A Mathematical Details and Proofs

In this section we provide the proofs of Proposition 3.1 and Proposition 3.2. The propositions are restated before each proof for convenience.

Proposition 3.1. *Let h be an invertible strictly increasing odd function which is strictly concave on $\mathbb{R}_+$ and define $\mathcal{T}_h$ by* (6)*. Then there exists a finite MDP where the fixed-point of $\mathcal{T}_h$ does not yield an optimal policy.*

Proof. We recall from (6) that the operator $\mathcal{T}_h$ is given by

$$(\mathcal{T}_h Q)(s,a) := \mathbb{E}_\rho \left[h\left(R_{t+1} + \gamma \max_{a'} (h^{-1} \circ Q)(S_{t+1}, a') \right) \,\middle|\, S_t = s, A_t = a \right]. \tag{1}$$

Hence an induced policy selects actions by $\arg\max_a h^{-1}(Q(s,a)) = \arg\max_a Q(s,a)$. We will now show that the optimal policy invariance property of $\mathcal{T}_h$ might break whenever the environment gives us a choice between a fully deterministic path and a path where the return variable is of maximum possible variance.

Let $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \rho)$ be a finite MDP with a non-terminal state s and suppose that $\mathcal{A}$ consists of two actions a, b. Suppose further that $\mathcal{R}$ and ρ are defined as follows: If we choose a in s, then ρ may send us to two different terminal states with equal probability, where we observe rewards 0 and $R > 0$ respectively. On the other hand if we select b, then ρ will send us deterministically to a single terminal state where we observe a reward r, which is defined by

$$h^{-1}\left(\frac{h(R)}{2} \right) < r < \frac{R}{2}. \tag{2}$$

To motivate the existence of such an r, note that h^{-1} is strictly convex on $\mathbb{R}_+$ with $h^{-1}(0) = 0$, since, by the condition of the proposition, h is strictly concave on $\mathbb{R}_+$ with $h(0) = 0$. Thus from Jensen's inequality we have

$$h^{-1}\left(\frac{h(R)}{2} \right) = h^{-1}\left(\frac{h(0)}{2} + \frac{h(R)}{2} \right) < \frac{h^{-1}(h(0))}{2} + \frac{h^{-1}(h(R))}{2} = R/2.$$

It is clear that an optimal policy chooses the non-deterministic path of a since the expected value is

$$0/2 + R/2 = R/2 > r.$$

However, in the transformed approach of (1) we find the expectation of each action in the values

$$h(R)/2 + h(0)/2 = h(R)/2$$

for a and $h(r)$ for b. Thus, given the way that r was defined in (2) we have

$$h^{-1}(h(r)) = r > h^{-1}(h(R)/2).$$

It follows that the fixed point inducing policy of (1), found by one iteration, suggests that we choose non-optimal $b \neq a$, which completes the proof of the proposition. □

The following lemma is used in the proof of Proposition 3.2 which is stated and proved directly after the lemma.

Lemma 2.1. *Let $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \rho)$ be a finite MDP. If η is any collection with measures of bounded first moments, then the induced Q-function $Q_{T^*\eta}$ of $T^*\eta$ can be expressed as*

$$Q_{T^*\eta}(s,a) = \mathbb{E}_\rho \left[R + \gamma \max_{a'} Q_\eta(S', a') \,\middle|\, s, a \right] = (\mathcal{T}^* Q_\eta)(s,a)$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Proof. The result follows from a simple reformulation by the finiteness of the MDP, linearity of integration over mixture distributions and a change of variables

$$\begin{aligned} Q_{T^*\eta}(s,a) &= \int_{\mathbb{R}} z \, d\,(T^*\eta)^{(s,a)} = \int_{\mathbb{R}} z \, d\left(\int_{\mathcal{R}\times\mathcal{S}} f_{r,\gamma}\#\eta^{(s',a^*)} \, d\rho(r,s' \mid s,a)\right)(z) \\ &= \int_{\mathcal{R}\times\mathcal{S}} \left(\int_{\mathbb{R}} z \, d\left(f_{r,\gamma}\#\eta^{(s',a^*)}\right)(z)\right) \, d\rho(r,s' \mid s,a) \\ &= \int_{\mathcal{R}\times\mathcal{S}} \left(\int_{\mathbb{R}} (r+\gamma z) \; d\eta^{(s',a^*)}(z)\right) \, d\rho(r,s' \mid s,a) \\ &= \mathbb{E}_\rho\left[R + \gamma \max_{a'} Q_\eta(S',a') \;\middle|\; s,a\right] = \left(\mathcal{T}^* Q_\eta\right)(s,a). \end{aligned}$$

This completes the proof of the lemma. □

Proposition 3.2. *Let ξ_0 be an initial collection of measures on J with supports contained in a closed bounded interval $I \subset J$. If we set*

$$\xi_k \coloneqq T_\varphi \xi_{k-1} = T_\varphi^k \xi_0$$

as the kth iteration of ξ_0 with respect to T_φ, then Q_k defined by

$$Q_k(s,a) \coloneqq \int_J \varphi^{-1}(w) \, d\xi_k^{(s,a)}(w)$$

satisfies the Bellman iteration $Q_k = \mathcal{T}^* Q_{k-1}$.

Proof. We recall from Definition 3.1 that

$$(T_\varphi \xi)^{(s,a)} \coloneqq \int_{\mathcal{R}\times\mathcal{S}} \left(\varphi \circ f_{r,\gamma} \circ \varphi^{-1}\right) \# \xi^{(s',a^*)} \, d\rho(r,s' \mid s,a),$$

where $a^* = a^*(s')$ is chosen according to

$$a^*(s') \in \left\{ \arg\max_{a'} \int_J \varphi^{-1}(w) \, d\xi^{(s',a')}(w) \right\}.$$

Since $\varphi \circ f_{r,\gamma} \circ \varphi^{-1}$ is continuous, hence measurable, and φ^{-1} by definition is integrable with respect to all involved measures, T_φ is well-defined. Moreover, given any collection ζ over J or $\mathbb{R}$ and any measurable function $f\colon J \to \mathbb{R}$ or $f\colon \mathbb{R} \to J$, we define $f\#\zeta \coloneqq \{ f\#\zeta^{(s,a)} \}$ as a joint push-forward.

Put $\eta_k \coloneqq \varphi^{-1}\#\xi_k$ and note that $\xi_k = \varphi\#\eta_k$ for all k. This implies $\eta_k = \varphi^{-1}\#T_\varphi\left(\varphi\#\eta_{k-1}\right)$. Since we are dealing with finite MDPs, any integral over $\mathcal{R}\times\mathcal{S}$ with respect to ρ given (s,a) can be represented as a finite sum. So from the linearity of push-forwards and finite sums we have

$$\begin{aligned} \eta_k^{(s,a)} &= \varphi^{-1}\# \int_{\mathcal{R}\times\mathcal{S}} \left(\varphi \circ f_{r,\gamma} \circ \varphi^{-1}\right) \# \left(\varphi\#\eta_{k-1}\right)^{(s',a^*)} \, d\rho(r,s' \mid s,a) \\ &= \left(\varphi^{-1} \circ \varphi\right) \# \int_{\mathcal{R}\times\mathcal{S}} f_{r,\gamma}\#\eta_{k-1}^{(s',a^*)} \, d\rho(r,s' \mid s,a) = \int_{\mathcal{R}\times\mathcal{S}} f_{r,\gamma}\#\eta_{k-1}^{(s',a^*)} \, d\rho(r,s' \mid s,a). \end{aligned}$$

Moreover, by a change of variables and the assumption that φ^{-1} is integrable, we find

$$\int_J \varphi^{-1}(w) \, d\xi_k^{(s,a)}(w) = \int_J \varphi^{-1}(w) \, d\left(\varphi\#\eta_k^{(s,a)}\right)(w) = \int_{\mathbb{R}} z \, d\eta_k^{(s,a)}(z).$$

Hence η_k corresponds precisely to iterations of the DRL optimality operator in (2), *i.e.*, $\eta_k = T^*\eta_{k-1}$ with an initial collection $\eta_0 = \varphi^{-1}\#\xi_0$. In particular, the induced Q-function sequence Q_k of ξ_k equals

$$Q_k(s,a) := \int_J \varphi^{-1}(w)\, d\xi_k^{(s,a)}(w) = \int_{\mathbb{R}} z\, d\eta_k^{(s,a)}(z) = Q_{\eta_k}(s,a).$$

Thus by Lemma 2.1, we find

$$Q_k(s,a) = Q_{\eta_k}(s,a) = Q_{T^*\eta_{k-1}}(s,a) = \left(\mathcal{T}^* Q_{\eta_{k-1}}\right)(s,a) = \left(\mathcal{T}^* Q_{k-1}\right)(s,a),$$

which implies that they are iterates of the Bellman operator in (3). It is now well-known from classical theory that since our MDP is finite with discount $\gamma < 1$ and ξ_0 induces a bounded Q-function Q_0, the subsequent iterates Q_k are bounded and will converge uniformly to the optimal value function Q^* as $k \to \infty$ (Szepesvári 2010). □

B Atari MDPs, Architecture and Hyperparameters

In this section we present implementation details of our C2D-Atari experiments. This includes the computational details for our networks and values for used hyperparameters. The Atari implementation of C2D used the ALE C++ library for simulations by encapsulating ALE in an environment class, which also handled storage of observed transitions in a circular replay buffer. Network computations and training was done in Python with TensorFlow 2.X by using a thin wrapper for the data exchange with C++.

B.1 Atari MDPs

Our Atari 2600 MDPs were induced by the settings in Table 1. Following DQN we repeat each action 4 times in ALE and represent an observation by max-pooling ALE screens #3 and #4 in the generated sequence of 4 grayscaled screens. This is done in order to remove flicker due to partial screen updates. Observation frames, rescaled to 84×84 pixels, are then stacked 4 times and rolled by each step taken by an agent to form states of tensor dimensions $(4, 84, 84)$. Thus, states now represent short temporal views of game dynamics. To induce non-determinism we use sticky actions, which is handled internally in ALE. In addition, every episode is terminated after roughly 30min (108k frames), which is the default termination time for single-actor algorithms.

Parameter	**Value**
ALE version	6.2
ALE color spectrum	Grayscaled
ALE frame dimensions	84×84
Max episode length	27k steps (108k frames)
Action repetition	4
State observation stacking	4
Terminal on life loss	True
Sticky actions	0.25
Discount γ	0.99

Table 1: Atari specific settings.

B.2 Architecture

The overall architecture of Figure 1 for the Atari implementation follows that of DQN. States are represented by a sequence of 4 max-pooled observations and given to an encoding function ψ consisting of three convolutional layers, interleaved by batch normalization + ReLU, and a ReLU-activated dense layer which computes a 512-feature sized vector $\psi(s)$. The encoded state $\psi(s)$ is then passed to a probability network $\mathbf{p}$ that computes $|\mathcal{A} \times N|$ probabilities $\mathbf{p}\left(\psi(s)\right)$ through a dense layer with softmax activation. The probabilities are also passed to a dense embedding layer ϕ which computes a 512-sized vector $\mathbf{e} \coloneqq \phi\left(\mathbf{p}\left(\psi(s)\right)\right)$, again with ReLU-activation. The embedding $\mathbf{e}$ and the feature vector $\psi(s)$ are concatenated to form an 1024-input vector, which is fed to an atom network $\mathbf{x}$. The resulting atoms $\mathbf{z}(\mathbf{e}, \psi(s))$ are computed by a dense layer of $|\mathcal{A} \times N|$ units with activation $\alpha \tanh(x/c)$. The combined output of the network is $(\mathbf{p}, \mathbf{x})$ which represents our discrete distributions, one for each available action at the current state s.

B.3 C2D Settings

Finally, Table 2 lists all other settings and hyperparameters used by C2D in our experiments.

Parameter	**Value**
TensorFlow version	2.5
Number of atoms N (IQN)	32
Optimizer (IQN)	ADAM
Learning rate (IQN)	$0.5 \cdot 10^{-4}$
ADAM epsilon (IQN)	$3.125 \cdot 10^{-4}$
ADAM global clip norm	10.0
Training volume (DQN)	50M steps (200M frames)
Replay buffer (DQN)	1M transitions (s, a, r, s')
Random history (Dopamine)	20k steps (80k frames)
Initial training ε (Dopamine)	1.0
Minimum training ε (Dopamine)	0.01
ε-decay schedule ($1.0 \to$ min. ε) (Dopamine)	250k steps (1M frames)
Target network update frequency (Dopamine)	8k steps (32k frames)
Training frequency (DQN)	Every 4th step
Batch size (Dopamine)	32 buffered transitions (s, a, r, s'), uniformly sampled.
Loss	Cramér distance $\int (F_\mu - F_\nu)^2 \, dw$
$h(x)$	$\operatorname{sign}(x)\left(\left(\sqrt{1+\|x\|}-1\right)+\epsilon x\right)$, $\epsilon = 0.001$
$h^{-1}(x)$	$\operatorname{sign}(x)\left(\left(\frac{\sqrt{1+4\epsilon(\|x\|+1+\epsilon)}-1}{2\epsilon}\right)^2-1\right)$, $\epsilon = 0.001$
Transformation scaling β	1.99
Homeomorphism $\varphi(x)$	$\beta h(x)$
Inverse $\varphi^{-1}(x)$	$h^{-1}(x/\beta)$
Support scale initialization	$\alpha = 50.0$ (trainable)
Internal output scaling c	5.0
Atom activation function	$\alpha \tanh x/c$

Table 2: C2D settings for our Atari experiments.

C Atari Mean Scores for C2D (Sticky Action)

Game	10M	50M	100M	200M
alien	613.0 (12.0)	1497.1 (214.7)	2826.9 (253.2)	4111.3 (340.8)
amidar	94.9 (1.7)	453.2 (25.3)	664.9 (77.1)	816.4 (68.3)
assault	2170.6 (297.4)	3162.6 (205.0)	3819.0 (541.3)	4997.0 (508.2)
asterix	2075.1 (191.4)	9818.6 (533.3)	19882.2 (3322.0)	62677.6 (8152.6)
asteroids	762.6 (51.1)	797.2 (34.9)	921.4 (66.8)	1075.0 (96.6)
atlantis	8487.2 (521.1)	976179.7 (8946.6)	927126.9 (16110.0)	940490.4 (19823.0)
bankheist	17.4 (3.9)	607.7 (100.9)	934.8 (11.5)	1040.3 (57.6)
battlezone	3307.9 (364.7)	29485.5 (949.4)	34422.8 (1031.1)	42040.7 (617.7)
beamrider	4326.7 (477.2)	8496.0 (539.6)	9473.1 (806.9)	10797.2 (777.7)
berzerk	576.1 (7.9)	767.3 (3.8)	791.3 (7.2)	831.6 (9.1)
bowling	26.6 (3.5)	81.7 (7.1)	88.6 (10.7)	97.9 (11.7)
boxing	-25.7 (0.8)	52.4 (1.3)	91.6 (6.2)	96.0 (1.6)
breakout	21.7 (12.6)	276.8 (14.3)	325.1 (5.0)	370.2 (8.4)
centipede	9298.1 (1314.1)	53349.9 (3118.4)	74364.0 (4338.5)	105440.0 (9567.5)
choppercommand	565.3 (77.5)	658.6 (218.2)	1272.6 (1610.7)	3118.6 (4640.5)
crazyclimber	102366.9 (1650.4)	124196.7 (1221.0)	134382.7 (2569.4)	142029.7 (30.9)
demonattack	7502.7 (1316.7)	30118.8 (4122.4)	55757.9 (8478.9)	95685.1 (196.6)
doubledunk	-22.9 (0.5)	-21.2 (0.9)	-21.7 (0.5)	-18.4 (2.7)
enduro	175.6 (71.2)	1297.6 (63.0)	1714.3 (59.2)	1885.2 (160.4)
fishingderby	-90.2 (0.0)	15.1 (0.4)	18.7 (0.9)	20.5 (0.2)
freeway	17.9 (0.8)	33.1 (0.1)	33.4 (0.0)	33.5 (0.1)
frostbite	660.9 (289.3)	3254.0 (173.8)	3483.5 (225.5)	4023.9 (135.6)
gopher	678.2 (171.9)	19015.8 (1765.9)	25183.6 (6676.9)	38405.3 (5558.2)
gravitar	174.6 (9.8)	605.9 (81.0)	765.9 (46.7)	975.9 (348.2)
hero	3385.2 (199.2)	15007.8 (2516.9)	21247.7 (495.4)	29424.4 (2582.7)
icehockey	-15.2 (0.2)	-9.3 (1.1)	-5.6 (0.9)	-5.0 (0.7)
jamesbond	248.7 (13.8)	871.3 (193.4)	3642.1 (2146.0)	9270.5 (3173.7)
kangaroo	2074.7 (865.0)	10271.2 (75.9)	10526.7 (741.3)	11512.9 (1308.5)
krull	2237.3 (177.3)	7705.5 (129.7)	8131.3 (152.2)	8713.5 (297.2)
kungfumaster	21452.6 (1944.8)	29539.3 (2798.6)	35743.8 (2124.8)	41563.1 (1753.4)
montezumarevenge	0.0 (0.0)	7.0 (1.4)	18.4 (16.2)	39.1 (15.7)
mspacman	1447.5 (173.2)	3592.8 (234.5)	4896.7 (269.6)	5508.9 (784.2)
namethisgame	2609.4 (339.0)	8001.1 (1213.2)	11231.4 (356.9)	13181.4 (418.4)
phoenix	5555.5 (1293.5)	13435.8 (1698.5)	20077.6 (3580.8)	22942.8 (2682.8)
pitfall	-43.3 (18.4)	-72.4 (18.2)	-109.4 (60.4)	-177.8 (97.2)
pong	-19.7 (1.2)	11.0 (3.7)	15.9 (1.4)	18.2 (1.3)
privateeye	131.8 (47.5)	-87.8 (51.8)	-41.7 (91.9)	5513.0 (7933.4)
qbert	1032.6 (143.9)	9033.6 (1996.4)	15585.1 (134.6)	20328.0 (2459.4)
riverraid	3391.4 (182.8)	12805.4 (194.0)	16284.8 (391.7)	19086.4 (146.5)
roadrunner	22377.3 (1106.9)	43889.8 (992.8)	45766.6 (1442.3)	48594.9 (916.6)
robotank	6.2 (1.7)	26.9 (3.9)	42.5 (5.1)	61.2 (2.6)
seaquest	433.9 (105.2)	3395.6 (259.5)	4072.1 (183.7)	4193.9 (191.7)
skiing	-22792.1 (536.1)	-27025.1 (2036.8)	-30319.5 (158.0)	-30591.0 (47.2)
solaris	1443.2 (81.3)	1152.0 (189.5)	1349.2 (312.8)	1538.4 (528.1)
spaceinvaders	633.2 (31.8)	1039.5 (25.8)	1333.7 (66.2)	1684.0 (118.5)
stargunner	1206.4 (84.3)	50894.5 (4573.2)	61879.0 (4897.6)	90312.8 (12514.9)
tennis	-23.8 (0.0)	-23.3 (0.8)	-23.8 (0.0)	-23.8 (0.0)
timepilot	1191.2 (63.8)	4322.8 (304.1)	6214.5 (546.3)	8156.5 (266.9)
tutankham	59.8 (29.6)	52.2 (24.6)	74.6 (17.8)	157.6 (25.5)
upndown	7222.4 (409.3)	16852.6 (394.9)	20112.0 (721.4)	25582.3 (1798.3)
venture	17.1 (5.0)	9.3 (12.5)	4.1 (5.0)	1.9 (3.3)
videopinball	23182.9 (2730.6)	199886.5 (11527.9)	230849.9 (11523.5)	342055.9 (76220.0)
wizardofwor	468.1 (48.0)	2901.9 (1149.8)	7466.4 (2461.0)	14566.2 (2521.3)
yarsrevenge	9673.8 (188.8)	20966.4 (14486.5)	52049.1 (34059.4)	87772.2 (3939.9)
zaxxon	962.5 (223.6)	6574.6 (1478.2)	11183.4 (198.1)	12113.7 (446.0)

Table 3: Sticky action raw scores for C2D over all 55 Atari games at various iterations in the training phase as suggested by (Machado et al. 2018). The scores, which are derived from moving averages over 5M frames for each seed, are computed as the mean over all available seeds with one standard deviation included in parentheses.

D Mean Learning Curves

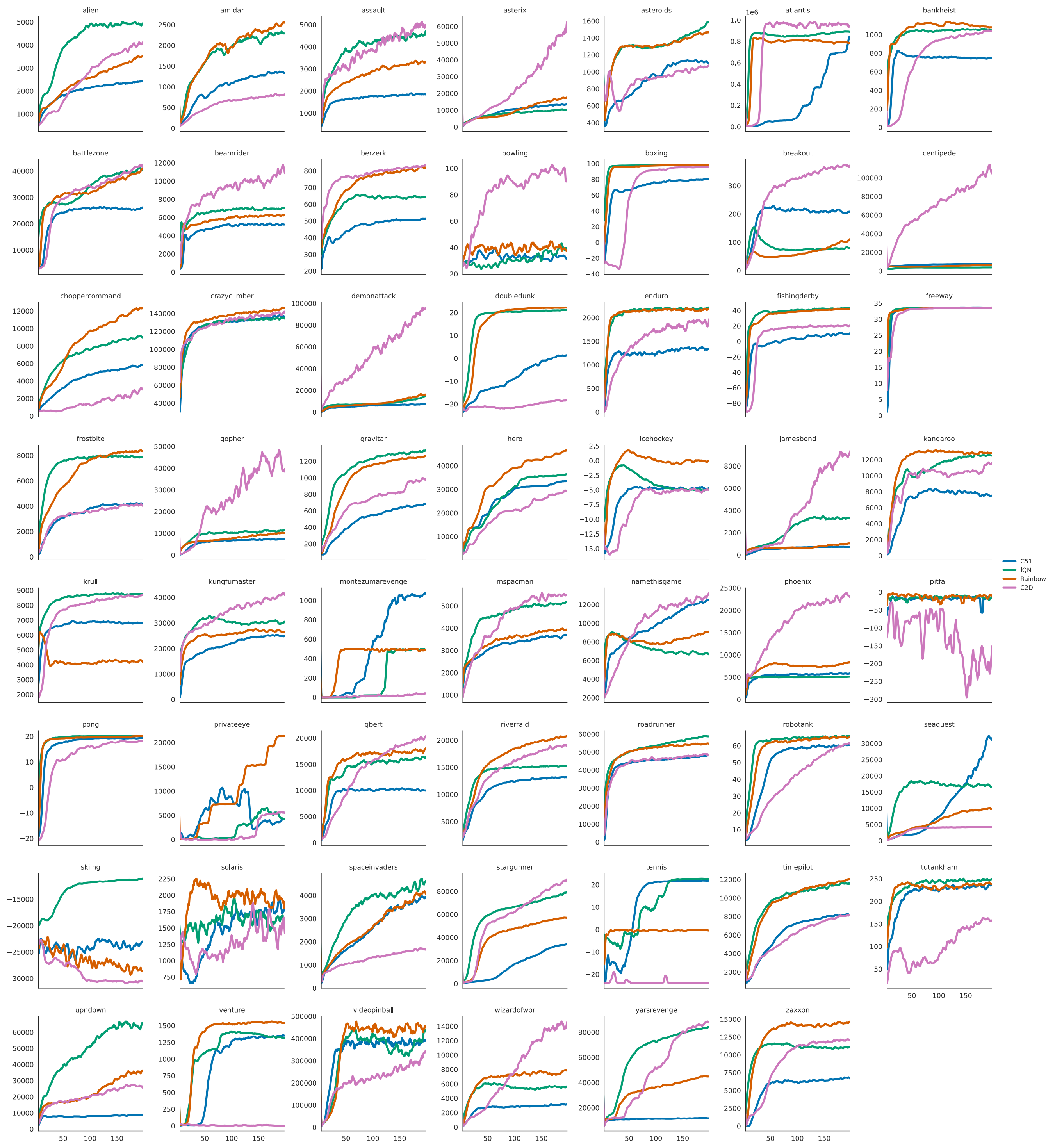

Figure 1: Mean learning curves for all games. Scores are computed by moving averages over 5M frames and the curves are the mean progression over all seeds. C2D used 3 seeds, and C51, IQN and Rainbow used 5 seeds (Dopamine, 2020).

E Mean Curves for the Maximal Possible Support

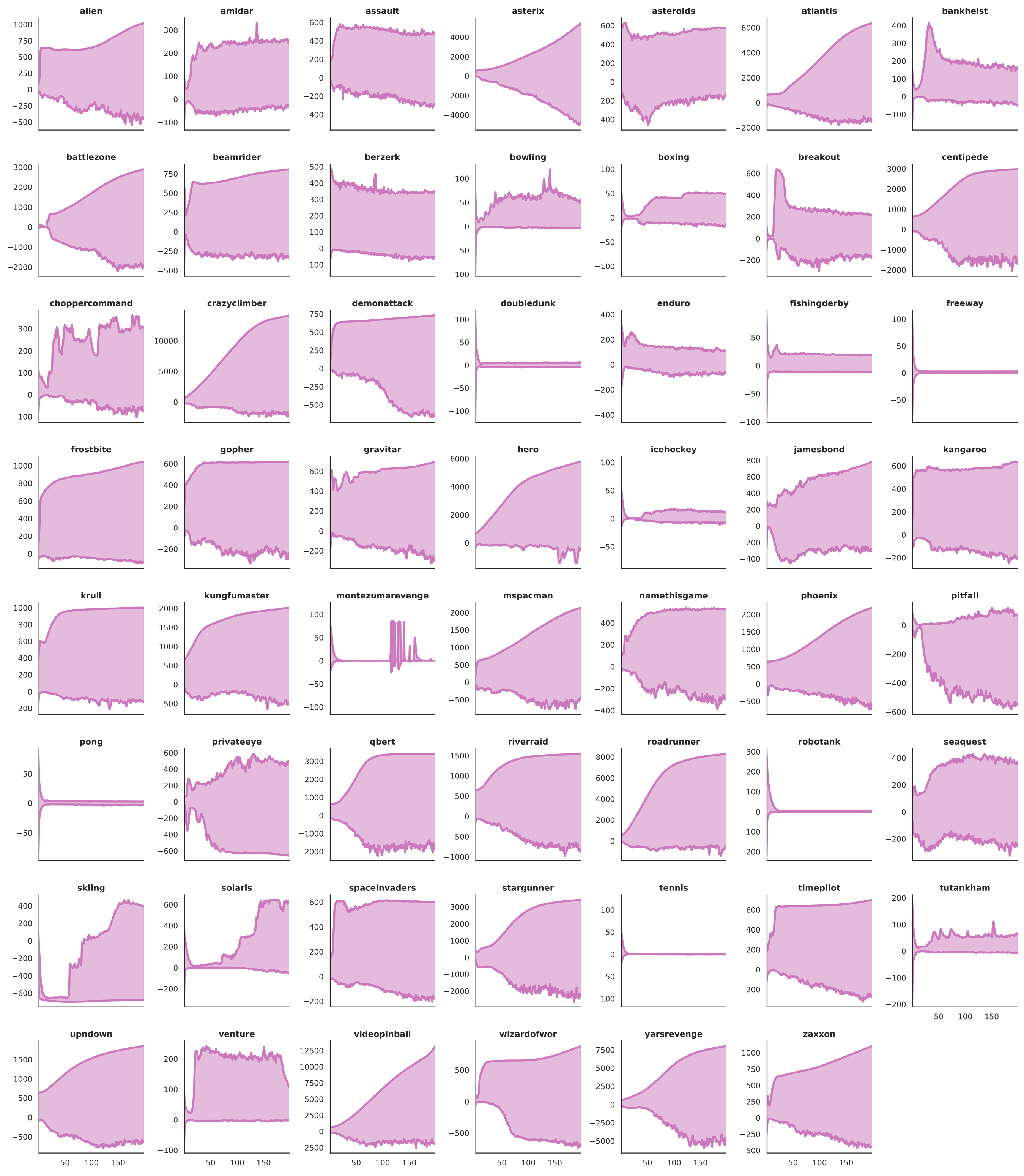

Figure 2: Mean curves for the maximal possible support of C2D over all frames across all 55 games. The larger value is taken as the maximum predicted atom over all actions on the last 1M frames, and the lower as the minimum.

F Sampling Efficiency: Mean and Median

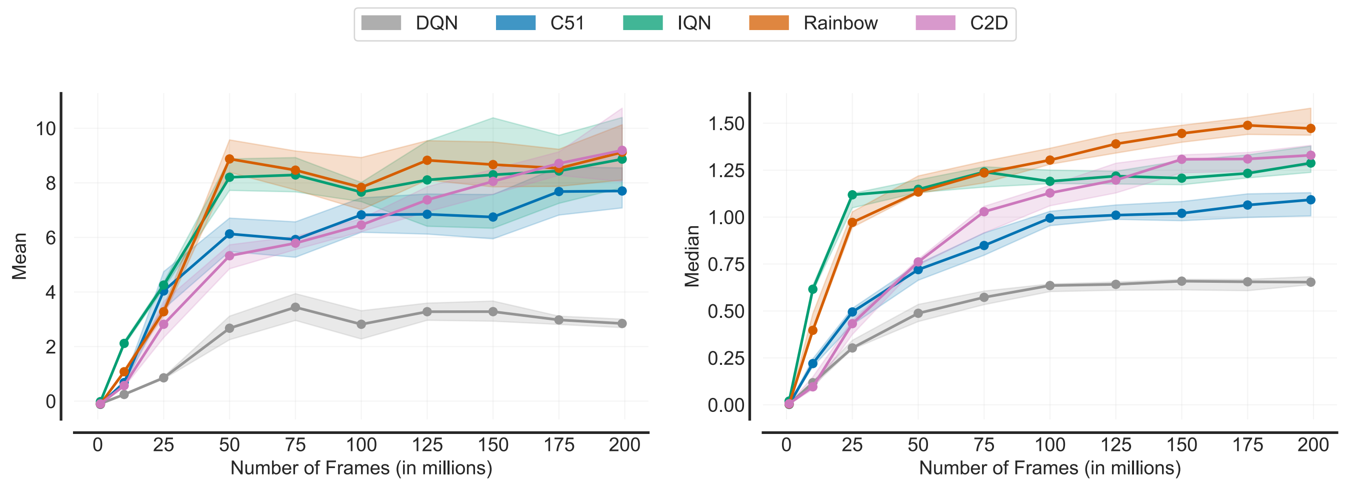

Figure 3: Aggregate mean and median metrics on Atari-200M over 55 games. The metrics are computed in accordance to the performance profiling methods given in (Agarwal et al. 2021). Dopamine results are computed over 5 runs and C2D used 3 runs.