Towards General Game Representations: Decomposing Games Pixels into Content and Style

Chintan Trivedi
ctriv01@um.edu.mt
Konstantinos Makantasis
konstantinos.makantasis@um.edu.mt
Antonios Liapis
antonios.liapis@um.edu.mt
Georgios N. Yannakakis
georgios.yannakakis@um.edu.mt

Institute of Digital Games,
University of Malta

Abstract

Learning pixel representations of games can benefit artificial intelligence across several downstream tasks including game-playing agents, procedural content generation, and player modeling. However, the generalizability of these methods remains a challenge, as learned representations should ideally be shared across games with similar game mechanics. This could allow, for instance, game-playing agents trained on one game to perform well in similar games with no re-training. This paper explores how generalizable pre-trained computer vision encoders can be used for such tasks by decomposing the latent space into content and style embeddings. The goal is to minimize the domain gap between games of the same genre when it comes to game content and ignore differences in graphical style. We employ a pre-trained Vision Transformer encoder and a decomposition technique based on game genres to obtain separate content and style embeddings. Our findings show that the decomposed embeddings achieve style invariance across multiple games while still maintaining strong content extraction capabilities. We argue that the proposed decomposition of content and style offers better generalization across game environments independently of the downstream task.

1 Introduction

Video game engines maintain an internal data representation of the game world capturing all necessary underlying variables describing the entities and objects existing within the virtual world [13, 18] such as the position of the player and the location of the opponent. When this data representation is fed through the *renderer* for generating the output image shown to the player it incorporates the visual styling of the game—defined in terms of textures and colors [21]. As a result, the visual *style* and the game *content*, represented as the internal game state, get entangled [34] onto a high dimensional space (i.e. the RGB pixels).

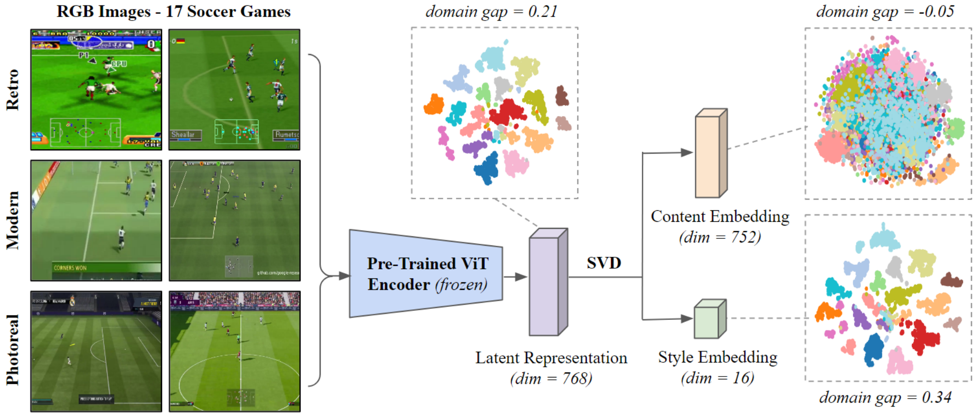
RGB Images - 17 Soccer Games
Retro
Modern
Photoreal
Pre-Trained ViT Encoder (frozen)
Latent Representation (dim = 768)
SVD
Content Embedding (dim = 752)
Style Embedding (dim = 16)
domain gap = 0.21
domain gap = -0.05
domain gap = 0.34

Figure 1: The proposed framework uses multiple games from the same genre to transform latent representations to two lower-dimensional embeddings: one for *content* and one for *style*. The t-SNE plot visualizes the domain gap in the latent representations and shows how these differences are flattened into a separate style subspace which filters out the style gap.

When artificial intelligence (AI) methods are applied to games [33], such internal data representations (e.g. the player's position) can be extracted directly from variables within the game engine [2, 15, 16, 22]. However, access to the game engine is a rarity; this severely limits the number of games—especially commercial-standard ones—available for game AI research. Pre-trained (and fine-tuned) computer vision models employed for extracting latent representations from game pixels are not generalizable and suffer from what is known as the *domain gap* problem [30] (see Fig. 1). Graphical styling differences in games—arising from varying color palettes and abstract object designs—may cause a shift in the underlying distributions of latent representations as extracted from pre-trained models [24]. Motivated by the lack of generic approaches that identify critical game state information (i.e. game content) solely from game frames, this paper introduces a method that processes the RGB pixels of the game footage and disentangles the game content from the game's style without relying on the game engine [25]. The introduced method learns to decompose content and style in a *generalizable* zero-shot manner [31] for any game, thereby eliminating the technical challenges [11] associated with fine-tuning neural network models.

We propose a latent decomposition technique on the latent space of a pre-trained Vision Transformer [4] model, trained on the Imagenet dataset [20] using the DINO self-supervised learning method [3]. We show that it is possible to recover style and content embeddings from this latent space extracted from the game pixels without further training. Our primary contribution involves using a simple statistical method (i.e. singular value decomposition) to find latent directions [5, 17, 27] that are unique across different games of the same genre, and define them as *style*, while the remaining latent directions that are shared across these games are defined as *content*. To validate our hypothesis, we test our method on two datasets. The Gen11 dataset introduced here contains 110k images from 11 game genres and 193 games that vary in style. The 3D-SSL dataset [26] contains game frames and corresponding internal game state information (e.g. player position) from 3 games of 3 difference genres. We conduct linear probing tasks [1] on the derived content and style embeddings to recover relevant game state information (i.e. content) and graphical style of the game (e.g. retro, modern, photorealistic). Our findings suggest that by using the introduced method we can recover information that exhibits style-invariance while maintaining efficient content extraction capacities in a generalizable fashion across game genres.

2 Background

Only a few indicative papers focus on generalized representation learning within the domain of video games. Trivedi *et al.* [26] provide preliminary results on learning game state representations with self-supervised learning (SSL) methods. They show that SSL models are superior to pre-trained ImageNet models for 3D games. In [24] Trivedi *et al.* train generalizable game models on 100k images across 175 games (10 game genres) and their findings suggest that contrastive learning is better for learning game representations compared to conventional supervised learning. Lee *et al.* [12] propose multi-game decision transformers able to generalize across different Atari games after training on data from several such games.

The application of self-attention [28] to computer vision has enabled many advancements that derive from the inductive biases [32] of the model architecture [4]. Caron *et al.* [3] observe that Vision Transformers, when paired with SSL methods, generalise better compared to convolutional networks. They note that via self-attention their ViT architecture can preserve spatial information in the images when compressing dimensionality for representation learning. Khan *et al.* [8] build upon this finding and showcase that such pre-trained encoders are sufficient as backbone models to deploy in games. Their study, however, only supports that pre-trained models are sufficient for training agents on a particular game without exploring whether those agents can also be transferred to other games in a zero-shot manner.

Computer vision studies that investigate image style and content disentanglement [7, 10] are largely focused on the image style transfer paradigm [6]. When it comes to games, style and content separation is possible [23]. Kim *et al.* [9] proposed to learn style and content representations in a car driving simulator, defining content as spatial information and everything else as style. This style-content disentanglement [14] cannot be applied to multiple games (even of the same genre) as we must cater for the game-specific differences in design choices and graphical fidelity of physical objects, especially those belonging to different game generations (e.g. low-bit retro games versus highly detailed modern games).

In contrast to all aforementioned studies, this paper aims to minimize the domain gap across games of the same genre. For that purpose, we propose a novel decomposition technique on the latent representations of pre-trained encoder models by projecting them onto separate content and style subspaces. The method appears to be generic across different game genres, provided sufficient and representative data samples of game images.

3 Style-Content Decomposition: Definitions

Kim *et al.* [9] define *style* as information that does not depend on pixel location and *content* as information that does. Within the car driving simulator used in their study, an example of style information would be weather condition while that of content would be placement of objects in the rendered image of the scene. Inspired by these definitions, we refer to *style* as the aesthetic design choice which influences visuals of the game, but has no bearing on the elements of the game world related to gameplay dynamics. *Content* instead can be viewed as elements of the game that influence the game-state existing within the predefined rules of the game world. These rules are often shared across games belonging to the same genre.

In contrast to [9], we work on multiple games from multiple genres and thus must formalize what core elements of gameplay should be considered for all these games. For this, we follow the causal dependency framework introduced by Trivedi *et al.* [24] that utilizes the broader accepted notion of "game genres" to create a separation between what counts as game content and what gets interpreted as style. Under this framework, the game genre (e.g. shooter genre versus car racing genre) only causes the content of the game to change, but not vice-versa. We hypothesize that a game may have an idiosyncratic visual style (e.g. skybox or wall textures). The content, however, is specified by conventions of the game genre: e.g. the types of game elements (ball, net), their visual representations (shape of a sniper rifle), their interactions (a soccer player may possess a ball by keeping it close), etc. In essence, our causal framework works under the assumption that different games from the same genre share the same content space but not necessarily the same style. An alternative way of looking at the differences between content and style within a game genre is their degree of subjectivity: the art style of a game can be viewed primarily as a subjective notion whereas content can be defined objectively.

Based on our definition of style and content, we highlight the challenge of using pre-trained computer vision models for processing game pixels. We define *domain gap* based on the framework defined earlier and quantify it as the influence of style in learning representations of different games belonging to the same genre. We therefore measure the domain gap in this paper based on the clustering quality metric *silhouette score* [19, 24], ranging between -1 and $+1$, which quantifies how clustered the representations of graphically different games of the same genre are in the embedding space. The higher the silhouette score, the larger the gap between clusters of representations of different games. When capturing content information that is assumed to share same information across graphically different games, we want this silhouette metric to be low: values close to 0 indicate no clear clusters. At the same time, the style information across different games should ideally exhibit high silhouette score values: values close to 1 indicate very well separated clusters.

Our hypothesis is that different games belonging to the same genre have the same content. This would mean that the latent representation of images of different games from the same genre would contain similar content. This would decrease the variance in the representations across games, and we can leverage this to disentangle style and content information from within the latent space of a pre-trained encoder.

4 Methodology

Figure 1 illustrates our overall methodology. We employ a pre-trained vision encoder on the images to obtain a latent representation for each frame. Once the game footage frames from a number of different games (of the same genre) are encoded, they display a certain degree of domain gap expressed through the silhouette score of the latent embedding (as in [26]). The latent representation is then decomposed into content and style via the use of the Singular Value Decomposition (SVD) method.

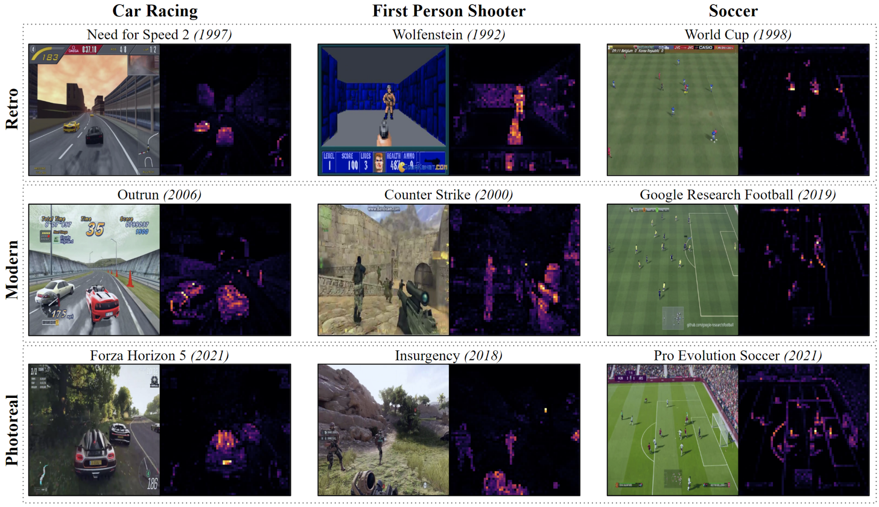

Figure 2: Visualizing the Spatial Attention Maps of DINO ViT-Base [3] across different game genres and graphical styles. The maps highlight the capability of self-attention layers [4] to extract content (e.g. car, enemy or player positions) while being invariant to the different game styles. The styles of retro, modern and photorealistic are shown in this example.

4.1 Computer Vision Encoder

For all experiments, we use a pre-trained Vision Transformer (ViT) [4] to extract latent representations of the game images. We use the ViT base encoder ($\sim$ 85M parameters) with patch size 8 and pre-trained on ImageNet dataset [20] employing the self-supervised DINO loss [3]. Fig. 2 presents the spatial self-attention maps used by the ViT architecture for 9 indicative games of our dataset. In particular, we obtain the attention activations from multi-head attention of the last transformer block, and average them across all attention heads [3]. This yields a 28x28 attention map, which we flatten into an attention representation of dimension 784. The attention maps depicted in Fig. 2 showcase the capability of self-attention layers to highlight a game's content while being invariant to the various graphical game styles.

4.2 Decomposing Game Content and Style via SVD

As mentioned earlier, the styles of different games belonging to the same genre can be substantially different. In contrast, their content—given that it reflects the game state—should not exhibit highly different patterns. Therefore, we hypothesize that the latent image representation factors that exhibit high variation across data points are associated with the game style, while those that do not significantly vary are associated with game content. To validate this hypothesis, we first identify those directions in the latent representation space that capture the most variation in the data and use them as a feature representation of game style; we call these direction *style components*. Then, we use the rest of the directions for representing content, which we call *content components*.

For that purpose, we use the SVD method [29]. Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{N \times P}$ representing a dataset consisting of N observations described by P features, SVD will compute the decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, where $\boldsymbol{U}$ is a unitary matrix, $\boldsymbol{S}$ is a diagonal matrix of singular

Table 1: Summary of the video game datasets used in this study.

Dataset Name	Gen11	3D-SSL
Game Genres	Football, Basketball, Bike Racing, Car Racing, Fighting, FPS, Hockey, Soccer, Table Tennis, Tennis, Volleyball	Car Racing (CARLA), FPS (VizDoom), Soccer (GRFE)
Total Games	193	3
Total Images	110,000	40,608
Labels	Graphics Styling (Retro, Modern, Photoreal)	Internal Game State (Positions of Player, Opponents and other Game Objects)
Usage	Latent Decomposition, Style Prediction	Content Extraction

values, and each of the columns of matrix $\boldsymbol{V}$ is associated with a *singular value*. The variance in the data that a column of $\boldsymbol{V}$ can explain is proportional to the square of its associated singular value. Based on that property, we can use the k columns of $\boldsymbol{V}$ associated with the k largest singular values to encode style and the remaining columns to encode content. The parameter k is selected empirically so that it maximizes the domain gap difference in games.

4.3 Datasets

Our proposed framework requires a specific dataset structure, where it is possible to categorize a game under a single game genre. For this purpose, we select the *Sports10* dataset introduced by Trivedi *et al.* [24], which contains gameplay images from 175 games across 10 sports game genres. In addition to Sports10 data, we consider data from the first-person shooter (FPS) genre, with $10,000$ images of 18 different FPS games in different graphic styles. The combined Gen11 dataset (as we name it here) contains 193 games across 11 game genres in total. In terms of graphical styling, it contains 58 games labeled as “retro”, 90 as “modern” and 45 as “photorealistic”. The properties of Gen11 data are provided in Table 1. We use this dataset for two purposes: (a) to quantify the domain gap across 11 different game genres and to learn a projection matrix for latent decomposition (see Section 3); (b) to assess our style embeddings using the three style labels (see Section 5.3).

In addition to Gen11, we use the 3D-SSL dataset [26] for evaluating content embeddings. This dataset contains paired images and game states obtained for three games by interfacing with the game engine to extract exact variables that describe the state of a game at any given instance. The properties of this dataset are also outlined in Table 1.

5 Results

This Section presents the decomposition analysis of the game genres considered (Section 5.1), and assesses the quality of content embeddings (Section 5.2) and style embeddings (Section 5.3) on simple linear probing tasks.

5.1 Decomposing Latent into Content and Style

To assess the decomposition method introduced in this paper, we apply SVD on the attention representation ($D = 768$) of each game genre (10k images) in the Gen11 dataset (see Section 4.3). The highest 256 singular values (out of 768) are shown in Fig. 3. We observe that

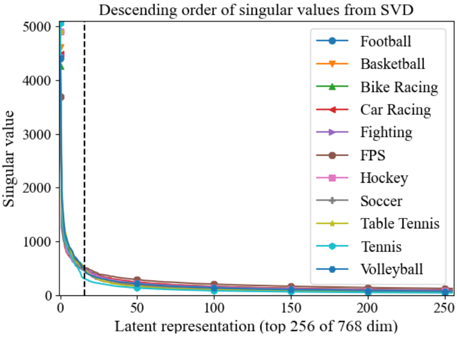

Figure 3: Top 256 singular values of SVD across all 768 dimensions of latent representation per genre in the Gen11 dataset.

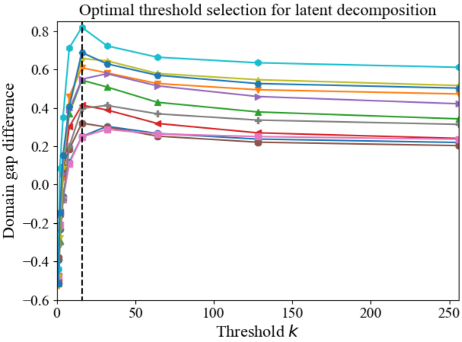

Figure 4: Domain gap difference across different k thresholds per genre in the Gen11 dataset.

most of the variance in the representations of different games within a genre comes from very few dimensions of the latent; this cut-off point (threshold k) is highlighted using the vertical dashed line ($k = 16$). The "flat" nature of this graph past the dashed line indicates that most information in the latent representations remains consistent across games despite having different graphical styling in its input. An encoder that is not robust to such variations would have a more gradual "decline" for the SVD values.

In order to assess the optimal value for the cut-off k, we use *domain gap difference* as a metric that consists of the silhouette score of the *style embeddings* (i.e. the k latent variables with the highest SVD score) minus the silhouette score of the *content embeddings* (i.e. the lowest $D-k$ latent variables according to the SVD score). Figure 4 shows this domain gap difference across different k thresholds. The optimal value for 9 out of 11 game genres is $k = 16$, with slightly higher domain gap differences at $k = 32$ for Football and Hockey genre. We select $k = 16$ as a general cutoff point across all game genres for consistency, yielding style embeddings of 16 dimensions and content embeddings of 752 dimensions.

In order to verify whether the embeddings derived above capture the notions of game *style* and game *content*, we run a linear probing analysis on both derived embeddings to predict elements of style versus features of content as detailed below.

5.2 Evaluating Content Embeddings

We follow the linear probing protocol introduced in [1] and run simple linear regression on game-state variables [26]. Indicative game-state variables for the three games of the 3D-SSL dataset [26] include the position of the players and the ball (94 variables for GRFE), the position of enemies (12 variables for VizDoom) and the position of the car and nearby traffic (7 variables for CARLA). We test four embeddings: (a) the latent representation of the pre-trained ViT, which represents the *upper bound* in terms of game state prediction potential for this experiment; (b) the latent representation of a randomly initialized ViT (RandViT) as a baseline to assess the impact of the training process; (c) the attention map of the pre-trained ViT, to assess whether features that only contain spatial information suffice for this task; (d) the content embeddings (dim=752) derived from our method. The performance of the linear probe is based on the coefficient of determination (R^2 score) of this model, averaged across all game state variables.

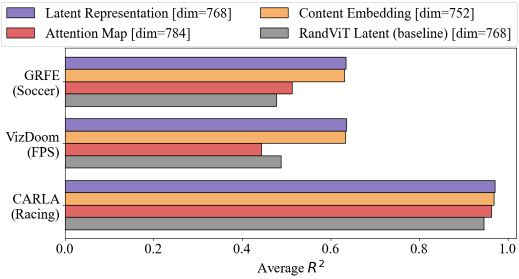

Figure 5: Linear probing results, measured as R^2 score averaged across all game state variables of the 3D-SSL dataset [26]. Results are averaged across all engine parameters of each game (94 for GRFE, 12 for VizDoom, 7 for CARLA).

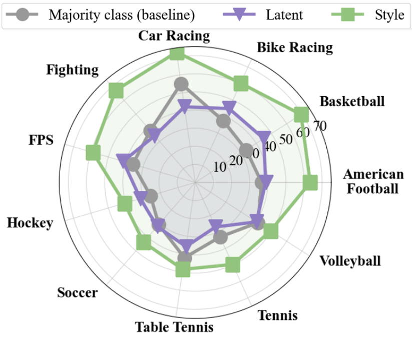

Figure 6: Test accuracy classifying a game's style (retro, modern, photorealistic) in the Gen11 dataset. Data is averaged from 10 cross-validation folds.

Table 2: Domain gap (as silhouette score) for different embeddings and game genres in the Gen11 dataset. Lower values indicate smaller domain gap.

	Football	Basketball	Bike Racing	Car Racing	Fighting	FPS	Hockey	Soccer	Table Tennis	Tennis	Volleyball
RandViT	0.147	0.201	0.093	-0.137	0.047	-0.144	0.030	0.353	0.317	0.626	0.167
Latent	0.138	0.290	0.236	0.161	0.218	0.104	0.140	0.213	0.340	0.441	0.276
Attention	-0.015	-0.012	0.021	0.056	-0.010	0.026	-0.096	-0.032	0.073	0.022	-0.007
Content	-0.010	-0.091	-0.048	-0.032	-0.153	-0.059	-0.030	-0.050	-0.133	-0.142	-0.199
Style	0.240	0.517	0.495	0.382	0.396	0.261	0.219	0.348	0.523	0.677	0.488

Figure 5 presents the key results of this experiment. As expected, the full latent representation of the pre-trained ViT reaches the highest R^2 scores. Moreover, the randomly initialized ViT performs poorly, with a relative decrease of 24%, 23% and 2.5% for GRFE, VizDoom, CARLA respectively. Evidently the CARLA dataset is fairly easy to predict, as even untrained models can perform relatively well. The attention layer underperforms as well, with a relative decrease of 19%, 30%, 0.8% for GRFE, VizDoom, CARLA respectively. While attention is good at detecting objects on the screen, it seems unable to distinguish between different object types which would require additional visual information. Finally, it is evident that content embeddings contain most of the information needed for these tasks, reaching R^2 scores very close to the upper bound (relative decrease from the full latent representation by 0.46%, 0.43% and 0.23% for GRFE, VizDoom, CARLA respectively).

Table 2 shows the silhouette scores for different embeddings per game genre in the Gen11 dataset. We note that content has low silhouette scores, i.e. exhibits a low domain gap between games of the same genre, lower than the full latent vector and (evidently) the style embeddings. Based on Fig. 5, these same content embeddings can predict critical game state information as well as the full latent vector. These two facts indicate that content embeddings do not exhibit differences based on the visuals of games within the same genre (of the Gen11 dataset) while also matching game engine parameters in games of the 3D-SSL dataset.

5.3 Evaluating Style Embeddings

In this section we test the capacity of the derived style embeddings to recover descriptive information about a game's graphic style in terms of three style labels: retro, modern and photorealistic graphics. We use a similar linear probing technique as in Section 5.2, but this time we learn to predict the graphics style class as a single-output classification task. The ground truth is in the manually ascribed style labels of the Gen11 dataset, which is used to train the classifier. We use a leave-three-games-out cross-validation scheme, with frames from three games of the same genre (one per style label) removed from training but used for testing. The Gen11 dataset is split as above randomly over 10 folds per genre, averaged and shown in Fig. 6. A majority vote classifier acts as a baseline, and is calculated on the training set in every fold.

For all 11 game genres in the Gen11 dataset, we notice that style embeddings are able to capture graphical styling information with higher accuracy than the baseline, with test accuracies as high as 72% (for the car racing genre). For comparison, we also test the capacity of the latent representations as input to this linear classifier. Despite the latent representations integrating style information, its classification accuracy is not able to reach that of the style embeddings; with fewer features (16), classification is easier for style embeddings than the full latent representation (768). The style prediction results highlighted in Fig. 6, combined with results in Table 2 showing higher silhouette scores for visual embeddings (i.e. spread out clusters even if the games all belong to the same genre), collectively provide strong evidence that the derived style embeddings are meaningful representations and efficient predictors of style across all game genres examined.

6 Discussion and Conclusion

Our findings provide evidence that only a few features of the latent representation of a pretrained Visual Transformer account for the style of a game, with remaining ones accounting for content information that can be re-used across other games in the same genre. By testing both style embeddings and content embeddings in fairly simple tasks (predicting graphical style and in-game variables respectively), we offer evidence that the split is a viable way forward for more complex downstream tasks. The clustering quality analysis also emphasizes the effectiveness of these embeddings in reducing the domain gap across different games of the same genre. Based on the experiments performed in this paper we can argue that the introduced method yields more generalizable and reusable game models, both for pixels to latent embeddings and for latent embeddings to pixels. We hope that the paper will spur further research in downstream applications with zero-shot transferability across games. Ultimately, how to best use the disentangled content and style embeddings for improving generalizability will vary according to the downstream task. Beyond the simple classifiers tested in this paper, in future studies we plan to test the the derived embeddings for a number of downstream tasks, including game-playing agents, player modeling and content generation.

7 Acknowledgments

This project has received funding from the EU's Horizon 2020 programme under grant agreement No 951911.

References

[1] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in Atari. In *Proc. of the 33rd Conference on Neural Information Processing Systems*, 2019.

[2] Matthew Barthet, Antonios Liapis, and Georgios N Yannakakis. Go-blend behavior and affect. In *Proc. of the IEEE Intl. Conf. on Affective Computing and Intelligent Interaction Workshops and Demos*, 2021.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision*, pages 9650–9660, 2021.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *Proc. of the 34th Conference on Neural Information Processing Systems*, 2020.

[6] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

[7] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. In *Proc. of the IEEE Winter Conference on Applications of Computer Vision*, pages 848–856, 2019.

[8] Mina Khan, P Srivatsa, Advait Rane, Shriram Chenniappa, Rishabh Anand, Sherjil Ozair, and Pattie Maes. Pretrained encoders are all you need. *arXiv: 2106.05139*, 2021.

[9] Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5820–5829, 2021.

[10] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Proc. of the IEEE/CVF international conference on computer vision*, pages 4422–4431, 2019.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

[12] Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. In *Proc. of the 36th Conference on Neural Information Processing Systems*, 2022.

[13] Michael Lewis and Jeffrey Jacobson. Game engines. *Communications of the ACM*, 45 (1):27, 2002.

[14] Xiao Liu, Spyridon Thermos, Gabriele Valvano, Agisilaos Chartsias, Alison O'Neil, and Sotirios A Tsaftaris. Measuring the biases and effectiveness of content-style disentanglement. In *Proc. of the British Machine Vision Conference*, 2021.

[15] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. The Arousal video Game AnnotatIoN (AGAIN) dataset. *IEEE Transactions on Affective Computing*, 13 (4), 2022.

[16] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris 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. *arXiv: 1912.06680*, 2019.

[17] Chi-Hieu Pham, Saïd Ladjal, and Alasdair Newson. Pca-ae: Principal component analysis autoencoder for organising the latent space of generative networks. *Journal of Mathematical Imaging and Vision*, 64(5):569–585, 2022.

[18] Steve Rabin. *Introduction to Game Development*. Charles River Media, Inc., 2005.

[19] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, et al. ImageNet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 2015.

[21] Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa. Video textures. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498, 2000.

[22] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv: 1712.01815*, 2017.

[23] Chintan Trivedi. Turning Fortnite into PUBG with deep learning (CycleGAN). https://towardsdatascience.com/turning-fortnite-into-pubg-with-deep-learning-cyclegan-2f9d3 2021. Accessed 8 May 2023.

[24] Chintan Trivedi, Antonios Liapis, and Georgios N Yannakakis. Contrastive learning of generalized game representations. In *Proc. of IEEE Conference on Games*, 2021.

[25] Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N Yannakakis. Game state learning via game scene augmentation. In *Proc. of the 17th International Conference on the Foundations of Digital Games*, 2022.

[26] Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N. Yannakakis. Learning task-independent game state representations from unlabeled images. In *Proc. of the IEEE Conference on Games*, 2022.

[27] Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. Warpedganspace: Finding non-linear rbf paths in gan latent space. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, pages 6393–6402, 2021.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of the 31st Conference on Neural Information Processing System*, 2017.

[29] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. *A practical approach to microarray data analysis*, pages 91–109, 2003.

[30] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312(C):135–153, 2018.

[31] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

[32] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. In *Proc. of the 35th Conference on Neural Information Processing System*, 2021.

[33] Georgios N Yannakakis and Julian Togelius. *Artificial intelligence and games*. Springer, 2018.

[34] Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 8447–8455, 2018.