Master’s Degree in Computer Science
Curriculum: Artificial Intelligence

Adaptively Combining Skill Embeddings for Reinforcement Learning Agents

Supervisors:
Prof. Davide Bacciu
Dott. Elia Piccoli

Examiner:
Prof. Marco Podda

Candidate:
Giacomo Carfì

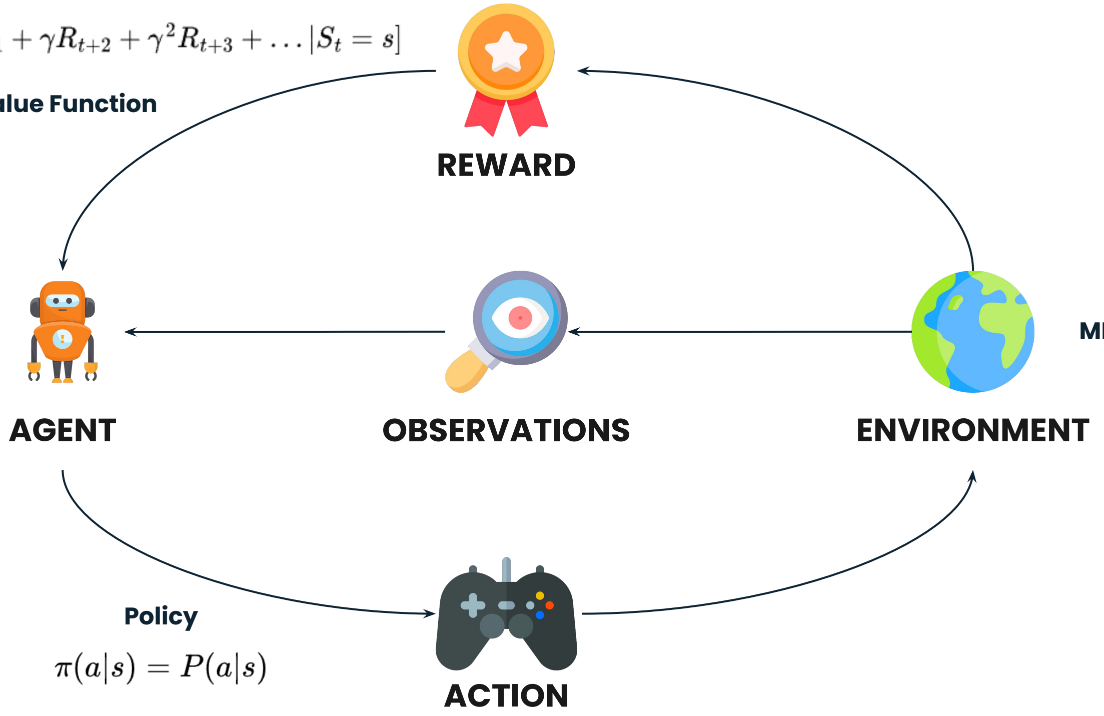
Academic Year: 2023-2024

01. REINFORCEMENT LEARNING

INTRODUCTION

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

Value Function



MDP = (S, A, P, R, γ)

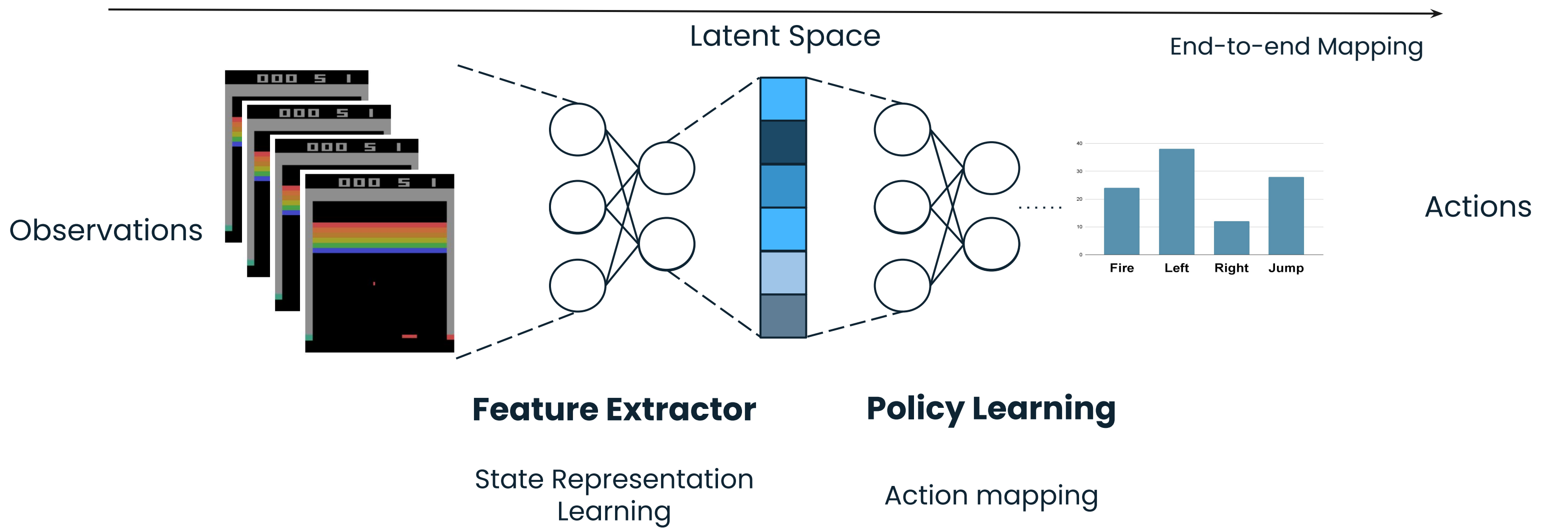
Policy

$$\pi(a|s) = P(a|s)$$

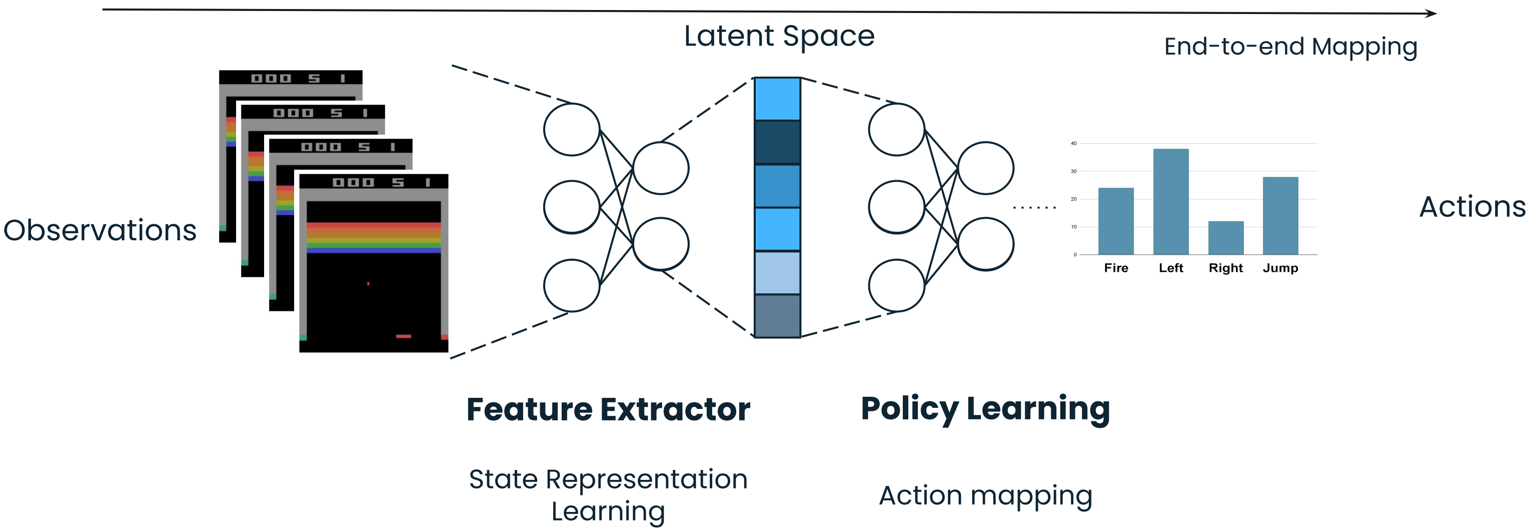
Goal Find a policy π that maximizes the **Value Function**

INTRODUCTION

01. PROBLEM FORMULATION



01. PROBLEM FORMULATION



Problem:

- State representation learning is **executed from scratch** each time for each new task.
- **No re-use** of previously learned knowledge.
- **No composition** of different knowledge.
- Agent's focus should be on **policy learning.**

01. PROBLEM FORMULATION

I N T R O D U C T I O N

SKILLS DEFINITION

- How can we **represent prior knowledge** for an RL agent to **simplify** the **state representation learning** and in order to **re-use** already trained abilities?

COMBINATION

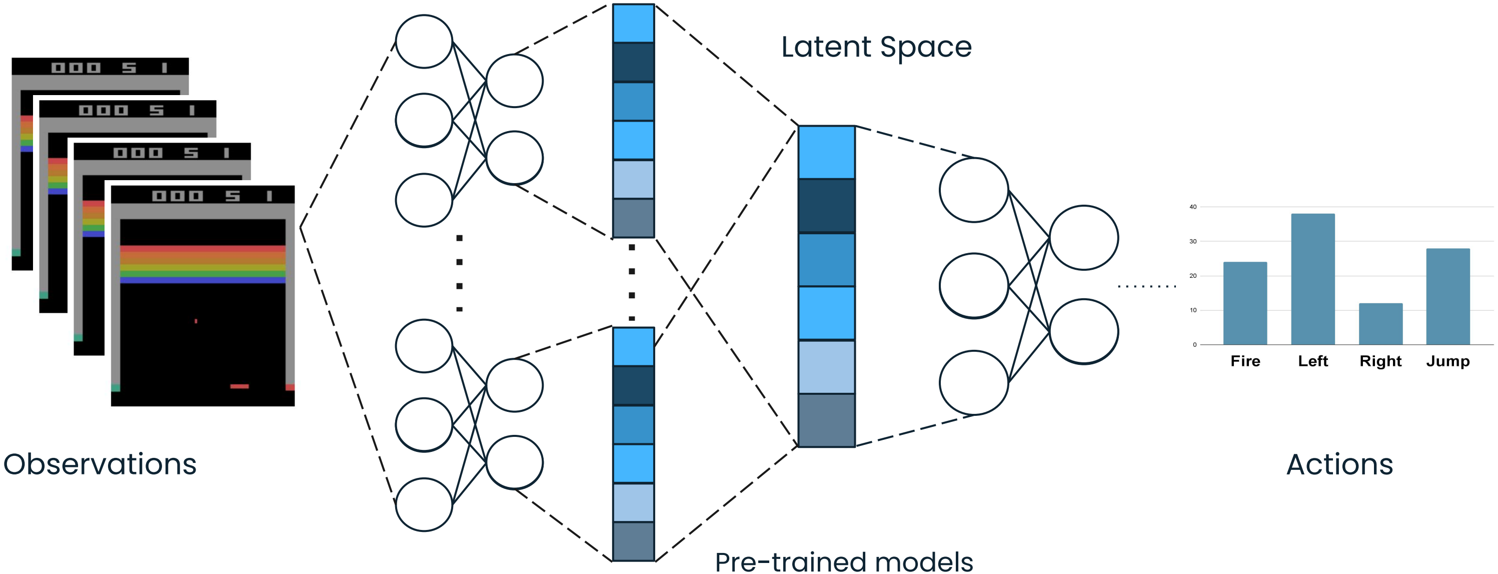
- How can we **combine** different information coming from different skills or **choose** the best one to achieve agent’s goals?

02. SKILLS DEFINITION

Foundational Models for Skills Representation

- Self-supervised or Unsupervised models.
- Trained on huge heterogeneous dataset.
- Extract hidden patterns.
- Fine-tuned on specific tasks.

02. GENERAL ARCHITECTURE



Pre-trained models provide different perspectives of the current state

Feature Extractor

State Representation Learning

Policy Learning

Action mapping

This provides the **re-use** of prior skills.

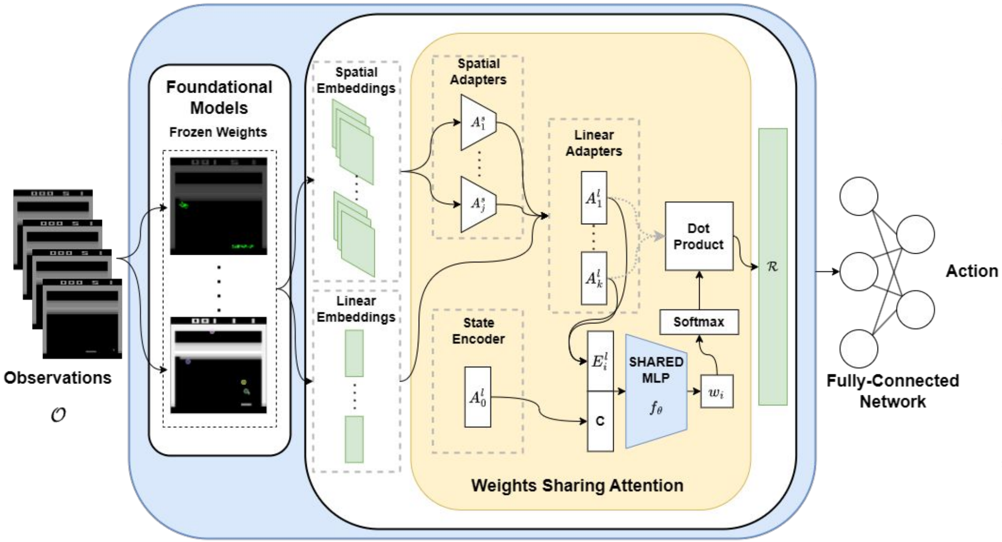
02. COMBINATION MODULES

Combining various pre-trained environment state representations

- Handle various type of information.
- Capable of capturing the most relevant information from each individual skill.
- Fast in gathering information and mixing it.

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



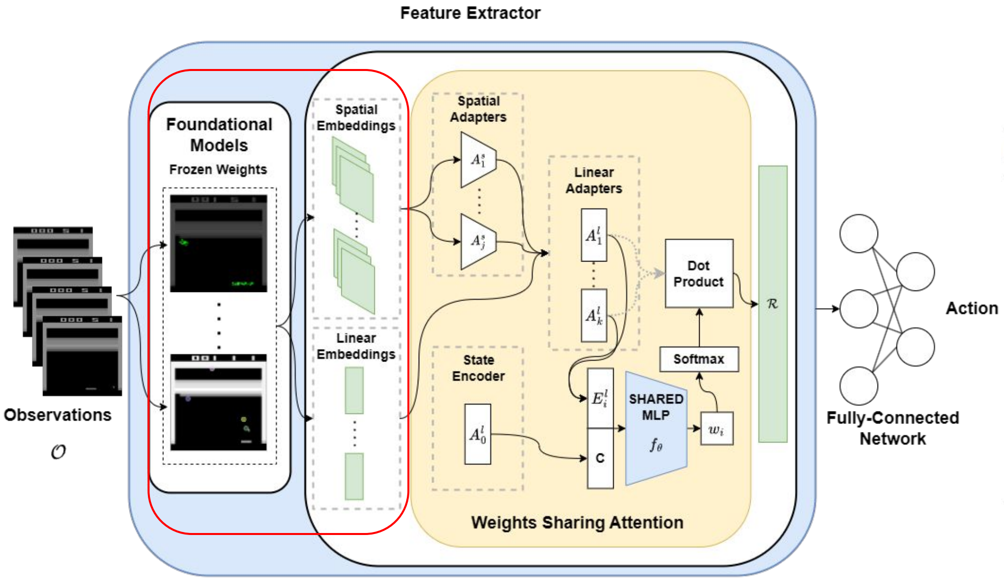
Combination Module in Yellow

Algorithm 5 Weight Sharing Attention

$\mathcal{C} = A_0(\mathcal{E}(\mathcal{O}))$
for FM ψ in Ψ **do**
  $x = \psi_i(\mathcal{O})$
  $E_i = A_i(x)$
  $w_i = f_\theta(\mathcal{C}, E_i)$
end for
$\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i$

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



Combination Module in Yellow

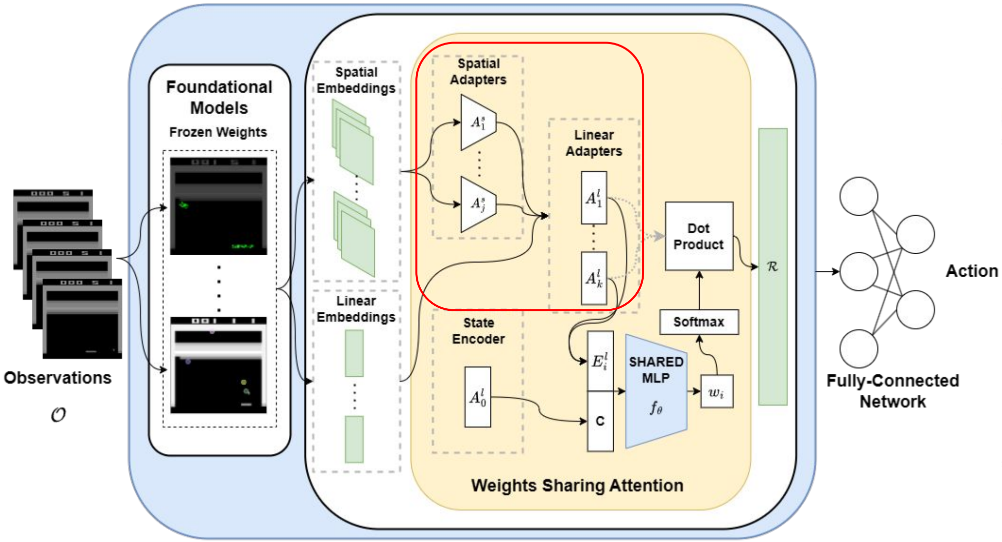
Algorithm 5 Weight Sharing Attention

$\mathcal{C} = A_0(\mathcal{E}(\mathcal{O}))$
for FM ψ in Ψ **do**
    $x = \psi_i(\mathcal{O})$
    $E_i = A_i(x)$
    $w_i = f_\theta(\mathcal{C}, E_i)$
end for
$\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i$

METHODOLOGY

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



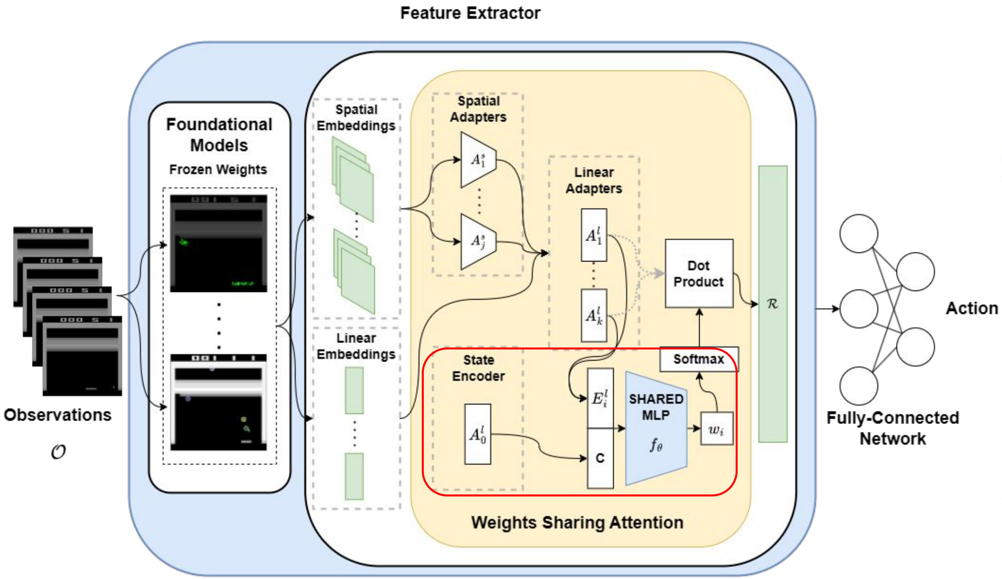
Combination Module in Yellow

Algorithm 5 Weight Sharing Attention

1: $\mathcal{C} = \mathcal{A}_0(\mathcal{E}(\mathcal{O}))$
2: **for** FM ψ in Ψ **do**
3: $\quad x = \psi_i(\mathcal{O})$
4: $\quad \boxed{E_i = \mathcal{A}_i(x)}$
5: $\quad w_i = f_\theta(\mathcal{C}, E_i)$
6: **end for**
7: $\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i$

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



Combination Module in Yellow

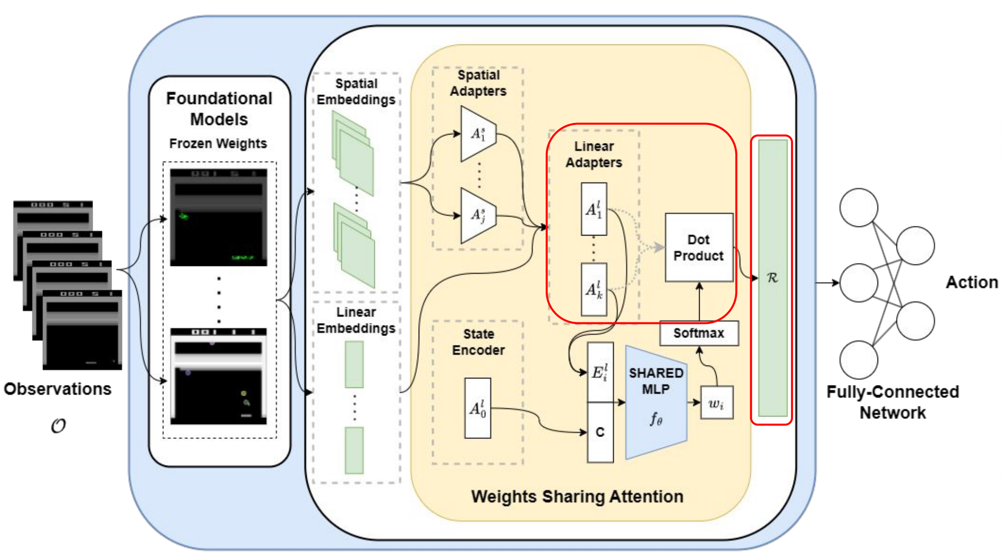
Algorithm 5 Weight Sharing Attention

$\mathcal{C} = A_0(\mathcal{E}(\mathcal{O}))$
for FM ψ in Ψ **do**
  $x = \psi_i(\mathcal{O})$
  $E_i = A_i(x)$
  $w_i = f_\theta(\mathcal{C}, E_i)$
end for
$\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i$

M E T H O D O L O G Y

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



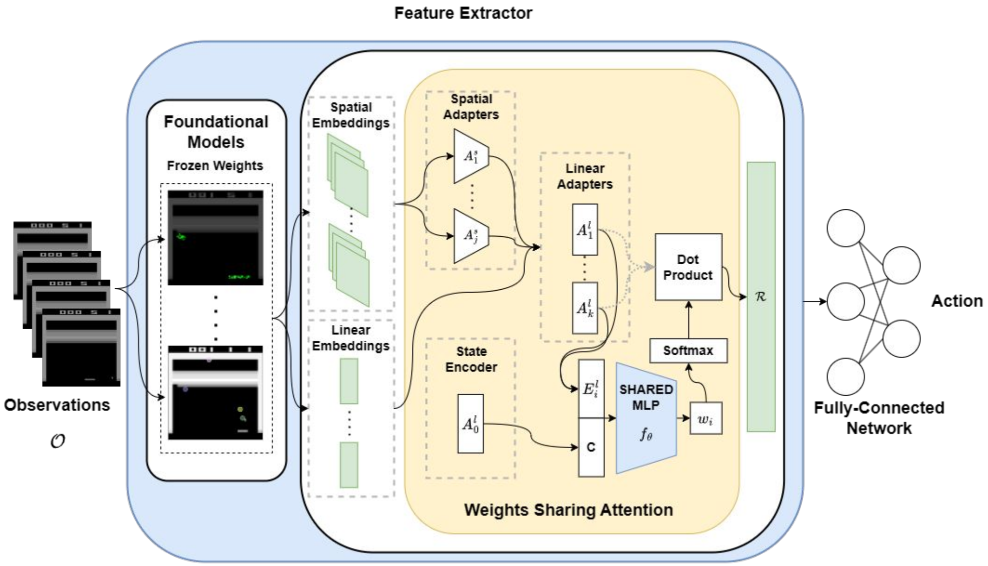
Combination Module in Yellow

Algorithm 5 Weight Sharing Attention

$\mathcal{C} = A_0(\mathcal{E}(\mathcal{O}))$
for FM ψ in Ψ **do**
  $x = \psi_i(\mathcal{O})$
  $E_i = A_i(x)$
  $w_i = f_\theta(\mathcal{C}, E_i)$
end for
$\boxed{\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i}$

02. COMBINATION MODULES

Weight Sharing Attention (WSA)



Combination Module in Yellow

PRO

- Combines encodings of different types.
- Can be scaled to an arbitrary number of skills.
- Provides explainability.

02. COMBINATION MODULES

Other Combination Modules

Combination Type	Configuration
Linear Combination (LIN)	$\mathcal{R} = E_1 \oplus E_2 \oplus, \ldots, \oplus E_k$
Fixed Linear Combination (FIX)	$\mathcal{R} = E_1 \oplus E_2 \oplus, \ldots, \oplus E_k$
Convolutional Combination (CNN)	$\mathcal{R} = conv(E_1 \oplus, \ldots, \oplus E_k)$
Mixed Combination (MIX)	$\mathcal{R} = E_1^l \oplus, \ldots, \oplus E_p^l \oplus conv(E_1^s \oplus, \ldots, \oplus E_q^s)$
Reservoir Combination (RES)	$IN = E \times \mathbf{W}_{\mathbf{in}}$ $H = IN \times \mathbf{W}_{\mathbf{res}}$ $\mathcal{R} = tanh(IN + H)$
DotProduct Attention (DPA)	$\mathbf{W} = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$ $\mathcal{R} = \sum_{i=0}^{\lvert\Psi\rvert} w_i * E_i$

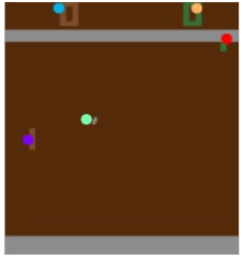
METHODOLOGY

03. SETUP

E X P E R I M E N T S

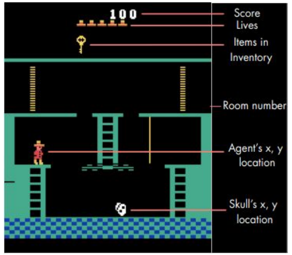
Skill Selection

Kulkarni et al. (2019)



Object Keypoints Detection
(**OKK, OKE**)

Anand et al. (2019)



State Representation
(**SR**)

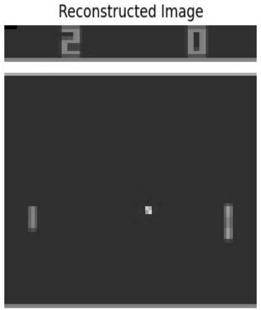
Goel et al. (2018)

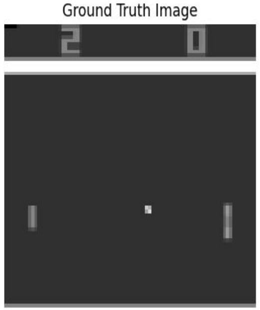
Video Object Segmentation
(**VOS**)

State Encoder

Inspired by *Nature CNN*

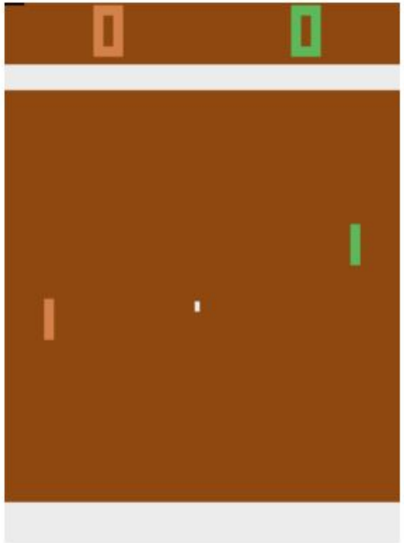
Mnih et al. (2015)



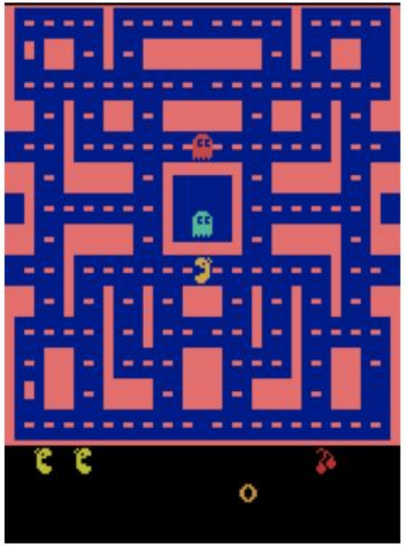


03. SETUP

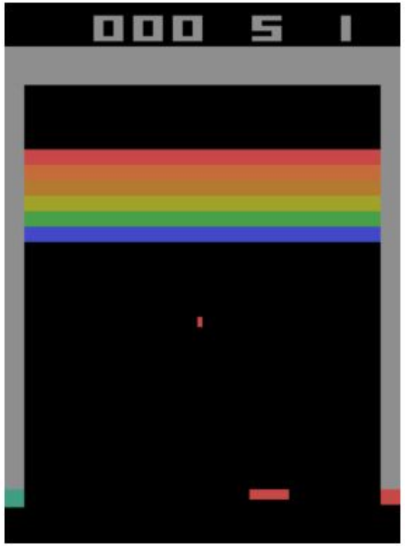
Environments



Pong



Ms. Pacman



Breakout

Discrete action space and observations

03. SETUP

SKILLS PRE-TRAINING

- Creation of the dataset using a random agent collecting **1M** frames per game.

AGENTS TRAINING

- Skill weights are **frozen** during agents' training.
- Max **10M** steps in training.
- Evaluation each 40.000 steps for **100 episodes.**
- **No hyperparameters search** for agents with skills.

03. PRELIMINARY STUDY

- Tested all the combination modules.
- Single layer with **256** units for Policy Learning network.
- Agents are trained using **early stopping** for those who show no improvement for **5 consecutive evaluations.**
- **PPO** as learning algorithm.

Feature Extractor	Configuration
LIN	-
FIX	256, 512, 1024
CNN	1, 2, 3
MIX	-
RES	512, 1024, 2048
DPA	256, 512, 1024
WSA	256, 512, 1024

EXPERIMENTS

03. TOP 3 PERFORMER

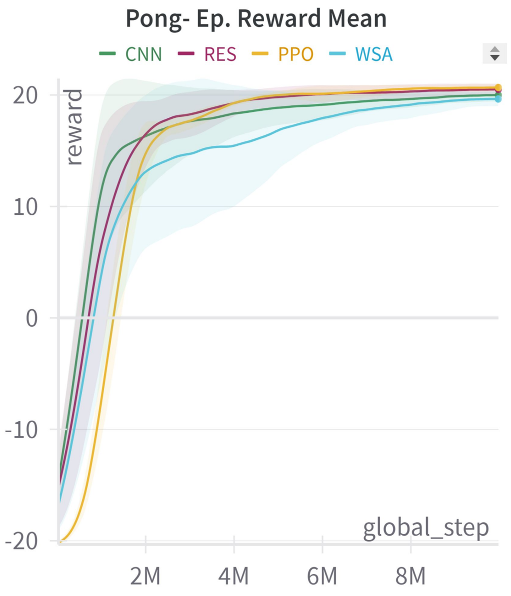
- Tested only the **top 3** combination modules.
- Single layer with **256** units for *Fully-Connected* network.
- **No early stopping**, agents trained for full 10M steps.
- Training results are averaged across **4 runs** per combination module per game. **Seeds are fixed.**
- Compared with an end-to-end **PPO** using already-tuned hyperparameters.

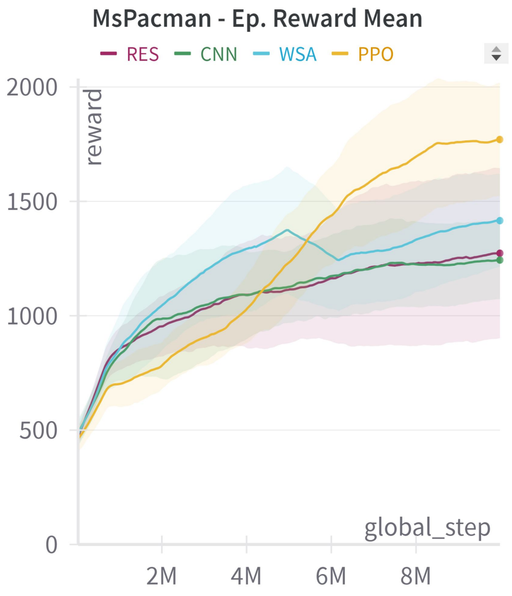
Environment	Configuration
Pong	WSA (1024) RES (1024) CNN (2)
Ms. Pacman	WSA (256) RES (1024) CNN (2)
Breakout	WSA (256) FIX (512) CNN (3)

03. TOP 3 PERFORMER

EXPERIMENTS

Learning Curves during Training





WSA (and others) is better than PPO in the early stages.

03. TOP 3 PERFORMER

E X P E R I M E N T S

Evaluation

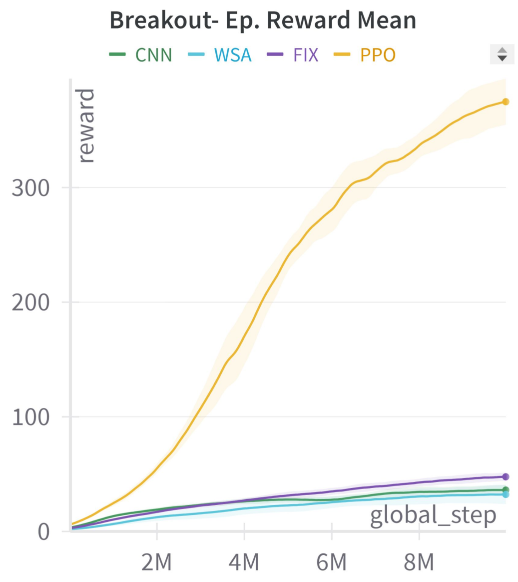
Environment	Agent	Reward
Pong	**WSA** RES **CNN** **PPO**	**21 ± 0.00** 20.85 ± 0.29 **21 ± 0.00** **21 ± 0.00**
Ms. Pacman	**WSA** RES CNN PPO	**2530.20 ± 23.09** 1369.27 ± 565.23 1801.30 ± 20.95 2258.40 ± 1.42

- Equivalent results in Pong.
- WSA is better than PPO in Ms. Pacman - better generalization.

03. TOP 3 PERFORMER

E X P E R I M E N T S

Learning Curves during Training



Not good results

03. BREAKOUT: OUT OF DISTRIBUTION DATA

Hypothesis - Underfitting Problems

- Only some components of the agents are updated during training.
- Policy learning network too small to capture relevant information.

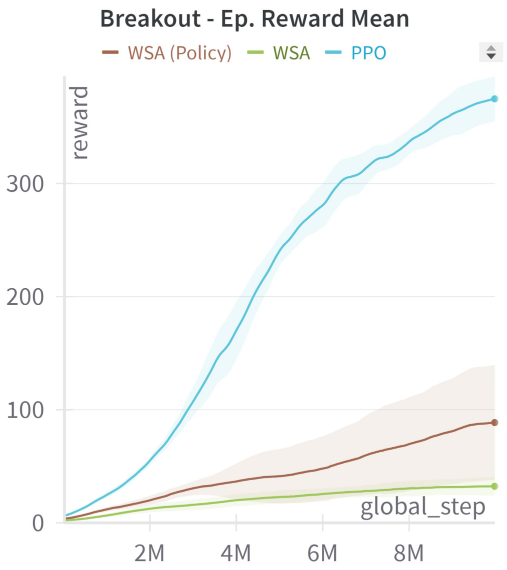
New Experiments increasing Policy Network

From single layer of **256** units
to three layers of **1024, 512, 256** units respectively
Using **ReLU** activation function.

03. BREAKOUT: OUT OF DISTRIBUTION DATA

EXPERIMENTS

Hypothesis – Underfitting Problems



Little improvements for WSA in training.

03. BREAKOUT: OUT OF DISTRIBUTION DATA

Hypothesis – Distributional Shift

- First stages of the game the agent can focus on just bounce the ball back.
- Late game stages, agents needs to be more precise.
- Training dataset is missing of late game scenarios.
- Misleading Skills.

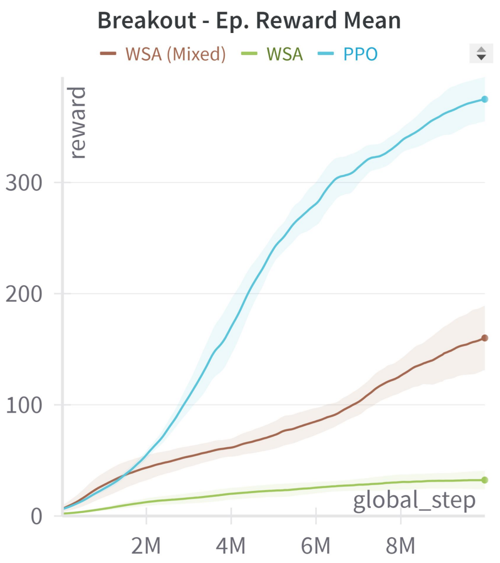
New Experiments retraining the skills on mixed data

We collected new dataset using both a **random** agent and an **expert** agents that plays Breakout to obtain early and late game scenarios.

03. BREAKOUT: OUT OF DISTRIBUTION DATA

EXPERIMENTS

Hypothesis - Distributional Shift

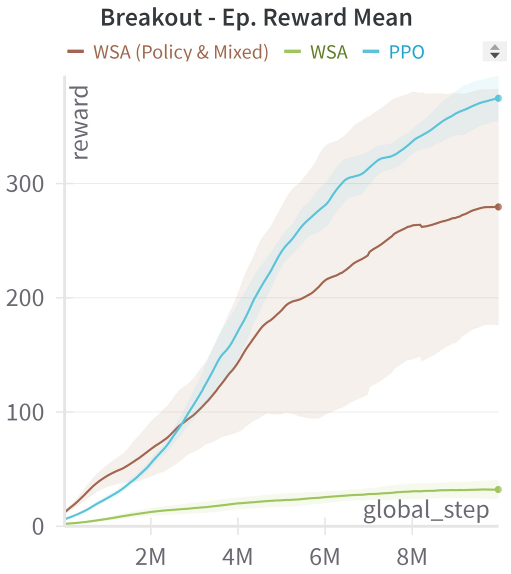


WSA achieved the biggest jump in performance.

03. BREAKOUT: OUT OF DISTRIBUTION DATA

EXPERIMENTS

Combination of increased policy network and using mixed data



WSA achieved the best performance.

03. BREAKOUT: OUT OF DISTRIBUTION DATA

Evaluation

Starting Point

Agent	Reward
WSA FIX CNN	99.58 ± 6.66 87.17 ± 6.87 65.98 ± 1.62

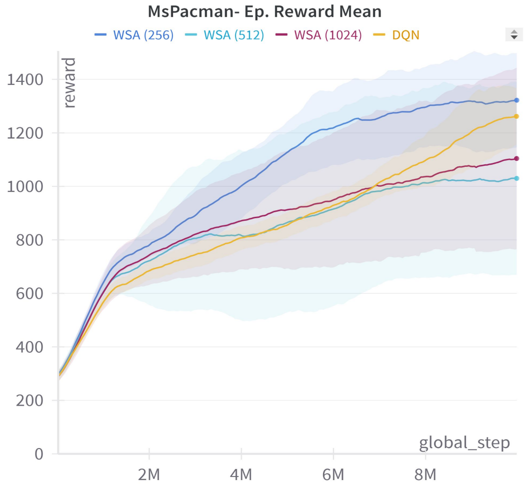
Results

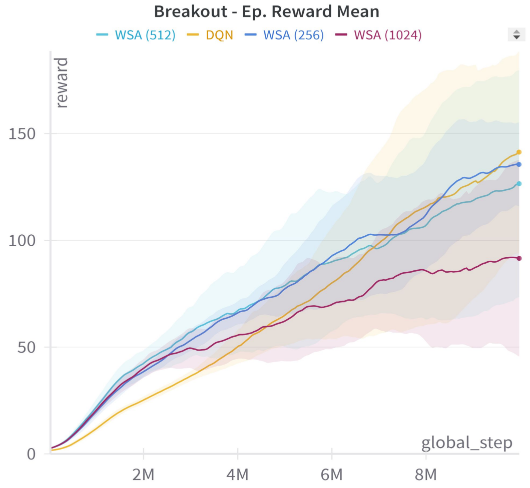
Strategy	Agent	Reward
Policy & Mixed	WSA FIX CNN	387.15 ± 0.43 71.06 ± 5.04 68.51 ± 1.85
	PPO	**413.51 ± 1.10**

03. DEEP Q-LEARNING TESTS

E X P E R I M E N T S

Training





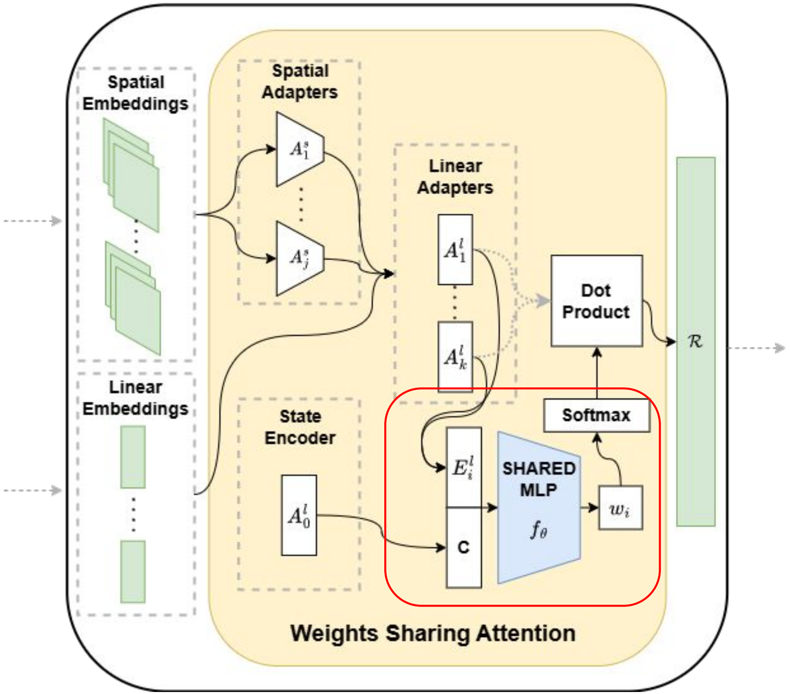
Evaluation

Environment	Agent	Reward
Ms. Pacman	**WSA** (256) DQL	**2047.27 ± 231.18** 1701.00 ± 490.41
Breakout	**WSA** (256) DQL	**213.14 ± 39.37** 166.65 ± 20.19

03. WSA EXPLAINABILITY

What skills does the agent use in different situations?

- Assigns different weights to skills.
- Analyze them in test phase to understand which skills are most important in specific contexts.

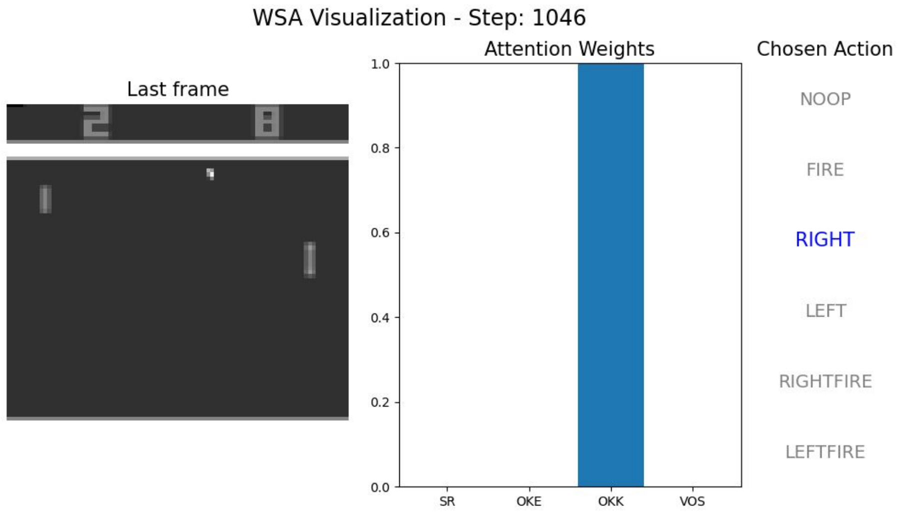


EXPERIMENTS

03. WSA EXPLAINABILITY

EXPERIMENTS

Pong

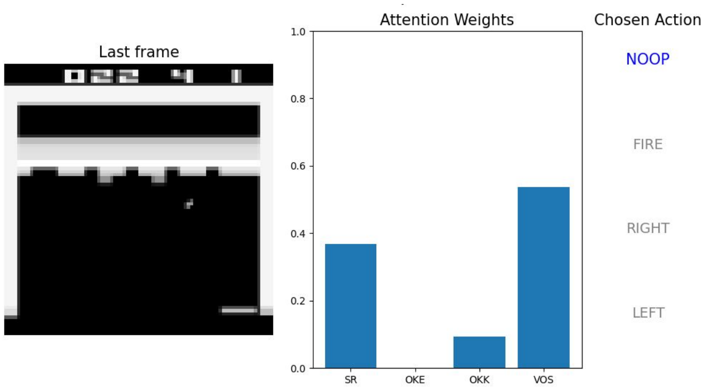


The individual skills are already very informative.

03. WSA EXPLAINABILITY

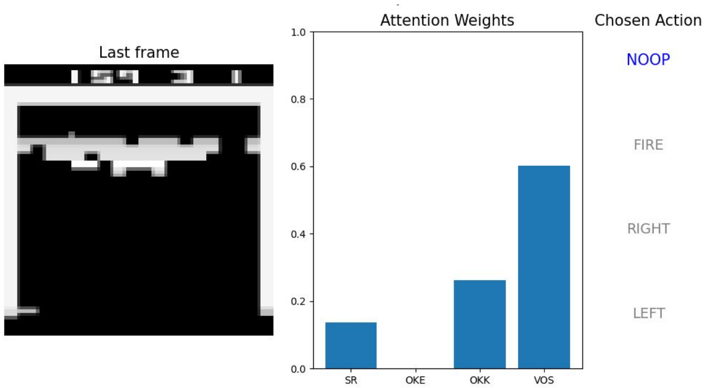
EXPERIMENTS

Combination of multiple skills



First Scenario
Strong presence of SR.

Second Scenario
SR is less informative.



04. CONCLUSIONS

CONCLUSIONS

Discussion

- We analyzed the **end-to-end mapping problem** of current RL algorithms.
- We proposed a set of **skills** to equip the agent with prior knowledge.
- We proposed multiple ways of **combining various encodings**, and in particular we proposed **WSA** as general and scalable combination method.
- We obtained **comparable** results with an end-to-end **PPO** agent and **better** results w.r.t **DQL** without fine-tuning the hyperparameters.

04. CONCLUSIONS

Future Works

- Performing **Hyperparameters Search** could improve the performance of the agents.
- **More Experiments** are needed to obtain more reliable results on average.
- Test WSA on **other benchmarks.**
- Use **different skills** perhaps scaling to very big FMs.
- More in-depth study **WSA Explainability.**

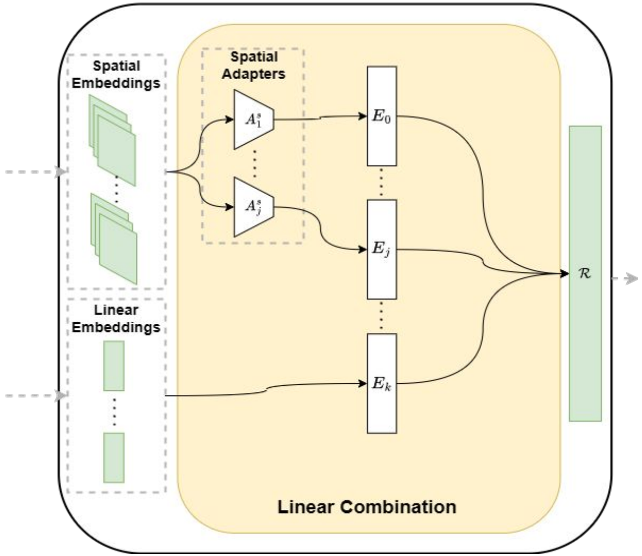
CONCLUSIONS

THANKS FOR YOUR ATTENTION!

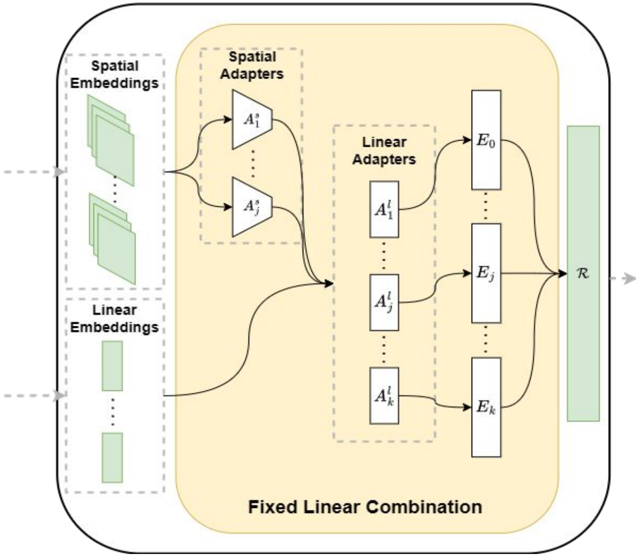
ANY QUESTIONS?

02. COMBINATION MODULES

Linear Combination Modules



LIN



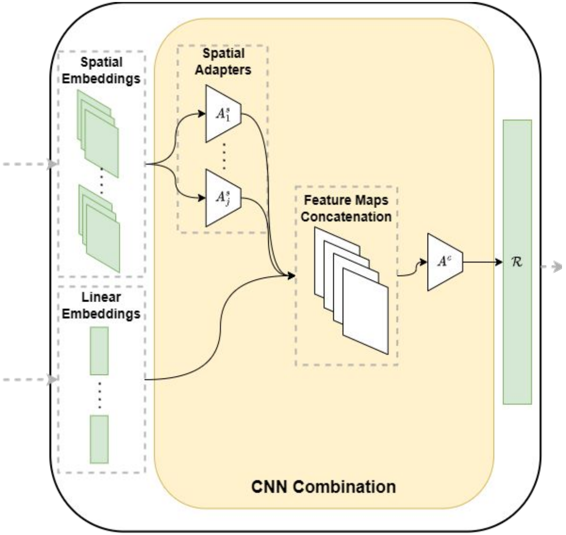
FIX

$$\mathcal{R} = E_1 \oplus E_2 \oplus, \ldots, \oplus E_k$$

02. COMBINATION MODULES

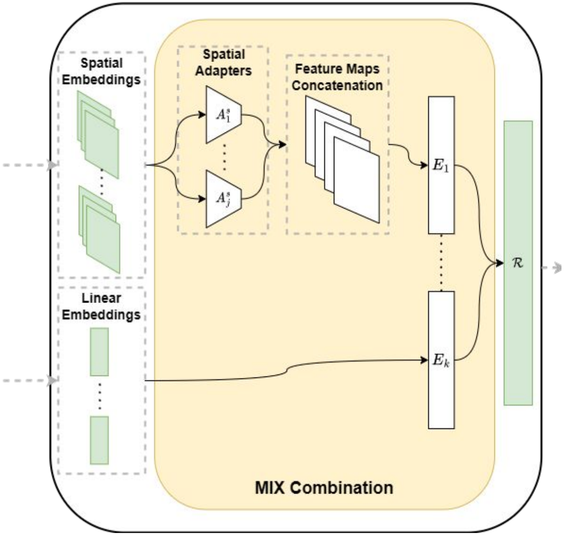
Convolutional Combination Modules

CNN



$$\mathcal{R} = conv(E_1 \oplus, \ldots, \oplus E_k)$$

MIX

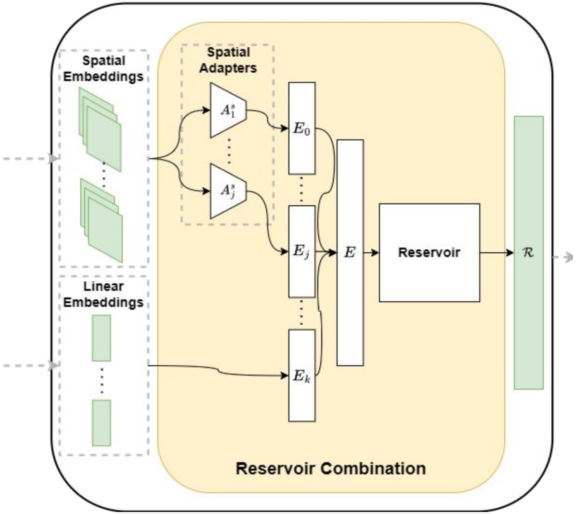


$$\mathcal{R} = E_1^l \oplus, \ldots, \oplus E_p^l \oplus conv(E_1^s \oplus, \ldots, \oplus E_q^s)$$
$$p + q = |\Psi|$$

02. COMBINATION MODULES

Others Combination Modules

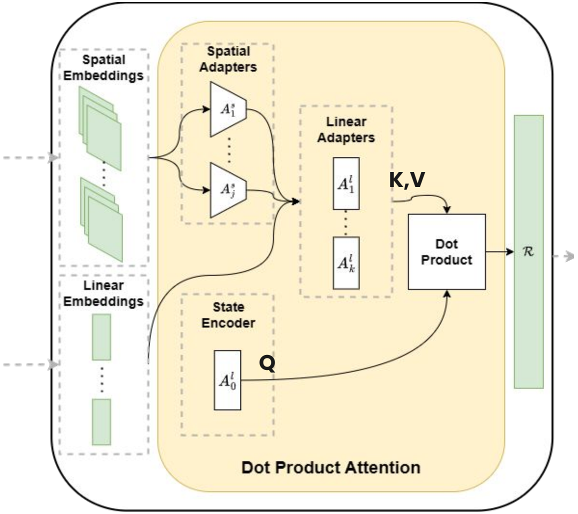
RES



$$IN = E \times \mathbf{W}_{\mathbf{in}}$$
$$H = IN \times \mathbf{W}_{\mathbf{res}}$$

$$\mathcal{R} = tanh(IN + H)$$

DPA

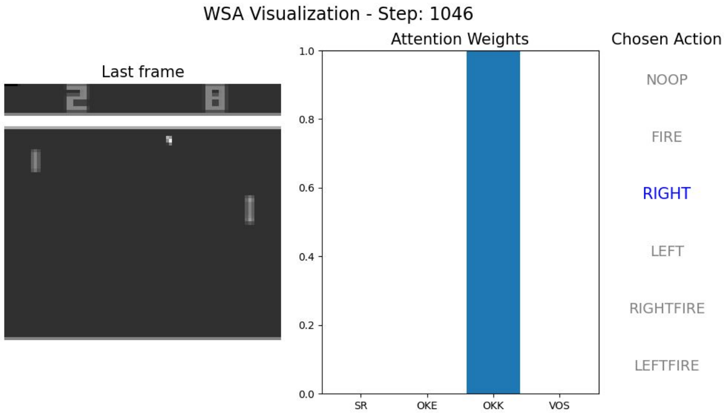


$$\mathbf{W} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\mathcal{R} = \sum_{i=0}^{|\Psi|} w_i * E_i$$

METHODOLOGY

03. WSA EXPLAINABILITY



Optimizations

- Regularization techniques like **Dropout**, **Batch Normalization**.
- Changing activation function, from ReLU to **Linear** or **Sigmoid**.
- Adding a penalty term to the loss considering **attention weights entropy**.

Conclusions

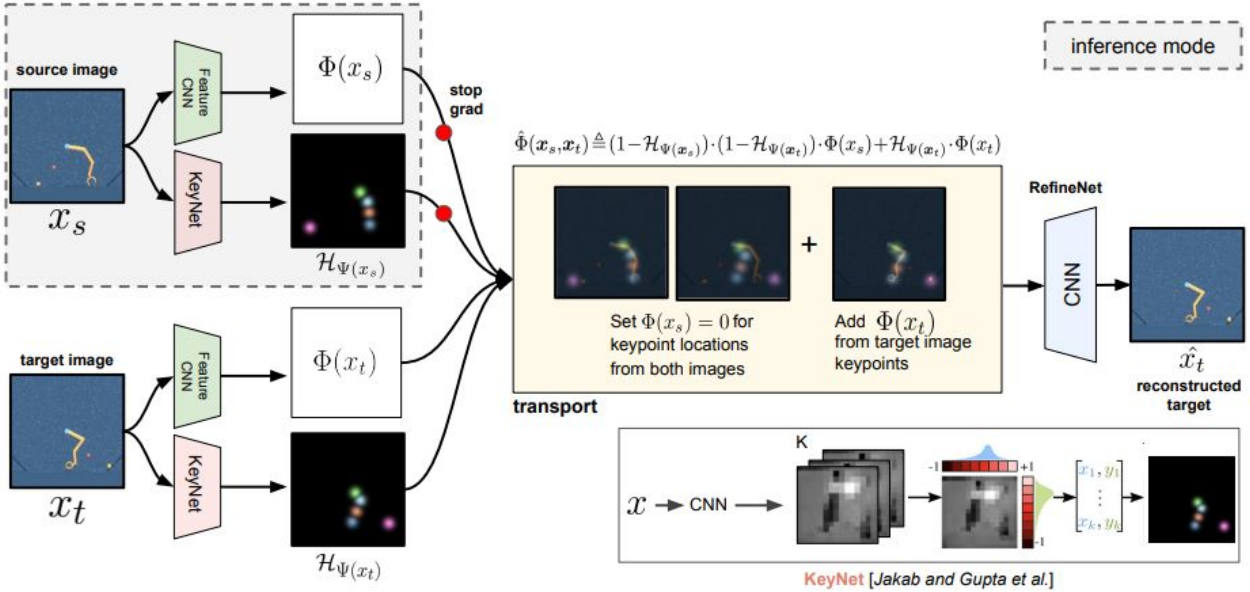
- The individual skills are already very good because in their reference works they were used to improve the performance of an agent.

03. LEARNABLE PARAMETERS

Feature Extractor	Configuration
LIN	8.7M
FIX	4.9M - 19.4M
CNN	4.2M
MIX	4.5M
RES	0.3M - 1.1M
DPA	8.7M - 34.7M
WSA	8.7M - 34.7M
PPO	1.6M

03. SKILLS ARCHITECTURE

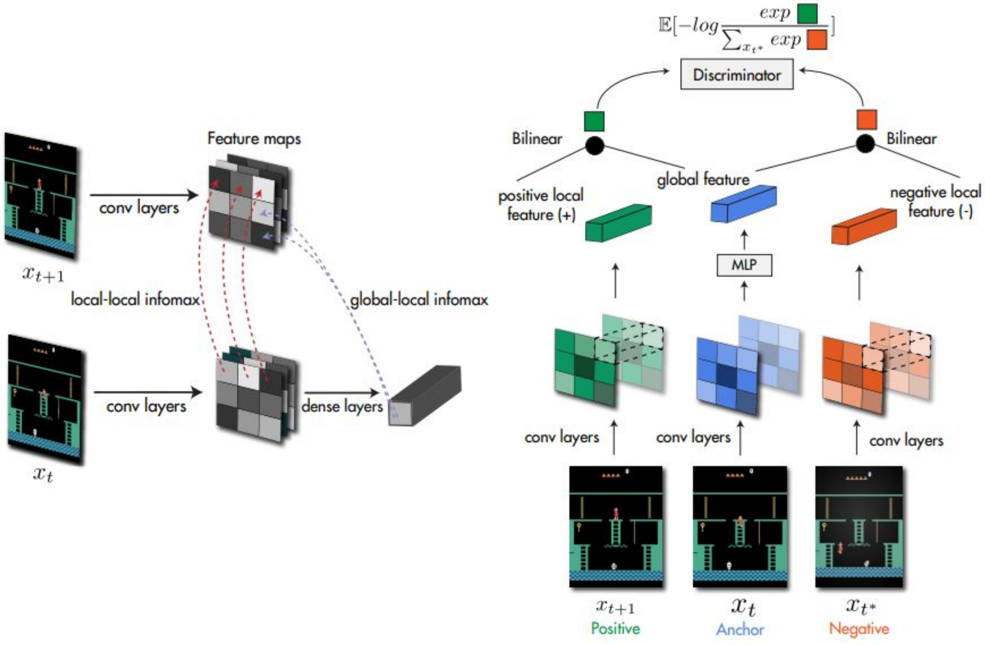
Kulkarni et al. (2019)



Object Keypoints Detection

03. SKILLS ARCHITECTURE

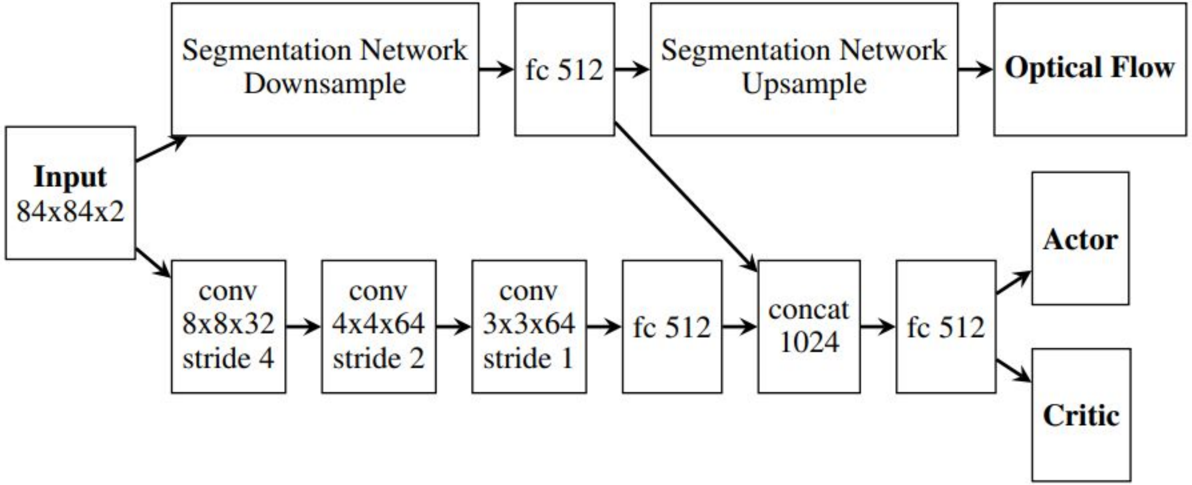
Anand et al. (2019)



State Representation

03. SKILLS ARCHITECTURE

Goel et al. (2018)



Video Object Segmentation

01. REINFORCEMENT LEARNING

INTRODUCTION

Environments as Markov Decision Process (MDP)
MDP = (*S*, *A*, *P*, *R*, γ)

- ***S*** is the set of states an agent can be in.
- ***A*** is the set of actions an agent can take.
- ***P*** is the transition probability function.
- ***R*** is the reward function.
- γ is the discount factor.

$$P^a_{s,s'} = P(S_{t+1} = s' | S_t = s, A_t = a)$$

$$\gamma \in [0, 1]$$

$$R^a_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

01. DEEP Q-LEARNING

Algorithm 3 Deep Q-Learning Algorithm

```
Initialize replay buffer D
Initialize online network Q with random weights w
Initialize target network Q⁻ with weights w⁻ = w
for each episode do
    Initialize S
    for each step of the episode do
        Choose A from S using ε-greedy policy
        Take action A, observe R, S'
        Store transition (S, A, R, S') in D
        Sample random minibatch of transitions (s, a, r, s') from D
        Compute target y = r + γ max_a' Q(s', a', w⁻)
        Compute loss L = (y − Q(s, a, w))^2
        Update weights w by minimizing the loss
        Every C step, update target network weights w⁻ = w
        S ← S'
    end for
    Until S is terminal
end for
```

01. PROXIMAL POLICY OPTIMIZATION

Algorithm 4 Proximal Policy Optimization Algorithm

Initialize policy network $\pi(a|s, \mathbf{w})$ and value network $V(s, \mathbf{w})$
for each iteration **do**
  for each epoch **do**
    Collect a batch of data by running the policy in the environment
    Compute the advantage function A_t
    Compute the probability ratio $r_t(\mathbf{w})$
    Compute the clipped objective function $L(\mathbf{w})$
    Compute the value function loss $L_v(\mathbf{w})$
    Update the policy network by minimizing $L(\mathbf{w})$
    Update the value network by minimizing $L_v(\mathbf{w})$
  end for
end for

01. REINFORCEMENT LEARNING

INTRODUCTION

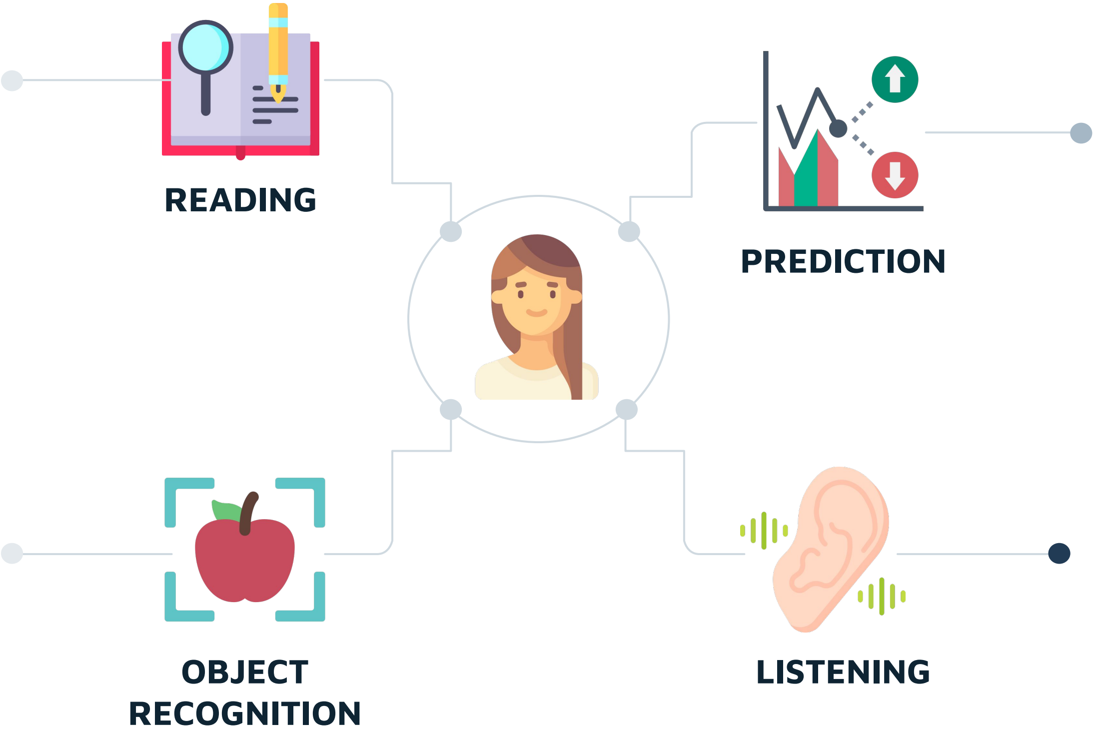
Return $$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Policy $$\pi(a|s) = P(a|s)$$

Value Function $$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \ldots | S_t = s] = \mathbb{E}_\pi[G_t | S_t = s]$$

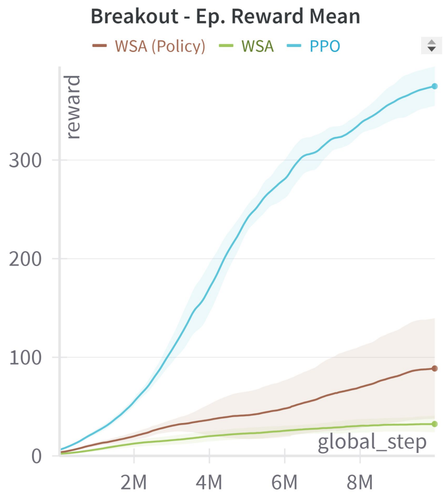
Goal Find a policy π that maximizes the **Value Function**

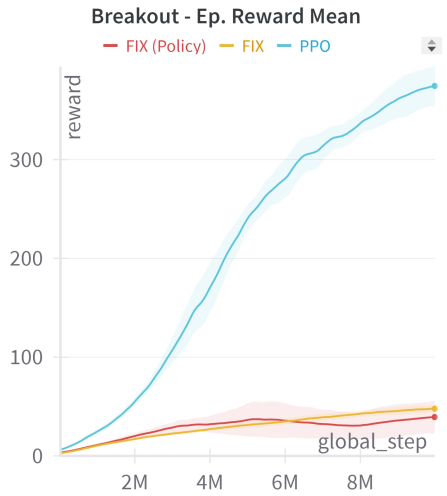
01. PROBLEM FORMULATION

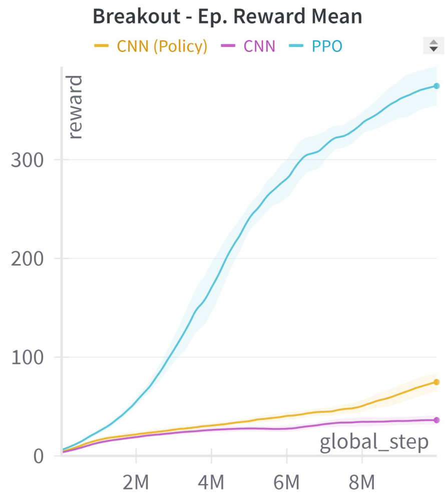


03. BREAKOUT: OUT OF DISTRIBUTION DATA

Hypothesis – Underfitting Problems







Little improvements for WSA in training

GAME OVER