arXiv:2512.06917v1 [cs.LG] 7 Dec 2025

Know your Trajectory - Trustworthy Reinforcement Learning deployment through Importance-Based Trajectory Analysis

Clifford F[1], Devika Jay[1], Abhishek Sarkar[2], Satheesh K Perepu[2], Santhosh G S[1], Kaushik Dey[2], Balaraman Ravindran[1]

[1]Centre for Responsible AI, IIT Madras, India
[2]Ericsson Research, Bangalore, India
na20b014@smail.iitm.ac.in, devika@cerai.in, abhishek.sarkar@ericsson.com, perepu.satheesh.kumar@ericsson.com, santhoshgs013@gmail.com, deykaushik@ericsson.com, ravi@dsai.iitm.ac.in

Abstract

As Reinforcement Learning (RL) agents are increasingly deployed in real-world applications, ensuring their behavior is transparent and trustworthy is paramount. A key component of trust is explainability, yet much of the work in Explainable RL (XRL) focuses on local, single-step decisions. This paper addresses the critical need for explaining an agent's long-term behavior through trajectory-level analysis. We introduce a novel framework that ranks entire trajectories by defining and aggregating a new state-importance metric. This metric combines the classic Q-value difference with a "radical term" that captures the agent's affinity to reach its goal, providing a more nuanced measure of state criticality. We demonstrate that our method successfully identifies optimal trajectories from a heterogeneous collection of agent experiences. Furthermore, by generating counterfactual rollouts from critical states within these trajectories, we show that the agent's chosen path is robustly superior to alternatives, thereby providing a powerful "Why this, and not that?" explanation. Our experiments in standard OpenAI Gym environments validate that our proposed importance metric is more effective at identifying optimal behaviors compared to classic approaches, offering a significant step towards trustworthy autonomous systems.[1]

Introduction

The increasing sophistication of Reinforcement Learning (RL) has enabled the training of agents for complex tasks, accelerating their deployment in diverse real-world systems. However, for these autonomous agents to be deemed trustworthy and responsible, their decision-making processes must be explainable. The field of Explainable RL (XRL) aims to provide high-fidelity, human-comprehensible explanations for an agent's behavior.

While a significant portion of XRL research has concentrated on local explanations, justifying a specific action in a given state (Amitai, Septon, and Amir 2024), these methods fall short of clarifying an agent's long-term strategy. Understanding the overarching "story" of an agent's behavior, encapsulated by its trajectory, is crucial for deployment in safety-critical domains. For instance, knowing why a self-driving car chose a particular route over another is more informative than knowing why it braked at a single intersection.

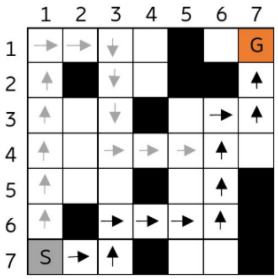


Figure 1: An agent's observed trajectory (black) versus a longer, suboptimal alternative (gray). Our goal is to explain why the black path was chosen by demonstrating its superior importance.

To address this gap, we propose a framework for explaining entire trajectories by leveraging causal inference concepts. Our approach produces explanations by answering contrastive questions like "Why was this path taken?" and "Why was an alternative path not taken?". Answering such questions involves generating counterfactuals, what could have happened, which is a powerful tool for human-like reasoning.

This paper makes the following contributions:

- We introduce a novel state-importance metric that augments the standard Q-value difference with a "radical term" representing the agent's goal affinity, allowing for a more robust evaluation of state criticality.
- We propose a complete pipeline to rank entire trajectories by aggregating our state-importance metric, enabling the identification of the most salient and representative behaviors from a large dataset of experiences.
- We empirically validate our approach by generating counterfactuals from top-ranked trajectories, demonstrating that our method effectively highlights the optimality of the agent's chosen path comparing viable alternatives.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Code and set up to reproduce our experiments are available at: https://github.com/clif-ford/XRL_Codebase

This work advances the state of deployable AI by providing a practical method for generating high-level strategic explanations, fostering greater trust and transparency in complex autonomous systems.

Related Work

Explainable RL (XRL) is a well developing field, with methods broadly categorized into Feature Importance (FI), Learning Process and MDP (LPM), and Policy Level (PL) explanations (Milani et al. 2024). Our work falls under the PL category, with a specific focus on trajectory-level explanations.

PL methods aim to explain high-level policy decisions. This includes summarizing key transitions (Amir and Amir 2018), converting complex recurrent policies into interpretable formats like finite state automata (Danesh et al. 2021), or extracting prototypical "landmark" states from experience (McCalmon et al. 2022).

Several works have specifically targeted trajectory explanations. The HIGHLIGHTS method (Amir and Amir 2018) provides summaries by selecting states with the highest potential impact on future rewards, based on Q-values. While effective, it summarizes behavior through discrete states rather than analyzing the full trajectory sequence. Other approaches have used offline data to cluster trajectories and train surrogate policies to identify dissimilarities, attributing importance to clusters that cause the largest policy divergence (Deshmukh et al. 2024). However, the interpretability of these clusters can be a challenge. (Frost et al. 2022) uses a learned policy to answer counterfactual queries by performing rollouts from matched states in a source domain, presenting these what-if scenarios to a user. More recently, visualization techniques have been used to abstract trajectories by clustering latent state representations, helping to visualize major state transitions for non-experts (Takagi et al. 2024).

Our work builds on these ideas by proposing a novel, principled metric for quantifying trajectory importance directly from the agent's value function. Unlike methods that rely on clustering or summarizing discrete states, our approach evaluates the entire sequential path, providing a holistic and contrastive explanation through counterfactual analysis.

Methodology

Our framework is designed to identify and explain the most significant trajectories from an agent's experience. The core of our approach lies in a novel definition of state and trajectory importance.

State Importance: A Classic View

We begin with the intuitive notion that a state is important if the choice of action within it has a significant impact on future rewards. This is classically defined using the agent's Q-values (Amir et al. 2016). The importance of a state s, denoted $I(s)$, is the difference between the values of the best and worst possible actions:

$$I(s) = \max_a Q^\pi(s,a) - \min_a Q^\pi(s,a)$$

This quantity, $\Delta Q(s)$, captures the potential advantage available in state s. A high value indicates a critical decision point where a suboptimal action can be costly.

A Modified Importance Metric

While $\Delta Q(s)$ measures the potential gain, it does not capture the agent's confidence or decisiveness in pursuing the optimal action. A state might have a high $\Delta Q(s)$, but if the agent's policy is nearly uniform over several good actions, the state is less critical than one where a single action is decisively superior.

To address this, we introduce a modified state-action importance metric that incorporates the agent's affinity for reaching the goal. We define this as:

$$I(s,a) = \Delta Q(s) \times R(s,a)$$

Here, $R(s,a)$ is a "radical term" that quantifies the agent's commitment to its chosen path. We explored several formulations for $R(s,a)$:

1. **Normalization (Naive)**: Measures how much better the chosen action is relative to the average action: $r(s,a) = (Q(s,a) - \mu_Q(s))/\sigma_Q(s)$.
2. **Bellman Error**: Uses the temporal difference error, $|Q(s,a) - (r + \gamma Q(s',a'))|$, as a measure of deviation from optimality.
3. **Entropy-Based Confidence**: Measures the decisiveness of the policy $\pi(a|s)$. We define confidence as normalized negative entropy: $r(s) = 1 - (H(\pi(s))/\log|\mathcal{A}|)$, where $r(s) \to 1$ for a deterministic policy.
4. **Value-Based Goal Proximity**: Uses the state-value function $V(s)$ as a proxy for closeness to the goal. This can be normalized using a known range, $r(s) = (V(s) - V_{\min})/(V_{\max} - V_{\min})$, or relative to the goal state's value, $r(s) = |V(s)/V(s_{\text{final}})|$.

Through experimentation, we found that the value-based goal proximity metric (V_{goal}) provided the most consistent and meaningful results, as it directly encodes progress towards the task objective.

Trajectory Importance and Explanation

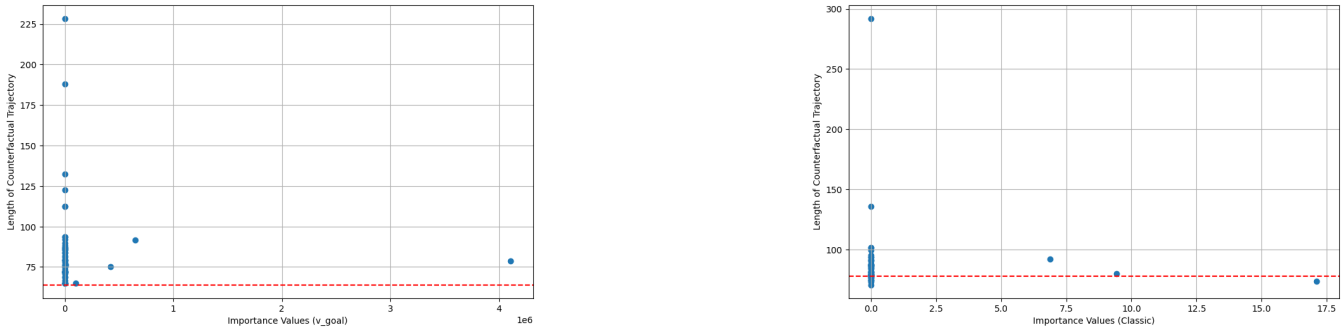
To evaluate an entire trajectory, we aggregate the importance scores of its constituent state-action pairs. For a trajectory $\tau = \{(s_0,a_0),(s_1,a_1),\ldots,(s_T,a_T)\}$, its importance is the average state-action importance:

$$I_\tau = \frac{1}{|\tau|}\sum_{(s,a)\in\tau} I(s,a) = \frac{1}{|\tau|}\sum_{(s,a)\in\tau} \Delta Q(s) \times R(s,a)$$

This score allows us to rank a large collection of trajectories, identifying those that are most representative of the agent's optimal strategy.

Explanation Pipeline

Our full pipeline for generating trajectory explanations is as follows:



(a) Our Method (V-Goal)

(b) Classic Method (ΔQ)

Figure 2: Acrobot counterfactual trajectory lengths. The red line is the original trajectory's length. (a) For our method, all counterfactuals are longer (worse) than the original. (b) For the classic method, some counterfactuals are shorter (better), indicating it did not select a truly optimal trajectory to explain.

1. **Data Collection**: Collect a dataset of trajectories and populate a Q-table from a trained agent's critic. For continuous state spaces, we discretize the state representations.
2. **Importance Calculation**: For each state-action pair (s, a) in every trajectory, calculate our modified importance metric $I(s, a)$.
3. **Trajectory Ranking**: Compute the aggregate importance I_τ for each trajectory and rank them to find the top-k most important ones. We select the trajectory from this set with the best outcome (e.g., highest reward, shortest length).
4. **Counterfactual Generation**: For the top-ranked trajectory, generate counterfactuals. At each state s_i along the original path, we forbid the original action a_i and force the agent to take a different action, after which it follows its policy π. This produces a set of alternative trajectories.
5. **Contrastive Explanation**: Compare the original trajectory with the generated counterfactuals on metrics like total reward, length, and importance score. An optimal original trajectory should be demonstrably better than its counterfactuals, providing a powerful explanation for the agent's behavior.

This pipeline provides a concrete method for answering "Why this path and not another?" by showing the consequences of deviation.

Experiments and Results

We conducted experiments in OpenAI Gym environments (Brockman et al. 2016), Acrobot-v1 and LunarLander-v2, using agents trained with the PPO algorithm. We focused on the scenario where trajectories are collected throughout the training process, resulting in a dataset containing both optimal and suboptimal behaviors. Our framework must be able to distinguish between them.

Identifying Optimal Trajectories

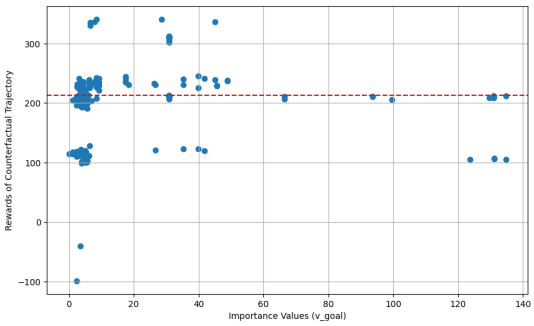
We first evaluated the ability of different radical terms ($R(s, a)$) to identify the best trajectories. For each metric, we ranked all collected trajectories and computed the average length and reward of the top 5.

Method	**Avg. Length**	**Avg. Reward**
Acrobot-v1 Environment		
Classic (ΔQ only)	70.0	-69.0
Naive Normalization	70.0	-69.0
Entropy-Based	73.2	-72.2
Bellman Error	70.8	-69.8
V-Normalization	70.0	-69.0
V-Goal	**68.8**	**-67.8**

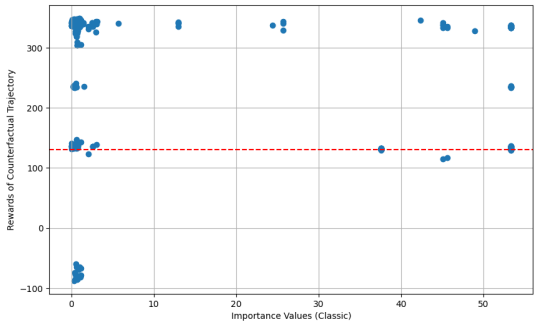
Table 1: Performance of top-5 ranked trajectories in Acrobot. Lower length and higher (less negative) reward are better. Our **V-Goal** metric identifies the most optimal set of trajectories.

The results for the Acrobot environment are shown in Table 1. In this task, success is measured by achieving the goal in the fewest steps, resulting in a higher (less negative) reward. While the differences are subtle, the trajectories ranked highest by our 'V-Goal' metric are consistently the most efficient, with the shortest average length (68.8) and highest average reward (-67.8). This provides initial evidence that incorporating goal affinity helps refine the selection of optimal trajectories. The distinction becomes much clearer in the more complex LunarLander environment.

The data in Table 2 clearly shows the superiority of the V-Goal' metric. In LunarLander, a successful landing yields a high reward, while crashing or running out of time (max 1000 steps) results in poor scores. Our 'V-Goal' metric is the only method that consistently identifies successful landing trajectories, achieving an average reward over 200 and an average trajectory length of 319 steps. In contrast, the classic method and others select trajectories that hit the time limit, indicating failed or meandering attempts. This suggests that incorporating goal proximity via the value function is essen-



(a) Our Method (V-Goal)



(b) Classic Method (ΔQ)

Figure 3: LunarLander counterfactual trajectory rewards. The red line represents the original trajectory's reward. (a) For our method, all counterfactuals yield lower rewards. (b) For the classic method, some counterfactuals result in higher rewards.

Method	**Avg. Reward**	**Avg. Length**
LunarLander-v2 Environment		
Classic (ΔQ only)	116.87	1000.0
Bellman Error	117.37	1000.0
Naive Normalization	188.12	433.2
Entropy-Based	121.27	871.0
V-Normalization	120.59	1000.0
V-Goal	**207.13**	**319.2**

Table 2: Performance of top-5 ranked trajectories in LunarLander. Higher reward and lower length are better. The results starkly highlight the effectiveness of our proposed metric.

tial for distinguishing truly optimal, task-achieving behavior from prolonged, suboptimal attempts.

Counterfactual Explanations

Having established 'V-Goal' as our best metric, we generated counterfactuals for the single best trajectory it identified and compared them with those from the top trajectory identified by the classic ΔQ metric. A successful explanation would show that deviations from the original trajectory lead to worse outcomes (e.g., longer paths, lower rewards).

Figure 2 shows the results for Acrobot. For the trajectory selected by our 'V-Goal' metric (Fig. 2a), every generated counterfactual trajectory was longer (worse) than the original. This provides a strong, clear explanation: the agent followed the optimal path, and any deviation would have been suboptimal. In contrast, for the trajectory selected by the classic ΔQ metric (Fig. 2b), several counterfactuals were shorter than the original, indicating that the classic method failed to identify a truly optimal trajectory, thus providing a confusing or incorrect explanation.

We observed the same pattern for LunarLander, shown in Figure 3. Counterfactuals from the trajectory identified by our method consistently resulted in lower rewards, while the classic method again selected a trajectory from which better paths could be found. These results robustly demonstrate that our modified importance metric is superior for identifying and explaining optimal agent behavior.

Discussion

Our experiments demonstrate that when an agent is still learning, with an experience buffer containing both successful and suboptimal trajectories, our framework excels at identifying and explaining optimal behavior.

However, when analyzing trajectories generated exclusively by a fully trained, optimal agent, the task becomes more challenging. In this scenario, most trajectories are nearly identical in quality, offering less explanatory insight. Future work could address this by focusing not on ranking trajectories but on identifying the few most **critical states** within a single optimal trajectory. Generating counterfactuals from only these pivotal moments could provide more concise and impactful explanations, even for highly optimized agents, maintaining our focus on trajectory-level analysis while adapting to the nuances of expert behavior.

Conclusion and Future Work

In this paper, we introduced a framework for generating trajectory-level explanations in Reinforcement Learning. By defining a state-importance metric that accounts for both Q-value advantage and goal affinity, our method identifies and ranks optimal trajectories from heterogeneous experience data. Through contrastive counterfactuals, we provide clear, intuitive explanations for an agent's long-term strategy, demonstrating why its chosen path was superior to alternatives.

Our empirical results show that this approach is more effective than classic importance metrics, providing a more reliable foundation for trustworthy and deployable AI systems. Understanding and interrogating high-level behavior is a critical step towards deploying RL safely in the real world.

For future work, we plan to extend this framework to automatically identify critical decision points within trajectories and explore scenarios where the agent's policy and value functions are unknown. Techniques from Inverse Reinforcement Learning (IRL) could infer a reward function explaining observed trajectories, after which our importance-based analysis can explain the inferred policy.

References

Amir, D.; and Amir, O. 2018. HIGHLIGHTS: Summarizing Agent Behavior to People. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, 1168–1176. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Amir, O.; Kamar, E.; Kolobov, A.; and Grosz, B. J. 2016. Interactive teaching strategies for agent training. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, 804–811. AAAI Press. ISBN 9781577357704.

Amitai, Y.; Septon, Y.; and Amir, O. 2024. Explaining reinforcement learning agents through counterfactual action outcomes. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press. ISBN 978-1-57735-887-9.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. arXiv:1606.01540.

Danesh, M. H.; Koul, A.; Fern, A.; and Khorram, S. 2021. Re-understanding Finite-State Representations of Recurrent Policy Networks. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2388–2397. PMLR.

Deshmukh, S. V.; Dasgupta, A.; Krishnamurthy, B.; Jiang, N.; Agarwal, C.; Theocharous, G.; and Subramanian, J. 2024. Explaining RL Decisions with Trajectories. arXiv:2305.04073.

Frost, J.; Watkins, O.; Weiner, E.; Abbeel, P.; Darrell, T.; Plummer, B.; and Saenko, K. 2022. Explaining Reinforcement Learning Policies through Counterfactual Trajectories. arXiv:2201.12462.

McCalmon, J.; Le, T.; Alqahtani, S.; and Lee, D. 2022. CAPS: Comprehensible Abstract Policy Summaries for Explaining Reinforcement Learning Agents. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022*, Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, 889–897. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS). Publisher Copyright: © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved; 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022 ; Conference date: 09-05-2022 Through 13-05-2022.

Milani, S.; Topin, N.; Veloso, M.; and Fang, F. 2024. Explainable Reinforcement Learning: A Survey and Comparative Review. *ACM Comput. Surv.*, 56(7).

Takagi, Y.; Tabalba, R.; Kirshenbaum, N.; and Leigh, J. 2024. Abstracted Trajectory Visualization for Explainability in Reinforcement Learning. arXiv:2402.07928.

Appendix

Supplementary Counterfactual Results

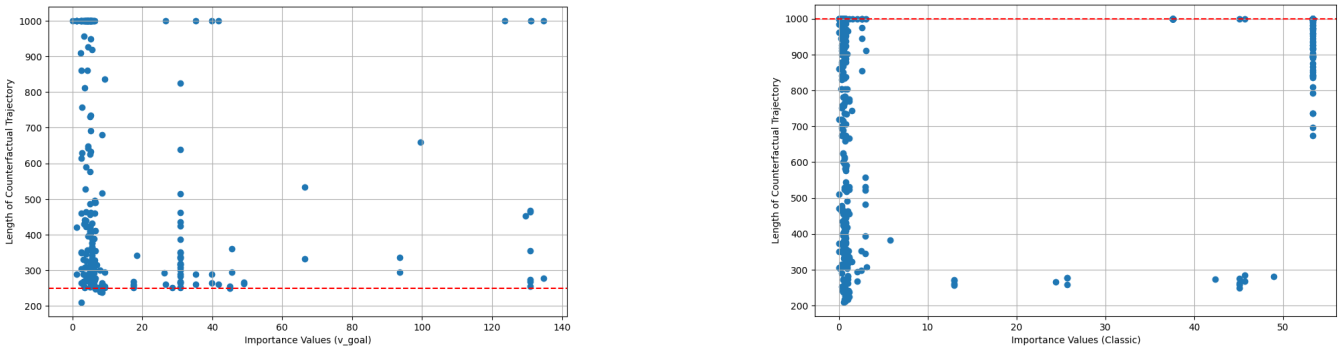
As discussed in the main paper's counterfactual analysis, our 'V-Goal' metric consistently identifies trajectories that are superior to their alternatives. The main text demonstrated this using reward values and trajectory length for LunarLander (Figure 2). This appendix provides supplementary evidence using trajectory length as the metric (Figure 4), further reinforcing our claim that the identified optimal trajectory is robustly better than its counterfactuals across multiple performance dimensions.

Exploration of KL-Divergence Metric

During our research, we explored using KL-divergence as a radical term, $r(s) = KL(\pi(s), X(s))$, where $\pi(s)$ is the agent's policy and $X(s)$ is a reference distribution over actions. The intuition was that a high divergence would indicate a confident, non-uniform policy. However, this metric was ultimately not included in our final framework for two main reasons:

1. **High Variance**: The performance was highly sensitive to the choice of the reference distribution $X(s)$ and varied significantly across different environments.
2. **Lack of Clear Rationale**: It was difficult to establish a principled, general method for choosing the reference distribution $X(s)$ for any given environment. While we hypothesized certain distributions might be suitable for specific agent behaviors (as shown in Table 3), this could not be consistently validated.

Due to this lack of stability and clear justification, we concluded that the KL-divergence metric was not robust enough for a general-purpose explanation framework.



(a) Our Method (V-Goal)

(b) Classic Method (ΔQ)

Figure 4: LunarLander counterfactual trajectory lengths. The red line represents the original trajectory's length. (a) Counterfactuals from our method's selected trajectory are probabilistically longer. (b) The classic method's selected trajectory has counterfactuals that are probabilistically shorter.

Distribution	When to Use
Uniform	When no clear preference exists in action selection; suitable for exploratory, early-stage agents or when the action tendencies are uncertain.
Gaussian	When actions follow a central tendency with some variability (common in continuous action spaces). Ideal for confident, near-deterministic agents.
Exponential	When large actions are rare and small actions are frequent (e.g., sparse high-reward events). Suitable for exploitative agents.
Dirichlet	When some actions are preferred over others but there remains significant variability. Useful for environments with multiple viable paths to success.
Beta	When actions have bounded probabilities (0–1) and model uncertainty in preference; suitable for tasks balancing exploration and exploitation.

Table 3: Initial hypotheses for choosing a reference distribution $X(s)$ for the KL-divergence metric.