META-LEARNING WITH AUTO-GENERATED TASKS FOR PREDICTING HUMAN BEHAVIOUR IN NORMAL FORM GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, machine learning methods have been successfully applied to predict human behaviour in strategic settings. However, as available data on human behaviour is not always large enough and people's reasoning processes in different types of games are various, learning to get a satisfactory prediction model is a challenge. In this paper, we employ a meta-learning method to improve the learning performance in predicting human behaviour in normal form games. In particular, we first design a behavioural predictor as a deep neural network that captures mixed human behaviour features based on cognitive psychology theory. Given a collected dataset of experimental human behaviour and the learned mixed features, we then construct tasks with unsupervised learning methods and use meta-learning to improve the predictor's generalisation performance on new games. Experimental results show that our proposed meta-learning method considerably increases the accuracy and generalisation for predicting human strategic behaviour.

1 INTRODUCTION

The Nash equilibrium in game theory provides an important solution concept, namely that the optimal outcome of a game is one in which there is no incentive for perfectly rational players to deviate. However, the perfectly rational assumption does not apply to many practical scenarios and these equilibrium strategies may lead to suboptimal outcomes. Experimental results in behavioural game theory literature have proven this (Camerer, 2003; Camerer et al., 2004; Haruvy & Stahl, 2007; Ho et al., 2004), and a wide range of models for predicting human behaviour in games are developed by incorporating the cognitive biases and limitations derived from observations of play and insights from cognitive psychology. For example, quantal best response model (McKelvey & Palfrey, 1995) features that people become more likely to make errors as those errors become less costly, and quantal level-k model (Costa-Gomes et al., 2001) captures the idea that humans can perform only a limited number of iterations of strategic reasoning. Quantal cognitive hierarchy model (Wright & Leyton-Brown, 2014), which combined both insights of quantal response and iterative reasoning, is the state-of-the-art behavioural model for predicting human play in normal form games (NFGs) (Wright & Leyton-Brown, 2017; 2019).

In recent years, machine learning methods have been successfully applied for predicting and understanding human behaviours in decision making problems (such as risk choice problems (Peterson et al., 2021) and NFGs (Hartford et al., 2016)). An NFG involves at least two players and a player should consider other players' possible choices when making her own decision. Given the insight of iterative reasoning, a deep neural network predictor containing feature layers and action response layers was designed in (Hartford et al., 2016), which achieved state-of-the-art performance by learning on a combined experimental dataset with more than 12k observations over 128 NFGs. Although this dataset is the largest dataset of human behaviour on NFGs to date, their final network used only one action-response layer as the networks with more than one action-response layer showed signs of over-fitting: performance on the training set improved steadily as the action-response layers are added but test set performance suffered. Indeed, predicting human behaviour in games is challenging because available human behaviour data is not always large and people's reasoning processes in different types of games are various.

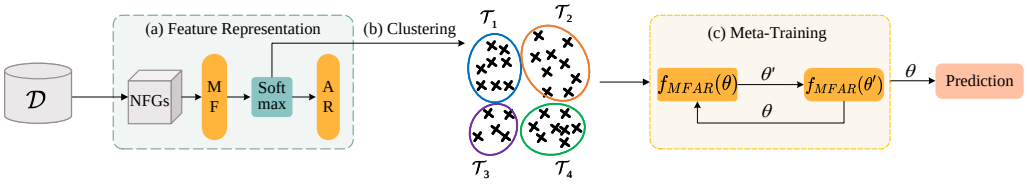


Figure 1.1: Illustration of the approach in this paper. (a) Extracting features from dataset $\mathcal{D}$ by modelling the underlying predictor as a designed neural network MFAR, (b) clustering with the learned features to generate tasks $\{\mathcal{T}_t\}$ and (c) meta-training on these tasks with a meta-learning method of second-order gradient optimisation to get a predictor with parameters θ.

To address this challenge, we design a deep neural network (MFAR) that combines several non-strategic human behaviour features with the architecture of a mixture of experts (MoE) (Jacobs et al., 1991) and then propose a meta-learning method (MFAR-ML) to get a predictor with good generalisation performance on new games (see Figure 1.1). To perform meta-learning, we need to construct tasks from the current problem of learning on the dataset of all NFGs in it, where a *task* is to learn from a batch of the whole human behaviour dataset. However, most NFGs have no specific actual meaning (in analogy with the languages of texts and the species of animals in pictures) that is proper to be used to label and divide them into tasks. Given this, we use an unsupervised learning method to automatically generate tasks by clustering the extracted mixture of features that learned with our designed neural network MFAR.

The main contribution of this paper is as follows: (1) We design a deep neural network that embodies mixed human behaviour features that follows cognitive psychology theory. (2) We propose a meta-learning method to improve prediction performance by learning on tasks that generated by clustering the extracted mixture of features of human behaviour. (3) We empirically evaluate our approaches on experimental datasets and show that it significantly outperforms state-of-the-art on both the accuracy and generalisation for predicting human strategic behaviour.

2 Related Work

2.1 Prediction in Normal-Form Games

The research on predicting human behaviours in NFGs is mainly focused on the behavioural game theory literature, and has gradually attracted the attention of the machine learning community in recent years. The models proposed in behavioural game theory literature (Camerer, 2003; Camerer et al., 2004; Haruvy & Stahl, 2007; Ho et al., 2004) are mainly defined to describe previously identified cognitive processes such as quantal best response (McKelvey & Palfrey, 1995) and limited iterative strategic reasoning (Costa-Gomes et al., 2001). Quantal cognitive hierarchy model (Camerer et al., 2004), combined both insights of quantal response and iterative reasoning, is the state-of-the-art model in the behavioural game theory literature for predicting human play in NFGs (Wright & Leyton-Brown, 2017; 2019). Afterwards, a deep learning approach (Hartford et al., 2016) was proposed to automatically perform cognitive modelling without relying on expert knowledge in behavioural game theory. Comparing to these methods, we defined a new deep neural network with mixture of features and improve the learning performance with meta-learning.

2.2 Meta-Learning

Meta-learning is one of the fastest-growing areas of research in machine learning. Meta-learning explores the common laws in data through machine learning methods, and finds a sufficient representation model of this law. This law is used to complete other tasks and improve the generalisation ability and training efficiency of the model. Typical meta-learning algorithms include model agnostic meta-learning (MAML) (Finn et al., 2017), prototypical networks (ProtoNets) (Snell et al., 2017) and construct tasks for unsupervised meta-learning (CACTUs) (Hsu et al., 2018), et.al. MAML (Finn et al., 2017) trains a meta-model as initialisation and adapts the meta-model to new tasks by gradient descent for several steps. Making a prototypical representation of each class and categoriz-

ing a query point (that is, a new point) depending on how far off they are from each other is the basic idea of ProtoNets (Snell et al., 2017). CACTUs (Hsu et al., 2018) automatically construct tasks from unlabeled data using unsupervised learning, and then apply meta-learning to the generated tasks. In this paper, we also use unsupervised learning as one of the two ways of constructing tasks for meta-learning, and the difference is that we design a neural network that combing several non-strategic human behaviour features with MoE to get exacted features for this task construction. Moreover, compared with most researches that apply meta-learning to multi-task problems, our main purpose of using meta-learning is to improve the learning performance under the condition of small samples.

While most meta-learning literature focus on image classification problems, some of them study prediction problems such as human motion prediction (Gui et al., 2018) and human decision making (Peterson et al., 2021). A proactive and adaptive meta-learning (PAML) method was proposed in (Gui et al., 2018) by combing model-agnostic meta-learning and model regression networks for human motion prediction. The work in (Peterson et al., 2021) studied to use large-scale experiments and machine learning to discover theories of human decision-making. They fine-tune the models that were previously already fitted to the aggregate data to data from individuals in their replication dataset in order to obtain a model with an individual difference. These challenges, however, are different from the NFGs studied in this paper, where a player should take into account other players' potential choices when making her own.

3 PRELIMINARY

3.1 NORMAL-FORM GAMES

Formally, an NFG can be defined by a tuple $G = (N; A; u)$, where $N = \{1, 2, ..., n\}$ is the set of n players; $A = A_1 \times A_2 \times ... \times A_n$ is the set of possible action profiles; A_i is the set of actions available to player $i \in N$. Moreover, agents' strategies can be defined as a function $u_i : A \to R$, each of which maps from an action profile to a utility for player i. A mixed strategy of player i is a probability distribution over her set of possible action profiles. Rational agents play to maximise their expected utilities and game theory studies how to play optimally in these settings with multiple players. Although game theory solution concepts such as *Nash equilibrium* have many appealing properties, experimental evidence shows that Nash equilibrium often fails to describe human behaviour in games (Goeree & Holt, 2001), even among professional game theorists (Becker et al., 2005).

3.2 PREDICTING HUMAN BEHAVIOUR IN NFGS

Formally, we denote an experimental dataset $\mathcal{D} = \{(G_i, a_{ij}|j = 1, \ldots, J_i)|i = 1, ..., I\}$ as a set containing I elements. Each element is a tuple containing a game G_i and a set of J_i pure actions a_{ij}, each played by a human subject in G_i. A *behavioural model* (a *predictor*) is a mapping from a game description G and a vector of parameters θ to a predicted distribution over each action profile a in G, which we denote $Pr(a|G, \theta)$. The statistical frequency of the action choices in each game can be viewed as a label for predictor learning.

The main two category of behavioural models for human playing in NFGs are *cognitive psychology models* and *deep neural network models*, in which the former one is aim to describe previously identified cognitive processes and the later one is designed and learned by deep neural networks. The state-of-the-art deep neural network predictor consists well-designed feature layers and action response layers (Hartford et al., 2016). We name their approach as neural network with *feature layers* and *action response layers* (FAR) for the simplicity of description later in this paper.

4 METHODS

In this section, we present our method of meta-learning with auto-generated tasks for human behaviour prediction. First, we design a deep neural network with *mixture of features* and *action response layers* (MFAR) to roughly model and be learned to get an underlying behavioural predictor. Then, we propose three methods to *construct tasks* from the experimental human behaviour dataset and use *meta-learning* to improve the prediction performance.

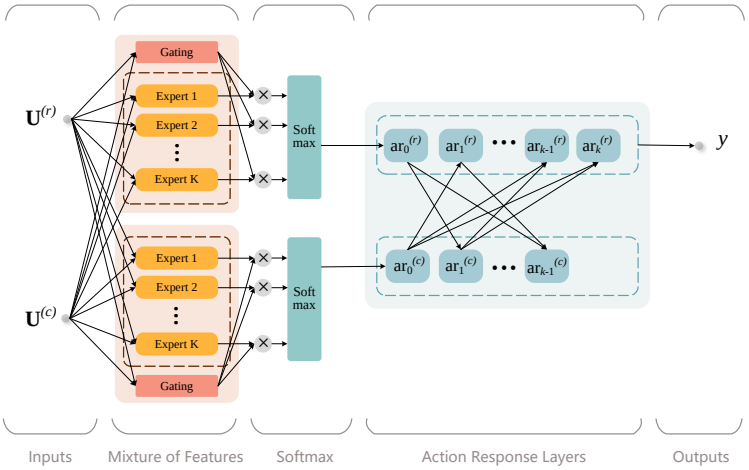


Figure 4.1: Modelling the underlying predictor (MFAR) as a deep neural network with mixture of features and action response layers, where each non-strategic human behaviour feature is modelled by an expert in MoE and the iterative strategic reasoning processes are modelled by the action response layers.

4.1 Modelling the underlying predictor

Following the FAR designed in (Hartford et al., 2016), we model our network MFAR with two parts: mixture of features (MF) and action response (AR) layers, as shown in Figure 4.1. Unlike the feature layers in FAR, MFAR combines human behavioural features with the architecture of MoE (Jacobs et al., 1991), which is an ensemble learning method that looks to explicitly tackle a predictive modelling problem in terms of subtasks leveraging expert models.

Mixture of features The behavioural feature layers take the row and column player's normalised utility matrices $\mathbf{U}^{(r)}$ and $\mathbf{U}^{(c)} \in \mathbb{R}$ of an NFG as input, and output a mixed non-strategic behavioural feature (i.e., a probability distribution for each player over all of her actions), where the row player has m actions and the column player has n actions. Five non-strategic human behaviours based on cognitive psychology theory (Wright & Leyton-Brown, 2019) are chosen as features/experts in MFAR: *Maxmax payoff*, *Maxmin payoff*, *Minmax regret*, *Minmin unfairness* and *Maxmax welfare*. Details of these features' mathematical formulation and network structures are given in Appendix A.1. To combine these features, MFAR uses an MoE architecture to model the mixture of features as shown in Figure 4.1, in which each expert models a non-strategic human behaviour feature and the weights of experts (i.e., the outputs of gating) depends on the inputs (i.e., the payoff matrix of a game). The gating network is a fully-connected network with one or two layers, and receives the same input as the expert networks. Formally, the column player's output of MF can be represented as $h\left(\mathbf{U}^{(c)}\right) = \sum_{k=1}^{K} g\left(\mathbf{U}^{(c)}\right)_k f_k\left(\mathbf{U}^{(c)}\right)$, where $g\left(\mathbf{U}^{(c)}\right)_k$ is the kth logit of the output of $g\left(\mathbf{U}^{(c)}\right)$, indicates the probability for expert f_k. Here, $f_k, k = 1 \ldots K$ are K expert networks and g represents the gating network that ensembles the results from all experts.

This is different with the way in (Wright & Leyton-Brown, 2019) that the learned combing weights are fixed and independent with inputs. Obviously, the dependence in this MoE architecture allows the combing weights (i.e., values of $g\left(\mathbf{U}^{(c)}\right)$ and $g\left(\mathbf{U}^{(r)}\right)$) vary with different games (i.e., $\mathbf{U}^{(c)}$ and $\mathbf{U}^{(r)}$ of an NFG G). This is more realistic, because in different games, a person may follow different criteria in mind to make decisions.

To note, the experts in this designed MFAR do not retain any learnable parameters, only the gating network have learnable parameters. Moreover, to reduce model parameters, the parameters of both the row and the column player's gating network are shared in MFAR.

Action response layers After the softmax layer, the column player's distribution over each action profile is calculated and can be seen as a mixed level-0 heuristic for her, i.e., $\text{ar}_0^{(c)} = h^{(c)}$. We use the same architecture AR layers as that defined in (Hartford et al., 2016). Thus, we can compute expect utility of the row player with respect to the vector of beliefs about the column player's choice of actions $\text{ar}_0^{(c)}$ and takes a softmax over it to get $\text{ar}_1^{(r)}$ as follows:

$$\text{ar}_1^{(r)} = \text{softmax}\left(\lambda \sum_j u_{1,j}^{(r)} \text{ar}_{0,j}^{(c)}, \ldots, \lambda \sum_j u_{m,j}^{(r)} \text{ar}_{0,j}^{(c)}\right) \tag{1}$$

Where λ is a scalar sharpness parameter to sharpen the resulting distribution. Following this way, other units of action response layers can be calculated. Although multiple action-response layers tested in the deep network in (Hartford et al., 2016), their final network used only one action-response layer because more than one action-response layer showed signs of over-fitting: performance on the training set improved steadily as they added AR layers but test set performance suffered. However, statistics in (Ho et al., 2004) show that cognitive levels many people are more than one. In this paper, we expect to use meta-learning eliminate to this over-fitting and effectively utilise multi-layer action-response architecture.

Representational generality of our model Thus we have defined the model of MFAR, in which MF embodies several human behaviour features that follows cognitive psychology theory. If we don't specify the features in MF, MFAR is more general than FAR as the later one can be derived from the former one by setting all outputs of gating units as one. Therefore, as FAR, MFAR can express several cognitive psychology models such as the quantal cognitive hierarchy (Wright & Leyton-Brown, 2014), quantal level-k (Stahl II & Wilson, 1994), cognitive hierarchy (Camerer et al., 2004) and level-k (Costa-Gomes et al., 2001). Moreover, given its neural network architecture, MFAR is more general than that proposed in (Wright & Leyton-Brown, 2019) takes weighted linear or logit specification to ensemble the non-strategic behaviours.

Given this, MFAR can be extended to a model with high representational generality and can be learned with the experimental human behaviour dataset to get an underlying behavioural predictor. Moreover, to improve the predictor's learning generalisation on new games, we next propose methods of constructing tasks from the human behaviour dataset and meta-learning on these tasks to get our final model.

4.2 CONSTRUCTING TASKS

To perform meta-learning, we need to construct behavioural prediction tasks $\{\mathcal{T}_t\}$ from the problem of predicting over all games $\{G_i\}$ in $\mathcal{D}$. Here, a task $\mathcal{T}_t$ is to learn from a batch of the whole human behaviour dataset. $\mathcal{T}_t$ consists of K training games and statistics of actions human subjects played, R query games and statistics of actions human subjects played. That is, in a task, we have $K + R$ games and statistics of actions.

In order to construct tasks, although it is not easy to use the actual meaning of NFGs (such as the languages of texts and the species of animals in some pictures), we can facilitate some abstract features to do this. For example, in an NFG that all players have more actions available than in another NFG, humans likely take more effort to make a decision. Moreover, for the two NFGs with and without dominant equilibrium, humans' decision-making process should also be very different. In addition, human population are heterogeneous because for sub-populations of players using different behavioural rules (i.e., action selection principles.) (Haruvy & Stahl, 2007). Given this, we can divide the whole predicting problem into several tasks so that NFGs in a same task have some similarity. Next, we propose three methods to construct tasks given different features (one is human-specific and the other two are automatically generated).

Constructing tasks with game theoretic features Research results in (Wright & Leyton-Brown, 2017) show that the performance of a prediction model is sensitive to the selected games with different properties of dominance solvability and equilibrium structure. However, the neural networks both designed in FAR and MFAR have not featured this. Given this, we partition the games by (1) whether an iterated removal of dominant strategies (either strict or weak) might solve a game and how many iterations were necessary, and (2) the number and type of Nash equilibria that each game possesses. These game theoretic properties are listed in Appendix A.2.

Algorithm 1 Meta-learning for predicting human behaviour in NFGs

Input: experimental dataset $\mathcal{D}$ including the games and statistics of actions human played, predictor $\mathcal{P}_\theta$ with initial parameters θ
Parameter: α, β: meta-learning rate hyper-parameters
Output: Learned predictor $\mathcal{P}_\theta$

```
Randomly initialise predictor parameters θ.
Extract features and construct tasks {T_t} from D by using one of the three methods proposed
while not done do
    Sample batch of tasks from {T_t}
    for all T_t in the batch do
        Evaluate ∇L_{T_t}(P_θ) with respect to K games in this task
        Compute adapted parameters with gradient descent: θ'_t = θ − α∇_θ L_{T_t}(P_θ)
    end for
    Update θ = θ − β∇_θ Σ_{T_t} L_{T_t}(P_{θ'_t})
end while
return P_θ
```

Constructing tasks by deep clustering with payoff matrix For the original game set $\{G_i\}$, according to its potential characteristics, the tasks of meta-learning can be constructed by DeepCluster (Caron et al., 2018), a deep clustering approach that simultaneously learns the neural network's parameter settings and the clustering of the output features. Generally, we haven't the label information that conforms to the game category, so we use an auto-encoder network (Rumelhart et al., 1985) for unsupervised learning in the deep clustering method. The auto-encoder network is composed of encoder and decoder, and guides the neural network to learn a mapping relationship by taking the input data as supervision information. When training the network, we expect to get a reconstructed output that can accurately characterise the games in $\{G_i\}$, and cluster the output information (i.e., a partition of the game set $\{G_i\}$ to get tasks $\{\mathcal{T}_t\}$). The detailed network structure and hyper-parameters of the adopted auto-encoder is listed in Appendix A.5.

Constructing tasks by clustering with mixture of features As shown in Figure 1.1, another way to realise the auto-generation of tasks is using the learned mixture of features in MFAR, i.e., using $h\left(\mathbf{U}^{(c)}\right)$ and $h\left(\mathbf{U}^{(r)}\right)$ of a game G with payoff matrix $\mathbf{U}^{(c)}$ and $\mathbf{U}^{(r)}$. Specifically, once we have got an underlying predictor by learning with NFAR from all games in $\mathcal{D}$, we can take $h\left(\mathbf{U}^{(c)}\right)$ and $h\left(\mathbf{U}^{(r)}\right)$ as the features of a game, select the coefficient of association to measure similarity of these sample features, and carry out the clustering with Gaussian mixture models (Diaz-Rozo et al., 2020) according to the distribution form of features in space. In addition, as $h\left(\mathbf{U}^{(c)}\right) = \sum_{k=1}^{K} g\left(\mathbf{U}^{(c)}\right)_k f_k\left(\mathbf{U}^{(c)}\right)$, we also can use only $g\left(\mathbf{U}^{(c)}\right)$ or $\left\{f_k\left(\mathbf{U}^{(c)}\right)\right\}$ of a game as its features for clustering. Intuitively, $g\left(\mathbf{U}^{(c)}\right)$ and $\left\{f_k\left(\mathbf{U}^{(c)}\right)\right\}$ respectively represents weights and values of non-strategic human behaviours, and may work in different ways. We also compared with using these two features for clustering as ablation experiments and the results are shown in Appendix A.6.

4.3 META-TRAINING

Given the constructed tasks, we use meta-learning to improve predictor's the learning performance. The meta-learning algorithm we use in this paper is model agnostic meta-learning (MAML) (Finn et al., 2017), which is one of the state-of-the-art meta-learning algorithms. Through a meta-learning process that gains knowledge from a large number of behaviour prediction tasks $\{\mathcal{T}_t\}$ with small samples, MAML enables the meta-learner to produce a model θ with high accuracy and generalisation. The loss $\mathcal{L}_{\mathcal{T}_t}(\mathcal{P}_\theta)$ is defined by a certain function to evaluate the prediction performance on task $\mathcal{T}_t$ with model $\mathcal{P}_\theta$. In addition, our task construction method via unsupervised learning makes this meta-learning can be seen as a variant of CACTUs(Hsu et al., 2018).

Thus, we have presented all the key parts of our method and here we present our algorithm of training a predictor with meta-learning in Algorithm 1. The input includes experimental human behaviour dataset $\mathcal{D}$, the network model of the predictor $\mathcal{P}_\theta$ (such as FAR, MFAR or other models) with parameters θ, and two hyper-parameters of meta-learning rates α, β, where α controls the

learning rate in the inner-loop and β controls the learning rate in the outer-loop. The tasks $\{\mathcal{T}_t\}$ can be constructed by using one of the methods proposed in previous section. Then we use MAML to learn to get the model $\mathcal{P}_\theta$ by using both the inner-loop and outer-loop training.

5 EXPERIMENTAL EVALUATION

The goal of our experimental evaluation are designed to answer the following research questions: (1) Can the MoE architecture in MFAR improve the learning accuracy? (2) Which one of the three ways for constructing tasks is most effective? (3) Can meta-training improve the learning accuracy and model generalisation? The experimental setup and main results are presented in this section and other results are provided in Appendix.

5.1 EXPERIMENTAL SETUP

Following (Hartford et al., 2016), we collect a dataset that combines observations from 9 human-subject experimental studies conducted by behavioural economists, in which human subjects were paid to select actions in NFGs. The payment depended on the subject's actions and the actions of their unseen opposition who chose an action simultaneously. Given this, we get a dataset with 11827 observations of 127 unique games for experimental evaluation. Details of the combination of the dataset are listed in Appendix A.3

We set meta-learning rate of the inner loop and outer loop in meta-training for all the models to 0.005 and use Adam as the meta-optimiser. The initial learning rate is 0.0002, $\beta_1 = 0.9, \beta_2 = 0.99$ and $\epsilon = 10^{-8}$. The gating network and action response layers use L_1 regularisation with a parameter of 0.01. During the meta-training period, we use 10 meta-gradient updates to train 31k epochs. We train all of these networks on a single computer with an Intel i7 CPU, 32 GB of RAM.

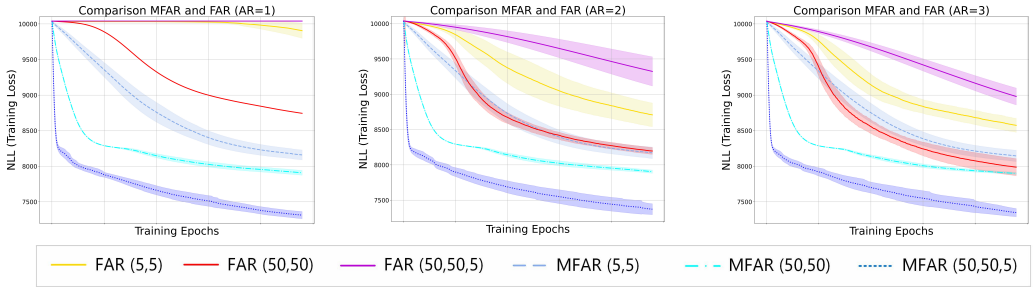
We focus on the model's ability to predict the distribution over the row player's action, rather than just its accuracy in predicting the most likely action. As a result, we fit models to maximise the likelihood of training data $\mathbb{P}(\mathcal{D}|\theta)$ (where θ denotes the parameters of the model and $\mathcal{D}$ is the dataset) and evaluate them in terms of negative log-likelihood on the test set.

5.2 RESULTS

Benefit of Mixture of Features We compare MFAR with FAR on a variety of configurations of the MF layers and AR layers. Hyperparameters of the final MFAR and FAR are listed and compared in Appendix A.4. Figure 5.1 (a) shows the effect of using a different number of gating units and layers in MF on performance with one AR layer. We found that the two-layer gating network with 5 units of MFAR (blue curve) performed generally in training, but well in testing. Meanwhile, a two-layer gating network with 50 units makes MFAR perform better on both the training set and test set. Clearly, adding a third gating layer results in better training performance, but the three-gating network (dark blue curve) makes the model easier to overfit, therefore, the test loss is about 13.17% higher than that of the two-gating network model. After training 25k epochs, the test loss of MFAR (50,50) is 9.16% lower than that of the best trained FAR model (red curve). In summary, combined with training loss and test results, MFAR (50,50) is the best model when the number of AR layers is one.

Figure 5.1 (b) and (c) consider the effect of varying the number of AR layers on performance. When the AR layer is two or higher, the test performance of FAR (50,50) is worse and overfitting occurs. On the contrary, for MFAR (50,50) model (sky blue curve), different AR layers have little effect on the test accuracy of the model. Hence, we picked AR = 1 since it performs well during training and reduces the negative log-likelihood of test loss to the lowest of all AR layers. Therefore, our final underlying network used only one AR layer. Our model combines human behavioural features with the architecture of MFAR achieves the better performance under the same amount of training data, and has made a good balance between efficiency and effectiveness. Thus, our final underlying network contains two layers of 50 gating units and one AR layer.

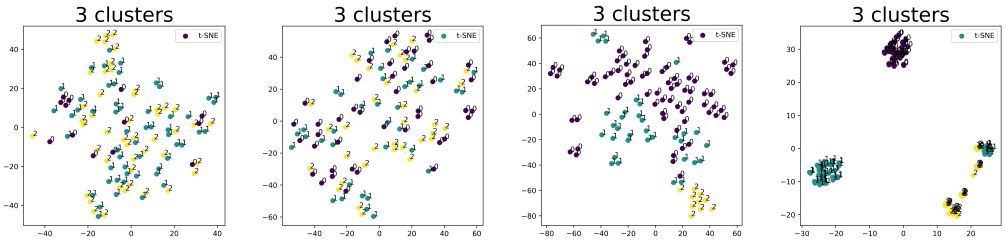
Task Construction By using each of the methods proposed in section 4.2, we construct three tasks and visually analyse these tasks' clustering effects with t-sne. As shown in Figure 5.2 (a)



(a) 1 AR layer (b) 2 AR layers (c) 3 AR layers

Model		FAR (5,5)	FAR (50,50)	FAR (50,50,5)	MFAR (5,5)	MFAR (50,50)	MFAR (50,50,5)
NLL ±95% CI	AR=1	1657.75±5.83	1591.26±1.97	1665.23±0.05	1453.76±15.15	**1457.74±9.84**	1649.74±91.32
	AR=2	1455.54±23.60	1577.11±38.14	1511.98±37.46	1451.08±25.27	1461.56±14.90	1587.18±66.28
	AR=3	1433.84±22.25	1602.21±54.96	1449.94±17.30	1452.77±24.20	1466.56±13.23	1631.72±125.88

Figure 5.1: Loss curves of original FAR and MFAR in the training process with different AR layers and different neural networks are shown in the top three figures and the performance of testing is reported in the bottom table. The lower loss MFAR (50,50) model (shown as a sky blue curve) performs well in the training process and test results. The shadow parts of curves represents 95% confidence intervals, with results from 10 different random seeds.



(a) D (b) NE (c) DC (d) MF

Figure 5.2: Constructed tasks' visualisation via t-sne for constructing methods based on (a) games' dominance solvability (D), (b) games' equilibrium structure (NE), deep clustering with payoff matrix (DC) and (d) clustering with mixture of features (MF).

(b), it is difficult to see obvious clustering effect from the tasks constructed with the properties of dominance solvability (D) and Nash equilibrium structure (NE). The tasks constructed with deep clustering (DC) are visually shown in Figure 5.2 (c), and the main characteristics of the data are clearly distinguished. The best clustering effect belongs to the tasks constructed with mixture of features (MF), in which the tasks are divided into three distinct types of data, as shown in the Figure 5.2 (d). Additional results of the tasks constructed by methods with more different settings are given in Appendix A.6.

Results of Prediction Performance We choose the negative log-likelihood to measure the prediction performance of these models, as shown in Table 1. Both FAR and MFAR have trained 25k epochs, but the assessment differs. FAR needs to be retrained 100 times on 10-fold cross-validation to acquire better results, whereas MFAR does not need fine-tuning and has far superior verification performance than FAR. We evaluate the performance of the model on the total dataset with 10-fold cross-validation after training 31k epochs for FAR-ML and MFAR-ML. It can be seen from Table 1 that the overall performance of MFAR and its meta-training process is significantly better than FAR and FAR-ML. In particular, MFAR-ML(MF-3C) performs well in clustering results and obtains the best performance among all task construction methods for MFAR-ML. Detailed clustering results analysis and DBI indicators are shown in Appendix A.6. For the test loss, MFAR-ML(MF-3C) im-

Table 1: Prediction performance of FAR, MFAR and both of them with four different construction of tasks. DC-3C and DC-4C represent the use of DC to construct three and four classes of tasks for meta-training, respectively. Similarly, the same is true for MF-3C and MF-4C.

Method	NLL ±95%CI	
	Test Loss	Training Loss
FAR	1497.25±2.85	13230.50±4.46
FAR-ML(D)	959.02±5.85	8566.22±12.21
FAR-ML(NE)	938.89±4.32	8449.56±3.12
FAR-ML(DC-3C)	951.03±0.36	8531.46±0.24
FAR-ML(MF-3C)	1376.75±0.17	12389.40±0.38
FAR-ML(DC-4C)	1067.66±0.98	9606.39±5.57
FAR-ML(MF-4C)	990.95±1.02	8918.52±0.43
MFAR	**794.68±0.76**	7151.28±1.28
MFAR-ML(D)	883.62±4.67	7952.44±2.54
MFAR-ML(NE)	889.79±5.41	8008.09±1.81
MFAR-ML(DC-3C)	886.47±0.23	7978.13±2.48
MFAR-ML(MF-3C)	**744.84±1.15**	6702.91±0.56
MFAR-ML(DC-4C)	893.10±0.68	8037.84±0.78
MFAR-ML(MF-4C)	773.35±0.49	6957.54±0.18

Table 2: Generalisation (with 95% confidence intervals) performance on the validation set. We compared the model performance of FAR and MFAR with or without a meta-training process.

	FAR	FAR-ML(MF-3C)	MFAR	MFAR-ML(MF-3C)
Generalisation Error	1703.93±80.81	1451.48±11.47	1243.37±16.94	**1109.56±14.68**

proves the accuracy by 50.30% compared with the previous best model, which is quite accurate and far superior to FAR.

As shown in the above results of clustering findings and prediction performance, clustering with a mixture of features is the best method for MFAR-ML. With three tasks, MFAR-ML(MF) has the highest prediction accuracy in a 10-fold cross-validation when it comes to clustering findings. Hence, we finally choose clustering with a mixture of features to generate MFAR-ML tasks.

Results of Generalisation Performance We validated the generalizability of our approach to new tasks using 80% trained and 20% validated proportional datasets. Instead of using the same 10-fold cross-validation method as the results in Table 1, we use the hold-out method. The training set is used for meta-training and the validation set is applied to detect the generalisation performance of the model. As shown in Table 2, meta-training largely improves the generalisation of the model. In particular, the meta-training effect on MFAR is more significant, exceeding the existing state-of-the-art level. In conclusion, MFAR with a meta-training process can further capture iterative strategic reasoning with good generalisation performance in the face of new games.

6 Discussion and Conclusions

In this work, we have proposed a method of meta-learning with auto-generated tasks for predicting human behaviour in NFGs. Compared with FAR, a state-of-the-art human strategic behaviour predictor with a deep neural network, our MFAR performs well both when using itself and using it in meta-training processes. Experimental results show that MFAR and MFAR-ML dramatically outperform FAR in prediction performance. Among all the task construction methods, clustering with a mixture of features has achieved the best performance, which means that the unsupervised learning method significantly outperforms the classical game theoretic method to generate meta-learning tasks. However, it is still an open problem of exploring a perfect representation for constructing meta-learning tasks for NFGs. In addition, extending our approach to more complex environments such as extensive-form games and stochastic games is a direction of future work.

REFERENCES

Tilman Becker, Michael Carter, and Jörg Naeve. Experts Playing the Traveler's Dilemma. Diskussionspapiere aus dem Institut für Volkswirtschaftslehre der Universität Hohenheim 252/2005, Department of Economics, University of Hohenheim, Germany, January 2005. URL https://ideas.repec.org/p/hoh/hohdip/252.html.

Colin F. Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 09 2003.

Colin F. Camerer, Teck-Hua Ho, and Juin-Kuan Chong. A Cognitive Hierarchy Model of Games*. *The Quarterly Journal of Economics*, 119(3):861–898, 08 2004. ISSN 0033-5533. doi: 10.1162/0033553041502225. URL https://doi.org/10.1162/0033553041502225.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 139–156, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01264-9.

David J Cooper, John B Van Huyck, et al. Evidence on the equivalence of the strategic and extensive form representation of games. *Journal of Economic Theory*, 110(2):290–308, 2003.

Miguel Costa-Gomes, Vincent P. Crawford, and Bruno Broseta. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235, 2001. doi: https://doi.org/10.1111/1468-0262.00239. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00239.

Javier Diaz-Rozo, Concha Bielza, and Pedro Larrañaga. Machine-tool condition monitoring with gaussian mixture models-based dynamic probabilistic clustering. *Engineering Applications of Artificial Intelligence*, 89:103434, 2020. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2019.103434. URL https://www.sciencedirect.com/science/article/pii/S095219761930332X.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1126–1135. JMLR.org, 2017.

Jacob K. Goeree and Charles A. Holt. Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, 91(5):1402–1422, December 2001. doi: 10.1257/aer.91.5.1402. URL https://www.aeaweb.org/articles?id=10.1257/aer.91.5.1402.

Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José M. F. Moura. Few-shot human motion prediction via meta-learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 441–459, Cham, 2018. Springer International Publishing.

Jason Hartford, James R. Wright, and Kevin Leyton-Brown. Deep learning for predicting human strategic behavior. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2432–2440, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Ernan Haruvy and Dale O. Stahl. Equilibrium selection and bounded rationality in symmetric normal-form games. *Journal of Economic Behavior and Organization*, 62(1):98–119, 2007. ISSN 0167-2681. doi: https://doi.org/10.1016/j.jebo.2005.05.002. URL https://www.sciencedirect.com/science/article/pii/S0167268105002271.

Ernan Haruvy, Dale O Stahl, and Paul W Wilson. Modeling and testing for heterogeneity in observed strategic behavior. *Review of Economics and Statistics*, 83(1):146–157, 2001.

Teck Ho, Colin Camerer, and Juin-Kuan Chong. A cognitive hierarchy model games. *The Quarterly Journal of Economics*, 119:861–898, 02 2004. doi: 10.1162/0033553041502225.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning, 2018. URL https://arxiv.org/abs/1810.02334.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10:6–38, 1995.

Martin J Osborne et al. *An introduction to game theory*, volume 3. Oxford university press New York, 2004.

Joshua C. Peterson, David D. Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L. Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214, 2021. doi: 10.1126/science.abe2629. URL https://www.science.org/doi/abs/10.1126/science.abe2629.

Brian W Rogers, Thomas R Palfrey, and Colin F Camerer. Heterogeneous quantal response equilibrium and cognitive hierarchies. *Journal of Economic Theory*, 144(4):1440–1467, 2009.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Dale O Stahl and Ernan Haruvy. Level-n bounded rationality and dominated strategies in normal-form games. *Journal of Economic Behavior & Organization*, 66(2):226–232, 2008.

Dale O Stahl and Paul W Wilson. On players' models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1):218–254, 1995.

Dale O Stahl II and Paul W Wilson. Experimental evidence on players' models of other players. *Journal of economic behavior & organization*, 25(3):309–327, 1994.

J. R. Wright and Kevin Leyton-Brown. Level-0 meta-models for predicting human behavior in games. *Proceedings of the fifteenth ACM conference on Economics and computation*, 2014.

James R. Wright and Kevin Leyton-Brown. Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior*, 106:16–37, 2017. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2017.09.009. URL https://www.sciencedirect.com/science/article/pii/S0899825617301574.

James R. Wright and Kevin Leyton-Brown. Level-0 models for predicting human behavior in games. *J. Artif. Int. Res.*, 64(1):357–383, jan 2019. ISSN 1076-9757. doi: 10.1613/jair.1.11361. URL https://doi.org/10.1613/jair.1.11361.

A APPENDIX

A.1 NON-STRATEGIC HUMAN BEHAVIOURS IN NFGS

We use the architecture of MoE to ensemble five non-strategic behaviours (Wright & Leyton-Brown, 2019). The mathematical formulation of these five non-strategic behaviours are as follows:

- **Maxmax payoff**. A maxmax action for a player is the best action in best case. That is: $f^{\text{maxmax}}(i) = \frac{1}{l}$ if $i \in \arg\max_{i\in\{1,\dots,m\}} \max_{j\in\{1,\dots,n\}} u_{i,j} \in \mathbf{U}^{(r)}$ and $f^{\text{maxmax}}(i) = 0$ otherwise.

- **Maxmin payoff**. In contrast, a maxmin action for a player is the action with the best worst-case guarantee. That is: $f^{\text{maxmin}}(i) = \frac{1}{l}$ if $i \in \arg\max_{i\in\{1,\dots,m\}} \min_{j\in\{1,\dots,n\}} u_{i,j} \in \mathbf{U}^{(r)}$ and $f^{\text{maxmin}}(i) = 0$ otherwise.
- **Minmax regret**. The minimax regret criterion is the criteria that minimise the maximum regret. It is a kind of analysis that evaluates the maximum regret a player will have by choosing an action and calculate the best actions that will make the regret minimum. The regret is defined as $r(i,j) = \max_i u_{i,j} - u_{i,j}, u_{i,j} \in \mathbf{U}^{(r)}$ and the Minimax regret is: $f^{\text{minmax regret}}(i) = \frac{1}{l}$ if $i \in \arg\min_{i\in\{1,\dots,m\}} \max_{j\in\{1,\dots,n\}} r_{i,j}$ and $f^{\text{minmax regret}}(i) = 0$ otherwise.
- **Minmin unfairness**. Fairness of outcomes is a common feature of human play in strategic situations and the minmin unfairness can be defined as: $f^{\text{minmin unfairness}}(i) = \frac{1}{l}$ if $i \in \arg\max_{i\in\{1,\dots,m\}} \min_{j\in\{1,\dots,n\}} |u^{(r)}_{i,j} - u^{(c)}_{i,j}|$ and $f^{\text{minmin unfairness}}(i) = 0$ otherwise.
- **Maxmax welfare**. Finally, a nonstrategic player might choose an action that produces the best overall benefit to the players collectively. That is: $f^{\text{maxmax welfare}}(i) = \frac{1}{l}$ if $i \in \arg\max_{i\in\{1,\dots,m\}} \max_{j\in\{1,\dots,n\}} |u^{(r)}_{i,j} + u^{(c)}_{i,j}|$ and $f^{\text{maxmax welfare}}(i) = 0$ otherwise.

The network structure of these five non-strategic behaviours are as follows:

- **Maxmax payoff**. We denote $H^{(1)}$ and $H^{(2)}$ as the output of the first and second expert layers, respectively, and $H^{(1)}_r$ is its maximum row pooling output. Let $w_1 = w_2 = 1$, and $b = 0$ where is some scalar $b \geq min_{i,j} U^{(r)}_{i,j}$ for $min_{i,j} U^{(r)}_{i,j} = 0$. Then

$$H^{(1)} = relu(U^{(r)} + b)$$
$$H^{(2)} = relu(cH^{(1)}_r)$$

  where $U^{(r)}$ is the payoff matrix of row players, and b is the bias of the neural network.

  That is, all the elements in each row of $H^{(2)}$ equal an positive affine transformation of the maximum element from the corresponding row in $U^{(r)}$.

$$f^{(1)}_i = softmax(H^{(2)})$$

  Therefore, as $c \to \infty$, $f^{(1)}_i \to f^{maxmax}(i)$ as required.
- **Maxmin payoff**. Max Min Payoff is derived similarly to Max Max except with $w_1 = -1$, and $b_1 = b$ where $b \geq max_{i,j} U^{(r)}_{i,j}$; we remain $w_2 = c$ as some large positive constant.Then $H^{(1)}$ reduces to,

$$H^{(1)} = relu(-U^{(r)} + b)$$
$$H^{(2)} = relu(cH^{(1)}_r)$$
$$f^{(1)}_i = softmax(H^{(2)})$$

  Therefore, as $c \to \infty$, $f^{(1)}_i \to f^{maxmin}(i)$ as required.

- **Minmax regret**. Let $b_1 = 0$, and we keep $w_2 = c$ as some large positive constant.Then $H^{(1)}$ reduces to,

$$H^{(1)} = relu(U_c^{(r)} - U^{(r)})$$

Where $U_c^{(r)}$ is the maximum column pooling of the output of $U^{(r)}$.

$$H^{(2)} = relu(cH_r^{(1)})$$

$$f_i^{(1)} = softmax(H^{(2)})$$

Therefore, as $c \to \infty$, $f_i^{(1)} \to f^{minmax}(i)$ as required.

- **Minmin unfairness**.

$$H^{(1)} = \left| U^{(r)} - U^{(c)} \right|$$

Which gives us a measure of "unfairness" as the absolute difference between the two payoffs.

$$f_i^{(1)} = softmax(cH_r^{(1)})$$

Therefore, as $c \to \infty$, $f_i^{(1)} \to f^{minmin}(i)$ as required.

- **Maxmax welfare**.

$$H^{(1)} = \left| U^{(r)} + U^{(c)} \right|$$

Which represent the overall welfare as the absolute sum between the two payoffs.

$$f_i^{(1)} = softmax(cH_r^{(1)})$$

Therefore, as $c \to \infty$, $f_i^{(1)} \to f^{maxmax}(i)$ as required.

A.2 Game Theoretic Properties

Three game theoretic properties of dominance solvability are as follows (Osborne et al., 2004):

- Weak dominance solvable. Dominant strategies are considered as better than other strategies, no matter what other players might do. In game theory, there are two kinds of strategic dominance. One is weakly dominance, of which a strategy is that provides at least the same utility for all the other player's strategies, and strictly greater for some strategy.
- Strict dominance solvable. The other kind of strategic dominance is strictly dominance, of which a strategy is that always provides greater utility to a player, no matter what the other player's strategy is.
- Not dominance solvable. Not dominance solvable means there is no strictly or weakly dominant strategy dominance solvable for a game.

Three game theoretic properties of equilibrium structure are as follows (Osborne et al., 2004):

- Single Nash equilibrium, which is pure. A Pure strategy Nash equilibrium is an action with the property that no single player i can obtain a higher payoff by choosing an action different from a_i, given every other player i adheres to their choice a_j.
- Single Nash equilibrium, which is mixed. A Mixed strategy Nash equilibrium is a mixed strategy action profile with the property that single player cannot obtain a higher expected payoff according to the player's preference over all such lotteries.
- Multiple Nash equilibria. This features a game with multiple Nash equilibria which are pure or/and mixed.

Table 3: Our datasets consist of 127 unique games, each experiment allows participants to play 3 to 20 games. The "games" column shows how many games of the complete datasets we utilised, and the "num"column represents how many observations were included in each dataset.

Source	(Hartford et al., 2016)		Our work	
	Games	Num	Games	Num
(Stahl II & Wilson, 1994)	10	400	10	400
(Stahl & Wilson, 1995)	12	576	12	576
(Costa-Gomes et al., 2001)	**18**	**1296**	**16**	**1252**
(Goeree & Holt, 2001)	**10**	**500**	**3**	**300**
(Cooper et al., 2003)	**8**	2992	**16**	2992
(Haruvy et al., 2001)	15	869	15	869
(Haruvy & Stahl, 2007)	20	2940	20	2940
(Stahl & Haruvy, 2008)	18	1288	18	1288
(Rogers et al., 2009)	17	1210	17	1210
All 9	128 games, **113 unique**	12071	**127 unique**	11827

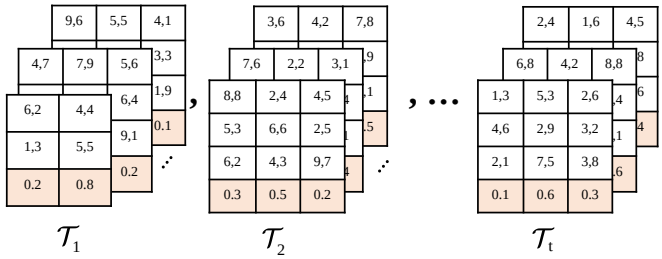


Figure A.1: The normal form game data set diagram. Given such a matrix as input we aim to predict the distribution over the row player's choice of actions defined by the observed frequency of selected actions shown on the bottom.

A.3 DATA

The dataset we collect comes from 9 human experimental experiments that behavioural economists conducted, in which participants made decisions in NFGs. We used the open source code for the FAR model, but collated a slightly different dataset. The payoff matrices of 2×4 in (Costa-Gomes et al., 2001) and (Goeree & Holt, 2001) are not included, hence the FAR model on our dataset performs differently from how it did in the original work (Hartford et al., 2016).

We divided the payoff matrix of both sides of a game into two for our dataset, which increased the number of datasets, but there was a draw. In a symmetric game, the payoff matrices of both sides were the same, however, the number of actions was different. In this case, we kept the payoff matrix belonging to the row player, removed other payoff matrices that were similar to it, and finally reduced the number of data sets from 147 to a unique 127. The comparison between the dataset used in (Hartford et al., 2016) and the one in our work is shown in Table 3.

A.4 HYPER-PARAMETERS AND MORE DETAILS OF THE EMPIRICAL RESULTS

In the mixture of experts structure, the experts do not retain any learnable parameters, only the gating network and meta-learning component have learnable parameters and hyperparameters. Comparing hyperparameters of FAR and MFAR Models, is shown in Table 4.

A.5 THE ARCHITECTURE OF THE AUTO-ENCODER

The auto-encoder is used to extract the main characteristics of the return matrix of both sides of the game. The network architecture and parameters are shown in Table 5.

Table 4: Detailed hyper-parameters of FAR and MFAR models

	FAR	**MFAR**
Feature layers	2 hidden layers	2 gating layers
Feature units	(50, 50)	(50, 50)
Action Response layers	1 layer	1 layer
AR units	50	50
Optimizer	Adam	Adam
Optimizer Learning rate	0.0002	0.0002
β_1	0.9	0.9
β_2	0.99	0.99
Dropout	0.2	/
L1 regularization	0.01	0.01
Training epoch	25000	32500

Table 5: The network structure and hyper-parameters of auto-encoder

Hyper-parameters	**Value**
Hidden layers	2 layers
Neural units (per layer)	(8,4,2)
Loss function	MSE
Batch size	10
Optimiser	Adam
Learning rate	0.01
Training epoch	500

A.6 Ablation Study

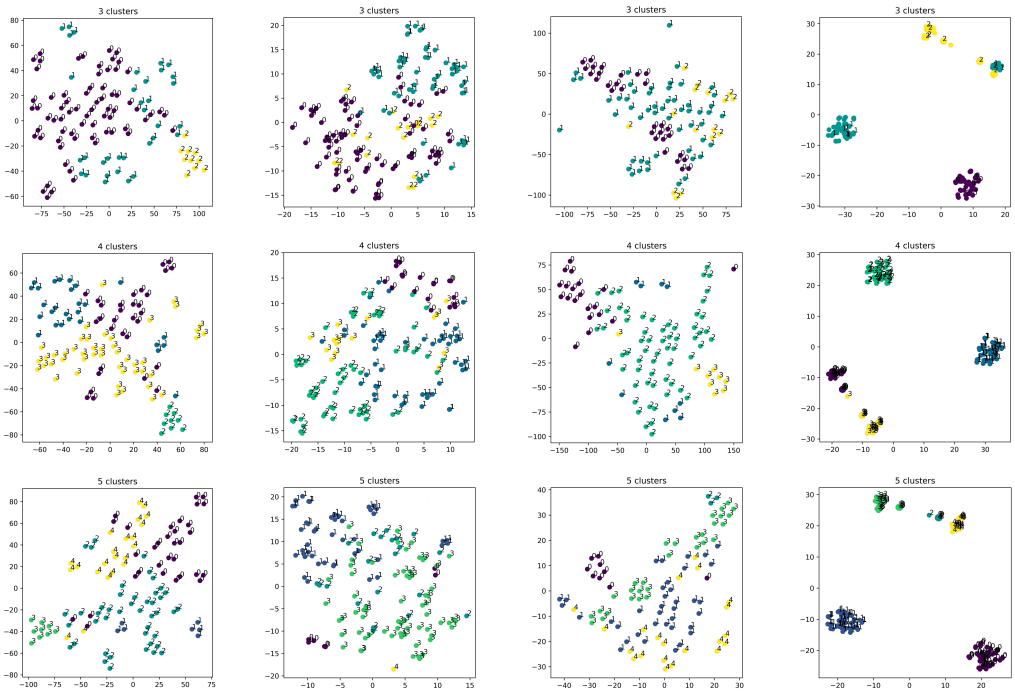
We carried out an extensive ablation study, as shown in Figure A.2, to assess the efficiency of task construction methods. As the output of MF is $h\left(\mathbf{U}^{(c)}\right)=\sum_{k=1}^{K} g\left(\mathbf{U}^{(c)}\right)_k f_k\left(\mathbf{U}^{(c)}\right)$, we have analysed that weights and values of non-strategic human behaviours (i.e., $g\left(\mathbf{U}^{(c)}\right)$ and $\left\{f_k\left(\mathbf{U}^{(c)}\right)\right\}$) of a game can also be used for clustering. Hence, we also tested these two methods and the results are shown in Figure A.2(a) and (b). Each column represents one of the four clustering approaches discussed. Clustering creates a visualisation for 3 to 5 categories of the same work from top to bottom. Clearly, MF outperforms the other three clustering algorithms on all tasks, whereas the others fail to classify the data. However, we discovered a problem with too little data labelled 2 when clustering into 5 classes in the MF clustering findings, which was easily caused by overfitting in MAML training. As a result, clustering into three or four tasks is the optimum option for MF.

To better assess the quality of clustering, we quantified the clustering performance using an internal evaluation index, the DBI index, to make the results more convincing. As the t-SNE visualisation is strongly influenced by the parameter, we conducted ablation experiments on the parameter and used different parameter values for t-SNE visualisation of the data, as shown in Figure A.3, and the results were all consistent with those in the paper, as shown in Table 6.

As shown by Table 6, the clustering results of MF are overall better than the clustering results of the other methods. Although the DBI metric is best when MF is divided into 4 classes, the total number of samples is only 127. Considering that too few samples would affect the training effect, the final model chose MF clustered into 3 classes as the task division for meta-learning.

Table 6: DBI value under different t-SNE parameter settings

Parameter Value	**D**	**NE**	**DC**	**MF-3**	**MF-4**	**MF-5**
10	5.5809336	8.7850499	1.5515862	0.5790317	0.266343	0.3537695
15	4.8725202	8.7397419	1.3158266	0.4650905	0.207548	0.2459401
20	5.1242206	8.8724134	1.0976955	0.499458	0.104965	0.1780445
25	5.7262574	8.8242524	1.1597403	0.4785413	**0.089039**	0.2362738
30	4.741271	8.5563453	1.0422968	0.4269047	**0.087788**	0.3767202



(a) DC (b) Gating Out (c) Experts Out (d) MF

Figure A.2: Ablation analyses clustering results of four task construction methods. 2D t-SNE exhibit distinct clusters that correspond to performance where deep clustering with payoff matrix (DC), clustering with the gating out, clustering with experts out, and clustering with mixture of features (MF).

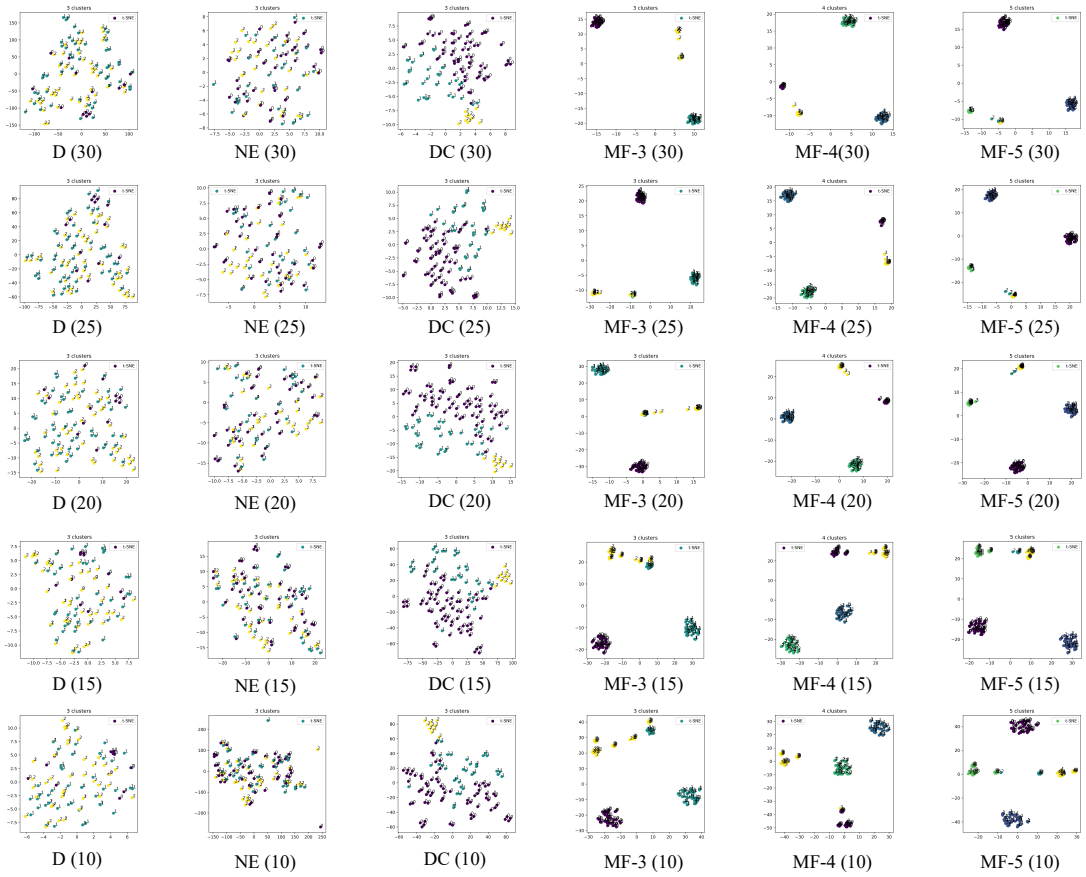


Figure A.3: Clustering results under different t-SNE parameter settings