

## 1 Content

- 2 • Figure A certifies that our implementation of DQN is comparable to that of DQN in  
3 Dopamine. It also shows that iDQN with  $K = 1$  yields similar performances to DQN.
- 4 • Table 1 gathers all the hyperparameters of iDQN.
- 5 • Algorithm 1 shows the pseudo-code of iDQN.
- 6 • In Figure B, the individual scores of iDQN on the 54 Atari games are presented along with  
7 the baselines' scores.
- 8 • In Figure C, the individual scores of iDQN on the 54 Atari games are presented along with  
9 the scores of more advanced methods.
- 10 • Figure D provides more insights into the neural network's architecture used in iDQN. It also  
11 illustrates how the rolling step is performed.

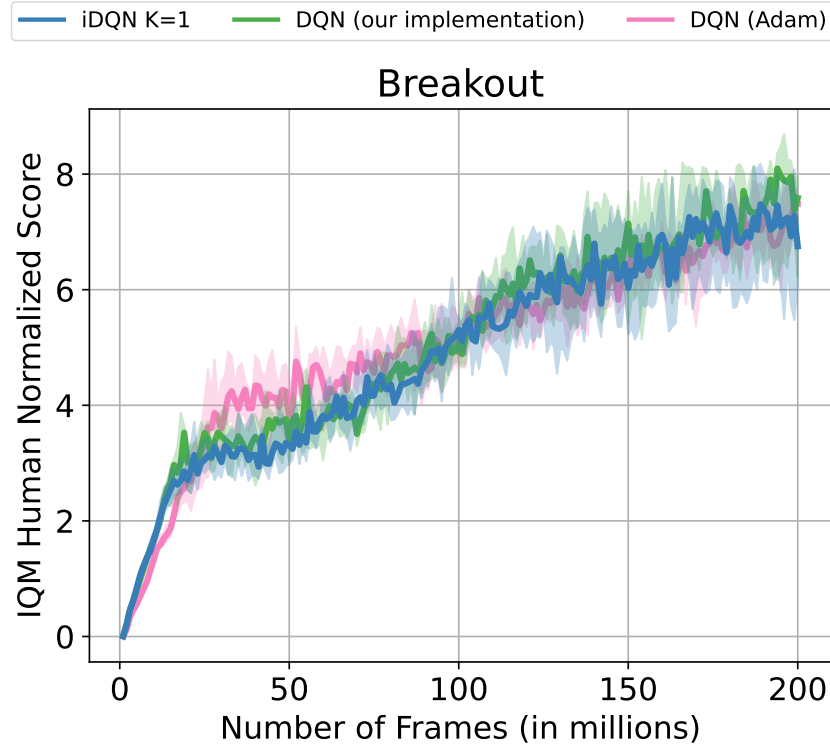


Figure A: Our implementation of DQN yields the same performances as the implementation of Dopamine (DQN (Adam)). This certifies that we can compare the results released in Dopamine with our method. Both DQN implementations and iDQN with  $K = 1$  have a similar behavior. This certifies the trustworthiness of our code base.

Table 1: Summary of all hyperparameters.  $\text{Conv}_{a,b}^d C$  is a 2D convolutional layer with  $C$  filters of size  $a \times b$  and of stride  $d$ , and FCE is a fully connected layer with  $E$  neurons.

Environment	
$\gamma$	0.99
$H$	27000
full action space	No
reward clipping	$\text{clip}(-1, 1)$
DQN	
number of epochs $N$	200
number of training steps per epochs $n$	250000
type of the replay buffer $\mathcal{D}$	FIFO
initial number of samples in $\mathcal{D}$	20000
maximum number of samples in $\mathcal{D}$	1000000
gradient step frequency $G$	4
target update frequency $T$	8000
starting $\epsilon$	1
ending $\epsilon$	0.01
$\epsilon$ linear decay duration	250000
batch size	32
learning rate	$6.25 \times 10^{-5}$
Adam $\epsilon$	$1.5 \times 10^{-4}$
torso architecture	$\text{Conv}_{8,8}^4 32 - \text{Conv}_{4,4}^2 64 - \text{Conv}_{3,3}^1 64 -$
head architecture	$-\text{FC}_{512} - \text{FC}_{n_{\mathcal{A}}}$
activations	ReLU
initializer	Xavier uniform
iDQN	
rolling step frequency $R$	6000
target update frequency $T$	30
sampling policy $\mu$	uniform

---

**Algorithm 1** iDQN. The modifications added to DQN are marked in green.

---

```

1: Inputs: number of epochs  $N$ , number of training steps per epoch  $n$ , sampling head policy  $\mu$ ,
   online and target parameters  $\theta = \bar{\theta}$ , replay buffer  $\mathcal{D}$ , gradient step frequency  $G$ , rolling step
   frequency  $R$ , target update frequency  $T$ .
2:
3:  $i \leftarrow 0$  ▷ number of overall training steps
4: performance  $\leftarrow$  empty list
5: for  $N$  epochs do
6:    $j \leftarrow 0$  ▷ number of training steps within an epoch
7:    $s \leftarrow \text{env.init}()$ 
8:   absorbing  $\leftarrow$  false; sum_reward  $\leftarrow$  0; n_episodes  $\leftarrow$  0
9:   while  $j < n$  and absorbing = false do
10:    sample  $k \sim \mu$  ▷ sample a neural network head
11:    sample  $a \sim \epsilon$ -greedy  $Q_k(s, \cdot | \theta)$ 
12:     $(s', r, \text{absorbing}) \leftarrow \text{env.step}(a)$ 
13:     $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s, a, r, s')\}$ 
14:     $s \leftarrow s'$ ; sum_reward  $+= r$ 
15:    if absorbing = true then
16:       $s \leftarrow \text{env.init}()$ 
17:      n_episodes  $+= 1$ 
18:    end if
19:
20:    if  $i = 0[G]$  then
21:       $d \sim \mathcal{U}(\mathcal{D})$ 
22:       $\theta \leftarrow \text{Adam\_update}(\mathcal{L}, d, \theta, \bar{\theta})$  ▷  $\mathcal{L}$  is defined in (2)
23:    end if
24:    if  $i = 0[R]$  then
25:       $(\theta, \bar{\theta}) \leftarrow \text{rolling\_step}(\theta, \bar{\theta})$  ▷ explained in Section 4
26:    end if
27:    if  $i = 0[T]$  then
28:       $\bar{\theta} \leftarrow \theta$ 
29:    end if
30:     $i += 1$ ;  $j += 1$ 
31:  end while
32:  performance.append( $\frac{\text{sum\_reward}}{\text{n\_episodes}}$ )
33: end for
34: return  $\theta$ 

```

---

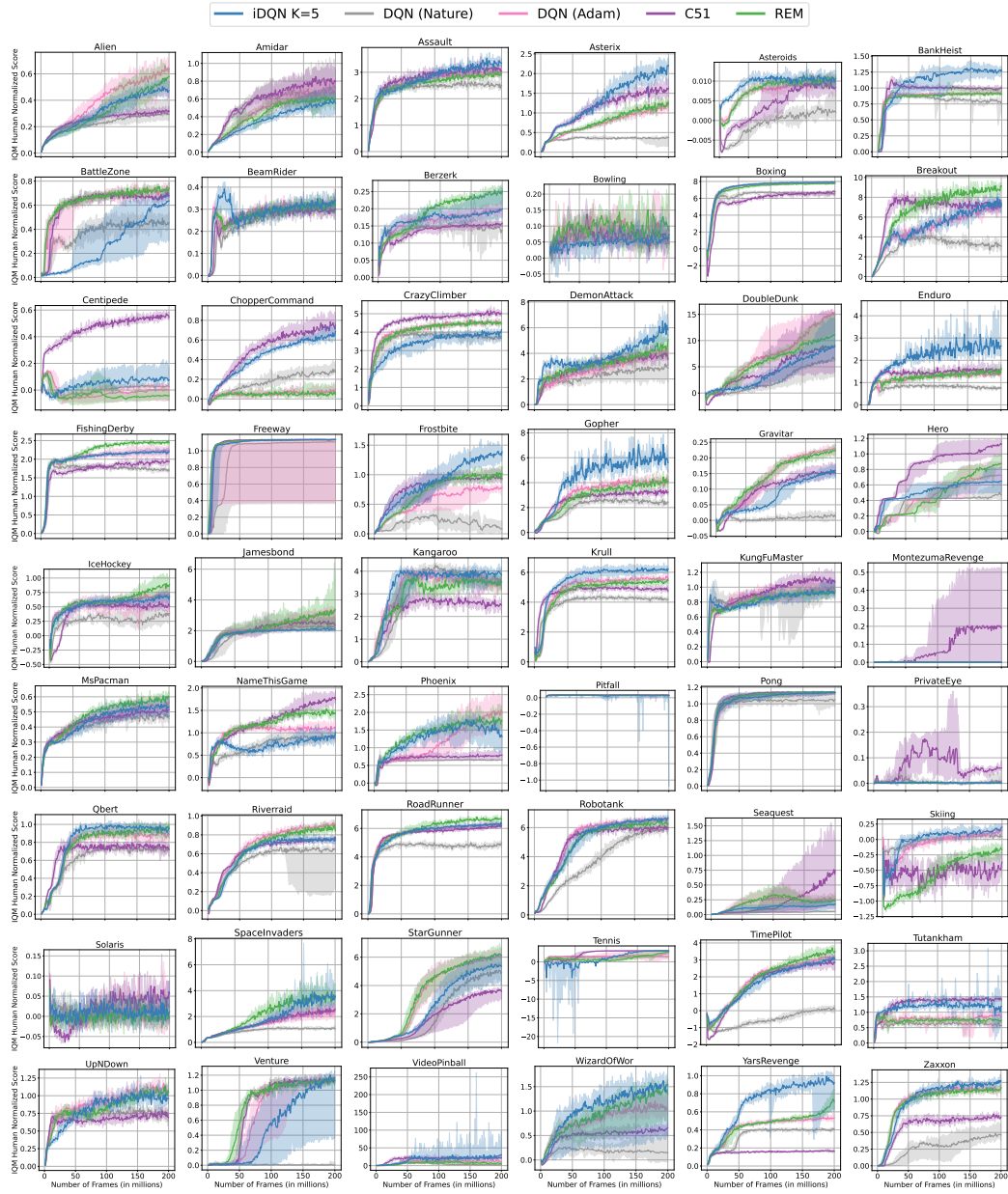


Figure B: Performances of iDQN ( $K = 5$ ) on the 54 Atari games along with the considered baselines.

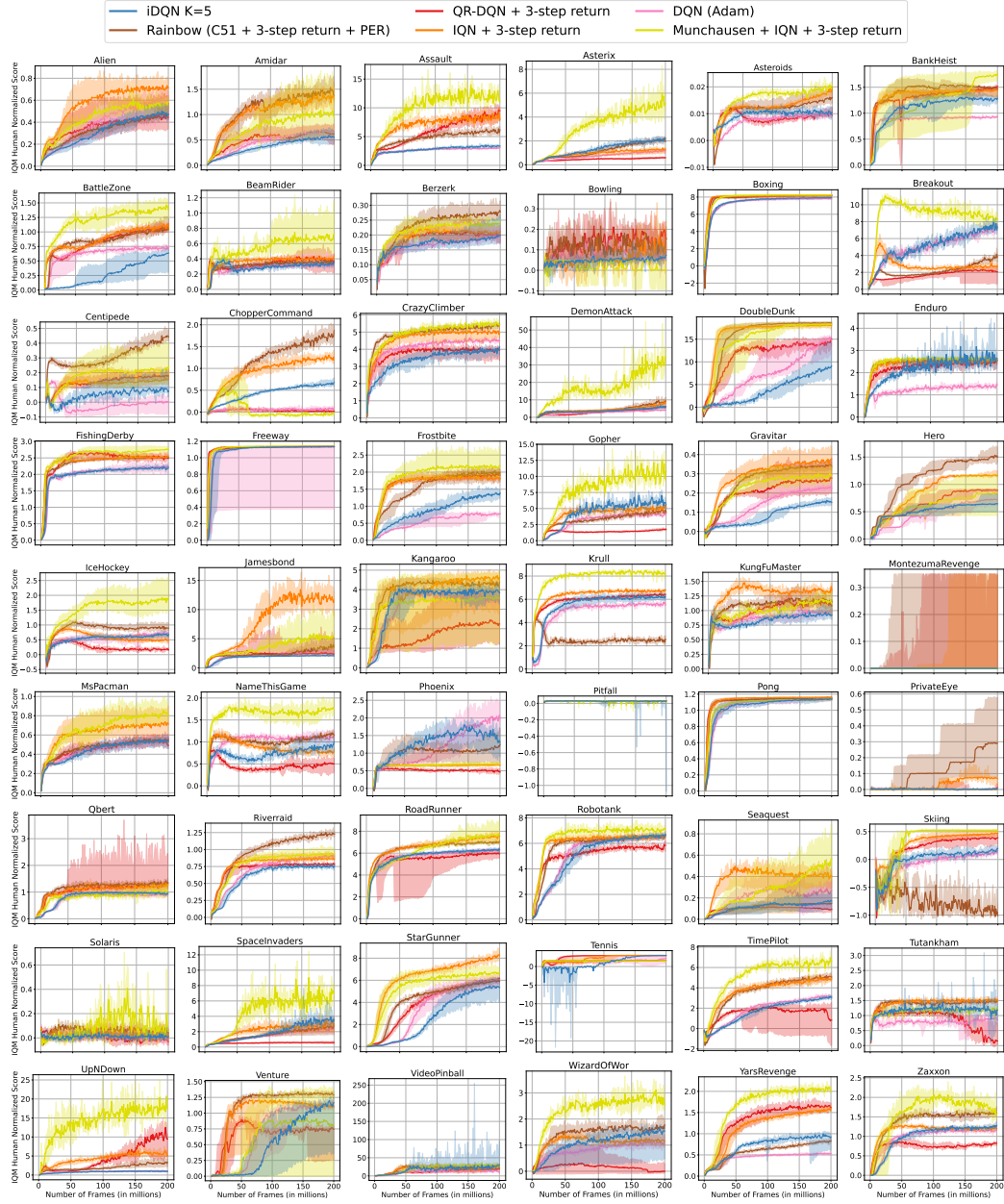


Figure C: Performances of iDQN ( $K = 5$ ) on the 54 Atari games along with other improvements over DQN.

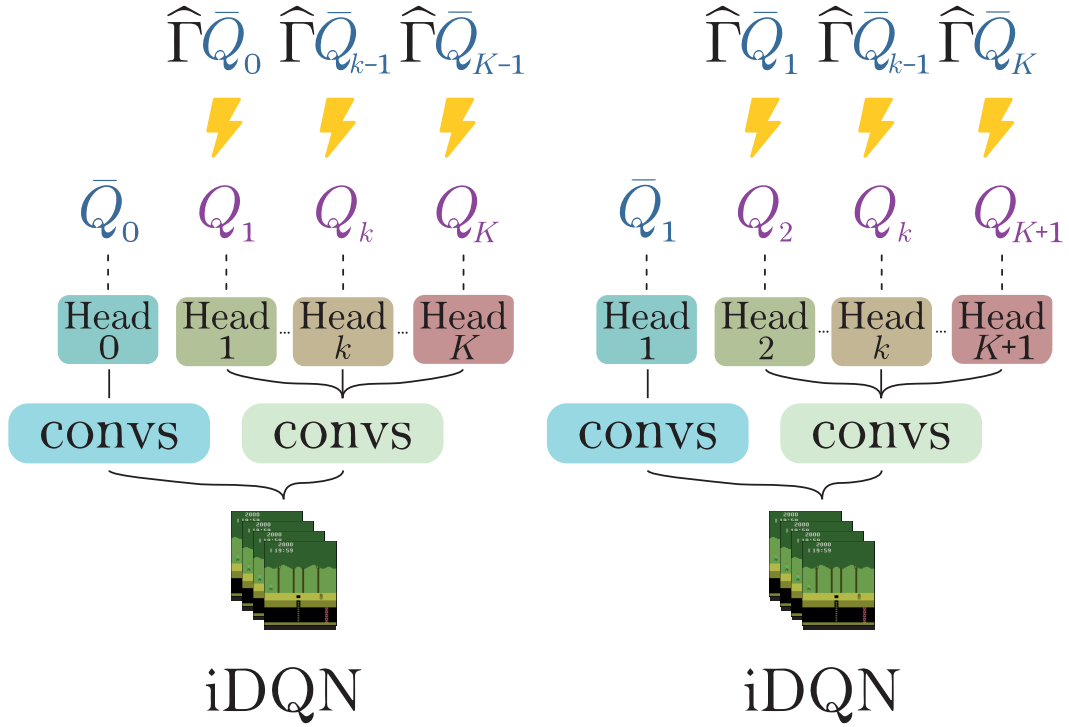


Figure D: Illustration of the rolling step. The first head of the neural network's architecture is not trained. Therefore, the convolutional layers for the first head are computed separately to ensure their outputs are still the ones it has been trained on.