Perceptual Similarity for Measuring Decision-Making Style and Policy Diversity in Games

Anonymous authors
Paper under double-blind review

Abstract

Defining and measuring decision-making styles, also known as playstyles, are crucial in gaming, where they reflect a broad spectrum of individuality and diversity. However, finding a universally applicable measure for these styles poses a challenge. Building on the *Playstyle Distance*, the first unsupervised metric to measure playstyle similarity based on observation-action pairs by identifying comparable states with discrete representations, we introduce three enhancements to increase accuracy: multiscale analysis with varied state granularity, a perceptual kernel rooted in psychology, and the utilization of the intersection-over-union method for efficient evaluation. These innovations not only advance measurement precision but also offer insights into human cognition of similarity. Across two racing games and seven Atari games, our techniques significantly improve the precision of playstyle classification, achieving an accuracy exceeding 90% with fewer than 512 observation-action pairs, less than half an episode of these games. Furthermore, our experiments with *2048* and *Go* demonstrate the potential of discrete playstyle measures in puzzle and board games. We also develop an algorithm for assessing decision-making diversity using these measures. Our findings illuminate promising avenues for real-time game analysis and the evolution of artificial intelligence with diverse playstyles.

1 Introduction

The pursuit of diversity in decision-making is one of the intrinsic motivations that drive human behavior, resulting in individuality and creativity (Rheinberg, 2020). This is evident in the context of games, where decision-making manifests as various playstyles, each reflecting individual strategies (Bean & Groth-Marnat, 2016). Alongside the pursuit of diversity, another central aspect of decision-making focuses on achieving optimal performance. Significant advances in decision-making performance have been witnessed, especially with the development of Deep Reinforcement Learning (DRL) (Russell & Norvig, 2020). The effectiveness of DRL was first showcased in arcade video games (Mnih et al., 2015). Subsequent applications to board games emphasized its potential, achieving superhuman skills (Silver et al., 2018). This success expanded into various types of game, from Agent57's superhuman performance in Atari games to groundbreaking feats in Dota 2 and StarCraft II (Badia et al., 2020; Berner et al., 2019; Vinyals et al., 2019). Beyond gaming, DRL applications extend to robot control (Andrychowicz et al., 2020) and natural language processing (Ouyang et al., 2022), among others.

Yet, while DRL continues to show promise in diverse applications, understanding and analyzing playstyles remain a complex endeavor. Data from diverse sources are essential to improve agent strength or efficiency (Fan & Xiao, 2022), just as acquiring skills from different styles is vital for agents to generalize across tasks (Eysenbach et al., 2019). Although a robust playstyle measure fosters a spectrum of playing strategies, it also reveals the inherent challenges in measuring these styles, particularly in environments without built-in features for style measurement. Consequently, achieving precise playstyle measurement remains a formidable task (Tychsen & Canossa, 2008).

There are several methods to evaluate playstyles, from heuristic rules design to in-game features exploration (Tychsen & Canossa, 2008; Bontchev & Georgieva, 2018; Mader & Tassin, 2022). Supervised learning

facilitates the discrimination of playstyles (Brombacher et al., 2017), while unsupervised clustering offers behavioral insights (Ferguson et al., 2020). Another avenue involves contrastive learning to identify playstyles among players (McIlroy-Young et al., 2021; Agarwal et al., 2021). Through these methods, the notion of playstyle can be gauged using distance or similarity measures across game datasets, addressing dynamic and evolving challenges in different scenarios. The concept is reflected in the work by Lupu et al. (2021), which specifies the policy diversity using the divergence of action distributions but requires parametrized policies.

The recent innovation by Lin et al. (2021) introduces the *Playstyle Distance* measure, which stands out by directly measuring playstyle from in-game observation-action pairs. Unlike common methods that compare latent feature distributions or rely on parametrized policies, this approach measures action distribution distances directly from gameplay samples, reducing the reliance on predefined style labels or extensive training sets for learning latent features or policies. Its effectiveness hinges on the critical role of state discretization. By discretizing observations, we can identify comparable states, allowing for a direct comparison of action distributions of each state, thereby defining the decision-making style on the expected distance of policies.

While the *Playstyle Distance* offers a pivotal advance, our research endeavors to elevate this foundation. We introduce techniques to improve playstyle measurement, harnessing the advantage of discrete states. Initially, we leverage multiscale analysis with varied state granularity, emulating human judgment of similarity from multiple attributes and viewpoints (Medin et al., 1993). We then derive a perceptual kernel from psychophysics (Fechner, 1966) in psychology to obtain a probabilistic similarity value, which is more in harmony with human comprehension than distance values. Moreover, incorporating the concept of the *Jaccard index* (Murphy, 1996), we broaden the focus of measurement, harnessing all observed game data to improve measurement accuracy. These techniques not only improve precision in playstyle measurement, but also shed light on understanding similarity through the lens of human cognition. From their fusion emerges the *Playstyle Similarity* measure.

To underscore our contributions, we propose three improvements for playstyle measurement. With these improvements, we achieve over 90% accuracy in playstyle classification tasks with fewer than 512 observation-action pairs, which is less than half an episode in the examined games including two racing games and seven Atari games. Furthermore, our experiments with *2048* and *Go* demonstrate the potential of discrete playstyle measures in puzzle and board games. Additionally, we introduce an algorithm to measure the diversity of decision-making, which showcases the applicability of our measures beyond zero-shot classification tasks. This algorithm is based on comparing the similarity of a new trajectory to previous trajectories using a probabilistic similarity threshold. These explorations enhance our understanding of game analysis and AI training, harmoniously merging playstyle measurement with human-centric similarity.

2 Background and Related Works

In this section, we discuss playstyle in depth, provide a historical overview, and highlight the importance of discrete representation in the creation of general playstyle measures.

2.1 Playstyle and Measurement

Establishing a universally accepted playstyle measure is a formidable challenge, as perceptions of playstyle are influenced by myriad factors and often harbor subjective nuances. Consequently, any playstyle measure should specify its evaluative parameters transparently to ensure that its measurements are persuasive. Historically, tailored metrics, characterized by heuristic rules or specific in-game features, often presented the most precision for dedicated case studies. For example, the study by Lample & Chaplot (2017) used measures such as object counts, kills, and deaths in shooting games. However, due to their inherent manual nature, these measures are often domain-specific.

To achieve a wider applicability, some researchers have resorted to supervised learning to identify styles (Brombacher et al., 2017). However, this method requires labeled training data and may encounter difficulties in detecting styles not present in the training set. Unsupervised clustering offers a different angle, emphasizing latent feature distances for classification (Ferguson et al., 2020). But this approach may obscure the semantic meaning of the measures, particularly when image data is the primary source. A notable approach is the

Behavioral Stylometry proposed by McIlroy-Young et al. (2021). This measure, crafted for chess playstyle assessment, encodes chess moves into a game vector, aggregates these vectors to represent a style, and then compares this representation against a reference set. Central to this method is the contrastive learning technique *Generalized End-to-End* (Wan et al., 2018), used to learn latent features to identify the most similar player in the given datasets.

For a more generalized measurement of playstyle, one could consider measuring the similarity of policies. It is common to regularize the policy with a prior model (Schulman et al., 2015; Garg et al., 2023). Methods that extend to specify similarity or diversity by comparing the action distribution of two policies have also been explored (Agarwal et al., 2021; Lupu et al., 2021). Notably, these methods often require a parametrized policy for comparisons. This limitation is addressed by the *Playstyle Distance* measure (Lin et al., 2021). Instead of emphasizing latent features or parametrized policies, *Playstyle Distance* focuses on the action distributions of given samples. Observations are discretized and then used for determining which action samples are comparable. Such a method resonates more with human instinct, echoing the case-by-case assessment we often deploy.

2.2 Framework of Playstyle Distance

To delve deeper into the generality and importance of the *Playstyle Distance* in playstyle measurement, we examine its foundation as follows. A pivotal component of its methodology is the use of the Vector Quantized-Variational AutoEncoder (VQ-VAE), which specializes in discrete representations by mapping continuous encodings to the nearest vectors in a predefined codebook (van den Oord et al., 2017). Building upon VQ-VAE, the Hierarchical State Discretization (HSD) in *Playstyle Distance* ensures a concise and hierarchical state space. This is essential for identifying overlapping discrete states while preserving the feature integrity of observation reconstruction and gameplay details.

Central to this framework is the discrete state encoder, denoted as ϕ. Observations o and their associated actions a are mapped to datasets $M_i \sim Style_i$. The encoder ϕ translates these observations into a compact state representation s, formulated as:

$$S_\phi : \phi(o) \to s$$

In the initial *Playstyle Distance* approach, the hierarchical encoder ϕ has the capability to generate multiscale discrete states. However, the foundational literature employs only a singular state space for computations; the hierarchical structure is used to control the state space and maintain the quality of discrete representation learning. Using a state space that is too large or too small a state space may lead to unstable measurements.

From state s, action distributions are deduced using a sampling distribution:

$$\{a|(o,a) \in M, \phi(o) \to s\} \Rightarrow \pi_M(s)$$

Here, π represents the policy, depicting action distributions for a given state. Subsequently, the distances between these distributions are determined using the metric $D(\pi_X, \pi_Y)$, where the 2-Wasserstein distance (W_2) serves as the standard (Vaserstein, 1969). Recognized for measuring the 'effort' to transform one distribution into another, the Wasserstein distance is apt for policy comparisons, analogous to quantifying the 'effort' to transition between playstyles.

The essence of this measure can be succinctly captured as:

$$\begin{aligned} &d_\phi(M_A, M_B) = \frac{1}{2} d_\phi(M_A|M_B) + \frac{1}{2} d_\phi(M_B|M_A), \\ &\text{where } d_\phi(M_X|M_Y) = \mathbb{E}_{o \sim M_Y, \phi(o) \in \phi(M_X) \cap \phi(M_Y)}[D(\pi_X(\phi(o)), \pi_Y(\phi(o)))] \end{aligned} \tag{1}$$

To summarize, the *Playstyle Distance* framework presents a method to contrast playstyles, reducing the need for predefined heuristics and datasets, thereby increasing the applicability in various games. For a graphical representation of the framework, see Figure 1.

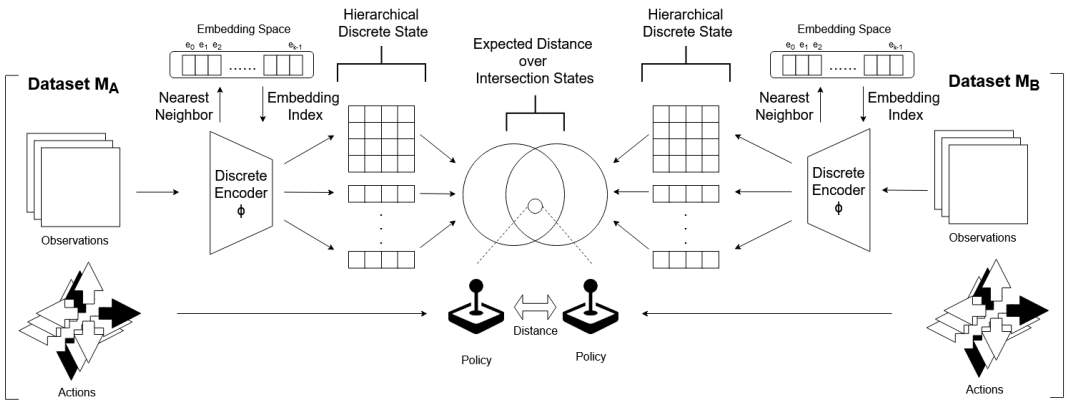


Figure 1: Illustration of the *Playstyle Distance* computation using a hierarchical discrete state encoder ϕ. The Venn diagram highlights the intersection of discrete states for distance calculation.

3 Discrete Playstyle Measures

In this section, we delve into a series of discrete playstyle measures derived from *Playstyle Distance*. We first discuss the limitations of *Playstyle Distance*. We then expand it into a multiscale approach by leveraging the hierarchical structure of states. Subsequently, we explore converting the action distribution distance into a perceptual similarity rooted in cognitive psychology, utilizing a perceptual kernel function. Thirdly, we broaden our scope from merely intersection states to the union of states, aiming for a more efficient estimation of all observed data. Concluding the section, we integrate these improvements into a comprehensive measure we term as *Playstyle Similarity*.

3.1 Effect of Multiscale States

Playstyle Distance hinges on discrete states for performing style measurements. Consequently, it resorts to a constrained state space sourced from the HSD model. A sample count threshold is applied to the intersection state to ensure the quality of action distribution, necessitating at least t samples in both datasets under comparison; failing which, the state is excluded from the intersection. This filtering sometimes discards important information; thus, we propose using multiscale analysis to alleviate this problem.

Human cognition perceives similarity as a convergence of multiple attributes leading to a holistic understanding (Goldstone & Barsalou, 1998). Hence, we advocate for employing varied granularity of discrete states to augment measurement capabilities, analogous to human judgment that varies from a broad view to intricate details. The HSD model's design inherently possesses a large state space for observational reconstruction and the discernment of gameplay nuances (Lin et al., 2021). Though the state space may be large, intersections are not void if observations are sufficiently similar or come from identical gameplay. It is worth noting that Lin et al. (2021) were able to identify intersection states even with unprocessed screen pixels in Atari games. In a different scenario, when treating each state as the same, we can invariably find the single intersection state. In this context, distance simply gauges the action distribution over the entirety of the game, akin to traditional methods deploying post-game action statistics.

Broadly speaking, we can enhance the original state encoder function, ϕ, evolving it into a state encoder mapping, Φ, wherein Φ is an assemblage of mapping functions, $\phi \in \Phi$, $S_\Phi : \Phi(o) \rightarrow \{\phi(o) \in S_\phi | \phi \in \Phi\}$. Consequently, the projected state of dataset M is defined as: $\Phi(M) \rightarrow \{\phi(o) \in S_\phi | o \in M, \phi \in \Phi\}$. We can then reinterpret Equation 1 as:

$$d_\Phi(M_A, M_B) = \frac{d_\Phi(M_A|M_B)}{2} + \frac{d_\Phi(M_B|M_A)}{2}, \text{where } d_\Phi(M_X|M_Y) = \frac{1}{|\Phi|} \sum_{\phi \in \Phi} d_\phi(M_X|M_Y) \tag{2}$$

This reformulated measure demonstrates superior accuracy in playstyle classification tasks in our experiments, even negating the need for a sample threshold count t. This improvement is likely due to the integration of hierarchical discrete states of varying granularity, which dilutes the impact of outliers during distance computation and leverages more useful details. Furthermore, the adoption of a multiscale state space effectively mitigates the trade-off between a compact intersection space and the preservation of intricate information details. This balance becomes crucial in complex games or those that require a vast state space to encode trajectory data.

3.2 Perception of Similarity

One potential shortcoming of the *Playstyle Distance* stems from the nature of distance itself. While distance is a common measure for determining similarity, a larger distance value conveys primarily that two entities are different, without giving much insight into the degree of their similarity. For example, a given point in 2D space, the candidate points with the same distance to the given point form a circle. As the distance increases, the size of this candidate circle also increases, and the similarity information is diluted.

Human cognition has been observed to exhibit the *Magnitude Effect*, suggesting diminished sensitivity to larger stimuli (Kahneman & Tversky, 1979). This is in line with the *Weber–Fechner Law* in psychophysics, where the relationship between stimulus and perception is logarithmic; as the magnitude of stimuli increases, sensitivity diminishes (Fechner, 1966). Drawing from the concept of similarity, we can infer that a smaller distance provides more definitive information about the similarity. As distance grows, the distinction becomes vaguer. Therefore, we argue for a measure that reflects higher sensitivity to smaller distances, emulating human perceptual behaviors.

We propose a probability-based model for similarity. In this model, greater similarity (i.e., smaller distance) corresponds to a probability closer to 100%, while lesser similarity (larger distance) approaches 0%. This proposed probability function aligns with the logarithmic human perceptual sensitivity to differences. Specifically, we use the exponential kernel to describe the probability of similarity, with the mapping function given by $P(d) = \frac{1}{e^d}$, where d is the distance value. This choice is analogous to the radial basis function kernel (Vert et al., 2004). This perceptual kernel is the if and only if kernel under our assumptions from human cognition and probability. We provide a proof using differential equations in Appendix A.1.

This exponential transformation can also be found in the Bhattacharyya distance and coefficient (Bhattacharyya, 1946). The Bhattacharyya coefficient $BC(P, Q)$ measures the similarity between two probability distributions P and Q, and it is related to the overlapping region between these two distributions. It is defined as $BC(P, Q) = \int_{\mathcal{X}} \sqrt{P(x)Q(x)}dx$. The Bhattacharyya distance, derived from the coefficient, is $D_B(P, Q) = -ln(BC(P, Q))$, and the inversion is $BC(P, Q) = exp(-D_B(P, Q))$.

Thus, we define a new playstyle measure $PS^{\cap}_{\Phi}(M_A, M_B)$ with probability of similarity in Equation 3:

$$PS^{\cap}_{\Phi}(M_A, M_B) = \frac{\sum_{s \in \bigcup_{\phi \in \Phi} \phi(M_A) \cap \phi(M_B)} P(D^M_{\Phi}(\pi_{M_A}(s), \pi_{M_B}(s)))}{|\bigcup_{\phi \in \Phi} \phi(M_A) \cap \phi(M_B)|}, \text{where } D^M_{\Phi}(\pi_X, \pi_Y) = \frac{D(\pi_X, \pi_Y)}{\overline{D}^M_{\Phi}} \tag{3}$$

The measure has been simplified by adopting a uniform average distance instead of an expected value. This not only streamlines calculations but also underscores the significance of encoder granularity. In particular, an intricate encoder with a vast state space may be accorded greater weight, especially if the intricate encoder reveals more intersection states. To match our probabilistic framework (Appendix A.1) we rescale the distances with a constant, $\overline{D}^M_{\Phi}$, ensuring the expected distance converges to 1. The constant $\overline{D}^M_{\Phi}$ can be calculated by averaging all observed distance on each discrete state in comparisons. Collectively, our revamped measure provides a probabilistic lens to interpret similarity, firmly rooted in cognitive theory and tailored for human comprehension. There is more discussion about the role of the distance metric in Appendix A.2.1, including the implications of adopting the Bhattacharyya distance metric.

3.3 Beyond Intersection

Before presenting our final measure, it is pertinent to revisit the foundational concept of the *Playstyle Distance*: the intersection of states. Identifying comparable states before measuring policy similarities encounters challenges when the intersecting samples are limited. A smaller intersection proportion can result in unstable or insufficient samples for measuring playstyles. Such a small intersection could indicate two scenarios. First, distinct state-visiting distributions might signify different playstyles. In contrast, uncontrollable factors external to playstyles, such as environmental randomness or decisions from other players, may also play a role, indicating the necessity for more extensive sampling.

A prudent approach would assess the proportion of intersecting samples relative to the total observed samples. In the realm of collection comparison, the *Jaccard index* (Murphy, 1996), also known as *Intersection over Union*, emerges as a prevalent similarity measure. The *Jaccard index* can serve as an effective playstyle measure under specific conditions. It is particularly apt when game observations clearly delineate playstyle distinctions. For instance, in deterministic environments where states can be distinctly segmented by different actions, the *Jaccard index* appears to be a fitting measure. However, complications arise when certain states recur due to game rules or every state is visited. The task of distinguishing different playstyles based solely on observations becomes considerably challenging. This is evident in single-state games, such as K-arm bandits (Sutton & Barto, 2018), where measuring playstyles only from states becomes an impractical endeavor.

Despite potential challenges, our empirical findings suggest that the *Jaccard index* serves as a handy measure, when the state space is large and the randomness in the game is low. The incorporation of the *Jaccard index* into a playstyle measure with a multiscale state space is expressed in Equation 4:

$$J_\Phi(M_A, M_B) = \frac{|\bigcup_{\phi\in\Phi} \phi(M_A) \cap \phi(M_B)|}{|\bigcup_{\phi\in\Phi} \phi(M_A) \cup \phi(M_B)|} \tag{4}$$

3.4 Playstyle Similarity

Throughout our exploration, we have derived and discussed various discrete playstyle measures. Collating these insights, we introduce a comprehensive measure termed as the *Playstyle Similarity*. Defined as $PS^{\cup}_{\Phi}(M_A, M_B)$, it synthesizes our earlier discussions into a singular measure as illustrated below:

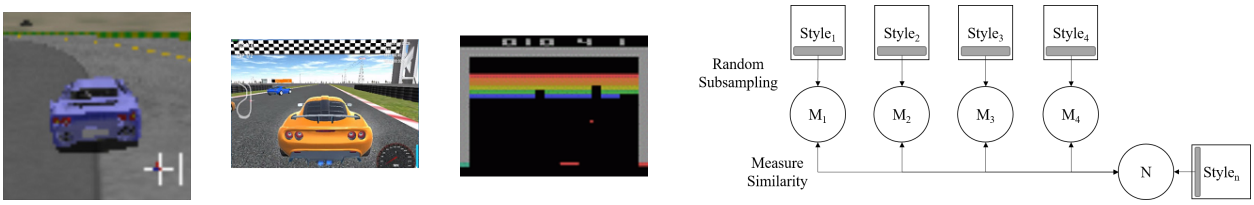
$$\begin{aligned} PS^{\cup}_{\Phi}(M_A, M_B) &= J_\Phi(M_A, M_B) \times PS^{\cap}_{\Phi}(M_A, M_B) \\ &= \frac{\sum_{s\in\bigcup_{\phi\in\Phi} \phi(M_A)\cap\phi(M_B)} P(D^M_\Phi(\pi_{M_A}(s), \pi_{M_B}(s)))}{|\bigcup_{\phi\in\Phi} \phi(M_A) \cup \phi(M_B)|} \end{aligned} \tag{5}$$

What makes this measure novel is its unique treatment of intersection states. While the *Jaccard index* assigns a uniform weight (of 1) to each intersecting state regardless of the similarity between the action distributions, our approach infuses a more nuanced probability-based weighting. The values range between 0 and 1, increasing proportionally with similarity. This modification alleviates the potential limitation of using the *Jaccard index* for playstyle measurements.

Furthermore, our approach ensures a consistent interpretation of zero values. For states not part of the intersection, where the distance between action distributions is maximal (approaching infinity), they can be understood as totally dissimilar.

4 Experiment Settings

In this section, we explain the specifics of our experiment setup, focusing on the datasets, sources of encoder models, and our playstyle classification methodology.



(a) TORCS (b) RGSK (c) Atari (d) Playstyle classification task

Figure 2: Three game platforms and the illustration of zero-shot playstyle classification tasks.

4.1 Game Platforms, Datasets, and Model Source

Our study encompasses three distinct game platforms, as depicted in Figures 2a, 2b, and 2c:

1. **TORCS**: This racing game features stable, controlled rule-based AI players (Yoshida et al., 2017). The datasets derived from TORCS include a total of 25 playstyles based on driving speed and noise characteristics. Each observation consists of a sequence of 4 consecutive RGB images with a size of 64×64. The action space is 2-dimensional and continuous.
2. **RGSK - Racing Game Starter Kit**: This racing game, available on the Unity Asset Store (Juliani et al., 2020), showcases human players. From RGSK, we have data from a total of 24 players, exhibiting individual playstyles. Each observation from this game comprises 4 consecutive RGB images of size 72×128, with 27 discrete actions.
3. **Atari games with DRL agents**: The dataset spans 7 different Atari games (Bellemare et al., 2013) from this platform. Each game includes 20 AI models, all of which demonstrate varied playstyles. These AI models originate from the DRL framework, *Dopamine* (Castro et al., 2018). Each observation involves 4 consecutive grayscale images of resolution 84×84. The action space is discrete and varies depending on the game.

It is crucial to clarify that our research did not involve the training of new encoder models. Instead, we leveraged three pretrained encoder models and corresponding datasets for each game, provided by Lin et al. (2021). The code and associated resources are available in their official release.[1]

4.2 Playstyle Classification and State Space Levels

Our playstyle classification adheres to the zero-shot methodology. As depicted in Figure 2d, we start with a query dataset N, sampled from a playstyle $Style_n$. We then compare this to multiple reference datasets M, each sampled from different playstyles $Style$. We perform 100 rounds of random subsampling for each playstyle; our primary performance metric for this task is the accuracy of playstyle classification. If dataset N exhibits the highest similarity to a reference dataset M_i, it suggests that $Style_n = Style_i$. The reported accuracy represents an average, derived from results obtained using the three discrete encoder models.

Regarding the discrete state space levels, three tiers have been considered:

1. Space size 1, a basic mapping with state space 1, which maps all observations identically.
2. Space size 2^{20}, as suggested by *Playstyle Distance*.
3. Space size $256^{64\sim144}$ or 256res, a level trained by HSD for the base hierarchy, depending on the resolution **res** of convolution features from the game screens.

5 Results and Discussion

In this section, we assess the efficacy of our methods. Initially, we demonstrate how a multiscale state space can enhance the accuracy of *Playstyle Distance* without relying on a sample count threshold to preserve the

[1] `https://paperswithcode.com/paper/an-unsupervised-video-game-playstyle-metric`

Table 1: Playstyle accuracy (%) when employing various discrete state spaces in the multiscale version of *Playstyle Distance*, with a sample threshold count t for intersecting states. Conventional methods, such as supervised learning (**SL**) and contrastive learning (**CL**), are not suited for cases lacking playstyle labels in the training datasets. In contrast, discrete playstyle measures do not suffer from this limitation, as they do not require playstyle labels for training the discrete encoders.

	SL	CL	1	2^{20}, t=2	2^{20}, t=1	256res, t=2	256res, t=1	**mix**, t=2	**mix**, t=1
TORCS	4.00	4.00	35.16	70.68	60.83	4.60	60.76	74.77	**75.05**
RGSK	4.17	4.17	80.14	78.39	93.43	5.62	26.58	88.51	**94.21**

quality of distance measurement. Subsequently, we compare several baselines, illustrating that probabilistic values for measuring similarity offer a viable alternative to distance values. Lastly, we incorporate all observed data to evaluate measures across all platforms. We also discuss the continuous playstyle spectrum in Appendix A.4, demonstrating the consistency of measure values under slightly different behaviors.

5.1 Multiscale State Space Efficacy

To evaluate the proposed multiscale state space and compare it with *Playstyle Distance*, we utilize the TORCS and RGSK platforms. Each dataset sampled from the given playstyles comprises 1024 observation-action pairs. Additionally, we examine the intersecting states' sample threshold count t, where an intersecting state requires at least t samples in both compared datasets for a stable action distribution estimation.

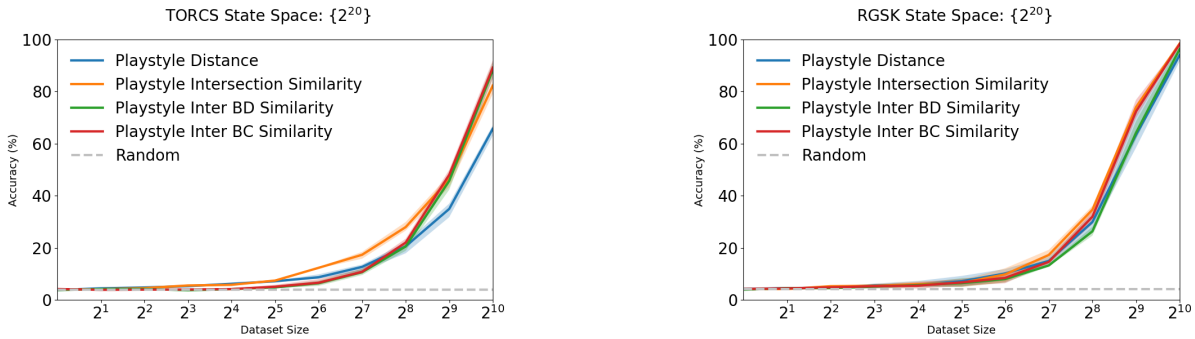
In our comparisons, we initially show that conventional methods, such as supervised learning (**SL**) and contrastive learning (**CL**), fall short for zero-shot playstyle classification due to the lack of playstyle labels or groups in the training datasets. We then focus our comparisons on *Playstyle Distance*, as detailed in Equation 2. For the multiscale version, which incorporates three discrete state spaces—$\{1, 2^{20}, 256^{\text{res}}\}$—we simplify our terminology by using the label "mix." The results, as shown in Table 1, clearly indicate that using a multiscale discrete state space not only yields superior results but also obviates the need for a sample threshold count t for intersecting states.

5.2 Probability vs. Distance

In addition to introducing the multiscale discrete state space, another key contribution of our work is the proposal to use probabilistic similarity from a perceptual perspective rather than employing negative distance as a measure of similarity. To elucidate the benefits of this modification, we examine the relationship between accuracy and dataset size of the sampled observation-action pairs. These pairs are evaluated under a single discrete state space $\{2^{20}\}$, without employing a sample count threshold, to provide a clear assessment of the transformation from distance to similarity. Further comparisons with different discrete state spaces can be found in Appendix A.2.

We evaluate several measures in this comparison:

- *Playstyle Distance*: $-d_\Phi$
- *Playstyle Intersection Similarity*: $PS_\Phi^\cap$
- *Playstyle Inter BD Similarity*: $PS_\Phi^{\cap BD}$, a variant of $PS_\Phi^\cap$ that employs the Bhattacharyya distance in place of the 2-Wasserstein distance
- *Playstyle Inter BC Similarity*: $PS_\Phi^{\cap BC}$, the Bhattacharyya coefficient version, which omits the scaling coefficient before the perceptual kernel $\frac{1}{e^d}$
- *Random*: A uniform random baseline that is a common result from supervised learning or contrastive learning if there is no style label or group (like self and others) information in the training data.



(a) TORCS (b) RGSK

Figure 3: Comparison of Efficacy: Probabilistic vs. Distance Approaches. The plot illustrates the relationship between accuracy (Y-axis) and size of the sampled observation-action pairs (X-axis).

Results presented in Figure 3 suggest that probabilistic similarity tends to yield better results compared to distance-based similarity, offering improved explainability in terms of measure values. Among the methods evaluated, the 2-Wasserstein distance with a perceptual kernel and the Bhattacharyya coefficient emerge as superior candidates. Additional experiments, detailed in Appendix A.4, confirm the consistency of measure values across slightly varied playstyles, further supporting this adaptation.

5.3 Full Data Evaluation

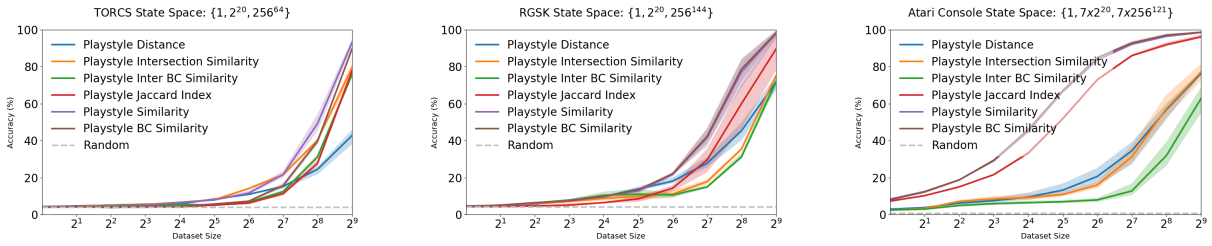
In this section, we perform a comprehensive evaluation of various playstyle measures, including leveraging full data with union operations. The evaluation method mirrors the one presented in Section 5.2, but expands the scope beyond racing games and adopts a multiscale state space.

Given the page limitations of the main paper, detailed results for each Atari game have been moved to Appendix A.3.1. Instead, leveraging the consistent observation and action space shared across Atari games, we propose a unified Atari console evaluation. This evaluation views each DRL agent's gameplay on individual games as distinct playstyles, yielding 7×20 unique playstyles. As for the discrete state space, we include a single shared state mapping in addition to two hierarchical discrete encoders from the seven games; thus, there are totally $1 + 7 \times 2$ discrete state encoders. For actions, rather than aligning their semantics across games, we expand the action set to the largest count in Atari games, which is 18. This is based on the assumption that variations in game content can be interpreted as different states.

The platforms included in the experiments span TORCS, RGSK, and Atari games. We compare the following measures:

- *Playstyle Distance*: $-d_\Phi$
- *Playstyle Intersection Similarity*: $PS_\Phi^\cap$
- *Playstyle Inter BC Similarity*: $PS_\Phi^{\cap BC}$
- *Playstyle Jaccard Index*: J_Φ
- *Playstyle Similarity*: $PS_\Phi^\cup$
- *Playstyle BC Similarity*: $PS_\Phi^{\cup BC}$, the union version of *Playstyle Inter BC Similarity*
- *Random*: A uniform random baseline.

Results displayed in Figure 4 show that the *Playstyle Similarity* outperforms its counterparts. Moreover, the *Jaccard index* has proven to be useful in practice. Our combined Atari console evaluation further underscores the robustness and adaptability of our measures.



(a) TORCS (b) RGSK (c) Atari Console

Figure 4: Playstyle Measure Evaluation in TORCS, RGSK, and Atari Console. The plots showcase the efficacy of different measures in the context of the "Full Data Evaluation" subsection. The shaded area indicates the range between min and max accuracy among three encoder models.

Table 2: Accuracy of game 2048 model identification.

	Accuracy
Playstyle Distance	98.90%
Playstyle Intersection Similarity	98.52%
Playstyle Inter BC Similarity	98.90%
Playstyle Jaccard Index	49.22%
Playstyle Similarity	71.26%
Playstyle BC Similarity	71.26%

Conclusively, our proposed *Playstyle Similarity* measure excels across all platforms. It is particularly impressive that it can identify playstyles with over 90% accuracy with just 512 observation-action pairs — less than half an episode across all tested games. This suggests the possibility of accurate playstyle prediction even before a game concludes, paving the way for real-time analysis.

6 Playstyle Measures Under High Uncertainty Games

To further assess the efficacy of our playstyle measures in games characterized by high uncertainty, we conducted experiments with the puzzle game 2048 (Szubert & Jaskowski, 2014) and the board game Go.

6.1 2048: High Randomness Puzzle Game

Known for its single-player format and high degree of randomness, 2048 presents challenges in generating identical trajectories. We trained a reinforcement learning agent over 10 million episodes, creating 10 distinct players by saving the model every 1 million episodes. For each player, we collected 1000 episodes, using the first 500 as the reference dataset and the remaining 500 as separate query datasets. This resulted in a total of 5000 query datasets for the experiment. The setup aimed to test the accuracy of playstyle measures in scenarios marked by high randomness and similar behaviors across small query datasets, simulating conditions that could challenge discrete playstyle measures. Given that our study is not focused on learning discrete states, we directly utilized the 4×4 full board as the discrete states. The experimental settings used for this analysis are listed in Appendix A.6.

The results in Table 2 demonstrate that measures incorporating the Jaccard index negatively impact accuracy. In games with high randomness, it is crucial to evaluate whether discrete states can identify key factors in the playstyle while maintaining a manageable state space to find comparable samples.

Table 3: Accuracy of Go player identification with MxM query and candidate games.

Accuracy on First 8 Moves	M=1	M=5	M=10	M=25	M=50	M=75	M=100
Playstyle Distance	7.0%	21.0%	42.0%	68.0%	82.0%	88.5%	91.5%
Playstyle Intersection Similarity	3.5%	17.5%	41.5%	73.5%	88.5%	93.0%	93.0%
Playstyle Inter BC Similarity	5.0%	19.0%	34.5%	65.0%	86.5%	91.5%	95.5%
Accuracy on First 16 Moves	M=1	M=5	M=10	M=25	M=50	M=75	M=100
Playstyle Distance	7.0%	20.0%	45.0%	69.0%	85.5%	91.0%	93.0%
Playstyle Intersection Similarity	1.5%	10.5%	33.5%	67.5%	85.0%	89.5%	95.0%
Playstyle Inter BC Similarity	2.0%	9.5%	21.0%	41.5%	67.5%	80.5%	90.0%
Accuracy on First 32 Moves	M=1	M=5	M=10	M=25	M=50	M=75	M=100
Playstyle Distance	8.0%	20.0%	44.0%	69.5%	85.5%	91.5%	94.0%
Playstyle Intersection Similarity	1.5%	7.5%	30.0%	59.5%	81.5%	91.0%	94.0%
Playstyle Inter BC Similarity	2.0%	6.0%	11.5%	23.5%	40.5%	51.0%	62.0%

6.2 Go: Two-Player Board Game

In addition to the inherent randomness of game environments, the inclusion of other players introduces further uncertainty in measuring playstyles. Consequently, we undertook a playstyle classification task using the two-player board game, Go. Data were sourced from Fox Weiqi, one of the largest online Go platforms, consisting of 200 players with skills ranging from 1D to 9D. Each player contributed 100 games for both reference and query datasets. The objective was to correctly identify players using the query datasets, demonstrating the potential of discrete playstyle measures in board games.

Given the absence of a discrete encoder specifically trained for Go and our focus not being on encoder training, we quantized move numbers to establish discrete states. This method segments a game into various stages and analyzes the action distribution within each stage. Notably, opening moves in board games, such as the first N moves in Go, considerably influence playstyle.

In our dataset, a game is represented as a sequence of pairs $[(state_0, action_0), (state_1, action_1), ...]$. Initially, we discard pairs not decided by the player; for instance, in games where the player starts, only even-indexed pairs are considered. We then categorize move numbers by dividing them by integers (e.g., 2, 4, 8), which forms a hierarchical discrete state space. For example, dividing by 2 groups the first two moves are assigned to the same state. Similarly, dividing by 4 groups the first four moves are assigned to the same state. This hierarchical approach allows us to examine how early-game decisions influence overall playstyle. For mid to late game stages, this method is not suitable since actions should focus more on the actual board states.

We tested initial move sets N={8,16,32} with move number steps {2,4,8} as discrete states. Since move numbers are features for discrete states, the Jaccard Index remains consistently at 100%; therefore, we compared three different discrete playstyle measures: Playstyle Distance, Playstyle Intersection Similarity, and Playstyle Inter BC Similarity. The results are shown in Table 3 for the MxM games query and candidate setting, with detailed accuracy tables available in Appendix A.7.

Our findings indicate that discrete playstyle measures can achieve significant accuracy with sufficient samples directly from the measure definition, without any training or predefined style labels on Go. Additionally, perceptual similarity with distance scaling (Playstyle Intersection Similarity) can further enhance accuracy with adequate samples. A deeper analysis incorporating a hierarchical state space and Jaccard Index would be a promising direction for future research on playstyles in board games.

7 Diversity Measurement in DRL

With these discrete playstyle measures, we can design an algorithm to quantify the diversity among DRL models, which is hard to define and measure formally in DRL. Algorithm 1 provides a simple method to

quantify diversity in decision-making by measuring the similarity between a new trajectory and observed trajectories. If a new trajectory is not similar enough to any observed trajectories, we count it as a different one. We conduct a series of experiments on this algorithm in Section A.5, including the analysis of different DRL algorithms and the diversity of a policy with varying levels of stochasticity.

Algorithm 1 Measuring Policy Diversity

Input: Policy π, Environment $\mathcal{E}$, Similarity measure M
Input: Similarity threshold t, Number of trajectories N
Initialize S (store trajectories) and diverse trajectory count $d = 0$
for $i = 1$ **to** N **do**
    Generate a trajectory $\tau_i \sim \pi, \mathcal{E}$
    Set $is_diverse =$ **true**
    for each τ_j in S **do**
        Compute similarity $M(\tau_i, \tau_j)$
        if $M(\tau_i, \tau_j) \geq t$ **then**
            $is_diverse =$ **false**
            break
        end if
    end for
    if $is_diverse$ **then**
        $d = d + 1$
    end if
    Store τ_i in S
end for
Output: Return d (diverse trajectory count) and N (total trajectories)

8 Conclusion and Future Works

In this research, we introduced three techniques to enhance discrete playstyle measures: adopting a multiscale state space, using perceptual similarity rooted in human cognition, and applying the *Jaccard index* to observed data. These advancements have been incorporated into playstyle measurement for the first time and collectively give rise to our playstyle measure, *Playstyle Similarity*. This measure stands out in terms of accuracy and explainability, requiring minimal predefined rules and data. Notably, the integration of a multiscale state space expands the measure's applicability, particularly for games that demand intricate state representations. Furthermore, our literature review and theoretical proof about human perception bridge the gap between AI-driven playstyle similarity measures and human cognition.

The *Playstyle Similarity* measure offers significant potential for real-time game analysis and AI training with specific playstyles, such as diversity analysis or human-like behaviors (Fujii et al., 2013). As an example, we propose an algorithm to quantify the diversity among DRL models. These insights emphasize that AI development can extend beyond simple measures like scores or win rates, encompassing more behavioral patterns. Additionally, the quantification of policy diversity becomes more tangible.

In conclusion, despite our validation efforts across several platforms, many games still remain unexplored. Furthermore, some playstyles can be shared across different games, existing in similar scenarios for game recommendation systems (Fear, 2023). Beyond gaming, playstyle measures could be pivotal for other decision-making topics, such as AI safety (Amodei et al., 2016) and language model agents.

Broader Impact Statement

We advise caution when using these measures for analyzing policy diversity due to potential sensitivities to discretization techniques and inherent explainability challenges in neural networks. Additionally, employing unsupervised methods could raise privacy concerns, as they facilitate targeted advertising or fraudulent activities without the need for costly playstyle labeling.

References

Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2020.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning (ICML)*, 2020.

Anthony Bean and Gary Groth-Marnat. Video gamers and personality: A five-factor model to understand game playing style. *Psychology of Popular Media Culture*, 2016.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2013.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019.

Anil Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, 1946.

Boyan Bontchev and Olga Georgieva. Playing style recognition through an adaptive video game. *Computers in Human Behavior*, 2018.

Patrick Brombacher, Johannes Masino, Michael Frey, and Frank Gauterin. Driving event detection and driving style classification using artificial neural networks. In *IEEE International Conference on Industrial Technology (ICIT)*, 2017.

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A research framework for deep reinforcement learning. *CoRR*, abs/1812.06110, 2018.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In Jennifer G. Dy and Andreas Krause (eds.), *International Conference on Machine Learning (ICML)*, 2018.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations (ICLR)*, 2019.

Jiajun Fan and Changnan Xiao. Generalized data distribution iteration. In *International Conference on Machine Learning, (ICML)*, 2022.

Andrew Fear. Playstyle analysis for game recommendations, 6 2023. US Patent Application US20230177583A1.

Gustav Theodor Fechner. *Elements of Psychophysics*. Holt, Rinehart and Winston, 1966. Originally published in 1860.

Mark Ferguson, Sam Devlin, Daniel Kudenko, and James Alfred Walker. Player style clustering without game variables. In Georgios N. Yannakakis, Antonios Liapis, Penny Kyburz, Vanessa Volz, Foaad Khosmood, and Phil Lopes (eds.), *International Conference on the Foundations of Digital Games (FDG)*, 2020.

Nobuto Fujii, Yuichi Sato, Hironori Wakama, Koji Kazai, and Haruhiro Katayose. Evaluating human-like behaviors of video-game agents autonomously acquired with biological constraints. In *Advances in Computer Entertainment Technology (ACE)*, 2013.

Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent RL without entropy. In *International Conference on Learning Representations (ICLR)*, 2023.

Robert L. Goldstone and Lawrence W. Barsalou. Reuniting perception and conception. *Cognition*, 1998.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *CoRR*, abs/1809.02627, 2020.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 1979.

Guillaume Lample and Devendra Singh Chaplot. Playing FPS games with deep reinforcement learning. In Satinder Singh and Shaul Markovitch (eds.), *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Chiu-Chou Lin, Wei-Chen Chiu, and I-Chen Wu. An unsupervised video game playstyle metric via state discretization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob N. Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning (ICML)*, 2021.

Stéphanie Mader and Eloïse Tassin. Playstyles in tetris: Beyond player skill, score, and competition. In *International Conference on Entertainment Computing (IFIP ICEC)*, 2022.

Reid McIlroy-Young, Yu Wang, Siddhartha Sen, Jon M. Kleinberg, and Ashton Anderson. Detecting individual decision-making style: Exploring behavioral stylometry in chess. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner. Respects for similarity. *Psychological Review*, 1993.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Allan H. Murphy. The finley affair: A signal event in the history of forecast verification. *Weather and Forecasting*, 1996.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Falko Rheinberg. Intrinsic motivation and flow. *Motivation Science*, 2020.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 2018.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, second edition, 2018.

Marcin Grzegorz Szubert and Wojciech Jaskowski. Temporal difference learning of n-tuple networks for the game 2048. In *IEEE Conference on Computational Intelligence and Games (CIG)*, 2014.

Anders Tychsen and Alessandro Canossa. Defining personas in games using metrics. In *International Academic Conference on the Future of Game Design and Technology (FuturePlay)*, 2008.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 1969.

Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 2004.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 2019.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez-Moreno. Generalized end-to-end loss for speaker verification. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2018.

Naoto Yoshida, Alexander Khimulya, Billy Zhou, and Arun Kumar. Gym-TORCS. `https://github.com/ugo-nama-kun/gym_torcs`, 2017.

A Appendix

A.1 A Proof of the Perceptual Kernel

In the main paper, we claim that $P(d) = \frac{1}{e^d}$ is the if and only if kernel function (as discussed in Section 3.2). We provide a proof of this claim using differential equations.

We make some assumptions about the perceptual kernel. First, $P(d)$ is a function that maps the distance d between two given action distributions to a probability value describing their similarity. Since distance is a continuous random variable, we use a probability density function $f(d)$ to describe the mapping function $P(d)$.

We might intuitively think of $P(d)$ as equal to $f(d)$, even though the probability density value is not the same as the probability value. Thus, we use a cumulative distribution function $F(d)$ to describe $P(d)$. We redefine a real-valued random variable D as the distance variable, where $d \in D$, and a random variable X, where $x \in X$, and $D \to X : X(d) = -d$. The probability of similarity, denoted as $P(X \leq x)$, is derived from the distance value d. Thus, $F_X(x) = P(X \leq x)$, and from the assumptions of similarity, $\lim_{x \to -\infty} F_X(x) = 0$, and $F_X(0) = 1$. Also, $F_X(x)$ can be described with a probability density function $f_X(x)$ as follows:

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt \tag{6}$$

The corresponding equation to $F_D(d) = P(d)$ becomes:

$$F_D(d) = \int_{d}^{\infty} f_D(t)dt \tag{7}$$

Additionally, we adopt another assumption from the field of psychophysics known as the *Weber–Fechner law* Fechner (1966). Fechner's law states that the relationship between stimulus S and perception p is logarithmic and can be described as a differential equation:

$$dp = k\frac{dS}{S} \tag{8}$$

Here, k is a constant depending on the sense and type of stimulus.

By integrating the equation, we obtain:

$$p = k \ln S + C \tag{9}$$

Where C is a constant of integration, and it is defined in Fechner's law assuming that the perceived stimulus becomes zero at some threshold stimulus S_0, where $p = 0$, and $S = S_0$. Thus, C can be calculated as follows:

$$C = -k \ln S_0 \tag{10}$$

Combining Equation 9 and Equation 10, Fechner's law Fechner (1966) is:

$$p = k \ln \frac{S}{S_0} \tag{11}$$

Now, we discuss the roles of p and S in our similarity scenario to construct a probability density function $f_D(d)$. We assume that similarity weakens as the distance increases, and there is a finite maximal similarity when the distance is 0. Additionally, the density value and distance are always non-negative.

We first assume that p is the density value of similarity and consider Equation 11. There are two cases:

1. If S represents distance and k is positive, this is incorrect since p approaches $-\infty$ as $S \to 0$.
2. If we change the growth direction of distance so that k is negative, this is still incorrect since p still approaches ∞ as $S \to 0$.

Considering invert Equation 11 as follows:

$$S = S_0 \exp\left(\frac{p}{k}\right) \tag{12}$$

Now, we assume that S is the density value of similarity and consider Equation 12. There are two cases:

1. If p represents distance and k is positive, this is incorrect since S approaches ∞ as $p \to \infty$, although there is a finite maximal value S_0 when $p = 0$.
2. If we change the growth direction of distance so that k is negative, this meets our desire since S approaches 0 as $p \to \infty$, there is a finite maximal value S_0 when $p = 0$, and the density value is always non-negative.

Finally, we can simplify the equations by assuming, for the sake of simplicity, that S_0 equals 1. This assumption is based on the intuition that the trend of decreasing similarity and increasing distance is similar around distance 0 in various scenarios:

$$f_D(d) = \exp\left(\frac{d}{k}\right) \tag{13}$$

Returning to Equation 7, $F_D(d)$ can be described as follows:

$$\begin{aligned} F_D(d) &= \int_d^\infty f_D(t)dt \\ &= \int_d^\infty \exp\left(\frac{t}{k}\right)dt \\ &= \left(\lim_{t\to\infty} k\exp\left(\frac{t}{k}\right) + C'\right) - \left(k\exp\left(\frac{d}{k}\right) + C'\right) \\ &= -k\exp\left(\frac{d}{k}\right) \end{aligned} \tag{14}$$

Considering that the sum of density values must be 1 over $(-\infty, \infty)$, we rewrite Equation 6 as follows:

$$\begin{aligned} \lim_{x\to\infty} F_X(x) &= \int_{-\infty}^x f_X(t)dt \\ &= \int_{-\infty}^\infty f_X(t)dt \\ &= 1 \end{aligned} \tag{15}$$

The corresponding equation to $F_D(d) = P(d)$ becomes:

$$\begin{aligned} \lim_{d\to-\infty} F_D(d) &= \int_d^\infty f_D(t)dt \\ &= \int_{-\infty}^\infty f_D(t)dt \\ &= \int_{-\infty}^0 f_D(t)dt + \int_0^\infty f_D(t)dt \\ &= 0 + \int_0^\infty f_D(t)dt \\ &= 1 \\ \implies F_D(0) &= 1 \end{aligned} \tag{16}$$

Combining Equation 14 and Equation 16:

$$\begin{aligned} F_D(0) &= -k\exp\left(\frac{0}{k}\right) \\ &= -k \\ &= 1 \\ \implies k &= -1 \\ \implies F_D(d) &= \exp\left(\frac{d}{-1}\right) \\ \implies F_D(d) &= e^{\frac{1}{d}} \\ \implies P(d) &= e^{\frac{1}{d}} \end{aligned} \tag{17}$$

Therefore, we have verified the claim that $P(d) = \frac{1}{e^d}$. If there is a case where $S_0 \neq 1$, it is straightforward to derive the equations from Equation 12to 17.

Besides, the expected value of distance is 1 can be obtained by the equations as follows:

$$\begin{aligned}
\mathbb{E}[D] &= \int_{-\infty}^{\infty} x f_D(x) dx \\
&= 0 + \int_{0}^{\infty} x f_D(x) dx \\
&= \int_{0}^{\infty} x \frac{1}{e^x} dx \\
&= (\lim_{t\to\infty} \frac{-t-1}{e^t} + C') - (\frac{-0-1}{e^0} + C') \\
&= 0 - (-1) \\
&= 1
\end{aligned} \tag{18}$$

This concept of expected value is used to scaling the distance value in different scenarios, as described in Section 3.2 with the notation $\overline{D}_{\Phi}^{M}$.

A.2 Perceptual Similarity Under Different State Spaces

There are various methods for generating discrete representations, and the effectiveness of perceptual similarity may vary under these representations, especially when combined with our proposed multiscale state space. In this section, we explore the impact of different state space choices on perceptual similarity.

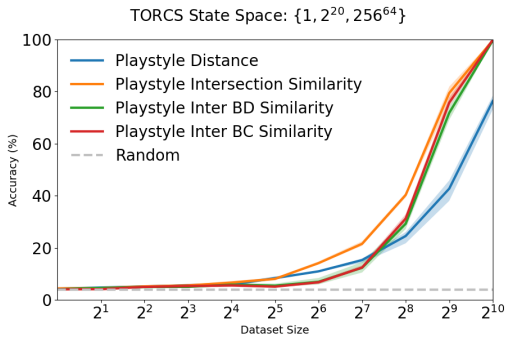
A.2.1 Bhattacharyya distance implementation

In this paper, we also provide some variants of *Playstyle Similarity*, which use Bhattacharyya distance or coefficient as an alternative to the 2-Wasserstein metric to assess the difference in playstyle from a different perspective. Bhattacharyya distance is related to the overlapping region between two distributions, and it is defined through Bhattacharyya coefficient BC. The value range of BC is $[0, 1]$, and the corresponding distance D_B is $D_B = -ln(BC)$. For discrete probability distribution, it is simple to compute the Bhattacharyya coefficient: $BC(P, Q) = \sum_{x\in\mathcal{X}} \sqrt{P(x)Q(x)}$. However, it is more challenging to calculate for continuous probability distributions, as in the case of actions in racing games like TORCS, since it involves the integration of probability density functions: $BC(P, Q) = \int_{x\in\mathcal{X}} \sqrt{p(x)q(x)}$. Thus, we adopt the formulation of multivariate normal distributions of Bhattacharyya distance (D_B) (Bhattacharyya, 1946) as follows, where $p_i = \mathcal{N}(\mu_i, \Sigma_i)$:

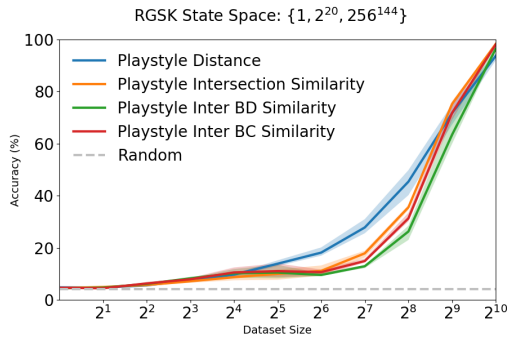
$$\begin{aligned}
&D_B(p_1, p_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln(\frac{\det\Sigma}{\sqrt{\det\Sigma_1 \det\Sigma_2}}) D_B, \\
&\text{where} \quad \Sigma = \frac{\Sigma_1 + \Sigma_2}{2}
\end{aligned} \tag{19}$$

Additionally, we clip the maximum Bhattacharyya distance to 10 to prevent an extremely large value from affecting the average scaling ($\frac{1}{e^{10}} = 0.00004539992 \approx 0\%$). The small value ϵ for dealing with singular matrices in matrix determinant calculation is set to 1e-8.

Recalling our earlier discussion, we mentioned that the Wasserstein distance can be likened to the 'effort' required to transition between different playstyle action distributions (as described in Section 2.2). The Bhattacharyya distance, in contrast, isn't about this 'effort'. Instead, it gauges the likelihood that two playstyles will result in the same action. This is due to its relation to the overlapping regions between two distributions. Thus, while the *Playstyle Similarity* is built on the idea of the effort needed to change playstyles, the *Playstyle BC Similarity* (or its variant *Playstyle BD Similarity*) is built on the frequency of identical actions. This distinction might relate to different roles within a game. For instance, a player in the game might be more concerned with the effort required to shift playstyles, while an observer might focus

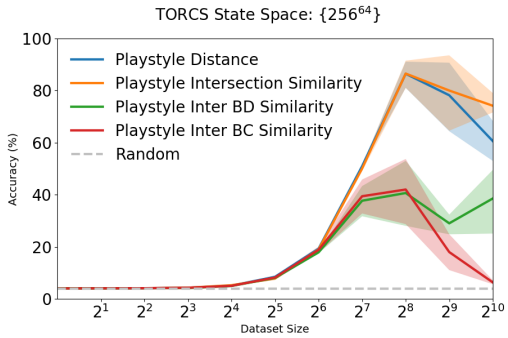


(a) TORCS

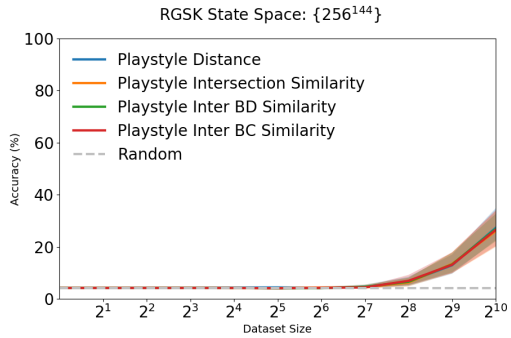


(b) RGSK

Figure 5: Comparison of Efficacy: Using multiscale state space $\{1, 2^{20}, 256^{\text{res}}\}$. The shaded area indicates the range between min and max accuracy among three encoder models.



(a) TORCS



(b) RGSK

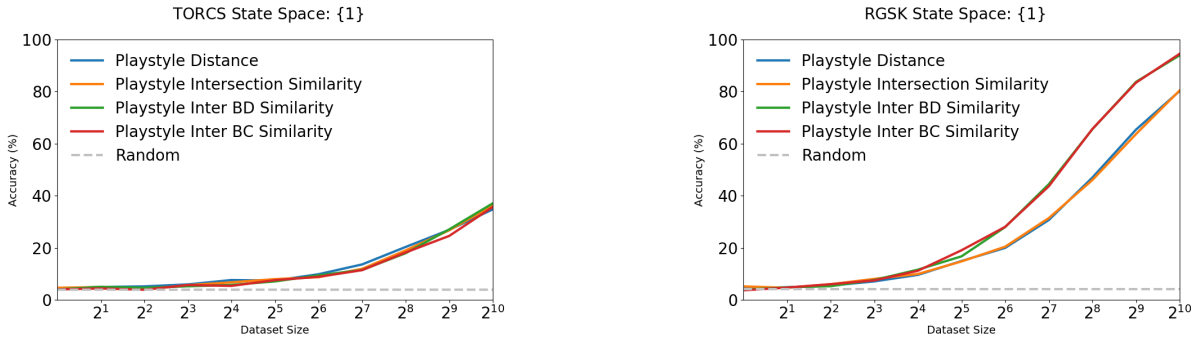
Figure 6: Comparison of Efficacy: Using only the base hierarchy of HSD with state space $\{256^{\text{res}}\}$. The shaded area indicates the range between min and max accuracy among three encoder models.

more on the actions they witness. Think of it this way: players exert effort, like moving their fingers to press buttons or manipulate a joystick or even a mental effort to change their belief of playing. The observer, on the other hand, sees only the outcome of these actions, without much insight into the effort involved.

A.2.2 Multiscale State Space with HSD

Figure 5 presents the results of experiments conducted with a multiscale state space $\{1, 2^{20}, 256^{\text{res}}\}$ generated from HSD models, as described in Section 4.2.

The results indicate an improvement in accuracy for TORCS, while there is no clear improvement in RGSK. Notably, in RGSK, the accuracy of the perceptual kernel with sample sizes 2^5 to 2^8 decreases, suggesting that detailed information for distinguishing these styles has a negative effect. To further investigate, we conducted two ablation studies to understand the effectiveness of the proposed measures for playstyle similarity. The first study focuses on using only the base hierarchy of HSD with a very large state space $\{256^{\text{res}}\}$, while the second study explores the use of a single-state state space $\{1\}$ to assess the measures. Figure 6 illustrates that the measurement is unstable when there are few intersecting samples in a very large state space. However, the negative effect of detailed information is mitigated when considering intersection over union, as shown in Figure 4. Figure 7 shows that even with single state space, this action statistic of dataset can offer some information to differentiate playstyle, especially in RGSK, where Lin et al. (2021) made thier human players follow some playstyles closely related to the keyboard actions, such as using nitro system or breaking in the racing game.



(a) TORCS (b) RGSK

Figure 7: Comparison of Efficacy: Using only the single state space {1}.

A.2.3 Discrete Representation from Downsampling

There are several existing methods for generating discrete representations. One conventional method for image data is downsampling to a lower resolution. While the downsampling parameters often require tuning for effective processing, it is a straightforward method that does not require training a neural network model. Previous work by Lin et al. (2021) attempted to use low-resolution downsampling as a discretization method, but they encountered challenges due to the lack of intersection states in their settings.

In our experiments, we explore the use of downsampling to create discrete representations with different state spaces. In TORCS, we map original game screen observations to three levels of state space:

1. 1: Basic mapping with state space 1, which maps all observations identically.
2. $16^{8\times8}$: Downsampling from 4 × [64,64,3] 256-intensity observations to 1 × [8,8,1] 16-intensity observations.
3. $16^{8\times8\times4}$: Downsampling from 4 × [64,64,3] 256-intensity observations to 4 × [8,8,1] 16-intensity observations.

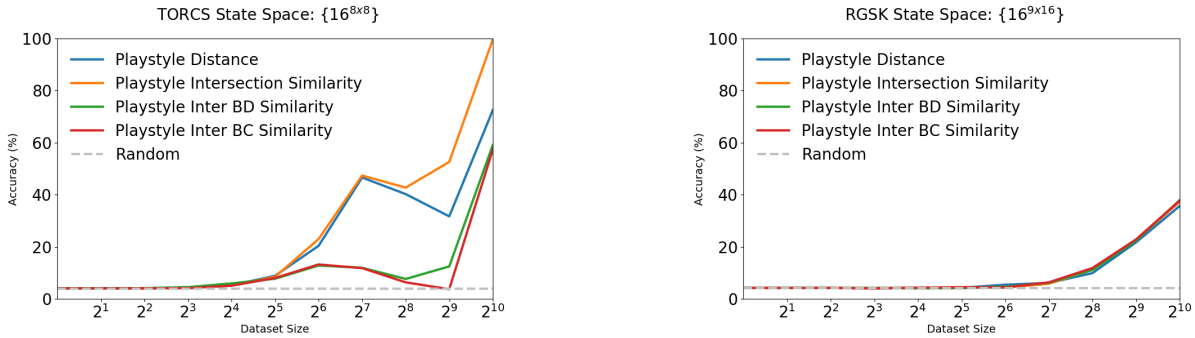
For RGSK, we similarly map original game screen observations to three levels of state space:

1. 1: Basic mapping with state space 1, which maps all observations identically.
2. $16^{9\times16}$: Downsampling from 4 × [72,128,3] 256-intensity observations to 1 × [9,16,1] 16-intensity observations.
3. $16^{9\times16\times4}$: Downsampling from 4 × [72,128,3] 256-intensity observations to 4 × [9,16,1] 16-intensity observations.

The results in Figure 8-10 show that downsampling can be a viable discretization method in some cases, but overall, the measurement is either unstable or shows no significant difference compared in these measures. These results highlight the importance of having discrete representations with high quality, providing proper granularity for playstyle features.

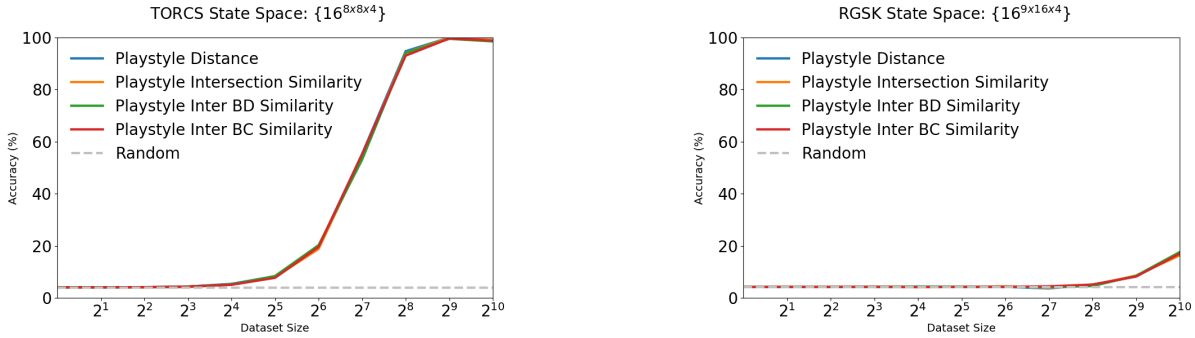
A.3 More Results of Full Data Evaluation

In this supplementary section, we delve deeper into the results to provide a comprehensive analysis of measures evaluations under different state spaces.



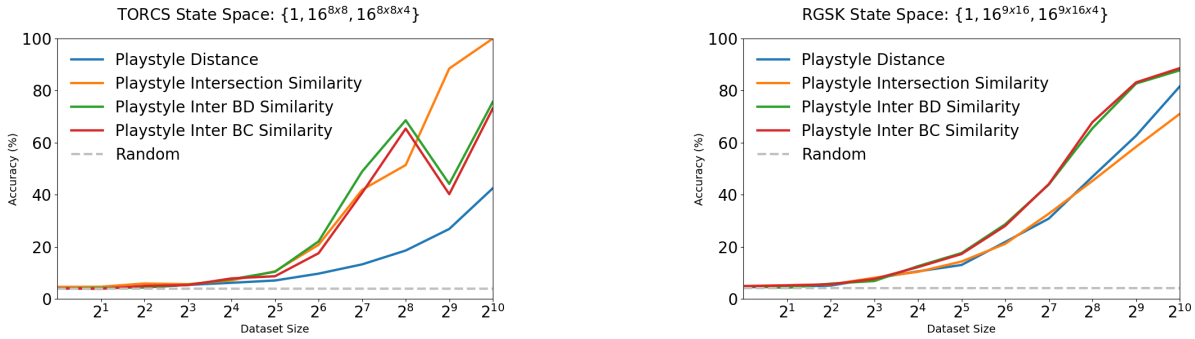
(a) TORCS (b) RGSK

Figure 8: Evaluation of discrete representations using downsampling, considering intersection states in TORCS with state space $\{16^{8\times8}\}$ and in RGSK with state space $\{16^{9\times16}\}$.



(a) TORCS (b) RGSK

Figure 9: Evaluation of discrete representations using downsampling, considering intersection states in TORCS with state space $\{16^{8\times8\times4}\}$ and in RGSK with state space $\{16^{9\times16\times4}\}$.



(a) TORCS (b) RGSK

Figure 10: Evaluation of discrete representations using downsampling, considering intersection states in TORCS with state space $\{1, 16^{8\times8}, 16^{8\times8\times4}\}$ and in RGSK with state space $\{1, 16^{9\times16}, 16^{9\times16\times4}\}$.

A.3.1 Individual Atari game results with multiscale state space

Figure 11 shows the relationship between playstyle classification accuracy and sampled dataset size for the seven Atari games. *Playstyle Similarity* ($PS_{\Phi}^{\cup}$) and its variant *Playstyle BC Similarity* ($PS_{\Phi}^{\cup BC}$) have nearly the same performance, and *Playstyle Jaccard Index* (J_{Φ}) can have a decent result. This evidence justifies that some playstyles, especially in a deterministic environment, can be differentiated solely with observations, which explains why the work by Eysenbach et al. (2019) considers states only for diversity.

A.3.2 Atari game results with a smaller state space

Figure 12 shows the relationship between playstyle classification accuracy and sampled dataset size for the seven Atari games and the combined version (Atari Console). These results show that measures with intersection over union still perform well in Atari games even with a smaller state space. Although it seems that *Playstyle Jaccard Index* is a decent and easy measure, we know that it theoretically does not work as long as all states are visited in the sampled dataset, as described in Section 3.3. This potential problem is discussed in more detail in Section A.3.3, where even with a state space of 2^{20}, the *Playstyle Jaccard Index* may not perform well.

A.3.3 TORCS and RGSK with a smaller state space

In this section, we conducted experiments using a reduced state space of 2^{20} for the two racing games, TORCS and RGSK, without employing the multiscale technique.

Figure 13 illustrates that the *Playstyle Jaccard Index* performs the poorest in TORCS and exhibits slightly inferior performance to *Playstyle Similarity* and *Playstyle BC Similarity* in RGSK. This observation provides valuable insights into the suitability of the *Playstyle Jaccard Index* for precise measurements, particularly in scenarios involving randomness (e.g., TORCS players employing different action noises) or where observations exhibit only slight variations (e.g., stable rule-based AI controllers in TORCS with slightly different target). Further investigation may be warranted to understand the reasons behind these performance differences.

A.4 Continuous Playstyle Spectrum in TORCS

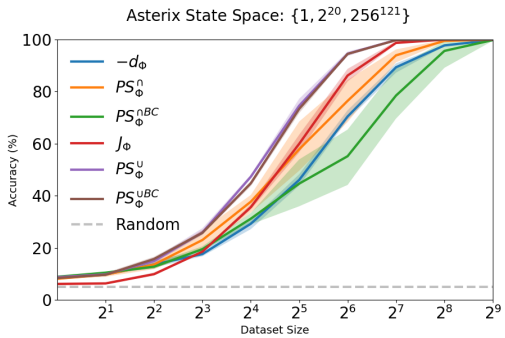
This experiment investigates the response of similarity measure values to variations in a continuous playstyle spectrum within the TORCS environment. It particularly focuses on whether these measures can accurately rank playstyles, ensuring precise predictions for the closest playstyle (Top-1 similarity) and maintaining a correctly ordered sequence of playstyles based on similarity measures.

Utilizing the TORCS dataset, which includes five levels of target speeds (60, 65, 70, 75, 80) and five levels of action noise, provides a broad spectrum for examining continuous playstyle changes. To illustrate, consider five playstyles labeled A through E, with A' being a variant closely aligned with A. We anticipate that the similarity between A' and A would be the highest, progressively decreasing towards E. This expectation sets the stage for our consistency test, wherein a similarity measure M should validate the order $M(A', A) > M(A', B) > M(A', C) > M(A', D) > M(A', E)$ (Corner Case).

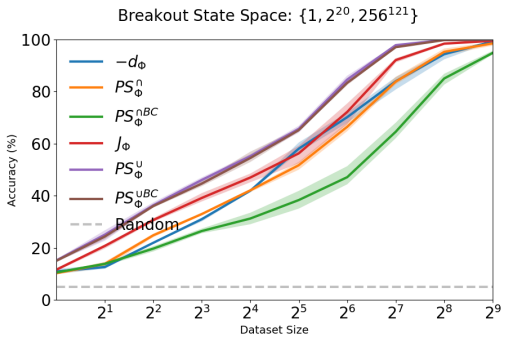
We assess the following measures:

1. Playstyle Distance with a 2^{20} state space (baseline)
2. Playstyle Distance using a mixed state space
3. Playstyle Intersection Similarity with a mixed state space
4. Playstyle Jaccard Index with a mixed state space
5. Playstyle Similarity with a mixed state space

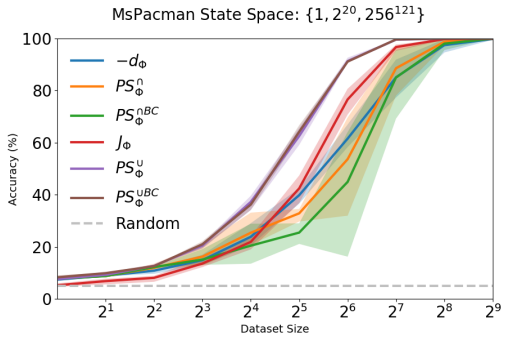
With 100 rounds of random subsampling (each dataset consisting of 512 observation-action pairs) and using the first Hierarchical State Discretization (HSD) model, we examine the consistency of similarity values as



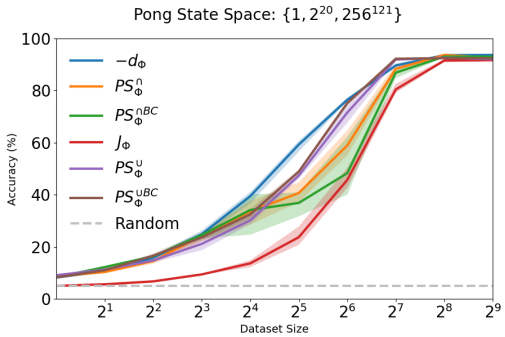
(a) Asterix



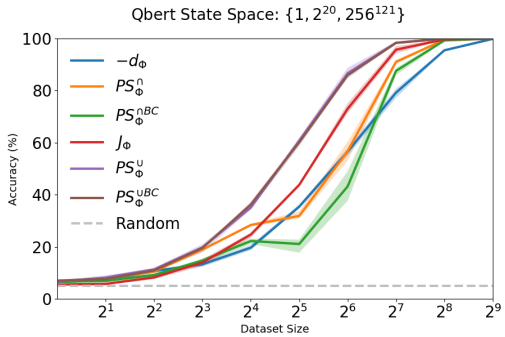
(b) Breakout



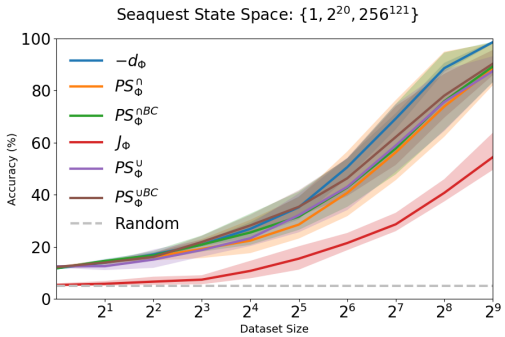
(c) MsPacman



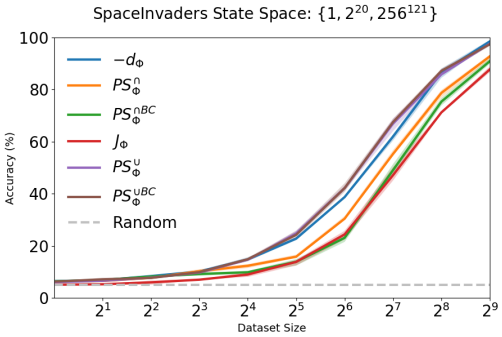
(d) Pong



(e) Qbert

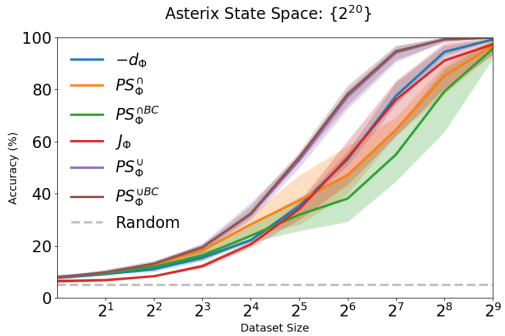


(f) Seaquest

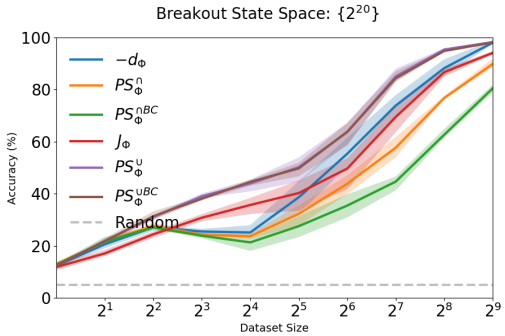


(g) SpaceInvaders

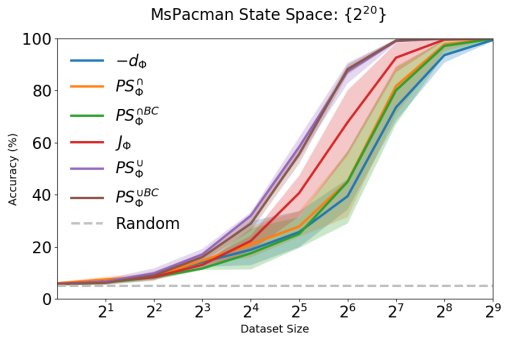
Figure 11: Playstyle Measure Evaluation in Atari games. The plots showcase the efficacy of different measures in the context of the "Full Data Evaluation" subsection. The shaded area indicates the range between min and max accuracy among three encoder models.



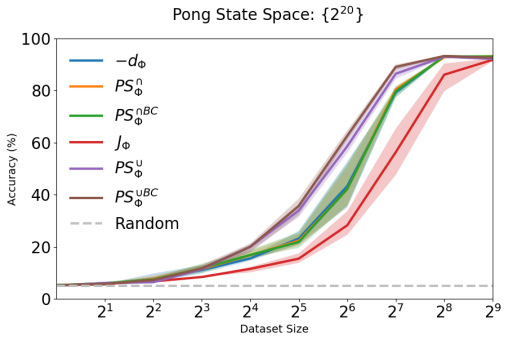
(a) Asterix



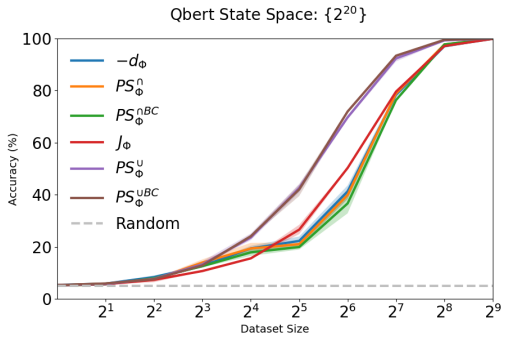
(b) Breakout



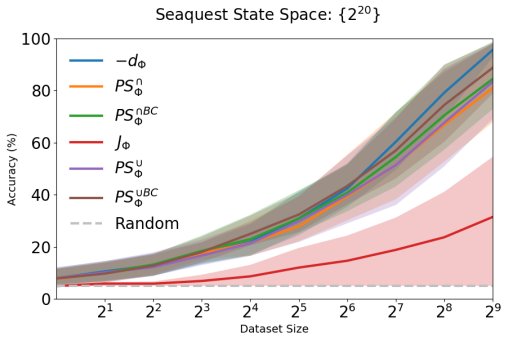
(c) MsPacman



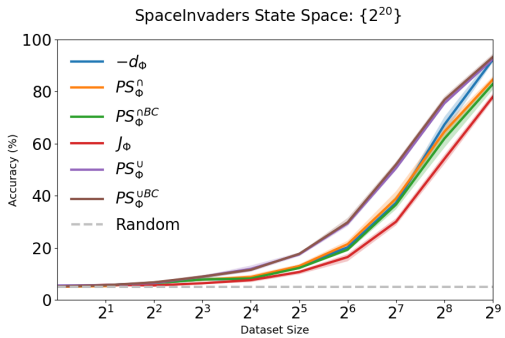
(d) Pong



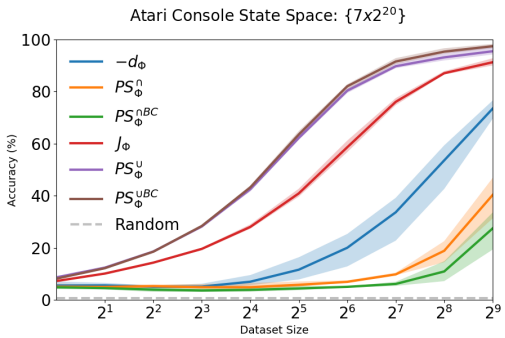
(e) Qbert



(f) Seaquest

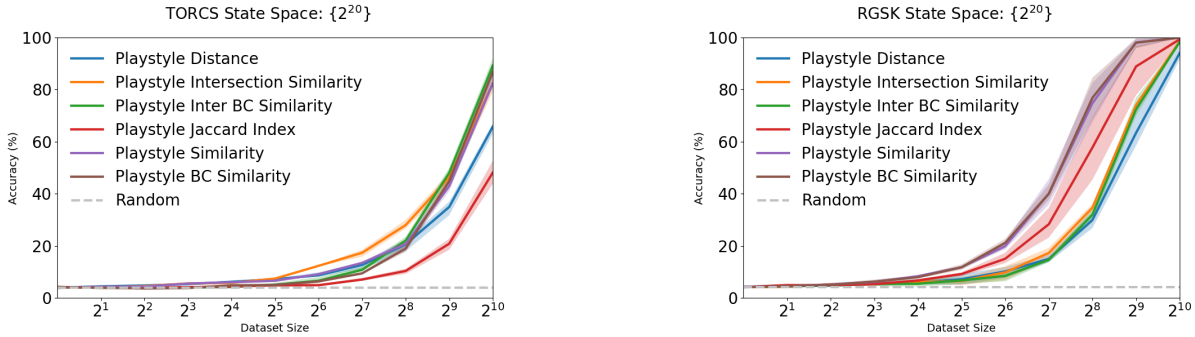


(g) SpaceInvaders



(h) Atari Console

Figure 12: Playstyle Measure Evaluation in Atari games. The plots showcase the efficacy of different measures with a 2^{20} state space from HSD models. The shaded area indicates the range between min and max accuracy among three encoder models.



(a) TORCS (b) RGSK

Figure 13: Playstyle Measure Evaluation in two racing games. The plots showcase the efficacy of different measures with a 2^{20} state space from HSD models. The shaded area indicates the range between min and max accuracy among three encoder models.

Table 4: Consistency count of two continuous playstyle spectrum cases.

	Corner Case	Center Case
Playstyle Distance (2^{20})	3	2
Playstyle Distance (mixed)	4	2
Playstyle Intersection Similarity (mixed)	8	2
Playstyle Jaccard Index (mixed)	5	1
Playstyle Similarity (mixed)	**9**	**3**

playstyle shifts. A decrease in similarity consistent with playstyle changes is marked with a (C), indicating measure reliability.

For instance, choosing Speed60N0 as a target playstyle, we observe how similarity measures adjust across a row or column in response to increasing speed or action noise levels. A practical demonstration reveals how the measure values consistently decrease across increasing speed levels and noise intensities, illustrating the measure's ability to capture playstyle divergence accurately.

Similar analyses extend to a Center Case scenario with Speed70N2 as the target playstyle, emphasizing the necessity for similarity measures to respect two key sequences for consistency:

1. $M(C', C) > M(C', B) > M(C', A)$
2. $M(C', C) > M(C', D) > M(C', E)$

These sequences confirm the measure's capacity to accurately reflect the gradual divergence of playstyles from a central reference point. The detailed findings, presented in Tables 4-14, underscore the nuanced performance of each evaluated measure. Additionally, we present the standard deviation of the measure values, denoted as (std), alongside the measure values themselves.

A.5 More Experiments on Diversity Measurement in DRL

This section introduces additional experiments demonstrating the application of our method for quantifying decision-making diversity, detailed in Algorithm 1. We analyze the Atari DRL agent dataset from the main paper, which includes 5 models per DRL algorithm, each contributing 5 trajectories, totaling 25 trajectories per algorithm (Mnih et al., 2015; Bellemare et al., 2017; Hessel et al., 2018; Dabney et al., 2018). Using Algorithm 1, we assess the diversity of trajectories produced by each algorithm. Results in Table 15, averaged across three discrete encoder models with a similarity threshold of $t = 0.2$, show the capacity of models trained

Table 5: Corner Case uses Playstyle Distance (2^{20}): consistent count = 3

Speed60N0	Speed60 (C)	Speed65	Speed70	Speed75	Speed80
N0 (C)	-0.0044 (0.0012)	-0.0048 (0.0010)	-0.0055 (0.0010)	-0.0069 (0.0011)	-0.0100 (0.0014)
N1 (C)	-0.0054 (0.0011)	-0.0064 (0.0015)	-0.0069 (0.0016)	-0.0086 (0.0013)	-0.0108 (0.0016)
N2	-0.0064 (0.0010)	-0.0058 (0.0010)	-0.0068 (0.0013)	-0.0086 (0.0010)	-0.0097 (0.0015)
N3	-0.0073 (0.0011)	-0.0078 (0.0013)	-0.0071 (0.0010)	-0.0100 (0.0017)	-0.0118 (0.0016)
N4	-0.0092 (0.0013)	-0.0083 (0.0011)	-0.0089 (0.0013)	-0.0110 (0.0015)	-0.0132 (0.0017)

Table 6: Corner Case uses Playstyle Distance (mix): consistent count = 5

Speed60N0	Speed60 (C)	Speed65	Speed70	Speed75 (C)	Speed80
N0 (C)	-0.0018 (0.0005)	-0.0022 (0.0005)	-0.0028 (0.0006)	-0.0037 (0.0006)	-0.0060 (0.0007)
N1 (C)	-0.0024 (0.0005)	-0.0028 (0.0007)	-0.0036 (0.0008)	-0.0043 (0.0007)	-0.0058 (0.0008)
N2	-0.0028 (0.0005)	-0.0027 (0.0006)	-0.0031 (0.0006)	-0.0045 (0.0007)	-0.0056 (0.0008)
N3 (C)	-0.0030 (0.0005)	-0.0033 (0.0007)	-0.0035 (0.0006)	-0.0049 (0.0007)	-0.0060 (0.0009)
N4	-0.0038 (0.0006)	-0.0037 (0.0006)	-0.0040 (0.0006)	-0.0055 (0.0008)	-0.0071 (0.0011)

Table 7: Corner Case uses Playstyle Intersection Similarity (mix): consistent count = 8

Speed60N0	Speed60 (C)	Speed65 (C)	Speed70 (C)	Speed75 (C)	Speed80 (C)
N0 (C)	0.8014 (0.0206)	0.7502 (0.0187)	0.7170 (0.0233)	0.6538 (0.0231)	0.5829 (0.0246)
N1 (C)	0.6927 (0.0234)	0.6865 (0.0240)	0.6646 (0.0218)	0.6254 (0.0265)	0.5508 (0.0250)
N2	0.6260 (0.0266)	0.6499 (0.0257)	0.6354 (0.0299)	0.5709 (0.0254)	0.5450 (0.0284)
N3	0.5813 (0.0282)	0.5857 (0.0250)	0.5721 (0.0302)	0.5507 (0.0276)	0.4825 (0.0321)
N4 (C)	0.5420 (0.0268)	0.5390 (0.0298)	0.5322 (0.0310)	0.4708 (0.0288)	0.4544 (0.0328)

Table 8: Corner Case uses Playstyle Jaccard Index (mix): consistent count = 5

Speed60N0	Speed60 (C)	Speed65	Speed70 (C)	Speed75 (C)	Speed80 (C)
N0 (C)	0.0938 (0.0059)	0.0863 (0.0042)	0.0841 (0.0044)	0.0840 (0.0048)	0.0816 (0.0042)
N1	0.0845 (0.0046)	0.0853 (0.0045)	0.0833 (0.0040)	0.0821 (0.0043)	0.0815 (0.0040)
N2	0.0827 (0.0045)	0.0816 (0.0037)	0.0830 (0.0047)	0.0812 (0.0043)	0.0799 (0.0042)
N3	0.0821 (0.0040)	0.0820 (0.0048)	0.0790 (0.0042)	0.0806 (0.0040)	0.0773 (0.0049)
N4	0.0780 (0.0046)	0.0762 (0.0045)	0.0757 (0.0038)	0.0750 (0.0042)	0.0760 (0.0041)

Table 9: Corner Case uses Playstyle Similarity (mix): consistent count = 9

Speed60N0	Speed60 (C)	Speed65 (C)	Speed70 (C)	Speed75 (C)	Speed80 (C)
N0 (C)	0.0753 (0.0046)	0.0650 (0.0035)	0.0608 (0.0034)	0.0555 (0.0026)	0.0472 (0.0035)
N1 (C)	0.0590 (0.0035)	0.0579 (0.0033)	0.0549 (0.0034)	0.0513 (0.0035)	0.0446 (0.0031)
N2	0.0519 (0.0036)	0.0533 (0.0033)	0.0525 (0.0033)	0.0458 (0.0032)	0.0440 (0.0034)
N3 (C)	0.0482 (0.0033)	0.0473 (0.0034)	0.0448 (0.0031)	0.0435 (0.0032)	0.0382 (0.0030)
N4 (C)	0.0421 (0.0034)	0.0413 (0.0031)	0.0402 (0.0035)	0.0349 (0.0030)	0.0337 (0.0032)

Table 10: Center Case uses Playstyle Distance (2^{20}): consistent count = 2

Speed70N2	Speed60	Speed65	Speed70	Speed75	Speed80
N0	-0.0068 (0.0013)	-0.0068 (0.0012)	-0.0065 (0.0011)	-0.0068 (0.0012)	-0.0097 (0.0017)
N1	-0.0073 (0.0011)	-0.0069 (0.0013)	-0.0073 (0.0014)	-0.0084 (0.0014)	-0.0095 (0.0014)
N2 (C)	-0.0079 (0.0010)	-0.0075 (0.0015)	-0.0069 (0.0013)	-0.0087 (0.0014)	-0.0095 (0.0016)
N3 (C)	-0.0085 (0.0012)	-0.0080 (0.0013)	-0.0075 (0.0011)	-0.0090 (0.0014)	-0.0103 (0.0013)
N4	-0.0105 (0.0013)	-0.0093 (0.0014)	-0.0094 (0.0013)	-0.0112 (0.0015)	-0.0127 (0.0016)

Table 11: Center Case uses Playstyle Distance (mix): consistent count = 2

Speed70N2	Speed60	Speed65	Speed70	Speed75	Speed80
N0	-0.0033 (0.0006)	-0.0029 (0.0006)	-0.0029 (0.0006)	-0.0033 (0.0005)	-0.0046 (0.0007)
N1	-0.0035 (0.0006)	-0.0032 (0.0007)	-0.0034 (0.0007)	-0.0036 (0.0007)	-0.0046 (0.0006)
N2 (C)	-0.0038 (0.0005)	-0.0035 (0.0008)	-0.0031 (0.0007)	-0.0037 (0.0006)	-0.0047 (0.0009)
N3 (C)	-0.0039 (0.0006)	-0.0036 (0.0006)	-0.0034 (0.0005)	-0.0040 (0.0007)	-0.0047 (0.0007)
N4	-0.0046 (0.0006)	-0.0037 (0.0005)	-0.0040 (0.0006)	-0.0046 (0.0006)	-0.0060 (0.0009)

Table 12: Center Case uses Playstyle Intersection Similarity (mix): consistent count = 2

Speed70N2	Speed60	Speed65	Speed70	Speed75	Speed80
N0 (C)	0.6774 (0.0241)	0.6944 (0.0206)	0.6995 (0.0233)	0.6625 (0.0246)	0.6286 (0.0265)
N1	0.6295 (0.0252)	0.6609 (0.0259)	0.6603 (0.0247)	0.6456 (0.0253)	0.6031 (0.0281)
N2 (C)	0.5962 (0.0323)	0.6376 (0.0308)	0.7100 (0.0231)	0.6086 (0.0291)	0.6034 (0.0273)
N3	0.5891 (0.0272)	0.5844 (0.0277)	0.5999 (0.0261)	0.5764 (0.0297)	0.5486 (0.0256)
N4	0.5129 (0.0306)	0.5664 (0.0274)	0.5503 (0.0278)	0.5135 (0.0297)	0.4841 (0.0253)

Table 13: Center Case uses Playstyle Jaccard Index (mix): consistent count = 1

Speed70N2	Speed60	Speed65	Speed70	Speed75	Speed80
N0	0.0823 (0.0046)	0.0829 (0.0047)	0.0827 (0.0046)	0.0844 (0.0043)	0.0831 (0.0051)
N1	0.0836 (0.0050)	0.0820 (0.0048)	0.0846 (0.0043)	0.0850 (0.0045)	0.0855 (0.0050)
N2 (C)	0.0821 (0.0050)	0.0825 (0.0043)	0.0930 (0.0049)	0.0845 (0.0042)	0.0842 (0.0045)
N3	0.0806 (0.0042)	0.0827 (0.0044)	0.0795 (0.0049)	0.0839 (0.0043)	0.0780 (0.0045)
N4	0.0802 (0.0046)	0.0792 (0.0046)	0.0799 (0.0048)	0.0782 (0.0047)	0.0804 (0.0050)

Table 14: Center Case uses Playstyle Similarity (mix): consistent count = 3

Speed70N2	Speed60	Speed65	Speed70	Speed75	Speed80
N0 (C)	0.0554 (0.0040)	0.0573 (0.0039)	0.0579 (0.0043)	0.0568 (0.0034)	0.0522 (0.0037)
N1 (C)	0.0527 (0.0036)	0.0555 (0.0041)	0.0560 (0.0035)	0.0549 (0.0038)	0.0518 (0.0037)
N2 (C)	0.0486 (0.0038)	0.0522 (0.004)	0.0652 (0.0043)	0.0512 (0.0035)	0.0498 (0.0035)
N3	0.0486 (0.0033)	0.0487 (0.0035)	0.0479 (0.0037)	0.0485 (0.0038)	0.0438 (0.0030)
N4	0.0417 (0.0033)	0.0452 (0.0033)	0.0444 (0.0037)	0.0407 (0.0035)	0.0399 (0.0035)

under the same DRL algorithm to generate diverse game episodes within 25 attempts. The IQN algorithm displays higher diversity across games, consistent with its risk sampling feature.

We also apply various levels of stochasticity using the Dopamine framework to illustrate another use case of this measure. Our focus is on the first IQN model from Dopamine, which is expected to adapt to a wide array of playstyles due to its risk functions and robust performance capabilities. To encourage diversity, we employ the Boltzmann distribution, commonly known as softmax, varying the temperature (z) to influence decision-making:

$$\pi(s) = \text{Softmax}\left(\frac{A(s)}{z}\right) \tag{20}$$

This approach, inspired by Fan & Xiao (2022), uses the advantage function (A) crucial in action selection in reinforcement learning, where a higher advantage indicates a preferable action.

We explore four levels of randomness ($z \in \{0.0001, 0.001, 0.01, 0.1\}$), anticipating increased diversity with greater randomness. We test similarity thresholds (t) of 0.5, 0.2, and 0.05 across seven Atari games with 100 trajectories each. Results are illustrated through shaded curves in Figures from Figure 14 to Figure 16, demonstrating the efficacy of our diversity measure.

Notably, games like *Seaquest* (Figure 17b) exhibit high diversity even at lower randomness levels, indicating intrinsic complexity in terms of playstyles. In contrast, *Qbert* (Figure 17a) becomes more monotonous since the goal is to achieve a higher score in the puzzle game. This observation suggests another application for our measure: identifying the complexity of game content. The time complexity of Algorithm 1 is $O(N^2)$, given

Table 15: Averaged diverse trajectory count of different DRL algorithms across 7 Atari games in 25 episodes.

DRL Algorithm	Asterix	Breakout	MsPacman	Pong	Qbert	Seaquest	SpaceInvaders
DQN	6.00	6.00	5.33	4.00	6.00	11.00	25.00
C51	6.00	7.00	7.00	6.00	7.00	21.00	25.00
Rainbow	8.67	5.33	8.00	5.00	5.00	24.00	25.00
IQN	25.00	14.00	10.00	9.00	10.00	25.00	25.00

Table 16: Accuracy of Go player identification with the Playstyle Distance.

First 8 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	7.0%	14.5%	13.5%	16.0%	22.5%	23.0%	23.0%
Candidate 5	12.5%	21.0%	26.5%	32.5%	36.0%	40.5%	41.0%
Candidate 10	12.5%	33.0%	42.0%	54.5%	57.0%	61.0%	64.0%
Candidate 25	14.0%	47.0%	54.0%	68.0%	76.5%	80.0%	87.0%
Candidate 50	17.0%	48.5%	59.5%	78.0%	82.0%	88.0%	91.5%
Candidate 75	17.0%	48.5%	59.5%	79.5%	85.5%	88.5%	92.0%
Candidate 100	17.5%	51.5%	62.5%	81.0%	87.0%	90.5%	91.5%
First 16 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	7.0%	17.5%	18.0%	20.5%	25.5%	27.0%	27.0%
Candidate 5	11.0%	20.0%	23.5%	33.0%	39.0%	40.5%	44.0%
Candidate 10	13.0%	35.0%	45.0%	53.5%	58.0%	62.0%	68.0%
Candidate 25	17.5%	48.5%	53.5%	69.0%	79.0%	83.5%	87.5%
Candidate 50	17.5%	48.0%	61.0%	79.0%	85.5%	89.0%	92.0%
Candidate 75	17.0%	50.5%	65.5%	83.0%	88.0%	91.0%	93.0%
Candidate 100	16.5%	54.5%	65.5%	81.0%	88.5%	90.0%	93.0%
First 32 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	8.0%	18.0%	15.0%	20.5%	26.0%	27.0%	27.5%
Candidate 5	12.0%	20.0%	24.0%	29.0%	38.0%	42.0%	43.0%
Candidate 10	14.0%	34.5%	44.0%	51.0%	57.0%	62.0%	65.0%
Candidate 25	17.5%	47.0%	55.5%	69.5%	80.0%	83.5%	86.5%
Candidate 50	17.0%	50.5%	59.5%	77.0%	85.5%	88.5%	91.5%
Candidate 75	17.0%	51.5%	63.0%	82.5%	89.0%	91.5%	94.0%
Candidate 100	18.0%	55.0%	65.0%	83.0%	89.5%	91.0%	94.0%

the number of trajectories N. Future research could investigate more efficient methods, perhaps leveraging approximations or advanced data structures for quicker similarity checks.

The algorithm we introduce for diversity quantification shifts our understanding of gaming from the subjective to the quantitative. While various methodologies for measuring diversity exist in different domains, our approach is particularly apt to video game playing. In addition, recognizing and quantifying this diversity can inform the development of more adaptive DRL models, thereby addressing specific challenges in gaming and artificial intelligence. This new measure contributes to our progress toward models that are not only efficient but also demonstrate a variety of adaptable strategies, opening up vast avenues for future research.

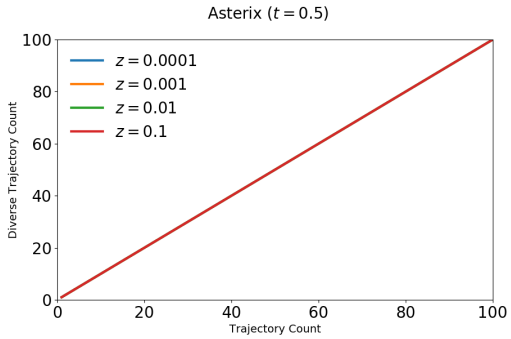
A.6 Game 2048 Experiment Details

We employed temporal difference learning to train our 2048 agents, utilizing training code available on GitHub[2]. We set the learning rate (α) to 0.01 and maintained all other default settings. As this study does not focus on optimizing discrete representations, we directly use the full board as the discrete states for playstyle analysis. The potential state space for the 4×4 version of 2048 is calculated to be 16^{16} (Figure 18).

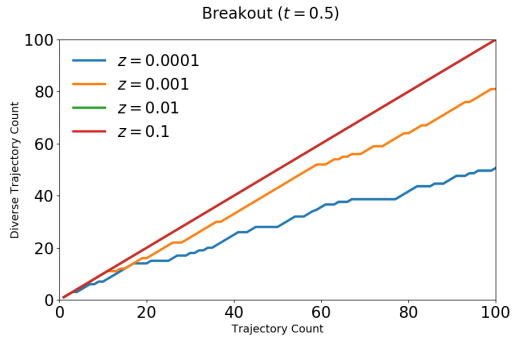
A.7 Go Experiment Details

Our Go setting follows the standard 19x19 rules, and the action space includes 362 actions (361 move actions and a pass). Tables 16-18 list the detailed accuracies for different query and candidate sizes in Section 6.2.

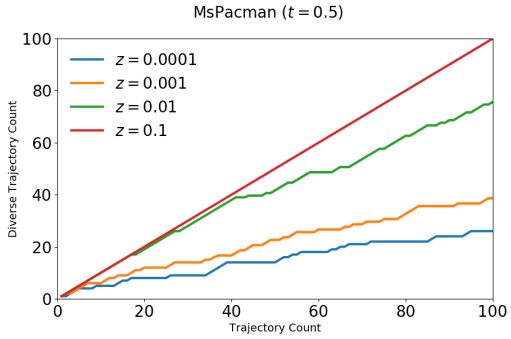
[2] https://github.com/moporgic/TDL2048-Demo



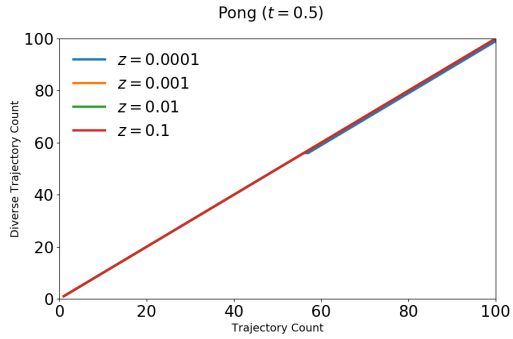
(a) Asterix



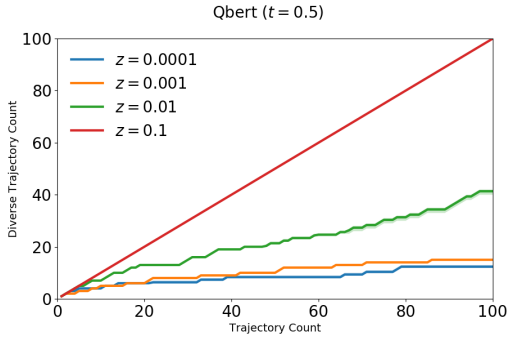
(b) Breakout



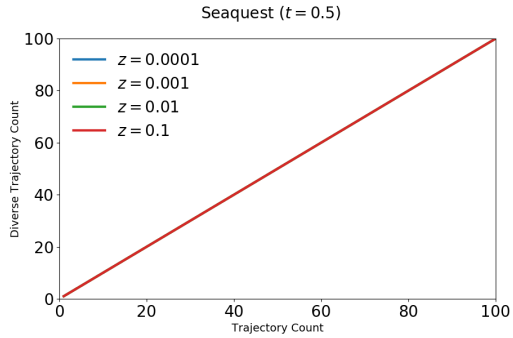
(c) MsPacman



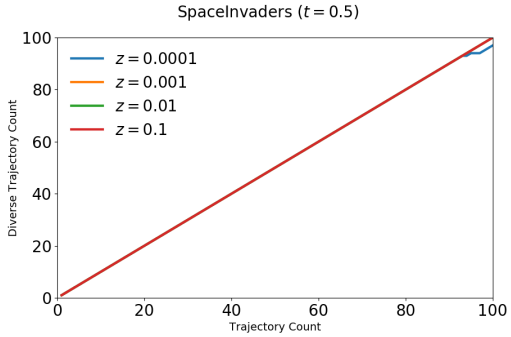
(d) Pong



(e) Qbert

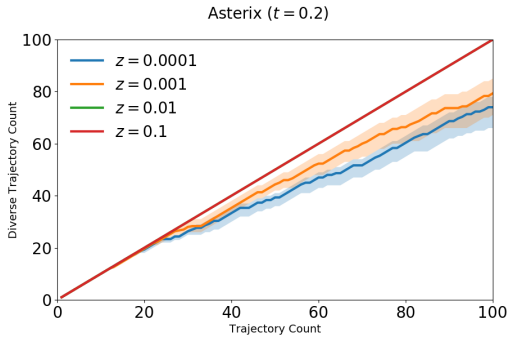


(f) Seaquest

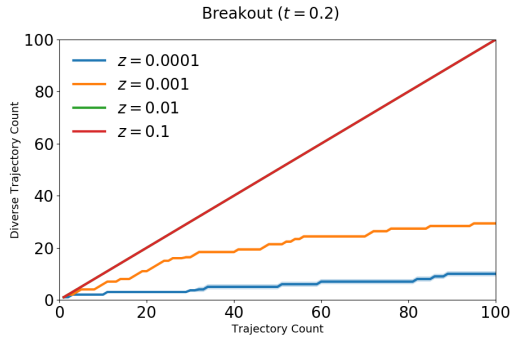


(g) SpaceInvaders

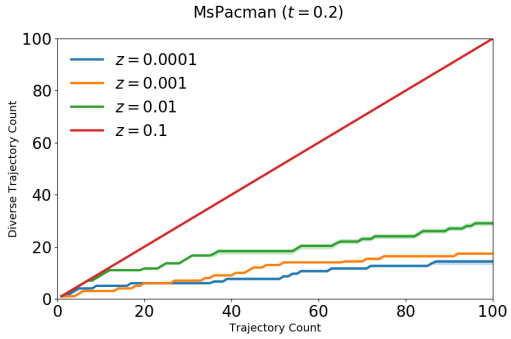
Figure 14: Diversity Measurement of IQN Models in Seven Atari Games ($t = 0.5$)



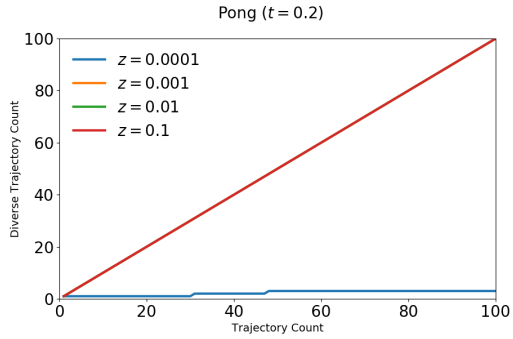
(a) Asterix



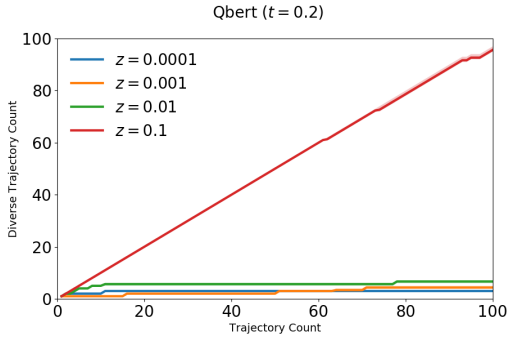
(b) Breakout



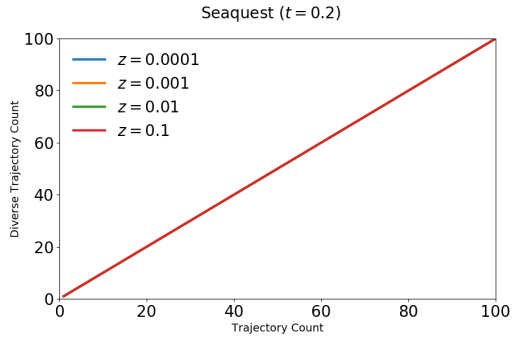
(c) MsPacman



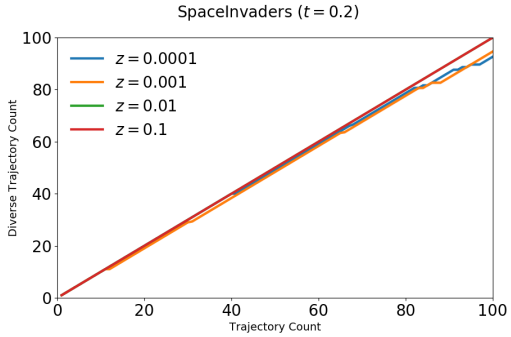
(d) Pong



(e) Qbert

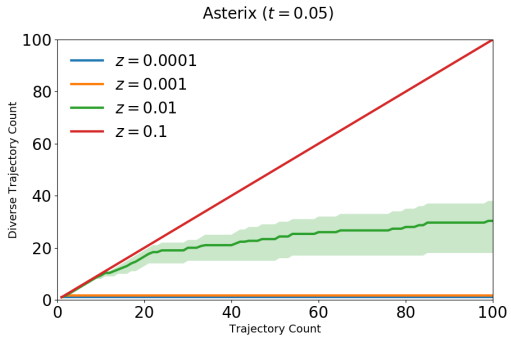


(f) Seaquest

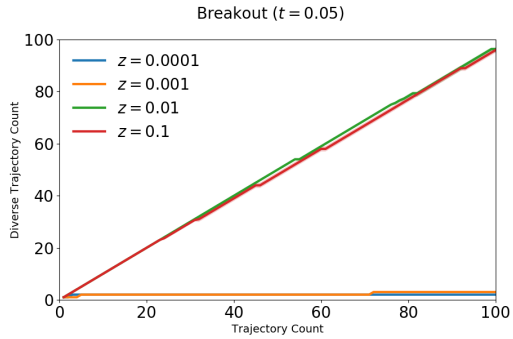


(g) SpaceInvaders

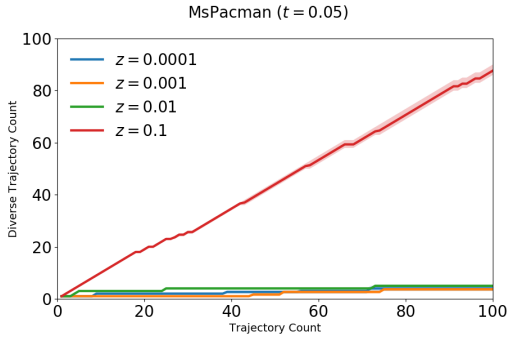
Figure 15: Diversity Measurement of IQN Models in Seven Atari Games ($t = 0.2$)



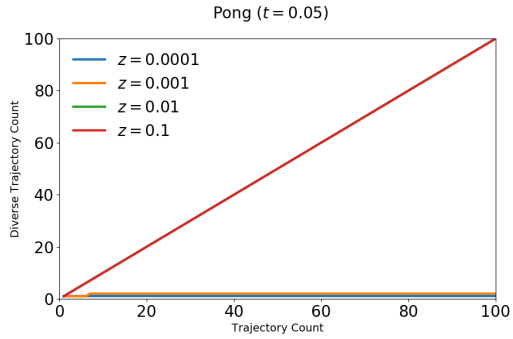
(a) Asterix



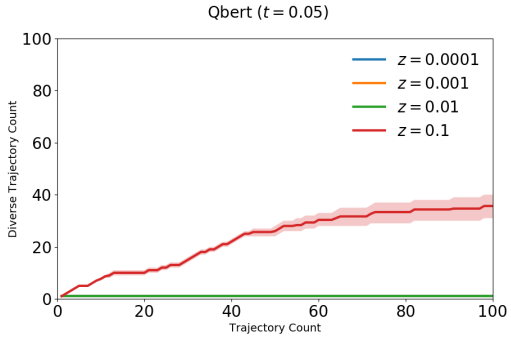
(b) Breakout



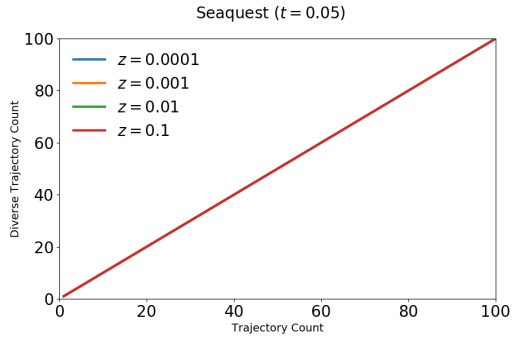
(c) MsPacman



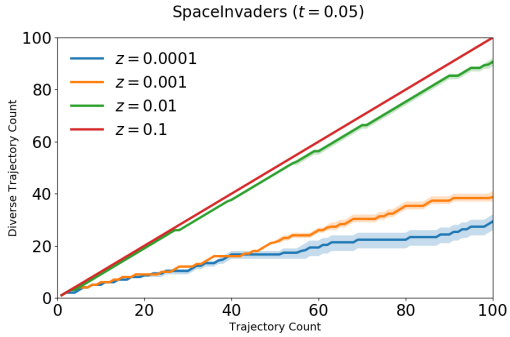
(d) Pong



(e) Qbert



(f) Seaquest



(g) SpaceInvaders

Figure 16: Diversity Measurement of IQN Models in Seven Atari Games ($t = 0.05$)

(a) Qbert: A 2D puzzle game where your goal is to change every cube in a pyramid to a target color. To do this, control the on-screen character, Q*bert, and make it jumps on top of the cube, avoiding obstacles and enemies.

(b) Seaquest: A 2D survival shooting game. The player sails a submarine to shoot at sharks and enemy submarines to rescue divers swimming in the water.

Figure 17: Game screens of Qbert and Seaquest.

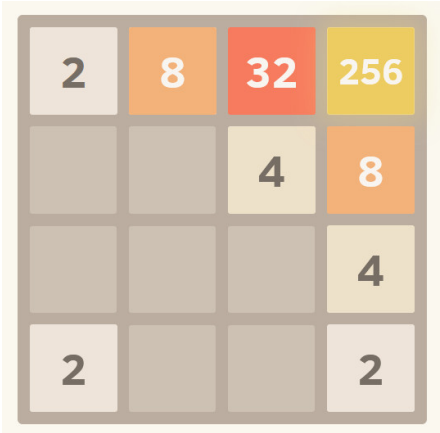


Figure 18: The game screen of the puzzle game 2048.

Table 17: Accuracy of Go player identification with the Playstyle Intersection Similarity.

First 8 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	3.5%	11.0%	15.0%	18.0%	22.5%	24.5%	25.5%
Candidate 5	6.5%	17.5%	22.0%	26.0%	35.0%	39.0%	41.0%
Candidate 10	14.5%	27.5%	41.5%	52.0%	58.0%	62.0%	68.0%
Candidate 25	14.5%	43.5%	54.0%	73.5%	82.5%	86.5%	89.0%
Candidate 50	20.0%	52.0%	64.0%	80.5%	88.5%	91.5%	93.5%
Candidate 75	18.0%	51.0%	68.0%	84.0%	89.5%	93.0%	92.0%
Candidate 100	18.5%	53.5%	68.5%	85.0%	91.0%	93.5%	93.0%
First 16 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	1.5%	12.5%	13.0%	16.0%	21.0%	23.0%	25.0%
Candidate 5	5.0%	10.5%	20.0%	22.5%	29.0%	33.0%	34.0%
Candidate 10	12.0%	23.5%	33.5%	39.5%	51.5%	55.0%	54.5%
Candidate 25	16.5%	42.0%	54.0%	67.5%	76.0%	82.0%	84.5%
Candidate 50	15.0%	50.0%	62.5%	74.5%	85.0%	89.0%	91.0%
Candidate 75	17.5%	52.5%	67.0%	79.5%	88.0%	89.5%	93.0%
Candidate 100	17.0%	51.5%	66.0%	80.0%	89.5%	91.5%	95.0%
First 32 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	1.5%	6.0%	11.0%	15.5%	21.5%	22.0%	22.5%
Candidate 5	4.0%	7.5%	15.0%	20.5%	22.5%	27.0%	31.0%
Candidate 10	12.0%	23.0%	30.0%	36.5%	42.5%	46.5%	48.0%
Candidate 25	16.0%	34.5%	46.5%	59.5%	73.0%	76.0%	78.5%
Candidate 50	16.0%	43.0%	59.0%	73.5%	81.5%	87.5%	90.5%
Candidate 75	16.5%	50.5%	60.0%	78.0%	87.5%	91.0%	93.0%
Candidate 100	16.5%	52.5%	62.5%	79.0%	88.5%	92.5%	94.0%

Table 18: Accuracy of Go player identification with the Playstyle Inter BC Similarity.

First 8 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	5.0%	9.0%	10.0%	14.5%	17.5%	19.5%	21.5%
Candidate 5	7.5%	19.0%	25.5%	32.5%	40.5%	45.0%	45.5%
Candidate 10	10.0%	25.0%	34.5%	48.5%	58.0%	62.0%	67.5%
Candidate 25	12.0%	41.0%	50.0%	65.0%	77.0%	84.5%	86.5%
Candidate 50	14.0%	45.0%	62.0%	76.5%	86.5%	91.0%	93.5%
Candidate 75	17.0%	44.0%	62.5%	80.5%	90.5%	91.5%	94.0%
Candidate 100	17.0%	50.5%	63.5%	82.0%	91.0%	94.0%	95.5%
First 16 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	2.0%	6.0%	6.0%	8.5%	12.5%	13.0%	14.0%
Candidate 5	4.0%	9.5%	15.5%	19.0%	25.0%	28.5%	28.0%
Candidate 10	7.0%	15.0%	21.0%	27.5%	37.5%	44.5%	47.0%
Candidate 25	11.5%	20.5%	34.5%	41.5%	54.0%	57.0%	63.5%
Candidate 50	12.0%	29.5%	42.5%	57.5%	67.5%	78.0%	79.5%
Candidate 75	15.0%	35.5%	48.0%	58.5%	73.5%	80.5%	87.0%
Candidate 100	14.5%	42.0%	50.0%	62.0%	79.0%	86.5%	90.0%
First 32 Moves	Query 1	Query 5	Query 10	Query 25	Query 50	Query 75	Query 100
Candidate 1	2.0%	3.0%	3.0%	5.5%	7.0%	8.0%	11.0%
Candidate 5	5.0%	6.0%	6.5%	9.5%	10.0%	13.5%	16.0%
Candidate 10	5.5%	9.5%	11.5%	17.0%	21.0%	18.5%	21.0%
Candidate 25	8.0%	16.5%	19.0%	23.5%	35.0%	33.0%	37.0%
Candidate 50	10.0%	22.5%	27.5%	37.0%	40.5%	49.0%	50.5%
Candidate 75	13.5%	24.5%	34.0%	37.5%	45.0%	51.0%	55.5%
Candidate 100	18.0%	28.5%	37.0%	42.0%	51.5%	56.5%	62.0%