Crossmodal attentive skill learner: learning in Atari and beyond with audio–video inputs

Crossmodal attentive skill learner: learning in Atari and beyond with audio–video inputs

Cite this article as: Dong-Ki Kim, Shayegan Omidshafiei, Jason Pazis and Jonathan P. How, Crossmodal attentive skill learner: learning in Atari and beyond with audio–video inputs, Autonomous Agents and Multi-Agent Systems https://doi.org/10.1007/s10458-019-09439-5

Autonomous Agents and Multi-Agent Systems manuscript No.
(will be inserted by the editor)

Crossmodal Attentive Skill Learner

Learning in Atari and Beyond with Audio-Video Inputs

Dong-Ki Kim · Shayegan Omidshafiei · Jason Pazis† · Jonathan P. How

Received: date / Accepted: date

Abstract This paper introduces the Crossmodal Attentive Skill Learner (CASL), integrated with the recently-introduced Asynchronous Advantage Option-Critic (A2OC) architecture [16] to enable hierarchical reinforcement learning across multiple sensory inputs. Agents trained using our approach learn to attend to their various sensory modalities (e.g., audio, video) at the appropriate moments, thereby executing actions based on multiple sensory streams without reliance on supervisory data. We demonstrate empirically that the sensory attention mechanism anticipates and identifies useful latent features, while filtering irrelevant sensor modalities during execution. Further, we provide concrete examples in which the approach not only improves performance in a single task, but accelerates transfer to new tasks. We modify the Arcade Learning Environment (ALE) [8] to support audio queries[1], and conduct evaluations of crossmodal learning in the Atari 2600 games H.E.R.O. and Amidar. Finally, building on the recent work of Babaeizadeh et al. [5], we open-source a fast hybrid CPU-GPU implementation of CASL.[2]

Work was supported by Boeing Research & Technology, ONR MURI Grant N000141110688, BRC Grant N000141712072, IBM (as part of the MIT-IBM Watson AI Lab initiative), and AWS Machine Learning Research Awards program. We also thank the three anonymous reviewers for their helpful suggestions.

† Work done prior to Amazon involvement of Jason Pazis, and does not reflect views of the Amazon company.

Dong-Ki Kim
Massachusetts Institute of Technology
E-mail: dkkim93@mit.edu

Shayegan Omidshafiei
Massachusetts Institute of Technology
E-mail: shayegan@mit.edu

Jason Pazis
Amazon Alexa
E-mail: pazisj@amazon.com

Jonathan P. How
Massachusetts Institute of Technology
E-mail: jhow@mit.edu

[1] ALE-audio code available at `https://github.com/shayegano/Arcade-Learning-Environment`
[2] CASL code available at `https://github.com/shayegano/CASL`

Keywords hierarchical learning · reinforcement learning · multimodal learning

1 Introduction

Intelligent agents should be capable of using local sensory streams to realize long-term goals. In recent years, the combined progress of computational capabilities and algorithmic innovations has afforded reinforcement learning (RL) [42] approaches the ability to achieve this desiderata in domains with large state-action spaces, exceeding expert-level human performance in tasks such as Atari and Go [32, 39]. Nonetheless, many of these algorithms thrive primarily in well-defined mission scenarios learned in isolation from one another; such monolithic approaches are not sufficiently scalable to missions in which the goals may be less clearly defined and sensory inputs found salient in one domain may be less relevant than in another.

How should agents learn effectively in domains of high dimensionality, where tasks are durative, agents receive sparse feedback, and sensors compete for limited computational resources? One promising avenue is hierarchical reinforcement learning (HRL), focusing on problem decomposition for learning transferable skills. Temporal abstraction enables exploitation of domain regularities to provide the agent hierarchical guidance in the form of options or sub-goals [23, 33, 43]. Options help agents improve learning by mitigating scalability issues in long-duration missions, by reducing the effective number of decision epochs. In the parallel field of supervised learning, temporal dependencies have been captured proficiently using attention mechanisms applied to encoder-decoder based sequence-to-sequence models [6, 26]. *Attention* empowers the learner to focus on the most pertinent stimuli and capture longer-term correlations in its encoded state, for instance to conduct neural machine translation or video captioning [46, 47]. Recent works also show benefits of spatio-temporal attention in RL [31, 40].

One can interpret the temporal abstraction approaches discussed above as conducting dimensionality reduction on the axis of time. In view of the scalability benefits afforded by dimensionality reduction, this paper proposes an RL paradigm exploiting hierarchies in the dimensions of *time* and *sensor modalities*. Our approach enables agents to learn rich skills that attend to and exploit pertinent crossmodal (multi-sensor) signals at the appropriate moments. The introduced crossmodal skill learning approach largely benefits an agent learning in a high-dimensional domain (e.g., a robot equipped with multiple high-dimensional sensors, such as those yielding audio and video observation streams). Instead of the expensive operation of processing and/or storing data from all sensors, we demonstrate that our approach enables such an agent to focus on sensors that are most important. This, in turn, leads to more efficient use of the agent's limited computational and storage resources (e.g., its finite-sized memory).

This paper focuses on combining two sensor modalities: audio and video. While these modalities have been previously used for supervised learning [34], to our knowledge they have yet to be exploited for crossmodal skill learning in RL. We provide concrete examples where the proposed HRL approach not only improves performance in a single task, but accelerates transfer to new tasks. We demonstrate the attention mechanism anticipates and identifies useful latent features, while filtering irrelevant sensor modalities during execution. The attention mechanism also impacts the gradient computation during training and filters out noisy gradients,

which results in stabilized learning and increased learning speed. We also show first ever results in the Arcade Learning Environment with audio-video inputs, where we modified the environment to support agent audio queries. In addition, we provide insight into how our model functions internally by analyzing the interactions of attention and memory. Building on the recent work of Babaeizadeh et al. [5], we open-source a fast hybrid CPU-GPU implementation of our framework. Finally, note that despite the focus on audio-video sensors in this paper, the framework presented is general and readily applicable to other sensory inputs.

2 Related Work

2.1 Multimodal and Crossmodal Learning

Our work is most related to crossmodal learning approaches that take advantage of multiple input sensor modalities. Fusion of multiple modalities or sources of information is an active area of research. Works in the diverse domains of sensor fusion in robotics [13, 27, 36], audio-visual fusion [7, 9, 34, 41], and image-point cloud fusion [2, 10] have shown that models utilizing multiple modalities tend to outperform those learned from unimodal inputs.

In general, approaches for multimodal fusion can be broadly classified depending on the means of integration of the various information sources. Filtering-based frameworks (e.g., the extended Kalman filter) are widely used to combine multi-sensor readings in the robotics community [13, 27, 36]. In machine learning, approaches based on graphical models [7, 9] and conditional random fields [24] have been used to integrate multimodal features [2, 10]. More recently, deep learning-based approaches that learn a representations of features across multiple modalities have been introduced [15, 21, 34, 41].

2.2 Attention

A variety of attentive mechanisms have been considered in recent works, primarily in application to supervised learning. Temporal attention mechanisms have been successfully applied to the field of neural machine translation, for instance in Bahdanau et al. [6] and Luong et al. [26], where encoder-decoder networks work together to transform one sequence to another. Works also exist in multimodal attention for machine translation, using video-text inputs [11]. Spatial attention models over image inputs have also been combined with Deep Recurrent Q-Networks [17] for RL [40]. Works have also investigated spatially-attentive agents trained via RL to conduct image classification [4, 31]. As we later demonstrate, the crossmodal attention-based approach used in this paper enables filtering of irrelevant sensor modalities, leading to improved learning and more effective use of the agent's memory.

2.3 Hierarchical Reinforcement Learning

There exists a large body of HRL literature, targeting both fully and partially-observable domains. Our work leverages the options framework [43], specifically

the recent Asynchronous Advantage Option-Critic (A2OC) [16] algorithm, to learn durative skills. HRL is an increasingly-active field, with a number of recent works focusing on learning human-understandable and/or intuitive skills. In Andreas et al. [3], annotated descriptors of policies are used to enable multitask RL using options. Multitask learning via parameterized skills is also considered in Da Silva et al. [14], where a classifier and regressor are used to, respectively, identify the appropriate policy to execute, then map to the appropriate parameters of said policy. Construction of *skill chains* is introduced in Konidaris and Barto [22] for learning in continuous domains. In Machado et al. [28], option discovery is conducted through eigendecomposition of MDP transition matrices, leading to transferable options that are agnostic of the task reward function. FeUdal Networks [44] introduce a two-level hierarchy, where a manager defines sub-goals, and a worker executes primitive actions to achieve them. A related HRL approach is also introduced in Kulkarni et al. [23], where sub-goals are hand-crafted by a domain expert. Overall, the track record of hierarchical approaches for multitask and transfer learning leads us to use them as a basis for the proposed framework, as our goal is to learn scalable policies over high-volume input streams.

While the majority of works utilizing multi-sensory and attentive mechanisms focus on supervised learning, our approach targets RL. Specifically, we introduce an HRL framework that combines crossmodal learning, attentive mechanisms, and temporal abstraction to learn durative skills.

3 Background

This section summarizes Partially Observable Markov Decision Processes (POMDPs) and options, which serve as foundational frameworks for our approach.

3.1 POMDPs

This work considers an agent operating in a partially-observable stochastic environment, modeled as a POMDP $\langle \mathbb{S}, \mathbb{A}, \mathbb{O}, \mathcal{T}, \mathcal{O}, \mathcal{R}, \gamma \rangle$ [20]. $\mathbb{S}$, $\mathbb{A}$, and $\mathbb{O}$ are, respectively, the state, action, and observation spaces. At each timestep, the agent executes action $a \in \mathbb{A}$ in state $s \in \mathbb{S}$, transitions to state $s' \sim \mathcal{T}(s, a, s')$, receives observation $o \sim \mathcal{O}(o, s', a)$, and reward $r = \mathcal{R}(s, a)$. The value of state s under policy $\pi : Dist(\mathbb{S}) \rightarrow \mathbb{A}$ is the expected return $V_\pi(s) = \mathbb{E}[\sum_{k=0}^{T} \gamma^k \mathcal{R}(s^{t+k}, a^{t+k}) | s^t = s]$, where $Dist(\mathbb{S})$ denotes the state distribution, s^t and a^t denote the state and action at timestep t, respectively, T is the horizon, and $\gamma \in [0, 1)$ is the discount factor. The objective is to learn an optimal policy π^*, which maximizes the value.

As POMDP agents only receive noisy observations of the latent state, policy π typically maps from the agent's belief (distribution over states) to the next action. Recent work has introduced Deep Recurrent Q-Networks (DRQNs) [17] for RL in POMDPs, leveraging recurrent Neural Networks (RNNs) that inherently maintain an internal state $\boldsymbol{h^t} \in \mathbb{R}^H$ to compress input history until timestep t, where $H \in \mathbb{N}$ is the dimension of the internal state. Throughout the paper, we give scalars either lowercase (e.g., s^t) or uppercase (e.g., H) variable names, column vectors lowercase names in bold typeface (e.g., $\boldsymbol{h^t}$), and matrices uppercase names with bold typeface (e.g., $\boldsymbol{W_m}$).

3.2 Options

The framework of *options* provides an RL agent the ability to plan using temporally-extended actions [43]. Option $\omega \in \Omega$ is defined by initiation set $\mathbb{I} \subseteq \mathbb{S}$, intra-option policy $\pi_\omega : \mathbb{S} \rightarrow Dist(\mathbb{A})$, and termination condition $\beta_\omega : \mathbb{S} \rightarrow [0, 1]$, where $Dist(\mathbb{A})$ denotes the action probability distribution. A policy over options π_Ω chooses an option among those that satisfy the initiation set. The selected option executes its intra-option policy until termination, upon which a new option is chosen. This process iterates until the goal state is reached.

Recently, the Asynchronous Advantage Actor-Critic framework (A3C) [30] has been applied to POMDP learning in a computationally-efficient manner by combining parallel actor-learners and Long Short-Term Memory (LSTM) cells [19]. Asynchronous Advantage Option-Critic (A2OC) extends A3C and enables learning option-value functions, intra-option policies, and termination conditions in an end-to-end fashion [16]. The option-value function models the value of state $s \in \mathbb{S}$ in option $\omega \in \Omega$,

$$Q_\Omega(s, \omega) = \sum_a \pi_\omega(a|s)\Big(\mathcal{R}(s, a) + \gamma \sum_{s'} \mathcal{T}(s'|s, a)U(s', \omega)\Big), \tag{1}$$

where $a \in \mathbb{A}$ is a primitive action and $U(s', \omega)$ represents the option utility function,

$$U(s', \omega) = (1 - \beta_\omega(s'))Q_\Omega(\omega, s') + \beta_\omega(s')(V_\Omega(s') - c). \tag{2}$$

A2OC introduces deliberation cost, c, in the utility function to address the issue of options terminating too frequently. Intuitively, the role of c is to impose an added penalty when options terminate, enabling regularization of termination frequency. The value function over options, V_Ω, is defined,

$$V_\Omega(s') = \sum_\omega \pi_\Omega(\omega|s')Q_\Omega(\omega, s'), \tag{3}$$

where π_Ω is the policy over options (e.g., an epsilon-greedy policy over Q_Ω). Assuming use of a differentiable representation, option parameters can be learned using gradient descent. Readers are referred to Harb et al. [16] for more details.

4 Approach

Our goal is to design a mechanism that enables the learner to modulate high-dimensional sensory inputs, focusing on pertinent stimuli that may lead to more efficient skill learning. This section presents motivations behind attentive skill learning, then introduces the proposed framework.

4.1 Attentive Mechanisms

Before presenting the proposed architecture, let us first motivate our interests towards attentive skill learning. One might argue that the combination of deep learning and RL already affords agents the representation learning capabilities

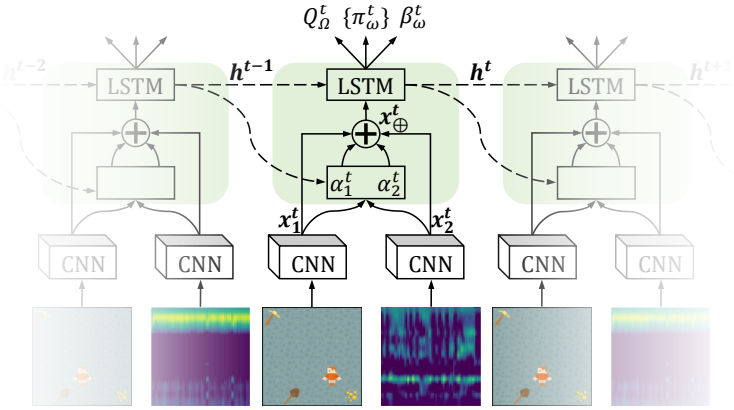


Fig. 1: CASL network architecture enables attention-based learning over multi-sensory inputs. CASL uses convolutional neural networks (CNNs) to extract sensor features $\boldsymbol{x_1^t}, \boldsymbol{x_2^t}$, which are then combined with attention α_1^t, α_2^t. Finally, LSTM outputs the option value Q_Ω^t, the intra-option policy π_ω^t, and the termination condition β_ω^t. Green highlighted region indicates crossmodal attention LSTM cell, trained via backpropagation through time.

necessary for proficient decision-making in high-dimensional domains; i.e., why the need for crossmodal attention?

Our ideas are motivated by the studies in behavioral neuroscience that suggest the interplay of attention and choice bias humans' value of information during learning, playing a key factor in solving tasks with high-dimensional information streams [25]. Research studying learning in the brain also suggest a natural pairing of attention and hierarchical learning, where domain regularities are embedded as priors into skills and combined with attention to alleviate the curse of dimensionality [35]. Other work also suggests attention plays a role in the intrinsic curiosity of agents during learning, through direction of focus to regions predicted to have high reward [29], high uncertainty [37], or both [38].

In view of these studies, we conjecture that crossmodal attention, in combination with HRL, improves representations of relevant environmental features that lead to superior learning and decision-making. Specifically, using crossmodal attention, agents combine internal beliefs with external stimuli to more effectively exploit multiple modes of input features for learning. As we later demonstrate, our approach captures temporal crossmodal dependencies, and enables faster and more proficient learning of skills in the domains examined.

4.2 Crossmodal Attentive Skill Learner

We propose Crossmodal Attentive Skill Learner (CASL), a novel framework for HRL. One may consider many blueprints for integration of multi-sensory attention into the options framework. Our proposed architecture is primarily motivated by the literature that taxonomizes attention into two classes: *exogeneous* and *endogeneous*. The former is an involuntary mechanism triggered automatically

by the inherent saliency of the sensory inputs, whereas the latter is driven by the intrinsic and possibly long-term goals, intents, and beliefs of the agent [12]. Previous attention-based neural architectures take advantage of both classes, for instance, to solve natural language processing problems [45]; our approach follows a similar schema.

The CASL network architecture is visualized in Fig. 1, with pseudocode presented in Algorithm 1. Let $M \in \mathbb{N}$ be the number of sensor modalities (e.g., vision, audio, etc.) and $\boldsymbol{x_m^t} \in \mathbb{R}^{X_m}$ denote extracted features from the m-th sensor at timestep t, where $m \in \{1, \ldots, M\}$ and $X_m \in \mathbb{N}$ is the dimension of the extracted features. For instance, $\boldsymbol{x_m^t}$ may correspond to feature outputs of a convolutional neural network given an image input. Note that we assume the number of extracted features for each modality is the same (i.e., $X_1 = X_2 = \ldots = X_M = X$). Given extracted features for all M sensors at timestep t, as well as hidden state $\boldsymbol{h^{t-1}}$, the proposed crossmodal attention layer learns the relative importance of each modality $\boldsymbol{\alpha^t} \in \Delta^{M-1}$, where Δ^{M-1} is the $(M-1)$-simplex:

$$\boldsymbol{z^t} = \tanh\Big(\underbrace{\sum_{m=1}^{M}(\boldsymbol{W_m x_m^t} + \boldsymbol{b_m})}_{\text{Exogeneous attention}} + \underbrace{\boldsymbol{W_h h^{t-1}} + \boldsymbol{b_h}}_{\text{Endogeneous attention}}\Big), \tag{4}$$

$$\boldsymbol{\alpha^t} = \text{softmax}\left(\boldsymbol{W_z z^t} + \boldsymbol{b_z}\right), \tag{5}$$

where $\boldsymbol{z^t} \in \mathbb{R}^Z$ is the internal embedding feature vector with the dimension $Z \in \mathbb{N}$, weight matrices $\boldsymbol{W_m} \in \mathbb{R}^{Z \times X}, \boldsymbol{W_h} \in \mathbb{R}^{Z \times H}, \boldsymbol{W_z} \in \mathbb{R}^{M \times Z}$ and bias vectors $\boldsymbol{b_m} \in \mathbb{R}^Z, \boldsymbol{b_h} \in \mathbb{R}^Z, \boldsymbol{b_z} \in \mathbb{R}^M$ are trainable parameters, respectively, and nonlinearities are applied element-wise. Then, the attended feature vector $\boldsymbol{x_\oplus^t}$ at timestep t is obtained by two possible options:

$$\boldsymbol{x_\oplus^t} = \begin{cases} \sum_{m=1}^{M} \alpha_m^t \boldsymbol{x_m^t}, & \text{(Summed attention)} \\ [(\alpha_1^t \boldsymbol{x_1^t})^T, \ldots, (\alpha_M^t \boldsymbol{x_M^t})^T]^T, & \text{(Concatenated attention)} \end{cases} \tag{6}$$

where $\alpha_m^t \in \mathbb{R}$ denotes the relative importance for m-th sensor.

For example, consider Fig. 1, where there are two sensor inputs of image and audio ($M = 2$). Assume the extracted feature dimension X of 3872, the LSTM hidden state dimension H of 16, and the internal embedding feature dimension Z of 16. Through a post-processing process, such as applying the convolutional neural network filters, the extracted feature vectors $\boldsymbol{x_1^t}, \boldsymbol{x_2^t} \in \mathbb{R}^{3872}$ are obtained, where $\boldsymbol{x_1^t}$ and $\boldsymbol{x_2^t}$ denote the extracted features from the image and audio data at timestep t, respectively. Then, we calculate the attention values $\boldsymbol{\alpha^t} \in \mathbb{R}^2$ through Eqs. (4) and (5). One possible learned attention values are $\boldsymbol{\alpha^t} = \{\alpha_1^t, \alpha_2^t\} = \{0.75, 0.25\}$, meaning that the agent should focus on the image data 3 times more than the audio data at timestep t. Finally, by Eq. (6), the attended feature vector $\boldsymbol{x_\oplus^t}$ is calculated.

Both exogeneous attention over sensory features $\boldsymbol{x_m^t}$ and endogeneous attention over LSTM hidden state $\boldsymbol{h^{t-1}}$ are captured in Eq. (4). The sensory feature extractor used in experiments consists of convolutional layers. Attended features $\alpha_m^t \boldsymbol{x_m^t}$ may be combined via summation or concatenation (Eq. (6)), then fed to an LSTM cell. The LSTM output captures temporal dependencies used to estimate option

Algorithm 1 Crossmodal Attentive Skill Learner (CASL)

Initialize global step counter $T \leftarrow 1$
repeat
  Initialize episode step counter $t \leftarrow 1$
  repeat
    Get sensor features at time t: $\{\boldsymbol{x_1^t}, \boldsymbol{x_2^t}, \ldots, \boldsymbol{x_m^t}\}$
    Process attended features $\boldsymbol{x_\oplus^t}$ using Eq. (6)
    if t == 1 or Option termination β_ω^{t-1} is True **then**
      Get option ω^t from policy over options π_Ω (see Section 3.2)
    end if
    Get primitive action a^t from intra-option policies π_ω
    Execute action a^t in the environment → Get reward r^t
    Update parameters using A2OC update rules (see Section 3.2 and Harb et al. [16])
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until Episode ends
until $T > T_{\max}$

values, intra-option policies, and termination conditions ($Q_\Omega^t, \pi_\omega^t, \beta_\omega^t$ in Fig. 1, respectively),

$$Q_\Omega^t(s,\omega) = \left\{\boldsymbol{W_Q h^t} + \boldsymbol{b_Q}\right\}_\omega, \tag{7}$$

$$\pi_\omega^t(a|s) = \left\{\text{softmax}(\boldsymbol{W_{\pi,\omega} h^t} + \boldsymbol{b_{\pi,\omega}})\right\}_a, \tag{8}$$

$$\beta_\omega^t(s) = \left\{\sigma(\boldsymbol{W_\beta h^t} + \boldsymbol{b_\beta})\right\}_\omega, \tag{9}$$

where weight matrices $\boldsymbol{W_Q} \in \mathbb{R}^{dim(\Omega)\times H}$, $\boldsymbol{W_{\pi,\omega}} \in \mathbb{R}^{dim(\mathbb{A})\times H}$, $\boldsymbol{W_\beta} \in \mathbb{R}^{dim(\Omega)\times H}$ and bias vectors $\boldsymbol{b_Q} \in \mathbb{R}^{dim(\Omega)}$, $\boldsymbol{b_{\pi,\omega}} \in \mathbb{R}^{dim(\mathbb{A})}$, $\boldsymbol{b_\beta} \in \mathbb{R}^{dim(\Omega)}$ are trainable parameters for the current option ω, $dim(\Omega)$ and $dim(\mathbb{A})$ refer to the dimension of the option and action space, respectively, and $\sigma(\cdot)$ is the sigmoid function. Note that $\{\cdot\}_\omega$ and $\{\cdot\}_a$ refer to the values for option ω and action a, respectively. Network parameters are updated using gradient descent. Entropy regularization of attention outputs $\boldsymbol{\alpha^t}$ was found to encourage exploration of crossmodal attention behaviors during training. For the hyper-parameter values that we used in our evaluations, refer to Section 5.6.

5 Evaluation

The proposed framework is evaluated on a variety of learning tasks with inherent reward sparsity and transition noise. We evaluate our approach in three domains: a door puzzle domain, a mining domain, and the Arcade Learning Environment (ALE) [8]. These environments include challenging combinations of reward sparsity and/or complex audio-video sensory input modalities that may not always be useful to the agent. The first objective of our experiments is to analyze performance of CASL in terms of learning rate and transfer learning. The second objective is to understand relationships between attention and memory mechanisms (as captured in the LSTM cell state). Finally, we modify ALE to support audio queries and evaluate crossmodal learning in the Atari 2600 games with long time horizons,

(a) The agent receives a reward for opening door 1 with key 1.

(b) The agent receives a reward for opening door 2 with key 2.

Fig. 2: Door puzzle domain. The agent must pick up the key and then open the correct door (depending on key color and audio) to receive a reward of +1. Otherwise, the agent receives a zero reward, making the domain challenging due to the sparse reward function.

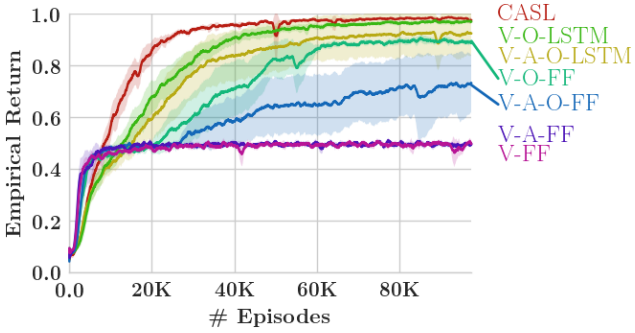
H.E.R.O. and Amidar. Atari games are complex benchmark domains, closer to real-world settings with long-term environment interactions. Experimental details are summarized in Section 5.6.

5.1 Crossmodal Learning and Transfer

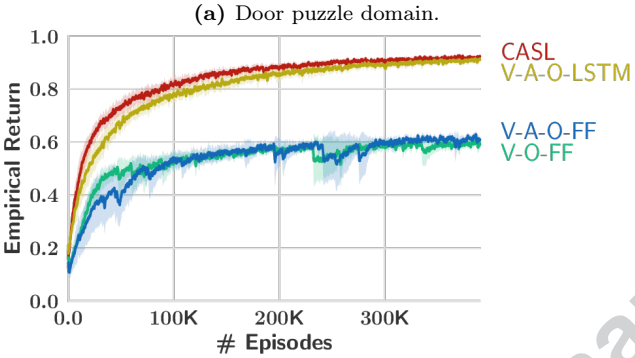
We first evaluate crossmodal attention in a sequential door puzzle game, where the agent spawns in a 5×5 gridworld with two locked doors (door 1 and 2) and a key at fixed positions. The agent can use four actions (move up, down, right, left ; $dim(\mathbb{A}) = 4$) to navigate around the domain. The key type (either key 1 or key 2) is randomly generated, and its observable color indicates the associated door (see Fig. 2). The agent also hears a sound when adjacent to the key (where the sound is dependent on key type), and hears noise otherwise. The agent must find and pick up the key (which then disappears), then find and open the correct door to receive +1 reward. This is the only situation wherein the agent receives a reward, making the reward space quite sparse. The game terminates upon the opening of either door or the timestep exceeds a pre-specified value T.

The agent's sensory inputs $\boldsymbol{x}_{\boldsymbol{m}}^{\boldsymbol{t}}$ are the grayscale domain images and audio spectrogram. Note that the audio input is represented visually by converting it into the spectrogram (see Section 5.6 for details) so that the convolutional filters can be applied. This task was designed in such a way that audio is not necessary to achieve the task – the agent can certainly focus on learning a policy mapping from visual features to open the correct door. However, audio provides potentially useful signals that may accelerate learning, making this a domain of interest for analyzing the interplay of attention and sensor modalities.

Attention Improves Learning Rate Figure 3a shows ablative training results for several network architectures. The three LSTM-based skill learners (including CASL) converge to the optimal value. Interestingly, the network that ignores audio inputs (V-O-LSTM) converges faster than its audio-enabled counterpart (V-A-O-LSTM), indicating the latter is overwhelmed by the extra sensory modality. We



(a) Door puzzle domain.

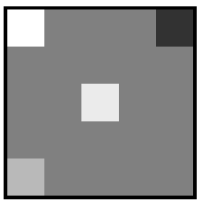


(b) Door puzzle transfer learning.

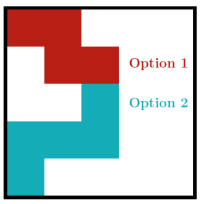
Fig. 3: Ablative analysis demonstrating that CASL improves learning rate compared to other networks. Abbreviations: **V**: video, **A**: audio, **O**: options, **FF** feedforward net, **LSTM**: Long Short-Term Memory net. Mean and 95% confidence interval computed for 10 independent runs are shown in all figures.

conduct t-tests with $p < 0.05$ based on the mean and standard deviation (std) of the area under the learning curve (AUC). Introduction of crossmodal attention enables CASL to converge faster than all other networks, showing the largest AUC with statistical significance (e.g., $p = 0.0001$ against V-O-LSTM and $p = 0.0077$ against V-A-O-LSTM). The feedforward networks all fail to attain optimal value, with the non-option cases (V-A-FF and V-FF) repeatedly opening one door due to lack of memory of key color. Notably, the option-based feedforward nets exploit the option index to implicitly remember the key color, leading to higher value. Interplay between explicit memory mechanisms and use of options as pseudo-memory may be an interesting line of future work.

Attention Accelerates Transfer We also evaluate crossmodal attention for transfer learning (Fig. 3b), using the more promising option-based networks. The door puzzle domain is modified to randomize the key position, with pre-trained options from the fixed-position variant used for initialization. All networks benefit from an empirical return jumpstart of nearly 0.2 at the beginning of training, due to the skill transfer. Once again, CASL converges the fastest with statistical significance (e.g., $p = 0.0017$ against V-A-O-LSTM for the t-test with $p < 0.05$), indicating more

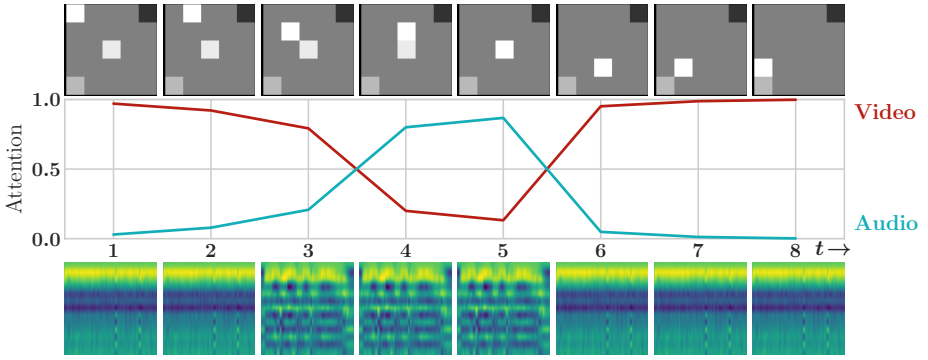


(a) Example episode of the door puzzle game. The agent should open the door 2 (the bottom-left door) corresponding to the lighter key color.



(b) Option trajectory. The agent learns two options sequentially to complete the task, first picking up the key, then navigating to the correct door.

(c) Option termination. Option 1 is terminated upon key pickup, at timestep $t = 4$. Option 2 does not terminate until the agent opens the door.



(d) Top and bottom rows show images and audio spectrogram sequences, respectively, in the door puzzle domain (a). Attention weights $\boldsymbol{\alpha}^t$ are plotted in center.

Fig. 4: Visualizations of option trajectories, option termination, and crossmodal attention. In the example episode shown in (a), the agent should open the door 2. In (b), the agent learns two options, one for picking up the key and the other for opening the correct door. The option termination in (c) is also sparse and only happens once for option 1. In (d), the agent uses visual information but also audio information up to $t = 5$. After picking up the key, the audio attention becomes near zero and the agent mostly relies on the visual information to navigate to the correct door.

effective use of the available audio-video data. While the asymptotic performance of CASL is only slightly higher than the V-A-O-LSTM network, the reduction in number of samples needed to achieve a high score (e.g., after 100K episodes) makes it advantageous for domains with high sampling cost.

Visualizations of Attention-based Option In this section, we visualize the option trajectories, option terminations, and crossmodal attention to understand the options learned within CASL. We trained the agent with two options in the sequential door puzzle game. Recall that the agent in the door puzzle game is positively rewarded by picking up the key and opening the correct door, depending on the type of the randomly generated key.

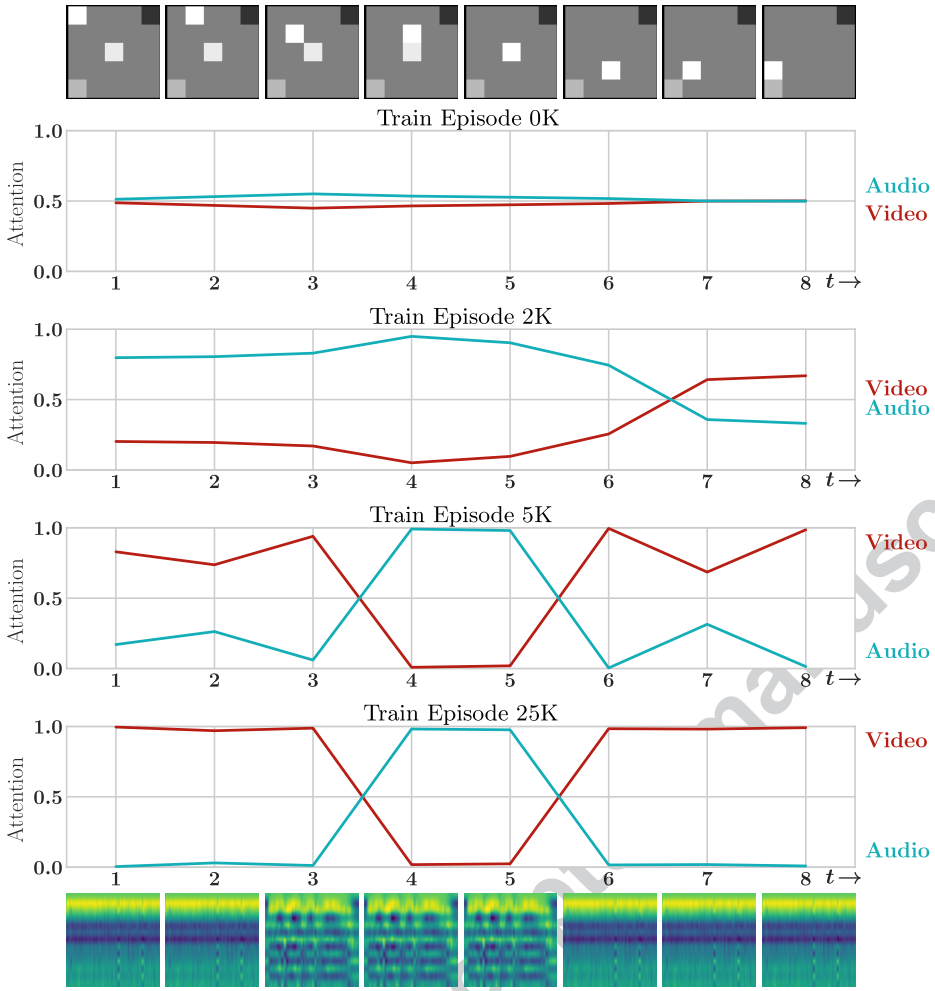
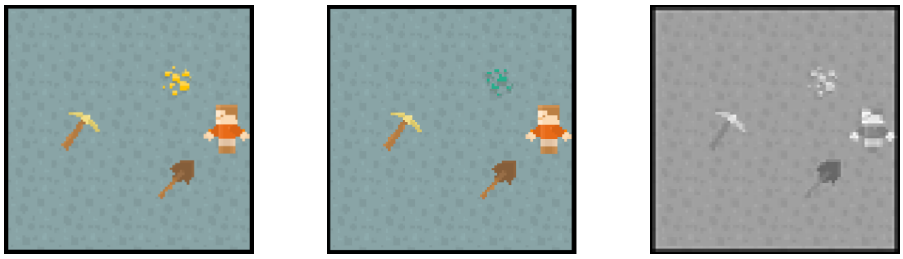


Fig. 5: An example of attention values during training. Initially, the agent pays almost the same amount of attention across the two sensors (i.e., $\boldsymbol{\alpha}^{\boldsymbol{t}} \approx \{0.5, 0.5\}$). Then, the agent explores different attention values during training and converges to the attention values shown at the train episode of 25K, where the agent focuses on the audio information when it is near the key (around $t = 4$).

The option trajectory visualization in Fig. 4b explains what options are learned with CASL: the option 1 is for navigating to the key whereas option 2 is for opening the correct door. Related to the option trajectory, Fig. 4c shows that the agent learned to terminate option 1 and change to option 2 at the moment of picking up the key. Notably, the option termination is sparse: only option 1 terminates and the option 2 does not terminate until the agent opens the door. This observation is consistent with A2OC [16], where the usage of a deliberation cost encourages the agent to learn temporally-extended options with sparse terminations. A closer look at the interactions between attention and sensory modalities in Fig. 4d reveals that



(a) Gold ore should be mined with the pickaxe.

(b) Iron ore should be mined with the shovel.

(c) Ore type indistinguishable by agent's visual input.

Fig. 6: Mining domain, where the agent must pick an appropriate tool (pickaxe or shovel) to mine either (a) gold or (b) iron ore. In its visual input (c), the agent observes identical grayscale sprites for both ore types, but unique audio features when near the ore, making long-term audio storage necessary for selecting the correct tool.

the agent primarily relies on visual information until it reaches the vicinity of the key (around $t = 4$), at which point the attention placed on audio signal increases. Once the key is acquired, audio attention becomes near zero and the agent mostly relies on the visual information to navigate to the correct door. We hypothesize that this capability of CASL that decides relative importance of sensory modalities enables the agent to solve the task using fewer samples than the non-attentive baselines (Fig. 3).

Attention Visualization During Training We include an example of how the attention values evolve during training to provide more comprehensive insight. In particular, we visualize the attention values at different train episodes in the door puzzle domain (see Fig. 5). Initially, the agent pays almost the same amount of attention across the two sensors ($\boldsymbol{\alpha}^t \approx \{0.5, 0.5\}$). Then, the agent explores various attention values during training, such as the values shown at the train episode of 2K in Fig. 5, where the agent unnecessarily pays much attention to the audio data even after acquiring the key. At the train episode 5K, the agent begins to notice that it should focus on the audio information only when it is near the key (around $t = 4$), but the attention values are noisy. Finally, the attention values converge to the attention values shown at the train episode of 25K in Fig. 5.

5.2 Interactions of Attention and Memory

Attention Necessary to Learn in Some Domains Temporal behaviors of the attention mechanism are also evaluated in a 2D mining domain, where the agent must pick an appropriate tool (pickaxe or shovel) to mine either gold or iron ore in a 5×5 gridworld (Figs. 6a to 6c). Critically, the agent observes identical images for both ore types, but unique audio features when the agent is near the ore, making long-term audio storage necessary for selection of the correct tool. The agent has four actions (move up, down, right, left ; $dim(\mathbb{A}) = 4$), and receives $+10$ reward for correct tool selection, -10 for incorrect selection, and -1 step cost. The episode

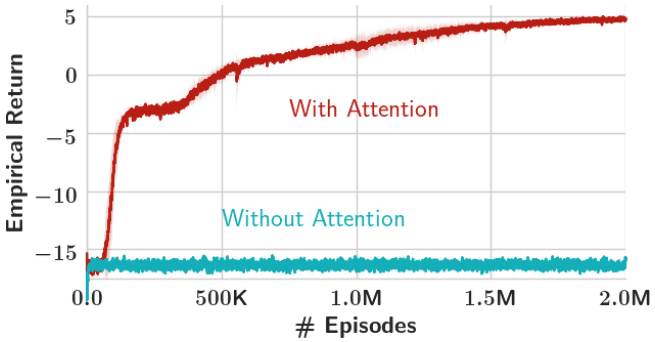


Fig. 7: In the mining domain, the non-attentive network fails to learn, whereas the attentive network succeeds. Mean and 95% confidence interval computed for 5 independent runs are shown in all figures.

terminates upon the selection of either tool or the timestep exceeds a pre-specified value T. Compared to the door puzzle game, the mining domain is posed in such a way that the interplay between audio-video features is emphasized. Specifically, an optimal policy for this task must utilize both audio and video features: visual inputs enable detection of locations of the ore, agent, tools, whereas audio is used to identify the ore type.

Visual occlusion of the ore type, interplay of audio-video features, and sparse positive rewards cause the non-attentive network to fail to learn in the mining domain, as opposed to the attentive case (Fig. 7). Figure 8a plots a sequence of frames where the agent anticipates salient audio features as it nears the ore at $t = 6$, gradually increasing audio attention, then sharply reducing it to 0 after hearing the signal.

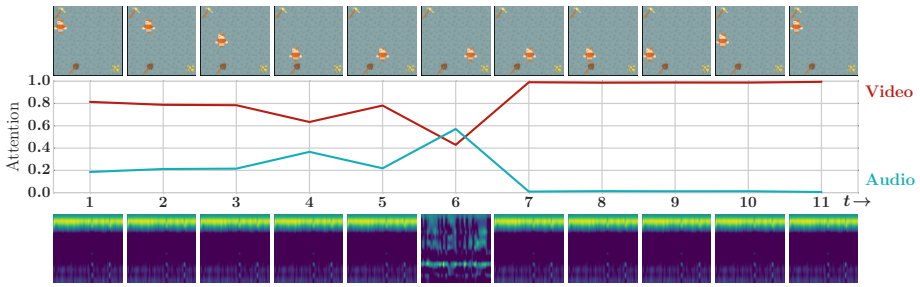
A Closer Look at Attention and Memory While the anticipatory nature of cross-modal attention in the mining domain is interesting, it also points to additional lines of investigation regarding interactions of attention and updates of the agent's internal belief (as encoded in its LSTM cell state). Specifically, one might wonder whether it is necessary for the agent to place any attention on the non-useful audio signals prior to timestep $t = 6$ in Fig. 8a, and also whether this behavior implies inefficient usage of its finite-size memory state.

Motivated by the above concerns, we conduct more detailed analysis of the interplay between the agent's attention and memory mechanisms as used in the CASL architecture (Fig. 1). We first provide a brief overview of LSTM networks to enable more rigorous discussion of these attention-memory interactions. At timestep t, LSTM cell state $\boldsymbol{c^t} \in \mathbb{R}^C$ encodes the agent's memory given its previous stream of inputs, where $C \in \mathbb{N}$ is the cell state dimension. The cell state is updated as follows,

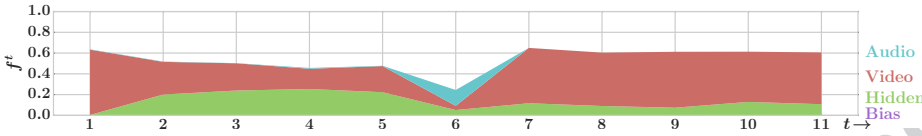
$$\boldsymbol{f^t} = \sigma\left(\boldsymbol{W_f}[\boldsymbol{x}^{\boldsymbol{t}}_{\oplus}, \boldsymbol{h^{t-1}}] + \boldsymbol{b_f}\right), \tag{10}$$

$$\boldsymbol{i^t} = \sigma\left(\boldsymbol{W_i}[\boldsymbol{x}^{\boldsymbol{t}}_{\oplus}, \boldsymbol{h^{t-1}}] + \boldsymbol{b_i}\right), \tag{11}$$

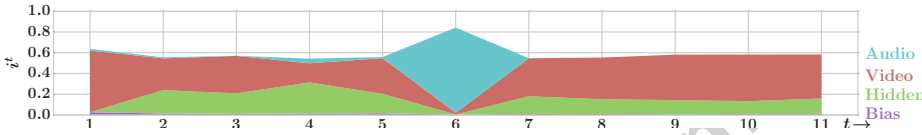
$$\boldsymbol{c^t} = \boldsymbol{f^t} \odot \boldsymbol{c^{t-1}} + \boldsymbol{i^t} \odot \tanh\left(\boldsymbol{W_c}[\boldsymbol{x}^{\boldsymbol{t}}_{\oplus}, \boldsymbol{h^{t-1}}] + \boldsymbol{b_c}\right), \tag{12}$$



(a) Agent anticipates salient audio features as it nears the ore, increasing audio attention until $t = 6$. Audio attention goes to 0 upon storage of ore indicator audio in the LSTM memory. Top and bottom rows show images and audio spectrogram sequences, respectively. Attention weights $\boldsymbol{\alpha^t}$ plotted in center.



(b) Average forget gate activation throughout episode. Recall $f^t = 0$ corresponds to complete forgetting of the previous cell state element.



(c) Average input gate activation throughout episode. Recall $i^t = 1$ corresponds to complete throughput of the corresponding input element.

Fig. 8: Example interaction of crossmodal attention and LSTM memory in the mining domain. The agent first approaches the gold ore (located at the bottom right) to identify the ore type (until $t = 6$), and then navigates to pick up the pickaxe (located at the top left). At $t = 6$, the attended audio input causes forget gate activation to drop, and the input gate activation to increase, indicating major overwriting of memory states. Relative contribution of audio to the LSTM forget and input activations drops to zero after the agent hears the necessary audio signal.

where $\boldsymbol{f^t} \in \mathbb{R}^C$ is the forget gate activation vector, $\boldsymbol{i^t} \in \mathbb{R}^C$ is the input gate activation vector, $\boldsymbol{h^{t-1}}$ is the previous hidden state vector, $\boldsymbol{x^t_\oplus}$ is the attended feature vector, and $\odot$ denotes the Hadamard product. Weights $\boldsymbol{W_f}, \boldsymbol{W_i}, \boldsymbol{W_c} \in \mathbb{R}^{C\times(X+H)}$ and biases $\boldsymbol{b_f}, \boldsymbol{b_i}, \boldsymbol{b_c} \in \mathbb{R}^C$ are trainable parameters. The cell state update in Eq. (12) first forgets certain elements ($\boldsymbol{f^t}$ term), and then adds contributions from new inputs ($\boldsymbol{i^t}$ term). Note that a forget gate activation of 0 corresponds to complete forgetting of the previous cell state element, and that an input gate activation of 1 corresponds to complete throughput of the corresponding input element.

Our goal is to not only analyze the overall forget/input activations throughout the gameplay episode, but also to quantify the relative impact of each contributing variable (audio input, video input, hidden state, and bias term) to the overall activations. Many methods may be used for analysis of the contribution of explana-

tory variables in nonlinear models (i.e., Eqs. (10) to (12)). We introduce a means of quantifying the correlation of each variable with respect to the corresponding activation function. In the following, we focus on the forget gate activation, but the same analysis applies to the input gate. First, expanding the definition of forget gate activation (Eq. (10)) assuming use of concatenated attention (Eq. (6)) yields,

$$\boldsymbol{f^t} = \sigma\left([\boldsymbol{W_{f,a}}, \boldsymbol{W_{f,v}}, \boldsymbol{W_{f,h}}, \boldsymbol{b_f}][\alpha_a \boldsymbol{x_a}, \alpha_v \boldsymbol{x_v}, \boldsymbol{h^{t-1}}, \boldsymbol{I}]\right), \tag{13}$$

where $\boldsymbol{x_a}$ and $\boldsymbol{x_v}$ are the audio and video input features, respectively, weights $\boldsymbol{W_{f,a}}, \boldsymbol{W_{f,v}} \in \mathbb{R}^{C \times X}$ are the forget gate weights for the audio and video input features, respectively, and $\boldsymbol{I}$ are is the identity matrix. Define $\boldsymbol{\hat{f}^t_m}$ as the forget gate activation if the m-th contributing variable were removed. For example, if audio input $\boldsymbol{x_a}$ were to be removed, then,

$$\boldsymbol{\hat{f}^t_a} = \sigma\left([\boldsymbol{W_{f,v}}, \boldsymbol{W_{f,h}}, \boldsymbol{b_f}][\alpha_v \boldsymbol{x_v}, \boldsymbol{h^{t-1}}, \boldsymbol{I}]\right). \tag{14}$$

Define the forget gate activation residual as $\tilde{f}^t_m = \text{avg}(|\boldsymbol{f^t} - \boldsymbol{\hat{f}^t_m}|)$ (i.e., the average difference in output resulting from the removal of the m-th contributing variable). Then, let us define a 'pseudo correlation' of the m-th contributing variable with respect to the true activation,

$$\rho(\tilde{f}^t_m) = \frac{\tilde{f}^t_m}{\sum_a \tilde{f}^t_a}. \tag{15}$$

This provides an approximate quantification of the relative contribution of the m-th variable (audio input, video input, hidden unit, or bias) to the overall activation of the forget and input gates.

Armed with this toolset, we now analyze the interplay between attention and LSTM memory. First, given the sequence of audio-video inputs in Fig. 8a, we plot overall activations of the forget and input LSTM gates (averaged across all cell state elements), in Fig. 8b and Fig. 8c, respectively. Critically, these plots also indicate the relative influence of each gate's contributing variables to the overall activation, as measured by Eq. (15).

Interestingly, prior to timestep $t = 6$, the contribution of audio to the forget gate and input gates is essentially zero, despite the positive attention on audio (in Fig. 8a). At $t = 6$, the forget gate activation drops suddenly, while the input gate experiences a sudden increase, indicating major overwriting of previous memory states with new information. The plots indicate that the attended audio input is the key contributing factor of both behaviors. In Fig. 8a, after the agent hears the necessary audio signal, it moves attention entirely to video; the contribution of audio to the forget and input activations also drops to zero. Overall, this analysis indicates that the agent attends to audio in anticipation of an upcoming pertinent signal, but *chooses* not to embed it into memory until the appropriate moment. Attention filters irrelevant sensor modalities, given the contextual clues provided by exogeneous and endogeneous input features; it, therefore, enables the LSTM gates to focus on learning when and how to update the agent's internal state.

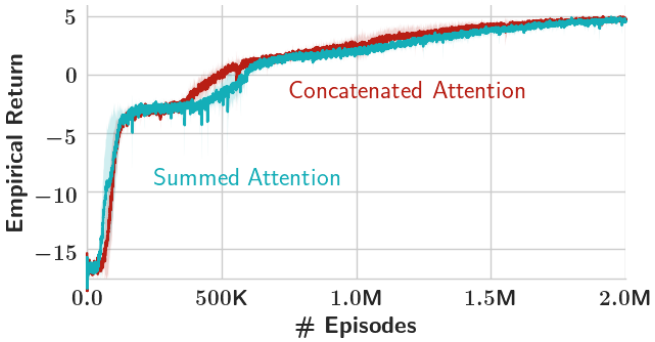


Fig. 9: Comparisons between the summed and concatenated attention, showing that the concatenated attention performs slightly better than the summed attention approach. Mean and 95% confidence interval computed for 5 independent runs are shown in all figures.

5.3 Summed vs. Concatenated Attention

Equation (6) explains two possible choices for combining the attended features $\alpha_m^t \boldsymbol{x_m^t}$: either by summation $\boldsymbol{x_\oplus^t} = \sum_{m=1}^{M} \alpha_m^t \boldsymbol{x_m^t}$ or by concatenation $\boldsymbol{x_\oplus^t} = \left[(\alpha_1^t \boldsymbol{x_1^t})^T, \ldots, (\alpha_M^t \boldsymbol{x_M^t})^T\right]^T$. In this section, we conduct an empirical analysis of these two possibilities, and to clarify, except for this section, all the results in this paper are based on concatenated attention. Figure 9 compares two versions of CASL in the mining domain, one with summed attention and the other with concatenated attention. Interestingly, the t-test result with $p < 0.05$ based on mean/std of AUC shows that the concatenated attention has slightly better performance than the summed attention ($p = 0.0383$). However, given the comparable performance, one might prefer the summed attention as it reduces the input size to the LSTM and thus saves computational resources as compared to the concatenated attention, whose size increases linearly in the number of sensory modalities.

5.4 The Arcade Learning Environment

Preliminary evaluation of crossmodal attention was also conducted in the Arcade Learning Environment (ALE) [8]. We modified ALE to support audio queries, as it previously did not have this feature.

Experiments were conducted in the Atari 2600 games H.E.R.O. and Amidar (Table 1). This line of investigation considers impacts of crossmodal attention on Atari agent behavior, even without use of multiple (hierarchical) options; these results use CASL with a *single* option, hence tagged "no-options" in the table.[3] Amidar was one of the games in which Deep Q-Networks failed to exceed human-level performance [32]. The objective in Amidar is to collect rewards in a rectilinear maze while avoiding patrolling enemies. The agent is rewarded for

[3] Note that HRL with a single option is equivalent to a normal RL with primitive actions. Therefore, a single option CASL corresponds to A3C [30] but with the crossmodal attention mechanism.

Table 1: Preliminary results for learning in Atari 2600 games. The crossmodal attention learner, even *without* options, achieves high score for non-hierarchical methods. We emphasize these are not direct comparisons due to our method leveraging additional sensory inputs, but are meant to highlight the performance benefits of crossmodal learning.

Algorithm	Sensory Inputs	Amidar Score	H.E.R.O. Score
DQN [32]	Video	740	19950
A3C [30]	Video	284	28766
GA3C [5]	Video	218	–
Ours (*no* options)	Audio & Video	900	32985

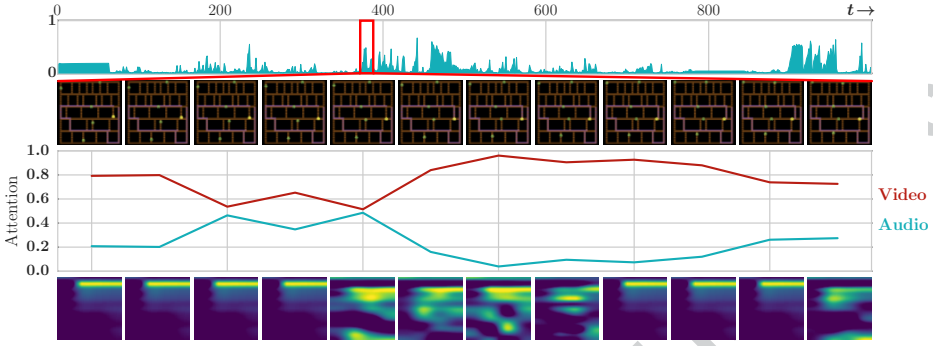


Fig. 10: Example interaction of crossmodal attention in Amidar, where pathway vertices critical for avoiding enemies make an audible sound if not previously crossed. The agent anticipates and increases audio attention when near these vertices. Top row shows audio attention over 1000 frames, with audio/video/attention frames highlighted for zoomed-in region of interest.

painting segments of the maze, killing enemies at opportune moments, or collecting bonuses. Background audio plays throughout the game, and specific audio signals play when the agent crosses previously-unseen segment vertices. Figure 10 reveals that the agent anticipates and increases audio attention when near these critical vertices, which are especially difficult to observe when the agent sprite is overlapping them (e.g., zoom into video sequences of Fig. 10).

Our crossmodal attentive agent (no-options) achieves a mean score of 900 in Amidar, over 30 test runs, outperforming the other non-hierarchical methods. A similar result is achieved for the game H.E.R.O., where our agent beats other non-hierarchical agents. Also, our agent trained using 3 options achieved a mean score of 1175 in Amidar, over 30 test runs. Note that this score is higher than our CASL agent *without* options. The score is also higher than our underlying hierarchical method (A2OC)'s score of 880. We emphasize these are not direct comparisons due to our method leveraging additional sensory inputs, but are meant to highlight the performance benefits of crossmodal learning. We also note that the state-of-the-art hierarchical approach FeUdal [44] beats our agent's score, the

future investigation of the combination of audio-video attention with their approach may be of interest.

5.5 Notes on Scalability

CASL may require a large number of train episodes to converge. We note that related deep learning approaches, such as GA3C [5], A3C [30], and DQN [32], share similar computation cost. In general, the computation cost is also affected by the reward sparsity. Due to the sparse feedback in the door puzzle (Fig. 2) and mining (Fig. 6) domains, the agent is required to perform a sufficient exploration, making the domain inherently difficult to solve and resulting in an increased number of train episodes. Additionally, we train deep neural networks with a large number of train parameters from scratch for each experiment, resulting in the increased computation cost. In problems with larger input dimensions and higher problem complexity, the high computation cost can be mitigated by the use of widely employed practical methods in deep learning literature. For instance, denser reward functions can be used to ease the learning complexity and reduce the computation cost [1]. The time for learning the feature extraction (e.g., CNNs in Fig. 1) can be also reduced by fine-tuning from pre-trained models instead of learning from randomly initialized weights [18].

5.6 Experimental Details

We summarize additional details and hyperparameters in our experiments in this section.

Domains Regarding the sound input in the door puzzle domain, the agent hears the key-specific sound when adjacent to the key. Specifically, if the Euclidean distance between the agent and key location is less than 1.5, then the agent hears the sound dependent on the key type. Otherwise, the agent hears the noise. Similarly, the agent in the 2D mining domain hears the sound dependent on the ore type if the Euclidean distance between the agent and ore location is less than 1.5. Otherwise, the agent hears the noise. Regarding the maximum timestep T, both the door puzzle and mining domain use $T = 30$. Additionally, the door puzzle domain has a transition noise of 0.2 so that the agent moves in a random direction every timestep with 0.2 probability. Lastly, in ALE, we clip reward values between -1 and 1 to stabilize learning.

Network Architecture The sensory feature extractor used in all experiments consists of 3 convolutional layers, each with 32 filters of size 3×3, stride 2, and ReLU activations. The input to each convolutional neural network is a 84×84 re-scaled, gray-scale data. The output of the convolutional layers for each sensor is then flattened, which has the dimension X of 3872. Regarding the LSTM, we use a single layer LSTM with 16 cells $(H = C = 16)$ in the door puzzle domain, and 128 cells $(H = C = 128)$ in the other domains.

Audio Conversion The audio sensory data is represented visually so that the convolutional layers can be applied to extract features. Specifically, given an audio signal, we convert it into the spectrogram by computing Mel Frequency Cepstral Coefficients (MFCCs) based on the signal's sample rate. We use the default parameters for computing MFCC features (e.g., the window length of 0.025s, window step of 0.01s, cepstrum number of 13, filter number of 26, and fast Fourier transform size of 512) in the door puzzle and 2D mining domain. For ALE domains, we use the same parameters, except the window length of 0.01s and window step of 0.003s with the audio frequency of 30720Hz.

Hyper-Parameters In all experiments, we train the agent with 32 parallel CPU threads, the discount factor γ of 0.99, either every 5 or 10 timesteps for the asynchronous update, the entropy regularization of 0.01, the attention regularization of 0.001, and the Adam optimizer with a learning rate of 0.0001. Regarding the dimension of the internal embedding feature vector, we use the the same value of the cell state's dimension for simplicity (i.e., $Z = C$). For the option-based approaches (including CASL), we use 2 options ($dim(\Omega) = 2$) with the deliberation cost of 0.001, the margin cost of 0.0002, and the option epsilon of 0.15 in the door puzzle domain, and 3 options ($dim(\Omega) = 3$) with the deliberation cost of 0.03, the margin cost of 0.0002, and the option epsilon of 0.10 in Amidar (with options experiment).

6 Contribution

This work introduced the Crossmodal Attentive Skill Learner (CASL), integrated with the recently-introduced Asynchronous Advantage Option-Critic (A2OC) architecture [16] to enable hierarchical reinforcement learning across *multiple* sensory inputs. We provided concrete examples where CASL not only improves performance in a single task, but accelerates transfer to new tasks. We demonstrated the learned attention mechanism anticipates and identifies useful sensory features, while filtering irrelevant sensor modalities during execution. We modified the Arcade Learning Environment [8] to support audio queries, and evaluations of crossmodal learning were conducted in the Atari 2600 games H.E.R.O. and Amidar. Finally, building on the recent work of Babaeizadeh et al. [5], we open-source a fast hybrid CPU-GPU implementation of CASL. This investigation indicates crossmodal skill learning as a promising avenue for future works in HRL that target domains with high-dimensional, multimodal inputs.

References

1. Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. (2018). Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations (ICLR)*.
2. Alvis, C. D., Ott, L., and Ramos, F. (2017). Online learning for scene segmentation with laser-constrained CRFs. In *International Conference on Robotics and Automation (ICRA)*, pages 4639–4643.

3. Andreas, J., Klein, D., and Levine, S. (2016). Modular multitask reinforcement learning with policy sketches. *arXiv preprint arXiv:1611.01796*.
4. Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.
5. Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., and Kautz, J. (2017). Reinforcement learning through asynchronous advantage actor-critic on a GPU. In *International Conference on Learning Representations (ICLR)*.
6. Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
7. Beal, M. J., Attias, H., and Jojic, N. (2002). Audio-video sensor fusion with probabilistic graphical models. In *European Conference on Computer Vision (ECCV)*, pages 736–750.
8. Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
9. Bengio, S. (2002). An asynchronous hidden markov model for audio-visual speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1237–1244.
10. Cadena, C. and Košecká, J. (2014). Semantic segmentation with heterogeneous sensor coverages. In *International Conference on Robotics and Automation (ICRA)*, pages 2639–2645.
11. Caglayan, O., Barrault, L., and Bougares, F. (2016). Multimodal attention for neural machine translation. *arXiv preprint arXiv:1609.03976*.
12. Carrasco, M. (2011). Visual attention: The past 25 years. *Vision research*, 51(13):1484–1525.
13. Chambers, A. D., Scherer, S., Yoder, L., Jain, S., Nuske, S. T., and Singh, S. (2014). Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments. In *American Control Conference (ACC)*.
14. Da Silva, B., Konidaris, G., and Barto, A. (2012). Learning parameterized skills. *arXiv preprint arXiv:1206.6398*.
15. Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M., and Burgard, W. (2015). Multimodal deep learning for robust RGB-D object recognition. In *International Conference on Intelligent Robots and Systems (IROS)*.
16. Harb, J., Bacon, P.-L., Klissarov, M., and Precup, D. (2017). When waiting is not an option: Learning options with a deliberation cost. *arXiv preprint arXiv:1709.04571*.
17. Hausknecht, M. and Stone, P. (2015). Deep recurrent Q-learning for partially observable MDPs. *arXiv preprint arXiv:1507.06527*.
18. He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.
19. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
20. Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134.
21. Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

22. Konidaris, G. and Barto, A. G. (2009). Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1015–1023.
23. Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3675–3683.
24. Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pages 282–289.
25. Leong, Y. C., Radulescu, A., Daniel, R., DeWoskin, V., and Niv, Y. (2017). Dynamic interaction between reinforcement learning and attention in multidimensional environments. *Neuron*, 93(2):451–463.
26. Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
27. Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., and Siegwart, R. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3923–3929.
28. Machado, M. C., Bellemare, M. G., and Bowling, M. (2017). A laplacian framework for option discovery in reinforcement learning. *arXiv preprint arXiv:1703.00956*.
29. Mackintosh, N. J. (1975). A theory of attention: Variations in the associability of stimuli with reinforcement. *Psychological Review*, 82(4):276.
30. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1928–1937.
31. Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2204–2212.
32. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
33. Nachum, O., Gu, S. S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3306–3317.
34. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, pages 689–696.
35. Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., and Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience*, 35(21):8145–8157.
36. Nobili, S., Camurri, M., Barasuol, V., Focchi, M., Caldwell, D. G., Semini, C., and Fallon, M. (2017). Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots. In *Robotics: Science and Systems (RSS)*.
37. Pearce, J. M. and Hall, G. (1980). A model for pavlovian learning: Variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological*

Review, 87(6):532.
38. Pearce, J. M. and Mackintosh, N. J. (2010). Two theories of attention: A review and a possible integration. *Attention and Associative Learning: From Brain to Behaviour*, pages 11–39.
39. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.
40. Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A., and Ignateva, A. (2015). Deep attention recurrent Q-network. *arXiv preprint arXiv:1512.01693*.
41. Srivastava, N. and Salakhutdinov, R. (2014). Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980.
42. Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*.
43. Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.
44. Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1703.01161*.
45. Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015). Grammar as a foreign language. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2773–2781.
46. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. (2016). Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1480–1489.
47. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., and Fei-Fei, L. (2015). Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, pages 1–15.