

This is a repository copy of *Player Style Clustering without Game Variables*.

White Rose Research Online URL for this paper:
<https://eprints.whiterose.ac.uk/162213/>

Version: Accepted Version

Proceedings Paper:

Ferguson, Mark, Devlin, Sam, Kudenko, Daniel orcid.org/0000-0003-3359-3255 et al. (1 more author) (Accepted: 2020) *Player Style Clustering without Game Variables*. In: *Proceedings of the International Conference on the Foundations of Digital Games (FDG) 2020*. ACM . (In Press)

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:
<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Player Style Clustering without Game Variables

MARK FERGUSON, University of York, UK

SAM DEVLIN, Microsoft Research, UK

DANIEL KUDENKO, Leibniz University, Germany

JAMES ALFRED WALKER, University of York, UK

Player clustering when applied to the field of video games has several potential applications. For example, the evaluation of the composition of a player base or the generation of AI agents with identified playing styles. These agents can then be used for either the testing of new game content or used directly to enhance a player’s gaming experience. Most current player clustering techniques focus on the use of internal game variables. This raises two main issues: (1) the availability of game variables, as source code access is required to log them and hence limits the data sources that can be used, and (2) the choice of game variables can introduce unintended bias in the types of play style extracted. In this work, a hybrid unsupervised frame encoder and a ‘reference-based’ clustering algorithm are both proposed and combined to allow clustering from raw game play videos. It is shown that the proposed methods are most beneficial when the types of play styles are unknown.

CCS Concepts: • **Information systems** → **Clustering**; • **Applied computing** → **Computer games**.

Additional Key Words and Phrases: Player Clustering, Game Analytics, Representation Learning

ACM Reference Format:

Mark Ferguson, Sam Devlin, Daniel Kudenko, and James Alfred Walker. 2020. Player Style Clustering without Game Variables. In *International Conference on the Foundations of Digital Games (FDG '20)*, September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3402942.3402960>

1 INTRODUCTION

The ability to elicit play styles from player data can be key to identifying issues within games. For example, if it was discovered that one play style was disproportionately used, this could highlight a possible balancing issue of the game mechanics. This information can then be used to improve the player experience [4].

In addition to understanding the player base, AI agents can be generated based on extracted play styles. These AI agents can be used to automate new level testing [11], [7] or be implemented directly into the game with the aim of improving the player experience. Further, the play styles extracted could be used to expand the player persona set, as discussed by Guerrero-Romero *et al.* [8].

To date, clustering over manually selected game variables are commonly used to discover play styles within player data. This has a number of potential problems: (1) manually selecting game variables is time consuming (2) the use of manually selected game variables may introduce an unintended bias in the types of play styles extracted, and (3) the clustering hyperparameters need to be tuned for different games.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2020 Copyright held by the owner/author(s).

Manuscript submitted to ACM

We tackle these issues with the following contributions: (1) We propose a hybrid encoder architecture, that aims to encode raw frames into representations that contain both visual frame information and high level game variables. This removes the need to manually select game variables and allows the use of other data sources, such as ‘streams’ and ‘lets play’ videos. (2) We propose a novel referenced based clustering technique. It can be applied with a fixed set of hyperparameters and hence avoids the need for tuning. We demonstrate the benefits in two game environments.

The two contributions can be combined to allow the use of visual game play information (e.g. recorded game videos) to extract player styles. To evaluate this, a mixture of AI agent and human data was generated for different play styles to enable the use of ground truth labels. These have been used to show that the proposed techniques are especially useful when the type of play styles are unknown

2 RELATED WORK

2.1 Clustering

The use of clustering to extract information from a dataset is commonly used in many scientific fields. This in turn has led to a large number of clustering techniques being developed, a number of which were discussed by Xu and Tian [16] including but not limited to methods based on: ‘partition’, ‘hierarchy’ and ‘distribution’.

Drachen et al. [5] tested common clustering algorithms on game variables from ‘Destiny’. The game variables used included ‘proportion of kills with each gun’, hence play styles extracted could often be described based on ‘weapon choice’. This reinforces that domain knowledge is required to select suitable game variables to extract all existing play styles. However, it is recognised that chosen game variables could be transferred between games of the same genre.

Further, Hayes and Beling [9] experimented with hierarchical based clustering to find ‘build order’ strategies in ‘StarCraft : Brood War’. In this case, analysis was applied to the ‘military composition’ of each player, which is a vector of the total amount of different units and buildings a player has at the end of a time step. The extracted play styles often favoured certain units or buildings. Similarly, this technique could be transferred to any game with a ‘build order’.

An alternative approach to the unsupervised clustering techniques previously discussed, is to train a classifier to identify strategies or player roles, as described by Eggert et al. [6] within ‘Dota 2’. Such a technique not only requires extraction of features such as ‘lane’, ‘damage type’, etc, but also a labelled dataset of strategies. This is not only commonly unfeasible in new games, but also limits the possible strategies to those defined by an expert.

Feature extraction of previous work often resulted in non-time-series data, hence standard clustering can be applied. However, when considering sequences of frame encodings, time series clustering was required. Aghabozorgi et al. [1] surveyed the current techniques, in which three main types of clustering were identified. From these, ‘whole time-series clustering’ was deemed most suitable as this allows a direct comparison between two sequences, whereas the other techniques look at clustering subsequences.

‘Whole time-series clustering’ can be broken down to either ‘shape’, ‘feature’ and ‘model’ based. The latter two are not suitable in this work, due to the necessity to transform the data to a feature vector or parameterised model respectively. Whereas ‘shape’ based works on the raw data, by matching the shape of the two time series. It would be beneficial for the chosen technique to be robust to temporal shift so that players operating the same strategy on different time scales are considered similar. Due to this Dynamic Time Warping (DTW) [3] was used. This involves the stretching or compression of the first time series in order to find its optimal matching pattern with the second.

2.2 State Representation Learning

As stated previously, the aim of this work is to cluster play styles without the use of game variables, based solely on the game screen the player is observing. However, issues arise when comparing the raw frames directly. The reason for this is that simple transformations, for example image shift or rotations, while having no effect on the play style, can result in dissimilarity between previously similar frames and when compounded over the sequence of frames can cause matching issues. Any frame processing solution must be performed in an unsupervised manner or it would nullify the previous benefit of removing the need for human choice. Hence the idea of unsupervised state representation learning can be used to encode frames into a more robust intermediate representation.

The choice of state representation is key to the success of any downstream task, whether that be classification, clustering or reinforcement learning. In order to remove the need for domain knowledge for the choice of a representation, several techniques have been developed in order to generate these representations in an unsupervised manner. Lesort et al. [13] identified four types of approach to train these encoders: ‘Observation Reconstruction’, ‘Forward Model’, ‘Reverse Model’ and ‘Prior Knowledge Constraints’.

For ‘Observation Reconstruction’ the loss term is defined as the difference between the true observation and the recreated observation. An example of this approach is the Autoencoder [10] where a decoder is trained, in parallel to the encoder, to recreate the observation from the representation. The method was further developed by Kingma and Welling [12] into a Variational Autoencoder (VAE), by adding a regulating term relating to the KL divergence.

For the ‘Forward Model’ approach, the aim is to predict the next observation. The Action-Conditional Video Prediction [14] technique attempts to do this based on either solely the current representation or combined with the player’s action.

As the ‘Reverse Model’ approach requires the player’s action, it is expected to be least useful as access to these is not guaranteed.

For the ‘Prior Knowledge Constraints’ approach, objective or loss functions are defined to be maximised or minimised respectively. An example of this is Spatiotemporal DeepInfomax (ST-DIM) [2], where two mutual information objective functions are defined. Firstly, between sequential local features in the same location and secondly, between the global features and next frame’s local features

3 PROPOSED APPROACHES

3.1 Proposed Reference Based Clustering

This method was developed following an observation regarding the distance matrix generated from initial experiments. Although the direct distance between points in the same cluster was not small, the distance from these points to other points were similar.

The proposed algorithm goes through four stages. Firstly, a gradient matrix is generated based on the distance matrix. The gradient value when comparing points x and y , in a set of D points, is equal to the m value that minimises $\sum_{z \in Z} ((m \times dis_{x,z} + c) - dis_{y,z})^2$ where $Z = \{1, 2, \dots, D\} - \{x, y\}$. However if x and y are equal, then the value is set to zero.

A graph is then generated with a node per data point. After which an edge is created from each node $N_i \in Nodes$ to node N_j where $j = \text{argmax}(grad[i])$.

For each subgraph a path is traced between the two highest degree nodes. If both nodes have a degree greater than 3 and the path has an edge either greater than 1.3 or less than 0.7, then that edge is removed. However a maximum of

one edge is removed with a higher edge taking precedence over a lower one. This is then applied recursively to any generated subgraphs.

Finally pairs of subgraphs are considered and if the average distance is higher than 0.8, they are merged. This is repeated until no further merging occurs.

3.2 Proposed Hybrid Encoder

In this work consideration was given to the type of feature that would be useful to cluster play styles, as this dictates the encoder approach that should be used. It is suggested that a combination of purely visual information and high level game variables are required to fully encapsulate all play styles.

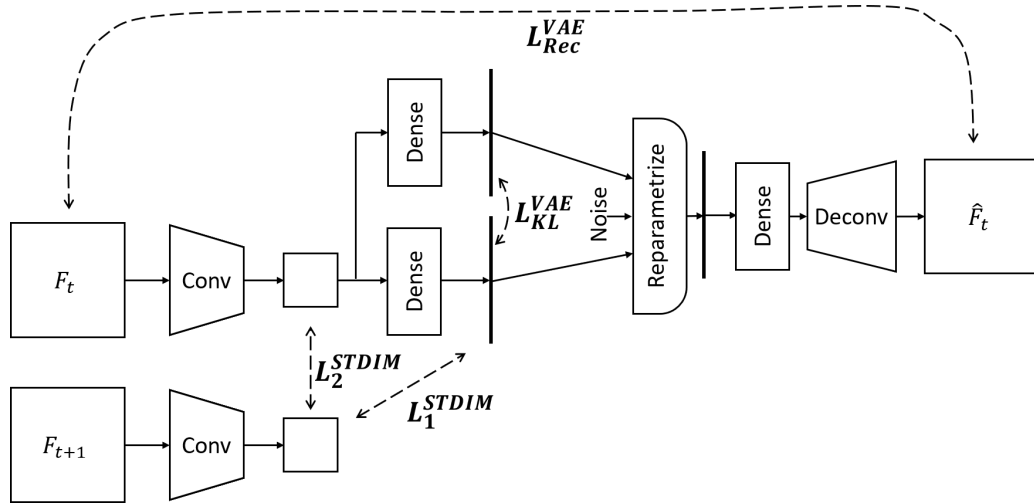


Fig. 1. Proposed STDIM-VAE network architecture

This lead to the proposed hybrid STDIM-VAE encoder shown in Figure 1. The VAE section is based on Kingma and Welling’s [12] previous work and aims to encode the visual information. Whereas the ST-DIM section is based on Anand et al.’s [2] previous work and aims to encode the high level game variables. The losses from each section are combined but adjusted so each component returns values within similar ranges, so no one component is dominant.

4 EXPERIMENTAL SETUP

To evaluate the proposed techniques, the ‘Black Smoke’ and ‘VizDoom’ environments were chosen. These allowed experiments on both 2D fully observable and 3D partially observable environments. In addition, it was chosen to use KMeans, Gaussian Mixture Models (GMMs) and Hierarchical clustering algorithms as a set of baselines.

In ‘Black Smoke’, part of the GVGAI Framework [15], the player must navigate to the exit while being chased by ‘smoke’ that damages the player. Demonstrations of two play styles were generated based on previously defined personas [11]. The ‘*Runner*’ aims to get to the exit, whilst the ‘*Completionist*’ aims to break blocks en route to the exit. These were artificially generated using a two-step look ahead agent with added random noise. The agent’s ‘position’ and running total of ‘blocks broken’ are also recorded.

For ‘VizDoom’ two features were chosen to define play styles, namely ‘weapon choice’ and ‘player position’. Weapon choice simply dictated that when possible, a weapon should be sought and used. Whereas player location dictated if a player should traverse a given route (engaging with enemies encountered) or camp in a given location. Figure 2 shows a breakdown of the properties for each play style and how they have been grouped into subsets. The data was generated by the lead author playing in each play style.

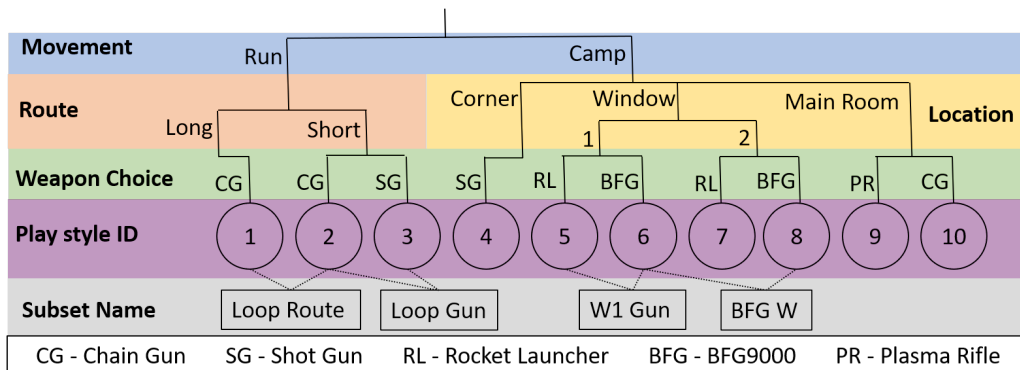


Fig. 2. Play Styles and Subsets

As ‘VizDoom’ is of the same genre as previous clustering research [5], where suitable game variables were transferred between the two games. This allows a direct comparison to previous literature using the same game variables.

For each environment, 5 encoders were trained, each with the Adam optimiser with a learning rate of $5e^{-4}$ and a batch size of 64. Although, ‘VizDoom’ required a frame down sampling of 5 and a frame limit of 400. The training times can be seen in Table 1.

Table 1. Mean Encoder Training Times (mins \pm SD)

Encoder	Black Smoke	VizDoom
VAE	37.31 \pm 00.10	28.16 \pm 03.98
STDIM	172.83 \pm 35.80	142.54 \pm 22.59
STDIM-VAE	216.55 \pm 26.24	17.82 \pm 28.89

Throughout, ground truth labels were stored and Adjusted Mutual Information (AMI) was used to evaluate cluster quality.

5 RESULTS AND DISCUSSION

For ‘Black Smoke’ clustering was applied to two datasets each with a different noise level. It was found the ‘Blocks Broken’ game variable performed best and hence was compared to the frame encoding techniques in Figure 3.

Evaluations conducted on the time series sequences of variables, ‘X’ and ‘Y’, performed reasonably well in contrast to a poorer performance when using only the final value. Additional improvements were seen when combining the time series of these two variables.

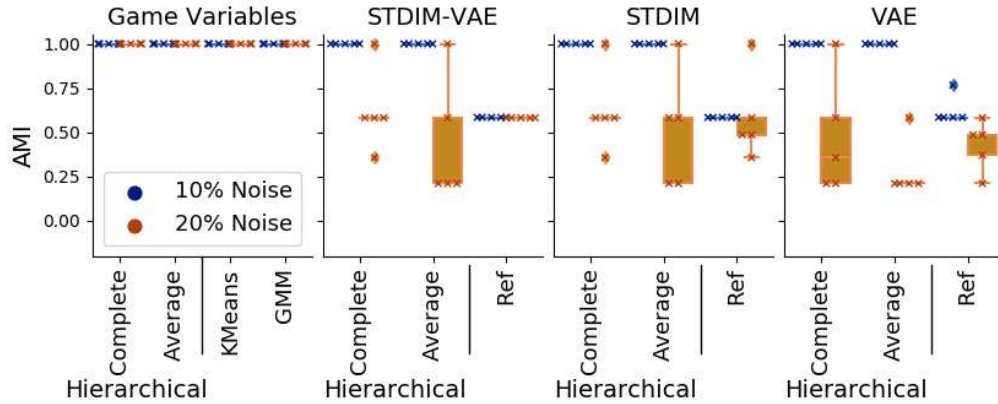


Fig. 3. Comparing Clustering Approaches on Black Smoke

When examining the clusters produced by the encoding methods, it was found for low noise levels, the performances were comparable. It was found ‘reference-based’ clustering would over cluster but the clusters were still pure. However, when tested on a higher noise level dataset, the encoding techniques performed worse.

Next experiments were conducted on ‘VizDoom’, in this case the ‘proportion of kills with each weapon’ game variables was used as a baseline, the results are shown in Figure 4. It was noted, combining additional variables resulted in a noticeable drop in performance.

The first set of experiments, shown in Figure 4a, focused on play styles based on weapon choice. This choice was made as previous literature [5] found this often was a defining feature of a play style. The results reinforced that clustering based on game variables performs well, if the play styles are well encapsulated by the selected variables. The encoding techniques often perform slightly worse on the same dataset. However, this is acceptable given the removal of the need to choose game variables.

The second set of experiments switched focus to play styles based on player position, the results can be seen in Figure 4b. This explored the introduction of unanticipated play styles within the data. The results show a significant reduction in performance when using game variables, which further supports that clustering is highly dependent on selecting suitable variables. However, better performance was achieved when using an encoding, which suggests that they are beneficial when unknown play styles exist in the data.

On certain subsets, a high variance in performance is observed. This could be attributed to either the randomness in network initialisation and data sampling, or that each training run learns different features. Further experiments are required for validation.

The final experiment used all previously defined play styles, the results can be seen in Figure 4c. In this case, the encoding methods outperform the use of game variables. This is due to the fact that the encoding methods have been shown to split movement-based play styles better. This is further validated, when using gun based ground truth, the game variables perform well. This shows the game variable clusters are split based on weapon alone. This suggests poor game variable selection could hide the existence of additional play styles.

6 CONCLUSION AND FUTURE WORK

The main purpose of this work was to enable play style clustering based on visual information alone when game variables are not accessible. This was achieved by the use of a hybrid frame encoder, that learns representations that contain both high level game variables and visual information, and a clustering approach that does not require prior knowledge as to how many play styles exist within the data.

The proposed methods have been combined and compared, to player clustering on expert selected game variables in two different game environments. When access to game variables is possible and the types of play styles are known, suitable game variables that fully capture play styles can be chosen. Under these conditions the use of game variables is the best option. These assumptions put restrictions on the type of data sources can be used and could introduce an unintended bias in the play styles clustered based on the game variables chosen. Therefore, if the data in question breaks either of the two requirements mentioned above the proposed method of clustering based on encoding sequence is more beneficial. This is shown by an AMI score 0.868 compared to 0.603 on a dataset containing both expected and unexpected play styles.

There are several avenues that could be explored to build on this work. The first would be to include experiments on additional game environments with a greater range of play styles. Further, it would be interesting to explore if the proposed methods can determine team-based strategies rather than individual behaviours.

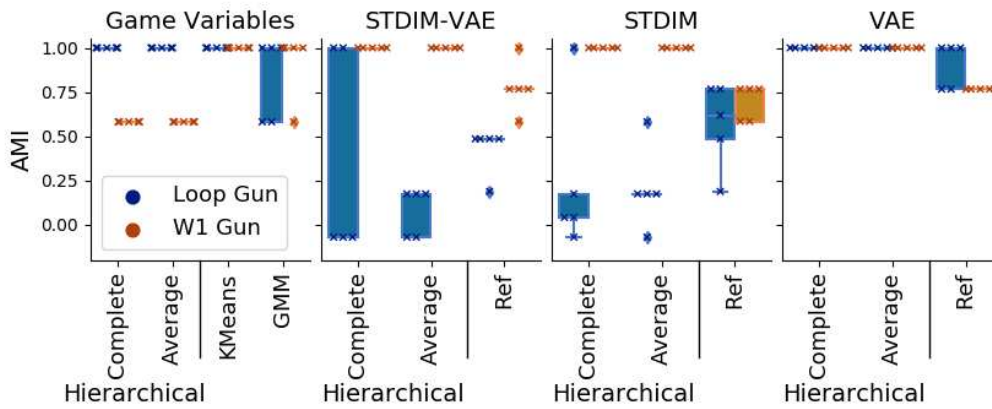
ACKNOWLEDGMENTS

This work was supported by Microsoft Research PhD Scholarship Programme and the Digital Creativity Labs funded by EPSRC/AHRC/Innovate UK (EP/M023265/1).

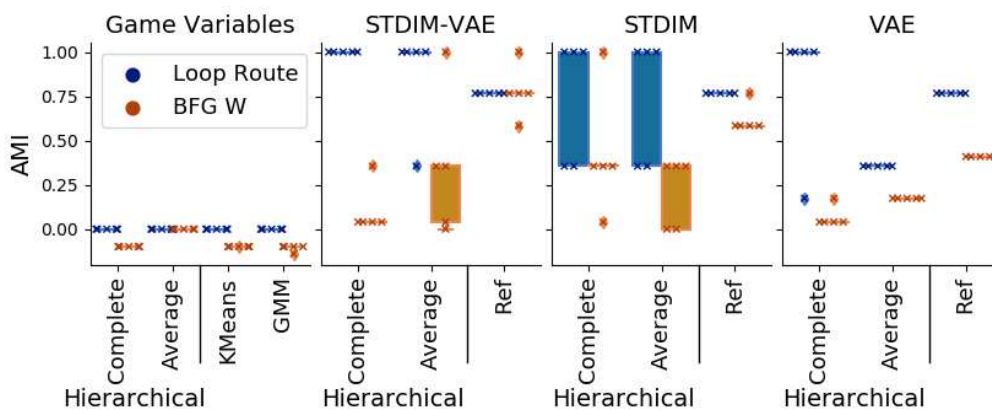
REFERENCES

- [1] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering—A decade review. *Information Systems* 53 (2015), 16–38.
- [2] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. 2019. Unsupervised State Representation Learning in Atari. *arXiv preprint arXiv:1906.08226* (2019).
- [3] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, 359–370.
- [4] Alessandro Canossa, Magy Seif El-Nasr, and Anders Drachen. 2013. Benefits of game analytics: Stakeholders, contexts and domains. In *game analytics*. Springer, 41–52.
- [5] Anders Drachen, James Green, Chester Gray, Elie Harik, Patty Lu, Rafet Sifa, and Diego Klabjan. 2016. Guns and guardians: Comparative cluster analysis and behavioral profiling in destiny. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.
- [6] Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. 2015. Classification of player roles in the team-based multi-player game dota 2. In *International Conference on Entertainment Computing*. Springer, 112–125.
- [7] Cristina Guerrero-Romero, Annie Louis, and Diego Perez-Liebana. 2017. Beyond playing to win: Diversifying heuristics for VGGAI. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 118–125.
- [8] Cristina Guerrero-Romero, Simon M Lucas, and Diego Perez-Liebana. 2018. Using a Team of General AI Algorithms to Assist Game Design and Testing. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.
- [9] Roy Hayes and Peter Beling. 2018. Unsupervised Hierarchical Clustering of Build Orders in a Real-Time Strategy Game. *The Computer Games Journal* 7, 1 (2018), 5–26.
- [10] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.
- [11] Christoffer Holmgård, Michael Cerny Green, Antonios Liapis, and Julian Togelius. 2018. Automated playtesting with procedural personas through MCTS with evolved heuristics. *arXiv preprint arXiv:1802.06881* (2018).
- [12] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [13] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-François Goudou, and David Filliat. 2018. State representation learning for control: An overview. *Neural Networks* 108 (2018), 379–392.

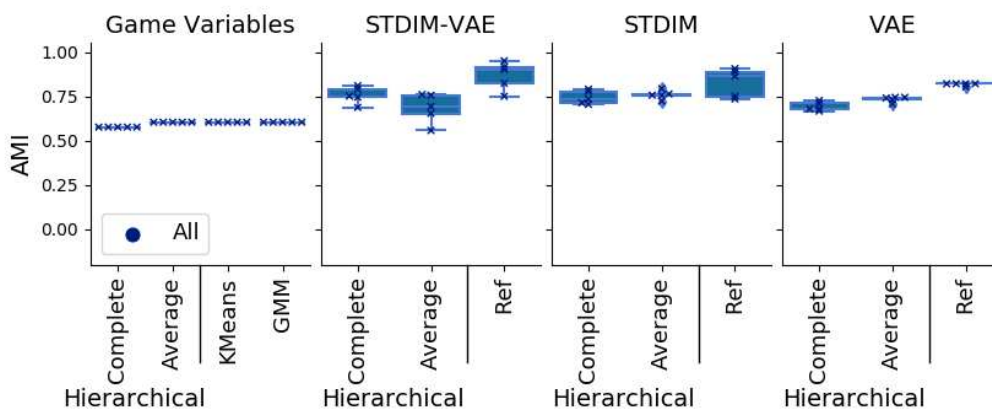
- [14] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. 2015. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*. 2863–2871.
- [15] Ruben Rodriguez Torrado, Philip Bontrager, Julian Togelius, Jialin Liu, and Diego Perez-Liebana. 2018. Deep Reinforcement Learning for General Video Game AI. In *Computational Intelligence and Games (CIG), 2018 IEEE Conference on*. IEEE.
- [16] Dongkuan Xu and Yingjie Tian. 2015. A comprehensive survey of clustering algorithms. *Annals of Data Science* 2, 2 (2015), 165–193.



(a) Gun Based Play Styles



(b) Movement Based Play Styles



(c) All Play Styles

Fig. 4. Comparing Clustering Approaches on VizDoom