Conjugated Discrete Distributions for Distributional Reinforcement Learning

Björn Lindenberg[*], Jonas Nordqvist, Karl-Olof Lindahl

Department of Mathematics, Linnæus University, Växjö, Sweden

Abstract

In this work we continue to build upon recent advances in reinforcement learning for finite Markov processes. A common approach among previous existing algorithms, both single-actor and distributed, is to either clip rewards or to apply a transformation method on Q-functions to handle a large variety of magnitudes in real discounted returns. We theoretically show that one of the most successful methods may not yield an optimal policy if we have a non-deterministic process. As a solution, we argue that distributional reinforcement learning lends itself to remedy this situation completely. By the introduction of a conjugated distributional operator we may handle a large class of transformations for real returns with guaranteed theoretical convergence. We propose an approximating single-actor algorithm based on this operator that trains agents directly on unaltered rewards using a proper distributional metric given by the Cramér distance. To evaluate its performance in a stochastic setting we train agents on a suite of 55 Atari 2600 games using sticky-actions and obtain state-of-the-art performance compared to other well-known algorithms in the Dopamine framework.

1 Introduction

In recent years, the resurgence of the distributional perspective in modern algorithms for reinforcement learning (RL) has represented a paradigm shift. Initially only used in risk-sensitive methods with parameterized continuous functions or in theoretical analysis, the approach of combining deep learning with distributional learning of return variables has been empirically proved to give superior performance. This is possibly due to a much richer set of predictions than value-based algorithms and different learning trajectories under gradient descent (Bellemare, Dabney, and Munos 2017; Lyle, Bellemare, and Castro 2019). As such, *distributional reinforcement learning* (DRL) is now the foundation of many state-of-the-art RL-algorithms such as the quantile functions of (Dabney et al. 2018a,b; Yang et al. 2019).

Transformations Another prevalent method in modern algorithms is to shape or transform an underlying *Markov decision process* (MDP) in order to improve performance. This

---
[*] Correspondence: bjorn.lindenberg@lnu.se
Appendix: https://github.com/bjliaa/c2d

is in itself an old idea in reinforcement learning. Transformations are particularly important for algorithms that are supposed to function over a wide variety of different environments with varying reward magnitudes, where adaptation to different settings is a common way to benchmark the robustness and ability of an algorithm.

Ideally, one would want the fixed points of the Bellman operators in both the original MDP and the transformed MDP to induce the same optimal policy, *i.e.*, to obtain *optimal policy invariance*. Early attempts at shaping rewards directly where shown to require specific forms in terms of impractical potential functions if we were to keep the invariance (Ng, Harada, and Russell 1999). A much more general approach is to consider non-linear equations that correspond to transformed Bellman operators. An investigation of such models and operator classes can be found in (van Hasselt et al. 2019). Regardless, many modern algorithms still employ reward shaping by clipping rewards for improved performance, even though this implies that the underlying MDP may never be solved in a theoretical sense.

An attempt to learn on unaltered rewards was presented in (Pohlen et al. 2018), where an alternative Bellman operator $\mathcal{T}_h$ was introduced. The operator uses a function to generate updates with downscaled expectations. This allowed for deep Q-learning over environments with wildly different return magnitudes. At the time of writing, this method is used successfully by some of the most powerful known distributive algorithms (Kapturowski et al. 2018; Schrittwieser et al. 2020; Badia et al. 2020). However, the proof of invariance for $\mathcal{T}_h$ in (Pohlen et al. 2018) required fully deterministic MDPs and it was left as an open question whether a similar property holds in a stochastic setting.

A New Distributional Operator We show in our work that the previously mentioned invariance for $\mathcal{T}_h$ may not hold true if we are given a non-deterministic MDP. Specifically, we show that the fixed-point policy similarity may break for a class of operators which includes $\mathcal{T}_h$. We show that this problem is rectified in DRL by the introduction of a new generalized operator T_φ, the so-called *conjugated distributional operator*. The operator, which transforms random outcomes by a chosen homeomorphism, can properly handle all its transformations while keeping the invariance property intact.

Our Algorithm To test the effects of training on unaltered rewards in DRL, we propose in the context of approximate DRL a single-actor algorithm *conjugated discrete distributions* (C2D). The algorithm extends C51 in (Bellemare, Dabney, and Munos 2017) by using discrete distributions with freely moving supports and learns conjugate distributions by sampling T_φ. The extension still implies sampling done through distributional DQN, but instead of the fixed support and projection operator of C51, C2D uses parameterized embedded *probability functions* and *atom functions* to represent discrete measures. Moreover, since supports are no longer shared between measures, C2D replaces the cross-entropy loss of C51 by the squared *Cramér distance*. This implies a DRL-algorithm that does minimization directly against a proper distributional metric.

Evaluation on Stochastic Games The algorithm was evaluated on a suite of 55 Atari 2600 games where the use of sticky actions (Machado et al. 2018) induced non-deterministic MDPs. For an "apples-to-apples" comparison with other algorithms we used the Dopamine framework protocol (Castro et al. 2018), where all involved agents were trained using a joint set of hyperparameters, including the sticky action probability. In the evaluation, C2D obtained state-of-the-art performance, comparable to Rainbow (Hessel et al. 2018), with a significant performance improvement over C51.

Organization This paper is organized as follows. In Section 2 we state the foundations of DRL through the lens of measure theory, which is the established framework of (Rowland et al. 2018). In Section 3, this allows for a much clearer picture of the connection between related work and our theoretical results in terms of the pushforward functions used to define our operator. We close Section 3 by discussing the three pillars of our C51 extension for approximate DRL. These are comprised of; *distributional DQN* to sample operations, *discrete measures* for distributional approximations and the *Cramér distance* combined with a method for its computation. In subsequent sections we then explicitly present implementation details of C2D and present evaluation results on Atari with an ending discussion. A comprehensive view of graphs, data and parameters can be found in the appendix along with proofs.

2 Setting

Let $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$ be finite state, action and reward spaces. We consider an agent-environment interaction which produces random sequences

$$S_0, A_0, R_1, S_1, A_1, R_1, \ldots, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, \ldots$$

according to the usual feedback loop of reinforcement learning. Our model for the interaction is a finite MDP $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \rho)$, where ρ is the transition kernel which maps $(s, a) \in \mathcal{S} \times \mathcal{A}$ to joint distributions $\rho(r, s' \mid s, a)$ over $\mathcal{R} \times \mathcal{S}$. In addition, the agent may sample its actions according to a stationary policy π which maps states $s \in \mathcal{S}$ to distributions $\pi(s)$ over $\mathcal{A}$.

2.1 Distributional Reinforcement Learning

Given a policy π and a state-action pair (s, a) at time t we have a future discounted return in the form of the random variable

$$Z_\pi(s, a) = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\Bigg|\, S_t = s, A_t = a , \quad (1)$$

where $\gamma \in (0, 1)$ is a fixed discount factor.

Denote by $\eta_\pi^{(s,a)} \in \mathcal{P}(\mathbb{R})$ the probability distribution of $Z_\pi(s, a)$ and consider the *collection* η_π of all such probability distributions as the image of $(s, a) \mapsto \eta_\pi^{(s,a)}$. Then the expected return for any state-action pair can be found as the first moment

$$Q_\pi(s, a) := \mathbb{E}\left[Z_\pi(s, a)\right] = \int_\mathbb{R} z \, d\eta_\pi^{(s,a)}(z),$$

i.e., the inherent expected value of being in some state s, taking action a while following π.

We note that any collection η of probability distributions with bounded first moments defines a bounded Q-function, say Q_η where $Q_\eta(s, a) := \int_\mathbb{R} z \, d\eta^{(s,a)}(z)$. It follows that we can define a *greedy* policy with respect to η that selects any action a^* in $\arg\max_a Q_\eta(s, a)$.

Given a probability distribution $\mu \in \mathcal{P}(\mathbb{R})$ and a measurable function $f : \mathbb{R} \to \mathbb{R}$, the *pushforward measure* $f\#\mu \in \mathcal{P}(\mathbb{R})$ is defined by $f\#\mu(A) := \mu\left(f^{-1}(A)\right)$, for all Borel sets $A \subseteq \mathbb{R}$. We may think of the pushforward measure $f\#\mu$ as obtained from μ by shifting the support of μ according to the map f. When given a real reward r and a probability measure μ of some random variable Z we denote the pushforward measure arising from adding r to the γ-discounted value of Z by $f_{r,\gamma}\#\mu$, where $f_{r,\gamma}(z) := r + \gamma z$. For brevity we denote any translation $r + z$ by f_r.

The framework introduced in (Bellemare, Dabney, and Munos 2017; Rowland et al. 2018) defines the *Bellman distributional optimality operator* T^* on any (s, a)-indexed collection η of probability measures by

$$(T^*\eta)^{(s,a)} := \int_{\mathcal{R} \times \mathcal{S}} f_{r,\gamma}\#\eta^{\left(s', a^*(s')\right)} \, d\rho(r, s' \mid s, a), \quad (2)$$

where $a^*(s')$ is any action in $\arg\max_{a'} Q_\eta(s', a')$. The mixture distribution of (2) can be seen as the "expected" pushforward measure found while acting greedily with respect to η. It can be shown that T^* in (2) is not a contraction map in any metric space of distributional collections (Bellemare, Dabney, and Munos 2017). This is in contrast to the contraction map on bounded functions given by the *Bellman optimality operator*

$$(\mathcal{T}^*Q)(s, a) := \mathbb{E}_\rho\left[R + \gamma \max_{a'} Q(S', a') \,\Big|\, s, a\right]. \quad (3)$$

Regardless we have the following result.

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*

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

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

Proof. In Appendix. □

Hence, the induced Q-function sequence Q_k of $\eta_k :=$ $(T^*)^k \eta_0$ satisfies $\forall\, (s,a)$,

$$Q_{k+1}(s,a) = \int_{\mathbb{R}} z\, d\eta_{k+1}^{(s,a)}(z) = (\mathcal{T}^* Q_k)(s,a). \quad (4)$$

Indeed, given any starting collection η_0 with bounded moments we find that Q_k converges to the *optimal Q-function* Q^* with respect to the uniform norm as $k \to \infty$, and generates an *optimal policy* $\pi^*(s) \in \arg\max_a Q^*(s,a)$ (Bertsekas and Tsitsiklis 1996; Szepesvári 2010; Bellemare, Dabney, and Munos 2017; Sutton and Barto 2018).

3   Our Approach and Related Work

We begin this section by motivating and defining a generalized *conjugated optimality operator* for DRL, which is key to our single-actor algorithm. We close the section by discussing our approach to approximate distributional reinforcement learning based on this operator.

Value-based Transformation Methods If we are training an agent which uses parameterized networks in an MDP that displays relatively high variance in both action-values and reward signals, then we may want to consider transformations of the involved quantities to improve stability. As an example, the original DQN-implementation in (Mnih et al. 2015) clips rewards of varying orders into the interval $[-1, 1]$. However, we note that this will encourage sequence-searching strategies instead of policies that can distinguish between actions that have relatively large differences in real returns. Thus, the clipping procedure may drastically change the underlying problem and its solution. An alternative approach presented in (Pohlen et al. 2018) uses a procedure with unaltered rewards, which is well-suited for deterministic environments and is successfully used by several well-known algorithms (Kapturowski et al. 2018; Schrittwieser et al. 2020; Badia et al. 2020). The method scales with an invertible strictly increasing odd function,

$$h(x) := \text{sign}(x)\left(\sqrt{1 + |x|} - 1\right) + \epsilon x, \quad 0 < \epsilon \ll 1, \quad (5)$$

which is (strictly) concave on $\mathbb{R}_+ := \{\, x \in \mathbb{R} \mid x \geq 0 \,\}$. The operation $\mathcal{T}_h Q$ on Q-functions is for each (s,a) then given by

$$\mathbb{E}_\rho\left[h\left(R + \gamma \max_{a'}(h^{-1} \circ Q)(S_{t+1}, a')\right) \,\Big|\, s, a\right]. \quad (6)$$

However, finding a proper operator for transformations when working with Q-functions directly may not be straight forward. As correctly pointed out in (Pohlen et al. 2018), iteration by (6) may not yield optimal policy invariance in a stochastic setting. Explicitly we have the following result.

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. In Appendix. □

Conjugated Optimality Operator The situation can be completely remedied in the setting of DRL by properly applying the conjugation in (6) on all outcomes, and by choosing actions that respect the transformation. More precisely, let Z, W be random variables satisfying $W = h(Z)$. If we put μ_Z and μ_W as their respective distributions, then $\mu_Z = h^{-1} \# \mu_W$ and $\mu_W = h \# \mu_Z$. It follows that if $Z' := r + \gamma Z$, then

$$W' := h(Z') = h(r + \gamma Z) = h\left(r + \gamma h^{-1}(W)\right)$$

obeys the law of $\left(h \circ f_{r,\gamma} \circ h^{-1}\right) \# \mu_W$. Moreover, for the expectation we obtain

$$\mathbb{E}[Z] = \int_{\mathbb{R}} z\, d\mu_Z = \int_{h(\mathbb{R})} h^{-1}(w)\, d\mu_W.$$

We are thus led to the following operator which works directly on collections of distributions for the transformed random variables.

Definition 3.1. Let $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \rho)$ be a finite MDP, J an open interval in $\mathbb{R}$ and $\varphi: \mathbb{R} \to J$ a homeomorphism. Furthermore, let $\xi := \{\, \xi^{(s,a)} \,\}$ be a collection of probability measures over J, such that φ^{-1} is integrable with respect to all measures in ξ. Put $g := \varphi \circ f_{r,\gamma} \circ \varphi^{-1}$. Then the *conjugated distributional optimality operator* T_φ on collections ξ is for each (s,a) defined by

$$(T_\varphi \xi)^{(s,a)} := \int_{\mathcal{R} \times \mathcal{S}} g \# \xi^{(s', a^*)}\, d\rho(r, s' \mid s, a),$$

where a^* is chosen uniformly such that

$$\int_J \varphi^{-1}(w)\, d\xi^{(s', a')}(w)$$

is maximized.

Since $g = \varphi \circ f_{r,\gamma} \circ \varphi^{-1}$ will always be continuous, hence measurable, and since we focus on distributions where φ^{-1} is integrable, T_φ is well-defined. The fact that T_φ now correctly mirrors iterations by T^* in (2) is stated by the following result.

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*

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

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

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

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

Proof. In Appendix. □

3.1 Approximate DRL

In order to probe the possible benefits of learning in conjugate space by T_φ in Definition 3.1, we present in this section the necessary approximation concepts needed for our single-actor algorithm. This includes the approximate operator method of C51 in (Bellemare, Dabney, and Munos 2017), where we estimate any operation of T_φ through a single observed transition. Moreover, compared to C51 we focus on representative measures taken from a larger class of discrete distributions where the aim is to train agents via a proper distributional metric.

Distributional DQN   The DQN-agent presented in (Mnih et al. 2015) employs a deep Q-Network $Q(s, a; \boldsymbol{\theta})$ to approximate action values. During learning, an older periodically updated clone $Q(s, a; \boldsymbol{\theta}^-)$ is used as a stabilizing critic on current estimations. That is, given uniformly drawn transitions (s, a, r, s') from a *replay buffer* the learning algorithm of DQN approximates (3) by computing gradients on squared *temporal difference errors* $\delta(s, a, r, s')^2$ where

$$\delta(s, a, r, s') := r + \gamma Q(s', a^*; \boldsymbol{\theta}^-) - Q(s, a; \boldsymbol{\theta}).$$

A similar approach for approximate DRL is through a parameterized distribution network $\xi(s, a; \boldsymbol{\theta})$. The network is coupled with an older copy $\xi(s, a; \boldsymbol{\theta}^-)$. Learning targets are taken as single sample approximations of $T_\varphi \xi(s, a; \boldsymbol{\theta}^-)$ and set to

$$\nu(r, s') := \left(\varphi \circ f_{r,\gamma} \circ \varphi^{-1}\right) \# \eta(s', a^*; \boldsymbol{\theta}^-). \quad (7)$$

Alternatively we can also choose the target $\Pi\nu(r, s')$ for some distributional projection Π. It follows that in order to extract gradients and push our current estimation towards the target we need a *distributional temporal difference error*

$$\Delta(s, a, r, s') := d\left(\nu(r, s'), \eta(s, a; \boldsymbol{\theta})\right), \quad (8)$$

where d is some divergence or distance on a suitable space of probability measures.

Discrete Distributions   In most real applications our distributional collections ξ will need to be implemented by a parameterized function that outputs approximate distributions $\xi(s, a, \boldsymbol{\theta})$. The original approach of C51 approximated distributions with discrete measures on a fixed set of 51 atomic classes, *i.e.*, *categorical reinforcement learning*. A larger set of discrete measures can be considered if we use a fixed number of quantiles together with quantile regression, which is done in (Dabney et al. 2018b,a) and with a fully parameterized quantile method in (Yang et al. 2019).

In our work we take a fully parameterized approach and focus on discrete measures of the form

$$\xi(s, a, \boldsymbol{\theta}) := \sum_{i=1}^{N} p_i \delta_{x_i},$$

where $\{p_i\}$ represents probabilities in the mixture distribution for the atomic classes $\{x_i\}$, and where N is some predefined number of atoms to use in the approximation.

Algorithm 1: Squared Cramér distance $\ell_2^2(\mu, \nu)$ for discrete distributions

Input Distributions $\mu := \sum_i p_i \delta_{x_i}$ and $\nu := \sum_j q_j \delta_{y_j}$
  # Extended signed measure support
  $(w_k) := \mathrm{sort}(x_1, \ldots, x_{n_\mu}, y_1, \ldots, y_{n_\nu})$
  $n \leftarrow n_\mu + n_\nu$
  $(\Delta w_k) := (w_{k+1} - w_k), \quad k = 1, 2, \ldots, n-1$
  for $k = 1$ **to** n **do**
    # Signed measure mass
    $r_k \leftarrow \begin{cases} p_i, & w_k = x_i \in (x_1, \ldots, x_{n_\mu}), \\ -q_j, & w_k = y_j \in (y_1, \ldots, y_{n_\nu}). \end{cases}$
    $P_k \leftarrow \sum_{l=1}^{k} r_l$
  end for
  $L \leftarrow \sum_{k=1}^{n-1} P_k^2 \Delta w_k$
Output L

Distributional Losses   The C51 algorithm employs parameterized distributions on a fixed support and implicitly use the *Kullback-Liebler divergence* (KL) in (8) combined with a projection of $\nu(r, s')$ onto the support. However, it is well-known that KL is not a proper metric, requires a common support and has no concept of the underlying geometry. With varying supports it is then only natural to consider distances in (8) that measures similarity in outcomes instead of likelihoods, *i.e.*, whose gradients induce probability mass transport while being sensitive to differences in outcomes between distributions. A common metric in this regard, used in both analysis and application, is the *Wasserstein distance* which is connected to Kantorovich's formulation of optimal transport cost (Villani 2008; Bellemare, Dabney, and Munos 2017). The distance generates biased sample gradients and is hard to apply directly, but indirect applications together with quantile regression have been used to great effect in (Dabney et al. 2018b,a; Yang et al. 2019).

However, in this paper we will focus on another proper metric called the *Cramér distance*, which in the univariate case squared is directly proportional to the *statistical energy distance* (Rizzo and Székely 2016):

Definition 3.2.   For probability measures μ, ν in $\mathcal{P}(\mathbb{R})$ we define the *Cramér distance* by

$$\ell_2(\mu, \nu) := \left(\int_{\mathbb{R}} (F_\mu(w) - F_\nu(w))^2 \; dw\right)^{1/2},$$

where F_μ, F_ν are the CDFs of each measure respectively.

The Cramér distance has been successfully used in analysis of convergence properties of (2), but also in modified form in real applications with linear approximations (Rowland et al. 2018; Bellemare et al. 2019; Lyle, Bellemare, and Castro 2019). As a proper metric it has a couple of attractive features that incorporates the underlying geometry: Like the Wasserstein distance it is *translation invariant*, *i.e.*, $\ell_2(f_r\#\mu, f_r\#\nu) = \ell_2(\mu, \nu)$, and *scale sensitive* such that $\ell_2\left((\gamma z)\#\mu, (\gamma z)\#\nu\right) = \sqrt{\gamma}\ell_2(\mu, \nu)$ for $\gamma > 0$. Additionally in the context of deep learning with parameterized networks, it has the desirable property of generating *unbiased*

gradient estimators when squared and used as a loss (Belle-mare et al. 2017).

Compared to previous implementations of the Cramér distance for discrete distributions, the atomic classes we consider are no longer equally spaced and may vary freely via parameterization. Hence similar to quantile methods this will give us a larger set of representative measures, but where the added degrees of freedom will also demand a slightly more involved distance computation. The solution used in our work is summarized in Algorithm 1, where we exploit the fact that $F_\mu - F_\nu$ represents the distribution function of the signed measure $\mu - \nu$.

4 Our Learning Algorithm

By combining the concepts in Section 3, we propose in this section *conjugated discrete distributions* (C2D) as an algorithm for approximate DRL using a chosen homeomorphism φ for conjugate pushforwards. The algorithm uses parameterized networks $\xi(s, a; \boldsymbol{\theta})$ to represent discrete measures $(\mathbf{p}, \mathbf{x}) \coloneqq \sum_{i=1}^{N} p_i \delta_{x_i}$ with N number of atoms, and our chosen homeomorphism φ dictates greedy actions according to a^* in Definition 3.1. During training we maintain an older clone $\xi(s, a; \boldsymbol{\theta}^-)$ together with the target approximating technique of (7) and we use Algorithm 1 to judge our current estimations. The learning algorithm is summarized in Algorithm 2.

Algorithm 2: Learning with C2D

Input Homeomorphism φ, transition (s, a, r, s')
    # Current estimation
    $\mu(\boldsymbol{\theta}) \coloneqq \xi(s, a; \boldsymbol{\theta})$
    # Chosen target
    $(\mathbf{p}(a'), \mathbf{x}(a')) \coloneqq \xi(s', a'; \boldsymbol{\theta}^-)$ for a' in $\mathcal{A}$
    $a^* \coloneqq \arg\max_{a'} \sum_{i=1}^{N} p_i(a') \varphi^{-1}(x_i(a'))$
    $(\mathbf{p}, \mathbf{x}) \coloneqq \xi(s', a^*; \boldsymbol{\theta}^-)$
    $\nu \coloneqq (\mathbf{p}, \varphi(r\mathbf{1} + \gamma\varphi^{-1}(\mathbf{x})))$ (elementwise)
    # Squared Cramér distance
    $L(\boldsymbol{\theta}) \coloneqq \ell_2^2(\nu, \mu(\boldsymbol{\theta}))$
Output $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$

4.1 Implementation Details

The functional model of C2D, shown in Figure 1, builds upon the representation network of DQN in (Mnih et al. 2015). An incoming state s is encoded into a feature vector $\psi(s)$, which is then passed to a parameterized probability function $\mathbf{p}(\psi)$ and a parameterized atom function $\mathbf{x}(\phi, \psi)$ that jointly outputs discrete distributions for all actions. Similar to IQN and FQF in (Dabney et al. 2018a; Yang et al. 2019) we employ an embedding ϕ of all probabilities to get a stabilizing coupling effect between probabilities and atomic classes. The embedding in C2D consists of an single dense layer with ReLU activation. The result vector $\phi(\mathbf{p})$ is concatenated with the feature vector $\psi(s)$ to yield an input for the atom function $\mathbf{x}(\phi, \psi)$. Moreover, since the supports only need to be ordered in distance computations we let the output of $\mathbf{x}$ vary unordered by the parameterization.
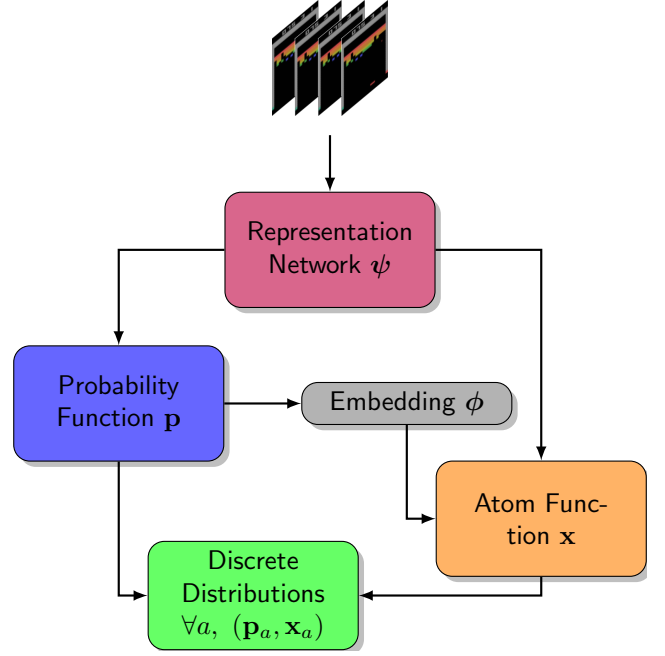

Figure 1: The functional model of C2D generates discrete distributions for all actions given an incoming state. A coupling effect between estimated probabilities $\mathbf{p}$ and atomic classes $\mathbf{x}$ is induced by a parameterized embedding ϕ.

Homeomorphism Our chosen homeomorphism φ for the conjugated pushforward function in Algorithm 2 is a scaled variant of $h(x)$ in (5). Namely, $\varphi(x) \coloneqq \beta h(x)$ where β is close to but slightly less than 2. This is a less aggressive transformation than h while keeping the invertible contraction property intact (Pohlen et al. 2018). Note that this implies $\varphi^{-1}(y) = h^{-1}(y/\beta)$ where $h^{-1}(x)$ is given by

$$\text{sign}(x) \left(\left(\frac{\sqrt{1 + 4\epsilon(|x| + 1 + \epsilon)} - 1}{2\epsilon} \right)^2 - 1 \right).$$

Adaptive Scaling To make the model adaptable to varying orders of return magnitudes in different environments we use the activation

$$\alpha \tanh(x/c)$$

for all supports. Here c is an implementation defined scaling hyperparameter for the output of the internal dense layer of $\mathbf{x}$, and α is a single trainable weight which is slowly tuned by gradients of the Cramér distance to properly accommodate for the scale of discounted returns in its present environment.

5 Evaluation on Stochastic Atari 2600 Games

In this section we gauge the performance of C2D by presenting our experiments on stochastic Atari 2600 games, where simulations are based upon the *Arcade Learning Environment* framework (Bellemare et al. 2013, ALE). We instruct ALE to generate non-deterministic environments by using
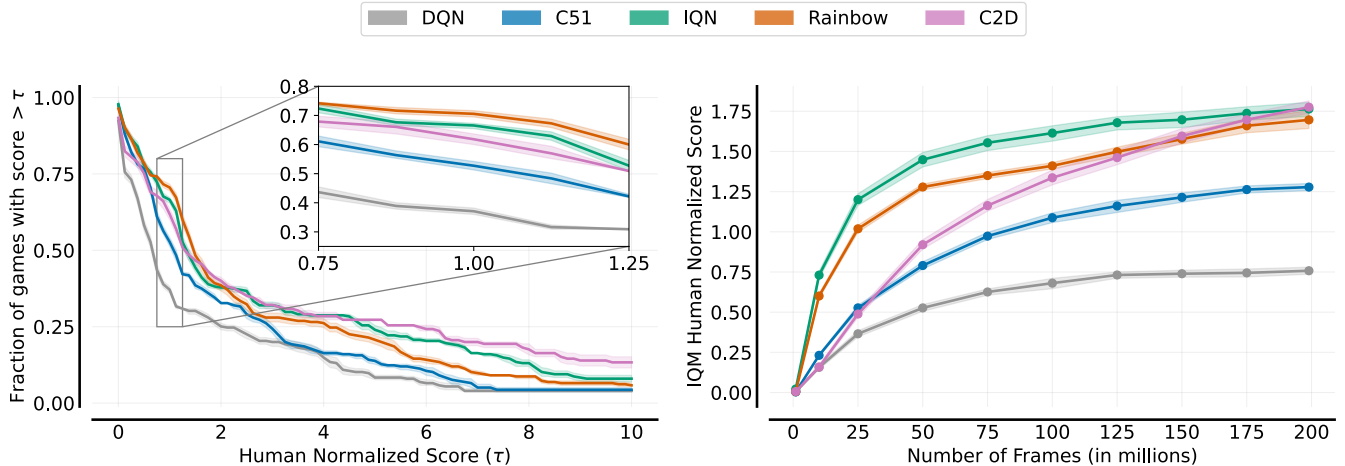
Figure 2: Accumulated statistics on 55 stochastic Atari games computed in accordance to the performance profiling methods given in (Agarwal et al. 2021). The left shows the fraction of all games that after 200M frames achieve a score higher than threshold τ. In particular, at $\tau = 1$ we see the fraction of games with human-like performance or better. The right shows training progression for the 25% trimmed mean. Dopamine results are computed over 5 runs and C2D used 3 runs.

sticky actions (Machado et al. 2018). This implies that environments will have a non-zero probability of previous action repetition and this MDP altering effect is implemented internally in ALE.

The Dopamine Protocol For an "apples-to-apples" comparison we adopt the Dopamine protocol summarized in Table 1 and evaluate training performance over a suite of 55 stochastic Atari 2600 games. The sticky action probability, in accordance to the protocol, is set to the default value of 0.25 in ALE. As is common for single-actor algorithms evaluated on Atari, training procedures mostly follow the original implementation of DQN in (Mnih et al. 2015), which includes 1M-sized replay buffers, episode lengths capped at 108K frames (30 min) and a total training volume of 200M frames. There are however protocol specific settings such as the decay of ε-greedy action probabilities and the length of the random replay history. The protocol also dictates a fixed period for copying parameters to target networks.

Parameter	Value
Min. training ε	0.01
ε-decay schedule ($1.0 \rightarrow$ min. ε)	1M frames
Min. history to start learning	80k frames
Target network update frequency	32k frames
Sticky actions	0.25

Table 1: Hyperparameters common to all implementations of the Dopamine protocol, where ε dictates probabilities for explorative ε-greedy actions.

C2D Specific Settings When applicable we set the hyperparameters of C2D as close as possible to other comparable algorithms. This includes setting $N = 32$ for the number of atomic classes used by the distributional approximations,

which can be compared to the size of the quantile fraction set in IQN. It also includes using *ADAM* as the network optimizer with learning rate $0.5 \cdot 10^{-4}$ and epsilon value $3.125 \cdot 10^{-4}$ (Kingma and Ba 2014).

Moreover, the padding in the three convolutional layers of the representation network ψ follows that of DQN, which implies substantially less network weights than similar implementations found in Dopamine. However, our function implementation of ψ will still differ from DQN by the incorporation of batch normalization layers between convolutions (Ioffe and Szegedy 2015). We found that this makes specialized weight initialization techniques for the deep network to become largely redundant.

We performed a preliminary search for C2D-specific parameters by measuring early training progression over six different environments with varying reward magnitudes. This resulted in using $\beta = 1.99$ for our transformation homeomorphism $\varphi(x) = \beta h(x)$. It also resulted in using $\alpha = 50$ and $c = 5$ for the support activation $\alpha \tanh(x/c)$ in each game. This implies that our *initial distributions* will have a maximum support interval $(-50, 50)$ to represent transformed outcomes in discounted returns. However since α is a trainable parameter, this maximum interval will either shrink or grow depending on both the environment and agent performance.

Scores and Baselines Under a similar set of hyperparameters, we compare and evaluate training performance in terms of human-normalized scores (Mnih et al. 2015). With three runs for each game, comparisons are done against other sticky-action algorithms found in the Dopamine framework (Castro et al. 2018). The discrete measures of C51 (Bellemare, Dabney, and Munos 2017) is of interest since we can get a direct comparison of fixed versus varying support. In addition, we may observe the effect of using a proper distributional metric such as the Cramér distance as a loss versus
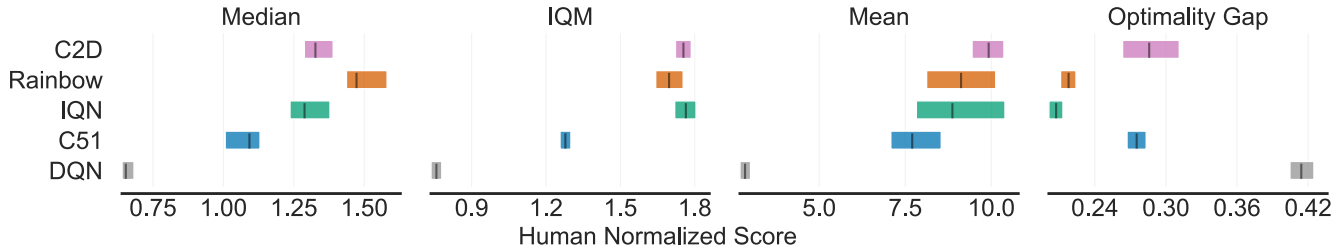
Figure 3: Aggregate metrics on Atari-200M over 55 games. The metrics are computed in accordance to the performance profiling methods given in (Agarwal et al. 2021). Dopamine results are computed over 5 runs and C2D used 3 runs.

the derived cross-entropy of KL. For comparisons with other strong distributional algorithms, we include IQN (Dabney et al. 2018a), which is based on quantile regression, and Rainbow (Hessel et al. 2018). Rainbow, with its myriad of techniques, is built upon C51 and is at the time of writing state-of-art within the Dopamine framework. It is of particular interest to us since performing close to Rainbow suggests that further improvements and superior performance can easily be obtained by adding any of the additional flavors found in Rainbow's non-distributional methods. Finally, as is the norm in most Atari 2600 evaluations, we also include DQN as a baseline.

5.1 Results

In order to have a rigorous evaluation methodology, we analyze results by using the profiling tools presented in (Agarwal et al. 2021). In Figure 2 we see the fraction of all games that achieve a human-normalized score higher than threshold τ. We also see the 25% trimmed mean (IQM) training progression over all frames. In Figure 3 we show 200M aggregate metrics measured in human-normalized scores over all 55 Atari games. These include the optimality gap, which measures the complement $1 - x$ of the mean x given that all scores are truncated by the human score. Thus, an optimality gap of 0 would indicate at least human-like performance over all games. To further showcase possible strengths and weaknesses, we present in Figure 4 four examples of mean training progressions, where C2D with its stated settings displayed significantly different trajectories compared to the other algorithms. More training graphs, support evolution and cross sections in raw scores is provided in the appendix.

The IQM training progression in Figure 2 indicates that C2D may achieve better long-term performance in this metric than the other baselines. This is also indicated in the overall mean at 200M in Figure 3, which is heavily weighed by scores in environments where RL-algorithms to a substantial degree outperform humans. We can see this reflected in Figure 2 where C2D maintains super-human performance in a significant portion of games ($\tau > 5$). However we note that the cross-section $\tau = 1$ indicates that C2D performs worse than IQN and Rainbow when it comes to number of games with human-like performance. In particular, the optimality gap and Figure 4 suggests that the algorithm may in the mean perform worse than C51 in games where it is weak. Moreover, the algorithm has a relatively slow initial progres-
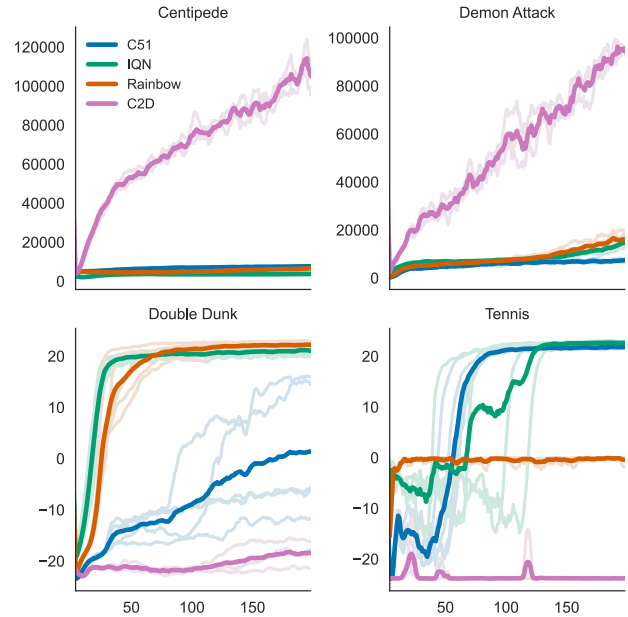

Figure 4: Moving averages (5M frames) of mean raw scores in four environments. With its stated settings, C2D excels in games with frequent rewards of relatively high variance and magnitude. However, C2D is often slow to learn or sensitive to exploration in environments with sparse rewards of low magnitude.

sion, requiring more than 150M frames before it has comparable performance to IQN and Rainbow in the trimmed mean metric. In the median we find that C2D is on par with IQN.

6 Discussion

We have shown that previous value-based methods for learning with a transformed Q-function operator may not hold in a stochastic setting. As a theoretical solution in DRL, we have introduced and shown that the generalized operator T_φ of Definition 3.1 properly reflects learning in a space of distributions that is derived from a chosen transformation homeomorphism. We have also proposed an approximate DRL algorithm based on T_φ in C2D, which has been evaluated on a suite of stochastic Atari games. The learning al-

gorithm produced state-of-the-art performance with respect to the strongest single-actor algorithms in the Dopamine framework. Specifically given the available seeds, we found that C2D outperformed both IQN and Rainbow in the averaged mean over all environments. In the median and IQM we found that C2D has IQN-comparable performance. We could also infer from the optimality gap that C2D is generally weaker than C51 in its low-performing environments.

Since T_φ induces guaranteed theoretical Q-function convergence we argue that the operator serves as a sound basis for approximate DRL involving unaltered rewards. Given a stochastic setting, it seems hard to define a practical and fully transformation correct operator for value-based learning, *i.e.*, a Q-function operator which handles transformations of the underlying random variables and where we have optimal policy invariance. For a deeper look into this problem, see (van Hasselt et al. 2019). However, the ease at which we may handle transformations through DRL is further indicative of the strength of the distributional formalism versus its classical counterpart.

A possible explanation for the success of C2D in the mean is that environments that heavily skew this statistic have frequent non-zero rewards with high variance in discounted returns. That is, the required conjugate distributional statistics for agent networks are then frequently updated by the Cramér distance, with large disparities between action-values in real discounted outcomes. In comparison and directly related to the IQM statistic, C2D often has significantly worse learning progression than IQN and sometimes subpar performance in environments with sparse rewards and low variance in returns, with a tendency to get stuck in non-optimal regions due to the inherit poor exploration of ε-greedy actions. Slowly updating statistics by moving probabilities and supports via the Cramér distance seems to be a harder optimization problem, sensitive to exploration. Although similar to the reasoning made in (Dabney et al. 2018a), we argue that a rainbowesque version of C2D should easily obtain superior performance by adding from Rainbow's non-distributional arsenal, which contains techniques such as multi-step sampling, noisy networks and double distributional DQN.

Interesting future directions for exploring the full scope of approximate DRL derived from T_φ would be through the explorative and distributive methods of algorithms such as R2D2, MuZero or Agent57 (Kapturowski et al. 2018; Schrittwieser et al. 2020; Badia et al. 2020). This could include: A significant increase in training frames to observe long-term convergence behavior combined with an ablation study of the effects of proper transformations. Better representation functions through recurrent or residual networks. Better operational approximations through planning, multi-step sampling or full episode sampling. Better exploration methods for statistics gathering in order to avoid premature convergence on non-optimal greedy regions in difficult environments.

Acknowledgments

References

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A.; and Bellemare, M. G. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Badia, A. P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z. D.; and Blundell, C. 2020. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, 507–517. PMLR.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 449–458. JMLR. org.

Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

Bellemare, M. G.; Le Roux, N.; Castro, P. S.; and Moitra, S. 2019. Distributional reinforcement learning with linear function approximation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2203–2211.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1996. *Neuro-dynamic programming*. Athena Scientific.

Castro, P. S.; Moitra, S.; Gelada, C.; Kumar, S.; and Bellemare, M. G. 2018. Dopamine: A Research Framework for Deep Reinforcement Learning. *arXiv e-prints*, arXiv:1812.06110.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018a. Implicit Quantile Networks for Distributional Reinforcement Learning. In *International Conference on Machine Learning*, 1096–1105.

Dabney, W.; Rowland, M.; Bellemare, M. G.; and Munos, R. 2018b. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hessel, M.; Modayil, J.; van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M. G.; and Silver, D. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. In McIlraith, S. A.; and Weinberger, K. Q., eds., *AAAI*, 3215–3222. AAAI Press.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 448–456. Lille, France: PMLR.

Kapturowski, S.; Ostrovski, G.; Quan, J.; Munos, R.; and Dabney, W. 2018. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, arXiv:1412.6980.

Lyle, C.; Bellemare, M. G.; and Castro, P. S. 2019. A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4504–4511.

Machado, M. C.; Bellemare, M. G.; Talvitie, E.; Veness, J.; Hausknecht, M.; and Bowling, M. 2018. Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents. *J. Artif. Int. Res.*, 61(1): 523–562.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, 278–287.

Pohlen, T.; Piot, B.; Hester, T.; Gheshlaghi Azar, M.; Horgan, D.; Budden, D.; Barth-Maron, G.; van Hasselt, H.; Quan, J.; Večerík, M.; Hessel, M.; Munos, R.; and Pietquin, O. 2018. Observe and Look Further: Achieving Consistent Performance on Atari. *arXiv e-prints*, arXiv:1805.11593.

Rizzo, M. L.; and Székely, G. J. 2016. Energy distance. *wiley interdisciplinary reviews: Computational statistics*, 8(1): 27–38.

Rowland, M.; Bellemare, M.; Dabney, W.; Munos, R.; and Teh, Y. W. 2018. An Analysis of Categorical Distributional Reinforcement Learning. In *International Conference on Artificial Intelligence and Statistics*, 29–37.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Szepesvári, C. 2010. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1): 1–103.

van Hasselt, H.; Quan, J.; Hessel, M.; Xu, Z.; Borsa, D.; and Barreto, A. 2019. General non-linear bellman equations. *arXiv preprint arXiv:1907.03687*.

Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Yang, D.; Zhao, L.; Lin, Z.; Qin, T.; Bian, J.; and Liu, T.-Y. 2019. Fully Parameterized Quantile Function for Distributional Reinforcement Learning. *Advances in Neural Information Processing Systems*, 32: 6193–6202.