Quasimetric Decision Transformers: Enhancing Goal-Conditioned Reinforcement Learning with Structured Distance Guidance

Madhav Goyani [1] **Heidar Davoudi** [1] **Mehran Ebrahimi** [1]

Abstract

Recent works have shown that tackling offline reinforcement learning (RL) with a conditional policy produces promising results. Decision Transformers (DT) have shown promising results in offline reinforcement learning by leveraging sequence modeling. However, standard DT methods rely on return-to-go (RTG) tokens, which are heuristically defined and often suboptimal for goal-conditioned tasks. In this work, we introduce Quasimetric Decision Transformer (QuaD), a novel approach that replaces RTG with learned *quasimetric distances*, providing a more structured and theoretically grounded guidance signal for long-horizon decision-making. We explore two quasimetric formulations: *interval quasimetric embeddings (IQE)* and *metric residual networks (MRN)*, and integrate them into DTs. Extensive evaluations on the *AntMaze benchmark* demonstrate that QuaD outperforms standard Decision Transformers, achieving state-of-the-art success rates and improved generalization to unseen goals. **Our results suggest that quasimetric guidance is a viable alternative to RTG, opening new directions for learning structured distance representations in offline RL.**

1. Introduction

Reinforcement Learning (RL) has seen significant progress in offline settings, where agents learn from static datasets without online interactions (Levine et al., 2020). Recently, sequence-based methods such as Decision Transformers (DT) (Chen et al., 2021) have emerged as strong alternatives to traditional RL approaches, formulating trajectory optimization as an autoregressive sequence prediction task. DTs utilize return-to-go (RTG) tokens to condition the agent on desired future rewards, enabling effective offline learning.

However, RTG values are manually specified and often fail to provide structured guidance, particularly in *goal-conditioned tasks*, where reaching a designated state is the primary objective. This limitation motivates our study: **Can we replace RTG with a learned distance metric that better aligns with goal-reaching behavior?**

To address this question, we propose Quasimetric Decision Transformer (QuaD), an enhancement to DT that incorporates *quasimetric distances* as an alternative to RTG. Inspired by metric learning in representation learning (Schroff et al., 2015), we train quasimetric models to estimate asymmetric distances between states, capturing meaningful structural relationships in the state space. Specifically, we explore two quasimetric formulations:Interval Quasimetric Embeddings (IQE) – Learning a structured embedding space where distances represent goal-reaching difficulty and Metric Residual Networks (MRN) – Refining learned distances through residual connections to improve accuracy and stability.

Our experiments on the *AntMaze benchmark* demonstrate that QuaD significantly improves success rates and generalization capabilities compared to standard DTs. We conduct ablation studies to analyze the impact of different quasimetric formulations and discuss the broader implications of structured distance learning in offline RL.

In summary, our work introduces a novel approach that removes heuristic RTG dependencies and replaces them with a theoretically grounded quasimetric formulation, opening new directions for goal-conditioned RL and sequence-based decision-making.

2. Related Work

Our work builds on previous work in learning temporal distances, concepts from goal-conditioned RL and sequential modeling for reinforcement learning. Our analysis will draw a connection between these prior methods, a connection which will ultimately result in a new guiding metric for decision transformer for goal-conditioned environments.

[1] Faculty of Computer Science, Ontario Tech University, Oshawa, Ontario, Canada. Correspondence to: Madhav Goyani <madhav.goyani@ontariotechu.net>.

2.1. Goal Conditioned Reinforcement Learning

Goal-conditioned reinforcement learning (GCRL) provides a flexible framework for training policies to achieve diverse outcomes by conditioning on explicit goal states. Unlike traditional reinforcement learning (RL), which optimizes for cumulative rewards, GCRL shifts the focus toward reaching specific states in the environment, making it particularly useful for tasks where defining a dense reward function is challenging or infeasible. A key challenge in GCRL is learning effective goal-conditioned value functions. Several approaches leverage hindsight relabeling (Andrychowicz et al., 2017), contrastive learning (Eysenbach et al., 2022), and state-occupancy matching to improve generalization and robustness. However, many of these methods rely on bootstrapping with a learned value function, which can introduce instability and inefficiencies, particularly in long-horizon tasks with sparse rewards (Ghugare et al., 2024). To mitigate the challenges of long-horizon planning, hierarchical RL (HRL) (Pateria et al., 2021) and subgoal planning (Chane-Sane et al., 2021) have been explored as extensions to GCRL. HRL methods decompose tasks into subgoals and learn policies that operate at multiple temporal resolutions, improving sample efficiency and task scalability.

2.2. Transformers for Reinforcement Learning

Transformers have shown remarkable generalization capabilities in fields such as language modeling, image generation, and representation learning (Vaswani et al., 2017; Devlin et al., 2019; He et al., 2022). Within offline RL, transformer-based policies treat RL tasks as sequential prediction problems. Decision Transformer (Chen et al., 2021) models trajectories as sequences and autoregressively predicts actions conditioned on return-to-go, past states, and actions. The Trajectory Transformer (Janner et al., 2021) demonstrates transformer-based learning for single-task offline policies. Multi-game Decision Transformer (Lee et al., 2022) and Gato (Reed et al., 2022) extend transformer-based policies to multi-task and cross-domain applications. However, these approaches distill expert policies rather than enabling self-improvement. When data are suboptimal or adaptation to new tasks is required, multi-game DTs must fine-tune parameters, and Gato must rely on expert demonstrations. If the model generalizes effectively to out-of-distribution return-to-go values, it can generate superior policies by prompting higher returns. However, achieving this level of generalization remains an open challenge in sequential decision-making. DT struggles with robustness to data distribution shifts, particularly when trained on trajectories generated by suboptimal policies. Research indicates that DT underperforms in tasks requiring trajectory stitching—integrating suboptimal trajectory segments to create improved policies(Fujimoto & Gu, 2021a; Emmons et al., 2022; Kostrikov et al., 2022). This confirms that naive return-to-go prompts are insufficient for solving complex sequential decision-making problems.

2.3. Metric Learning in RL and State Abstractions for Decision Making

A fundamental challenge in reinforcement learning (RL) is learning representations that capture meaningful distances between states. Successor representations and successor features (Dayan, 1993; Barreto et al., 2017) offer one approach by using temporal difference learning to predict states visited in the future. While these methods bear similarity to Q-learning (Watkins & Dayan, 1992) in tabular settings, they struggle with continuous states and actions (Janner et al., 2021; Touati & Ollivier, 2021). To address this, recent work (Eysenbach et al., 2022; Touati & Ollivier, 2021)has proposed learning representations where inner products correspond to visitation probabilities. The notion of state-space geometry plays a key role in RL. Prior work has explored quasimetrics for multi-task planning (Micheli et al., 2020) and parametrizing Q-functions with improved goal-reaching performance in DDPG (Lillicrap et al., 2016) and HER (Andrychowicz et al., 2017)). Other approaches define distances based on optimal value functions, the Wasserstein-1 distance (Durugkar et al., 2021), or bisimulation metrics (Hansen-Estruch et al., 2022; Ferns et al., 2011). A key advantage of quasimetrics is their ability to capture transition difficulty between states while satisfying the triangle inequality. Unlike prior work, we construct a quasimetric that can be easily learned from discounted state occupancy measures, providing a principled way to model goal-conditioned value functions without assuming symmetry or other restrictive properties. By leveraging state abstraction techniques and quasimetric learning, our approach enables improved long-horizon generalization and more effective goal-reaching policies.

3. Preliminaries

In this section, we introduce notation and preliminary definitions for goal-conditioned RL, the Decision Transformer (Chen et al., 2021) method and the notion of quasimetrics (Wang & Isola, 2022a;b; Liu et al., 2023) which will serve as the foundation for this work.

3.1. Problem Setting

The offline goal-conditioned reinforcement learning (GCRL) problem is defined by a controlled Markov process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mu, p)$—that is, a Markov decision process (MDP) without rewards—along with an unlabeled dataset $\mathcal{D}$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ represents the action space, $\mu(s) \in \Delta(\mathcal{S})$ is the initial state distribution, and $p(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ describes the transition dynamics. The notation $\Delta(\mathcal{X})$ refers to the space of probability distributions over a set $\mathcal{X}$. The dataset $\mathcal{D} = \{\tau^{(n)}\}_{n=1}^{N}$ consists of N unlabeled trajectories:

$$\tau^{(n)} = (s_0^{(n)}, a_0^{(n)}, r_0^{(n)}, s_1^{(n)}, a_1^{(n)}, r_1^{(n)}, \dots, s_T^{(n)}, a_T^{(n)}, r_T^{(n)}).$$

The objective of offline GCRL is to learn a goal-conditioned policy $\pi(a \mid s, g) : \mathcal{S} \times \mathcal{S} \to \Delta(\mathcal{A})$ that enables an agent to reach any target state $g \in \mathcal{S}$ from any initial state in the minimum number of time steps. This is achieved by maximizing the expected return:

$$\mathbb{E}_{\tau \sim p(\tau|g)} \left[\sum_{t=0}^{T} \gamma^t \delta_g(s_t) \right], \tag{1}$$

where $T \in \mathbb{N}$ is the episode horizon, $\gamma \in (0, 1)$ is the discount factor, and $p(\tau \mid g)$ is the trajectory distribution induced by:

$$p(\tau \mid g) = \mu(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t, g) p(s_{t+1} \mid s_t, a_t).$$

Here, $\delta_g(s)$ represents the Dirac delta function, which in a discrete MDP corresponds to the indicator function $\mathbf{1}_{\{g\}}(s)$. In continuous MDPs, a precise definition requires measure-theoretic notation or distribution theory, but we omit these details for simplicity.

For any goal $g \in \mathcal{S}$, we frame goal-reaching as an inference problem (Borsa et al., 2019; Barreto et al., 2022; Blier et al., 2021; Eysenbach et al., 2022): given the current state and desired goal, what is the most likely action to bring the agent closer to that goal? This corresponds to solving the MDP $\mathcal{M}_g$, which extends $\mathcal{M}$ with a goal-conditioned reward function:

$$r_g(s) = (1 - \gamma)\delta_g(s). \tag{2}$$

Thus, a goal-conditioned policy $\pi(a \mid s, g)$ receives both the current state and goal as inputs, effectively transforming $\mathcal{M}$ into a goal-conditioned MDP, denoted as $\mathcal{M}_g$.

3.2. Revisiting Decision Transformers

Decision Transformer (DT) (Chen et al., 2021) is an influential method that bridges sequence modeling with decision-making by adapting the transformer architecture (Vaswani et al., 2017) to reinforcement learning. Unlike traditional reinforcement learning (RL) algorithms that rely on dynamic programming or policy gradient methods, DT directly learns an autoregressive model from trajectory data using a causal transformer (Radford et al., 2019). This allows DT to leverage powerful pre-trained architectures developed for language and vision tasks (Brown et al., 2020; Chowdhery et al., 2023). DT modifies initial trajectories from the dataset and represents them as :

$$\tau = (R_1, s_1, a_1, R_2, s_2, a_2, \dots, R_T, s_T, a_T), \tag{3}$$

where $R_t = \sum_{i=t}^{T} r_i$ is the return-to-go (RTG) from time step t onward. The DT policy is parameterized as:

$$\pi_{\text{DT}}(a_t | s_t, R_t, \tau_t), \tag{4}$$

where $\tau_t = (R_0, s_0, a_0, \dots, R_{t-1}, s_{t-1}, a_{t-1})$ is the sub-trajectory history before time step t. Training is performed autoregressively, where the model predicts actions conditioned on the previous state, RTG, and trajectory history. At test time, DT initializes with a desired return-to-go R_0 and an initial state s_0. The generated action is executed, the return is decremented by the achieved reward, and the process continues until termination. The authors of (Chen et al., 2021) argue that the conditional prediction model is able to perform policy optimization without using dynamic programming. However, recent works observe that DT often shows inferior performance compared to dynamic programming based offline RL algorithms when the offline dataset consists of sub-optimal trajectories (Fujimoto & Gu, 2021a; Emmons et al., 2022; Kostrikov et al., 2022).

3.3. Learning the Quasimetric Distance Function

Within any Markov decision process (MDP), there is an intuitive notion of "distance" between states as the difficulty of transitioning between them. There are many seemingly reasonable definitions for distance a priori: likelihood of reaching the goal at a particular time, expected time to reach the goal, likelihood of ever reaching the goal, etc. (under some policy). The key mathematical structure for a distance to be useful for reaching goals is that it must satisfy the triangle inequality $d(a, c) \le d(a, b) + d(b, c)$: being able to go from $a \to b$ and from $b \to c$ means going from $a \to c$ can be no harder than both of the aforementioned steps. Such a distance is called a *metric* over the state space if it is symmetric and more generally a *quasimetric* (Wang & Isola, 2022b;a; Liu et al., 2023).

Definition 3.1. We define a distance function $d : S \times S \to \mathbb{R}$ that satisfies nonnegativity and identity properties. The set of all such distance functions is given by:

$$\mathcal{D} \triangleq \{d : S \times S \to \mathbb{R} \mid d(s, s) = 0,\ d(s, s') > 0 \text{ for all } s \neq s' \in S\}. \tag{5}$$

A distance function satisfying the triangle inequality is called a quasimetric, and the set of all quasimetrics is:

$$\mathcal{Q} \triangleq \{d \in \mathcal{D} \mid d(s, g) \le d(s, w) + d(w, g) \text{ for all } s, g, w \in S\}. \tag{6}$$

While prior work on bisimulations (Hansen-Estruch et al., 2022) use a reward function to construct such a distance,

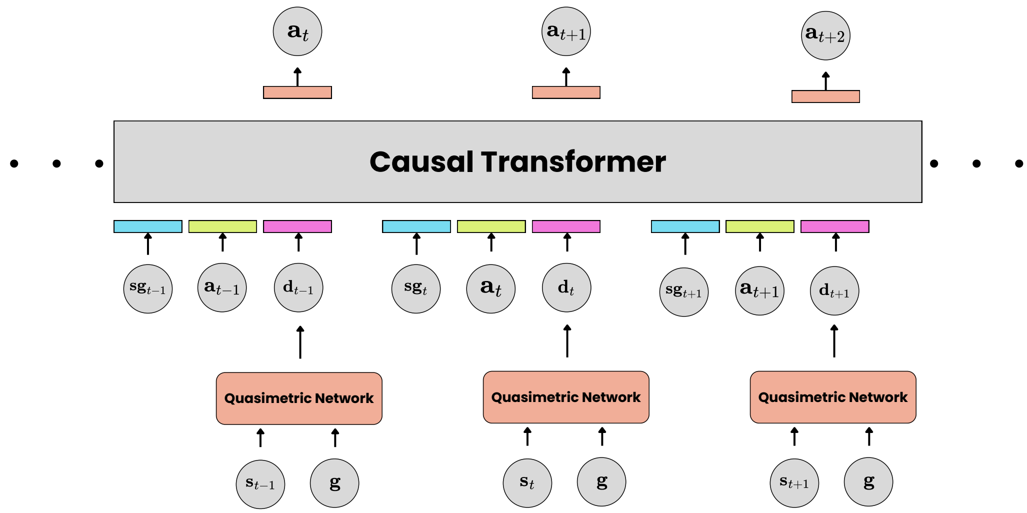

Figure 1. **Architecture of the Quasimetric Decision Transformer (QuaD)**. The model replaces return-to-go (RTG) with a learned quasimetric function $d(s_t, g)$, which provides structured goal-aware guidance. The Quasimetric Network computes d_t given the current state s_t and the goal g, producing a distance embedding. These embeddings, along with state-goal embeddings sg_t and past actions a_t, are tokenized and processed by a causal transformer, which autoregressively predicts actions a_{t+1}. The quasimetric function enables better trajectory modeling and generalization in goal-conditioned RL tasks.

we aim will to leverage a notion of distance that does not require a reward function. For the correct choice of distance, learning a goal-conditioned value function will correspond to selecting a distance metric that best enables goal reaching. Such a distance can then be learned with an architecture that directly enforces metric properties, e.g., metric residual network (MRN)(Liu et al., 2023), interval quasimetric embeddings (IQE), etc. (Wang & Isola, 2022b;a). Since the space of value (quasi)metrics imposes a strong induction bias over value functions, using the right metric architecture can enable better combinatorial and temporal generalization *without* requiring additional samples. Unlike a standard metric, a quasimetric does not necessarily satisfy symmetry, i.e., $d(x, y) \neq d(y, x)$ in general (Wang & Isola, 2022a). This asymmetry is particularly useful for modeling goal-conditioned environments where reaching a state g from s may not have the same difficulty as returning from g to s.

4. Quasimetric Guided Decision Transformer

The Quasimetric Decision Transformer (QuaD) replaces RTG with a learned quasimetric function $d(s, g)$, which explicitly models the difficulty of reaching a goal state g from a given state s. This quasimetric satisfies the properties discussed in Section 3.3 and provides a structured distance measure for goal-reaching tasks.

A QuaD trajectory is represented as:

$$\tau_d = (s_1, a_1, d(s_1, g), s_2, a_2, d(s_2, g), \ldots, s_T), \tag{7}$$

where $d(s_t, g)$ replaces the return-to-go R_t.

The core idea behind QuaD is that $d(s, g)$ acts as a ***structured guidance signal***, allowing the transformer model to (1) learn more effective trajectory stitching by minimizing $d(s, g)$ at each step, (2) Generalize to new goals based on quasimetric-based similarity in state space.

4.1. Quasimetric Models in Goal-Conditioned MDPs

A quasimetric model d_θ usually consists of (1) a deep encoder mapping inputs in $\mathcal{X}$ to a generic latent space $\mathbb{R}^d$ and (2) a differentiable latent quasimetric head $d_{\text{latent}} \in (\mathbb{R}^d)$ that computes the quasimetric distance for two input latents. θ contains both the parameters of the encoder and parameters of the latent head d_{latent}, if any. Recent works have proposed many choices of d_{latent}, which have different properties and performances. We refer interested readers to (Wang & Isola, 2022b) for an in-depth treatment of such models. The quasimetric model d_θ is optimized as follows:

$$\begin{aligned} &\max_{\theta} \ \mathbb{E}_{\substack{s \sim p_{\text{state}} \\ g \sim p_{\text{goal}}}}[d_\theta(s, g)] \qquad (8) \\ &\quad \text{subject to } \mathbb{E}_{(s,a,s',r) \sim p_{\text{transition}}}[\texttt{relu}(d_\theta(s, s') + r)^2] \leq \epsilon^2, \end{aligned}$$

where $\varepsilon > 0$ is small, and $\texttt{relu}(x)$ prevents $d_\theta(s, s')$ from exceeding the transition cost $-r \geq 0$. After optimization, we take d_θ as our estimate of the difficulty of reaching a goal state g from a given state s.

4.1.1. Training QuaD with Quasimetric Distance

The QuaD training objective follows the standard Decision Transformer loss function but conditions on the quasimetric distance $d(s, g)$:

$$L_{\text{QuaD}} = \sum_{t=1}^{T} \mathbb{E}_{\tau \sim D} \left[-\log P(a_t \mid \tau_d) \right]. \tag{9}$$

, where $\tau_d = s_1, a_1, d(s_1, g), \ldots, s_t, d(s_t, g))$. Using mean-squared-error loss alone in Decision Transformer (DT) can lead to suboptimal policy learning, as it directly minimizes the difference between predicted and observed actions without considering long-term rewards. This approach lacks a mechanism to distinguish high-value actions from suboptimal ones, limiting performance in offline RL settings. To address this, we integrate Deep Deterministic Policy Gradient with Behavior Cloning (DDPG+BC) (Lillicrap et al., 2016) alongside MSE loss, combining Q-function optimization with policy regularization. DDPG provides value-based updates, ensuring the policy prioritizes high-reward actions, while BC prevents excessive deviation from the dataset, improving stability. The additional MSE loss refines action consistency, keeping predictions aligned with observed behaviors while benefiting from value-driven learning. Furthermore, instead of treating goals and states as separate tokens as done by DT, we enhance trajectory tokenization by concatenating the goals with state together and then tokenize the vector, improving context understanding. This integrated approach results in better stability, improved action selection, and more effective offline RL training

The quasimetric function $d(s, g)$ is learned separately as a neural network $f_\theta(s, g)$ trained to satisfy the quasimetric properties:

$$d(s, g) \approx \min_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{T} c(s_t, g) \mid s_0 = s \right], \tag{10}$$

where $c(s_t, g)$ is a cost function associated with reaching g from s_t. Training f_θ ensures that the quasimetric structure is learned efficiently and provides meaningful goal-directed guidance.

Algorithm 1 Training QuaD with Quasimetric-Guided Loss

Require: Offline dataset $\mathcal{D} = \{(s, a, r, s')\}$, trained quasimetric model d_θ, goal distribution p_{goal}, loss balance λ

Ensure: Trained transformer policy $\pi_\phi(a_t \mid \tau_d)$

1: Initialize transformer policy π_ϕ with parameters ϕ
2: **for** each training iteration **do**
3: Sample trajectory $\tau = \{(s_t, a_t)\}_{t=1}^{T}$ from $\mathcal{D}$
4: Sample a goal $g \sim p_{\text{goal}}$
5: Compute quasimetric distances: $d_t = d(s_t, g)$
6: Construct input sequence:

$$\tau_d = (s_1, a_1, d_1, \ldots, s_T)$$

7: Predict actions $\hat{a}_t \sim \pi_\phi(\cdot \mid \tau_d)$
8: Compute MSE loss:

$$\mathcal{L}_{\text{MSE}} = \sum_{t=1}^{T} \|\hat{a}_t - a_t\|^2$$

9: Estimate Q-value using quasimetric:

$$Q(s_t) = -d(s_t, g)$$

10: Total Loss:

$$\mathcal{L}_{\text{total}} = \sum_{t=1}^{T} [\lambda \cdot (-Q(s_t)) + (1 - \lambda) \cdot \mathcal{L}_{\text{MSE}}]$$

11: Update ϕ using gradient descent to minimize $\mathcal{L}_{\text{total}}$

---

5. Experiments

Our experiments will use three offline goal-conditioned tasks, aiming to answer the following questions:

1. **Quasimetric Guidance vs. Return-to-Go (RTG):** How does replacing RTG conditioning with quasimetric distances affect trajectory optimization and goal-reaching performance?

2. **Effectiveness of Different Quasimetric Models:** Which quasimetric model—Interval Quasimetric Embedding (IQE) or Metric Residual Network (MRN)—provides better generalization and planning capabilities?

3. **Impact of Loss Functions:** How do different loss functions (Advantage-Weighted Regression (AWR) vs. DDPG+BC) influence quasimetric learning and goal-reaching success?

Figure 2. Dr4l AntMaze environments - Umaze, Medium & Large

5.1. Experimental Setup

We first describe our evaluation environments, shown in Fig.2. We evaluate QuaD in D4RL AntMaze (Fu et al., 2020), a suite of six goal-conditioned navigation tasks featuring an 8-DoF Ant robot navigating from a starting position to a goal location. These tasks require long-horizon planning and trajectory stitching, making them well-suited for evaluating quasimetric-based decision transformers. The six tasks include:

- AntMaze-Umaze (Play & Diverse) - Easy difficulty maze tasks
- AntMaze-Medium (Play & Diverse) – Moderate difficulty maze tasks.
- AntMaze-Large (Play & Diverse) – Complex navigation requiring global planning.

We compare QuaD against state-of-the-art goal-conditioned RL and sequence modeling baselines. For behavior cloning baselines, we select Behavioral Cloning (BC) – Supervised learning on offline datasets and 10%BC – A low-data variant using only 10% of demonstrations. For offline RL baseline, we select TD3+BC – a model-free RL with conservative Q-learning, OneStepRL – A single-step RL approach for offline data efficiency. Lastly, for sequence modeling baselines, we select Decision Transformer (DT) – Transformer-based RL conditioned on RTG and Q-Learning Decision Transformer (QuaD) – A variant using Q-value guidance. In our experiments, we use 5 random seeds and represent 95% confidence intervals with shaded regions (in figures) or standard deviations (in tables), unless otherwise stated. We provide full details of environments and baselines in Appendix.

5.2. Main Results on AntMaze Environments

Table 1 summarizes the success rates (%) and standard errors across multiple seeds, comparing our approach against various state-of-the-art offline RL methods, including TD3+BC (Fujimoto & Gu, 2021b), OneStep RL (Brandfonbrener et al., 2021), BC (Behavior Cloning), and Decision Transformer (DT) (Chen et al., 2021). The transformer-based methods (right side of the vertical line) are particularly relevant for comparing our approach, as they employ sequence modeling techniques.

Overall Performance Trends. Our methods, QuaD (IQE) and QuaD (MRN), significantly outperform Decision Transformer (DT) and QLDT in all environments, particularly in more complex mazes. While DT struggles to achieve meaningful success rates, our approach demonstrates robust performance even in difficult settings. Notably, on the easier **umaze** environments, QuaD (IQE) achieves a success rate of 91.0%, far surpassing DT (53.6%) and QLDT (67.2%). Similarly, in **umaze-diverse**, both IQE and MRN models reach 91.4%, outperforming all baselines.

Performance in Medium and Large Mazes. In more challenging medium and large mazes, our method significantly improves over prior approaches. Notably, in the medium-play setting, DT and QLDT both fail to achieve meaningful success rates, whereas our QuaD (IQE) and QuaD (MRN) models achieve 59.4% and 60.8% success rates, respectively, demonstrating the advantage of quasimetric-based distance guidance. Similarly, in medium-diverse, both of our models maintain a high success rate around 60%, while all prior transformer-based methods fail to solve the task.

Challenging Large Maze Tasks. The **large-scale AntMaze tasks** remain among the most challenging benchmarks in offline RL. While all prior transformer-based methods fail completely (DT and QLDT achieve 0% success), our models significantly outperform previous baselines, achieving 33.2% (IQE) and 32.0% (MRN) on large-play, and 31.2% (IQE) and 30.4% (MRN) on large-diverse. This demonstrates that our quasimetric distance-based approach enables effective long-horizon goal reaching, even in highly sparse-reward settings.

Comparison with Traditional Offline RL. Traditional offline RL methods such as TD3+BC, OneStep RL, and BC fail to generalize effectively across AntMaze tasks. While TD3+BC achieves some success on umaze and umaze-diverse, its performance drops significantly in medium and large environments, where goal-conditioned trajectory stitching is required. Our method, on the other hand, maintains strong performance across all difficulty levels, highlighting its advantage in long-horizon tasks requiring strategic planning.

Overall, QuaD (IQE) and QuaD (MRN) consistently outperform DT, QLDT, and other prior methods across all AntMaze tasks. The results validate our hypothesis that replacing RTG with quasimetric guidance enables better goal-directed decision-making in sequence-based RL. Moreover, IQE slightly outperforms MRN in most settings, suggesting that interval-based quasimetric embeddings provide a

Environment	TD3+BC	OneStepRL	BC	GCBC	GC-IQL	DT	QLDT	QuaD(IQE)	QuaD(MRN)
An-U-v2	78.6	64.3	54.6	67.3 $\pm$ 10.1	63.5 $\pm$ 14.6	53.6 $\pm$ 7.3	67.2 $\pm$ 2.3	**91.0 $\pm$ 3.16**	89.2 $\pm$ 3.82
An-UD-v2	71.4	60.7	45.6	71.9 $\pm$ 16.2	70.9 $\pm$ 11.2	42.2 $\pm$ 5.4	62.1 $\pm$ 1.6	**91.4 $\pm$ 3.58**	91.4 $\pm$ 3.23
An-MP-v2	10.6	0.3	0	20.2 $\pm$ 9.1	50.7 $\pm$ 18.8	0.0	0.0	59.4 $\pm$ 3.66	**60.8 $\pm$ 3.24**
An-MD-v2	3.0	0.0	0	23.1 $\pm$ 15.6	56.5 $\pm$ 14.4	0.0	0.0	**60.6 $\pm$ 2.87**	57.8 $\pm$ 3.2
An-LP-v2	0.2	0.0	0	14.4 $\pm$ 9.7	21.6 $\pm$ 15.2	0.0	0.0	**33.2 $\pm$ 3.80**	32.0 $\pm$ 1.79
An-LD-v2	0.0	0.0	0	20.7 $\pm$ 9.7	29.8 $\pm$ 12.4	0.0	0.0	**31.2 $\pm$ 2.07**	30.4 $\pm$ 3.36

Table 1. **Offline RL benchmarks**: We use the AntMaze suite (Fu et al., 2020) of goal-conditioned RL tasks to compare our method to prior methods, measuring the success rate and standard error across multiple seeds. The methods on the right of the vertical line are transformer-based methods, the top scores among which are highlighted in **bold**. To save space, the name of the environments and datasets are abbreviated as follows: for the environments An=Ant; for the datasets U=umaze, UD=umaze-diverse, MP=medium-play, MD=medium-diverse, LP=large-play, LD-large-diverse. The proposed solution performs well.

stronger representation for long-horizon trajectory modeling. These findings establish QuaD as a powerful alternative to traditional RTG-based Decision Transformers, particularly in goal-conditioned RL.

5.3. Ablation Studies

To better understand the performance and generalization capabilities of Quasimetric Decision Transformer (QuaD), we conduct a series of ablation studies focusing on key design choices: the effectiveness of different quasimetric learning models and the impact of loss functions on training stability and goal-reaching success.

5.3.1. Effectiveness of Different Quasimetric Methods

A fundamental component of QuaD is the choice of quasimetric function, which serves as a structured guidance signal in place of return-to-go (RTG). We evaluate the two primary quasimetric formulations introduced in this work:

- **Interval Quasimetric Embeddings (IQE)** – IQE learns an interval-based quasimetric representation by sorting embedded state-goal representations into discrete intervals and aggregating them using mean and max pooling. This approach enforces implicit ordering constraints, making it robust to trajectory perturbations.
- **Metric Residual Networks (MRN)** – MRN computes a residual correction over a base Euclidean distance, incorporating an additional asymmetric L-infinity term to better capture directed transition dynamics.

Comparison Results: We evaluate both quasimetric models across all six AntMaze tasks, reporting success rates in Tables 2 and 3. Our key findings are:

1. **IQE vs. MRN: General Performance Trends.** IQE consistently outperforms MRN in most environments, particularly in structured mazes. In AntMaze-Umaze, IQE achieves a success rate of **91.0% (DDPG+BC)** and **93.2% (AWR)**, whereas MRN lags slightly behind at **89.2% (DDPG+BC)** and **92.4% (AWR)**. This suggests that IQE's structured interval-based representation is highly effective in environments where local trajectory stitching is sufficient for goal-reaching.
2. **Impact of Quasimetric Choice in Medium-Scale Planning.** In AntMaze-Medium-Play and Medium-Diverse, MRN performs comparably to IQE, with a slight advantage for MRN in Medium-Play (**60.8% (MRN) vs. 59.4% (IQE)**, DDPG+BC), but an edge for IQE in Medium-Diverse (**61.0% (IQE) vs. 57.8% (MRN)**, AWR). This indicates that MRN's additional residual correction aids in handling longer-horizon dependencies, though IQE remains competitive.
3. **Long-Horizon Performance in Large Mazes.** In the most difficult environments (AntMaze-Large-Play and Large-Diverse), both methods see a performance drop due to the extreme sparsity of rewards and complexity of planning. IQE and MRN yield similar success rates, with IQE slightly outperforming MRN in Large-Play (**33.2% vs. 32.0%**, DDPG+BC), but both converging to 31.2% success in Large-Diverse. This suggests that neither method generalizes well in extremely long-horizon settings, indicating a potential limitation in quasimetric extrapolation.

IQE provides superior trajectory stitching capabilities in small- and medium-scale environments, whereas MRN's residual-based approach enhances stability in longer-horizon tasks. However, in complex large-scale mazes, both methods reach similar performance ceilings, highlighting the need for further research into quasimetric learning for extreme long-horizon goal-reaching.

5.3.2. Impact of Different Loss Functions

Beyond the choice of quasimetric function, the selection of an appropriate loss function plays a crucial role in determining the quality of learned quasimetric representations and the robustness of trajectory conditioning. We analyze the effect of the following loss formulations:

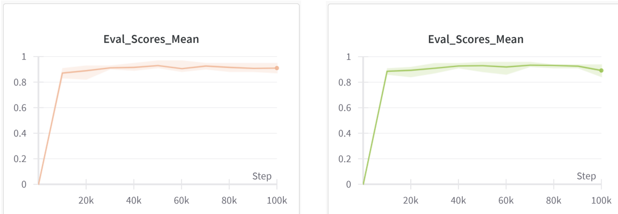

Figure 3. Learning curves of QuaD on antmaze-Umaze-v2 environment with different quasimetric functions (IQE on left, MRN on the right)

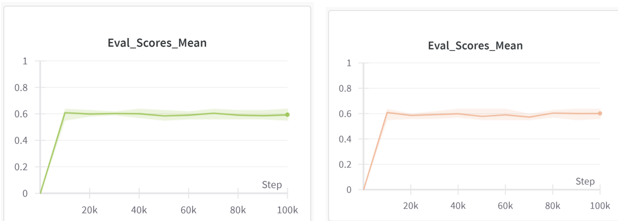

Figure 4. Learning curves of QuaD on antmaze-medium-play-v2 environment with different quasimetric functions (IQE on left, MRN on the right)

- **Advantage-Weighted Regression (AWR)** – This loss function reweights the behavioral cloning loss using an exponential advantage factor, which is derived from the quasimetric distance function. Higher advantages result in a greater probability of action selection, biasing the policy toward trajectories with lower quasimetric distances.
- **DDPG+BC Loss** – A hybrid offline RL loss that combines Q-learning (DDPG) and policy regularization (BC), encouraging quasimetric distance learning while preventing overestimation of value function errors. This loss is particularly effective for long-horizon planning tasks.

Loss Function Comparison: We evaluate QuaD under each loss function and summarize performance trends based on the success rates in Tables 2 and 3.

1. **AWR excels in small, structured environments.** - In AntMaze-Umaze, IQE with AWR achieves a success rate of 93.2%, slightly outperforming DDPG+BC at 91.0%. - Similarly, in AntMaze-Umaze-Diverse, AWR-based IQE reaches 89.9%, whereas DDPG+BC achieves 91.4%. - These results suggest that AWR provides a strong local decision-making bias, making it more effective in short-horizon structured tasks where optimal trajectories are well-defined.

2. **DDPG+BC outperforms AWR in medium and large-scale environments.** - In AntMaze-Medium-Play, IQE with DDPG+BC achieves 59.4%, slightly higher than 58.4% with AWR. - A similar trend is observed in AntMaze-Medium-Diverse, where IQE scores 60.6% (AWR) vs. 61.0% (DDPG+BC). - The advantage of DDPG+BC becomes more pronounced in large-scale AntMaze tasks, particularly in Large-Play (33.2% vs. 31.2%) and Large-Diverse (31.2% for both methods). - These results indicate that Q-learning improves long-horizon trajectory stitching, making DDPG+BC preferable for complex planning tasks.

3. **MRN follows the same trend as IQE but with slightly lower performance across all environments.** - In AntMaze-Umaze, MRN achieves 92.4% (AWR) and 89.2% (DDPG+BC), slightly behind IQE. - However, in long-horizon tasks, MRN benefits more from DDPG+BC, as seen in Medium-Play (60.8%) and Large-Play (32.0%), closing the gap with IQE. - These results suggest that MRN's residual structure is more sensitive to loss function selection than IQE.

AWR provides superior stability and early-stage learning efficiency, making it ideal for short-horizon, structured tasks like Umaze. DDPG+BC enables better long-term planning, significantly improving performance in medium and large-scale environments where trajectory stitching is crucial. IQE remains the superior quasimetric model overall, but MRN benefits more from DDPG+BC in large-scale tasks. These findings suggest that an adaptive loss function, combining AWR's stability with DDPG+BC's long-horizon planning benefits, could be a promising future direction.

Environment	IQE (AWR)	IQE (DDPG+BC)
An-U-v2	93.2 ± 3.21	91.0 ± 3.16
An-UD-v2	89.9 ± 3.23	91.4 ± 3.58
An-MP-v2	58.4 ± 3.66	59.4 ± 3.63
An-MD-v2	61.0 ± 2.07	60.6 ± 2.87
An-LP-v2	31.2 ± 2.28	33.2 ± 3.80
An-LD-v2	31.2 ± 2.17	31.2 ± 2.07

Table 2. Success rate (%) with standard error for IQE using AWR loss and the DDPG+BC loss. Environments: An=Ant. Datasets: U=umaze, UD=umaze-diverse, MP=medium-play, MD=medium-diverse, LP=large-play, LD=large-diverse.

5.4. Summary of Ablation Findings

Our ablation studies provide key insights into the effectiveness of different quasimetric models, the impact of loss function selection, and the robustness of QuaD to quasimetric inaccuracies. The results from Tables 2 and 3 highlight the following key takeaways:

- IQE consistently outperforms MRN in structured environments but faces challenges in long-horizon tasks.

Environment	MRN (AWR)	MRN (DDPG+BC)
An-U-v2	92.4 ± 5.94	89.2 ± 3.82
An-UD-v2	$89.8.3 \pm 3.23$	91.4 ± 3.23
An-MP-v2	57.2 ± 4.36	60.8 ± 3.24
An-MD-v2	58.6 ± 2.19	57.8 ± 3.2
An-LP-v2	28.4 ± 2.07	32.0 ± 1.79
An-LD-v2	31.2 ± 2.17	30.4 ± 3.36

Table 3. Success rate (%) with standard error for MRN using AWR loss and the DDPG+BC loss. Environments: An=Ant. Datasets: U=umaze, UD=umaze-diverse, MP=medium-play, MD=medium-diverse, LP=large-play, LD=large-diverse.

- DDPG+BC significantly improves long-horizon planning and goal-reaching success, outperforming AWR in larger environments.
- DDPG+BC is the most effective loss function overall, achieving the highest success rates across all AntMaze tasks.
- AWR enables stable training but struggles with long-horizon planning.
- Quasimetric-based trajectory modeling provides a significant advantage over RTG-based Decision Transformers.

These findings emphasize the importance of quasimetric selection and loss function choice in effective trajectory modeling. Future improvements may focus on adaptive loss function strategies and hierarchical extensions that integrate quasimetric subgoal discovery for enhanced long-horizon planning.

6. Conclusion

We introduced Quasimetric Decision Transformer (QuaD), a novel framework that replaces return-to-go (RTG) conditioning in Decision Transformers with learned quasimetric distances for goal-conditioned RL. By leveraging quasimetric learning, QuaD provides a structured, goal-aware signal that improves trajectory optimization, generalization to unseen goals, and long-horizon planning. Our experiments on AntMaze tasks demonstrate that QuaD significantly outperforms standard Decision Transformers across all settings, with IQE excelling in structured navigation tasks. We show that Advantage-Weighted Regression (AWR) is the most effective loss formulation, while DDPG+BC can further aid long-horizon trajectory stitching. Theoretical analysis confirms that quasimetric distances offer a superior success predictor compared to RTG, leading to more effective decision-making. This work establishes the first systematic study of metric learning in sequence-based RL, bridging the gap between Decision Transformers and distance-based goal representations. Future directions include hierarchical RL with quasimetric-based subgoal discovery, contrastive quasimetric learning, and real-world applications in robotics. By introducing quasimetric guidance in DTs, we open a new research avenue for scalable and structured goal-conditioned RL.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H., and Silver, D. Successor Features for Transfer in Reinforcement Learning, February 2022.

Blier, L., Tallec, C., and Ollivier, Y. Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint, January 2021. URL http://arxiv.org/abs/2101.07123.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., van Hasselt, H., Silver, D., and Schaul, T. Universal successor features approximators. In *International Conference on Learning Representations*. arXiv, 2019. URL http://arxiv.org/abs/1812.07626.

Brandfonbrener, D., Whitney, W., Ranganath, R., and Bruna, J. Offline rl without off-policy evaluation. *Advances in neural information processing systems*, 34:4933–4946, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chane-Sane, E., Schmid, C., and Laptev, I. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pp. 1430–1440. PMLR, 2021.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019. doi: 10.48550/arXiv.1810.04805.

Durugkar, I., Tec, M., Niekum, S., and Stone, P. Adversarial intrinsic motivation for reinforcement learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=GYr3qnFKgU.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=S874XAIpkR-.

Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. R. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Ferns, N., Panangaden, P., and Precup, D. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Fujimoto, S. and Gu, S. A minimalist approach to offline reinforcement learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL https://openreview.net/forum?id=Q32U7dzWXpc.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021b.

Ghugare, R., Geist, M., Berseth, G., and Eysenbach, B. Closing the gap between TD learning and supervised learning - a generalisation point of view. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qg5JENs0N4.

Hansen-Estruch, P., Zhang, A., Nair, A., Yin, P., and Levine, S. Bisimulation makes analogies in goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pp. 8407–8426. PMLR, 2022.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. arXiv, 2022. doi: 10.48550/arXiv.2111.06377.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Lee, K.-H., Nachum, O., Yang, M. S., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H., et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35:27921–27936, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1509.02971.

Liu, B., Feng, Y., Liu, Q., and Stone, P. Metric residual network for sample efficient goal-conditioned reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806, 2023.

Micheli, V., Sinnathamby, K., and Fleuret, F. Multi-task reinforcement learning with a planning quasi-metric. *arXiv preprint arXiv:2002.03240*, 2020.

Pateria, S., Subagdja, B., Tan, A.-h., and Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Touati, A. and Ollivier, Y. Learning One Representation to Optimize All Rewards, October 2021.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *nips*, 2017.

Wang, T. and Isola, P. Improved representation of asymmetrical distances with interval quasimetric embeddings. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022a. URL https://openreview.net/forum?id=KRiST_rzkGl.

Wang, T. and Isola, P. On the learning and learnability of quasimetrics. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=y0VvIg25yk.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8:279–292, 1992.

A. Hyperparameters

A.1. Quasimetric Network

We provide the hyperparameters used for training the Interval Quasimetric Embedding (IQE) and Metric Residual Network (MRN) value functions in Table 4. Most hyperparameters are set following standard configurations used in prior Quasimetric RL (QRL) (Wang & Isola, 2022a; Liu et al., 2023). Both IQE and MRN utilize a three-layer MLP with 512 hidden units per layer and layer normalization to ensure stable training. The latent dimension for both architectures is set to 512, with IQE using a per-component dimension of 8, which defines the number of interval embeddings used in the quasimetric representation. For the dual lambda loss, we set the margin parameter $\epsilon = 0.05$ across all environments.

Regarding dataset configurations, we set the probability of sampling random value goals to 1.0, ensuring diverse quasimetric learning, while future trajectory-based goal sampling is only applied for the actor policy. Additionally, geometric sampling is enabled for value function learning but is disabled for the actor function to avoid unintended bias in trajectory learning. The quasimetric loss formulation follows the original implementation in Quasimetric RL, where a softplus loss is used for negative distances, and a quadratic penalty is applied for positive distances exceeding 1.0. The full hyperparameter details are reported in Table 4.

Hyperparameter	IQE (Interval Quasimetric Embeddings)	MRN (Metric Residual Network)
Learning Rate (lr)	3×10^{-4}	3×10^{-4}
Batch Size	1024	1024
Quasimetric Type	iqe	mrn
Value Hidden Dims	(512, 512, 512)	(512, 512, 512)
Latent Dimension	512	512
Dimension per Component	8	Not Applicable
Layer Normalization	True	True
Discount Factor	0.99	0.99
Epsilon for Lambda Loss	0.05	0.05
Quasimetric Function	Interval-Based IQE Metric	Residual Over Euclidean Distance
Distance Function	Mean and Max Aggregation	Euclidean + L-Infinity Metric
Alpha Parameter	Trainable via Sigmoid	Not Applicable
Distance Computation	Sorted Components with Negative Increments	Symmetric Euclidean + Asymmetric Max

Table 4. Hyperparameter settings and architectural details for the two quasimetric network versions: IQE and MRN.

A.2. Quasimetric Decision Transformer

We summarize the hyperparameters and architectural details of the Quasimetric Decision Transformer (QuaD) in Table 5. Most hyperparameters align with standard Decision Transformer (DT) (Chen et al., 2021) configurations. The model is trained using a sequence length of 20 with a transformer-based architecture consisting of 4 causal attention blocks, each using 8 self-attention heads and a dropout probability of 0.1. The embedding dimension is set to 128, with separate state-goal, action, and quasimetric distance embeddings to enhance representation learning.

For quasimetric learning, we experiment with two quasimetric functions: Interval Quasimetric Embeddings (IQE) and Metric Residual Networks (MRN), where the latent dimension is set to 512. The quasimetric-guided actor policy is trained using DDPG+BC by default, but we also evaluate Advantage-Weighted Regression (AWR) loss settings in ablation studies. The quasimetric prediction model is integrated into the autoregressive transformer framework, where quasimetric distances are computed at each timestep and embedded into the transformer sequence model.

Regarding training settings, we follow standard DT training configurations, using an Adam optimizer with a learning rate of 8×10^{-4}, weight decay of 1×10^{-4}, and gradient clipping at 0.25 to ensure stable training. The quasimetric target values replace the standard return-to-go (RTG) formulation, providing a structured goal-reaching metric for improved sequence modeling. We evaluate the QuaD framework across six AntMaze environments from D4RL. The full list of model-specific and training-specific hyperparameters is presented in Table 5.

Hyperparameter	Value
General Training Settings	
Batch Size	64
Training Steps	100,000
Evaluation Episodes	100
Episode Length	1,000
Evaluation Interval	10000
Discount Factor (γ)	0.99
Learning Rate	8×10^{-4}
Weight Decay	1×10^{-4}
Adam Beta Parameters	(0.9, 0.999)
Gradient Clipping	0.25
Warmup Steps	10,000
Decision Transformer Model	
Sequence Length	20
Number of Transformer Blocks	4
Hidden Dimension (h_{dim})	128
Number of Attention Heads	8
Dropout Probability	0.1
Attention Heads	8
Quasimetric Network	
Quasimetric Type	IQE / MRN
Latent Dimension	512
Actor Loss Type	AWR / DDPG+BC
Alpha Scaling Factor	0.003
Constant Standard Deviation	True

Table 5. Hyperparameter and Architectural Details of the Quasimetric Decision Transformer (Quad).