A Value Function Basis for Nexting and Multi-step Prediction

Andrew Jacobsen **Vincent Liu** **Roshan Shariff** **Adam White** **Martha White**

Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8
{ajjacobs, vliu1, roshan.shariff, amw8, whitem}@ualberta.ca

Abstract

Humans and animals continuously make short-term cumulative predictions about their sensory-input stream, an ability referred to by psychologists as *nexting*. This ability has been recreated in a mobile robot using modern reinforcement learning approaches, but in practice there are limitations on how many predictions we can learn. In this paper, we investigate inferring new predictions from a minimal set of learned General Value Functions. We show that linearly weighting such a collection of value function predictions enables us to also make accurate multi-step predictions about future outcomes, and provide a closed-form solution to estimate this linear weighting. We also show that a similar approach can produce accurate estimates of value functions which we did not explicitly train to predict.

Keywords: Reinforcement Learning, Multi-Step Prediction

1 Introduction

The ability to continually make predictions about one's sensory-motor stream is an important aspect of forming awareness of one's environment. In particular, it has been shown that both humans and animals continually make large numbers of short-term cumulative predictions about their sensory input at many different time-scales [Fedus et al., 2019, Pezzulo, 2008, Carlsson et al., 2000, Brogden, 1939]. This ability is referred to as *nexting*. Recent work in Reinforcement Learning has been able to recreate this ability in a mobile robot by using a collection of *General Value Functions* (GVFs)[Sutton et al., 2011], learned online and in parallel [Modayil et al., 2014].

There are are limitations, however, on the number of predictions that an agent can make in a continual learning setting. Each nexting prediction that the agent makes has its own cost in terms of memory and computation. With a large enough collection of nexting predictions — say in the millions — it becomes infeasible for the agent to be able to update them all. Furthermore, in this setting we want our agents to be able to make *new* predictions during run-time. This can be problematic since each GVF may require different hyperparameters (learning rate α, trace decay rate λ, etc.) to learn accurate predictions. Because of this, not only does the agent have to learn each new prediction from scratch, but may have to do so multiple times in order to find the right hyperparameters — all before being able to actually use that prediction.

Instead of learning all possible predictions from scratch, in this paper we investigate whether an agent can use a small set of sufficiently informative nexting predictions to infer the answers to other questions. We show that a small collection of GVF predictions can be used to accurately estimate the answers to predictive questions that the agent has not explicitly learned. We introduce a simple linear transformation which uses a collection of GVF predictions to estimate 1) other GVFs with arbitrary discounting parameters, and 2) n-horizon predictions.

This work has a similar motivation to multi-scale Successor Representations (SRs) [Momennejad and Howard, 2018], and Universal Value Function Approximators (UVFAs)[Schaul et al., 2015]. SRs, in fact, can be represented as GVFs. This work differs from multi-scale SRs, because they assume a tabular setting and use a different weighting scheme with approximate laplace transforms. UVFAs focus on learning value functions that generalize across goal states, using neural networks.

2 General Value Functions

In this section, we define the concepts of return and value and their extension to more general predictions. Consider a Markov Reward Process defined by state-space $\mathcal{S}$, transition function $P : \mathcal{S} \times \mathcal{S} \mapsto [0, 1]$, and reward function $r : \mathcal{S} \mapsto \mathbb{R}$ defined as $r(s) = \mathbb{E}\big[R_{t+1}|S_t = s\big]$, where R_t and S_t are random variables representing the reward and state at time t respectively. We define the return at time t to be

$$G_t \coloneqq \sum_{j=0}^{\infty} \gamma^j r(S_{t+j})$$

where $\gamma \in [0, 1)$ is a constant discounting factor. Given a state $s \in \mathcal{S}$, we define the *value* of state s to be the expected return from state s

$$v(s) \coloneqq \mathbb{E}\big[G_t|S_t = s\big]$$

The function $v(s)$ is referred to as the *Value Function*. A *General Value Function* (GVF) [Sutton et al., 2011] extends the above definition of value, by allowing $r(S_t)$ to be *any* function of the current state — not just a reward signal — and letting the discounting factor be a function of state as well, $\gamma_t \coloneqq \gamma(S_t)$. In this paper we consider only constant discounting factors, thus the above definition of return does not change. GVF predictions with $r(\cdot)$ set to the observations correspond to nexting predictions [Modayil et al., 2014].

3 Predicting Future Outcomes with General Value Functions

In this section, we explain how a set of value functions predictions on this time step can be used to approximate outcomes n-steps into the future. We start by assuming that you have access to the actual returns into the future, and then discuss implications when estimating the returns using value functions.

Suppose we are interested in reconstructing an unknown time series $y_1, \ldots, y_t, \ldots \in \mathbb{R}$. Suppose further that we know the discounted sum of this time series, for several discounts $\gamma_1, \ldots, \gamma_k \in [0, 1)$:

$$G_{t,\gamma_i} = \sum_{j=0}^{\infty} \gamma_i^j y_{t+j+1}.$$

Our goal is to reconstruct various aspects of y given only $G_{t,\gamma_1}, \ldots, G_{t,\gamma_k}$. We are primarily interested in reconstructing y_n for a variety of horizons $n \in \mathbb{N}$. We might also be interested in reconstructing G_γ for some $\gamma \notin \Gamma = \{\gamma_1, \ldots, \gamma_k\}$.

In general, obtaining exact reconstructions is not possible, because we only have k known quantities and yet we want to reconstruct y_n for all $n \in \mathbb{N}$. Our only recourse is to *approximate* y. To do so, define the function $f : \mathbb{N} \to \mathbb{R}$, with $f(t) \coloneqq y_t$. We will try to find a $\hat{f}$ that minimizes the distance to f.

To begin with, we can think of f as an element of an infinite-dimensional vector space — the space of all functions $\mathbb{N} \to \mathbb{R}$. We can define an inner product on this space: $\langle f, g \rangle = \sum_{t=0}^{\infty} f(t)g(t)$, for $f, g \in \mathbb{N} \to \mathbb{R}$, which in turn gives us a norm $\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\sum_{t=0}^{\infty} f(t)^2}$. An element of this function space is the function $t \mapsto \gamma^t$. We can see that the discounted sum is actually an inner product: $G_\gamma = \langle \gamma^t, f \rangle$. In other words, we have a system of k linear equations in the unknown f:

$$\begin{aligned} \langle \gamma_1^t, f \rangle &= G_{t,\gamma_1} \\ &\vdots \\ \langle \gamma_k^t, f \rangle &= G_{t,\gamma_k} \end{aligned}$$

As mentioned above, it is impossible to recover the infinite-dimensional f by solving this system; we can however recover the component of f that lies in the k-dimensional subspace spanned by $\gamma_1^t, \ldots, \gamma_k^t$. Define

$$\hat{f}_\theta(t) \coloneqq \sum_{i=1}^{k} \theta_i \gamma_i^t$$

for $\theta \in \mathbb{R}^k$ as a linear combination of the functions $\{\gamma_1^t, \ldots, \gamma_k^t\}$ with coefficients $\theta_1, \ldots, \theta_k$. We want to find the coefficients θ that minimize the squared distance $\|\hat{f}_\theta - f\|^2$. The solution to this problem is

$$\theta = K^{-1} \begin{bmatrix} \langle \gamma_1^t, f \rangle \\ \vdots \\ \langle \gamma_k^t, f \rangle \end{bmatrix},$$

where K is a $k \times k$ matrix whose entries are given by $K_{ij} = \langle \gamma_i^t, \gamma_j^t \rangle$. Fortunately, the entries of K can actually be computed in closed form when the discounting factors are constant:

$$K_{ij} = \langle \gamma_i^t, \gamma_j^t \rangle = \sum_{t=0}^{\infty} \gamma_i^t \gamma_j^t = \frac{1}{1 - \gamma_i \gamma_j}$$

for $0 \leq \gamma_i, \gamma_j < 1$. We can also use ℓ_2 regularization when estimating θ:

$$\theta_\lambda = (K + \lambda I)^{-1} \begin{bmatrix} \langle \gamma_1^t, f \rangle \\ \vdots \\ \langle \gamma_k^t, f \rangle \end{bmatrix}$$

where λ is the weight of the regularization and I is an identity matrix.

With this approximation to f, we can return to the problem of approximating aspects of the series y. We can obtain n-horizon predictions by using

$$y_n = f(n) \approx \hat{f}_\theta(n) = \sum_{i=1}^{k} \theta_i \gamma_i^n.$$

We can estimate the discounted sum $\langle \gamma^t, f \rangle$ for some $\gamma \notin \Gamma$ using

$$\sum_{i=0}^{\infty} \gamma^t y_{t+1} = \langle \gamma^t, f \rangle \cong \langle \gamma^t, \hat{f}_\theta \rangle = \sum_{i=1}^{k} \theta_i \langle \gamma^t, \gamma_i^t \rangle = \sum_{i=1}^{k} \frac{\theta_i}{1 - \gamma_i \gamma}.$$

When making predictions about the future, we do not have access to exact returns. Rather, we will estimate value functions—estimate expected returns—to use within the above formulas. For exact value functions, we can obtain the same approximations as above for expected n-step values and expected discounted sums. This is appropriate, as any direct multi-step prediction method using squared error is estimating the expected value n steps in the future. The approximation of the value functions themselves will introduce additional approximations to above.

4 Approximation Error

In practice we do not have access to the true returns G_{t,γ_i}, but instead have to estimate them. We want to characterize the error that results from the transition from true returns G_{t,γ_i} to expected returns $\mathbb{E}[G_{t,\gamma_i}]$.

4.1 Idea 1: Characterization in terms of variance

One simple way we can characterize the approximation error is by considering the expected difference between the solution θ obtained when using true returns and when using expected returns. Let G_Γ be the vector of such that $G_\Gamma[i] = G_{t,\gamma_i}$, and V_Γ be the vector of corresponding expected returns. Consider the solutions $\theta \coloneqq K^{-1}G_\Gamma$ and $\hat{\theta} \coloneqq K^{-1}V_\Gamma$. In expectation, we have

$$\begin{aligned}
\mathbb{E}[||\theta - \hat{\theta}||^2] &= \mathbb{E}[||K^{-1}G_\Gamma - K^{-1}V_\Gamma||^2] \\
&= \mathbb{E}[||K^{-1}(G_\Gamma - V_\Gamma)||^2] \\
&= \mathbb{E}[(G_\Gamma - V_\Gamma)^\top (K^{-1})^\top K^{-1}(G_\Gamma - V_\Gamma)]
\end{aligned}$$

Consider a special case where K^{-1} happens to be orthonormal, then

$$\begin{aligned}
\mathbb{E}[||\theta - \hat{\theta}||^2] &= \mathbb{E}[(G_\Gamma - V_\Gamma)^\top (G_\Gamma - V_\Gamma)] \\
&= \mathbb{E}[\sum_{i=1}^{k} (G_{t,\gamma_i} - \mathbb{E}[G_{t,\gamma_i}])^2] \\
&= \sum_{i=1}^{k} \mathbb{E}[(G_{t,\gamma_i} - \mathbb{E}[G_{t,\gamma_i}])^2] \\
&= \sum_{i=1}^{k} Var[G_{t,\gamma_i}]
\end{aligned}$$

Thus in this special case, the approximation error is a sum of variances of each of the G_{t,γ_i}. This implies two things: first, that having more GVFs does not necessarily improve the approximation, in fact we ought to have a "less-is-more" approach; and second, that the approximation will be better when using lower values of γ since these will be lower variance.

In general, K^{-1} is not orthonormal. In this case, we end up with

$$\mathbb{E}[||\theta - \hat{\theta}||^2] = \sum_{i,j=1}^{k} \Lambda_{i,j} Cov(G_{t,\gamma_i}, G_{t,\gamma_j})$$

Where $\Lambda = (K^{-1})^\top K^{-1}$. Notice that, in general, this will be worse than the above approximation error. TODO: do we ever get negative covariance between returns?? This *might* suggest that a useful heuristic for selecting Γ is ensuring that $K^{-1} = (\Gamma\Gamma^\top)^{-1}$ is orthonormal (or at least close to orthonormal). In an episodic setting, one way for K^{-1} to be orthonormal is when K^{-1} is a DFT matrix. This would require using complex-valued γ values with $|\gamma_i| = 1$ for each γ_i. Similarly, in a continuing task we can get an orthonormal K^{-1} by truncating the returns to some number of steps $\tau \in \mathbb{N}$ and using a τ-sample DFT matrix.

Note that the approximation error in the solution θ tells us the prediction errors that result from moving from true returns to expected returns in a straight forward way.

$$\mathbb{E}[||\boldsymbol{\gamma}^\top \theta - \boldsymbol{\gamma}^\top \hat{\theta}||^2] = \sum_{i,j=1}^{k} \Lambda'_{i,j} Cov(G_{t,\gamma_i}, G_{t,\gamma_j})$$

where now $\Lambda' = (K^{-1})^\top \boldsymbol{\gamma}\boldsymbol{\gamma}^\top K^{-1}$ and $\boldsymbol{\gamma}$ is the k-dimensional vector such that $\boldsymbol{\gamma}_i = \gamma_i^n$ for n-horizon prediction and $\boldsymbol{\gamma}_i = \frac{1}{1-\gamma\gamma_i}$ for estimating a GVF prediction with discount parameter γ.

4.2 Idea 2: Decomposition of approximation errors

Suppose our goal is to approximate V^{γ}.

Let $V^{\gamma,n}$ be the value reconstructed from $V^{\gamma_1}, ..., V^{\gamma_n}$ (n true values).

Let $V^{\gamma,k}$ be the value reconstructed from $V^{\gamma_1}, ..., V^{\gamma_k}$ (k true values).

Let $\hat{V}^{\gamma,k}$ be the value reconstructed from $\hat{V}^{\gamma_1}, ..., \hat{V}^{\gamma_k}$ (k estimated value).

The approximate error is bounded by

$$|V^{\gamma} - \hat{V}^{\gamma,k}|^2 \leq |V^{\gamma} - V^{\gamma,n}|^2 + |V^{\gamma,n} - V^{\gamma,k}|^2 + |V^{\gamma,k} - \hat{V}^{\gamma,k}|^2.$$

The first term is likely to go to zero as $n \to \infty$ (TODO).

For the third term, assume the given estimated values are bounded with high probability (TODO: justify this assumption (cite Sajed, Chung, and White, 2018). Intuitively, higher gamma return has higher magnitude, and thus looser bound), i.e.,

$$\Pr(|\hat{V}^{\gamma_i} - V^{\gamma_i}| \leq \frac{\epsilon}{1-\gamma_i},\ \forall i \in [k]) \geq \delta.$$

Following from Andrew's note, let $G_{\Gamma} = [\hat{V}^{\gamma_1}, ..., \hat{V}^{\gamma_k}]^T$, $V_{\Gamma} = [V^{\gamma_1}, ..., V^{\gamma_k}]^T$, $w = [\frac{1}{1-\gamma\gamma_1}, ..., \frac{1}{1-\gamma\gamma_k}]$ and $\boldsymbol{\epsilon} = [\frac{\epsilon}{1-\gamma_1}, ..., \frac{\epsilon}{1-\gamma_k}]^T$, then

$$\begin{aligned}
V^{\gamma,k} - \hat{V}^{\gamma,k} &= w\theta - w\hat{\theta} \\
&= w(\theta - \hat{\theta}) \\
&= w(K^{-1})(G_{\Gamma} - V_{\Gamma}) \\
&\leq w(K^{-1})\boldsymbol{\epsilon} \ \because \text{the assumption we made} \\
&= \sum_{j=1}^{k}\sum_{i=1}^{k} \boldsymbol{\epsilon}_j w_i (K^{-1})_{ij} \\
&\leq \sum_{j=1}^{k}\sum_{i=1}^{k} \frac{\epsilon (K^{-1})_{ij}}{(1-\gamma_i)(1-\gamma_j)} \ \because \boldsymbol{\epsilon}_j w_i = \frac{\epsilon}{(1-\gamma\gamma_i)(1-\gamma_j)} \leq \frac{\epsilon}{(1-\gamma_i)(1-\gamma_j)} \\
&\leq \frac{\epsilon}{(1-\gamma_{max})^2} \sum_{j=1}^{k}\sum_{i=1}^{k} (K^{-1})_{ij}
\end{aligned}$$

The bound suggests that if there are estimation errors in value functions, we should use the inverse Kernel matrix which has smaller sum of values. It justifies why ℓ_2 regularization helps in our experiments (is it true?). This equation might be able to write as a function of $\gamma, \gamma_1, ..., \gamma_k$ (if we have the close-form expression of K^{-1}).

5 Experimental Results

In this section we give two simple demonstrations of an agent's ability to infer the answers to questions it has not been trained to predict. We imagine a scenario in which the agent is performing a simple prediction task for a number of evaluation steps. Mid-way through the task, the agent adds a new prediction to be evaluated.

5.1 Predicting Future Observations

To demonstrate our method's ability to make n-horizon predictions, we tested our approach on the Mackey-Glass time series, a single-variable dataset derived from the time-delay differential equation:

$$\frac{\partial y(t)}{\partial t} = \alpha \frac{y(t-\tau)}{1 + y(t-\tau)^{10}} - \beta y(t)$$

In this experiment, we used $\tau = 17$, $\alpha = 0.2$, and $\beta = 0.1$, starting from an initial value of $y(0) = 1.2$. The agent makes predictions at a horizon of 6 steps for $1,000,000$ steps, and adds a horizon 12 prediction mid-way through. We gave the agent a GVF basis consisting of 100 GVFs and constant discount factors $\gamma_i = 1.0 - i/101$, for $i = 1, \ldots, 100$. The GVFs

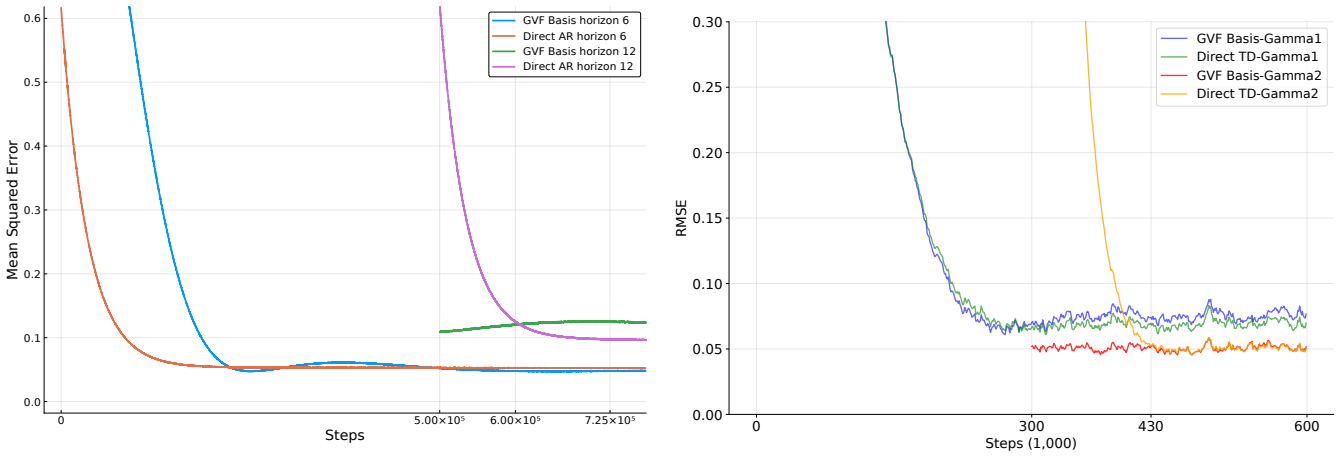

Figure 1: **Left:** Online Mean Squared Error (MSE) for the GVF Basis and Direct AR. For the first 500,000 steps of training, the agent makes only horizon 6 predictions. At step 500,000 the agent adds a horizon 12 prediction to evaluate. Performance stabilizes at around 725,000 steps, so we omit the final 250,000 steps. **Right:** Online Root Mean Squared Error (RMSE) for the GVF Basis and Direct TD. In this figure "Gamma1" is 0.9 and "Gamma2" is 0.8.

were trained using TD(0) with linear value function approximation. The features given to the GVFs was a history of the previous 4 observations. We additionally included predictions made using a linear autoregressive model (*"Direct AR"*), with a history of 4 observations, as an optimal baseline. Note that the baseline was allowed to directly train on each of the horizons of interest, while our method was never explicitly trained to do any n-horizon prediction.

Results on this task are shown in figure 1 (Left). We can see that at the start of training, the GVF basis predictions at horizon 6 take a bit longer to learn, but still ends up reaching the same performance level as the baseline without ever being trained to make this prediction. At the 500,000 step mark, the agent begins making horizon 12 predictions. Note that the baseline agent using direct AR has to wait almost 100,000 steps before it can obtain a reasonably accurate horizon 12 prediction! Furthermore, note that the GVF basis could have just as easily made predictions for an arbitrary number of horizons, all of which can be made immediately.

Our method is slightly less accurate than the final performance of the baseline at horizon 12. However, we note that the discounting values are chosen rather arbitrarily; a better understanding of discount selection strategy could lead to improved performance. Selecting optimal discounting factors is still currently a work in progress at the time of writing. We note also that in these experiments, the GVFs were given a history of observations in order to build up sufficient state information. We could have instead used the GVF predictions themselves as features at each step, as in Schlegel et al. [2018]. We find that this approach works much better in general, but is outside the scope of this paper.

5.2 Predicting other General Value Functions

We also tested the accuracy of our method for predicting other GVFs. The environment is a randomly generated Markov chain with 500 states and the branching factor of 5 (the number of successor states), where we can compute the true value functions exactly. *GVF basis* learns the GVFs of a set 10 discount factors linearly spaced between $[0.0, 0.99]$, Γ_{train}, and predicts the GVF of a different discount factor γ. *Direct TD* method learns the GVF of the discount factor γ directly.

Both methods use TD(0). We tuned the regularization constant over the values $\lambda \in \{0.0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and the learning rate over the set $\{0.4, 0.2, 0.1, 0.05, 0.025\}$. Online performance is shown in Figure 1 (Right). For the first 300,000 steps of training, the agent makes prediction for $\gamma_1 = 0.9$. We can see that the GVF Basis predictions are comparable to Direct TD predictions in the beginning of training. At step 300,000 the agent adds a prediction for $\gamma_2 = 0.8$. The GVF Basis can immediately make the new prediction accurately; the Direct TD method, however, needs to learn the value function from scratch, and takes roughly 430,000 steps to reach the same performance as the GVF Basis estimate.

6 Conclusions

In this work, we introduced a novel approach to infer new GVF predictions and multi-step predictions from a small set of learned GVFs. This work was focused more on whether a collection of GVF predictions *can* be used to make other

predictions, rather than on the general utility of this approach. In these initial experiments, the collection of discounting factors Γ was chosen naively; future work will investigate how these discounting factors can be chosen optimally to facilitate reconstructing multi-step and discounted cumulative predictions. It is possible that better performance can be attained by selecting Γ in a more principled way. We note also that discounted cumulative predictions are interesting in their own right for time series prediction problems such as section 5.1; each of the predictions made with a different discounting factor provides slightly different information about how the signal is expected to evolve over time. We believe that the fact that predictions of this sort also facilitate relatively accurate multi-step predictions could make them a subject of interest to the general time series forecasting community.

References

Wilfred J Brogden. Sensory pre-conditioning. *Journal of Experimental Psychology*, 25(4):323, 1939.

Katrina Carlsson, Predrag Petrovic, Stefan Skare, Karl Magnus Petersson, and Martin Ingvar. Tickling expectations: neural processing in anticipation of a sensory stimulus. *Journal of cognitive neuroscience*, 12(4):691–703, 2000.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

Joseph Modayil, Adam White, and Richard S Sutton. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior*, 22(2):146–160, 2014.

Ida Momennejad and Marc W. Howard. Predicting the future with multi-scale successor representations. *bioRxiv*, page 449470, October 2018. doi: 10.1101/449470. URL `https://www.biorxiv.org/content/10.1101/449470v1`.

Giovanni Pezzulo. Coordinating with the future: the anticipatory nature of representation. *Minds and Machines*, 18(2): 179–225, 2008.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320, 2015.

Matthew Schlegel, Adam White, Andrew Patterson, and Martha White. General value function networks. *arXiv preprint arXiv:1807.06763*, 2018.

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*, pages 761–768, 2011.