Neural Networks 145 (2022) 10–21

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2021 Special Issue on AI and Brain Science: Brain-inspired AI

Generalized attention-weighted reinforcement learning

Lennart Bramlage [a,b,*,1], Aurelio Cortese [b,*,2]

[a] Faculty of Technology, Bielefeld University, 33615, Germany
[b] Computational Neuroscience Labs, ATR Institute International, 619-0288, Japan

ARTICLE INFO

Article history:
Available online 11 October 2021

Keywords:
Self-attention
Decision-making
Value function approximation
Deep reinforcement learning
Representation learning
Feature binding

ABSTRACT

In neuroscience, attention has been shown to bidirectionally interact with reinforcement learning (RL) to reduce the dimensionality of task representations, restricting computations to relevant features. In machine learning, despite their popularity, attention mechanisms have seldom been administered to decision-making problems. Here, we leverage a theoretical model from computational neuroscience – the attention-weighted RL (AWRL), defining how humans identify task-relevant features (i.e., that allow value predictions) – to design an applied deep RL paradigm. We formally demonstrate that the conjunction of the self-attention mechanism, widely employed in machine learning, with value function approximation is a general formulation of the AWRL model. To evaluate our agent, we train it on three Atari tasks at different complexity levels, incorporating both task-relevant and irrelevant features. Because the model uses semantic observations, we can uncover not only which features the agent elects to base decisions on, but also how it chooses to compile more complex, relational features from simpler ones. We first show that performance depends in large part on the ability to compile new compound features, rather than mere focus on individual features. In line with neuroscience predictions, self-attention leads to high resiliency to noise (irrelevant features) compared to other benchmark models. Finally, we highlight the importance and separate contributions of both bottom-up and top-down attention in the learning process. Together, these results demonstrate the broader validity of the AWRL framework in complex task scenarios, and illustrate the benefits of a deeper integration between neuroscience-derived models and RL for decision making in machine learning.

1. Introduction

Reinforcement learning (RL) provides a powerful description of learning from experience in biological organisms (Doya, 2007; Sutton & Barto, 1998). Yet, RL algorithms become notoriously inefficient when the dimensionality of the problem is large (Bellman, 1957), as is generally the case in real-world scenarios. Animals and humans not only learn new tasks and generalize quickly from complex, noisy stimuli and experiences, but they usually adapt to changing conditions with unmatched dexterity. This ability could, in part, arise from abstractions and selective attention generating lower-dimensional task-state representations (Cortese, De Martino, & Kawato, 2019; Gazzaley & Nobre, 2012; Niv, 2019), effectively resolving the "curse of dimensionality" in RL. That is, the brain would represent a task only in terms of behaviorally relevant aspects (features) of the environment, foregoing unnecessary information. From this perspective, attention guides the selection of features on which RL operates, while RL value approximation further refines the attentional focus to currently relevant dimensions (Leong, Radulescu, Daniel, Dewoskin, & Niv, 2017; Niv et al., 2015).

In machine learning research, attention models have been applied to a variety of problems. Visual selective attention has enabled sequential methods for image processing, similar to saccades in the human eye (Ba, Mnih, & Kavukcuoglu, 2014; Xu et al., 2015). Top-down, goal-directed attention may be used to reconfigure general purpose neural networks to accommodate new goals (Luo, Roads, & Love, 2020). Self-attention (Bahdanau, Cho, & Bengio, 2014) has revolutionized natural language processing by all but replacing more complex recurrent models for sequential data (Vaswani et al., 2017). Nevertheless, explicit models of attention have only recently broken into the field of reinforcement learning (RL) for decision-making in artificial neural networks, and almost solely at the sensory (i.e., visual) level (e.g., see Manchin, Abbasnejad, and van den Hengel (2019), Sorokin, Seleznev, Pavlov, Fedorov, and Ignateva (2015), Yuezhang, Zhang, and Ballard (2018)). Yet, research in neuroscience suggests that attention mechanisms in the brain engage with representations all along the cortical hierarchy, selecting

* Corresponding authors.
E-mail addresses: bramlagelennart@gmail.com (L. Bramlage), cortese.aurelio@gmail.com (A. Cortese).
[1] Secondary corresponding author.
[2] Primary corresponding author.

https://doi.org/10.1016/j.neunet.2021.09.023

a variety of features, stemming not only from sensory measurements, but also from memory, prior knowledge and predictive forward-modeling (Chun, Golomb, & Turk-Browne, 2011; Farashahi, Rowe, Aslami, Lee, & Soltani, 2017; Mack, Love, & Preston, 2016; Martinez-Trujillo & Treue, 2004).

We take direct inspiration from neuroscience by translating a theoretical model of feature-based attentional learning, Attention-Weighted Reinforcement Learning (AWRL) (Leong et al., 2017; Niv, 2019; Niv et al., 2015), into a functional neural network model based on state-of-the-art machine learning methods. The proposed model uses the Multi-head Dot-Product Attention mechanism (i.e., self-attention), originally designed for different classes of problems in machine learning such as natural language and sequence processing (Bahdanau et al., 2014; Vaswani et al., 2017). We first derive a novel formulation of self-attention in feature space, for non-stationary RL paradigms. Self-attention introduces the exciting possibility of not only selecting task-relevant features but relating them to create compound representations of current observations. Thus, we formalize this approach by proving that the conjunction of self-attention with value approximation learning is a generalization of AWRL. It is important to note that, as opposed to the more common approach in deep RL models where feature extraction is usually learned in tandem with optimizing a value function, here we assume an arbitrary feature extraction process and a subsequent selection step performed by attention that is the focus of our work. This is motivated by empirical work in neuroscience showing that neurons in the first layers of sensory processing represent specific features and combinations thereof (Grunewald & Skoumbourdis, 2004; Jörntell et al., 2014), upon which attention can operate a selection mechanism (Ekman, Roelfsema, & de Lange, 2020), at various levels of abstraction (Gong & Liu, 2020) that is coupled with reinforcement learning (Leong et al., 2017; Niv et al., 2015). The viability, benefits, and shortcomings of this model are evaluated on canonical deep-RL benchmark tasks from the Atari framework (Bellemare, Naddaf, Veness, & Bowling, 2015).

In a series of three experiments, we seek to answer the following questions: **(i)** does the simple AWRL model implemented in deep-RL perform as theorized in the neuroscience literature, and can we improve performance with the generalized AWRL approach? **(ii)** Does the better performer of the two methods compare with established feature-based models in the same learning regime? **(iii)** What are the effects of conditioning attention on feature-channels or feature-content alone (corresponding to naïve interpretations of top-down and bottom-up attention, respectively)?

2. Attention-weighted RL

Attention mechanisms are prime candidates for the process of breaking down high-dimensional feature observations into lower-dimensional representations (Corbetta & Shulman, 2002). These proposed task-state representations (Niv, 2019) have several properties to support efficient learning: (i) they are implemented by a set of feature-specific weights, (ii) they are task-specific (Bar-Gad, Morris, & Bergman, 2003), and (iii) dynamically adjustable as the agent shifts task-focus (Frank & Badre, 2012). Physiologically, the inductive bias of perceiving the world as a collection of features is highly plausible. The visual system prominently implements a series of filters for orientation, shape, and color, whose compounds represent the scenes in front of our eyes (Hubel & Wiesel, 1962). These low-level features are extracted at the earliest stage of the visual pathway (V1), which is affected by learning and attention (Ekman et al., 2020; Henschke et al., 2020; Posner & Gilbert, 1999; Somers, Dale, Seiffert, & Tootell, 1999), supporting the idea that feature-based RL may be central in biological intelligence.

Previous work by Leong et al. has extended the general feature-based RL to incorporate an explicit attentional focus (Leong et al., 2017). This formulation exhibits a feature-specific attention weight $\phi_t(f)$ that leads to the following RL components:

Value function

$$V_t(S_c) = \sum_f \phi_t(f) v_t(f, S_c) \tag{1}$$

Reward prediction error

$$\delta_t = r_t - V_t(S_c) \tag{2}$$

Learning rule

$$v_{t+1}(f, S_c) \leftarrow v_t(f, S_c) + \alpha\,[\phi_t(f)\delta_t] \tag{3}$$

where $v_t(f, S_c)$ represents the value estimate of feature f under the currently examined stimulus S_c. The f variable represents a feature index. This corresponds to an addressing mechanism, such as spatial attention in abstract space. With this definition, the framework aligns well with the feature-similarity gain model (Martinez-Trujillo & Treue, 2004; Treue & Martinez Trujillo, 1999), as it implements a modulation of neural gain in the value function and learning rule.

Eqs. (1)–(3) highlight the bi-directional interaction between attention and RL (Leong et al., 2017). Eq. (1) describes how attention constrains value prediction to selected stimuli, thus influencing the corresponding reward prediction error by possibly increasing the prediction's magnitude. Larger prediction errors lead to more extensive updates to the feature-specific value estimates, again weighted by their respective attention weights. Through this process, attention modulates what the agent learns about by guiding decisions. Conversely, learning about attended features can eventually outweigh the influence of the attention bias if a given feature's value estimate is large enough. The gradual shift of attention may be implemented with an RL mechanism that dynamically updates learned feature weights for value estimation, based on reward prediction errors (as opposed to, e.g., basing attention on recent reward history) (Jones & Cañas, 2010; Kruschke, 1992; Leong et al., 2017).

3. The self-attention mechanism

The popular *Dot-Product Attention* (DPA) mechanism serves as a good candidate for a differentiable weighting function. DPA was initially developed for applications in natural language processing in the "transformer" architecture, where it replaced more intricate recurrent models (Bahdanau et al., 2014; Vaswani et al., 2017). The general processing is simple and geared towards sequential data (see inset in Fig. 1A for a graphical illustration of the mechanism). DPA creates three vector representations $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e} \in \mathbb{R}^{1\times d_k}$, termed query, key, and embedding respectively[3], for each discrete token in a sequence $\mathbf{S} \in \mathbb{R}^{L\times d_{in}}$ of length L by applying three parameterized linear mappings $q(\cdot), k(\cdot), e(\cdot)$. Note that the complete sequence may also be represented in matrix-form, such that $q, k, e : \mathbf{S} \mapsto \mathbf{Q}, \mathbf{K}, \mathbf{E} \in \mathbb{R}^{L\times d_k}$. Next, these matrices are used to redefine each individual input token as a weighted sum of embedding vectors. This step introduces the possibility of selecting and deselecting input tokens for further processing and capturing the relationship between every available pairing of input tokens by summing up their embedding vectors. Query

[3] In the original publication, the "embedding" is called "value" as a reference to database terminology. Here we use the term embedding to avoid confusion with value definitions in RL.

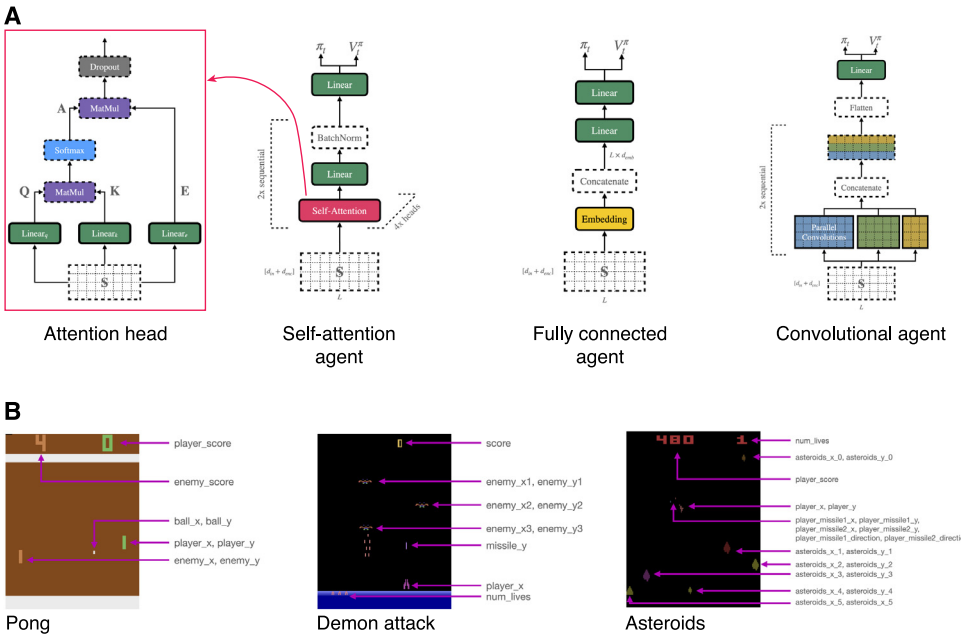

Fig. 1. (A), graph representations of the main self-attention agent, as well as two baselines, the fully connected and convolutional agents. Additionally, in the red inset, the graph representation of a single self-attention head. The self-attention model utilized 2 layers, each with 4 heads. The previous layer output (or stimulus input), represented by a sequence of vectors $\mathbf{S}$ is translated into three disparate matrices $\mathbf{Q}, \mathbf{K}, \mathbf{E}$. The pairwise dot-products of sequence elements $\mathbf{Q}_{i,:}$ and $\mathbf{K}_{j,:}$ correspond to the attention weight a_{ij}. The attention vectors $\mathbf{A}_{i,:}$ are input into a softmax and subsequently multiplied with the sequence of embeddings to create one weighted sum of $\mathbf{E}$ per sequence element. (B), screenshots and list of features for each of the three Atari games (*Pong*, *DemonAttack*, *Asteroids*) considered to test the self-attention model.

and key representations are used to generate the attention map $\mathbf{A} \in \mathbb{R}^{L \times L}$:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \quad \text{s.t. } a_{ij} \approx \langle q(\mathbf{S}_{i,:}), k(\mathbf{S}_{j,:})\rangle \tag{4}$$

Once the attention weights are computed, they are used to generate an attention-weighted sum of embedded inputs $e(\mathbf{S}_{i,:})$ per input sequence element to form the output of the self-attention operation:

$$\hat{\mathbf{S}} = \mathbf{A}\mathbf{E} \quad \text{s.t. } \hat{\mathbf{S}}_{i,:} := \sum_{j=1}^{L} a_{ij}\mathbf{E}_{j,:} \tag{5}$$

These processing steps are straightforward and effective at capturing relationships between sequence tokens, but remain agnostic towards order, the defining factor of a sequence. The standard solution to this problem is the introduction of a positional encoding, which is generally a static vectorization of the token index in the sequence (e.g., a sine–cosine encoding (Vaswani et al., 2017)). Each of the positional encoding vectors is either added or appended to their respective token before all other processing steps, such that $\mathbf{S}^p \in \mathbb{R}^{L \times [d_{in} + d_{enc}]}$. These annotations serve as feature identifiers in this work.

To capture different (e.g., hierarchical) relationships between sequence elements, multiple parallel processing steps are necessary. As such, multiple attention heads in a single DPA layer can be combined to form a *Multi-head Dot-Product Attention* (MHA — the central contribution of the original transformer architecture (Bahdanau et al., 2014; Vaswani et al., 2017)). In the MHA layer, instead of maintaining a single set of parameters for the linear transformations $q(\cdot), k(\cdot), e(\cdot)$, each layer holds h sets of weights per function, each of which processes the input sequence in exclusivity. The outputs are concatenated along the feature axis, such that $\hat{\mathbf{S}}_{concat} \in \mathbb{R}^{L \times [d_k \times h]}$. A fully connected layer then processes each sequence element as such $f : \mathbb{R}^{1 \times [d_k \times h]} \mapsto \mathbb{R}^{1 \times d_{out}}$.

4. Deep and generalized AWRL

Now that we have defined both the theoretical foundation and a candidate mechanism for our attentive deep-RL agent, we demonstrate that the MHA architecture, in fact, implements a generalized version of the computational AWRL model when coupled with value function learning.

In deep-RL, one rarely computes feature-based value function estimates as proposed in the AWRL framework (Eq. (1)), largely because input stimuli (such as images) are not easily separable into distinct features. However, under the assumption that distinct features (in matrix form as above) are used for value estimation, we can represent the feature-based value estimate with a subset of neural network weights from a fully connected linear layer, such that:

$$v(i, \mathbf{S}) = W_t^i \mathbf{S}_{i,:}^\top \quad \text{and } V_t(\mathbf{S}) = W_t \mathbf{S}^\top = \sum_{i=1}^{L} v(i, \mathbf{S}) \tag{6}$$

where W_t^i is the weight matrix solely associated with the ith input feature. Now, we choose to represent the attention function $\phi(i)$, which maps the feature index to a scalar attention weight, with the formulation of self-attention:

$$\phi_t(i) \equiv q_t(i) k_t(i)^\top \tag{7}$$

The value function estimate of the entire attention-weighted stimulus becomes (deep AWRL):

$$V_t(\mathbf{S}) = \sum_{i=1}^{L} \langle q_t(i), k_t(i)\rangle W_t^i \mathbf{S}_{i,:}^\top \tag{8}$$

This equation is a special case of Eq. (5), where only the diagonal elements of the attention matrix may take nonzero values. Note that for this to have an effect, instead of applying a softmax activation as in Eq. (4) one must use the similar inner product of

query and key representations. Alternatively, the softmax may be applied across the diagonal of the attention map, instead of each row, to limit the magnitude of attention weights (Appendix A).

Next, we can expand the model by exploring the full potential of the self-attention mechanism. Instead of selecting and weighing individual features in exclusivity, the self-attention mechanism introduces a relational inductive bias that allows features' mixing. To explain why this should be important, imagine the following scenario: when we are crossing a road, the speed of a car may be only relevant if it is coming towards us, but not when it is moving away. Thus, not only the individual feature (speed), but its relation to other features (e.g., direction) provides necessary information for decision-making. Hence, we reformulate AWRL as follows:

$$V_t(\mathbf{S}) = \underbrace{\sum_{i=1}^{L} q_t(i)k_t(i)^\top W_t^i e_t(\mathbf{S}_{i,:})^\top}_{\text{deep AWRL}}$$
$$\Longrightarrow \underbrace{\sum_{i=1}^{L} W_t^i \left(\sum_{j=1}^{L} q_t(i)k_t(j)^\top e_t(\mathbf{S}_{j,:}) \right)}_{\text{generalized AWRL}} \tag{9}$$

In this formulation, mixtures, while possible, are not mandatory. Depending on the requirements of the environment, optimizing this value function estimate will select and/or relate features as necessary and is therefore a general expression of AWRL. Notably, here we choose to map indices to attention weights, while in the introductory paragraph on self-attention, we chose the actual information content as arguments for the $q(\cdot)$ and $k(\cdot)$ functions. Both approaches have their advantages, but the former is a more accurate translation of the original AWRL model. Most applications of the self-attention mechanism apply a mixture of the two principles, in that the information content of a sequence element is extended with a positional encoding (Gehring, Auli, Grangier, Yarats, & Dauphin, 2017).

5. Attention models and baseline architectures

This section will introduce the attention model architecture as well as three state-of-the-art baseline models used as comparisons.

The proposed deep AWRL (d-AWRL) and generalized AWRL (g-AWRL) models comprise two sequential attention modules (Fig. 1A left). Each module consists of an MHA layer compiling four parallel heads, a linear up-scaling layer applied separately to each feature vector, and a batch-normalization layer. The two attention modules are succeeded by a final, fully connected layer that feeds into policy- and value-outputs in the case of the g-AWRL. See Fig. S1 for a comparison with 2 and 16 heads, indicating 4 heads are optimal. The d-AWRL model, true to the original formulation in Leong et al. (2017), forgoes this additional fully connected layer, such that value truly is computed on a feature-by-feature basis. By feeding the feature sequence directly into the value-estimation layer represented by a fully connected layer, the value estimates are calculated in exclusivity and subsequently summed up (see Eq. (6)). To ensure that no feature mixing is applied in each of the self-attention layers, a mask is applied to the attention maps, such that only values along the main diagonal are considered for feature representations. Notably, the vast majority of trainable parameters are shared between policy π_θ and value-function estimate V_ϕ, except for their final output step. This approach is relatively common in actor–critic models (Mnih et al., 2016; Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), and constitutes the basis of all proposed model and baselines.

The first baseline (FC) is a fully connected neural network (Fig. 1A center), i.e., every vector element in the input volume is connected to each neuron in the first network layer, and so on. To make the model more competitive, we employ an initial feature-wise embedding layer that allows the network to mutate each feature-vector before processing the input volume in its entirety. Note that this step adds a notion of location invariance to this model that would otherwise be absent. The network has two sequential, fully connected layers after the embedding layer. The number of neurons per layer is chosen such that the number of trainable parameters is similar to the attention model. However, it follows logically that the number of trainable parameters must increase quadratically as a function of the number of features.

The second baseline (CNN) is a convolutional neural network of the style commonly applied to natural language tasks (Fig. 1A right). Multiple parallel layers comprising filters of different sizes process the input volume (here, 3, 5, and 7), i.e., a matrix representing L features of dimensionality d_{in}. The feature maps created by each of these layers are then concatenated with each other along the channels dimension. In simple terms, this allows the convolutional layer to compute multi-step relationships between up to seven contiguous features at a time. The fully processed output at a single sequence index then includes multiple relationships along the channel-axis from a maximum distance of one to three neighbors. A subsequent layer will then be able to relate between three and seven one- to six-neighbor relationships at a time. Unlike the fully connected baseline, the basic convolutional model is limited in its capabilities to relate spatially distant vectors to each other by filter- and step-size. Specifically, this means that the same filter would never process the first and last feature in the observation simultaneously without further hierarchical processing.

Therefore, we include an advanced convolutional model (a-CNN) as the third baseline. While the base-case consists of a simple two-layer convolutional network with enough filters to match the attention models' number of parameters, the a-CNN is equipped with up to 20 sequential layers, depending on the number of features L in the observation space, such that it can cover the full length of the input volume with a single filter. Note that this model has a much higher number of parameters compared with all other baselines.

Finally, all models are trained using an updated version of the proximal policy optimization (PPO) algorithm that implements various specifications to facilitate stability and performance (Engstrom et al., 2020) (Appendix B). See Appendix C for hyperparameter details of all models.

6. Testing environment and selected games

For all experiments, we test the main model and the baselines on a selection of three Atari games from the Arcade learning environment (Bellemare et al., 2015). The choice of games is based on their respective observation space and task complexity, with particular attention given to the relevance of each feature in the input. For example, the game *Breakout* does provide a large observation space (35 individual features per time-step, stacked along four time-steps), but most features refer only to the location of one of the colored bars at the top. However, any agent can achieve a perfect *Breakout* score by merely keeping the ball from falling below the paddle, i.e., paying attention to ball and paddle positions. Most features are thus irrelevant, meaning that both the observation space and task are of low complexity. Hence, we choose the following 3 games: *Pong* (low observation and task complexity), *DemonAttack* (medium observation and task complexity), and *Asteroids* (high observation and very high task complexity) — see Appendix D for a description of the learning

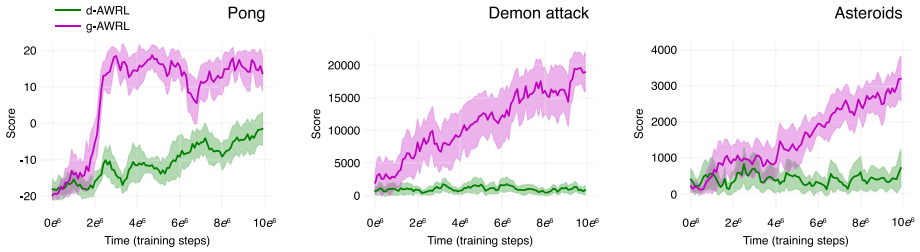

Fig. 2. Comparing performance scores of the original d-AWRL (green lines) and the generalized version g-AWRL (pink lines) on the three Atari games *Pong*, *DemonAttack*, *Asteroids*. The agents completed 10 training runs for each game (parameters were reinitialized each time). The thick line represents the mean, the shaded areas the S.E.M.

Table 1
Number of features for each game, under low and high noise conditions.

Task	*L* features	2*L*	4*L*
Pong	8	16	32
DemonAttack	10	20	40
Asteroids	41	82	164

environment, feature observations, and of each game with details on the available features. The third game is so demanding that modern convolutional approaches have difficulty matching human-level performance and thus makes for an excellent upper bound on the capabilities of our proposed model.

7. Experiment 1: deep vs generalized AWRL

The first set of experiments aims to investigate the potential benefits of computing the value function of compound features, as opposed to individual features (i.e., comparing the d-AWRL model to the g-AWRL approach).

7.1. Task description

The agents are tested on 3 different games: *Pong*, *DemonAttack*, and *Asteroids*, where most features are task-relevant. The number of features in observation space is charted in the second column (L features) in Table 1. The agents complete 10 training runs for each game, with each training run allowing $10e^6$ time-steps interactions with the environment. Parameters are newly initialized in each game and training run.

7.2. Results

Results from ten individual runs per game demonstrate a clear benefit of relating features before computing their respective value contribution. The learning curves clearly show the g-AWRL model reaches high performances in all 3 games, at various levels of difficulty. Conversely, the d-AWRL model fails to reach any competitive performance in all three games, only achieving minimal improvement on the *Pong* task. This model simply sums individual features' value-predictions. The same is true for its policy network, which effectively sums several logits, each based on a singular feature to arrive at a policy distribution. As such, the model is limited to capturing linear relationships between features, in addition to only one relational computation step as opposed to multiple hierarchical ones in the g-AWRL model. The performance on the game *Pong* (Fig. 2 left) may be explained by its simple mechanics and generally low observation complexity since the game can be played with some success (winning a majority of matches) by a linear function, minimizing the vertical distance between ball and player.

However, the other two games demand a higher fidelity of compound feature representations. In *DemonAttack*, (Fig. 2 center) the player needs to line up a spaceship, fire a shot, and observe whether the shot finds its target, all the while avoiding enemy fire. Additionally, to confirm that a shot will hit its target, *DemonAttack* requires two-dimensional evaluations of position. Predictably, the d-AWRL falls short of finding the necessary representations to derive accurate value-function estimates and policies.

The same challenges are exacerbated in *Asteroids*, (Fig. 2 right) where the number of objects to be tracked increases significantly, and each of the objects can move in sixteen directions instead of two as in *DemonAttack*.

The associated attention maps' patterns illustrate that both models converge on highlighting the most relevant task features (Fig. 3, the example maps are from the game *Pong*). While the d-AWRL agent is limited to selection only (hence the one-dimensional maps), it still identifies most of the features that explain the variance in the environment's value-function at any given time (Fig. 3A). The g-AWRL model (Fig. 3B), on the other hand, has the ability to either select or mix learned feature representations to find value-predictive task-states by creating a weighted sum of feature embedding vectors (see Fig. 4 for a graphical explanation). Thus, the model demonstrates that some tasks demand feature compilation to translate into lower-dimensional task-state representations through emergent behavior that follows the value-function optimization process. Indeed, by examining the attention maps of the g-AWRL, we find that feature compilation does occur. Two particular mixing patterns can be found in all game settings: (i) mixing the same feature across several time-steps (suggesting changes in position, score, or other dynamic values) (e.g., Fig. 3B-1 for *Pong*, where 'ball x' is integrated over two time-steps) and (ii) modal mixing of features in a single time-step (that is, mixing features of the same modality like the x-coordinates of two separate objects) (e.g., Fig. 3B-2 for *Pong*, where 'ball x' is mixed with 'player x'). The first style of compound features is ubiquitous in all attention-layers and heads. More so than relating different aspects of the observation, this behavior can be interpreted as time integration to formulate new low-level features. Such a composition captures information about speed and direction, which is then used in subsequent layers. Beyond the ability to track multiple features, this will allow the agent to relate arbitrary features inter-categorically. NLP applications of the self-attention mechanism illustrate this capacity, e.g. by mixing representations of surrounding tokens to identify a noun as the subject of a sentence. Similarly, our agent will compose a selection of feature vectors (what we deem a partial state representation) to feed to the actor and critic networks for policy generation and value prediction. The second mixing pattern occurs equally frequently, but it rarely exhibits the same amount of selective magnitude, which means that those features selected for compilation are rarely selected as sparsely

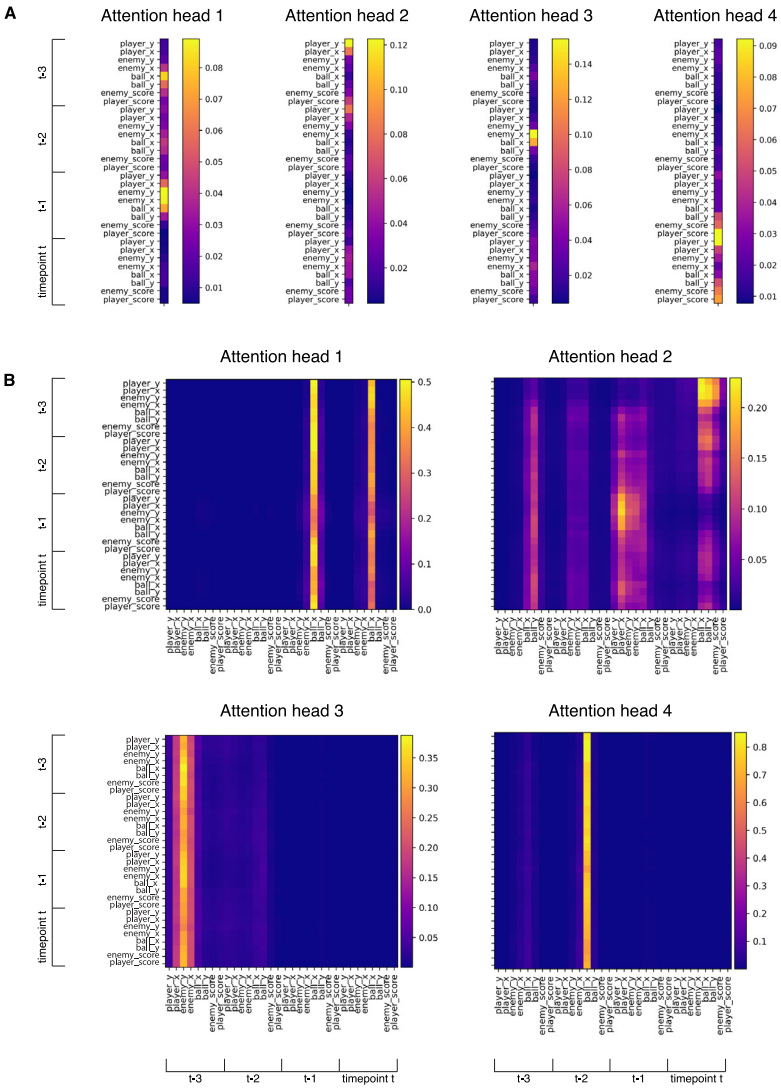

Fig. 3. Example attention maps from two representative agents, at the end of training on the game *Pong*. (A) Attention maps of the 4 attention heads in the first layer of a trained d-AWRL agent. Note the number of features is x4 the original eight, as four time-steps were always included as the input (t, t-1, t-2, t-3). The maps are 1-D vectors since the d-AWRL agent can only select features, but not mix them (no compositions). (B) Attention maps of the 4 attention heads in the first layer of a trained g-AWRL agent. As in (A), the number of features is x4 the original eight ones, as four time-steps were always included as the input (t, t-1, t-2, t-3). The maps are 2-D matrices since the g-AWRL agent can select and mix features, giving rise to compositions. The color bars represent the importance weight assigned to each feature or their combination in a given map.

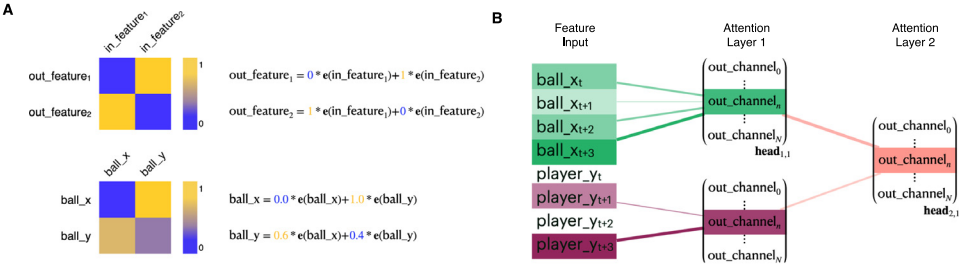

Fig. 4. (A) Illustration of a minimal attention map for a general and a concrete case in the game *Pong*. Row indices indicate the query, column indices the key feature, the dot product of which equals the attention value in that cell. Note that sequence, unlike in NLP applications, is irrelevant in our case. It is therefore not important that the row index feature is represented, at least in part, by itself. Rather, the row index may as well be arbitrary and serves only as a memory address. (B) Graph representation of compound features in successive self-attention layers and different attention heads in the g-AWRL agent. Each attention head has a number of output channels equal to the number of input features N. Several elements of the true architecture are omitted for clarity, such as the constant number of four attention heads per layer, the full number of input features and the multi-channel output of attention heads.

(with as large of an attention weight) as in the first mixing pattern. This behavior may suggest that, while feature compilation is necessary to achieve peak performances, selection is the far more common mode attention operates on for the generation of task-state representations. The expressiveness of differentiable feature representations even allows for amodal mixing, such as scores

with positions, which we observe less frequently (e.g., in Fig. 3B-2 top right). We assume further that conjunctive learning in the Atari context has many similarities to compositions of discrete features, with the added benefit of partial matches. I.e., where discrete feature space experiments will not yield partial reward if one of the target dimensions does not match the target feature, in Atari games and other dynamic environments smaller variations within the target dimensions might still correlate with future reward or specific policies. In other words, slight deviations in the target feature expressions should have minimum impact on the output distributions of actor and critic networks.

Lastly, we find that the g-AWRL agents tend to discard a large number of input features by choosing to represent them as the same feature multiple times. For example, the attention head in Fig. 3 B-4 for *Pong* elects to represent every input feature as the 'ball x'-position almost exclusively.

While we have discussed here the attention map from the g-AWRL agent trained on the game *Pong*, the maps obtained after training on the games *DemonAttack* and *Asteroids* display the same mixing patterns (see Figs. S2 and S3).

8. Experiment 2: generalized AWRL vs baselines

This second round of experiments examines whether the application of attention mechanisms harbors algorithmic benefits for a deep-RL agent, particularly in conditions that are closer to real-world cases (e.g., with high-dimensional, noisy input observations). Here, the g-AWRL agent competes against the FC, CNN, and a-CNN baselines.

8.1. Task description

As in Experiment 1, we use the 3 Atari games *Pong*, *DemonAttack*, and *Asteroids*. To test the resilience of the agents to high dimensionality and noise, we introduce distractor environments, which are parallel instances of each respective game whose features are appended to the observation input vector. The agent interacts with only one of these environments and has to identify the correct set of corresponding features. It is important to emphasize that while the distractor features are drawn from the same environment, they are independent instances of the game played by a decoupled, pre-trained agent. As such, they do not contribute any information to the observation, but they keep the distribution over feature values very similar (and make the attention task difficult). Thus, we extend the task-set by a low-noise and a high-noise version of each of the environments. The low-noise condition adds only a single distractor environment, where the number of features in each observation is doubled. In the high-noise version, a total of three distractor environments are added — quadrupling the size of the observation per time-step. The number of features in observation space is charted in Table 1, center and right columns. Additionally, we run a supplementary analysis using 7 distractor environments, and an analysis using random distractor features, to test if the ability of the g-AWRL model to select relevant features even among very high noise, or an equivalent level of uncorrelated noise, is preserved. Random features are sampled from different periodic functions to prevent large changes in the same feature within single frame updates. At the outset of each training run, the sequence of features in the observation window is randomized to minimize correlations between spatial and associative relationships. The agents complete 10 training runs for each game and conditions, with each training run allowing $10e^6$ time-steps interactions with the environment. Parameters are newly initialized in each game, condition and training run.

8.1.1. Results

The learning curves illustrate the strong advantage of the g-AWRL attention agent (Fig. 5). However, the a-CNN agent is a close match in terms of final performance and convergence speed in more complex tasks and high-noise cases (e.g., *DemonAttack*, *Asteroids*). The basic CNN model's top-performances in low-noise and low-complexity environments instead lend support to the notion that some form of spatial invariance significantly contributes to performance in feature space. The idea is intuitive, given that the fully connected baseline model maintains a large number of weights exclusively processing task-irrelevant features, presenting an ineffective architecture for the data.

Except for the *Asteroids* task, the basic CNN agent demonstrates stable improvements in low-noise environments and performance curves second only to the g-AWRL agent, likely due to its computation graph's low complexity. Another compelling explanation is the proximity of task-relevant features within the window of observation. With low dimensionality in observation space, each convolutional filter will create a mixed representation of almost the entire width of the observation (along the feature index axis). Effectively, each filter acts as one attention head, albeit with simplified computation graph and output sequence, and a lower number of parameters to train. As the number of features in the observation window increases, these benefits rapidly disappear since single convolutional filters will be less likely to capture multiple task-relevant and relatable features in a single operation. This issue does not affect the g-AWRL agent, which will consider all combinations of features locally and non-locally as training commences, depending on the query and key mappings' initialization. CNNs are theoretically able to implement such non-local processing as well, depending on the size of the applied kernels. With smaller kernels, several hierarchical layers are necessary to span the whole input sequence (Cordonnier, Loukas, & Jaggi, 2019).

Importantly, the g-AWRL is resilient to noise even when using very high numbers of irrelevant features (i.e., 7 distractor environments, game *Pong*) or random features (Fig. S4). Including random noise as irrelevant features (instead of distractor environments), has less of a detrimental effect on performance (Fig. S4). This is likely due to the differing distribution across time in the random features as opposed to real features drawn from a distractor environment. Analyzing the attention maps demonstrates that the agents discriminate the relevant features from the irrelevant copies (Fig. S5). Finally, we also test the agents in the basic environments without any noise (Fig. S6), and find that – all environments and noise levels considered – the g-AWRL consistently reaches the best performances.

9. Experiment 3: exclusive spatial- vs content-based attention

With the third experiment we investigate the differences in attention patterns between conditioning attention exclusively on feature content or spatial addresses. Spatial addresses here refer to the positional encoding that serves to index every feature vector. While features are randomly sorted at every new initialization of a learning environment, their position in the observation space does not change within a training run.

To solve a game, the agent will attempt to approximate the environment's value-function. In the spatial attention case, the agent is forced to select those feature channels that may hold task-relevant information using address-based attention. However, the observed information within the selected channels may not be relevant at any given moment. This process corresponds to a crude interpretation of feature search. Intelligent agents seek out features using top-down attention mechanisms that modulate the response of those neurons tuned to that feature

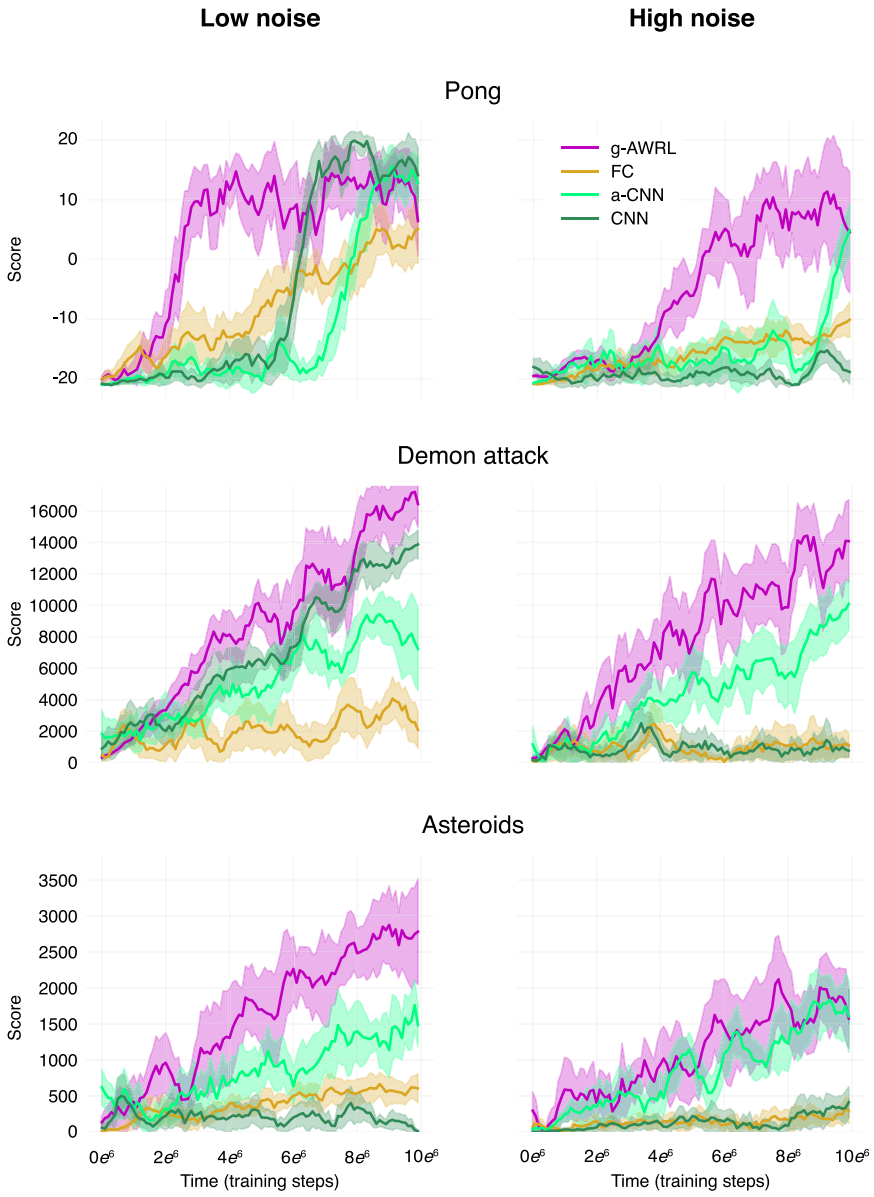

Fig. 5. Comparison of the g-AWRL vs baselines in low (2 × L features: L relevant, L irrelevant) and high noise conditions (4 × L features: L relevant, 3 × L irrelevant), in the three Atari games *Pong*, *DemonAttack*, *Asteroids*. The agents complete 10 training runs for each game and condition (parameters are reinitialized each time). The thick lines represent the mean, the shaded areas the S.E.M.

(Treue & Martinez Trujillo, 1999). In the content-based case, on the other hand, attention is conditioned exclusively on features that are currently available in the observation. This type of attention corresponds to stimulus-driven, i.e., bottom-up processes, where a salient observation will capture the attentional spotlight. Bottom-up attention is a necessary mechanism in biological intelligence that allows agents to react to essential stimuli on the fly, without engaging in a specific task (e.g., responding to the presence of a predator or an unexpected food-source).

9.1. Task description

Here, both attention models are built upon the g-AWRL architecture. The observations received from the environment are static for the computation of attention maps in the address-based case (spatial) and comprised of feature values in the content-based case (content). To clarify, the address-based agent will still receive the contents of selected features for later processing steps, and merely the selection process is limited to predefined positional encodings. For this experiment, the environments are fed as such to the agents, without noise. The agents complete 10 training runs, with each training run allowing $10e^6$ time-steps interactions with the environment. Parameters are newly initialized in each training run.

9.2. Results

In terms of performance, we find that both versions of the g-AWRL achieve, on average, lower scores compared with the main g-AWRL (Fig. 6A vs Fig. S6), highlighting the importance of integrating both attention streams for an agent to solve complex tasks. Overall, the purely address-based attention agent tends to achieve scores close to the zero-noise benchmarks of g-AWRL, while the purely content-based agent displays a severe reduction in top-scores and convergence speed. One particular issue of purely content-based attention, in this case, is that the agent has no broader understanding of the observations it makes. Without a sense of which feature channel is being attended, the

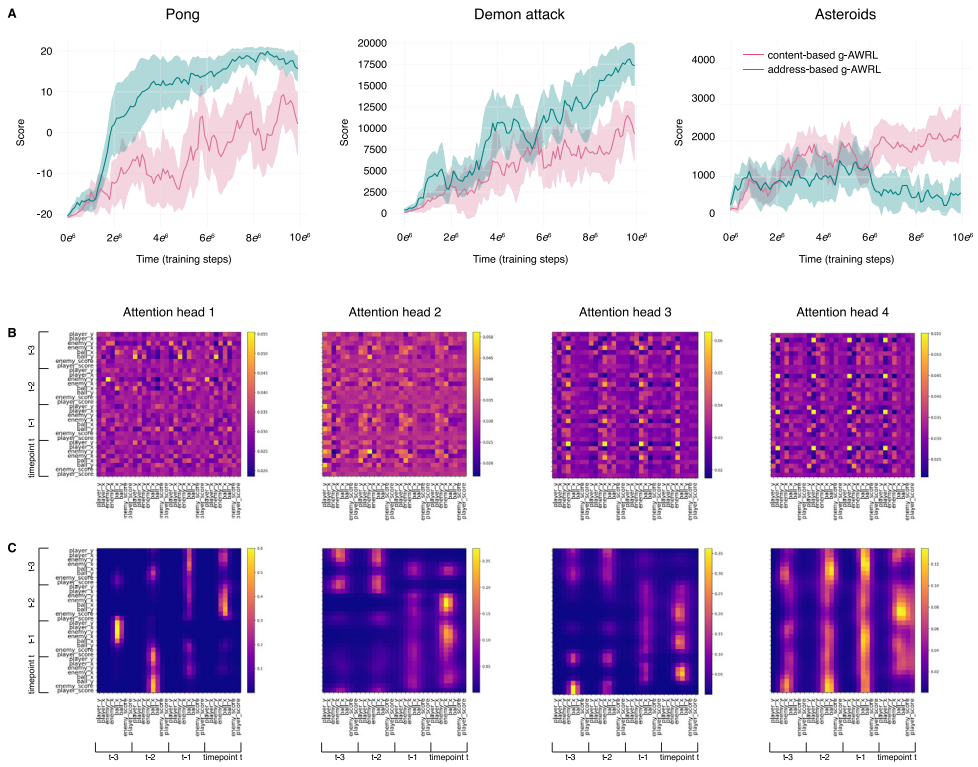

Fig. 6. Comparison of exclusive spatial- and content-based attention. (A) Performance curves in the three Atari games *Pong*, *Demon Attack*, *Asteroids* for the g-AWRL version with only spatial-based attention or content-based attention. Thick lines represent the mean, shaded areas the S.E.M. (B) Example attention maps of the 4 attention heads in the first layer of a g-AWRL agent equipped with content-based attention, performing the game *Pong* after training. (C) Example attention maps of the 4 attention heads in the first layer of a g-AWRL agent equipped with spatial-based attention, performing the game *Pong* after training.

agent is forced to identify salient feature values which might share the same neural codes regardless of the features they represent. Meanwhile, the address-based agent, while grounding attention weights on the addresses (which can be understood as an immutable feature identifier), will still transform the weighted features using the final linear layer in the self-attention module. Unlike in the purely content-based agent, all venues of information contained in the baseline g-AWRL architecture are still present and being processed, though in exclusion of each other. The one difference is that attention maps remain static and need to track task-relevant features continuously. This process is consistent with the original neuroscience AWRL formulation since the model does not explain temporal dynamics, with some features being task-relevant only some of the time. If task-state representations truly only highlight a set of features for a given time (regardless of their content) (Leong et al., 2017; Niv et al., 2015), then the address-based model can be understood as the most faithful translation of AWRL.

As expected, attention patterns vary significantly between purely content- vs. purely address-based attention (Fig. 6B-C). In the content-based case (for the game *Pong*, Fig. 6B), attention patterns fluctuate between observations. In line with previous reports (Mott, Zoran, Chrzanowski, Wierstra, & Rezende, 2019), we find that particular feature ranges trigger larger attention scores in a bottom-up fashion. For example, in *Pong*, the ball position will not be attended to by the agent unless it reaches a distinct set of values — specifically, positions in close proximity to the player and enemy agents. In this case the attention score is computed by the dot products of vector embeddings of that particular position. That is, the network adapts to finding specific feature expressions, or values, that correspond to better explanations of variance in the value-function. Conversely, in the address-based attention agent (for the game *Pong*, Fig. 6C), patterns do not fluctuate at all after training. The agent has no perceivable notion of either time or the current observation until the attention-weighted observation enters the value-function- and policy-networks. Attention is fully conditioned on task-learning. Thus, the agent is forced to highlight those feature channels that may or may not hold task-relevant information continuously. The corresponding attention patterns are close to static, highlighting these channels. Different attention heads converge to select and combine specific features like player position over time.

10. Discussion

This work presents an empirical application of the theoretical Attention-Weighted Reinforcement Learning (AWRL) framework, a model of feature-based attention in biological RL, in canonical deep-RL tasks. AWRL mathematically illustrates how organisms select multimodal stimulus dimensions to form lower-dimensional state representations subject to particular task-settings. This process is necessary since most real-world scenarios faced by biological organisms present few task-relevant features amid a majority of equally salient distractor or irrelevant features.

We first translate the AWRL framework into a trainable neural network architecture (d-AWRL), using the self-attention mechanism popularized in natural language processing (Bahdanau et al., 2014; Vaswani et al., 2017). We demonstrate that d-AWRL is a special case of self-attention combined with value function approximation and extend the model to allow for relational, beyond purely selective, processing. Our proposed g-AWRL approach offers a more effective solution to the binding problem — a formulation of the challenging task to select and relate stimuli, knowledge, and memory to represent the current experience neurally. In the original AWRL model, task-state representations correspond to neural codes that allow for accurate

value-prediction, conditioned on a given task. The same is true for our applied version, where feature-specific attention weights are made differentiable, and thus learnable, with respect to value-prediction errors in accordance with the Mackintosh model of attention (Mackintosh, 1975).

The results of the first experiment indicate that our generalized approach to solving the feature binding problem through feature compilation (and concomitantly avoid the limitation of only using features in isolation) is essential to perform competitively in environments were stimuli have more than a few dimensions such as Atari video games (Fig. 2A). The emergent dynamics of the attention mechanism suggest that agents integrate features over time, as well as modal features that provide information such as relative distances between objects. Similarly, elegant recent work with deep convolutional neural networks has highlighted how biological attention mechanisms can improve network classification performance mostly by modulating higher layers (Lindsay & Miller, 2018), which are generally associated with more complex features created by the compilation of simpler ones. This bears a strong resemblance to the brain, as higher visual areas — but also parietal or frontal regions as one moves along the cortical hierarchy, are related to more abstract representations (Bernardi et al., 2020; Cortese et al., 2020; Kikumoto & Mayr, 2020), and implicated in sophisticated RL-type behavioral/conceptual navigation (Brunec & Momennejad, 2019; Cortese, Lau & Kawato, 2020; Momennejad, Otto, Daw, & Norman, 2018; O'Doherty, Kringelbach, Rolls, Hornak, & Andrews, 2001).

The second experiment demonstrates that an RL agent endowed with self-attention has a notable advantage over comparable models in settings with many noisy features per observation, exhibiting stable performance retention as the number of distractor features increases — particularly if we only consider models with similar number of parameters. Notably, these environments come much closer to the challenges faced by biological organisms, as they are rich with salient, but task-irrelevant, stimuli. The performance retention most likely results from non-local processing, which allows the g-AWRL agent to consider every possible combination of features for its task-state representations. At the same time, more traditional models like the fully-connected- and both convolutional baselines tend to struggle to capture larger observation spaces concisely. Meanwhile, the g-AWRL collapses the space of possible feature combinations effectively by optimizing its value-function estimate with respect to its attention weights. Admittedly, considering all input features, and all possible combinations, may not be exactly biologically plausible. But ultimately, this may or may not be a limitation — in the current work it proves advantageous, and one can think of other situations in which it may not (e.g., limited processing capacity, speed of processing, size of the network in question), raising the need for alternative solutions in the form of memory, abstractions or even consciousness (Bengio, 2017; Dehaene, Lau, & Kouider, 2017; Ho, Abel, Griffiths, & Littman, 2019; Lengyel & Dayan, 2008).

In the first two experiments, the g-AWRL agents use observations where features are concatenated with a positional encoding. This vectorized positional index serves as an identifier of the feature. One possible interpretation that relates closely to neuroscience is that this embedding holds information on which neuron or cluster of neurons represents a particular feature. In physiological terms, attending to such pre-selected neurons would mean a modulation of their firing rates when preferred stimuli are detected (e.g., see the feature-similarity gain model (Martinez-Trujillo & Treue, 2004; Treue & Martinez Trujillo, 1999)). In the final experiment, we thus examine differences in performance and attention patterns when the attention scores were conditioned only on this identifier (top-down, task-driven) or only on the actual feature content (bottom-up, stimulus-driven). The results indicate that a combination of both is undoubtedly superior, as can be naturally expected under the lenses of biological attention (Knudsen, 2007).

With the initial success of the presented experiments, the popularity of self-attention mechanisms and attention in general, as well as the blossoming of neuroscience-inspired artificial intelligence, many avenues remain to be explored. For example, using linearly complex self-attention methods (Wang, 2020), or different forms of RL such as off-policy methods that would allow active-sampling and prioritized experience replay (Mattar & Daw, 2018; Moore & Atkeson, 1993; Schaul, Quan, Antonoglou, & Silver, 2016) for the integration of the Pearce–Hall and Mackintosh models of attention learning (Mackintosh, 1975; Pearce & Hall, 1980). Further, RL alone is unlikely to account for all effects of biological learning. For example, in decision-making tasks that require attending to different features in order to obtain rewards, human participants often perform better than chance after only 2–3 trials, an effect brought about by the ability to rapidly switch focus between features (Leong et al., 2017; Radulescu, Niv, & Ballard, 2019). This suggests that humans employ, at least in part, alternative strategies such as serial hypothesis testing (Donoso, Collins, & Koechlin, 2014; Radulescu, Niv et al., 2019; Radulescu, Niv, & Daw, 2019). In environments with a stable structure, alternative methods to gradient-based and reinforcement learning – such as particle filtering – could allow an agent to reach robust value-function estimates faster (Radulescu, Niv et al., 2019). Nevertheless, in environments with high numbers of continuous features, with stochastic and dynamic changes, gradual updates would probably remain superior. An interesting way forward would be to develop models that can adjust the way attention shifts – gradual updates vs. quick changes – based on environment or task characteristics.

The above points bring the discussion close to the delicate comparison between human and machine performances (Firestone, 2020). How does the g-AWRL agent described here compare with humans in terms of learning speed? As it currently stands, the proposed model is unlikely to reach human levels in tasks commonly used in human neuroscience or psychology experiments, such as the task illustrated in the original formulation of AWRL. This is mainly due to the slowness of gradient-based updates and the still significant number of tunable parameters in the g-AWRL model. Besides, humans have an exceptional ability to quickly garner *adequate* skill in any number of previously unknown tasks, implemented most likely by various higher cognitive functions (Cortese et al., 2019). Mastery, on the other hand, takes most of us ample amounts of training, especially in tasks that require complex motor behavior beyond accurate value estimates. State-of-the-art deep RL approaches, while requiring very large training sample data to reach even acceptable performances, tend to converge to top scores quite rapidly once initial success is observed. Our g-AWRL model displays very similar performance profiles – especially in the simpler game *Pong*, with steep learning curves after initial periods of low score. A compelling direction for future work will be a formal comparison of machine and human learning trajectories both over short and longer time horizons, and at different levels of task complexity. Together with further exploration of rapid value-function learning to supplement and accelerate the robust policy acquisition in deep RL algorithms, these lines of work will allow us to better understand the unique elements of human learning.

As a straightforward explorative extension of our current approach, one may focus control with the softmax activation function that is an integral part of the self-attention mechanism. The softmax turns the logits, representing how much each sequence element is related to the current query element, into a distribution. The temperature parameter T in the softmax governs the entropy of the resulting distribution. In our application of self-attention as a feature selector, lower temperatures

would enforce sparser feature selection, possibly resulting in attention heads that carry only a single feature representation. The effects on learning of sparse and dense heads need to be researched in future work, as sparser heads might lead to more compact task-state representations and even better interpretability of the model behavior. Unlike the explicit comparison of single-selection vs. feature-mixing in this paper, reducing the temperature parameter will allow the agent to mix features as long as they are weighted equally. Sparsity might lead the g-AWRL to approach the speed of particle filtering, which has been suggested to well reflect human choice behavior based on feature learning (Radulescu, Niv et al., 2019).

Finally, the majority of deep-RL models are trained end-to-end, meaning that each trainable parameter is optimized with respect to a value-function (in some cases, a policy) — from stimulus input to action output. This paradigm often results in impressive performances on narrow sets of challenges, but ultimately corresponds to a severe case of over-specialization and is not suitable for multi-tasking or actual task generalization settings. Ideally, pre-trained or out of the box models could be introduced as direct sensory modules to RL agents. In the visual domain, variational autoencoders (VAE) have achieved great success in representing interpolable classes at small scales, i.e., as small feature vectors (Kingma, Rezende, Mohamed, & Welling, 2014), and could be used as such. Yet, without modification the VAE approach has specific limitations that prove fatal to learning for decision-making. Most glaringly, the reliance upon pixel-based loss functions (measuring the importance of visual features based on the relative frequency of their associated pixel-values), and omitting smaller but semantically significant features from their representations. A recent approach for unsupervised representation learning attempts to remedy the problem by introducing a mutual information-based loss function (the Spatiotemporal Deep InfoMax) (Anand et al., 2020). In this way, a convolutional neural network will generate feature vectors from RGB-images that capture both local features of small image patches and a global representation relating all image patches. The representations are trained outside the RL cycle and promise to be general enough to be applied to several tasks within the same environment. Additionally, the authors introduce a creative temporal bias, in that the model does not encode simple images but image transitions, ensuring that the most salient parts of the image are captured in the limited representation space. The feature-based models presented in this work certainly need a robust stimulus-to-representation encoder to function in a broader range of scenarios. The Infomax approach is a promising candidate for a lightweight solution to this challenge. As a proof-of-concept, exploratory experiment, we show that pairing our g-AWRL approach with the Deep InfoMax leads to fast learning and significant improvements in reaching top scores in the game *Pong* (Fig. S7).

To conclude, the approach and results presented in this study provide theoretical and empirical evidence that endowing an agent with a simple attention mechanism paired with RL is one solution to learn appropriate behavioral policies in complex, noisy environments. We hope this work will further the fruitful exchange between neuroscience and machine learning.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank Miho Nagata and Mieko Hirata for administrative support, Dr. Giuseppe Lisi and Dr. Barbara Hammer for useful discussions. This work was supported by JST ERATO (Grant Number JPMJER1801), Japan; by the Innovative Science and Technology Initiative for Security (Grant Number JP004596) from the Acquisition, Technology & Logistics Agency (ATLA), Japan.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neunet.2021.09.023.

References

Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., & Devon Hjelm, R. (2020). Unsupervised state representation learning in atari. arXiv:1906.08226v5.

Ba, J., Mnih, V., & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. arXiv:1412.7755.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

Bar-Gad, I., Morris, G., & Bergman, H. (2003). Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Progress in Neurobiology (Oxford)*, *71*(6), 439–473.

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2015). The arcade learning environment: An evaluation platform for general agents. In *IJCAI 2015*.

Bellman, R. (1957). *Dynamic programming*. Princeton, New Jersey, USA: Princeton University Press.

Bengio, Y. (2017). The consciousness prior. arXiv:1709.08568.

Bernardi, S., Benna, M. K., Rigotti, M., Munuera, J., Fusi, S., & Daniel Salzman, C. (2020). The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, *0*(0).

Brunec, I. K., & Momennejad, I. (2019). Predictive representations in hippocampal and prefrontal hierarchies. *BioRxiv*, Article 786434.

Chun, M. M., Golomb, J. D., & Turk-Browne, N. B. (2011). A taxonomy of external and internal attention. *Annual Review of Psychology (Palo Alto, CA)*, *62*, 73–101.

Corbetta, M., & Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews. Neuroscience*, *3*(3), 201–215.

Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. arXiv:1911.03584.

Cortese, A., De Martino, B., & Kawato, M. (2019). The neural and cognitive architecture for learning from a small sample. *Current Opinion in Neurobiology (London)*, *55*, 133–141.

Cortese, A., Lau, H., & Kawato, M. (2020). Unconscious reinforcement learning of hidden brain states supported by confidence. *Nature Communications*, 1–14.

Cortese, A., Yamamoto, A., Hashemzadeh, M., Sepulveda, P., Kawato, M., & De Martino, B. (2020). Value shapes abstraction during learning. *BioRxiv*, 2020.10.29.361469.

Dehaene, S., Lau, H., & Kouider, S. (2017). What is consciousness, and could machines have it? *Science*, *358*(6362), 486–492.

Donoso, M., Collins, A. G. E., & Koechlin, E. (2014). Foundations of human reasoning in the prefrontal cortex. *Science*, *344*(6191), 1481–1486.

Doya, K. (2007). Reinforcement learning: Computational theory and biological mechanisms. *HFSP Journal*, *1*(1), 30–40.

Ekman, M., Roelfsema, P. R., & de Lange, F. P. (2020). Object selection by automatic spreading of top-down attentional signals in V1. *Journal of Neuroscience (New York, NY)*, *40*(48), 9250–9259.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., et al. (2020). Implementation matters in deep policy gradients: A case study on PPO and TRPO. arXiv:2005.12729.

Farashahi, S., Rowe, K., Aslami, Z., Lee, D., & Soltani, A. (2017). Feature-based learning improves adaptability without compromising precision. *Nature Communications*, *8*(1), 1768.

Firestone, C. (2020). Performance vs. competence in human-machine comparisons. *Proceedings of the National Academy of Sciences of the United States of America*.

Frank, M. J., & Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: computational analysis. *Cerebral Cortex*, *22*(3), 509–526.

Gazzaley, A., & Nobre, A. C. (2012). Top-down modulation: bridging selective attention and working memory. *Trends in Cognitive Science*, *16*(2), 129–135.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. arXiv:1705.03122.

Gong, M., & Liu, T. (2020). Biased neural representation of feature-based attention in the human frontoparietal network. *Journal of Neuroscience (New York, NY)*, *40*(43), 8386–8395.
Grunewald, A., & Skoumbourdis, E. K. (2004). The integration of multiple stimulus features by V1 neurons. *Journal of Neuroscience (New York, NY)*, *24*(41), 9185–9194.
Henschke, J. U., Dylda, E., Katsanevaki, D., Dupuy, N., Currie, S. P., Amvrosiadis, T., et al. (2020). Reward association enhances stimulus-specific representations in primary visual cortex. *Current Biology (London)*.
Ho, M. K., Abel, D., Griffiths, T. L., & Littman, M. L. (2019). The value of abstraction. *Current Opinion in Behavioral Sciences*, *29*, 111–116.
Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal Physiology*, *160*, 106–154.
Jones, M., & Cañas, F. (2010). Integrating reinforcement learning with models of representation learning. In *Proceedings of the annual meeting of the cognitive science society*.
Jörntell, H., Bengtsson, F., Geborek, P., Spanne, A., Terekhov, A. V., & Hayward, V. (2014). Segregation of tactile input features in neurons of the cuneate nucleus. *Neuron*, *83*(6), 1444–1452.
Kikumoto, A., & Mayr, U. (2020). Conjunctive representations that integrate stimuli, responses, and rules are critical for action selection. *Proceedings of the National Academy of Sciences of the United States of America*.
Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. arXiv:1406.5298.
Knudsen, E. I. (2007). Fundamental components of attention. *Annual Review of Neuroscience (Palo Alto, CA)*, *30*, 57–78.
Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychological Review (Washington, DC)*, *99*(1), 22–44.
Lengyel, M., & Dayan, P. (2008). Hippocampal contributions to control: The third way. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Advances in neural information processing systems (Vol. 20)* (pp. 889–896). Curran Associates, Inc.
Leong, Y., Radulescu, A., Daniel, R., Dewoskin, V., & Niv, Y. (2017). Dynamic interaction between reinforcement learning and attention in multidimensional environments. *Neuron*, *93*(2), 451–463.
Lindsay, G. W., & Miller, K. D. (2018). How biological attention mechanisms improve task performance in a large-scale visual system model. *Elife*, *7*.
Luo, X., Roads, B. D., & Love, B. C. (2020). The costs and benefits of goal-directed attention in deep convolutional neural networks. arXiv:2002.02342.
Mack, M., Love, B., & Preston, A. (2016). Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proceedings of the National Academy of Sciences*, *113*(46), 13203–13208.
Mackintosh, N. J. (1975). A theory of attention: Variations in the associability of stimuli with reinforcement. *Psychological Review (Washington, DC)*, *82*(4), 276–298.
Manchin, A., Abbasnejad, E., & van den Hengel, A. (2019). Reinforcement learning with attention that works: A self-supervised approach. arXiv:1904.03367.
Martinez-Trujillo, J. C., & Treue, S. (2004). Feature-based attention increases the selectivity of population responses in primate visual cortex. *Current Biology (London)*, *14*(9), 744–751.
Mattar, M. G., & Daw, N. D. (2018). Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 1.
Mnih, V., Badia, A. P., Mirza, M., Graves, A., Harley, T., Lillicrap, T. P., et al. (2016). Asynchronous methods for deep reinforcement learning. In *ICML 2016*.
Momennejad, I., Otto, A. R., Daw, N. D., & Norman, K. A. (2018). Offline replay supports planning in human reinforcement learning. *Elife*, *7*.
Moore, A. W., & Atkeson, C. G. (1993). Prioritized sweeping reinforcement learning with less data and less time. *Machine Learning*, *13*, 103–130.
Mott, A., Zoran, D., Chrzanowski, M., Wierstra, D., & Rezende, D. J. (2019). Towards interpretable reinforcement learning using attention augmented agents. arXiv:1906.02500.
Niv, Y. (2019). Learning task-state representations. *Nature Neuroscience*, *22*(10), 1544–1553.
Niv, Y., Daniel, R., Geana, A., Gershman, S., Leong, Y., Radulescu, A., et al. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience (New York, NY)*, *35*(21), 8145–8157.
O'Doherty, J., Kringelbach, M. L., Rolls, E. T., Hornak, J., & Andrews, C. (2001). Abstract reward and punishment representations in the human orbitofrontal cortex. *Nature Neuroscience*, *4*(1).
Pearce, J. M., & Hall, G. (1980). A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review (Washington, DC)*, *87*(6), 532–552.
Posner, M. I., & Gilbert, C. D. (1999). Attention and primary visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *96*(6), 2585–2587.
Radulescu, A., Niv, Y., & Ballard, I. (2019). Holistic reinforcement learning: The role of structure and attention. *Trends in Cognnitive Science*.
Radulescu, A., Niv, Y., & Daw, N. (2019). A particle filtering account of selective attention during learning. In *2019 conference on cognitive computational neuroscience*. Brentwood, Tennessee, USA: Cognitive Computational Neuroscience.
Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. In *ICLR 2016*.
Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv:1707.06347.
Somers, D. C., Dale, A. M., Seiffert, A. E., & Tootell, R. B. (1999). Functional MRI reveals spatially specific attentional modulation in human primary visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *96*(4), 1663–1668.
Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., & Ignateva, A. (2015). Deep attention recurrent Q-network. arXiv:1512.01693.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
Treue, S., & Martinez Trujillo, J. C. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, *399*(6736), 575–579.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in neural information processing systems (Vol. 30)* (pp. 5998–6008). Curran Associates, Inc.
Wang, X.-J. (2020). Macroscopic gradients of synaptic excitation and inhibition in the neocortex. *Nature Reviews. Neuroscience*, *21*(3), 169–178.
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. arXiv:1502.03044.
Yuezhang, L., Zhang, R., & Ballard, D. H. (2018). An initial attempt of combining visual selective attention with deep reinforcement learning. arXiv:1811.04407.