# Game Theory Meets Statistical Mechanics in Deep Learning Design

Djamel BOUCHAFFRA<sup>1</sup>, Fayçal YKHLEF<sup>2</sup>, Bilal FAYE<sup>3</sup>, Hanane AZZAG<sup>4</sup>, Mustapha LEBBAH<sup>5</sup> djamel.bouchaffra@gmail.com, ykhlef.faycal@gmail.com, faye@lipn.univ-paris13.fr, azzag@univ-paris13.fr, mustapha.lebbah@uvsq.fr

Abstract—We present a novel deep graphical representation that seamlessly merges principles of game theory with laws of statistical mechanics. It performs feature extraction, dimensionality reduction, and pattern classification within a single learning framework. Our approach draws an analogy between neurons in a network and players in a game theory model. Furthermore, each neuron viewed as a classical particle (subject to statistical physics' laws) is mapped to a set of actions representing specific activation value, and neural network layers are conceptualized as games in a sequential cooperative game theory setting. The feed-forward process in deep learning is interpreted as a sequential game, where each game comprises a set of players. During training, neurons are iteratively evaluated and filtered based on their contributions to a payoff function, which is quantified using the Shapley value driven by an energy function. Each set of neurons that significantly contributes to the payoff function forms a strong coalition. These neurons are the only ones permitted to propagate the information forward to the next layers. We applied this methodology to the task of facial age estimation and gender classification. Experimental results demonstrate that our approach outperforms both multi-layer perceptron and convolutional neural network models in terms of efficiency and accuracy.

#### I. Introduction

Deep learning (DL) based on graphical representations has proven effective, especially when domain-specific knowledge for feature extraction is limited [1]. For instance, DL models have demonstrated high performance in complex tasks such as medical image classification [2] [3], natural language processing, and speech recognition [4] [5] [6]. However, these deep learning models often function as black boxes, delivering impressive results in data classification without providing insights into understanding the model's internal workings as well as unraveling the causal mechanisms underlying their predictions [7] [8]. This lack of interpretability and explainability, essentially the ability to comprehend and trace cause and effect within a system, limits their applicability in domains they were not specifically trained for. In terms of their structural design, DL models exhibit several critical limitations: (i) they process information sequentially from one layer to the next without formally evaluating the individual contribution of each neuron, (ii) they have difficulty determining activation levels associated with groups of neurons within a layer, (iii) they struggle to identify the most informative neurons within a layer, often relying on random dropout techniques to mitigate noise and reduce overfitting, and (iv) most of the DL models lack probabilistic measures to express information uncertainty.

To address these limitations, several approaches exploiting game theory (GT) were proposed in the deep machine learning literature [9] [10] [11] [12] [13]. Likewise, [14] relied on game theory to improve prediction in ensemble learning. They defined the pruned ensemble as the minimal winning coalition made of the members that together exhibit moderate diversity. Moreover, [15] explored unlearnable example attacks using a game-theoretic approach, where the attack is modeled as a nonzero-sum Stackelberg game. [16] introduced an efficient approach that leverages a combination of deep learning techniques and game theory to enhance the performance and scalability of solving extensive-form games. These games, characterized by their complex decision-making processes with latent information, pose significant challenges in strategic planning. The research by [17] addresses the challenge of robots finding optimal paths while avoiding collisions with humans and other robots. Traditional Deep Reinforcement Learning (DRL) struggles with slow convergence in such complex scenarios. To improve performance, the study introduces a hybrid approach that integrates DRL with game theory. Furthermore, [11] have demonstrated that a deep neural network can be modeled as a non-atomic congestion game, irrespective of whether it is fully connected or only locally connected. Additionally, they have proved that optimizing the weight and bias vectors for a given training set is equivalent to finding the optimal solution for the associated non-atomic congestion game. Other applications of game theory to deep neural networks can be found in [18] and [19].

A different front emanates from the field of statistical mechanics (SM) has been investigated in order to gain insight into the understanding and optimization of deep learning models [20] [21]. [22] applied mean-field theories to analyze the information propagation in neural networks, which helps identify the 'edge of chaos' and dynamic isometry conditions for optimal learning and generalization. These theories provide a framework for initializing neural networks in a way that maximizes mutual information, enhancing their performance from the start. In the context of continual learning, statistical mechanics offers insights through the development of variational principles and mean-field potentials.

Our main contribution in this manuscript is fourfold:

 Fusion of GT and SM: A seamless combination of game theory and statistical mechanics in deep learning design is applied. In this setting that we propose under the name of 'NEUROGAME', the collaboration between neurons within layers in a neural network is grounded in game theory driven by statistical mechanics laws

- **Probabilistic Signal Transmission**: The flow of information, with a Gibbs distribution value, is propagated across layers in the network.
- Cortical Activation: A neuronal region of activation within the network is described as a coalition of players—connected neurons cooperating to optimize the payoff function.
- Information Filtering and Model Regularization: The coalition with the maximum payoff is deemed the winning coalition, and the contribution of each neuron within this coalition is quantified using the Shapley value. Neurons with high contributions form a strong coalition, and only these neurons transmit information forward to the next layer. In this very phase, some neurons are dropped out to a achieve a dynamic model regularization.

#### II. SOME BASICS OF GAME THEORY

The following definitions are essential to grasp some knowledge about game theory.

#### A. Simple Games

To gain insight into the proposed methodology, we introduce some principles related to game theory, focusing on the concepts of simple games and cooperative sequential games [23] [24] [25]. A simple game involves a set of n players; a set of strategies  $s_i \in S_i$  (possible actions) associated with each player  $i \in \mathcal{N} = \{1, 2, \dots, n\},\$ where  $s = (s_1, s_2, ..., s_n) \in \mathcal{S} = (\mathcal{S}_1 \times \mathcal{S}_2 \times ... \times \mathcal{S}_n)$  $\mathcal{S}_n$ ) is a set of pure strategy profiles; a set of payoffs (real values)  $v_i(s_1, s_2, \ldots, s_n) \in \mathbb{R} \ (v_i : \mathcal{S} \longrightarrow \mathbb{R})$ assigned to each player i for every possible list of strategy choices—where strategies translate into outcomes and each player has preferences over these outcomes represented by their payoffs—and a level of information or belief, which encompasses what players know and believe about the situation and one another, and what actions they observe before making decisions. The game is finite if S is finite.

#### B. Notion of Simple Coalition

We define the concept of coalition and the contribution of each player within this coalition. A simple coalition is a group of players  $\mathcal{C} \subset \mathcal{N}$  that cooperate to achieve a common goal. The set  $\mathcal{N}$  is often referred to as the grand coalition. Every coalition  $\mathcal{C}$  has a set of actions. If the payoffs  $v(\mathcal{C})$  associated with a coalition  $\mathcal{C}$  are freely redistributed among its members, this condition is known as the Transferable Utility Assumption (TUA). A coalitional game with transferable utility is a pair  $(\mathcal{N}, v)$ , in which  $\mathcal{N}$  is a finite set of players, and  $v: 2^{\mathcal{N}} \longrightarrow \mathbb{R}$  maps each coalition  $\mathcal{C}$  to a real-valued payoff function  $v(\mathcal{C})$  that the coalition members can distribute among themselves. We assume that  $v(\emptyset) = 0$ . Given a coalitional game  $(\mathcal{N}, v)$ , the

Shapley value associated with player  $i \in \mathcal{N}$  is given by:

$$\phi_i(\mathcal{N}, v) = \frac{1}{n!} \sum_{\mathcal{C} \subseteq \mathcal{N} \setminus \{i\}} |\mathcal{C}|! (n - |\mathcal{C}| - 1)! [v(\mathcal{C} \cup \{i\}) - v(\mathcal{C})].$$

$$\tag{1}$$

The Shapley value expresses the average marginal contribution of player i, averaging over all different coalitions with respect to which the grand coalition can be built starting from the empty one [9].

# III. NEUROGAME: GAME THEORY MEETS STATISTICAL MECHANICS

We present in this section the analogy between conventional DL and NEUROGAME, as well as the description of all the components needed to fully comprehend how this proposed deep learning model operates.

# A. Comparison between Conventional Deep Learning and NEUROGAME

we make the following correspondence between cooperative game theory and deep learning representation:

- 1) A layer of a deep neural network represents a game.
- A neuron in a layer of a deep neural network represents a player of the game. Neurons are viewed as particles interacting via statistical mechanics laws.
- 3) Each neuron is mapped to a set of actions representing its current state (a specific interval of neuron activation values). This set of actions acts as its strategy.
- 4) An input to the deep neural network structure corresponds to the information (or observation within the environment) that is available at any time of the game. In our setting, an input is a 2D image.
- 5) A neuronal region depicts a group of connected neurons  $(s_1, \ldots, s_n)$  that are located within a certain neighborhood in the cerebral cortex; it constitutes a simple coalition of players.
- 6) A payoff  $v_i(s_1, \ldots, s_n)$  assigned to this coalition expresses the worth of the actions exhibited by all players forming this coalition. This function is conveyed through the energy function assigned to a tuple of activations of neurons. This tuple is called a configuration state of the coalition. The coalitions with high payoffs are sought: They represent the winning coalitions. The payoff function reflects the quality level of the information available.
- 7) The contribution of a neuron within each winning coalition is expressed by its Shapley value expressed through equation (1). Neurons with high Shapley values are members of strong coalitions. Only strong coalitions, extracted from the winning coalitions, are permitted to forward the flow of information from one layer to the next.

#### B. Computation of a Coalition Payoff

This section describes the relationship between the energy function, the Gibbs distribution and the payoff (also known as utility) function assigned to a coalition. The computation of all three functions requires a definition of a neighborhood system between neurons that compose a coalition.

1) Neurons Neighborhood System:: For the sake of illustration and without loss of generality, we focus on neighbors of a neuron within a (3,3) neuronal grid. A (3,3) neighborhood system  $\mathcal{H}$  of a neuron located at coordinates (i,j) is the set  $\{(i-1,j), (i-1,j+1), (i,j+1), (i+1,j+1), (i+1$ 1), (i+1,j), (i+1,j-1), (i,j-1), (i-1,j-1)}.

This neighborhood system is needed during the clique structure used by the energy function.

Definition 1: A set of random variables is a Gibbs random field (GRF) on a set  $\Omega$  with respect to a neighborhood system  $\mathcal{H}$  if and only if its configuration obeys a Gibbs distribution. We now introduce the notion of configuration state that is needed in the evaluation of the energy and payoff functions.

Definition 2: A configuration state assigned to a simple coalition is a sequence of activation values of neurons that form this coalition.

This configuration state is denoted:  $\omega_i = (a^i_{s_1}, \dots, a^i_{s_n})$ where:  $a_{s_i}^i$  is the neuron activation value at location  $s_j$  in the coalition i and n is the size of the coalition.

2) Gibbs Distribution of a Configuration State:: During a regression or classification task, we aim for the activation of neurons to progressively increase from the first layer to the last layer in a deep neural network. This behavior is compatible with the energy minimization principle. The Gibbs (or Boltzmann) distribution relies on the energy function assigned to the i-th configuration state.

Definition 3: The Gibbs distribution function is defined as:

$$P(\omega_{i},T) = \frac{1}{Q}e^{\frac{-E(\omega_{i})}{k_{B}\times T}} = \frac{e^{\frac{-E(\omega_{i})}{k_{B}\times T}}}{\sum_{j=1}^{j=M}e^{\frac{-E(\omega_{j})}{k_{B}\times T}}},$$
 (2)
•  $P(\omega_{i},T)$  is the probability of the  $i$ -th configuration state

- at temperature T,
- $E(\omega_i)$  is the energy of the *i*-th configuration state,
- $k_B$  represents the Boltzmann constant  $(k_B \approx 1.38 \times$  $10^{-23}$ ),
- T is the temperature of the system,
- M denotes the number of all configuration states associated to all simple coalitions within a layer,
- $\bullet$  Q is the canonical partition function (normalizing factor).

This distribution shows that configuration states with lower energy will always be assigned a higher probability of being occupied than those with higher energy. However, the energy assigned to a configuration state is defined via a potential function expressed through the Ising model using bonding strengths (synaptic links) between neurons in a lattice structure. This energy, which is a Hamiltonian function, is therefore expressed as:

$$E(\omega) = \sum_{\langle p,q \rangle} b_{pq} \times \left(\frac{1}{a_p \times a_q}\right) + \sum_p f_p \times \left(\frac{1}{a_p}\right), \quad (3)$$

ullet  $b_{pq}$  is the bonding strength between two neighbor neurons p and q,

- $\bullet$   $f_q$  is the external magnetic field interacting with the
- ullet  $a_p$ , and  $a_q$  are non-zero activation values assigned to neuron p and q, respectively,
- $\bullet$  < p, q > is a pair of neighbor neurons.

If we set  $f_p=\alpha$  and  $b_{pq}=\beta$ , therefore the Ising model expressed via equation 3 can be rewritten as follows:

$$E(\omega) = \alpha \sum_{p} \left( \frac{1}{a_p} \right) + \beta \sum_{q \in \mathcal{H}(p)} \left( \frac{1}{a_p \times a_q} \right), \forall p \qquad (4)$$

where  $\mathcal{H}(p)$  is a (m,n) neighborhood system. The second summation is over pairs of neighboring neurons. The energy decreases when the activation values in a configuration state are high. In other words, a smaller energy means a higher neuronal activation. However, we consider the temperature T as dependent on the iteration number i during the training of NEUROGAME. It is expressed as follows:

$$T(i) = \frac{c \times 10^{23}}{\ln(1+i)},\tag{5}$$

where the numerator is a large constant value that ensures a high temperature at initialization. Therefore, using equation 2, one can compute the Gibbs distribution  $P(\omega_i, T)$ assigned to each configuration state.

- 3) Generation of Configurations States:: In order to compute the Gibbs distribution, one has to estimate the normalizing term that requires M configuration states. The set of configuration states contained in one layer is built using a grouping (set of neurons acting together) containing neurons that are close to each other. An element of this grouping can be a  $4\times4$  (or  $5\times5$ ) grid of neurons. A simple coalition in a layer is composed of neurons that are nearby with respect to a distance measure. We generate through this partitioning process M configuration states with different levels of neuron activations. Moreover, each configuration state is assigned a Gibbs distribution value.
- 4) Layer Neighborhood System:: We show in this step how a neighborhood system (a lattice structure) can be constructed in order to compute the energy associated to the Ising model. A neighborhood system is based on a metric (or distance) between neurons of a layer. This set of neighbors associated to neuron (i, j) is composed of the sites  $\{(i, j-1), (i-1, j), (i, j+1), (i+1, j)\}.$
- 5) The Payoff Function:: Since we are in the context of a collaborative game theory, therefore, the contribution of a group of players should induce a higher payoff than the one incurred by a single player within a simple coalition. Furthermore, a maximum payoff value should be assigned a minimum energy value. Using Boltzmann's distribution, this minimum energy value corresponds to a maximum Gibbs distribution value. We now define the payoff as being proportional to the Gibbs distribution:

Definition 4: The payoff function assigned to a simple coalition is expressed as follows:

Payoff
$$(\omega_i, T) = \ln\left(\frac{k_1}{1 - P(\omega_i, T)}\right),$$
 (6)

where  $k_1$  is a positive control parameter and the natural logarithm is applied to smooth this function.

One can notice that a high payoff corresponds to a low neuronal energy value. Neurons are supposed to behave as microscopic physical particles interacting seamlessly. The configuration state  $\omega^*$  with the maximum Payoff value is assigned the winning coalition among all simple coalition associated to the M possible configuration states.

Definition 5: A configuration state  $\omega^*$  with a maximum Payoff value is associated to a winning coalition among all possible simple coalitions.

The Payoff value represents the worth of the winning coalition. It tallies the total expected sum of payoffs the members of this coalition can gain by cooperating. However, instead of considering only one winning coalition, a set of p winning coalitions derived from p top choices of payoff values are considered.

6) The Concept of Strong Coalition:: The payoff value assigned to a configuration state of is needed during the Shapley value computation. This payoff corresponds to the utility function v used in the Shapley function expressed by equation 1. This payoff function requires the computation of:  $v(\mathcal{C} \cup \{i\}) - v(\mathcal{C})$ , which is the leading term in the Shapley value computation, associated to player i, denoted  $\phi_i(\mathcal{N}, v) = \text{Payoff}$ . This leading term corresponds to:

$$Payoff(\mathcal{C} \cup \{i\}) - Payoff(\mathcal{C}), \forall \mathcal{C} \subseteq (\mathcal{N} \setminus \{i\}). \tag{7}$$

The summation used in equation 1 consists in extracting all subsets  $\mathcal{C}$  from the simple coalition  $\mathcal{N}$  (set of players) that do not contain player i. The number of these subsets is  $2^{\mathcal{N}-1}$ . However, among all subsets, only those subsets whose cardinalities are greater or equal than 2 are considered, since the singletons do not form coalitions. Once the winning coalition is identified, members of this coalition who contributed most to the payoff are maintained; the other members with low contributions are dropped out. This regularization technique that is not based on randomness represents one key feature of novelty exhibited by NEUROGAME.

Definition 6: A strong coalition is composed of all neurons whose Shapley values are greater than a threshold value  $\rho$ .

Neurons with a high payoff (or high Gibbs distribution) are those that form the strong coalition. The threshold  $\rho$  is dynamic; it involves the contribution of each neuron forming a coalition within a network layer and the iteration number i: it is expressed specifically via the following function:

$$\rho(S_{c_j}^i, i) = Q_1(S_{c_j}^i) \times \ln(1+i). \tag{8}$$

The function 'ln' represents the natural logarithm, while  $Q_1(S_{c_j}^i)$  denotes the first quartile of the set S of Shapley values (sorted in ascending order) assigned to the set of neurons forming a winning coalition  $c_j$  for  $j=1,\ldots,p$ , with each coalition being of size n. If n is the number of these values, therefore, this first quartile is equal to (n+1)/4; it indicates that 25% of the data are below this point.

Theorem 1 (Shapley Threshold Behavior): For large values of n (coalition size) and i (iteration number during training), the function  $\rho(S_{c_j}^i,i)$  will tend to increase, with the growth rate influenced by ln(1+i).

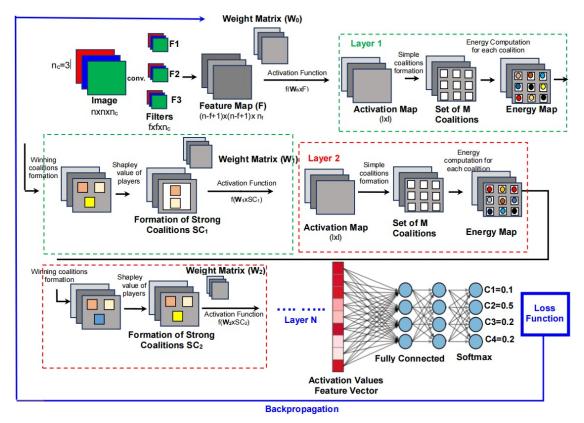
Proof: If i increases during training, the natural logarithm function ln(i) grows without bound, but it does so very slowly compared to linear functions. Therefore, ln(1+i) will continue to increase, but at a gradually slowing, logarithmic rate. However,  $Q_1(S^i_{c_j})$  depends on the Shapley values distribution. As we increase the coalition size n, the value of  $Q_1(S^i_{c_j})$  does not necessarily increase. It reflects the position within the ordered data rather than growing unbounded. Finally, the combined effect on  $\rho(S^i_{c_j},i)$  will be affected by the product of these two functions:  $Q_1(S^i_{c_j})$  and ln(1+i). In conclusion, the primary driver of the behavior of the threshold  $\rho(S^i_{c_j},i)$  for large i and n will be ln(1+i).

It is worth noting that as NEUROGAME learns, the selected coalitions grow progressively stronger.

#### C. The Different Phases in NEUROGAME

The following sequence of operations describes NEUROGAME:

- 1) The input is a colored (or grey-level) image with its three colors components red, green, blue ( $n_c = 3$ ), with a dimension equal to (n×n) for each color: ( $n \times n \times n_c$ ).
- 2) A convolution operation with three filters  $F_1$ ,  $F_2$  and  $F_3$ , each one with a dimension equal to  $f \times f$  is subsequently applied, to each color  $(f \times f \times n_c)$ . An arithmetic mean value is computed for each element of the three colored matrices after convolution.
- 3) The results of the convolution between filters and the image is represented by three feature map matrices with dimension  $(n f + 1) \times (n f + 1) \times n_c$ .
- 4) An activation function is applied to the product of the feature map matrices and the first weight matrix  $W_1$ , and the result is stored as the three activation map matrices with a predefined dimension  $(l \times l)$ .
- 5) Generation of M simple coalitions within each of the the three activation map matrices of the first layer. The value of M is equal to the dimension of a layer divided by n. Therefore, each simple coalition has a size of n neurons and is assigned a configuration state (activation values).
- 6) Computation of the energy value for each simple coalition (configuration state) within an activation map matrix. The set of energy values within an activation map of the first layer forms an energy map. Therefore, we obtained three energy maps.
- Selection of p-top choices simple coalitions given their payoff values. They are the p winning coalitions of each energy map.
- 8) Extraction of the strong coalitions amongst the winning ones. Neurons of the winning coalitions whose Shapley values exceed the threshold value  $\rho(S_{c_i}^l,i)$  are



**Fig. 1:** The training procedure of NEUROGAME showing the passage from *M simple coalitions* to *p winning coalitions* and then to *p strong coalitions* generated via the Shapley filtering process. The computation of the strong coalitions (integrated into a fully connected neural network) is repeated across all *k* layers until NEUROGAME converges. The feature vector extracted at this convergence point is composed of activation values of the last optimal strong coalitions.

- maintained and neurons with Shapley values that fall under this threshold value are removed.
- 9) This entire process continues during the first training iteration until reaching the last layer k. The activation values of the strong coalitions corresponding to the three energy maps are concatenated to form a feature vector assigned to the input image.
- 10) This feature vector is subsequently fed to a fully connected neural network with k hidden layers.
- 11) The Softmax operation is applied for the evaluation of the loss function during training.
- 12) All weights are updated using the opposite direction of the gradient of the loss function.

Figure 1 illustrates the NEUROGAME training procedure when the observation input is an image and the number of labels for a classification task is four  $(C_1, C_2, C_3, C_4)$ .

## D. NEUROGAME Layers and Information Propagation

In this section, we show how a layer of NEUROGAME is built. We also describe how the nformative signals are communicated to the next layer during training of the entire deep neural network.

1) NEUROGAME Layer:: A layer in this proposed deep neural network is composed of five components: (i) activation maps, (ii) a set of M coalitions, (iii) a set of energy maps, (iv)

a set of winning coalitions, and (v) a set of strong coalitions (refer to Figure 1).

2) Transmission of the Information:: The most informative signals generated from neurons pertaining to the strong coalitions (those that passed the  $\rho$  test) of layer l are forwarded to neurons of layer (l+1). In fact, these signals represent the image by the activation function of the product of two quantities: (i) The activation value of a neuron within a strong coalition in layer l, and (ii) the weight (synaptic link) that connects this neuron to a specific neuron of layer (l+1). These two quantities are the ones involved during a forward transmission of information during NEUROGAME training-based on backpropagation.

#### IV. EXPERIMENT

To demonstrate the effectiveness of the proposed methodology, we have performed two different classification tasks: 1) gender classification, and 2) simultaneous age and gender classification.

#### A. Datasets and Architecture

To assess NEUROGAME's performance, we used two benchmarked datasets designed for distinct classification tasks: CelebA (CelebFaces Attributes) [26] dataset for gender classification and UTKFace dataset [27] for age and gender classification concurrently. We now present

the architectures of the two baseline models for gender classification and simultaneous age and gender classification, alongside comparisons with our proposed NEUROGAME method.

#### Gender Classification: Multi-Layer Perceptron (MLP):

- Input: Images of size (64, 64, 3).
- Layers: Flattened input followed by dense layers (256 units ReLU, Batch-Normalization, Dropout; 128 units ReLU, Batch-Normalization, Dropout; 64 units ReLU, Batch-Normalization, Dropout).
- Output: Single unit with sigmoid activation for binary classification.

For comparison with NEUROGAME, a single NEUROGAME layer is added with a coalition size of (2,2) and a top-p value of 0.85.

# Simultaneous Gender and Age Classification: Convolutional Neural Network (CNN):

- Input: Images of size (128, 128, 1).
- Layers: Four Conv2D layers (32, 64, 128, 256 filters with (3, 3) kernels and ReLU activation), followed by max-pooling (2, 2).
- Flatten and two dense layers (256 units each, ReLU activation), each followed by a Dropout layer.
- Outputs: Gender (sigmoid activation), Age (ReLU activation).

For comparison, Conv2D layers are replaced with NEUROGAME layers (top-p=0.85, coalition size=(2,2)) to evaluate NEUROGAME's performance in classification tasks. In all NEUROGAME models, we applied a Convolutional layer with three filters to generate feature maps.

## B. Hyperparameter Tuning for NEUROGAME

To determine the optimal hyperparameter values for NEUROGAME, extensive experimentation was carried out. The results showed that the most effective configuration was achieved by setting  $\alpha$  to 0 and  $\beta$  to 1 (refer to Equation 4). With  $\alpha=0$ , the model's energy is determined exclusively through the interactions between neighboring neurons, thereby simplifying the system by excluding the contributions of individual neurons. Setting  $\beta=1$  preserves the original form of neighbor interactions without any additional weighting. For temperature estimation, a value of c=1 was found to be optimal (refer to Equation 5), while  $k_1=1$  was determined to be the best setting for the payoff calculation (refer to Equation 6).

## C. Gender Classification

For gender classification, we applied data augmentation through random cropping and horizontal flipping to increase training diversity and model robustness. The images were normalized by scaling pixel values to [0, 1] and dividing by 255. Models were trained with a batch size of 64 using Adam optimizer [28] and binary cross-entropy loss. Figure 2 shows that NEUROGAME achieved more effective reduction in loss

and better generalization, with a test loss of 0.2645 compared to MLP's 0.4335, as highlighted in Table I, demonstrating NEUROGAME's superior performance.

Model	Test Loss	Test Accuracy (%)
MLP	0.4335	80.19
NEUROGAME	0.2645	88.26

**TABLE 1:** Test performance comparison between MLP and NEUROGAME models on CelebA dataset.

Furthermore, the test accuracy of NEUROGAME is 88.26%, substantially higher than MLP model's test accuracy of 80.19%. This improvement in accuracy demonstrates NEUROGAME's superior capability in generalizing from the training data to unseen data, confirming that the incorporation of NEUROGAME's specialized layer results in enhanced model performance and reliability. To further assess the efficacy of NEUROGAME, we expanded our investigation to include in the next section a more intricate task: simultaneous classification of age and gender.

## D. Simultaneous Age and Gender Classification

In age and gender classification, we use two classifiers: the first focuses on gender with binary cross-entropy loss and accuracy as the metric, while the second predicts age as a continuous value using mean absolute error as the loss function. During inference, a correct age class prediction is considered a success. Both models are optimized using the Adam optimizer with a batch size of 32 and trained for 100 epochs without data augmentation. This setup thoroughly assesses the performance and robustness of CNN and NEUROGAME in age and gender classification.

Figures 3 and 4 compare CNN and NEUROGAME

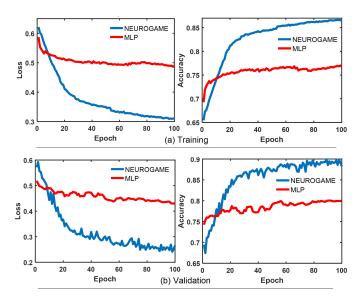


Fig. 2: Comparison of training and validation losses and accuracies between MLP and NEUROGAME models. NEUROGAME shows better generalization performance, as evidenced by the lower validation loss and improved validation metrics.

in terms of training and validation performance for

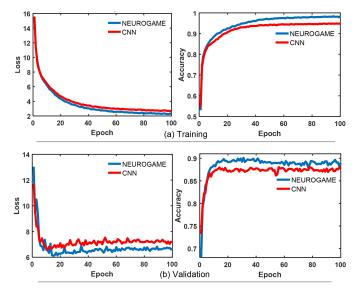
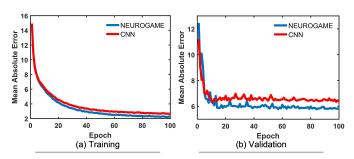


Fig. 3: Comparison of training and validation performance between CNN and NEUROGAME models for gender classification.



**Fig. 4:** Comparison of training and validation performance between CNN and NEUROGAME for age classification.

gender and age classification. NEUROGAME consistently outperforms CNN, demonstrating superior generalization with lower validation loss and better metrics. Precision was computed for each model across gender, age, and combined classifications. NEUROGAME, with fewer parameters, shows higher average precision than CNN. As shown in Table II, NEUROGAME achieves higher precision across all age groups in gender classification and generally leads in age classification. This was validated on UTKFace test set, where NEUROGAME maintained higher precision, especially in younger and middle age categories, highlighting its robustness in multitask learning.

Due to limited data in this age group, CNN model predicts an age of 93 years, while NEUROGAME predicts 101 years, closer to the ground truth, showing superior generalization. This indicates NEUROGAME's better handling of sparse data compared to CNN. The image was randomly selected, underscoring NEUROGAME's robustness and reliability. The two experiments highlight the effectiveness of NEUROGAME, especially in classification tasks, when compared to well-established ML models. Indeed, NEUROGAME has outperformed both MLP and CNN models in gender classification as well as in the simultaneous

Class	Gender		Age		Gender and Age	
	CNN	NEUROGAME	CNN	NEUROGAME	CNN	NEUROGAME
[0, 2]	89.61	91.63	85.60	69.96	79.31	64.11
[3, 6]	95.35	96.63	79.07	81.43	76.37	77.61
[7, 12]	93.92	96.03	72.68	74.75	71.27	73.44
[13, 17]	96.00	95.03	57.92	61.09	57.43	64.40
[18, 22]	97.62	98.04	60.93	64.18	60.07	63.46
[23, 26]	98.60	98.75	53.25	58.25	52.70	57.73
[27, 33]	98.41	98.46	74.45	71.36	73.69	70.63
[34, 44]	98.79	98.65	76.52	70.97	75.97	70.39
[45, 59]	98.33	97.81	77.28	73.00	76.87	72.41
[60, 69]	98.32	98.93	64.19	63.96	64.04	63.73
[70, 79]	95.00	98.05	58.80	62.66	58.23	62.09
[80, 89]	98.97	97.88	50.79	59.13	50.40	57.54
[90, 99]	95.35	96.30	36.50	55.47	35.04	54.74
[100, 116]	100.00	100.00	46.88	53.13	46.88	53.13

**TABLE II:** precisions (%) of CNN and NEUROGAME on UTKFace test set across age and gender categories.

classification of gender and age.

#### V. CONCLUSION AND PERSPECTIVES

We have developed a novel DL architecture, NEUROGAME, which integrates game theory and statistical physics principles. This allows neurons in the same layer to collaborate using the Shapley value function to assign contribution scores and perform controlled dropout, reducing overfitting. This Shapley-based regularization enhances network robustness and provides transparency within the architecture, functioning as a *glass-box framework*.

Comparative studies show NEUROGAME outperforms MLP and CNN in gender and joint gender-age classification, showing better generalization and accuracy. This research signals a paradigm shift in deep learning, paving the way for more interpretable, efficient, and effective neural networks. As a perspective, we will explore *the Banzhaf power index* to assess the influence of neuronal states in prediction tasks, potentially improving model generalization further.

#### REFERENCES

- R. Archana and P. Jeevaraj, "Deep learning models for digital image processing: a review," *Artificial Intelligence Review*, vol. 57, no. 11, pp. 1–23, 2024.
- [2] R. Kumar, P. Kumbharkar, S. Vanam, and S. Sharma, "Medical images classification using deep learning: a survey," *Multimedia Tools and Applications*, vol. 83, no. 7, 2024.
- [3] S. Bamber and T. Vishvakarma, "Medical image classification for alzheimer's using a deep learning approach," *Journal of Engineering* and Applied Science, vol. 70, p. 54, 2023.
- [4] J. S. Chou, P. L. Chong, and C. Y. Liu, "Deep learning-based chatbot by natural language processing for supportive risk management in river dredging projects," *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107744, 2024.
- [5] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Information Fusion*, p. 102422, 2024.
- [6] A. M. Hashan, C. R. Dmitrievich, M. A. Valerievich, D. D. Vasilyevich, K. N. Alexandrovich, and B. B. Andreevich, "Deep learning based speech recognition for hyperkinetic dysarthria disorder," in 2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), pp. 012–015, 2024.
- [7] J. Peters, D. Janzing, and B. Schlkopf, *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.
- [8] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.
- [9] M. Maschler, E. Solan, and S. Zamir, Game Theory. Cambridge University Press, 2nd ed., 2020.

- [10] W. Hu, X. Liu, and Z. Xie, "Ore image segmentation application based on deep learning and game theory," in World science: problems and innovations, pp. 71–76, 2022.
- [11] C. Ren, Z. Wu, D. Xu, and W. Xu, "A game-theoretic analysis of deep neural networks," in Algorithmic Aspects in Information and Management: 15th International Conference, AAIM 2021, Virtual Event, December 20–22, 2021, Proceedings, pp. 369–379, Springer International Publishing, 2021.
- [12] S. Li, X. Hu, and Y. Du, "Deep reinforcement learning and game theory for computation offloading in dynamic edge computing markets," *IEEE Access*, vol. 9, pp. 121456–121466, 2021.
- [13] L. Yin, S. Lin, Z. Sun, R. Li, Y. He, and Z. Hao, "A game-theoretic approach for federated learning: A trade-off among privacy, accuracy and energy," *Digital Communications and Networks*, vol. 10, no. 2, pp. 389–403, 2024.
- [14] H. Ykhlef and D. Bouchaffra, "An efficient ensemble pruning approach based on simple coalitional games," *Information Fusion*, vol. 34, pp. 28–42, March 2017.
- [15] S. Liu, Y. Wang, and X.-S. Gao, "Game-theoretic unlearnable example generator," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 21349–21358, 2024.
- [16] L. Meng, Z. Ge, P. Tian, B. An, and Y. Gao, "An efficient deep reinforcement learning algorithm for solving imperfect information extensive-form games," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 5823–5831, 2023.
- [17] Y. Xing, D. Hou, J. Liu, H. Yuan, A. Verma, and W. Shi, "Deep learning and game theory for ai-enabled human-robot collaboration system design in industry 4.0," in 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), (Las Vegas, NV, USA), pp. 8–13, 2024.
- [18] T. Hazra and K. Anjaria, "Applications of game theory in deep learning: a survey," *Multimedia Tools and Applications*, vol. 81, pp. 8963–8994, 2022.
- [19] T. Hazra, K. Anjaria, A. Bajpai, and A. Kumari, "Applications of game theory in deep neural networks," in *Applications of Game Theory in Deep Learning*, SpringerBriefs in Computer Science, Springer, Cham, 2024.
- [20] M. E. Tuckerman, Statistical Mechanics: Theory and Molecular Simulation. Oxford University Press, 2nd ed., 2023.
- [21] S. Kollmannsberger, D. D'Angella, M. Jokeit, and L. Herrmann, "Deep learning in computational mechanics," in *Deep Learning in Computational Mechanics*, pp. 55–84, Springer International Publishing, 2021.
- [22] C. Li, Z. Huang, W. Zou, and H. Huang, "Statistical mechanics of continual learning: Variational principle and mean-field potential," *Phys. Rev. E*, vol. 108, p. 014309, Jul 2023.
- [23] B. von Stengel, Game Theory Basics. Cambridge University Press, 2021.
- [24] A. Roth, Who Gets What-and Why. HarperCollins, 2015.
- [25] E. Mendelson and D. Zwillinger, Introducing Game Theory and its Applications. CRC Press, 2024.
- [26] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [27] Z. Zhang and Y. Song, "Utkface." https://susanqq.github. io/UTKFace/, 2017.
- [28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.