

A MASP Regularization and Implementation Details

A.1 Detailed Formulation of MASP

The Macro-Action Similarity Penalty (MASP) is a regularization term designed to enforce smoothness across the Q-values of similar actions (both primitive and macro-actions) in the augmented action space. To achieve this, we define a similarity matrix $\Sigma \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$, where each entry $\Sigma_{ij} \geq 0$ represents the degree of similarity between actions a_i and a_j . In our implementation, Σ is constrained to be symmetric and non-negative, but we do not require it to be positive-definite.

Given a batch of transitions $\{(s_i, a_i, r_i, s_{i+1})\}_{i=1}^n$, MASP adds the following regularization to the TD loss:

$$\mathcal{L}_{\text{MASP}} = \eta \cdot \frac{1}{n} \sum_{i=1}^n \|Q(s_i, \cdot; \theta) - \Sigma Q(s_i, \cdot; \theta)\|_2^2 \quad (3)$$

where $Q(s_i, \cdot; \theta) \in \mathbb{R}^{|\mathcal{A}|}$ is the vector of Q-values for all actions at state s_i , and η is a tunable regularization coefficient.

Intuition: This penalty encourages the Q-values of similar actions (according to Σ) to be close, effectively sharing credit among macro-actions that are functionally related. Unlike options or hierarchical RL, our penalty is “soft”—allowing some dispersion for exploration and differentiation.

A.2 Meta-learning Procedure for Σ

The similarity matrix Σ is meta-learned jointly with the agent’s main network parameters θ via meta-gradients. This is achieved as follows:

1. Inner Update (Agent Step):

- Sample a trajectory τ from the replay buffer.
- Perform a standard TD update with the MASP penalty, updating $\theta \rightarrow \theta'$ using Σ fixed.

2. Outer Update (Meta Step):

- Sample a new trajectory τ' .
- Evaluate the performance of the updated θ' using a meta-objective (the standard TD loss).
- Compute the meta-gradient of this meta-objective w.r.t. Σ (backpropagating through the inner update step).
- Update Σ with a separate learning rate β .

This approach is similar to the meta-gradient method introduced by Xu et al. [32]. In practice, we use automatic differentiation and checkpointing to efficiently compute $\frac{d\theta'}{d\Sigma}$.

Algorithmic Details:

- See Algorithm 1 in the main text for the pseudocode.
- Σ is parameterized as a free matrix with symmetry enforced by averaging with its transpose after each update.
- To prevent divergence, we clip the entries of Σ to the range $[0, 1]$ after each update.

A.3 Practical Implementation Details

Embedding Σ : For large action spaces, representing Σ explicitly can be expensive. We instead flatten Σ and use a learned projection:

$$e_\Sigma = W_{\text{emb}} \cdot \text{vec}(\Sigma)$$

where W_{emb} is a trainable matrix, and e_Σ is a low-dimensional embedding vector. This is concatenated with the state embedding and fed into the Q-network.

Additional Training Details:

- Both Σ and W_{emb} are updated via backpropagation from the main loss.
- The meta-objective for Σ updates does not include gradients through W_{emb} .
- To prevent Σ from degenerating to the identity or to a rank-one matrix, we add a small entropy penalty to its row-normalized version during meta-learning.

Computational Cost:

- The MASP regularization is vectorized using batched matrix multiplications.
- The additional overhead is roughly 20% over standard DQN (wallclock time), dominated by meta-gradient computation and matrix operations.

B Experimental Details and Hyperparameters

We summarize Atari preprocessing settings in Table 6 and the main algorithm hyperparameters in Table 7.

Hyperparameter	Value
Max frames per episode	108,000
Observation down-sampling	(84, 84)
Num. action repeats	4
Num. stacked frames	1
Terminal state on loss of life	<i>true</i>
Random noops range	30
Sticky actions	<i>true</i>
Frames max pooled	3 and 4
Grayscaled/RGB	Grayscaled
Action set	Full

Table 6: Atari pre-processing hyperparameters.

Reproducibility and Code Availability

To facilitate reproducibility, all code used for the experiments and MASP implementation is available at: <https://github.com/rl-submissions/macro-credit-masp>.

B.1 Compute Resources and Training Time

Our experiments were conducted on a compute cluster equipped with 192GB RAM and a combination of NVIDIA GPUs: 1x RTX 5090, 2x RTX 4090, and 4x RTX 3090 cards. All Atari experiments typically required approximately one week to complete per full experimental run (including all seeds, ablations, and sweeps). Training runs for StreetFighter environments generally completed in approximately two days.

C Additional Experimental Results

C.1 Atari Results

Full Atari performance for all methods and human-normalized scores are reported in Table 8.

C.2 Streetfighter II Experimental Details

Streetfighter II experiments were conducted with domain-specific preprocessing and macro-action settings, including a reduced action set and macro-actions corresponding to common combos. All environment and training hyperparameters are detailed in Table 9.

Hyperparameter	Value
Optimizer	Adam
Learning rate	6.25×10^{-5}
Batch size	32
Discount factor γ	0.99
Replay buffer size	1×10^6
Target network update period	10,000 steps
Gradient clipping	10
Exploration ϵ (initial / final)	1.0 / 0.01
Exploration decay schedule	1M steps
Multi-step returns (n)	3
Noisy nets	<i>true</i>
Distributional atoms	51
Distributional min/max values	-10 / 10
Dueling network	<i>true</i>
Prioritized replay α	0.5
Prioritized replay β	$0.4 \rightarrow 1.0$
Macro-action set size (k)	32 (see ablations)
Macro-action length	3–8
MASP penalty weight η	0.1, 0.3, 0.5, 0.7, 1 (swept)
Meta-learning rate β	0.001, 0.005, 0.01 (swept)
Σ embedding dim (e_Σ)	32

Table 7: Main hyperparameters used for Rainbow-DQN, macro-action augmentation, and MASP regularization in all experiments.

A visualization of the learned similarity matrix Σ for macro-actions in Street Fighter II can be found in Figure 3 in this appendix, which illustrates the clustering of related macro-actions discovered by the agent during training.

C.3 MiniGrid Experiments

Experimental Setup. We also tested MASP on MiniGrid [37] environments designed to require planning and structured exploration. We selected tasks such as DoorKey, FourRooms, LockedRoom, and ObstructedMaze-Full.

Implementation Details. Macro-actions consist of fixed sequences derived from typical interaction patterns (e.g., forward-forward-turn). We compare RAINBOW DQN, RAINBOW with macro-actions, and RAINBOW with MASP. Each agent is trained for 500,000 steps. Hyperparameter tuning for η was done per environment.

Results and Analysis. As shown in Table 10, MASP improves success rates over both baselines across all tasks. In simpler settings, the improvements are significant, and in more complex environments like ObstructedMaze-Full, MASP is critical to achieving high success.

C.4 Transferability

Another key motivation for the Macro-Action Similarity Penalty (MASP) framework is to improve transferability across tasks and environments. In principle, MASP regularization encourages the agent to learn more robust and generalizable representations by smoothing the Q-values among similar macro-actions, potentially facilitating adaptation to new tasks where action semantics overlap.

Transfer Protocol: In our experiments, we evaluated transferability by taking agents trained with MASP on a subset of tasks and fine-tuning them (with or without additional MASP updates) on related environments. In practice, the learned similarity matrix Σ and macro-action set can be reused or adapted for downstream tasks, reducing the need for retraining from scratch.

Game	Rainbow-DQN	+Macro-Actions	+MASP Score HN Score
Alien	6,022.9 ± 718.2	3,714.8 ± 246.0	7614.4 ± 388.5 (121.4)
Amidar	202.8 ± 23.4	185.8 ± 8.0	272.5 ± 7.0 (48.7)
Assault	14,491.7 ± 759.0	11,368.2 ± 1174.3	15665.7 ± 1682.2 (540.4)
Asterix	280,114.0 ± 23760.7	225,501.1 ± 25686.6	356834.2 ± 22386.9 (2102.2)
Asteroids	2,249.4 ± 191.5	2,093.2 ± 148.5	3544.9 ± 362.3 (52.3)
Atlantis	814,684.0 ± 42022.1	766,181.5 ± 52739.2	933826.3 ± 48727.3 (6721.3)
Bank Heist	826.0 ± 60.3	708.5 ± 20.8	1159.4 ± 70.6 (219.2)
Battle Zone	52,040.0 ± 4502.2	30,290.9 ± 2185.3	67537.1 ± 5791.1 (398.2)
Beam Rider	21,768.5 ± 1356.5	16,221.3 ± 1208.4	29374.2 ± 3309.8 (232.5)
Berzerk	1,793.4 ± 96.1	1,026.2 ± 36.7	2635.8 ± 122.0 (180.3)
Bowling	39.4 ± 4.2	35.1 ± 2.6	66.6 ± 6.0 (34.2)
Boxing	54.9 ± 2.7	50.6 ± 1.5	100.0 ± 10.3 (1000.0)
Breakout	379.5 ± 25.1	252.9 ± 22.2	884.4 ± 74.0 (1011.2)
Centipede	7,160.9 ± 211.7	4,487.6 ± 125.5	7489.6 ± 445.0 (72.4)
Chopper Command	10,916.0 ± 1034.1	7,464.7 ± 392.6	11592.2 ± 876.8 (73.3)
Crazy Climber	143,962.0 ± 13254.7	80,474.4 ± 7541.7	158672.8 ± 7774.1 (207.5)
Defender	47,671.3 ± 4264.2	30,844.1 ± 943.5	58679.4 ± 7205.4 (346.3)
Demon Attack	109,670.7 ± 2672.4	72,716.0 ± 5913.2	117663.3 ± 11120.4 (1014.3)
Double Dunk	-0.6 ± 0.0	-0.7 ± 0.1	-0.2 ± 0.0 (53.7)
Enduro	2,061.1 ± 83.3	1,324.7 ± 148.7	2266.6 ± 58.3 (67.6)
Fishing Derby	22.6 ± 2.8	22.6 ± 0.6	36.9 ± 1.0 (75.3)
Freeway	29.1 ± 2.7	25.2 ± 1.3	30.3 ± 3.7 (101.0)
Frostbite	4,141.1 ± 175.4	3,361.2 ± 356.4	5566.7 ± 317.3 (65.5)
Gopher	72,595.7 ± 6706.9	45,801.3 ± 5387.9	78992.5 ± 3703.2 (803.5)
Gravitar	567.5 ± 20.5	367.6 ± 42.6	645.5 ± 31.8 (60.5)
Hero	50,496.8 ± 3850.7	42,269.0 ± 1228.2	62730.1 ± 3469.7 (261.0)
Ice Hockey	-0.7 ± 0.1	-0.8 ± 0.0	-0.1 ± 0.0 (53.6)
Kangaroo	10,841.0 ± 1247.0	8,271.9 ± 1008.4	11225.7 ± 1304.9 (132.6)
Krull	6,715.5 ± 823.5	4,597.9 ± 342.5	8251.1 ± 858.9 (65.1)
Kung Fu Master	28,999.8 ± 2080.4	18,165.9 ± 1865.3	36837.2 ± 1430.2 (151.2)
Montezuma's Revenge	154.0 ± 16.9	124.0 ± 5.3	400.0 ± 18.1 (3.5)
Ms Pacman	2,570.2 ± 204.6	1,471.5 ± 185.3	2966.5 ± 360.5 (67.6)
Name This Game	11,686.5 ± 315.2	7,252.3 ± 180.3	12745.8 ± 685.4 (131.5)
Phoenix	103,061.6 ± 10294.9	89,925.5 ± 10494.5	116444.7 ± 5868.9 (231.2)
Pitfall	-37.6 ± 2.0	-49.8 ± 4.8	-12.8 ± 1.1 (3.3)
Pong	19.0 ± 1.4	19.0 ± 1.0	19.0 ± 0.3 (104.2)
Private Eye	1,704.4 ± 41.7	1,358.2 ± 94.9	2244.3 ± 150.9 (13.2)
Q Bert	18,397.6 ± 637.1	12,276.2 ± 854.1	22774.8 ± 1373.3 (417.0)
Road Runner	54,261.0 ± 2330.3	45,952.7 ± 1687.5	62633.7 ± 3796.5 (220.4)
Robotank	55.2 ± 6.4	50.5 ± 6.3	64.5 ± 7.6 (75.1)
Seaquest	19,176.0 ± 1428.9	13,917.1 ± 641.1	23768.8 ± 1476.9 (240.0)
Skiing	-11,685.8 ± 787.3	-15,069.0 ± 566.2	-10114.6 ± 575.5 (42.6)
Solaris	2,860.7 ± 153.5	1,764.4 ± 217.6	4488.9 ± 267.5 (75.3)
Space Invaders	12,629.0 ± 1413.2	7,496.4 ± 532.9	16668.2 ± 2050.5 (197.6)
Star Gunner	123,853.0 ± 4413.1	112,043.0 ± 12667.8	169778.8 ± 11049.3 (533.5)
Surround	7.0 ± 0.3	6.3 ± 0.3	8.72 ± 0.72 (69.8)
Tennis	-2.2 ± 0.2	-3.0 ± 0.4	12.6 ± 0.6 (60.6)
Time Pilot	11,190.5 ± 410.0	8,331.4 ± 931.3	15583.3 ± 707.4 (209.7)
Tutankham	126.9 ± 5.0	70.5 ± 7.9	179.6 ± 14.9 (48.2)
Venture	45.0 ± 3.4	25.0 ± 3.1	133.7 ± 8.4 (20.5)
Video Pinball	506,817.2 ± 16888.8	290,039.4 ± 11557.8	577339.3 ± 20750.9 (327.4)
Wizard of Wor	14,631.5 ± 948.6	10,125.1 ± 891.3	18866.5 ± 1771.3 (259.5)
Yarr's Revenge	93,007.9 ± 5494.8	55,584.7 ± 3232.2	103544.9 ± 3201.0 (134.8)
Zaxxon	19,658.0 ± 2229.6	17,553.0 ± 1147.8	26566.7 ± 2208.2 (103.3)

Table 8: Comparison between RAINBOW DQN [11], RAINBOW-DQN with macro-actions (+Macro-Actions), and RAINBOW DQN with macro-actions similarity penalty (+MASP). **Bold** indicates maximal raw performance between RAINBOW DQN and MASP. Human-normalized (HN) scores are shown in parentheses. Cells highlighted in pink denote $\text{HN} \geq 100$.

Hyperparameter	Value
Observation shape	(128, 128, 3)
Frame skip / Action repeat	2
Num. stacked frames	4
Reward clipping	[-1, 1]
Opponent	Random / AI Level 3
Max episode steps	18,000
Terminal on round loss	<i>true</i>
No-op start range	0–10
Sticky actions	<i>false</i>
Action set	Reduced (15 discrete moves)
Combo macro-actions	<i>true</i>
Macro-action set size (k)	24
Macro-action length	2–6
MASP penalty weight η	0.3, 0.5 (swept)
Meta-learning rate β	0.001, 0.003 (swept)
Σ embedding dim (e_Σ)	16
Optimizer	Adam
Learning rate	1×10^{-4}
Batch size	32
Replay buffer size	5×10^5
Discount factor γ	0.99
Target network update period	5,000 steps
Exploration ϵ (initial/final)	1.0 / 0.05
Exploration decay schedule	200k steps
Dueling network	<i>true</i>
Distributional RL	<i>true</i>

Table 9: Hyperparameters for Streetfighter II experiments. Settings reflect domain-specific differences, including observation size, combo macro-actions, and action set.

Task	Rainbow DQN	Macro-Actions	MASP
Door Key	0.83	0.88	1
Four Rooms	0.69	0.87	1
Locked Room	0.58	0.77	0.97
Obstructed Maze Full	0.57	0.84	0.91

Table 10: Comparison between the success rate of RAINBOW DQN, RAINBOW DQN with macro-actions and MASP for MiniGrid environments.

508 **Observations:** We found that MASP-trained agents generally adapted more quickly and achieved
509 higher initial performance on transfer tasks compared to standard Rainbow DQN baselines. This sug-
510 gests that MASP helps encode transferable structure in the Q-function and macro-action embeddings.

511 **Limitations:** The degree of transfer benefit depends on the similarity between source and target task
512 action spaces. Large discrepancies may require re-learning or adaptation of Σ .

Hyperparameter	Value
Max frames per episode	2,000
Num. action repeats	1
Num. stacked frames	1
Terminal state on loss of life	N/A
Random noops range	0
Sticky actions	<i>false</i>
Replay buffer size	5×10^4
Batch size	64
Learning rate	1×10^{-4}
Discount factor γ	0.99
Target network update period	1,000 steps
Exploration ϵ (initial / final)	0.2 / 0.01
Exploration decay schedule	50k steps
Multi-step returns (n)	1
Noisy nets	<i>false</i>
Dueling network	<i>false</i>
Macro-action set size (k)	8
Macro-action length	2–4
MASP penalty weight η	0.05, 0.1, 0.3 (swept)
Meta-learning rate β	0.001
Σ embedding dim (e_Σ)	8

Table 11: MiniGrid-specific hyperparameters. Only hyperparameters that differ from Atari are shown.

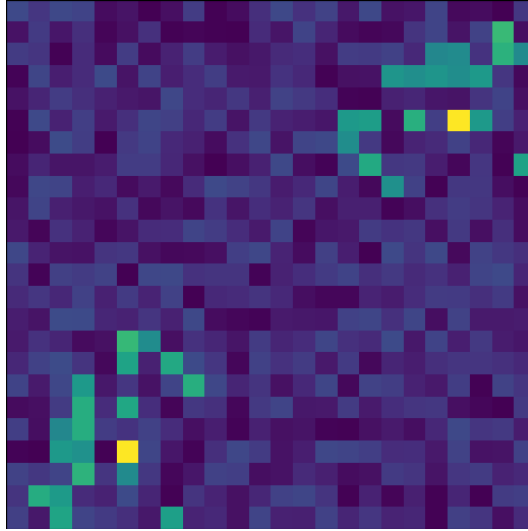


Figure 3: Sample Σ matrix from Street Fighter II experiments, illustrating the learned similarities between different macro-actions. Distinct clusters with higher values indicate groups of macro-actions that are functionally related or often co-activated. In contrast, the regions of the matrix with the lowest values and lacking visible structure correspond to primitive actions, which are entirely independent and dissimilar to each other.