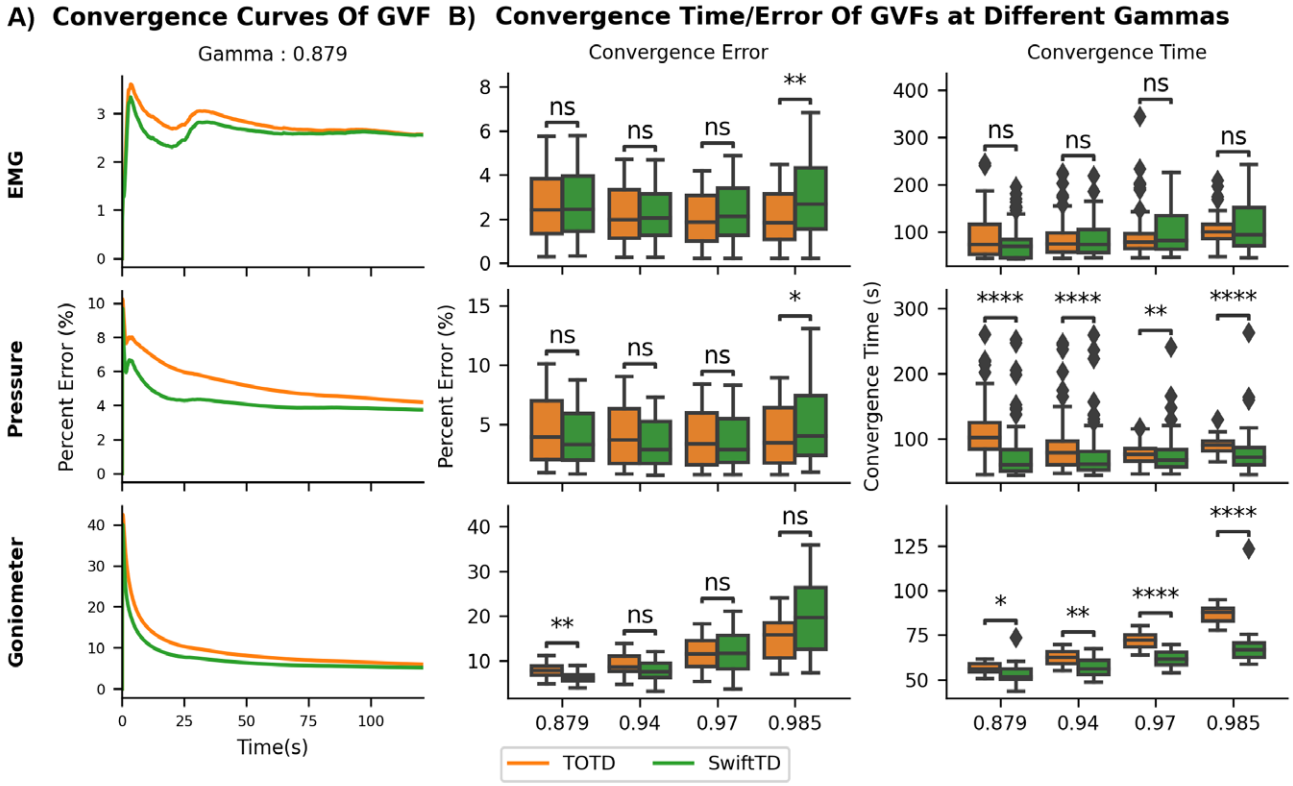

Fig. 4. Convergence error and convergence time performance metrics for evaluating TD learning algorithms. (A) The average RMSE convergence curves for EMG, underfoot pressure, and goniometer signals at the discount factor (γ) of 0.879. Error is reported as the percentage of the maximum cumulant value. (B) Boxplots contain the convergence errors and times for EMG, underfoot pressure, and goniometer signals (ns: $p \geq 0.05$, *: $p < 0.05$, **:$p < 0.01$, ***: $p < 0.001$, ****:$p < 0.0001$). Statistical annotations done using the *Statsannotation* package (v0.2.3) [21]

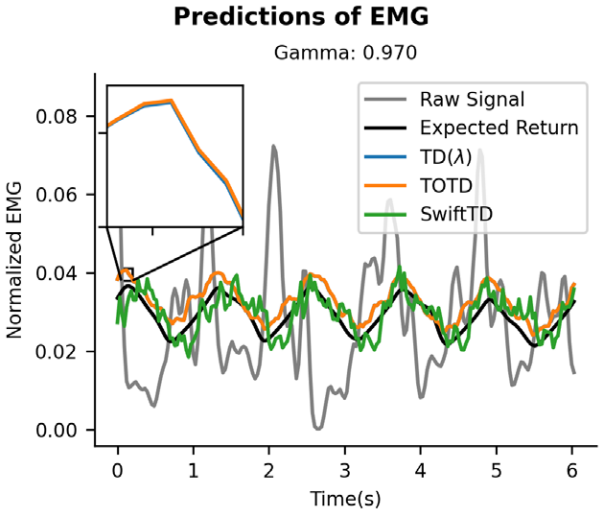

Fig. 5. Comparison of representative EMG signal, the expected return of predictions, and predictions made by three TD learning algorithms (TD(λ), TOTD, and SwiftTD) of P2's TAL muscle. Note the varying degrees of accuracy and anticipatory behavior exhibited by each algorithm compared to the expected return. TD(λ) predictions overlap the TOTD predictions.

analyses, and all comparisons were made between TOTD and SwiftTD. Figure 4A illustrates the convergence curves from TOTD and SwiftTD for three different sensor signals.

For EMG prediction, no significant differences in convergence error were observed across γ values of 0.879 (0.25s), 0.94 (0.5s), and 0.97 (1s) ($p = 0.996$, $p = 0.931$, $p = 0.267$). At γ = 0.985 (2s), however, TOTD demonstrated a 27.0% lower error than SwiftTD ($2.08\% \pm 1.2\%$ vs $2.85\% \pm 1.6\%$) ($p = 0.007$). Convergence times for both algorithms were not different across all γ values ($p = 0.495$, $p = 0.870$, $p = 0.516$, $p = 0.895$). TOTD generally minimized error when learning GVFs on EMG signals with higher γ values. The lack of difference in convergence time indicates that both algorithms sufficiently learned GVFs of EMG signals.

Convergence errors for predicting underfoot pressure were not different across γ values of 0.879 (0.25s), 0.94 (0.5s), and 0.97 (1s) ($p = 0.216$, $p = 0.246$, $p = 0.975$). However, at γ = 0.985 (2s), TOTD outperformed SwiftTD with a 21.4% lower convergence error ($3.98\% \pm 2.50\%$ vs. $5.06\% \pm 3.25\%$, $p = 0.034$). SwiftTD converged faster than TOTD in all cases, with the largest performance increase at γ = 0.879, where SwiftTD converged 33.9% faster ($71.0s \pm 36.4s$ vs $105.5s \pm 40.6s$) ($p = 0.003$, $p < 0.001$, $p < 0.001$, $p < 0.001$). Based on these results, GVFs of pressure signals learned using TOTD at γ = 0.985 minimized the convergence error. For lower γ values, SwiftTD minimized the convergence time with errors similar to those of TOTD.

The most noticeable convergence error difference for predicting joint angle signals was at γ = 0.879 (0.25s), where SwiftTD converged with a 20.5% lower average error compared to TOTD ($6.31\% \pm 1.27\%$ vs $7.94\% \pm 1.58\%$) ($p = 0.009$). SwiftTD converged faster than TOTD in all cases, with the biggest performance increase at γ = 0.985,

with SwiftTD converging faster by 19.1% ($69.7s \pm 13.8s$ vs $86.2s \pm 5.2s$) ($p < 0.001$). These results suggest that a lower γ value of 0.879 for the joint angle prediction task was suitable for minimizing convergence error with SwiftTD. Additionally, SwiftTD excelled in convergence time. For learning GVFs of goniometer signals at higher γ values, SwiftTD minimized convergence time with similar convergence errors compared to TOTD.

These results extend prior work using TD methods for predicting kinetic and kinematic signals [9], [10]. The current work showed that choices in prediction timescale and learning algorithms affect prediction accuracy and learning time of GVFs. Convergence error increased with higher γ values, indicating a potential trade-off between temporal abstraction and prediction accuracy. This highlights the importance of careful algorithm selection and parameter tuning based on the application requirements and type of signal.

Although valuable insights are extracted from this comparative analysis, it is important to acknowledge some limitations. This analysis compared convergence speed and error as primary evaluation metrics, but other factors may be relevant in real-world applications. For example, the computational complexity of each algorithm is not discussed but may be pertinent to real-time implementation. Additionally, the TD algorithms chosen for this comparison were a focused subset of those currently in use. Future studies could add other TD algorithms, such as Gradient TD, for a more comprehensive analysis. Future research will assess the prediction performance across each terrain to understand how a changing environment impacts learning.

IV. Conclusion

There currently remains a gap in empirical evidence regarding the most effective TD algorithms for predicting lower-limb sensor signals during walking. This study assessed the performance of TD learning algorithms for predicting walking-related sensor signals during walking tasks relevant to the control of exoskeletons and related standing-and-walking rehabiltiation technology. As a main contribution, evidence was found in this setting that SwiftTD favors learning efficiency, while TOTD favors learning accuracy. Careful consideration of TD learning algorithms is essential to balance the trade-off between accuracy and temporal abstraction when deployed. Future work includes assessing the impact of TD-learned predictive knowledge online in an exoskeleton controller to provide adequate and timely assistance for individuals with motor deficits. These findings reinforce the idea that an exoskeleton controller integrated with RL could adapt to an individual's unique biomechanics, paving the way for truly personalized assistive technologies.

References

[1] B. S. Armour, E. A. Courtney-Long, M. H. Fox, H. Fredine, and A. Cahill, "Prevalence and causes of paralysis—united states, 2013," *American journal of public health*, vol. 106, no. 10, pp. 1855–1857, 2016.

[2] C. M. Cirstea, "Gait rehabilitation after stroke: should we re-evaluate our practice?" pp. 2892–2894, 2020.

[3] D. Archangeli, B. Ortolano, R. Murray, L. Gabert, and T. Lenzi, "A Lightweight Powered Hip Exoskeleton with Parallel Actuation for Frontal and Sagittal Plane Assistance," *IEEE Transactions on Robotics*, pp. 1–17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10874165/

[4] S. A. Murray, K. H. Ha, C. Hartigan, and M. Goldfarb, "An Assistive Control Approach for a Lower-Limb Exoskeleton to Facilitate Recovery of Walking Following Stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 3, pp. 441–449, May 2015. [Online]. Available: https://ieeexplore.ieee.org/document/6876184/

[5] R. S. Sutton, "Reinforcement learning: An introduction," *A Bradford Book*, 2018.

[6] M. Tucker, M. Cheng, E. Novoseller, R. Cheng, Y. Yue, J. W. Burdick, and A. D. Ames, "Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3423–3430.

[7] Q. Zhang, J. Si, X. Tu, M. Li, M. D. Lewek, and H. Huang, "Toward task-independent optimal adaptive control of a hip exoskeleton for locomotion assistance in neurorehabilitation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024.

[8] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," *arXiv preprint arXiv:1803.11347*, 2018.

[9] P. M. Pilarski, T. B. Dick, and R. S. Sutton, "Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2013, pp. 1–8.

[10] A. N. Dalrymple, D. A. Roszko, R. S. Sutton, and V. K. Mushahwar, "Pavlovian control of intraspinal microstimulation to produce over-ground walking," *Journal of neural engineering*, vol. 17, no. 3, p. 036002, 2020.

[11] P. Faridi, J. K. Mehr, D. Wilson, M. Sharifi, M. Tavakoli, P. M. Pilarski, and V. K. Mushahwar, "Machine-learned adaptive switching in voluntary lower-limb exoskeleton control: Preliminary results," in *2022 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2022, pp. 1–6.

[12] G. Avdan, S. Onal, and B. K. Smith, "Maximum voluntary contraction (MVC) techniques to normalize lower limb muscle groups in young healthy subjects," *IIE Annual Conference Proceedings*, 2022.

[13] A. Searle, M. J. Spink, C. Oldmeadow, S. Chiu, and V. H. Chuter, "Calf muscle stretching is ineffective in increasing ankle range of motion or reducing plantar pressures in people with diabetes and ankle equinus: A randomised controlled trial," *Clinical biomechanics*, vol. 69, pp. 52–57, 2019.

[14] J. B. Travnik and P. M. Pilarski, "Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective kanerva coding," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 1443–1450.

[15] A. White, "Developing a preditive approach to knowledge," Ph.D. dissertation, University of Alberta, 2015.

[16] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 2011, pp. 761–768.

[17] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, pp. 9–44, 1988.

[18] H. Van Seijen, A. R. Mahmood, P. M. Pilarski, M. C. Machado, and R. S. Sutton, "True online temporal-difference learning," *Journal of Machine Learning Research*, vol. 17, no. 145, pp. 1–40, 2016.

[19] K. Javed, A. Sharifnassab, and R. S. Sutton, "SwiftTD: A fast and robust algorithm for temporal difference learning," *Reinforcement Learning Journal*, vol. 2, pp. 840–863, 2024.

[20] A. Kearney, A. J. Koop, and P. M. Pilarski, "What's a good prediction? challenges in evaluating an agent's knowledge," *Adaptive Behavior*, vol. 31, no. 3, pp. 197–212, 2023.

[21] F. Charlier, M. Weber, D. Izak, E. Harkin, M. Magnus, J. Lalli, L. Fresnais, M. Chan, N. Markov, O. Amsalem, S. Proost, A. Krasoulis, getzze, and S. Repplinger, "Statannotations," 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7213391