Towards General Models of Player Experience: A Study Within Genres

David Melhart
Institute of Digital Games
University of Malta
Msida, Malta
david.melhart@um.edu.mt

Antonios Liapis
Institute of Digital Games
University of Malta
Msida, Malta
antonios.liapis@um.edu.mt

Georgios N. Yannakakis
Institute of Digital Games
University of Malta
Msida, Malta
georgios.yannakakis@um.edu.mt

***Abstract*—To which degree can abstract gameplay metrics capture the player experience in a general fashion within a game genre? In this comprehensive study we address this question across three different videogame genres: racing, shooter, and platformer games. Using high-level gameplay features that feed preference learning models we are able to predict arousal accurately across different games of the same genre in a large-scale dataset of over $1,000$ arousal-annotated play sessions. Our genre models predict changes in arousal with up to 74% accuracy on average across all genres and 86% in the best cases. We also examine the feature importance during the modelling process and find that time-related features largely contribute to the performance of both game and genre models. The prominence of these game-agnostic features show the importance of the temporal dynamics of the play experience in modelling, but also highlight some of the challenges for the future of general affect modelling in games and beyond.**

***Index Terms*—general modelling, player modelling, affective computing, preference learning, arousal**

I. Introduction

Artificial general intelligence and artificial psychology define two critical long-term goals of artificial intelligence (AI). The intersection of the two would enable artificial systems to perform affect-based interactions in general settings. While games (board or digital) define the dominant application area for the study of general AI, limited emphasis has been given to the ways general AI systems are possible in games beyond the task of gameplaying [1], [2], including systems that create or even model player experience in a general fashion [3]. Arguably, studying *general models of player experience*—which aim at predicting the experience of play in a game-independent way—is still in its infancy. The handful of examples in this vein are limited by ad-hoc game testbeds, and experience models that are built on small-scale corpora [4]–[6].

Motivated by the lack of a comprehensive study on general player experience modelling, this paper explores the degree to which player experience can be modelled across games of the same genre in a general fashion. We assume that there exist features of play that are able to transfer aspects of player experience across games of the same genre. We also assume that such features can be used to build accurate models of player experience in a general fashion within a genre.

This project has received funding from the EU's Horizon 2020 programme under grant agreement No 951911, and from the University of Malta internal research grants programme Research Excellence Fund under grant agreement No 202003.

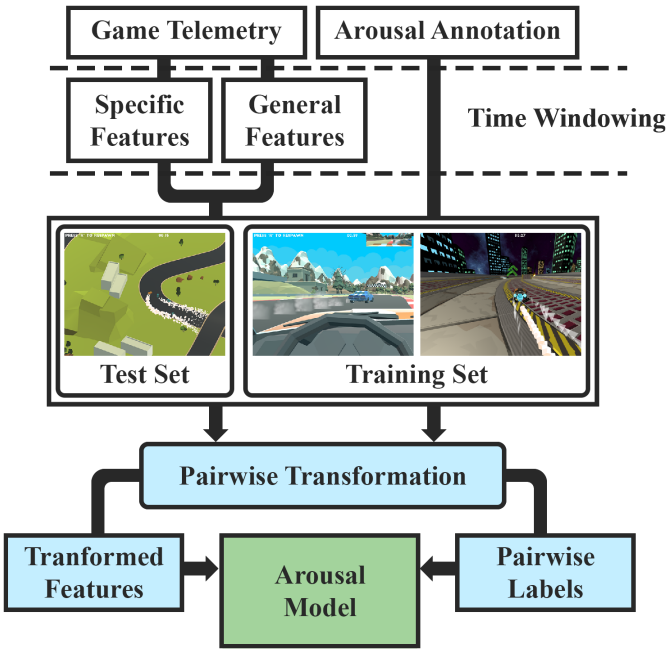

Fig. 1. Genre-based modelling pipeline for modelling arousal across different games of the same genre as presented in this study. Both genre-specific and general features are extracted from the raw telemetry. Models are trained on data from two games and tested on an unseen game within the genre. Preference learning is applied by using a Pairwise Transformation, in which the ranking problem is reformulated as binary classification of pairwise labels (see Section IV-A for more details).

To test our hypothesis, we first design features *specific* to each genre that can predict player experience within a game with high accuracy. Then we examine whether certain ad-hoc designed features that contain general information about gameplay can act as reliable *general* predictors of player experience across games of the same genre. We use the *Affect Game AnnotatIoN* (AGAIN) dataset [7], which includes telemetry and annotations of arousal for almost 1,000 play sessions of nine different games across three different genres (*Racing*, *Shooter*, and *Platformer*). We employ random forests for preference learning in order to build models that predict

arousal both in a specific game and across unseen games of each genre (see Figure 1). Viewing player arousal modelling as a relational learning problem [8], we test the capacity of the models to predict the *change in arousal* in a short time-frame compared to the previous session history. The key results show that the ad-hoc designed general features manage to predict the change of arousal with up to 74% accuracy on average across all genres and 86% in the best cases within shooter games. The core findings of the paper suggest that we can design general features that can predict player experience across unseen games of the same genre with high accuracy. More importantly, such features perform equally well compared to features that are tailored to predict player arousal within the same game that they were trained on.

To the best of our knowledge, we examine genre-based general modelling of player arousal through game telemetry for the first time. Only one similar study examined general arousal modelling [6]. However, here we take a more systematic approach and examine our results in the context of three different videogame genres: racing, shooter, and platformer games. We also reexamine ad-hoc gameplay features through an analysis of feature importance, explaining the results and process of our machine learning models. Our results highlight the importance of a similar temporal dynamic between games. This revelation puts previous general affect modelling approaches into context and foreshadows future challenges in the field.

II. Background

This section highlights related work on modelling players' affective states (Section II-A), and our ordinal approach to emotion modelling (Section II-B).

A. Player Affect Modelling

The field of games user research can generally be divided between static profiling and dynamic modelling [3]. While the former focuses mostly on high-level data aggregation [9] and pattern discovery [10], the latter involves predictive modelling. These modelling tasks can be further broken down into behavioural (i.e. what the player does) and affective (i.e. how the player feels) approaches. Examples for the former include behaviour [11] and churn prediction [12], while examples for the latter include experience [13], [14], motivation [15], and affect modelling [6], [16]. Since most of these studies rely on supervised machine learning, their main limitation is their data needs. Many studies focus on ad-hoc testbeds and game-dependent models. While the resulting models are useful for understanding how players interact with already published games, these models do not generalise well to unseen ones.

To answer this issue, general affect modelling [2] aims to create pre-trained models which can be applied to unseen games. If successful, such models can reduce the data needs of new projects. While research has begun in this field, studies in the literature are still rather sparse. For instance, Shaker *et al.* investigated manual [4] and automated feature mapping [5] through the use of transfer learning. While transfer learning offers a robust approach, interestingly, other studies have been just as successful in applying domain knowledge to hand-craft high-level general features of gameplay. Camilleri *et al.* used game-agnostic features such as playtime and encoded valence as goal oriented and goal opposed events to model arousal across games [6] with moderate success. Similarly, Bonometti *et al.* used activity count and diversity to abstract gameplay and model general engagement across six games [17]. However, a general limitation of these studies is the ad-hoc set of testbed games, which are often limited in scope or fall too far from each other. In this paper, we take a more structured approach to general modelling and investigate the robustness of domain-specific general features created in a top-down manner. As opposed to previous studies, which used ad-hoc setups, we investigate the proposed approach in three different genres, over nine different games.

B. Ordinal Player Modelling

Ordinal affect modelling aims to capture the relative processes behind emotional responses [8], [18]. Human cognition is prone to temporal biases [19] such as anchoring [20], habituation [21], adaptation [22], and other recency effects [23]. Therefore, focusing on the relative differences rather than absolute judgements can lead to more reliable observations and more robust predictions [8]. In the field of games user research, several papers contribute to a growing body of research proving the effectiveness of this approach; see [8], [24]–[28] among many. This approach evidently increases the inter-rater reliability and consistency of data annotations [25], [26], and yields models that have a higher generality across affective corpora [28] and dissimilar videogames [6].

A common issue with ordinal affect modelling is the lack of sufficiently labelled datasets. Because collecting pairwise comparisons through forced-choice surveys can be labour intensive (due to the number of comparisons growing quadratically when new options are introduced), most studies focus on traditional rating methods such as Likert scales. While absolute ratings can be converted to ordinal labels [8], bounded scales come with their own limitations [26]. A good compromise is to collect unbounded ratings, which can still be interpreted in an ordinal fashion but preserves the relative relationship between data points [18]. Inspired by the studies of Lopes *et al.* [29] and Camilleri *et al.* [6], we collect arousal in an unbounded continuous fashion and via the mean value within a time window to predict changes in arousal.

III. The AGAIN Dataset

This study employs the AGAIN dataset[1], which was designed to provide a diverse and robust database for general affect modelling in the domain of videogames [7]. The raw dataset includes $1,116$ playthroughs; after cleaning and preprocessing, the clean dataset (used in this paper) includes 122 participants and 995 playthroughs. More information on the games, the cleaning process, and the dataset are found in [7].

[1]The full dataset is available at https://again.institutedigitalgames.com/

Fig. 2. The 9 games of the AGAIN dataset, one genre per row. Top row is *racing* games, mid row is *shooter* games and bottom row is *platformer* games.

A. Games

The AGAIN dataset includes 9 games in total; 3 games for each of the *racing, shooter*, and *platformer* genres (see Fig. 2). The games were designed as casual representations of popular contemporary and classic games. Because the games had to fit into a 2-minute playtime, they are simplified and resemble a mobile game experience rather than console or PC gameplay. Nevertheless, the games were designed to provide a more realistic testbed with more contemporary aesthetics.

1) Racing: The AGAIN dataset includes three car-racing games, where players have to navigate in a closed-loop track until the timer runs out. In the order that they appear in Fig. 2, the three games in this set are: **TinyCars**, a retro top-down racing game; **Solid**, a rally game; and **ApexSpeed**, an arcade-like speed racer. All of these games feature three opponents who race against the player. The control scheme of the games is quite consistent; ApexSpeed stands out as the car moves automatically on a preset track with lane swapping mechanics.

2) Shooter: The set of shooters includes games where the goal is to eliminate opponents using projectile weapons. In the order that they appear in Fig. 2, the games in this set are **Heist!**, a first-person shooter with health regeneration mechanics; **TopDown**, a retro top-down shooter with unlimited ammo and health pick-ups; and **Shootout**, an arcade shooter. All of these games involve mouse-aim; however Shootout stands out as in this game, the player has no health and is not able to move; the game is only played for score.

3) Platformer: In the platformer games of AGAIN, players have to navigate in a 2D environment, eliminate or evade opponents, and solve light spatial puzzles. In the order that they appear in Fig. 2, the set includes **Endless**, an endless runner; **Pirates!**, a classic Mario-clone; and **Run'N'Gun**, a retro shoot-em up. Platform games are the most diverse in the dataset, with Endless featuring automatic forward movement (like ApexSpeed) and Run'N'Gun featuring weapon aiming.

TABLE I
THE GENERAL GAMEPLAY FEATURES OF AGAIN

Feature	Description
Time Passed	Time since the start of the recording
Player Score	Points earned by the player
Input Intensity	Number of key presses
Input Diversity	Number of unique key presses
Player Activity	Time spent pressing controls
Player Movement	Distance travelled and reticle moved
Bot Count	Number of bots visible
Bot Movement	Bot distance travelled
Bot Diversity	Number of unique bots visible
Object Intensity	Number of objects of interest
Object Diversity	Number of unique objects
Event Intensity	Number of events
Event Diversity	Number of unique events

B. Dataset

The clean AGAIN dataset consists of 122 players playing 995 sessions of 2-minute games [7]. The gender distribution of the participants skews towards men. One participant identified as non-binary, 43 as female, and 78 as male. The average age of participants was 33, ranging from 19 to 55. Most respondents were from the USA (100 participants); other countries were Brazil (10 participants), Italy (3), Canada (2), India (2), Czech Republic (1), Germany (1), and Romania (1). Most of the participants (114) are self-described gamers (either hard-core or casual) and play videogames daily or weekly. Most participants had either a PC, gaming console, or both and played a wide variety of game genres, including shooters, platformers, and driving games.

While the dataset includes both video footage and telemetry data, we focus on the latter in this study. The dataset contains genre-specific telemetry features, which are largely shared across games within the same genre. AGAIN provides 33, 35, and 42 genre-specific features for racing, shooter, and platformer games, respectively. Genre-specific features describe events, interactions, and states from the player's perspective; i.e. only bots and objects visible to the player are logged. Specific features encode the *gameplay context*, player status, bot status, and events both controlled by the player and controlled by the game; more details can be found at [7]. Some games lack gameplay features that other games have, even within the same genre. In case of a missing feature, we fill the missing values with zeroes; i.e. a large loop in the track is a central feature in Solid but missing from TinyCars, subsequently features referencing the loop have a constant value of zero in TinyCars.

Beyond these *specific features*, the AGAIN dataset also provides 13 *general features* [7]. Table I shows these general features and their short descriptions. These features describe the game on a higher level without introducing substantial domain knowledge to the data. Most general features are trivial to create, with the exception of object intensity and object diversity. What constitutes an object in each game varies, but in most cases, this includes passive elements the player can interact with (e.g. destructible elements and power ups).

Fig. 3. Continuous, unbounded arousal annotation with the RankTrace method through PAGAN [7]. The figure shows the Run'N'Gun platformer game.

AGAIN offers continuous, unbounded arousal annotations of each gameplay session recorded with the PAGAN annotation tool [30] (see Figure 3). The interface shows the full history of the annotation process and does not limit the value range of the affect label. Due to these properties, the collected *annotation trace* preserves the subjective and ordinal nature of the player experience [8] and makes the dataset optimal for modelling through preference learning (see Section IV-A).

C. Preprocessing

This paper uses the preprocessed, cleaned dataset from the AGAIN database, from which unresponsive participants and outliers have already been removed [7]. Because the collected data is irregularly spaced due to the online collection protocol, the dataset has also been resampled at 250ms intervals [7], [30]. In this section, we discuss the additional preprocessing steps we took for this paper. As windows of 250ms are not meaningful intervals in terms of human attention due to reaction time, we process the data into 3-second time windows. As presented in Section II-B, we derive the mean of annotation windows. Finally, we apply a 1-second annotation lag (shifted back compared to other features) to account for reaction time. Mariooryad and Busso suggest that although an optimal annotation lag can be found algorithmically, an ad-hoc value between 1 and 3 seconds is practically a good compromise when it comes to similar annotation tasks [31]. Through a preliminary experiment, we determined that an annotation lag of 1 second is sufficient to correct for the participants' reaction. Figure 4 illustrates the annotation metrics and how the annotation lag is applied. After this 1-second lag correction, the dataset consists of $40,836$ datapoints.

IV. Methods

In this paper, we use preference learning (PL) to construct models of arousal. In this section we describe the core aspects of PL (Section IV-A) and the particular algorithm used for our experiments: i.e. random forests (Section IV-B).

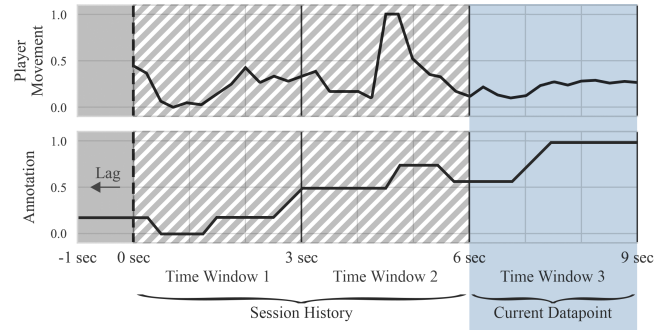

Fig. 4. The aggregation of 3-second time windows. The figure shows an example of comparing the highlighted Time Window 3 to the average of the Session History (Time Window 1 and Time Window 2).

A. Preference Learning

Preference learning is a supervised learning paradigm in which an algorithm learns to distinguish between datapoints in an ordinal manner [32]. The name of the method stems from early applications involving recommender systems and actual user preferences [33], but PL is directly applicable to any supervised learning task in which the target outputs can be treated as ordinal data. The core of the method is the transformation of the data by discarding the output labels but conserving the relationships they describe. The algorithm then learns to predict this relationship instead of any actual label of the data. Some PL techniques derive a ranking score, which can be used for pointwise predictions. PL by *pairwise comparison* has proven to be more robust, providing more stable predictions when the distribution of the labels is not normal or unknown [34], [35].

Formally, in pairwise PL for every pair of datapoints $(x, x') \in X$ and label $(\lambda_x, \lambda_{x'}) \in L$ we create two new points $(x - x')$ and $(x' - x)$ and two new labels, y and y'. In case of $\lambda_x \succ \lambda_{x'}$ (x is preferred to x') we assign $y = 1$ to $(x - x')$ and $y' = -1$ to $(x' - x)$, indicating the preference relation. During the $\lambda_x \succ \lambda_{x'}$ comparison a *Preference Threshold* (P_t) parameter can be applied. P_t takes a value between 0 and 1 and controls the required difference between two labels to be considered a preference. While the previously published baseline of the dataset [7] used consecutive time windows during the pairwise transformation, this paper compares datapoints to data averaged over all previous datapoints within a session. This processing method emphasises the temporality of the data by considering datapoints in relation to the session history. The pairwise transformation is applied to each query, i.e. within each play session separately. The reformulated problem can be solved by any binary classifier. Moreover, by keeping two observations per pair, the baseline of the transformed dataset is always 50%.

B. Random Forests

As explained above, through the pairwise transformation, the task of PL is reformulated as binary classification. In this paper, we use a Random Forest (RF) classifier as it provides

a robust method for modelling arousal. An RF is an ensemble learning method used for classification and regression. As the name suggests, RFs operate by constructing a multitude of randomly initialised independent decision trees during training and use the mode of their individual predictions as the meta output. Decision trees themselves are simple yet powerful machine learning algorithms for predictive modelling [36]; they operate by constructing an acyclical network of nodes, which splits the features of the given dataset into simpler decisions [37]. For our experiments we rely on the *Scikit-learn* Python library [38]. Scikit-learn implements decision trees through an optimised Classification And Regression Tree (CART) algorithm first proposed by Breiman [39]. The CART method uses a generalisation of the binomial variance to evaluate the impurity (and thus splitting criterion) of nodes [40]. It also relies on a process of "overgrowing" and pruning trees [37] to minimise training errors without overfitting. We set the number of estimators to 100 and the maximum depth of each tree to 10 for all experiments. Because RFs are stochastic, we repeat each experiment 20 times and present the average results of all runs.

V. Results

This section presents the key results of our experiments and is structured as follows. First we discuss the parameter tuning protocol (Section V-A) while in Section V-B we present the performance of *game models*, i.e. models trained and validated on the same game. Finally, in Section V-C we introduce and test *genre models*, i.e. models tested on an unseen game while trained on other games of the same genre. Figure 1 shows an example of our pipeline when it comes to genre models.

Reported significance is measured by two-tailed Student's t-tests with $\alpha = 0.05$, adjusted with the Bonferroni correction where applicable.

A. Cross Validation and Parameter Tuning

We use 10-fold cross-validation to test our results. The cross-validation folds are defined between subjects. Because 122 subjects cannot be divided evenly, each fold encompasses either 12 or 13 players. To make our results comparable to each other, the same cross-validation strategy is maintained with both game models and genre models. This means that in the former case, we train models on specific games and test them on unseen players of the same game, and in the latter case, we train models on two games in a given genre and test it on the *unseen players of the unseen game*.

During parameter tuning, we focus on the P_t parameter (see Section IV-A), which determines which changes of arousal should be discarded as marginal. In particular, we seek the best P_t parameter value between 0 and 0.5 with steps of 0.05. Increasing P_t generally leads to higher accuracies (as the separation between preferred and non-preferred classes is clearer), but there is a trade-off in the amount of discarded data. We pick the best P_t given that at least 50% of the available comparisons is maintained in the dataset. Extensive empirical experiments show that $P_t = 0.15$ yields the most accurate models.

TABLE II
Testing accuracies (%) of models trained and tested on the same game. Most accurate models are in bold.

Game	Specific	General	All
TinyCars	**64.8±1.2**	64.3±0.9	64.4±1.1
Solid	71.8±0.4	**73.2±0.7**	72.7±0.6
ApexSpeed	70.5±1.1	**71.9±1.3**	70.8±1.1
Heist!	79.4±0.6	79.4±0.7	**79.8±0.7**
TopDown	82.8±1.1	83.3±1.1	**83.5±1.1**
Shootout	85.8±0.8	85.8±0.8	**85.8±0.8**
Endless	**69.5±1.8**	69.1±1.8	68.9±1.7
Pirates!	69.5±1.6	68.9±1.7	**70.0±1.7**
Run'N'Gun	79.5±1.8	79.8±1.9	**79.8±1.9**

B. Game Models

Table II shows the test accuracies of models trained and tested on the same games. To measure the robustness of general features, we compare the genre-specific and general feature sets to each other and models using all available features. In 5 out of 9 games, a combined feature set of genre-specific and general features leads to the highest accuracies. Notably, shooter and platformer games benefit from these combined feature sets. Interestingly most racing games models based on general features outperform models based on both the combined and genre-specific feature set. While overall general features lead to better predictions than specific features, there are exceptions in TinyCars and Endless.

It should be noted that differences between game models with different inputs are not significant and often marginal. The best performing models are trained and tested on shooter games (average accuracy of 83%) followed by platformers, then racing games (average accuracy of 73% and 70% respectively). The lack of significant difference between feature sets shows the robustness of general features in capturing the complexity of gameplay within each genre. The lack of significant performance increase when combining the feature sets is possibly due to redundancies between genre-specific event telemetry and features such as Event Intensity and Event Diversity (see Table I) that accumulate gameplay events.

C. Genre Models

After acquiring a baseline performance of game models, we move on to genre-based modelling. Each *genre model* is trained on two games and tested on the remaining one within the genre; e.g. the model for TinyCars on Table III shows the results of a model trained on Solid and ApexSpeed and tested on TinyCars (see Figure 1). We are referring to models based on the test game in this section.

Table III shows the performance of our genre models. Results reveal the robustness of general features in comparison to genre-specific ones. In 6 out of the 9 games (all except TinyCars, Endless, and Pirates!), models trained on genre-specific features perform significantly worse than ones trained on feature sets containing general features. In these cases,

TABLE III
TESTING ACCURACIES (%) OF MODELS TRAINED ON TWO GAMES AND TESTED ON AN UNSEEN GAME IN THE SAME GENRE. MOST ACCURATE MODELS ARE IN BOLD.

	Genre Models			Game Models
Game	**Specific**	**General**	**All**	**Best**
TinyCars	66.3±1.6	66.2±1.5	**66.9±1.7**	64.8±1.2
Solid	70.6±0.6	72.2±0.5	72.3±0.6	**73.2±0.7**
ApexSpeed	67.2±1.0	**71.9±1.4**	69.9±1.2	71.9±1.3
Heist!	64.2±0.5	79.3±0.9	79.2±0.9	**79.8±0.7**
TopDown	76.3±1.0	83.5±1.1	**83.7±1.1**	83.5±1.1
Shootout	74.3±0.6	85.8±0.8	85.5±0.8	**85.8±0.8**
Endless	67.4±1.4	**70.0±2.0**	69.8±1.8	69.5±1.8
Pirates!	66.2±1.2	69.5±1.7	69.6±1.7	**70.0±1.7**
Run'N'Gun	62.1±0.9	74.6±1.4	78.0±1.7	**79.8±1.9**

models trained on the specific features have an average of -9% drop in accuracy. The effect is most prominent in games that feature enemy projectiles and some form of shooting mechanics (-12% on average). Interestingly, Run'N'Gun—while it includes shooting—does not feature mouse controls, suggesting that the reason for the performance difference between genre-specific and general models is not the different control scheme but the shooter and shooter-like gameplay dynamics. The difference in racing games is marginal (-2% on average) but still significant.

There is no significant difference between models trained on general, genre-specific and combined features. Furthermore, there is no significant difference between these models and the best game models of Section V-B (included on Table III). The average performance of the best genre models is the same as the game-specific models (70%, 83%, and 73% for the racing, shooter, and platformer games, respectively). Interestingly, models trained on Solid and ApexSpeed perform better on TinyCars than game-specific TinyCars models. While not significantly better, when predicting TinyCars, genre models show an average of $+2\%$ improvement across all feature sets, and the best fold from models trained on general features is $+11\%$ higher (up to 86%) than the best fold of the corresponding game model. A reason for this improvement could be that the fixed isometric view of TinyCars is interfering with the player experience, and the more conventional first- and third-person cameras of Solid and ApexSpeed provide a more consistent coupling between telemetry and arousal. Similarly, genre models tested on ApexSpeed, TopDown, and Endless also outperform game models trained and tested on these games, however in these cases the improvement is marginal (less than $+1\%$ on average).

D. Impact of individual telemetry features

To better understand our results and the reason behind the unexpected robustness of general features, we observe the top five most important features per genre. Feature importance is calculated as the *Mean Decrease Impurity* (MDI) [41], which measures the average amount by which a feature decreases the weighted impurity across all trees in the forest. The MDI value is normalised between 1 and 0, the latter meaning the feature is irrelevant. The ordinal importance of the features can be observed by ranking them by their corresponding MDI values. Here, we average the MDI values of features from different training folds and within a genre to get a bigger picture. Because there was no significant difference between the models trained on different feature sets, and to maximise the number of observed features, we use models trained on all (specific and general) features.

Table IV shows the top five features in each genre ranked by their MDI values. Across all models, Time Passed and Player Score are the most important features. As Player Score is generally increasing as the game progresses, just like Time Passed, it is also a time-related feature. The prominence of these features across the board explains the robustness of genre models when compared to game models. The importance of time makes sense in the context of the games included in AGAIN as they are all designed to be casual and arcade-like. Games like these are designed with an increasing intensity. When it comes to genre models, because time-related features are game-agnostic, the more diverse datasets of two games combined possibly emphasise these features, filtering out more specific ones. The higher MDI score of Time Passed and Player Score for genre models compared to game models supports this hypothesis.

Analysing Table IV by genre, we can see that features relating to player action are more prominent in racing games. This makes sense as the competition is based more on the individual's skill than adversary play in these games. In many cases, the player swiftly overtakes the bots (or is left behind), limiting their interaction. In shooter games, both game models and genre models focus more on the bots numbers and types and the health of either their avatars or the bots. It is surprising that while for shooters there is a starker disagreement between game models and genre models in terms of feature importance, these games produced the most robust models in both cases. However, on a second inspection, this can be attributed to the exceptional prominence of time-related features within this genre. The player's status and the bot are also important in platformer games. Unsurprisingly, the health of the bots (prominent for shooting games) is replaced with the movement of the bots as anticipating the bots' position is essential for winning in this genre.

VI. DISCUSSION

This study presented a robust approach to general affect modelling in videogames by investigating the generality of largely game-agnostic features across three different genres. Our results show that game intensity can be modelled based on simple general features (such as score and playtime) at an accuracy comparable to models based on hand-crafted genre-specific features. These features can be used to create general models that perform comparatively to game-specific models of arousal within genres. In quick and casual games—such as those featured in the AGAIN dataset—the intensity of the gameplay increases over time to such a degree that a relatively simple algorithm can achieve up to 86% average accuracy when predicting the change in player arousal. While games

TABLE IV

FEATURE IMPORTANCE AS DERIVED FROM THE RANDOM FORESTS, AVERAGED ACROSS GAMES OF THE SAME GENRE. FEATURES ARE LABELLED AS GENERAL (G) OR SPECIFIC (S). FEATURES PRESENT IN TOP FIVE FEATURES OF ALL MODELS ARE SHOWN IN BOLD.

Genre	Game Models (averaged)			Genre Models (averaged)		
		Feature	Score		Feature	Score
Racing	G	**Time Passed**	0.089	G	**Time Passed**	0.116
	G	**Player Score**	0.085	G	**Player Score**	0.110
	S	Player Gas Pedal	0.062	S	Player Gas Pedal	0.066
	G	Player Activity	0.045	G	Player Activity	0.038
	S	Bot Score	0.033	S	Bot Collision	0.038
Shooter	G	**Time Passed**	0.167	G	**Time Passed**	0.225
	G	**Player Score**	0.126	G	**Player Score**	0.162
	S	Bot Health	0.054	S	Bot Reloading	0.042
	G	Bot Count	0.051	S	Player Health	0.040
	G	Bot Diversity	0.050	G	Bot Diversity	0.037
Platformer	G	**Time Passed**	0.106	G	**Time Passed**	0.137
	G	**Player Score**	0.104	G	**Player Score**	0.132
	S	Player Damaged	0.039	S	Player Damaged	0.046
	G	Bot Movement	0.037	S	Player Death	0.035
	G	Player Movement	0.035	G	Bot Movement	0.032

with shooting mechanics were easier to predict, some models leave substantial room for improvement. The least successful general models only reached up to 65% when predicting the racing game TinyCars.

Unlike earlier studies [4], [6], we presented a systematic approach to the study of general affect modelling in games, investigating almost $1,000$ gameplay sessions across nine games and three genres. Results presented in this paper show a promising path forward for general affect modelling in videogames but also highlight the challenges ahead. Subsequent analysis of our results showed the prominence of time-related features, which could inform future applications. Normative datasets used for research into game-playing AI use arcade-type games [1], similar to the ones included in AGAIN and used in this study. Augmenting game-playing AI with affective models could help produce more human-like agents and believable characters [42], [43]. Similarly, general models of affect can be used in dynamic adaptation systems, and procedural content generation in an affective loop [3], without having to build specialised models. The surprising robustness of the models also means that the method can possibly be extended to other domains as well, outside of game research. Time-related features can be used to build general models of any user experience with a strong temporal dynamic. Future research should focus on applications of general models of affect in other human-computer interaction applications. While this study only focused on within-genre affect modelling, future studies should explore truly general approaches across different genres. We used top-down, hand-crafted features but the extraction of general features can be enhanced or automated by leveraging unsupervised feature extraction; transfer learning [5] could be useful in this direction. Since AGAIN also includes gameplay videos [7], the former can be achieved using deep learning, and pixel-to-affect modelling [13].

However, the study also highlights a disconnect between contemporary commercial console and computer games and arcade-type games used in different fields of game research. The observed trend between time and the average value of the annotation within a time window suggests a relatively easy task. As AGAIN only includes short, casual games, it is unlikely that the same results would hold for commercial games played over long periods. The models' reliance on time-related features could mean that the presented robustness is only applicable to similarly structured experiences and would not hold up across different contemporary industrial applications. Future studies should aim to verify the results observed here on longer games with a shifting level of intensity. An alternative avenue for research could be on other game-agnostic features and new processing methods for the output of the models, such as the average gradient of the annotation [6] as it is time-independent and subsequently likely to be more robust under longer periods of play.

VII. CONCLUSIONS

This paper examined an approach towards general player experience modelling in a large-scale study of almost $1,000$ play sessions. Experiments focused on general models within the *Racing*, *Shooter*, and *Platformer* genres. Results show that general features describing the player's input, the bots' actions, and the gameplay context on a high level are robust predictors of player arousal. Through two series of experiments, we created baseline game models based on genre-specific and general feature sets and genre models which pool data from two games and predict arousal of unseen players on an unseen game. Our best general models reached up to 74% accuracy on average across all genres and 86% at best within the shooter genre. The core findings of this paper suggest that there exist general in-game features that can predict player experience reliably and can be transferred to games of the same genre with high accuracy. The subsequent analysis of feature importance in the presented study highlights the prominence of time-related features in the machine learning models of player arousal. This result shows the importance of the temporal aspects of the player experience.

REFERENCES

[1] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, and S. Lucas, "General video game AI: Competition, challenges and opportunities," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[2] J. Togelius and G. N. Yannakakis, "General general game AI," in *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 2016.

[3] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018.

[4] N. Shaker, M. Shaker, and M. Abou-Zleikha, "Towards generic models of player experience," in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[5] N. Shaker and M. Abou-Zleikha, "Transfer learning for cross-game prediction of player experience," in *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, 2016.

[6] E. Camilleri, G. N. Yannakakis, and A. Liapis, "Towards general models of player affect," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*, 2017, pp. 333–339.

[7] D. Melhart, A. Liapis, and G. N. Yannakakis, "The Affect Game AnnotatIoN (AGAIN) dataset," *arXiv preprint arXiv:2104.02643*, 2021.

[8] G. N. Yannakakis, R. Cowie, and C. Busso, "The ordinal nature of emotions: An emerging approach," *IEEE Transactions on Affective Computing*, 2018.

[9] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," in *Proceedings of the Symposium on Computational Intelligence and Games (CIG)*, 2009.

[10] S. Makarovych, A. Canossa, J. Togelius, and A. Drachen, "Like a DNA string: Sequence-based player profiling in Tom Clancy's The Division," in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[11] S. C. Bakkes, P. H. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.

[12] M. Viljanen, A. Airola, J. Heikkonen, and T. Pahikkala, "Playtime measurement with survival analysis," *IEEE Transactions on Games*, vol. 10, no. 2, pp. 128–138, 2018.

[13] K. Makantasis, A. Liapis, and G. N. Yannakakis, "From pixels to affect: a study on games and player experience," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*. IEEE, 2019, pp. 1–7.

[14] D. Melhart, D. Gravina, and G. N. Yannakakis, "Moment-to-moment engagement prediction through the eyes of the observer: PUBG streaming on Twitch," in *Proceedings of the International Conference on the Foundations of Digital Games (FDG)*, 2020.

[15] D. Melhart, "Towards a comprehensive model of mediating frustration in videogames," *Game Studies*, vol. 18, no. 1, 2018.

[16] P. Lopes, A. Liapis, and G. N. Yannakakis, "Modelling affect for horror soundscapes," *IEEE Transactions on Affective Computing*, vol. 10, no. 2, pp. 209–222, 2017.

[17] V. Bonometti, C. Ringer, M. Ruiz, A. Wade, and A. Drachen, "From theory to behaviour: Towards a general model of engagement," *arXiv preprint arXiv:2004.12644*, 2020.

[18] G. N. Yannakakis, R. Cowie, and C. Busso, "The ordinal nature of emotions," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII) (ACII)*. IEEE, 2017, pp. 248–255.

[19] A. R. Damasio, *Descartes' error: Emotion, rationality and the human brain*. New York: Putnam, 1994.

[20] B. Seymour and S. M. McClure, "Anchors, scales and the relative coding of value in the brain," *Current Opinion in Neurobiology*, vol. 18, no. 2, pp. 173–178, 2008.

[21] R. L. Solomon and J. D. Corbit, "An opponent-process theory of motivation: I. temporal dynamics of affect," *Psychological Review*, vol. 81, no. 2, p. 119, 1974.

[22] H. Helson, *Adaptation-level theory: an experimental and systematic approach to behavior.* Harper and Row: New York, 1964.

[23] S. Erk, M. Kiefer, J. Grothe, A. P. Wunderlich, M. Spitzer, and H. Walter, "Emotional context modulates subsequent memory effect," *Neuroimage*, vol. 18, no. 2, pp. 439–447, 2003.

[24] H. P. Martinez, G. N. Yannakakis, and J. Hallam, "Don't classify ratings of affect; rank them!" *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 314–326, 2014.

[25] G. N. Yannakakis and H. P. Martinez, "Grounding truth via ordinal annotation," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*, 2015, pp. 574–580.

[26] G. N. Yannakakis and H. P. Martínez, "Ratings are overrated!" *Frontiers in ICT*, vol. 2, p. 13, 2015.

[27] R. Lotfian and C. Busso, "Practical considerations on the use of preference learning for ranking emotional speech," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 5205–5209.

[28] D. Melhart, K. Sfikas, G. Giannakakis, and G. Y. A. Liapis, "A study on affect model validity: Nominal vs ordinal labels," in *IJCAI Workshop on Artificial Intelligence in Affective Computing*. PMLR, 2020, pp. 27–34.

[29] P. Lopes, G. N. Yannakakis, and A. Liapis, "Ranktrace: Relative and unbounded affect annotation," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*, 2017, pp. 158–163.

[30] D. Melhart, A. Liapis, and G. N. Yannakakis, "PAGAN: Video affect annotation made easy," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*, 2019.

[31] S. Mariooryad and C. Busso, "Analysis and compensation of the reaction lag of evaluators in continuous emotional annotations," in *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII)*, 2013, pp. 85–90.

[32] J. Fürnkranz and E. Hüllermeier, "Preference learning," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 789–795.

[33] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2002, pp. 133–142.

[34] J. Fürnkranz and E. Hüllermeier, "Pairwise preference learning and ranking," in *Proceedings of the European Conference on Machine Learning*. Springer, 2003, pp. 145–156.

[35] V. Melnikov, P. Gupta, B. Frick, D. Kaimann, and E. Hüllermeier, "Pairwise versus pointwise ranking: A case study," *Schedae Informaticae*, vol. 25, pp. 73–83, 2016.

[36] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, "Deep learning for affective computing: Text-based emotion recognition in decision support," *Decision Support Systems*, vol. 115, pp. 24–35, 2018.

[37] R. J. Lewis, "An introduction to classification and regression tree (cart) analysis," in *Proceedings of the society for Academic Emergency Medicine (SAEM) annual meeting*, 2000.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[40] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[41] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," *Advances in neural information processing systems 26*, 2013.

[42] J. D. Miles and R. Tashakkori, "Improving the believability of non-player characters in simulations," in *Proceedings of the Conference on Artificial General Intelligence*. Atlantis Press, 2009.

[43] C. Pacheco, L. Tokarchuk, and D. Pérez-Liébana, "Studying believability assessment in racing games," in *Proceedings of the International Conference on the Foundations of Digital Games (FDG)*, 2018, pp. 1–10.