The Multiquadric Kernel for Moment-Matching Distributional Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Distributional reinforcement learning has gained significant attention in recent years due to its ability to handle uncertainty and variability in the returns an agent can expect to receive for each action it takes. A key challenge in distributional reinforcement learning is finding a measure of the difference between two distributions that is well-suited for use with the distributional Bellman operator, a function that takes in a value distribution and produces a modified distribution based on the agent's current state and action. In this paper, we address this challenge by introducing the multiquadric kernel to moment-matching distributional reinforcement learning. We show that this kernel is both theoretically sound and empirically effective. Our contribution is mainly of a theoretical nature, presenting the first formally sound kernel for moment-matching distributional reinforcement learning with good practical performance. We also provide insights into why the RBF kernel has been shown to provide good practical results despite its theoretical problems. Finally, we evaluate the performance of our kernel on a number of standard tasks, obtaining results comparable to the state-of-the-art.

1 Introduction

Reinforcement learning is a type of machine learning that involves training agents to take actions in an environment in order to maximize a reward signal. In traditional reinforcement learning, the agent learns a value function that estimates the expected future cumulative discounted reward (value) for each state-action pair. The value function tells the agent how "good" a particular state-action pair is, and the agent uses this information to make decisions about which actions to take in order to maximize its value. However, this approach can be problematic in some cases, such as when the environment is highly stochastic or when the reward signal is noisy.

Distributional reinforcement learning is a variation of reinforcement learning that addresses these challenges by learning a distribution of future cumulative discounted rewards (value distribution), rather than a single expected value, for each state-action pair. This distributional approach allows the agent to consider the uncertainty or variability in the rewards it may receive, and can lead to more robust and accurate decision-making.

Distributional reinforcement learning algorithms differ in two main ways: the representation of the value distribution for a given state-action pair and the measurement of the difference between distributions. Some algorithms utilize a discrete set of samples to represent the distribution (Bellemare et al., 2017; Dabney et al., 2017; Nguyen-Tang et al., 2021), while others employ implicit distributions parameterized by a neural network (Dabney et al., 2018; Yang et al., 2019). Additionally, the agent must have the ability to measure the difference between the predicted value distribution for a given state-action pair and the actual value distribution received, in order to update its predictions. This difference measure is crucial for the agent to learn and adapt.

The Kullbach-Leibler divergence is widespread in the probabilistic machine learning field but possesses some undesirable properties for use in distributional reinforcement learning. Instead, measures such as the Wasserstein metric, the Cramer distance, and the maximum mean discrepancy have shown better results (Bellemare et al., 2017; Dabney et al., 2017; Nguyen-Tang et al., 2021; Dabney et al., 2018; Yang et al., 2019).

The distributional Bellman operator is a key component of many distributional reinforcement learning algorithms. It is used to update the value distribution for a given state-action pair based on the observed rewards and transitions in the environment. The distributional Bellman operator is a contraction under certain conditions, which means that it guarantees that the distance between the updated and original value distribution will always be non-increasing, converging to the true value distribution in the limit (with convergence properties similar to those of the usual Bellman operator used, e.g., in Q-learning).

Our main contribution is expanding the class of known kernels under which the distributional Bellman operator is a contraction. Additionally, our results may also be of theoretical interest for understanding the properties of the distributional Bellman operator.

2 Background

This section introduces the necessary background in classical reinforcement learning, and distributional reinforcement learning, as well as some background on current state-of-the-art distributional reinforcement learning methods.

2.1 Reinforcement Learning

We consider the standard reinforcement learning (RL) setting, where interaction between agent and environment is modeled as a Markov Decision Process $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$ (Puterman, 1994), where $\mathcal{S} \subseteq \mathbb{R}^d$ and $\mathcal{A}$ denote the state and action space respectively. The stochastic reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a function that maps the state and action of an agent to a reward value with some element of randomness or uncertainty. $P(\cdot|\boldsymbol{x}, a)$ denotes the transition probabilities starting in state $\boldsymbol{x} \in \mathcal{S}$ and doing $a \in \mathcal{A}$, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi(\cdot|\boldsymbol{x})$ maps states to distributions over actions.

The Bellman equation is often used as the foundation for algorithms that solve MDPs, such as Q-learning. In general, the equation describes a necessary condition for an optimal solution in dynamic programming optimization. In the context of reinforcement learning, it is often used to describe the optimal state-action value function Q^*. According to the Bellman equation, described by (1), the long term reward, or value, of a state-action pair is equal to the sum of the immediate reward from the current action and the discounted expected reward from future actions:

$$Q^*(\boldsymbol{x}, a) = \mathbb{E}\left[R(\boldsymbol{x}, a)\right] + \gamma \mathbb{E}_P\left[\max_{a' \in \mathcal{A}} Q^*(\boldsymbol{x}', a')\right]. \tag{1}$$

A common approach to solving the Bellman equation involves using the Bellman operator. The Bellman operator, $\mathcal{T}^\pi$, maps state-action values to their next state-action values. The Bellman operator for a policy π is defined as follows:

$$\mathcal{T}^\pi Q(\boldsymbol{x}, a) := \mathbb{E}\left[R(\boldsymbol{x}, a)\right] + \gamma \mathbb{E}_P\left[Q(\boldsymbol{x}', a^\pi)\right]. \tag{2}$$

Alternatively, we can define the Bellman operator for an optimal policy, also known as the Bellman optimality operator as

$$\mathcal{T}^* Q(\boldsymbol{x}, a) := \mathbb{E}\left[R(\boldsymbol{x}, a)\right] + \gamma \mathbb{E}_P\left[\max_{a' \in \mathcal{A}} Q(\boldsymbol{x}', a')\right]. \tag{3}$$

The significance of the Bellman operators is that they are contraction mappings wrt the supremum norm. According to the Banach fixed-point theorem Banach (1922), that means that we have the following properties:

$$\lim_{k \to \infty} (\mathcal{T}^\pi)^k Q = Q^\pi, \tag{4}$$

$$\lim_{k \to \infty} (\mathcal{T}^*)^k Q = Q^*. \tag{5}$$

This will converge even starting with a non-optimal policy π with exploration techniques in a class known as GLIE (greedy in the limit with infinite exploration) (Singh et al., 2000). This means that if we adopt π in a "GLIE manner", then $Q^\pi = Q^*$ in the limit.

Deep Q-Network (DQN) (Mnih et al., 2013) approximates Q using a neural network, which we will denote Q_θ. Mnih et al. (2013) proceed to approximate Q^* using gradient descent on the squared temporal difference (TD), defined as

$$\delta_t^2 = \left[\mathcal{T}^\pi Q_\theta(\boldsymbol{x}_t, a_t) - Q_\theta(\boldsymbol{x}_t, a_t)\right]^2 . \tag{6}$$

Although this is still not guaranteed to converge due to the employment of the *deadly triad* (Sutton & Barto, 2018), namely *bootstrapping*, *off-policy learning*, and *function approximation*, practical performance is still good.

2.2 Distributional Reinforcement Learning

Distributional reinforcement learning focuses on learning the distribution of values in a given environment, rather than just the expected value. In distributional reinforcement learning, the algorithm typically uses a value function to approximate the distribution for each state-action pair. This value function is updated over time based on the rewards received and the predicted distribution of rewards for each state-action pair. By doing this, the algorithm can learn to anticipate the range of possible cumulative discounted rewards and make decisions that maximize the expected value.

For an agent following policy π, the value Z^π, or *value distribution* (Bellemare et al., 2017), is the sum of discounted rewards along the agent's trajectory:

$$Z^\pi(\boldsymbol{x}, a) := \sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{x}_t, a_t), \tag{7}$$

starting in state $\boldsymbol{x}_0 = \boldsymbol{x}$, with $a_0 = a$, $a_t \sim \pi(\cdot|\boldsymbol{x}_t)$, and $\boldsymbol{x}_t \sim P(\cdot|\boldsymbol{x}_{t-1}, a_{t-1})$. The objective of RL can be summarized as finding the optimal policy, that is the sequence of actions that an agent should take, in order to maximize the reward signal $Q^\pi(\boldsymbol{x}, a) := \mathbb{E}_{P,R,\pi}\left[Z^\pi(\boldsymbol{x}, a)\right]$ in a given environment. In other words, it is the strategy that the agent should follow to achieve the best possible outcome that is sought.

The distributional Bellman operator is a mathematical operator used in distributional reinforcement learning to estimate the probability distribution of future rewards for each state-action pair in an environment. This operator is based on the Bellman equation. The distributional Bellman operator extends the Bellman equation to estimate the probability distribution of future rewards, rather than just the expected value. This allows the operator to capture the uncertainty and variability of the rewards in the environment, and to make predictions about the range of possible outcomes. Given a random variable Z^π denoting the value distribution, the operator is defined as

$$\mathcal{T}^\pi Z^\pi(\boldsymbol{x}, a) :\overset{D}{=} R(x, a) + \gamma Z^\pi(X', A'), \tag{8}$$

where $A \overset{D}{=} B$ denotes equality in distribution, i.e. random variable A and B have the same distribution.

Bellemare et al. (2017) show that the distributional Bellman operator for a fixed policy ($\mathcal{T}^\pi$) is a contraction in the p-Wasserstein metric and Cramér distance, but not in KL-divergence, total varation or Kolomogrov distance. Nguyen-Tang et al. (2021) further show that $\mathcal{T}^\pi$ is a contraction in maximum mean discrepancy for carefully selected kernels.

2.2.1 Learning Categorical Distributions

One popular approach to distributional reinforcement learning is to approximate the distribution of values, Z, by a discretized or categorical variable. C51, which stands for "Categorical DQN" was introduced by Bellemare et al. (2017) as an extension of DQN. Like DQN, C51 uses a neural network to approximate the action-value function, but instead of outputting a single value for each action, it outputs a distribution over the possible returns. This distribution is represented using a set of discrete "atoms", which allows the agent to represent a wide range of possible returns. 51 discrete atoms per action were used in their Atari 2600 experiments, hence the name C51.

In C51, the distance between two distributions is calculated using the Kullback-Leibler (KL) divergence. One issue with using the KL divergence as a measure of the difference between two distributions in distributional reinforcement learning is that the distributional Bellman operator is not a contraction in KL divergence (Bellemare et al., 2017). This means that the KL divergence between the output of the distributional Bellman operator and the input distribution can be larger than the KL divergence between the output of the distributional Bellman operator and the true value distribution. This can lead to problems in distributional reinforcement learning, as it means that the agent's estimate of the value distribution may not necessarily get closer to the true distribution as it takes more actions and gathers more experiences.

For these reasons, KL divergence is generally not considered a good measure of the difference between two distributions in distributional reinforcement learning. Instead, other methods typically use measures such as the Wasserstein distance or the Cramer distance, which do satisfy the contraction property.

2.2.2 Quantile Regression

Quantile Regression DQN (QR-DQN) is a variant of DQN introduced by Dabney et al. (2017). To learn the value distribution, QR-DQN uses a variant of Q-learning called quantile regression Q-learning, which is based on quantile regression, a statistical technique for estimating the conditional quantile function of a random variable.

One advantage of QR-DQN is that it uses the Wasserstein distance, also known as the Earth Mover's distance, as a measure of the difference between two distributions. The Wasserstein distance is a well-known metric in the field of probability theory and statistics, and it has several desirable properties that make it a good choice for distributional reinforcement learning.

First, the Wasserstein distance is a true metric, meaning that it satisfies the triangle inequality and is symmetric. This makes it a more consistent and reliable measure of the difference between two distributions than the Kullback-Leibler (KL) divergence, which does not satisfy these properties.

Second, the Wasserstein distance is a smooth, continuous function, which makes it well-suited for use with gradient-based optimization methods. This is important in reinforcement learning, where algorithms often rely on gradient-based optimization to learn the value function.

Finally, the distributional Bellman operator is a contraction in Wasserstein distance. The contraction property of the distributional Bellman operator in Wasserstein distance is an important property in distributional reinforcement learning. It ensures, under certain conditions, that the agent's estimate of the distribution of returns will get closer to the true distribution as the agent interacts with the environment.

2.2.3 Implicit Quantile Regression

Implicit quantile regression DQN (IQR-DQN) (Dabney et al., 2018) is a variant of the Quantile Regression DQN (QR-DQN) (Dabney et al., 2017) algorithm for distributional reinforcement learning. Like QR-DQN, IQR-DQN is based on the DQN algorithm, but instead of learning a single value for each state-action pair, it learns a distribution of values. However, unlike QR-DQN, which uses an explicit quantile regression model to estimate the distribution of returns, IQR-DQN uses an implicit quantile model, which is a neural network that directly maps states and actions to quantiles of the return distribution.

Work on IQR-DQN (Dabney et al., 2018; Yang et al., 2019) has focused on developing efficient and effective algorithms for learning the implicit quantile model, as well as on demonstrating the performance of IQR-DQN on a variety of reinforcement learning tasks. Results have shown that IQR-DQN can learn more accurate value distributions than other distributional reinforcement learning methods and that it can achieve better performance.

2.2.4 Moment-matching

Nguyen-Tang et al. (2021) proposed using maximum mean discrepancy (MMD) as a target, rather than the Wasserstein distance. The squared maximum mean discrepancy between two distributions p and q relative to a kernel k is defined as

$$\mathrm{MMD}^2(p, q; k) = \mathbb{E}\left[k(X, X')\right] + \mathbb{E}\left[k(Y, Y')\right] - 2\mathbb{E}\left[k(X, Y)\right], \tag{9}$$

where X, X' are independent random variables with distribution p and Y, Y' are independent random variables with distribution q. Given two sets of samples $\{\boldsymbol{x}_i\}_{i=1}^m \sim p$ and $\{\boldsymbol{y}_i\}_{i=1}^m \sim q$, the following is an unbiased estimator of MMD

$$\widehat{\mathrm{MMD}}_u^2(\{\boldsymbol{x}_i\}, \{\boldsymbol{y}_i\}; k) := \frac{1}{m(m-1)} \sum_{i \neq j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + k(\boldsymbol{y}_i, \boldsymbol{y}_j) - 2k(\boldsymbol{x}_i, \boldsymbol{y}_j).$$

Nguyen-Tang et al. (2021) use a biased estimator $\widehat{\mathrm{MMD}}_b$ of MMD, but justify the choice by noting that the estimator in practice shows lower variance than its unbiased counterpart. The biased estimator, MMD_b, is defined as

$$\widehat{\mathrm{MMD}}_b^2(\{\boldsymbol{x}_i\}, \{\boldsymbol{y}_i\}; k) := \frac{1}{m^2} \sum_{i,j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + k(\boldsymbol{y}_i, \boldsymbol{y}_j) - k(\boldsymbol{x}_i, \boldsymbol{y}_j).$$

Kernels MMD^2 with the kernel $k(\boldsymbol{x}, \boldsymbol{y}) = -\|\boldsymbol{x} - \boldsymbol{y}\|^2$ is called the *energy distance*. We will call this kernel the negative Euclidean kernel. The energy distance is a natural choice, since the distributional Bellman operator is a contraction in the energy distance (Nguyen-Tang et al., 2021). Unfortunately, it is zero as long as the two distributions are equal in expectation (Székely & Rizzo, 2013). This means that we will not see the convergence in distribution, but rather in expectation, essentially reducing it to classical Q-learning. Nguyen-Tang et al. (2021) nonetheless look at the practical performance of the energy distance, but finds it performs unfavorably compared to the RBF kernel.

Székely & Rizzo (2013) showed that with kernels of the form $k(\boldsymbol{x}, \boldsymbol{y}) = -\|\boldsymbol{x}-\boldsymbol{y}\|^p$ for $p \in (0, 2)$, then $\mathrm{MMD}(\mu, \nu; k)$ is zero if and only if μ and ν are equal in distribution. Nguyen-Tang et al. (2021) further show that the distributional Bellman operator is a contraction in these kernels. Practical performance was evaluated for $p = 1$, but the results were not promising. We will call this the negative Manhattan kernel.

The RBF kernel is a common choice in Gaussian processes and other applications. It is defined as

$$k_h(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-h^2 \|\boldsymbol{x} - \boldsymbol{y}\|_2^2\right), \tag{10}$$

where h, is a free parameter. Nguyen-Tang et al. (2021) showed by counterexample that the distributional Bellman operator is not a contraction under MMD with the RBF kernel, but also found that the RBF kernel performs favorably compared to the negative Euclidean and negative Manhattan kernels, when using a mixture of RBF kernels with different length-scale values, defined as

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{l=1}^{10} \exp\left(-l^{-1} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2\right). \tag{11}$$

Pseudo-samples MMDQN (Nguyen-Tang et al., 2021) use the QR-DQN network structure to approximate Z. They define the N outputs per action to be (pseudo-)samples from Z_θ, and then minimize MMD_b^2 between the (pseudo-)samples from Z_θ and $\mathcal{T} Z_{\overline{\theta}}$. This means that true samples from the value distribution are never drawn, and that the approximation is completely deterministic. There is no underlying restrictions with the algorithm that would suggest that a sampling based approach would not also work.

Open question Nguyen-Tang et al. (2021) pose a question on what the sufficient conditions on the kernels are for contraction in MMD. They further question whether scale sensitivity is a necessary condition. We show in Section 4 that scale sensitivity is not a necessary condition for contraction in MMD.

3 Limitations of the Radial Basis Function for MMD

In this section, we aim to investigate the sensitivity of MMDQN (Nguyen-Tang et al., 2021) performance on Atari problems to RBF kernel parameters. While MMDQN has achieved remarkable performance across all 57 games, questions remain regarding how sensitive this performance is to change of kernel. The use of a single mixture kernel for all 57 games, may give the impression that the method is relatively insensitive to the kernel, however, we will show that this is not the case. We will also identify settings where the optimal RBF parameters are more challenging to find. It is worth noting that the authors do not claim that these parameters will generalize to different environments; quite the contrary, they state that the kernel's role in performance is crucial. They also state that identifying a kernel that performs well on all games is challenging. In this context, we explore the impact of the RBF bandwidth parameter on MMDQN performance in simple toy settings.

We noticed that for all the Atari games that MMDQN (Nguyen-Tang et al., 2021) released videos of the Z-distribution was uni-modal. Therefore, we designed a procedure to evaluate MMDs ability to fit multi-modal distributions with the RBF kernel. The procedure is described in Algorithm 1.

Algorithm 1 Procedure for evaluating the ability of MMD with kernel k to approximate a distribution.

```
Require: $k, P, \{x\}_{1:N}, T$
  p-values ← $Array[1:T]$
  for $t = 1:T$ do
      $y_i \sim P$    for $i = 1, \dots, N$
      $\{\Delta x\}_{1:N} \leftarrow \nabla_x \widehat{\text{MMD}}_b(\{x\}_{1:N}, \{y\}_{1:N}; k)$
      $\{x\}_{1:N} \leftarrow Optimiser(\{x\}_{1:N}, \{\Delta x\}_{1:N})$
      p-values[$t$] ← Anderson-Darling($\{x\}_{1:N}, P$)
  end for
  return p-values
```

For the experiments Anderson-Darling test experiment, we chose a Gaussian mixture model with two components, $\mathcal{N}(-2, 0.2)$ and $\mathcal{N}(2, 0.2)$ with weights 0.2 and 0.8 respectively. We tested the MQ with $h = 1$, $h = 10$, and $h = 100$. For the RBF we used the same kernel that Nguyen-Tang et al. (2021) used in their Atari experiments, viz. Equation 11. Search for a more effective kernel yielded no positive results. Even if such a kernel exists, these results show that it is not trivial to find such a kernel, and suggests that the performance on certain Atari games could have suffer as a result.

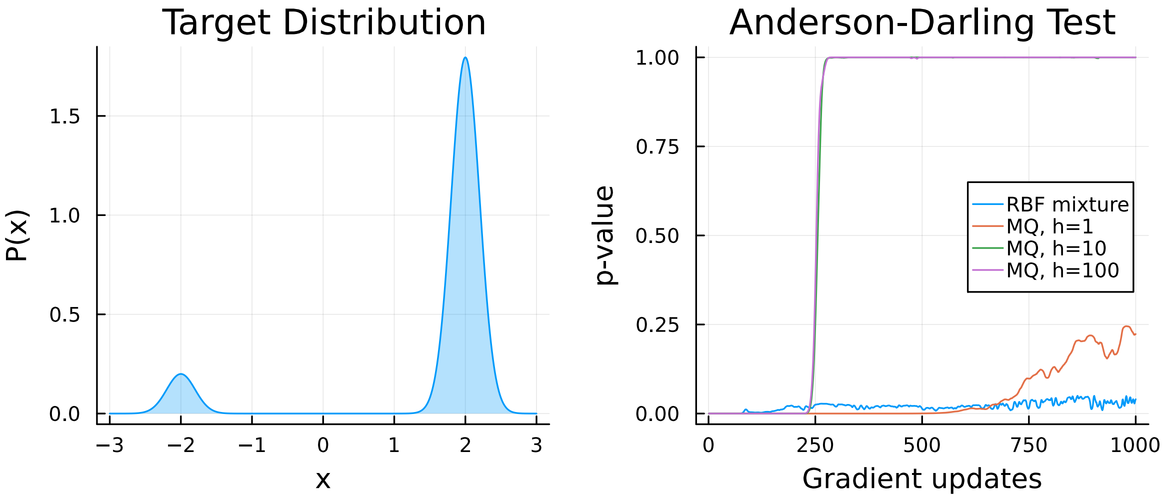

Figure 1: Figure showing target distribution and p-value for the Anderson-Darling test with the null-hypothesis that the samples are drawn from P against the alternative hypothesis that the samples are not drawn from P. We want to see a score close to 1. A learning rate of 0.1 was used for MQ and a learning rate of 0.01 was used for RBF. The MQ kernel was scaled by h^{-1} to avoid the magnitude of the gradients growing significantly with h.

4 Theoretical Work

In this paper, we propose to use the multiquadric kernel (MQ) (Hardy, 1971) for moment-matching distributional reinforcement learning. Figure 2 provides an illustrative comparison between the multiquadric kernel and the kernels discussed in section 2.2.4. Our choice of the multiquadric kernel is motivated by several desirable properties it possesses. Given a free parameter h, the multiquadric kernel is defined as

$$k_h(\boldsymbol{x}, \boldsymbol{y}) = -\sqrt{1 + h^2 \|\boldsymbol{x} - \boldsymbol{y}\|_2^2}. \tag{12}$$

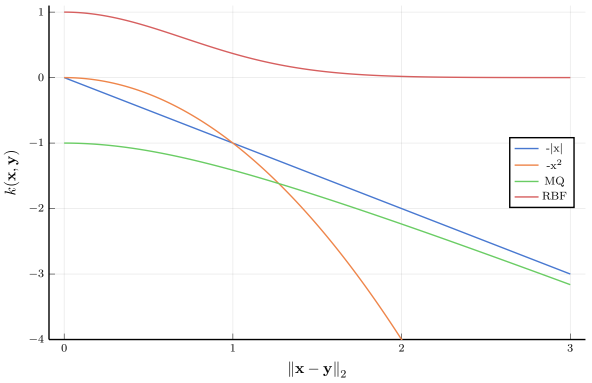

Figure 2: The multiquadric (MQ) kernel compared to the kernels discussed in section 2.2.4 as a function of the Euclidean distance between two points. $h = 1$ was chosen for both MQ and RBF.

First, under the multiquadric kernel, maximum mean discrepancy (MMD) is a metric in the family of valid distributions. In other words, it satisfies the triangle inequality, is symmetric, and is zero if and only if the distributions are equal. This is important in the context of distributional reinforcement learning, as it ensures that the distance measure used to compare the predicted and true distributions of returns is consistent and reliable.

Second, we demonstrate that the distributional Bellman operator is a contraction in MMD with the multiquadric kernel under certain conditions. This property is important in reinforcement learning, as it ensures that the agent's estimate of the distribution of returns will necessarily get closer to the true distribution as it takes more actions and gathers more experiences. This can help the agent to more accurately learn the distribution of returns and make good decisions based on that distribution.

Furthermore, the multiquadric kernel has smooth properties, resulting in a smooth maximum mean discrepancy when using the multiquadric kernel. If the loss function is not smooth, then the gradients of the loss function can be very noisy, which can make it difficult for the model to learn effectively. On the other hand, if the loss function is smooth, then the gradients will be relatively stable, which allows the model to make more consistent and reliable updates to its parameters during training. This can ultimately lead to faster convergence and better performance of the model. This is one of the motivations behind the use of the ℓ_2 metric and the Huber loss over ℓ_1 metric in deep learning.

Finally, the magnitude of the gradient of the kernel is upper bound by a constant. If the magnitude of the gradient is not upper-bounded, then the learning algorithm may oscillate or diverge, rather than converge to a stable and accurate solution. By upper bounding the magnitude of the gradient, it is possible to ensure that the learning algorithm converges to a stable and accurate solution in a reasonable amount of time. This is one of the reasons the Huber loss has become so popular in reinforcement learning, and why we often see gradient clipping being performed when the ℓ_2 metric is used as a loss function.

Concisely formulated, given a multiquadric kernel k_h, we have

1. $\mathrm{MMD}(\cdot, \cdot; k_h)$ is smooth,
2. $\sup_{\boldsymbol{x} \in \mathbb{R}^n} \| \frac{\partial}{\partial \boldsymbol{x}} \mathrm{MMD}(\boldsymbol{x}, \boldsymbol{y}; k_h) \| \leq C_h$.
3. $\mathrm{MMD}(\cdot, \cdot; k_h)$ is a metric on probability distributions,
4. The Bellman operator is a contraction in MMD with k_h.

Table 1 shows property 1 - 4 for a range of radial kernels, i.e. kernels on the form $k(\boldsymbol{x}, \boldsymbol{y}) = \psi(\|\boldsymbol{x} - \boldsymbol{y}\|)$. We observe that only the multiquadric kernel possesses all the aforementioned properties. As far as we know, the MQ kernel is currently the only kernel that satisfies all properties 1 - 4 and is the first non-scale-sensitive kernel with the contraction property. However, it is possible that other kernels meeting these criteria exist, but we have not seen any evidence of their discovery yet. Properties 1 and 2 can be verified by inspection. Proofs for properties 3 and 4 will follow.

Radial Kernel	$\psi(t)$	1	2	3	4
Multiquadric	$-\sqrt{1+h^2t^2}$	✓	✓	✓	✓
RBF	$\exp(-h^2t^2)$	✓	✓	✓	✗
Unrectified	$-\lvert t\rvert$	✗	✓	✓	✓
Unrectified	$-t^2$	✓	✗	✗	✓

Table 1: Property 1 - 4 for various radial kernels.

Our proof that the distributional Bellman operator is a contraction in MMD with the multiquadric kernel requires a few definitions and existing lemmas. We begin by defining *completely monotonic functions*, *conditionally (strictly) positive definite kernels*, and *pushforward measures*.

Definition 4.1. $f(x)$ is a completely monotonic function (Schilling et al., 2012, Definition 1.3) if it is infinitely differentiable and

$$(-1)^n\psi^{(n)}(x) \geq 0 \text{ for } x > 0 \text{ and } n = 0, 1, 2, \dots$$

Definition 4.2. A real-valued kernel k is called conditionally (strictly) positive definite if it is symmetric, i.e. $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{y}, \boldsymbol{x})$, and satisfies

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_i k(\boldsymbol{x}_i, \boldsymbol{x}_j) c_j^* \geq 0 \qquad (> 0)$$

for any $\boldsymbol{c} \in \mathbb{R}^m$ with

$$\sum_{i=1}^{m} c_i = 0.$$

Definition 4.3. $(f)_{\#}\mu$ is the distribution of a random variable $f(X)$, given $X \sim \mu$. This is known as the pushforward measure of μ by f.

The following three lemmas provide a basis for our main results in Theorem 4.1 and Theorem 4.2.

Lemma 4.1 ((Székely & Rizzo, 2013, Proposition 3))**.** $\mathrm{MMD}(\cdot, \cdot; k)$ is a metric on $\mathcal{P}(\mathcal{X})$ if and only if k is a conditionally strictly positive definite kernel.

Lemma 4.2 ((Micchelli, 1986, Theorem 2.1))**.** Let $g(t) := f(\sqrt{t})$. Then f is conditionally positive definite if and only if $-g'(t)$ is completely monotonic.

Lemma 4.3 ((Wendland, 2005, Corollary 8.20))**.** f is conditionally *strictly* positive definite if and only if it is conditionally positive definite and not a polynomial of order 2 or less.

Theorem 4.1 formally states property 3 of the multiquadric kernel.

Theorem 4.1. $\mathrm{MMD}(\cdot, \cdot; k_h)$ is a metric on $\mathcal{P}(\mathcal{X})$ for the multiquadric kernel $k_h(\boldsymbol{x}, \boldsymbol{y}) = -\sqrt{1 + h^2\|\boldsymbol{x} - \boldsymbol{y}\|^2}$

Proof. By utilizing Lemma 4.1, we only need to show that the multiquadric kernel is conditionally strictly positive definite.

We rewrite $k_h(\boldsymbol{x}, \boldsymbol{y})$ as $\psi(\|\boldsymbol{x} - \boldsymbol{y}\|^2)$. We can directly show that for $g(t) = -\sqrt{1+t}$ with $t > 0$, then $-g'(t) = \frac{1}{2\sqrt{1+t}}$ is completely monotonic. By induction, we see that its n-th derivative

$$\frac{d^n}{dt^n}\frac{1}{2\sqrt{1+t}} = \frac{\sqrt{\pi}(x+1)^{-n-1/2}}{2\Gamma\left(\frac{1}{2} - n\right)}$$

has an oscillating sign on $(0, \infty)$, due to the sign of the gamma function for strictly negative values. By Lemma 4.2, we conclude that $f(t)$ is positive definite. By Lemma 4.3, $f(t)$ is strictly positive definite. ☐

Lemma 4.4 is the basis for our proof that the distributional Bellman operator is a contraction in MMD with the multiquadric kernel (Theorem 4.2).

Lemma 4.4. $f(t; a, b) := \sqrt{a + t^2} - \sqrt{b + t^2}$ is positive definite for $a > b$.

In order to establish the positive definiteness of $f(t)$, we begin by taking the n-th derivative of $f(\sqrt{t})$ and demonstrating that it is completely monotonic. Utilizing Lemma 4.4, we can then conclude that $f(t)$ is positive definite. A comprehensive proof can be found in the appendix.

Theorem 4.2. The distributional Bellman operator is a contraction in MMD with the multiquadric kernel.

The detailed proof of Theorem 4.2 is given in the appendix. The main step is to show that it is sufficient that the kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\frac{1}{\gamma^2} + h^2\|\boldsymbol{x} - \boldsymbol{y}\|_2^2} - \sqrt{1 + h^2\|\boldsymbol{x} - \boldsymbol{y}\|_2^2}$$

is conditionally positive definite. The condition is true as a result of Lemma 4.4.

Using a similar proof to that of Theorem 4.2, we can show that for two distributions ν, μ with bounded support, the distributional Bellman operator is a contraction in MMD with the RBF kernel under certain conditions. Corollary 4.1 defines the bounds. The proof is reserved for the appendix.

Corollary 4.1. Let $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-h^2\|\boldsymbol{x} - \boldsymbol{y}\|^2)$. Then the distributional Bellman operator is a contraction in $\mathrm{MMD}^2(\mu, \nu; k)$ for distributions μ, ν with support S_μ, S_ν such that for some $\alpha > 0$

$$\sup_{\boldsymbol{x} \in S_\mu, \boldsymbol{y} \in S_\nu} \|\boldsymbol{x} - \boldsymbol{y}\| \leq \frac{\log(\gamma^{2-\alpha})}{(\gamma^2 - 1)h^2}.$$

The implication of the corollary is that to ensure convergence, a flat radial basis function (RBF) kernel with a small h-parameter may be necessary when the value distributions have a bounded range. Conversely, if the value distributions lack a bounded range, convergence cannot be guaranteed.

5 Experiments

To evaluate the performance of the multiquadric kernel, we conducted experiments on 8 Atari games from the Arcade Learning Environment (ALE). Our goal was to see whether our algorithm was able to achieve results similar to those obtained by MMDQN with the RBF kernel, a state-of-the-art distributional reinforcement learning algorithm, while also demonstrating the benefits of using a kernel with theoretical guarantees.

To show that we are not handpicking games that the multiquadric kernel performs better on, we use the 6 Atari games Nguyen-Tang et al. (2021) used for tuning (Breakout, Assault, Asterix, MsPacman, Qbert, and BeamRider). We have selected two additional games, one where MMDQN performed significantly better than QR-DQN (Tutankham), and one where it performed significantly worse (SpaceInvaders). We are interested in whether the multiquadric kernel, with its contraction property, will outperform the RBF kernel in environments where the RBF kernel struggles. By inspecting the original results presented in the MMDQN paper Nguyen-Tang et al. (2021), we see that most of the environments in which MMDQN significantly outperforms QR-DQN are part of the 6 games used for tuning. This suggests that appropriate kernel parameters are important for performance. For the multiquadric kernel, we only used Breakout to tune the kernel parameters.

To ensure that we do not involuntarily skew the experiments in favor of the multiquadric kernel, we used the original implementation of MMDQN by Nguyen-Tang et al. (2021), and all except the kernel parameters remained the same. Equation 11 defines the parameters used for the RBF mixture kernel. We investigate two parameter values for the multiquadric kernel, $h = 1$ and $h = 10$. The results in Figure 3 show similar performance between the multiquadric and RBF kernel, despite the hyperparameter optimization that has been done for the RBF kernel parameters over these games. This could explain the slightly worse performance in Asterix and Assault. Interestingly, the RBF kernel performed significantly worse on SpaceInvaders. Although no definitive conclusions can be made from this, it does support our argument that the performance is more sensitive to RBF kernel parameters. Raw result data is available at **http://example.com**[1].

---

[1] The URL to a repository containing all our results will be made available once the article is accepted for publication.

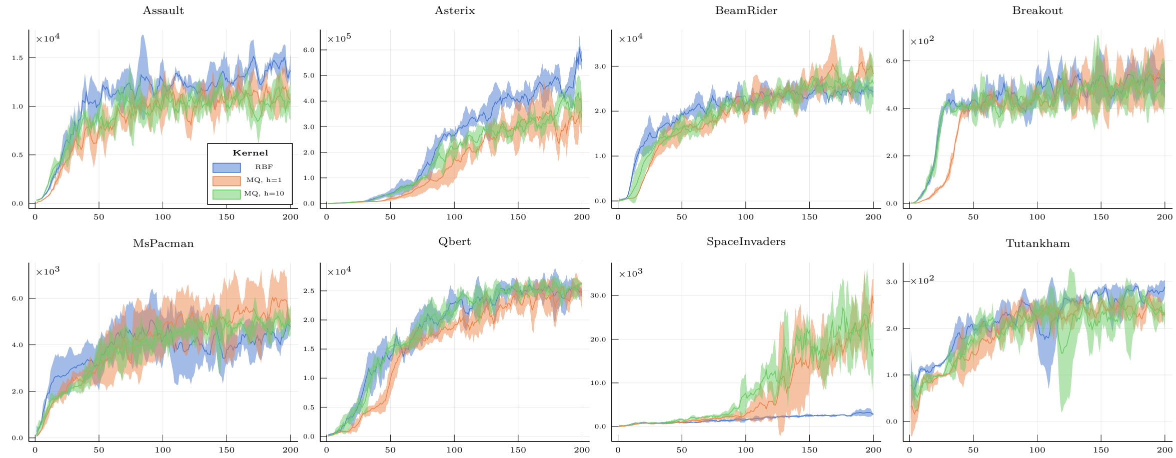

Figure 3: Training curves for the multiquadric (MQ) and RBF kernels on 8 Atari 2600 games. Curves are averaged over 3 seeds and smoothed over a sliding window of 5 iterations. 95% confidence intervals are shown. Reference values for the RBF kernel are from Nguyen-Tang et al. (2021).

6 Discussion and Conclusions

In this paper, we have explored the use of the multiquadric kernel for distributional reinforcement learning. We have shown that the multiquadric kernel has several key theoretical properties that have been studied in this context:

1. The distributional Bellman operator is a contraction in MMD with the multiquadric kernel.
2. MMD is a metric under the multiquadric kernel, which means the MMD between two random variables can only be zero when they are equal in distribution.

There are several reasons why a theoretically sound kernel might be preferred over a kernel that performs similarly but lacks theoretical guarantees:

- The contraction property provides a rigorous foundation for the algorithm and can give us confidence in the performance and behavior of the algorithm under different conditions. This is especially important in reinforcement learning, where the agent is learning to interact with and make decisions in a complex and potentially unknown environment.
- The properties discussed in section 4 can help to ensure the algorithm is well-behaved and robust and can help to avoid pathological or degenerate cases that might arise with kernels that lack the metric or contraction property.
- The discussed properties, particularly the metric and contraction property can facilitate the development of new techniques and variations on the algorithm, by providing a clear set of assumptions and guarantees that can be built upon or modified.
- MMD with MQ kernel's improved ability to model bimodal distribution can be critical for risk-sensitive policies where low returns are particularly detrimental, such as in investment management.

Overall, while it is certainly important to consider the practical performance of a kernel, a theoretically sound kernel can offer a number of additional benefits that can make it a compelling choice for moment-matching distributional reinforcement learning. Additionally, our experimental results show that the multiquadric kernel is stable with regards to parameter-changes, and that a mixture kernel is not necessary to achieve good results.

Furthermore, through Corollary 4.1, we have provided insight into the practical performance of the RBF kernel. The relationship between convergence, the parameter h, and the support of value distributions can provide insight into tuning h for each task. Alternatively, it could provide the basis for an algorithm that learns and changes h during training. Such an algorithm was proposed as future work by Nguyen-Tang et al. (2021). It is possible that the RBF kernel may encounter difficulties in such a scenario due to the absence of a convergence guarantee and that the multiquadric kernel may be a more suitable alternative.

Finally, our results demonstrate the potential of the multiquadric kernel for distributional reinforcement learning. Its contraction and metric property make it a promising choice for a wide range of reinforcement learning tasks. Further research is needed to fully explore the capabilities of the multiquadric kernel and to compare its performance to other kernels in more diverse and challenging environments.

References

Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3(1):133–181, 1922. URL `http://eudml.org/doc/213289`.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/bellemare17a.html`.

Will Dabney, Mark Rowland, Marc Bellemare, and Remi Munos. Distributional reinforcement learning with quantile regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, 10 2017. doi: 10.1609/aaai.v32i1.11791.

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1096–1105. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/dabney18a.html`.

Rolland L Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8):1905–1915, 1971.

Charles A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986. doi: 10.1007/BF01893414. URL `https://doi.org/10.1007/BF01893414`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL `https://arxiv.org/abs/1312.5602`.

Thanh Nguyen-Tang, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning via moment matching. In *AAAI*, 2021.

Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

René L Schilling, Renming Song, and Zoran Vondracek. *Bernstein Functions: Theory and Applications*, volume 37 of *De Gruyter Studies in Mathematics*. Walter de Gruyter GmbH Co.KG, Berlin/Boston, 2. aufl. edition, 2012. ISBN 3110252295.

Satinder Singh, Tommi Jaakkola, Michael Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38:287–308, 03 2000. doi: 10.1023/A:1007678930559.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. ISSN 0378-3758. doi: https://doi.org/10.1016/j.jspi.2013.03.018. URL `https://www.sciencedirect.com/science/article/pii/S0378375813000633`.

Holger Wendland. Scattered data approximation, 2005.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, December 2019. URL `https://www.microsoft.com/en-us/research/publication/fully-parameterized-quantile-function-for-distributional-reinforcement-learning/`.

A Proofs

Lemma A.1. $f(t;a,b) := \sqrt{a+t^2} - \sqrt{b+t^2}$ is positive definite for $a > b$.

Proof. Let $g(t) := f(\sqrt{t}) = \sqrt{a+t} - \sqrt{b+t}$. We use that a function $f(t)$ is positive definite if and only if $g(t)$ is completely monotonic. By induction, it follows that

$$\frac{d^n \sqrt{c+t}}{dt^n} = (c+t)^{1/2-n} \left(\frac{3}{2} - n\right)_n ,$$

where $(\cdot)_n$ is the Pochhammar symbol. Hence,

$$\begin{aligned}
&\frac{d^n \sqrt{a+t} - \sqrt{b+t}}{dt^n} \\
&= (a+t)^{1/2-n} \left(\frac{3}{2} - n\right)_n - (b+t)^{1/2-n} \left(\frac{3}{2} - n\right)_n \\
&= \left((a+t)^{1/2-n} - (b+t)^{1/2-n}\right) \left(\frac{3}{2} - n\right)_n .
\end{aligned}$$

The first term is positive for $n = 0$ and negative otherwise. The second term is positive for $n = 0$ and has sign $(-1)^{n+1}$ otherwise. It is therefore evident that $g(t)$ is completely monotonic and thus $f(t)$ is positive definite. ☐

Theorem A.1. The distributional Bellman operator is a contraction in MMD with the multiquadric kernel.

Proof. Let k be the multiquadric kernel, i.e.

$$k(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x} - \boldsymbol{y}) = -\sqrt{1 + h^2 \|\boldsymbol{x} - \boldsymbol{y}\|_2^2}.$$

Nguyen-Tang et al. (2021) prove that for the distributional Bellman operator to be a contraction it is sufficient to show that for all $\gamma \in (0,1)$

$$\mathrm{MMD}^2\left((f_{r,\gamma})_{\#}\mu, (f_{r,\gamma})_{\#}\nu; k\right) \le \gamma \,\mathrm{MMD}^2(\mu, \nu; k),$$

where $(f_{r,\gamma})_{\#}\mu$ denotes the pushforward measure of μ by $(f_{r,\gamma})_{\#}$. Nguyen-Tang et al. (2021) show in Lemma 4 that this holds with equality for all scale sensitive and shift-invariant kernels k, i.e. whenever $k(r+\gamma x, r+\gamma y) = \gamma^{\alpha} k(x,y)$ for some $\alpha > 0$. We will show that this condition holds for the multiquadric kernel.

First, we will define

$$\tilde{\psi}_\gamma(\boldsymbol{z}) := -\sqrt{\frac{1}{\gamma^2} + h^2 \|\boldsymbol{z}\|_2^2},$$

with

$$\tilde{k}_\gamma(\boldsymbol{x}, \boldsymbol{y}) := \tilde{\psi}_\gamma(\boldsymbol{x} - \boldsymbol{y}).$$

Note that $\psi(\gamma \boldsymbol{z}) = \gamma\tilde{\psi}_\gamma(\boldsymbol{z})$.

$$\begin{aligned}
\mathrm{MMD}^2\Big((f_{r,\gamma})_\#\mu, (f_{r,\gamma})_\#\nu; k\Big) &= \int\int k(\boldsymbol{z}, \boldsymbol{z}')(f_{r,\gamma})_\#\mu(d\boldsymbol{z})(f_{r,\gamma})_\#\mu(d\boldsymbol{z}') \\
&+ \int\int k(\boldsymbol{w}, \boldsymbol{w}')(f_{r,\gamma})_\#\nu(d\boldsymbol{w})(f_{r,\gamma})_\#\nu(d\boldsymbol{w}') \\
&- 2\int\int k(\boldsymbol{w}, \boldsymbol{w}')(f_{r,\gamma})_\#\mu(d\boldsymbol{z})(f_{r,\gamma})_\#\nu(d\boldsymbol{w}) \\
&= \int\int k(r+\gamma\boldsymbol{z}, r+\gamma\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') + \int\int k(r+\gamma\boldsymbol{w}, r+\gamma\boldsymbol{w}')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') \\
&- 2\int\int k(r+\gamma\boldsymbol{z}, r+\gamma\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w}) \\
&= \int\int \psi(\gamma\boldsymbol{z}-\gamma\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') + \int\int \psi(\gamma\boldsymbol{w}-\gamma\boldsymbol{w}')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') \\
&- 2\int\int \psi(\gamma\boldsymbol{z}-\gamma\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w}) \\
&= \int\int \gamma\tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') + \int\int \gamma\tilde{\psi}_\gamma(\boldsymbol{w}-\boldsymbol{w}')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') \\
&- 2\int\int \gamma\tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w}) \\
&= \gamma\,\mathrm{MMD}^2(\mu, \nu; \tilde{k}_\gamma)
\end{aligned}$$

Showing that $\mathrm{MMD}^2(\mu,\nu;\tilde{k}_\gamma) \leq \mathrm{MMD}^2(\mu,\nu;k)$, or equivalently $\mathrm{MMD}^2(\mu,\nu;k) - \mathrm{MMD}^2(\mu,\nu;\tilde{k}_\gamma) \geq 0$ is therefore sufficient.

$$\begin{aligned}
\mathrm{MMD}^2(\mu,\nu;k) - \mathrm{MMD}^2(\mu,\nu;\tilde{k}_\gamma) &= \Big(\int\int \psi(\boldsymbol{z}-\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') + \int\int \psi(\boldsymbol{w}-\boldsymbol{w}')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') \\
&- 2\int\int \psi(\boldsymbol{z}-\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w})\Big) - \Big(\int\int \tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') \\
&+ \int\int \tilde{\psi}_\gamma(\boldsymbol{w}-\boldsymbol{w}')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') - 2\int\int \tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w})\Big) \\
&= \int\int \psi(\boldsymbol{z}-\boldsymbol{z}') - \tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{z}')\mu(d\boldsymbol{z})\mu(d\boldsymbol{z}') \\
&+ \int\int \psi(\boldsymbol{w}-\boldsymbol{w}') - \tilde{\psi}_\gamma(\boldsymbol{w}-w')\nu(d\boldsymbol{w})\nu(d\boldsymbol{w}') \\
&- 2\int\int \psi(\boldsymbol{z}-\boldsymbol{w}) - \tilde{\psi}_\gamma(\boldsymbol{z}-\boldsymbol{w})\mu(d\boldsymbol{z})\nu(d\boldsymbol{w}) \\
&= \mathrm{MMD}^2(\mu,\nu;k-\tilde{k}_\gamma)
\end{aligned}$$

Since MMD^2 is non-negative for conditionally positive definite kernels, it is sufficient that

$$\psi(\boldsymbol{z}) - \tilde{\psi}_\gamma(\boldsymbol{z})$$

is conditionally positive definite for $\gamma \in (0,1)$. Lemma 4.4 shows that

$$\psi(\boldsymbol{z}) - \tilde{\psi}_\gamma(\boldsymbol{z}) = \sqrt{\frac{1}{\gamma^2} + h^2\|\boldsymbol{z}\|_2^2} - \sqrt{1 + h^2\|\boldsymbol{z}\|_2^2}$$

is positive definite for $\gamma \in (0,1)$, which concludes the proof. □

Corollary A.1. Let $k(\boldsymbol{x},\boldsymbol{y}) = \exp(-h^2\|\boldsymbol{x}-\boldsymbol{y}\|^2)$. Then the distributional Bellman operator is a contraction in $\mathrm{MMD}^2(\mu,\nu;k)$ for distributions μ,ν with support S_μ, S_ν such that for some $\alpha > 0$

$$\sup_{\boldsymbol{x}\in S_\mu,\boldsymbol{y}\in S_\nu} \|\boldsymbol{x}-\boldsymbol{y}\| \leq \frac{\log(\gamma^{2-\alpha})}{(\gamma^2-1)h^2}. \tag{13}$$

Proof. Utilizing the proof for Theorem 4.1, we note that all we have to do is show that

$$\gamma^\alpha k(\boldsymbol{x},\boldsymbol{y}) - k(\gamma\boldsymbol{x},\gamma\boldsymbol{y})$$

is conditionally positive definite when Equation (13) holds.

By Lemma 4.4, this means that

$$-\frac{d}{dz}\gamma^\alpha \exp(-h^2 z) - \exp(-h^2\gamma^2 z) = h^2\gamma^2\exp(-h^2\gamma^2 z) - \gamma^\alpha h^2 \exp(-h^2 z)$$

is conditionally positive definite. Let us begin by showing that the first derivative is negative.

$$\begin{aligned}
&h^2\gamma^2\exp(-h^2\gamma^2 z) - \gamma^\alpha h^2\exp(-h^2 z) \geq 0\\
\iff &\gamma^{2-\alpha}\exp(-h^2\gamma^2 z) - \exp(-h^2 z) \geq 0\\
\iff &\log(\gamma^{2-\alpha}) - h^2\gamma^2 z \geq -h^2 z\\
\iff &\log(\gamma^{2-\alpha}) \geq h^2\gamma^2 z - h^2 z\\
\iff &\frac{\log(\gamma^{2-\alpha})}{(\gamma^2-1)h^2}
\end{aligned}$$

Now we show that for $n \geq 1$

$$(-1)^n\frac{d}{dz^n}\gamma^\alpha\exp(-h^2 z) - \exp(-h^2\gamma^2 z) \geq 0$$

whenever

$$-\frac{d}{dz}\gamma^\alpha\exp(-h^2 z) - \exp(-h^2\gamma^2 z) \geq 0.$$

For $\gamma \in (0,1)$ we have

$$\begin{aligned}
(-1)^n\frac{d}{dz^n}\gamma^\alpha\exp(-h^2 z) - \exp(-h^2\gamma^2 z) &= \gamma^\alpha h^{2n}\exp(-h^2 z) - h^{2n}\gamma^{2n}\exp(-h^2\gamma^2 z)\\
&= h^{2n-1}\left(h^2\exp\left(-h^2 z\right) - h^2\gamma^{2n}\exp\left(-h^2\gamma^2 z\right)\right)\\
&\geq h^{2n-1}\left(h^2\exp\left(-h^2 z\right) - h^2\gamma^2\exp\left(-h^2\gamma^2 z\right)\right).
\end{aligned}$$

Given the first derivative $h^2\exp\left(-h^2 z\right) - h^2\gamma^2\exp\left(-h^2\gamma^2 z\right) \geq 0$ this shows that the $(-1)^n\frac{d}{dz^n}\gamma^\alpha\exp(-h^2 z) - \exp(-h^2\gamma^2 z)$ is also greater than or equal to zero. ☐