Hierarchical Integration of Predictive Representations of State from General Value Functions

Anonymous authors
Paper under double-blind review

Abstract

In this work, we investigate how predictive representations of state in the form of continually learned General Value Functions (GVFs) interact with downstream policy networks. Intelligent agents deployed in real-world environments need to adapt to changing conditions in their environment. Adapting to one's environment requires a model or representation of the environment on which to base decision-making. Models that take the form of predictions and GVFs have been shown to provide temporally abstracted predictive representations of state that can forecast useful elements of an agent's or environment's future behaviour. While GVFs have been concretely deployed in rehabilitation and robotic domains, existing approaches treat predictions as input features into model frameworks, without examining or comparing how best to integrate them into downstream learning processes. In this work, we compare multiple strategies for integrating observations and GVF predictions into another learning architecture: 1) actual observations solely in the input layer, 2) predictions solely in the input layer, 3) actual observations and predictions in the input layer, and 4) actual observations in the input layer and predictions in the later latent representations. We evaluate these strategies in a rehabilitation setting, using GVFs to learn predictive representations of kinetic and kinematic signals collected from wearable sensors on the lower limb during ambulation across varied terrains, and policy networks to classify walking terrain.

1 Introduction

Lower-limb exoskeletons have the potential to aid in the rehabilitation of walking for individuals who have suffered a variety of neuromuscular deficits, including stroke and spinal cord injury Luo et al. (2023). Day-to-day community ambulation requires the ability to modify walking to adapt to the characteristics of the environment and walking terrain Balasubramanian et al. (2014). However, current commercial exoskeleton controllers are not adaptive, relying on manual physical input by the user to switch to different fixed control strategies, limiting their effectiveness in real-world use Baud et al. (2021). There is a need to develop adaptive exoskeleton controllers to aid in safe transitioning and traversing across variable terrains. The development of these control strategies for assistive devices can enhance the overall autonomy and mobility of individuals who suffer from motor impairments.

Hierarchical Reinforcement Learning (HRL) offers a promising approach to simplifying complex tasks by breaking them down into multiple simpler sub-tasks that are adaptable over time. Previous work has utilized an architecture consisting of general value functions (GVFs) and RL agents for HRL to make gesture control decisions in an Android environment Comanici et al. (2022). We employ a hierarchical learning (HL) approach to reduce the control problem into two components: 1) a higher-level module to adaptively predict which terrain an individual is walking on, and 2) a lower-level module that provides predictive information of lower-limb sensor signals in the form of GVFs to the higher-level framework to make control decisions. GVFs produce temporal abstractions

of future signal values of multiple lower-limb sensors during walking, including electromyography (EMG), underfoot pressure, and knee joint angles.

Prior work has shown that predictions from GVFs can act as inputs for a policy network Pilarski et al. (2013) that controls the actions of a learning agent. However, integrating these predictions into the latent layers of a deep neural network might integrate abstract predicted information more effectively and improve decision-making Sherstan et al. (2020). We examine four integration methods with different combinations of actual and predicted signals into a policy network. The first method uses the actual sensor signals as input to the policy network. The second method solely utilizes predicted sensor signals in the input layer to the policy network. The third method involves inputting actual and predicted sensor signals together in the input of the policy network. The fourth method adds predictions into the latent space after encoding actual signals into the network input. We aimed to determine which approach leads to better performance in predicting the terrain an individual is walking across. This can create adaptive control policies for walking over different terrains using assistive devices.

2 Methods

2.1 Data Collection and Preprocessing

Ten (24.7 ± 3.2 years old, 3 female) individuals wore a suite of sensors while walking over different terrains. Sensors included EMG for muscle activity, goniometers for knee joint angles, and pressure-sensing insoles for weight distribution under the foot. EMG data were first filtered with a 1st-order high-pass filter (0.1 Hz), rectified, and then filtered using a 1st-order low-pass filter (5 Hz). The goniometer signals were filtered with a 2nd-order low-pass filter at 5 Hz. Signals were then down-sampled to 33 Hz and normalized between 0 and 1, resulting in 30 sensor signals to form the state space. Terrains included even ground, uneven ground, up and down stairs, up and down ramps, and turns, with transitions marked synchronously with data collection. Each participant dataset resulted in approximately 14,000–18,000 timesteps, or approximately 425–545 gait cycles.

Data were sequentially input into the learning architecture to simulate online learning. The first task involved learning predictions of walking-related sensor signals approximately 0.5 seconds into the future (17 timesteps at 33 Hz). GVFs were learned on-policy using true online temporal-difference learning (TOTD). The second task involved learning terrain predictions using actual and predicted signals, with an initial learning rate of $\alpha = 0.001$. We describe the specific computational methods used and the choices in the subsections below.

2.2 Selective Kanerva Coding

Function approximation methods are required to represent a continuous state space, such as a range of sensor signal values. We randomly distributed $K = 5000$ prototypes in the 30-dimensional state space. The locations of these prototypes were held constant. The 500, 100, and 25 closest prototypes to the current state became "active", or set to one, to indicate the location of the current state (defined by the values of the sensor signals) within the state space Travnik & Pilarski (2017). Using these sets of closest prototypes to the current state provides coarse and fine representations of the state Dalrymple et al. (2020). The prototype activations were outputted as a binary feature vector and used to generate the GVFs.

2.3 General Value Functions (GVFs)

RL agents use a value function to approximate the expected return, the future cumulative discounted sum of rewards an agent can expect from a particular state Sutton (1988). However, value functions can approximate any signal of interest, called a cumulant (Z) White (2015); Sutton et al. (2011). Value functions used in this fashion are called GVFs. A GVF learned online can be used to ask questions regarding the current behavior of robotic systems, making temporally abstracted predic-

tions of the cumulative discounted signal. GVFs utilize a γ value between 0 and 1 to represent how far the abstractions are made in the future. For these experiments, we set γ to 0.94. Timesteps into the future can be calculated using the equation: $timesteps = \frac{1}{1-\gamma}$. A GVF with γ = 0.94 will predict 17 timesteps into the future. With regard to learning GVFs of sensor signals, the prediction learning task required predicting gait-related signals in real-time. This was done using GVFs, with SKC representing the continuous state space and TOTD (described below) updating the predictions made by the GVFs (Figure 1b). This framework allows the GVFs to rapidly learn and predict signals from the lower limb, which we call the "lower-level module (fast learning)" (Figure 1c).

2.4 True Online Temporal-difference (TOTD) Learning

Temporal-difference (TD) learning is a core algorithm in RL, allowing agents to learn from their experiences without needing a model of the environment. Sutton (1988). TD learning has been used to learn GVFs due to their low computational cost and performance Van Seijen et al. (2016). TOTD is an up-to-date TD learning algorithm that follows the equivalent ideal mathematical outcomes. It has superior online performance in many settings compared to other TD learning methods Van Seijen et al. (2016). We set λ, the eligibility trace parameter, to 0.5 based on previously tested values.

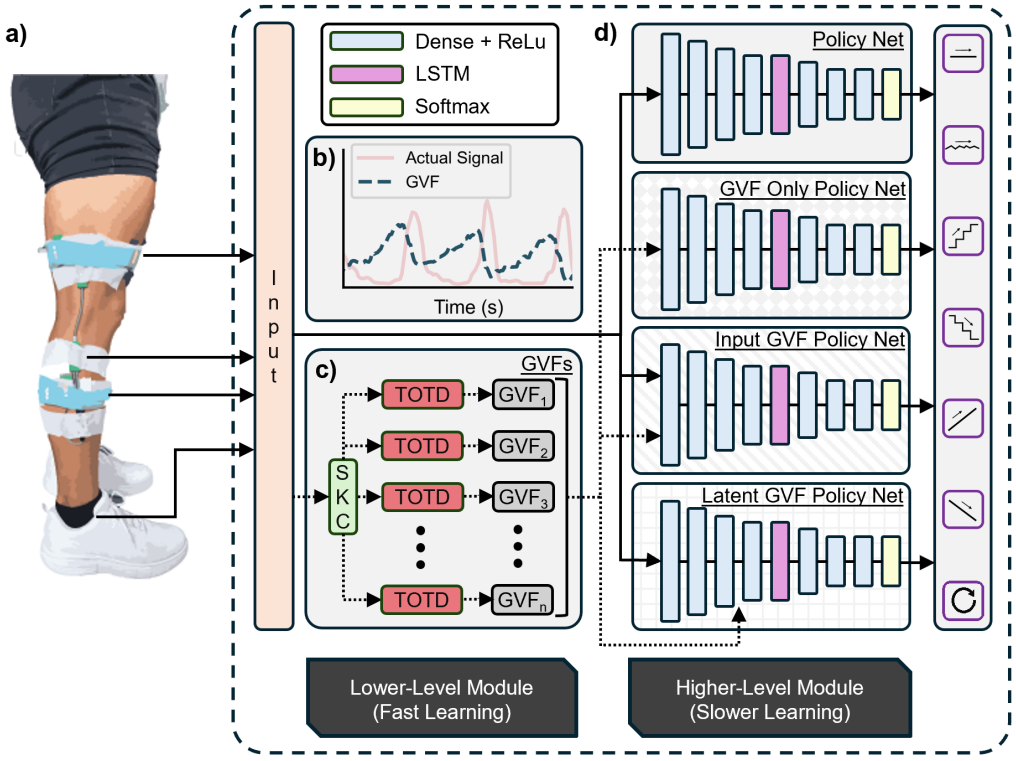

Figure 1: Flowchart of the HL architecture. a) Overall system diagram showing the flow from lower-limb sensor signal acquisition through prediction algorithms and terrain policy networks, b) Example graph of the actual signal from an EMG sensor placed on the lower-limb and the predictions of that signal from a GVF, c) Learned GVFs used for predicting future sensor signals, d) Four policy network configurations illustrating where GVF predictions and actual sensor signals were integrated into the networks.

2.5 Policy Network for Terrain Control Decisions

We updated the policy networks during training using actual and predicted gait-related signals to classify the appropriate terrain in real-time. For this task, we employed a continual deep learning network that utilized a replay buffer to sample previously acquired samples as batch inputs to our network Sherstan et al. (2020); Rolnick et al. (2019). We adopted the sampling strategy from the CLEAR framework to minimize catastrophic forgetting, which creates new batches for continual deep learning by allocating portions of the batch for new and replayed data Rolnick et al. (2019).

The policy network was updated with a 50-50 new-replay batch of size 32, where half of the data came from the most recently collected samples and the other half was randomly sampled from a replay buffer of size 1000. This approach ensured that the network benefited from both recent and past experiences. We deem the policy network the "higher-level module (slower learning)", as deep learning architectures typically require more iterations and time to learn than the GVFs.

Actual signals were acquired and input into the fast predictive state to obtain the predictions made by the GVFs. We compared embedding GVFs directly into the input stream compared to embedding GVFs into the network's latent space, and compared both to a control net without GVFs (only actual sensor signals). All policy networks (nets) used a series of decreasing-size encoding layers to transform the original data into compressed latent representations Sherstan et al. (2020). For the GVF Only Policy Net, the GVF predictions were directly input to the network. For the Input GVF Policy Net, the GVF predictions were directly appended to the actual signals and both were input into the network. For the Latent GVF Policy Net, the GVF predictions were appended to the post-encoding layer to the output of the actual signal input encoding. The Policy Net with no GVFs as inputs was trained as a control net. The architecture for the policy networks are shown in Figure 1d.

2.6 Statistics

Statistics: Data normality was assessed using the Shapiro-Wilk test. Due to the normal distribution, we used a repeated-measures ANOVA to compare accuracies between all three policy nets. We utilized pairwise comparisons with a Bonferroni correction to determine significant differences between individual terrain predictions between networks.

3 Experiments and Results

3.1 Learning Performance Over Time

Figure 2 illustrates the accuracy convergence curves for all the policy networks from the beginning to the end of learning the sequential gait data from all participants. Each policy network correctly

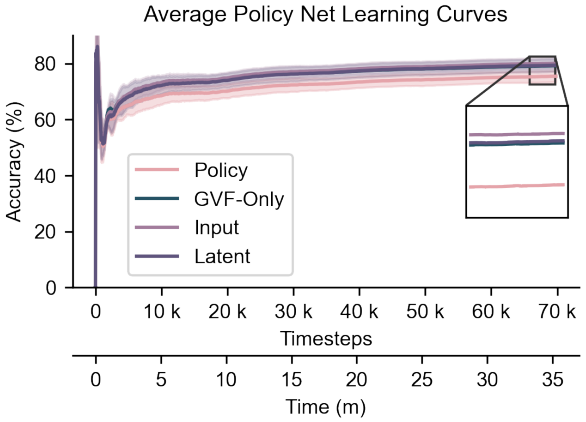

Figure 2: Convergence curves for policy nets.

classified the first terrain, resulting in a sharp initial spike in the accuracy curve at the start of learning. The accuracy decreased as the learning system observed more terrains, and misclassifications increased. As the learning systems accumulated more examples of each terrain, their learning curve accuracy increased and then plateaued at the end of learning.

3.2 Prediction Accuracy

To assess classification accuracy, we evaluated the output of the networks by comparing the assigned terrain labels with ground-truth terrain labels. By the end of learning, the Policy Net achieved a total accuracy of 75.4% ($\pm$ 2.0%), the GVF Only Policy Net achieved 79.1% ($\pm$ 2.5%), the Input GVF Policy Net achieved 80.0% ($\pm$ 2.0%), and the Latent GVF Policy Net achieved 79.3% ($\pm$ 2.2%) across all terrains. All Policy Networks that utilized predictions from GVFs demonstrated significantly higher end-of-learning accuracies compared to the No GVF Policy Net (Policy Net vs. GVF Only: $p < 0.01$, Policy Net vs. Input GVF: $p < 0.001$, Policy Net vs. Latent GVF: $p < 0.001$). However, there was no statistically significant difference in total accuracy between the three GVF Policy Networks (GVF Only vs. Input GVF: $p = 0.08$, Input GVF vs. Latent GVF: $p = 1$, GVF Only vs. Latent GVF: $p = 0.12$).

Figure 3 displays the confusion matrices for terrain classification by each of the policy networks. Table 1 presents the prediction accuracies of all four policy networks separated by terrain. We compared the classification accuracy for each individual terrain across the networks. For predicting upstairs terrain, no significant differences were observed between the Policy Net and the GVF Only Policy Net ($p = 1$), the Input GVF Policy Net ($p = 0.27$), and the Latent GVF Policy Net ($p = 0.18$). Likewise, there were no significant differences observed for downstairs prediction (GVF Only vs. Policy Net: $p = 0.11$, Input GVF vs. Policy Net: $p = 0.55$, Latent GVF vs. Policy Net:

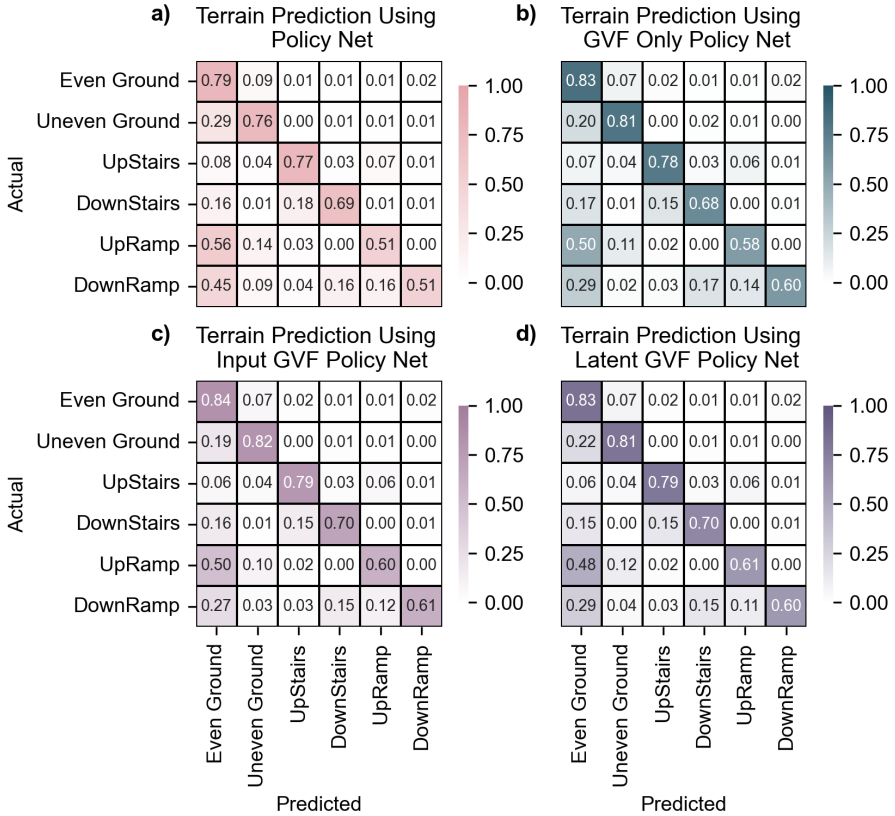

Figure 3: Confusion matrices of the terrain classifications made by the a) Policy Net, b) GVF Only Policy Net, c) Input GVF Policy Net, and d) Latent GVF Policy Net.

Table 1: Policy Net Accuracies Per Terrain

		Networks			
	Terrains	Policy Net	GVF Only Policy Net	Input GVF Policy Net	Latent GVF Policy Net
Accuracies (%)	Even Ground	79.2 (± 2.3)	83.2 (± 2.7)***	83.8 (± 2.4)***	83.0 (± 2.5)***
	Uneven Ground	75.6 (± 3.1)	80.7 (± 3.5)**	81.5 (± 2.7)***	80.4 (± 2.9)***
	Upstairs	76.1 (± 4.9)	76.3 (± 5.6)	77.9 (± 5.0)	78.0 (± 5.0)
	Downstairs	69.1 (± 1.7)	67.5 (± 2.7)	69.5 (± 2.1)	70.0 (± 1.9)
	Upramp	50.1 (± 5.6)	56.7 (± 5.1)*	58.8 (± 6.1)**	59.7 (± 6.2)**
	Downramp	49.7 (± 7.6)	58.2 (± 7.5)**	59.5 (± 7.0)**	58.8 (± 7.1)***

Comparisons made between Policy Net and specified GVF Net. $p \geq 0.05$, *: $p < 0.05$, **:$p < 0.01$, ***: $p < 0.001$

$p = 0.25$). For even ground classification, the GVF Only Policy Net improved accuracy by 4.0% ($p < 0.001$), the Input GVF Policy Net improved accuracy by 8.0% ($p < 0.001$), while the Latent GVF Policy Net improved accuracy by 5.6% ($p < 0.001$) when compared to the Policy Net. For uneven ground classification, the GVF Only Policy Net significantly improved the accuracy by 5.1% ($p = 0.003$), the Input GVF Policy Net improved the accuracy by 5.9% ($p < 0.001$), and the Latent GVF Policy Net showed an accuracy improvement of 4.8% ($p < 0.001$) when compared to the Policy Net. For up-ramp classification, all Policy Nets integrated with GVFs showed improvements in classification accuracy when compared to the Policy Net: 6.6% improvement for the GVF Only Policy Net ($p = 0.015$), 8.7% for the Input GVF Policy Net ($p = 0.001$), and 9.6% for the Latent GVF Policy Net ($p = 0.003$). For down-ramp classification, the GVF Only Policy Net showed an improvement of 8.5% ($p = 0.001$) over the Policy Net, while the Input GVF Policy Net showed an improvement of 9.8% ($p = 0.001$) and the Latent GVF Policy Net showed an improvement of 9.1% ($p < 0.001$). These results indicate that integrating predictive knowledge in the form of GVFs into policy networks can enhance classification performance for certain terrains, particularly those that are often misclassified without predictive information from GVFs. However, it remains unclear which integration method is optimal for classification performance.

4 Conclusion

In this work, we investigated how predictive representations of state from continually learned GVFs impacted the operation of another decision-making network. Specifically, we compared whether it was more effective to provide these predictions directly into the input layer of the network or the latent space of the post-encoding layers. We demonstrated that for the specific case example of terrain prediction during human locomotion, incorporating predictive knowledge improved the classification performance of a deep neural network. However, it remains unclear whether adding these predictions to the input or latent layers is superior, as significant accuracy gains were observed only in the prediction of some terrains (specifically those with confounding factors making it hard to differentiate between output classes). This implies that in this setting, predictive information can likely benefit decision-making under uncertainty, for instance, in identifying terrains prone to misclassification, such as upramps and downramps.

More broadly, our results provide evidence for the usefulness of predictions, as GVFs can provide meaningful predictive abstractions that inform downstream decision-making. Future work will move beyond terrain classification into a fully hierarchical reinforcement learning framework in which predictive representations will drive decision-making for control in a gamified environment.

References

Chitralakshmi K Balasubramanian, David J Clark, and Emily J Fox. Walking adaptability after a stroke and its assessment in clinical settings. *Stroke research and treatment*, 2014(1):591013,

2014.

Romain Baud, Ali Reza Manzoori, Auke Ijspeert, and Mohamed Bouri. Review of control strategies for lower-limb exoskeletons to assist gait. *Journal of NeuroEngineering and Rehabilitation*, 18(1):119, December 2021. ISSN 1743-0003. DOI: 10.1186/s12984-021-00906-3. URL https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-021-00906-3.

Gheorghe Comanici, Amelia Glaese, Anita Gergely, Daniel Toyama, Zafarali Ahmed, Tyler Jackson, Philippe Hamel, and Doina Precup. Learning how to interact with a complex interface using hierarchical reinforcement learning. *arXiv preprint arXiv:2204.10374*, 2022.

Ashley N Dalrymple, David A Roszko, Richard S Sutton, and Vivian K Mushahwar. Pavlovian control of intraspinal microstimulation to produce over-ground walking. *Journal of Neural Engineering*, 17(3):036002, may 2020. DOI: 10.1088/1741-2552/ab8e8e. URL https://dx.doi.org/10.1088/1741-2552/ab8e8e.

Shuzhen Luo, Ghaith Androwis, Sergei Adamovich, Erick Nunez, Hao Su, and Xianlian Zhou. Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning. *Journal of neuroengineering and rehabilitation*, 20(1): 34, 2023.

Patrick M. Pilarski, Travis B. Dick, and Richard S. Sutton. Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints. In *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–8, 2013. DOI: 10.1109/ICORR.2013.6650435.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. *Experience replay for continual learning*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Craig Sherstan, Shibhansh Dohare, James MacGlashan, Johannes Günther, and Patrick M. Pilarski. Gamma-nets: Generalizing value estimation over timescale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5717–5725, Apr. 2020. DOI: 10.1609/aaai.v34i04.6027. URL https://ojs.aaai.org/index.php/AAAI/article/view/6027.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina 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*, AAMAS '11, pp. 761–768, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 0982657161.

Jaden B. Travnik and Patrick 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)*, pp. 1443–1450. IEEE Press, 2017. DOI: 10.1109/ICORR.2017.8009451. URL https://doi.org/10.1109/ICORR.2017.8009451.

Harm Van Seijen, A. Rupam Mahmood, Patrick M. Pilarski, Marlos C. Machado, and Richard S. Sutton. True online temporal-difference learning. *J. Mach. Learn. Res.*, 17(1):5057–5096, January 2016. ISSN 1532-4435.

Adam White. *Developing a preditive approach to knowledge*. PhD thesis, University of Alberta, 2015.