## A. PixelCNN Hyper-parameters

The PixelCNN model used in this paper is a lightweight variant of the Gated PixelCNN introduced in (van den Oord et al., 2016a). It consists of a $7 \times 7$ masked convolution, followed by two residual blocks with $1 \times 1$ masked convolutions with 16 feature planes, and another $1 \times 1$ masked convolution producing $64$ features planes, which are mapped by a final masked convolution to the output logits. Inputs are $42 \times 42$ greyscale images, with pixel values quantized to $8$ bins.

The model is trained completely online, from the stream of Atari frames experienced by an agent. Optimization is performed with the (uncentered) RMSProp optimizer (Tieleman & Hinton, 2012) with momentum $0.9$, decay $0.95$ and epsilon $10^{-4}$.

## B. Methodology

Unless otherwise stated, all agent performance graphs in this paper show the agent's training performance, measured as the undiscounted per-episode return, averaged over 1M environment frames per data point.

The algorithm-comparison graphs Fig. 6 and Fig. 8 show the relative improvement of one algorithm over another in terms of area-under-the-curve (AUC). A comparison by maximum achieved score would yield similar overall results, but underestimate the advantage in terms of learning speed (sample efficiency) and stability that the intrinsically motivated and MMC-based agents show over the baselines.

## C. Convolutional CTS

In Section 4 we have seen that DQN-PixelCNN outperforms DQN-CTS in most of the 57 Atari games, by providing a more impactful exploration bonus in hard exploration games, as well as a more graceful (less harmful) one in games where the learning algorithm does not benefit from the additional curiosity signal. One may wonder whether this improvement is due to the generally more expressive and accurate density model PixelCNN, or simply its convolutional nature, which gives it an advantage in generalization and sample efficiency over a model that represents pixel probabilities in a completely location-dependent way.

To answer this question, we developed a convolutional variant of the CTS model. This model has a single set of parameters conditioning a pixel's value on its predecessors shared across all pixel locations, instead of the location-dependent parameters in the regular CTS. In Fig. 14 we contrast the performance of DQN, DQN-MC, DQN-CTS, DQN-ConvCTS and DQN-PixelCNN on 6 example games.

We first consider dense reward games like Q*BERT and ZAXXON, where most improvement comes from the use of the MMC, and the exploration bonus hurts performance. We find that in fact convolutional CTS behaves fairly similarly to PixelCNN, leaving agent performance unaffected, whereas regular CTS causes the agent to train more slowly or reach an earlier performance plateau. On the sparse reward games (GRAVITAR, PRIVATE EYE, VENTURE) however, convolutional CTS shows to be as inferior to PixelCNN as the vanilla CTS variant, failing to achieve the significant improvements over the baseline agents presented in this paper.

We conclude that while the convolutional aspect plays a role in the 'softer' nature of the PixelCNN model compared to its CTS counterpart, it alone is insufficient to explain the massive exploration boost that the PixelCNN-derived reward provides to the DQN agent. The more advanced model's accuracy advantage translates into a more targeted and useful curiosity signal for the agent, which distinguishes novel from well-explored states more clearly and allows for more effective exploration.

## D. The Hardest Exploration Games

Table 1 reproduces Bellemare et al. (2016)'s taxonomy of games available through the ALE according to their exploration difficulty. "Human-Optimal" refers to games where DQN-like agents achieve human-level or higher performance; "Score Exploit" refers to games where agents find ways to achieve superhuman scores, without necessarily playing the game as a human would. "Sparse" and "Dense" rewards are qualitative descriptors of the game's reward structure. See the original source for additional details.

Table 2 compares previously published results on the 7 hard exploration, sparse reward Atari 2600 games with results obtained by DQN-CTS and DQN-PixelCNN.

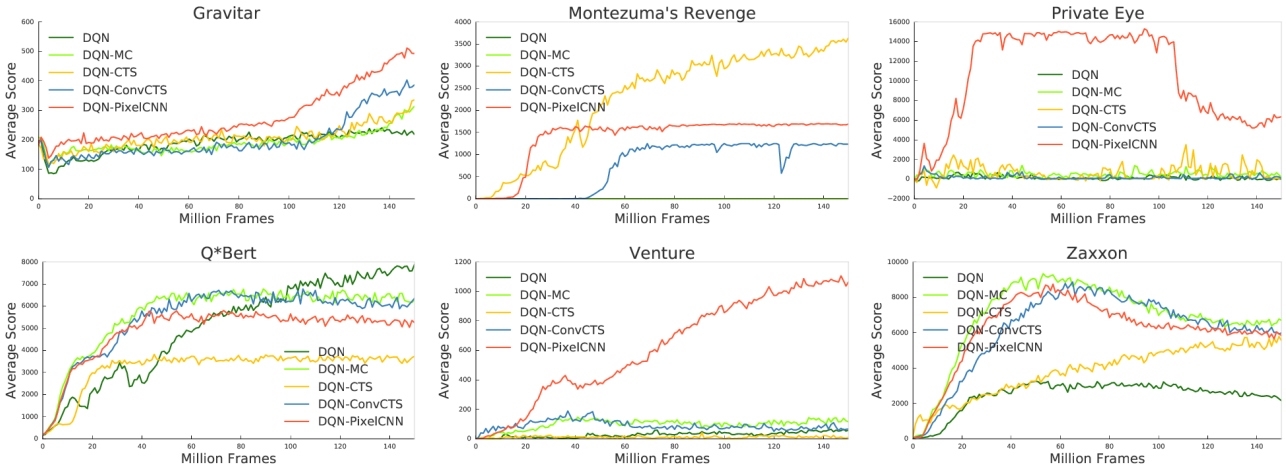


*Figure 14.* Comparison of DQN, DQN-CTS, DQN-ConvCTS and DQN-PixelCNN training performance.

| Easy Exploration | | Hard Exploration | |
|---|---|---|---|
| Human-Optimal | Score Exploit | Dense Reward | Sparse Reward |
| Assault | Asterix | Beam Rider | Alien | Freeway |
| Asteroids | Atlantis | Kangaroo | Amidar | Gravitar |
| Battle Zone | Berzerk | Krull | Bank Heist | Montezuma's Revenge |
| Bowling | Boxing | Kung-Fu Master | Frostbite | Pitfall! |
| Breakout | Centipede | Road Runner | H.E.R.O. | Private Eye |
| Chopper Cmd | Crazy Climber | Seaquest | Ms. Pac-Man | Solaris |
| Defender | Demon Attack | Up n Down | Q*Bert | Venture |
| Double Dunk | Enduro | Tutankham | Surround | |
| Fishing Derby | Gopher | | Wizard of Wor | |
| Ice Hockey | James Bond | | Zaxxon | |
| Name this Game | Phoenix | | | |
| Pong | River Raid | | | |
| Robotank | Skiing | | | |
| Space Invaders | Stargunner | | | |

*Table 1.* A rough taxonomy of Atari 2600 games according to their exploration difficulty.

| | DQN | A3C-CTS | Prior. Duel | DQN-CTS | DQN-PixelCNN |
|---|---|---|---|---|---|
| Freeway | 30.8 | 30.48 | **33.0** | 31.7 | 31.7 |
| Gravitar | 473.0 | 238.68 | 238.0 | 498.3 | **859.1** |
| Montezuma's Revenge | 0.0 | 273.70 | 0.0 | **3705.5** | 2514.3 |
| Pitfall! | -286.1 | -259.09 | **0.0** | **0.0** | **0.0** |
| Private Eye | 146.7 | 99.32 | 206.0 | 8358.7 | **15806.5** |
| Solaris | 3,482.8 | 2270.15 | 133.4 | 2863.6 | **5501.5** |
| Venture | 163.0 | 0.00 | 48.0 | 82.2 | **1356.25** |

*Table 2.* Comparison with previously published results on hard exploration, sparse reward games. The compared agents are DQN (Mnih et al., 2015), A3C-CTS ("A3C+" in (Bellemare et al., 2016)), Prioritized Dueling DQN (Wang et al., 2016), and the basic versions of DQN-CTS and DQN-PixelCNN from Section 4. For our agents we report the maximum scores achieved over 150M frames of training, averaged over 3 seeds.

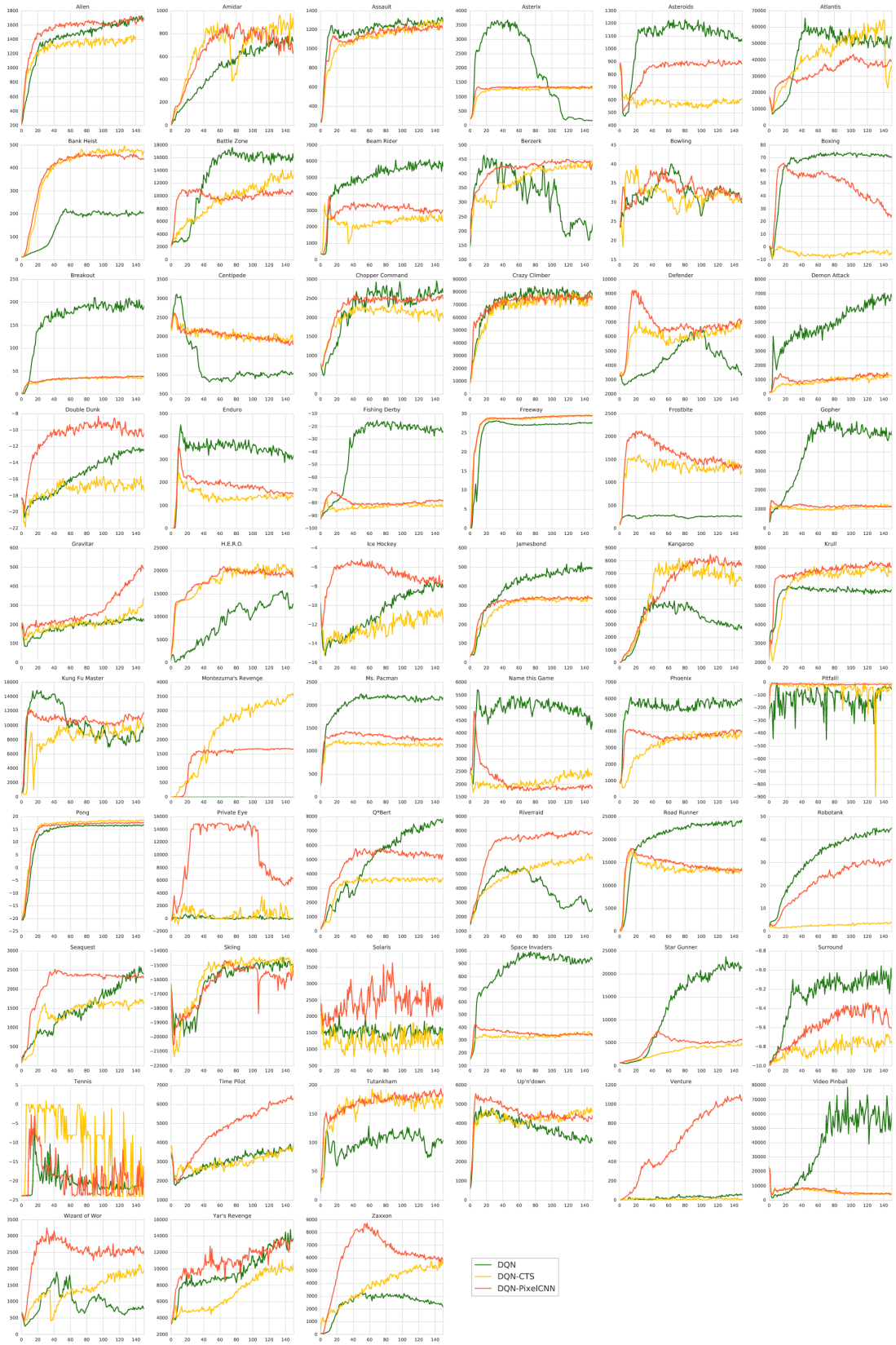


*Figure 15.* Training curves of DQN, DQN-CTS and DQN-PixelCNN across all 57 Atari games.

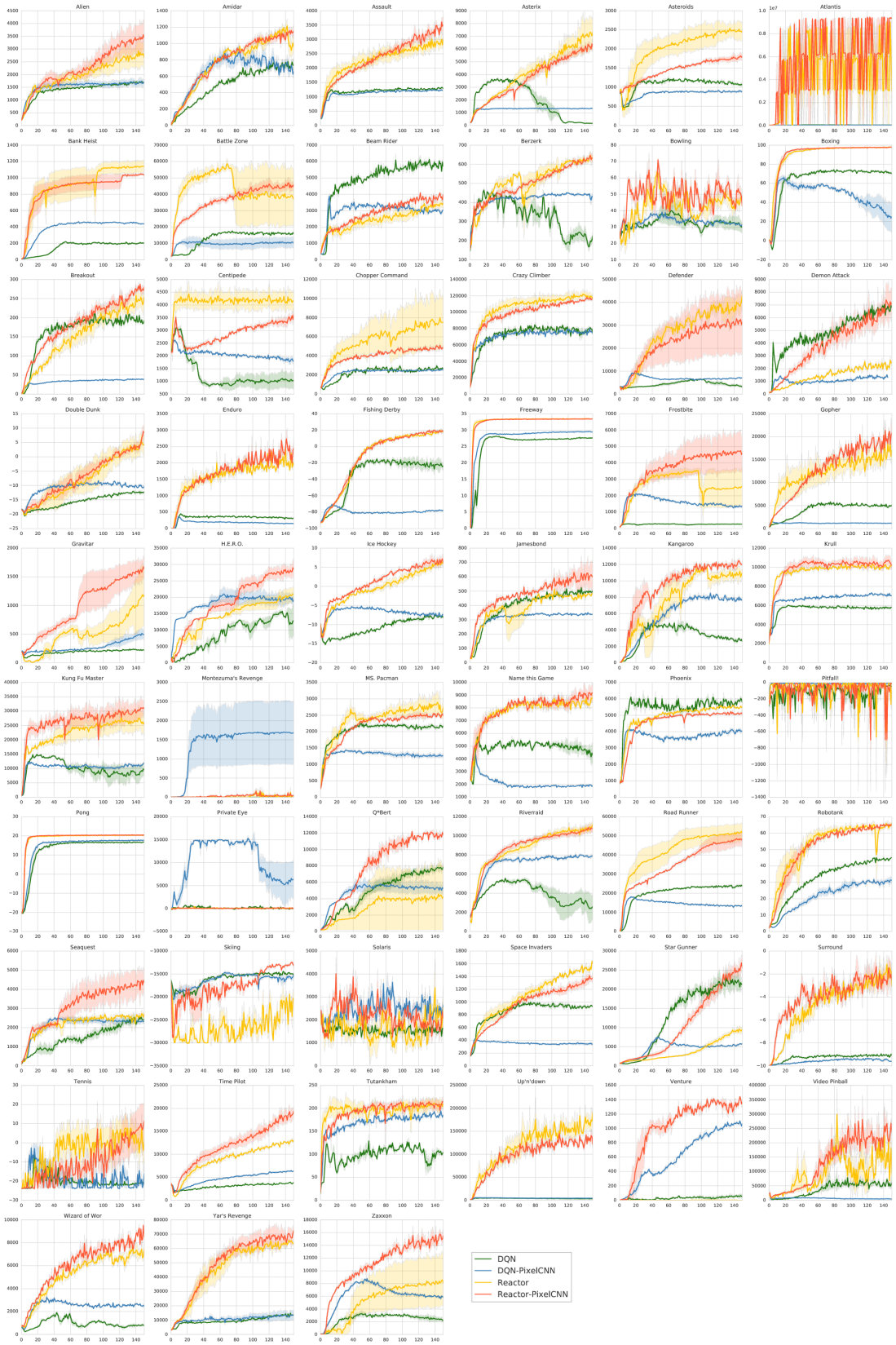


*Figure 16.* Training curves of DQN, DQN-PixelCNN, Reactor and Reactor-PixelCNN across all 57 Atari games.

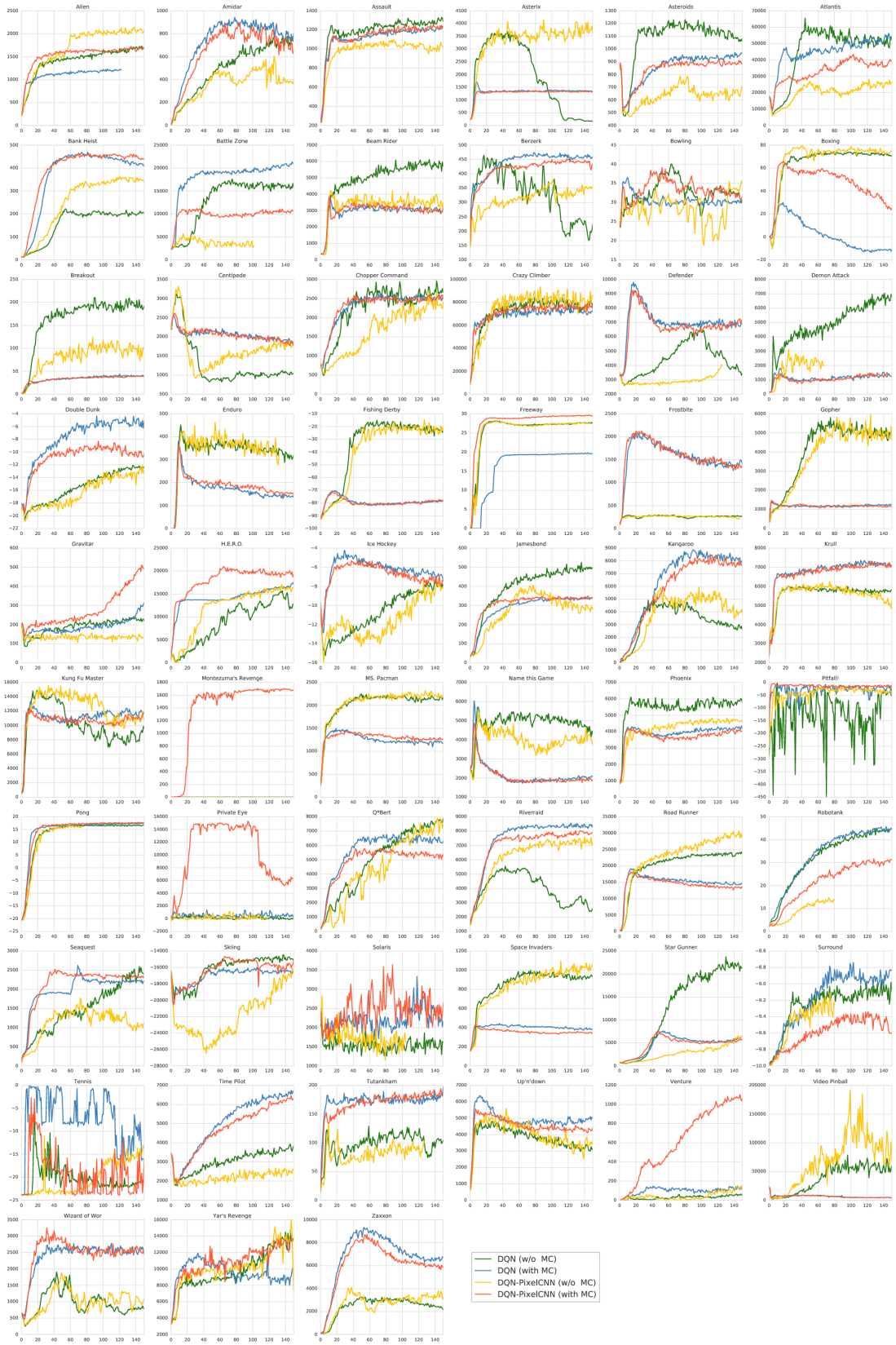


*Figure 17.* Training curves of DQN and DQN-PixelCNN, each with and without MMC, across all 57 Atari games.