An Unsupervised Game-Theoretic Approach to Saliency Detection

Yu Zeng, Mengyang Feng, Huchuan Lu, Gang Yang, and Ali Borji

***Abstract*— We propose a novel unsupervised game-theoretic salient object detection algorithm that does not require labeled training data. First, saliency detection problem is formulated as a non-cooperative game, hereinafter referred to as Saliency Game, in which image regions are players who choose to be "background" or "foreground" as their pure strategies. A payoff function is constructed by exploiting multiple cues and combining complementary features. Saliency maps are generated according to each region's strategy in the Nash equilibrium of the proposed Saliency Game. Second, we explore the complementary relationship between color and deep features and propose an iterative random walk algorithm to combine saliency maps produced by the Saliency Game using different features. Iterative random walk allows sharing information across feature spaces, and detecting objects that are otherwise very hard to detect. Extensive experiments over six challenging data sets demonstrate the superiority of our proposed unsupervised algorithm compared with several state-of-the-art supervised algorithms.**

***Index Terms*— Saliency, salient object detection, visual attention.**

I. Introduction

SALIENCY detection is a preprocessing step in computer vision which aims at finding salient objects in an image [1]. Saliency helps allocate computing resources to the most informative striking objects in an image, rather than processing the background. This is very appealing for many computer vision tasks such as object tracking, image and video compression, video summarization, image retrieval and classification. A lot of previous effort has been spent on this problem and has resulted in several methods [2], [3]. Yet, saliency detection in arbitrary images remains to be a very challenging task, in particular over images with several objects amidst high background clutter.

On the one hand, unsupervised methods are usually more economical than supervised ones because no training data is needed. But they usually require a prior hypothesis about salient objects, and their performance heavily depend on reliability of the utilized prior. Take a recently popular label propagation approach as an example (*e.g.* [5]–[8]). First, seeds are selected according to some prior knowledge (*e.g.* boundary background prior), and then, labels are propagated from seeds to unlabeled regions. They work well in most cases, but their results will be inaccurate if the seeds are wrongly chosen. For instance, when image boundary regions are chosen as background seed, the output will be unsatisfactory if the salient objects touch the image boundary (see the first row of Figure 1(d)).

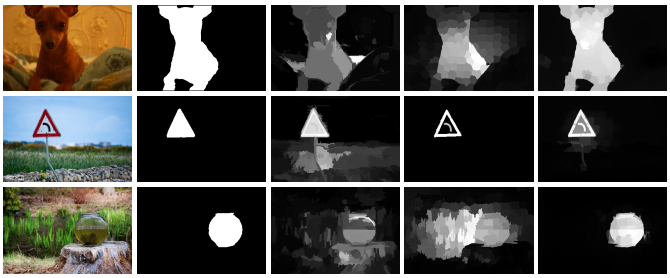

Fig. 1. Saliency detection results by several methods. (a) Input images, (b) Ground truth maps, (c) DRFI method [4], a supervised method based on handcrafted features, (d) MR method [5], an unsupervised method taking image boundary regions as background seeds, (e) Our method.

On the other hand, supervised methods are generally more efficient. Compared with unsupervised methods based on heuristic rules, supervised methods can learn more representative properties of salient objects from numerous training images. The prime example is deep learning based methods [9]–[12]. Owning to their hierarchical architecture, deep neural networks (*e.g.* CNNs [13]) can learn high-level features rich in semantic information. Consequently, these methods are able to detect semantically salient objects in complex backgrounds. However, off-line training a CNN needs a great deal of training data. As a result, using CNNs for saliency detection, although effective, is relatively less economical than unsupervised approaches.

In this paper, we attempt to overcome the aforementioned drawbacks. To begin with, the saliency detection problem is formulated as a Saliency Game among image regions. Our main motivation in formulating saliency and attention in a game-theoretic manner is the very essence of attention which is the competition among objects to enter high level processing. Most previous methods formulate saliency detection as minimizing one single energy function that

Manuscript received July 30, 2017; revised January 18, 2018; accepted April 30, 2018. Date of publication May 21, 2018; date of current version June 15, 2018. This work was supported by the Natural Science Foundation of China under Grant 91538201, Grant 61725202, and Grant 61472060. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guoliang Fan. *(Corresponding author: Huchuan Lu.)*

Y. Zeng, M. Feng, and H. Lu are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: zengyu@mail.dlut.edu.cn; mengyangfeng@gmail.com; lhchuan@dlut.edu.cn).

G. Yang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: yanggang@mail.neu.edu.cn).

A. Borji is with the Center for Research in Computer Vision and the Computer Science Department, University of Central Florida, Orlando, FL 32816 USA (e-mail: aliborji@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2838761

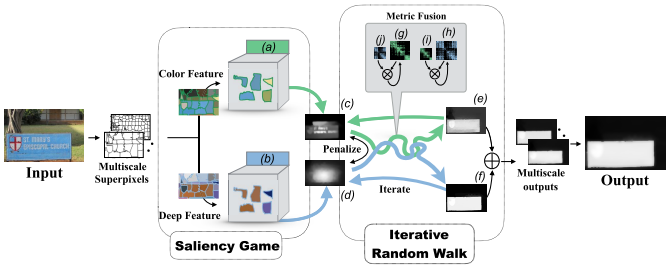

Fig. 2. The pipeline of our algorithm. The input image is segmented into superpixels in several scales. The following process is applied to each scale, and the average of the results in all scales is taken as the final saliency map. First, the *Saliency Game* among superpixels is formulated in two feature spaces ((*a*) and (*b*)), respectively to generate corresponding results ((*c*) and (*d*)). Second, an *Iterative Random Walk* is constructed to refine the results of the Saliency Game. Then outputs (*e*) and (*f*) are summed (with weights) as the result in one scale. In the Iterative Random Walk, complete affinity metric ((*g*) in deep feature space and (*h*) in color space) in one feature space is fused with neighboring affinity metric ((*i*) in color space and (*j*) in deep space) in another feature space. This is done to exploit the complementary relationship between the two feature spaces. Finally, we average the results in all scales to form the final saliency map.

incorporates saliency priors. Different image regions are often considered through adding terms in the energy function (*e.g.* [14]) or sequentially (*e.g.* [5]). If the priors are wrong, optimization of their energy function might lead to wrong results. In contrast, we define one specific payoff function for each superpixel which incorporates multiple cues including spatial position prior, objectness prior, and support from others. In addition, adopting two independent priors makes the proposed method more robust since when one prior is inappropriate, the other might work. The goal of the proposed Saliency Game is to maximize the payoff of each player given other players strategies. This can be regarded as maximizing many competing objective functions simultaneously. The game equilibrium automatically provides a trade-off, so that when some image region can not be assigned a right saliency value by optimizing one objective function (*e.g.* due to misleading prior), optimization of the other objective functions might help to give them a right saliency value. This approach seems very natural for attention modeling and saliency detection, as also features and objects compete to capture our attention.

In addition, it is known that one main factor for the astonishing success of deep neural networks is their powerful ability to learn high-level semantically-rich features. Using features extracted from a pre-trained CNN to build an unsupervised method seems a considerable option, as it allows utilizing the aforementioned strength while avoiding time-consuming training. However, rich semantic information comes with the cost of diluting image features. Therefore, we also use traditional color features as supplementary information. To make full use of the two complementary features for better detection results, we avoid directly taking the weighted sum of the raw results generated by the above Saliency Game in the two feature spaces. Instead, we further propose an Iterative Random Walk algorithm across two feature spaces, deep features and the traditional CIE-Lab color features, to refine saliency maps. In every iteration of the Iterative Random Walk, the energy function in the two feature spaces are regularized by the latest result of each other.

The framework of our algorithm is shown in Figure 2. Our algorithm is consist of two stages: Saliency Game and Iterative Random Walk. Each of the two stages uses both the color features and deep features. Each input image is segmented into different scale of superpixels. Two saliency maps are first produced through the *Saliency Game* among superpixels in color feature space and deep feature space. Then the two saliency maps are combined by the *Iterative Random Walk* exploiting complementary of different features. The final saliency map is formed by averaging the results over different scale of superpixels.

In a nutshell, the main contributions of our work include:

1) We propose a novel unsupervised Saliency Game to detect salient objects. Adopting two independent priors improves robustness. The nature of game equilibria assures accuracy when both priors are unsatisfactory,
2) Utilizing semantically-rich features extracted from a pre-trained CNN the proposed method is able to identify salient objects in complex scenes, in which traditional methods based handcrafted features may fail (see Figure 1(c)), and
3) An Iterative Random Walk algorithm across two feature spaces is proposed that takes advantage of the complementary relationship between the color feature space and the deep feature space to further refine the results.

II. Related Work

Some saliency works have followed an unsupervised approach. In [6], saliency of each region was defined as its absorbed time from boundary nodes, which measures its global similarity with all boundary regions. Yang *et al.* [5] ranked the similarity of image regions with foreground or background cues via graph-based manifold ranking. Saliency value of each image element was determined based on its relevance to given seeds. In [7], saliency pattern was mined to find foreground seeds according to prior maps. Foreground labels were propagated to unlabeled regions. Tong *et al.* [15] proposed a learning algorithm to bootstrap training samples generated from prior maps. These methods exploited either boundary background prior or foreground prior from a prior map, while we adopt two different priors in our method for robustness purposes. Priors only act as weak guidance with very small weights in the payoff function of our proposed Saliency Game.

Some deep learning based saliency detection methods have achieved great performance. In [9], two deep neural networks were trained, one to extract local features and the other to conduct a global search. Zhao *et al.* [10] proposed a multi-context deep neural network taking both global and local context into consideration. Li and Yu [11] explored high-quality visual features extracted from deep neural networks to improve the accuracy of saliency detection. In [12], high level deep features and low level handcrafted features were integrated in a unified deep learning framework for saliency detection. Pan *et al.* [16] propose several extremely computationally efficient methods for saliency detection based on the powerful

convolutional neural networks. All above methods needed a lot of time and many images for training. In this work, we are not against the CNN models, but we combine deep features with traditional color features in an unsupervised way, which result in an efficient unsupervised method complementary to CNNs that does on par with the above models that need labeled training data. Hopefully, this will encourage new models that can utilize both labeled and unlabeled data.

Furthermore, there are many computer vision and learning tasks in which game theory has been applied successfully. A grouping game among data points was proposed in [17]. Albarelli *et al.* [18] proposed a non-cooperative game between two sets of objects to be matched. A game between a region-based segmentation model and a boundary-based segmentation model was proposed in [19] to integrate two sub-modules. Erdem and Pelillo [20] formulated a multi-player game for transduction learning, whereby equilibria correspond to consistent labeling of the data. Miller and Zucker [21] showed that the relaxation labeling problem [22] is equivalent to finding Nash equilibria for polymatrix n-person games. However, to the best of our knowledge, game theory has not yet been used for salient object detection.

III. Saliency Game

Here, we formulate a non-cooperative game among superpixels to detect salient objects in an input image. The input image is firstly segmented into N superpixels by the SLIC algorithm [1] which act as players in the Saliency Game. $\mathcal{I} = \{1, 2, 3, \ldots, N\}$ denotes the enumeration of the set of superpixels. Each player chooses to be "background" or "foreground" as its pure strategy and its mixed strategy corresponds to this superpixel's saliency value. After showing their strategies, players obtain some payoff according to both their own and other players' strategies. Payoff is determined by a payoff function which incorporates position and objectness cues as well as support from others. We use each player's mixed strategy in the Nash equilibrium of the proposed Saliency Game as the saliency value of this superpixel in the output saliency map. Such an equilibrium corresponds to a steady state where each player plays a strategy that maximize its own payoff when the remaining players' strategies are kept fixed, which provides a globally plausible saliency detection result.

A. Game Setting

The pure strategy set is denoted as $\mathcal{S} = \{0, 1\}$, indicating "to be foreground" or "to be background", respectively. All superpixels' pure strategies are collectively called a pure strategy profile, denoted as $\boldsymbol{s} = (s_1, \ldots t, s_N)$. The strategy profile set is denoted as Θ. $\pi_{ij}(s_i, s_j)$ denotes a single payoff that superpixel i obtains, when playing pure strategy s_i against superpixel j who holds a pure strategy s_j, in their 2-person game. There are four possible values for $\pi_{ij}(s_i, s_j)$ that can be put into a 2×2 matrix B_{ij},

$$B_{ij} = \begin{pmatrix} \pi_{ij}(0,0) & \pi_{ij}(0,1) \\ \pi_{ij}(1,0) & \pi_{ij}(1,1) \end{pmatrix} \tag{1}$$

Payoff of superpixel i in pure strategy profile $\boldsymbol{s}$, where $\forall j \in \mathcal{I}$ the j-th superpixel's pure strategy s_j is the j-th component of vector $\boldsymbol{s}$, is denoted as $\pi_i(\boldsymbol{s})$. Payoff of superpixel i when it adopts a pure strategy t_i (not necessarily the i-th component of $\boldsymbol{s}$), while all other superpixels adopt pure strategies in pure strategy profile $\boldsymbol{s}$ is denoted as $\pi_i(t_i, \boldsymbol{s}_{-i})$. We make an assumption that the total payoff of superpixel i for playing with all others is the summation of payoffs for playing 2-player games with every other single superpixel. Formally, we assume that $\pi_i(\boldsymbol{s}) = \sum_{j \neq i} \pi_{ij}(s_i, s_j)$ and $\pi_i(t_i, \boldsymbol{s}_{-i}) = \sum_{j \neq i} \pi_{ij}(t_i, s_j)$.

A pure best reply for player i against a pure strategy profile $\boldsymbol{s}$ is a pure strategy such that no other pure strategy gives a higher payoff to i against $\boldsymbol{s}$. The i-th player's pure best-reply correspondence, which maps each pure strategy profile $\boldsymbol{s} \in \Theta$ to a pure strategy $s_i \in \mathcal{S}$, is denoted as $\beta_i : \Theta \to \mathcal{S}$:

$$\beta_i(\boldsymbol{s}) = \{s_i \in \mathcal{S} | \pi_i(s_i, \boldsymbol{s}_{-i}) \geq \pi_i(t_i, \boldsymbol{s}_{-i}), \forall t_i \in \mathcal{S}\}. \tag{2}$$

The combined pure best-reply correspondence $\beta : \Theta \to \Theta$ is defined as the cartesian product of all players' pure best-reply correspondence:

$$\beta(\boldsymbol{s}) = \times_{i \in \mathcal{I}} \beta_i(\boldsymbol{s}) \subset \Theta. \tag{3}$$

A pure strategy profile $\boldsymbol{s}$ is a pure Nash equilibrium if $\boldsymbol{s} \in \beta(\boldsymbol{s})$.

A probability distribution over the pure strategy set is termed as a mixed strategy. Mixed strategy of the i-th superpixel is denoted as a 2-dimensional vector $\boldsymbol{z}_i = (z_i^0, z_i^1)^{\mathrm{T}}$, while $z_i^0 = P(s_i = 0), z_i^1 = P(s_i = 1)$ and $z_i^0 + z_i^1 = 1$. The set of mixed strategies is denoted as Δ. A pure strategy thereby can be regarded as an extreme mixed strategy where only one component is 1 and the other one is 0, *e.g.* i-th player's pure strategy $s_i = 1$ is equivalent to its mixed strategy $\boldsymbol{z}_i = (0, 1)^{\mathrm{T}}$ because $P(s_i = 0) = 0$ and $P(s_i = 1) = 1$. Correspondingly, expected payoff of superpixel i for playing mixed strategy $\boldsymbol{z}_i$ against superpixel j holding mixed strategy $\boldsymbol{z}_j$ is denoted as $u_{ij}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{z}_i^{\mathrm{T}} B_{ij} \boldsymbol{z}_j$. We also denote $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$, $u_i(Z) = \sum_{j \neq i} u_{ij}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ and $u_i(\boldsymbol{w}_i, Z_{-i}) = \sum_{j \neq i} u_{ij}(\boldsymbol{w}_i, \boldsymbol{z}_j)$ to be mixed strategy version of $\boldsymbol{s}$, $\pi_i(\boldsymbol{s})$ and $\pi_i(t_i, \boldsymbol{s}_{-i})$. Similarly, a mixed Nash equilibrium is also defined to be a mixed strategy profile which is a mixed best reply to itself. These symbols or definitions are not stated here individually due to limited space.

From the definition of the Nash equilibrium above, it can be inferred that in a Nash equilibrium of a game, each player adopts a strategy that maximizes its own payoff when other players' strategies are fixed.

B. Payoff Function

We have assumed in Section III-A that the total payoff of superpixel i for playing with all others is the summation of every single payoff in its 2-person games with every other superpixel. Hence, here we focus on modeling payoff of every 2-person game. We define the payoff $\pi_{ij}(s_i, s_j)$ of superpixel i for its 2-person game with j as a weighted sum of three terms:

$$\pi_{ij}(s_i, s_j) = \lambda_1 \cdot \mathrm{pos}_i(s_i) + \lambda_2 \cdot \mathrm{obj}_i(s_i) + \mathrm{spt}_{ij}(s_i, s_j), \tag{4}$$

where $\mathrm{pos}_i(s_i)$, $\mathrm{obj}_i(s_i)$, and $\mathrm{spt}_{ij}(s_i, s_j)$ indicate the i-th superpixel's position prior, objectness prior and support that

superpixel j gives to superpixel i, respectively. λ_1 and λ_2 are parameters controlling the weight of the first two terms.

1) Position: Position prior term in the payoff function is formulated based on the observation that salient objects often fall at the image center. The position term should give a greater payoff when, a) Center superpixels choose to be foreground, and b) Boundary superpixels choose to be background. Assuming (x_0, y_0) to be the image center, and (x_i, y_i) to be the center coordinate of superpixel i, the position prior term is defined as follows,

$$\mathrm{pos}_i(s_i) = \begin{cases} \frac{1}{N}\exp\{-(x_i - x_0)^2 - (y_i - y_0)^2\} & \text{if } s_i = 1, \\ \frac{1}{N}(1 - \exp\{-(x_i - x_0)^2 - (y_i - y_0)^2\}) & \text{if } s_i = 0. \end{cases} \tag{5}$$

2) Objectness: Generally, objects attract more attention than background clutter. Hence superpixels which are part of an object are more likely to be salient. The objectness term should give a greater payoff when, a) Superpixels with high objectness choose to be foreground, and b) Superpixels with low objectness choose to be background.

We exploit the geodestic object proposal (GOP) [23] method to extract a set of object segmentations, and define the objectness of a superpixel according to its overlap with all GOP proposals as follows:

$$\mathrm{obj}_i(s_i) = \begin{cases} \frac{1}{N \cdot N_o}\sum_{j=1}^{N_o}\frac{\sum_{x,y} O_j(x,y) \times P_i(x,y)}{\sum_{x,y} P_i(x,y)} & \text{if } s_i = 1, \\ \frac{1}{N}(1 - \frac{1}{N_o}\sum_{j=1}^{N_o}\frac{\sum_{x,y} O_j(x,y) \times P_i(x,y)}{\sum_{x,y} P_i(x,y)}) & \text{otherwise.} \end{cases} \tag{6}$$

in which $\{O_j\}_{N_o}$ is the set of object candidate masks generated by the GOP method, where $O_j(x, y) = 1$ indicates that the pixel located at (x, y) of the input image belongs to the j-th object proposal, and $O_j(x, y) = 0$, otherwise. $P_i(x, y)$, $i \in \mathcal{I}$, denotes the mask of the i-th superpixel, where $P_i(x, y) = 1$ indicates that the pixel located at (x, y) of the input image belongs to the i-th superpixel, and $P_i(x, y) = 0$, otherwise.

3) Support: With a much larger weight in the payoff function (λ_1 and λ_2 being small), support from others is the main source of payoff obtained by each superpixel. When playing with an opponent, each superpixel judges if the opponent's strategy is right or wrong with its own stance, and provides a higher or lower even negative support to the opponent accordingly. More precisely,

- Each superpixel takes a neutral attitude to opponents who hold different pure strategies from itself, and provides them zero support.
- If an opponent adopts the same pure strategy as superpixel i,
  - if the opponent's strategy is similar to it, then superpixel i provides the opponent a great support in recognition of its choice.
  - else if the opponent is not similar with it, then superpixel i provides the opponent a small even negative support as punishment.

Formally, the support term is defined as follows,

$$\mathrm{spt}_{ij}(s_i, s_j) = \begin{cases} A(i,j) - \frac{\alpha}{N}\sum_{k=1}^{N} A(i,k) & \text{if } s_i = s_j, \\ 0 & \text{if } s_i \neq s_j, \end{cases} \tag{7}$$

where α is a positive constant, $A(i, j)$ is the affinity between superpixels i and j. We consider affinity in terms of color features and deep features, denoted as A^c and A^d respectively, and defined as below.

We extract deep features from the output of the last convolution layer of VGG16 [24] or FCN32s [25]. This is because features from the last layer of CNNs encode semantic abstraction of objects and are robust to appearance variations. Since the deep feature maps are too small, we use dilated convolution in the last three convolution blocks to obtain lager feature maps. The feature maps are resized to the input image size via linear interpolation. Affinity in terms of deep features is defined to their Gaussian weighted Euclidean distance:

$$A^d(i,j) = \exp\Big(-\frac{\left\|\boldsymbol{f}_i^d - \boldsymbol{f}_j^d\right\|^2}{\sigma^2}\Big), \tag{8}$$

in which $\boldsymbol{f}_i^d$ is the deep feature vector of superpixel i. Each superpixel is represented by the mean deep feature vector of all its contained pixels.

The semantically-rich deep features can help accurately locate the targets but fail to describe the low-level information. Therefore, we also employ color features as a complement to deep features. Inspired by [6], we use CIE-Lab color histograms to describe superpixels' color appearance. With CIE-Lab color space divided into 8^3 ranges, the color feature vector of the i-th superpixel is denoted as $\boldsymbol{f}_i^c$. Affinity between superpixels i and j in the color feature space is denoted as $A^c(i, j)$, which is defined to be their Gauss weighted Chi-square distance:

$$A^c(i,j) = \exp\Big(-\frac{\chi^2(\boldsymbol{f}_i^c, \boldsymbol{f}_j^c)}{\sigma^2}\Big). \tag{9}$$

So far, we have modeled the payoff $\pi_i(s_i, s_j)$ that superpixel i obtains by playing a 2-person pure strategy game with superpixel j. The expected payoff $u_i(\boldsymbol{w}_i, Z_{-i})$ that superpixel i obtains by playing mixed strategy game with all others adopting strategies in mixed strategy profile Z can be given based on the definition stated at the beginning of this section,

$$u_i(\boldsymbol{w}_i, Z_{-i}) = \sum_{j=1 \wedge j \neq i}^{N} \boldsymbol{w}_i^{\mathrm{T}} B_{ij} \boldsymbol{z}_j. \tag{10}$$

C. Computing Equilibrium

We use Replicator Dynamics [26] to compute the mixed strategy Nash equilibrium of the proposed Saliency Game. In Replicator Dynamics, a population of individuals play the game, generation after generation. A selection process acts on the population, causing the number of users holding fitter

strategies to grow faster. We use discrete time Replicator Dynamics to find the equilibrium of the game:

$$z_i^h(t+1) = z_i^h(t)\frac{const + u_i(\boldsymbol{e}^h, Z(t)_{-i})}{const + u_i(Z(t))}, \tag{11}$$

in which $z_i^h(t)$ represents the h-th component of the i-th player's mixed strategy at time t, $\boldsymbol{e}^h$ is a vector whose h-th component is 1, while other components are 0. We set the initial mixed strategies of player i to $z_i(0) = (0.5, 0.5)$, $\forall i \in \mathcal{I}$. $const$ is background birthrate for an individual, which is set to 0.1 empirically to make sure $const + u_i(\boldsymbol{e}^h, Z(t)_{-i})$ is positive for all i in $\mathcal{I}$ [27]. We iterate Eqn.11 at most 30 times or until $\forall i \in \mathcal{I}, |z_i(t) - z_i(t-1)| < \epsilon$. There could be multiple equilibria in a game, likewise in the proposed Saliency Game. Replicator Dynamics might reach different Nash equilibria if the initial state $z_i(0)$ is set to different interior points of Δ. We find that $z_i(0) = (0.5, 0.5)$ is a good initialization leading to plausible saliency detection.

In the proposed Saliency Game, each superpixel inspects strategies of all other superpixels and takes a stance by providing large or small even negative support. Usually, no matter what strategy a superpixel adopts, there are both protesters and supporters. Game equilibira provide a good trade off among different influences. Thus, in the equilibrium of the proposed Saliency Game with payoff function as defined in Eqn. 4, each superpixel chooses a strategy that suits itself best given its position, objectness, and support from others. Doing so has two advantages: 1) the center position prior and the objectness prior are almost independent, when one prior is unsatisfactory, the other may work.

As shown in the first row of Figure 3, the little pug appears away from image center, but since it is the only object in the image, the objectness prior identifies it correctly. 2) the two priors only serve as weak guidance and obtain small weights in the payoff function. Even when they are both unsatisfactory on some image regions, pressure from peers will impel these regions to get proper saliency values in the equilibrium of the game. As shown in the second row of Figure 3, although only heads of the people are high in position prior and objectness prior, the produced saliency map can highlight the entire object. From the third row of Figure 3, we can see that the proposed algorithm also suppresses background effectively when prior highlights background areas by mistake. Note that in order to illustrate effectiveness of the proposed Saliency Game, only color feature is used in the three shown cases.

IV. Iterative Random Walk

Traditional color features are of high-resolution, so saliency maps generated in color space are detailed and with clear edges. But due to lack of high-level information, sometimes they fail to locate the targets accurately (see Figure 4(c)). On the contrary, since deep features encode high-level concept of objects well, saliency maps generated in the deep feature space are able to find correct salient objects in an image. But due to several layers of convolution and pooling, these features are too coarse. Thus the generated saliency maps are indistinctive, as shown in Figure 4(d).

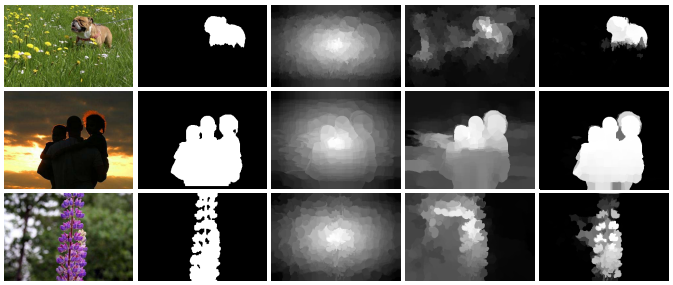

Fig. 3. The proposed Saliency Game still works well even when position prior and objectness prior are not very satisfying. (a) Input images. (b) Ground Truth maps. (c) and (d) Illustration of the position term and objectness term in Eqn. 5. (e) Saliency maps by Saliency Game using the color feature.

Accordingly, here, we use both complementary features for a better result. However, as shown in Figure 4(e), although the weighted sum of the two is slightly better, they are not satisfactory. To solve this problem, in this section, inspired by the metric fusion presented in [28], we propose an Iterative Random Walk method to best exploit this two complementary feature spaces. In the proposed Iterative Random Walk, metrics in the two feature spaces are fused as stated in [28] (corss fusion in Eqn. 16 is the work of Tu *et al.*), in addition, we also make the two random walk energy function regularized by the latest result of each other (cross regularization in Eqn. 17 and Eqn. 18 is our work). Figure 5 shows that both cross fusion and cross regularization contribute to the performance.

With superpixels as nodes, a neighbor graph and a complete graph are constructed in both feature space (deep and color features). The affinity between two superpixels is assigned to the edge weight. Four weight matrices are defined:

- $\mathcal{W}^d$ and W^d: weight matrices of neighbor and complete graphs in the deep feature space, respectively.
- $\mathcal{W}^c$ and W^c: weight matrices of neighbor and complete graphs in the color space, respectively.

In the complete graphs, there is an edge between every pair of nodes:

$$W^d(i,j) = A^d(i,j), \quad \forall j \in \mathcal{I}. \tag{12}$$

In the neighbor graphs, each node is connected only to its neighbors. We adopt a definition of 2-hoop neighbor which is frequently used in superpixel based saliency detection methods. The set of the i-th superpixel's neighbors is denoted as $\mathcal{N}(i) = \mathcal{N}_1(i) \cup \mathcal{N}_2(i) \cup \mathcal{N}_3(i)$, where $\mathcal{N}_1(i)$ indicates the set of superpixels who share at least one common edge with the i-th superpixel. $\mathcal{N}_2(i)$ and $\mathcal{N}_3(i)$ are defined as follows:

$$\mathcal{N}_2(i) = \{j | j \in \mathcal{N}_1(k), k \in \mathcal{N}_1(i), j \neq i\}, \tag{13}$$

$$\mathcal{N}_3(i) = \begin{cases} \emptyset & \text{if } i \notin \mathcal{B} \\ \{j | j \in \mathcal{B}, j \neq i\} & \text{if } i \in \mathcal{B} \end{cases}, \tag{14}$$

where $\mathcal{B}$ denotes the set of superpixels in image boundary. Weight matrices of the neighbor graph $\mathcal{W}^d$ and W^d are defined as follows,

$$\mathcal{W}^d(i,j) = \begin{cases} A^d(i,j) & \text{if } j \in \mathcal{N}(i), \\ 0 & \text{if } j \notin \mathcal{N}(i), \end{cases} \tag{15}$$

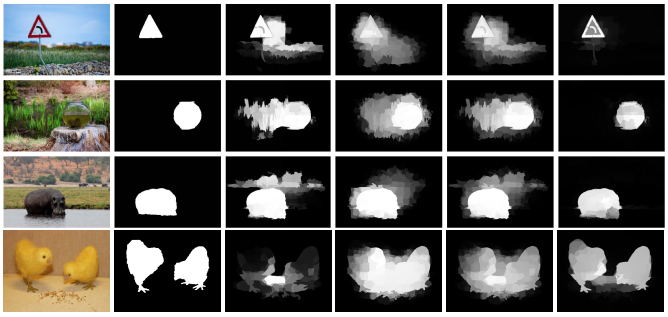

Fig. 4. Effect of the Iterative Random Walk. (a) Input images. (b) Ground Truth maps. (c) Saliency maps generated by our Saliency Game in the color feature space. (d) Saliency maps generated by our Saliency Game in the deep feature space. (e) The weighted summation of (c) and (d). (f) Saliency maps after refinement by the Iterative Random Walk proposed in this section.

in which $\mathcal{W}^c$ and W^c are defined similarly but using A^c. See Eqn.8 and 9 for definitions of A^d and A^c.

Following [28], we firstly fuse these four affinity matrices as follows,

$$\begin{cases} P^d_{(t+1)} = \mathcal{P}^c \times P^d_{(t)} \times \mathcal{P}^c, \\ P^c_{(t+1)} = \mathcal{P}^d \times P^c_{(t)} \times \mathcal{P}^d, \end{cases} \tag{16}$$

in which t is the number of iterations, $\times$ denotes matrix multiplication, $P^d_{(0)}(i,j) = W^d(i,j)/\sum_{j=1}^{N} W^d(i,j)$, $\mathcal{P}^d_{(0)}(i,j) = \mathcal{W}^d(i,j)/\sum_{j=1}^{N} \mathcal{W}^d(i,j)$. $P^c_{(0)}$ and $\mathcal{P}^c_{(0)}$ are defined similarly but using $W^d(i,j)$ and $\mathcal{W}^d(i,j)$.

Then, using the fused affinity matrices, we let the results in the two feature spaces regularize each other. Two random walk energy functions are defined as follows,

$$E^d_{(t+1)}(\boldsymbol{l}) = \sum_{i,j} P^d_{(t)}(i,j)(l_i - l_j)^2 + \beta \sum_{i=1}^{N} (l_i - l^c_{i(t)})^2, \tag{17}$$

$$E^c_{(t+1)}(\boldsymbol{l}) = \sum_{i,j} P^c_{(t)}(i,j)(l_i - l_j)^2 + \beta \sum_{i=1}^{N} (l_i - l^d_{i(t)})^2, \tag{18}$$

in whcih $\boldsymbol{l}$ is the label vector, l_i is the i-th superpixel's label, and β is a parameter.

By minimizing the two energy functions above, we have,

$$\boldsymbol{l}^d_{(t+1)} = \arg\min_{\boldsymbol{l}} E^d_{(t+1)}(\boldsymbol{l}) = (L^d_{(t+1)} + \beta I)^{-1} \boldsymbol{l}^c_{(t)}, \tag{19}$$

$$\boldsymbol{l}^c_{(t+1)} = \arg\min_{\boldsymbol{l}} E^c_{(t+1)}(\boldsymbol{l}) = (L^c_{(t+1)} + \beta I)^{-1} \boldsymbol{l}^d_{(t)}, \tag{20}$$

where L is the Laplacian matrix. $\boldsymbol{l}^c_{(0)}$ and $\boldsymbol{l}^d_{(0)}$ are set to the results of the Saliency Game stated in Section III. After T rounds, the iteration converges and the final saliency map is obtained as:

$$S = (1-\rho) \cdot \boldsymbol{l}^c_{(T)} + \rho \cdot \boldsymbol{l}^d_{(T)}, \tag{21}$$

in which ρ controls the weight of the two results.

As shown in Figure 4, through the Iterative Random Walk, semantic information from deep features helps locate the target object accurately, and information from the color space helps cut the whole salient object clearly. Also, objects that could not be detected in one feature space can be detected with the help of results from the other feature space. For example, in the fourth row of Figure4(c), when using color features, the chicken are not detected due to their low color contrast against the background. When use deep features (Figure 4(d)), the chicken can be detected but are only coarsely highlighted. After processed by Iterative Random Walk (Figure 4(f)), the chicken are segmented precisely.

V. Experiments and Results

In this section, we evaluate the proposed method on 6 benchmark datasets: ECSSD [29] (1000 images), PASCAL-S [30] (850 images), MSRA-5000 [31] (5000 images), HKU-IS [11] (4447 images), DUT-OMRON [5] and SOD [32].

We compare our algorithm with 12 state-of-the-art methods including BL [15], MB+ [33], DRFI [4], DSR [34], HS [29], LEGS [9], MCDL [10], MST [35], LR [36], RC [37], wCO [14], and KSR [38]. Results of different methods are provided by authors or achieved by running available codes. We also make our code public as many researches have practiced.[1][2]

A. Parameter Setting

All parameters are set once fixed over all the datasets. We segment an image into 100, 150, 200, and 250 superpixels (i.e., 4 segmentation images), run the algorithm on each segmentation image, and average the four outputs to form the final saliency map. σ^2 is set to 0.1 and ϵ is set to 10^{-4}. The parameters controlling the weight of each term in the payoff function (Eqn. 4) are set to $\lambda_1 = 2.1 \times 10^{-6}$, $\lambda_2 = 9 \times 10^{-7}$, respectively. α in Eqn. 7 is set to 0.007. β in Eqn. 19 and Eqn. 20 is set to 1. In Eqn. 21, we set $T = 20$, $\rho = 0.7$.

B. Evaluation Metrics

We use precision-recall curve, F-measure curve, F-measure and AUC to quantitatively evaluate the experimental results. The precision value is defined as the ratio of salient pixels correctly assigned to all salient pixels in the map to be evaluated, while the recall value corresponds to the percentage of the detected salient pixels with respect to all salient pixels in the ground-truth map. The F-measure is an overall performance indicator computed by the weighted harmonic of precision and recall. We set $\beta^2 = 0.3$ as suggested in [39] to emphasize the precision.

Given a saliency map with intensity values normalized to the range of 0 and 1, a series of binary maps are produced by using several fixed thresholds in $[0, 1]$. We compute the precision/recall pairs of all the binary maps to plot the precision-recall curves and the F-measure curves. As suggested in [39], we use twice the mean value of the saliency maps as the threshold to generate binary maps for computing F-measure. Notice that some works have reported slightly different F-measures using different thresholds.

[1] https://github.com/zengxianyu/uga
[2] http://ice.dlut.edu.cn/lu/

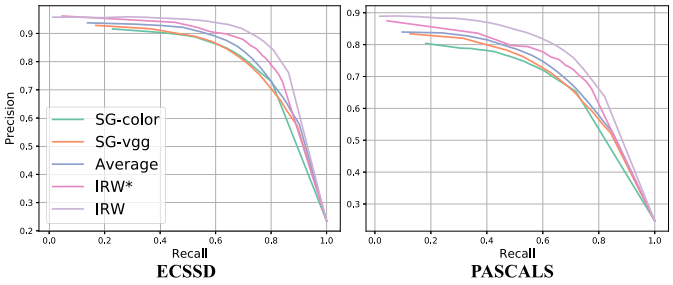

Fig. 5. Effect of each component of the proposed method. Left) PR curves on ECSSD dataset. Right) PR curves on PASCAL-S dataset. SG-color: Saliency Game with color features. SG-vgg: Saliency Game with VGG features. Average: a weighted sum of SG-color and SG-vgg. IRW*: Saliency Game refined by the Iterative Random Walk without metric cross fusion. IRW: Saliency Game refined by the Iterative Random Walk with metric cross fusion.

C. Ablation Study

To demonstrate the effectiveness of each component of our algorithm, we test the proposed Saliency Game and the Iterative Random Walk (with and without metric fusion) separately on ECSSD and PASCAL-S datasets. PR curves in Figure 5 show that:

- The proposed saliency game algorithm achieves favorable performance, even when using only simple color features.
- The average results of Saliency Game with color features and deep features are better than the result of using only one kind of features.
- Combining the results of deep features and color features by the Iterative Random walk leads to better performance than simply averaging the results.
- The comparision of results of Iterative Random Walk (IRW), Iterative Random Walk without metric fusion (IRW*), and weighted sum of results in the two feature spaces (Average), demonstrates the advantage of both cross regularization and metric fusion in the Iterative Random Walk.

D. Computational Complexity

Let N denote the number of superpixels and D denote the dimensions of features. Constructing the affinity matrix involves computing distance between each pair of superpixels. Take euclidean distance as an example, computing distance between two D-dimensional vectors requires D operations of multiplication. There are N^2 paris of superpixel, so DN^2 operations of multiplication are required to construct the affinity matrix.

Let T denote the number of iterations in Eqn.11. Computing the Equilibria of the Saliency Game needs to multiply a $N \times N$ matrix with a $N \times 2$ matrix T times, in which $2TN^2$ operations of multiplication are required. Therefore, complexity of the proposed Saliency Game is $O(DN^2+2TN^2)$, which is $O(N^3)$ since N is usually larger than D and T.

The proposed Iterative Random Walk involves multiplying two $N \times N$ matrices several times, inversing two $N \times N$ matrix, and multiplying a $N \times N$ matrix with a N-demensional vector $2T$ times. Complexity of these three operations are $O(N^3)$, $O(N^3)$ and $O(2TN^2)$ respectively. Since N is usually larger than T, the complexity of the Iterative Random Walk is $O(N^3)$.

TABLE I

THE AVERAGE RUN-TIME (IN SECONDS) OF OUR METHODS AND SEVERAL STATE-OF-THE-ARTS. SUPERVISED METHODS ARE IN BOLD

Methods	BL	DSR	wCO	LR	HS
Run-time	21.5161	3.4796	0.1484	10.0259	0.3821
Code	Matlab	Matlab	Matlab	Matlab	EXE
Methods	**RC**	**DRFI**	**MCDL**	**LEGS**	**Ours**
Run-time	0.1360	8.0104	2.2521	1.9050	1.3819
Code	C	Matlab	Python	Matlab+C	Matlab

TABLE II

F-MEASURE SCORES (THE LARGER THE BETTER). OURS-F DENOTES OUR METHODS WITH FCN FEATURES, AND OURS-V DENOTES OUR METHODS WITH VGG FEATURES. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN RED AND GREEN, RESPECTIVELY. SUPERVISED METHODS ARE MARKED IN BOLD

dataset	ECSSD	PASCALS	MSRA	HKU-IS	SOD	OMRON
BL	.6838	.5742	.7840	.6597	.5723	.4989
DSR	.6618	.5575	.7841	.6774	.5962	.5243
HS	.6347	.5314	.7671	.6359	.5212	.5108
RC	.4560	.4039	.5754	.5008	.4184	.4058
wCO	.6764	.5999	.7937	.6770	.5987	.5277
LR	.5629	.4791	.6940	.5546	.4843	.4531
MST	.6778	.6095	.7803	.6574	.5916	.5178
MB+	.6746	.6077	.7911	.6641	.5894	.5195
DRFI	.7329	.6182	-	.7219	.6470	.5505
MCDL	7959	.6912	-	.7573	.6772	.6250
LEGS	.7851	-	-	.7229	.6834	.5916
KSR	.7817	.7039	-	.7468	.6679	.5911
Ours-F	.8215	.7062	.8666	.8015	.6896	.5981
Ours-V	.8214	.6905	.8693	.7979	.6852	.6190

The proposed method is implemented in MATLAB on a PC with a 3.6GHz CPU and 32GB RAM. It takes 1.38 seconds to process an image on average. Comparisons of the proposed method with state-of-the-arts in speed performance are shown in Table I. Speed of our method is comparable to state-of-the-arts. In addition, although speed of some supervised methods are close to our methods, they usually require a lot of time to train. For example, MCDL costs 31 hours for training [10].

E. Comparison With State-of-the-Art Methods

In this section, we show performance comparisons of the proposed method against 12 state-of-the-arts on six benchmark datasets. Among methods, LR, DSR, RC, BL, HS, MST, wCO, MB+ are unsupervised methods. DRFI, LEGS, MCDL, KSR are supervised methods. For a fair comparison, we do not provide evaluation results of DRFI, LEGS, MCDL, and KSR methods on MSRA-5000 dataset since these methods all randomly select images from this dataset for training. Further, since LEGS also selects images from PASCAL-S dataset, we do not show its performance over the PASCAL-S dataset.

As shown in Figure 6, Table II and Table III, our method with FCN features or VGG features both compare favorably against state-of-the-art approaches. The performance of using FCN features and VGG features are close, and is slightly better when using FCN features. A possible reason is that FCN is

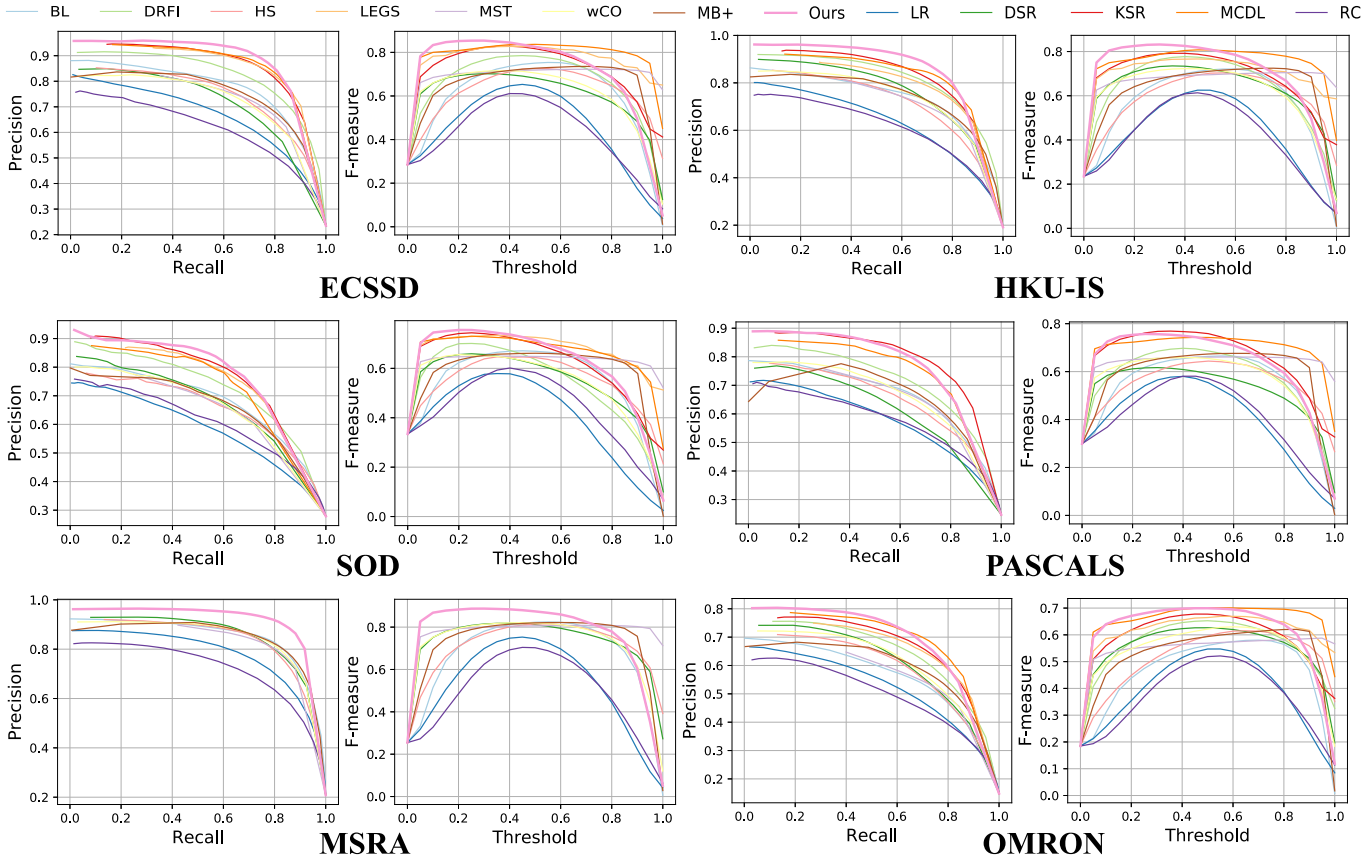

Fig. 6. Comparison of our method against-state-of-the-arts in terms of PR curves and F-measure curves.

TABLE III

AUC SCORES (THE LARGER THE BETTER). OURS-F DENOTES OUR METHODS WITH FCN FEATURES, AND OURS-V DENOTES OUR METHODS WITH VGG FEATURES. SUPERVISED METHODS ARE MARKED IN BOLD. THE BEST AND THE SECOND BEST RESULTS ARE SHOWN IN RED AND GREEN, RESPECTIVELY

dataset	ECSSD	PASCALS	MSRA	HKU-IS	SOD	OMRON
BL	.9143	.8671	.9535	.9140	.8503	.8778
DSR	.8619	.8118	.9382	.9008	.8208	.8787
HS	.8838	.8362	.9279	.8782	.8145	.8586
RC	.8342	.8139	.8951	.8530	.7924	.8476
wCO	.8814	.8482	.9360	.8952	.8026	.8846
LR	.8619	.8119	.9225	.8645	.7787	.8556
MST	.8713	.8307	.9133	.8814	.7858	.8529
MB+	.9026	.8608	.9491	.9159	.8319	.8891
DRFI	.9404	8950	-	.9435	.8813	9157
MCDL	.9186	.8699	-	.9175	.8163	.9014
LEGS	.9235	-	-	.9026	.8268	.8841
KSR	.9268	.9012	-	.9099	.8403	.8921
Ours-F	.9272	.8724	.9583	.9183	.8481	.8869
Ours-V	.9091	.8614	.9463	.8974	.8184	.8680

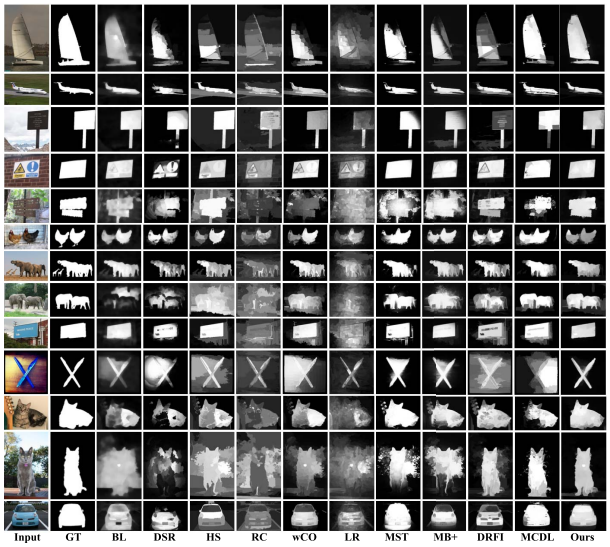

Fig. 7. Visual comparison of our method against state-of-the-arts.

trained with pixel-level supervision, thereby its features are more suitable for pixel-level tasks such as saliency detection.

Visual comparison of the proposed method against state-of-the-art on different datasets is shown in Figure 7.

F. *Sensitivity Analysis*

We analyze sensitivity of the proposed Saliency Game to parameters λ_1, λ_2, α and sensitivity of the Iterative Random Walk to parameters β and ρ. For simplicity, we let $\lambda_1 = \lambda_2 = \lambda$ and analyze sensitivity to λ_1 and λ_2 together as λ. As shown in Figure 8, performance of the Saliency Game is relatively stable when $\alpha < 10^{-2}$ and $\lambda \in [10^{-8}, 10^{-4}]$. Performance of the Iterative Random Walk is relatively stable when β varies between 0.5 and 10. The Iterative Random Walk achieve its best performance when $\rho = 0.8$.

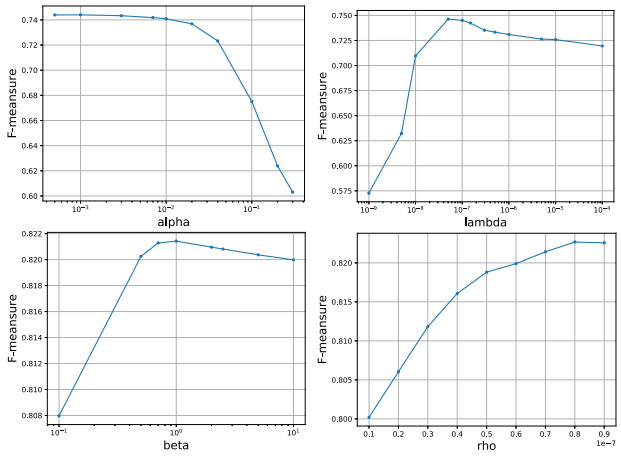

Fig. 8. Sensitivity of the Saliency Game to parameters α, λ and Iterative Random Walk to β, ρ, evaluated on ECSSD dataset.

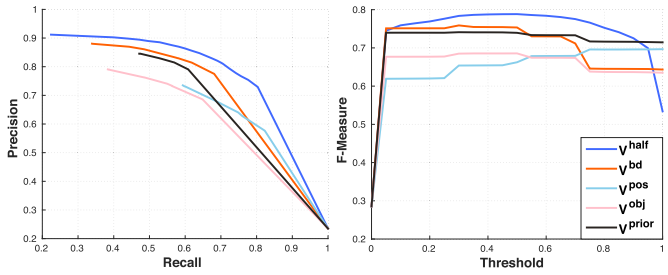

Fig. 9. Effect of different initialization, evaluated on PASCALS dataset. V^{half}, V^{pos}, V^{obj}, V^{bd} and V^{prior}: Saliency detection results corresponding to different initial state V^{half}, V^{pos}, V^{obj}, V^{bd} and V^{prior}, respectively.

The proposed Saliency Game is a special category of games named polymatrix games [40], where each player plays a two-player game against each other and his payoff is then the sum of the payoffs from each of the two-player games [41]. Howson *et al.* [40] showed that every polymatrix game has at least one equilibrium. Therefore, the proposed Saliency Game also has at least one, but could have more than one equilibria. Replicator Dynamics is invoked to find a Nash equilibrium of the game, in which different Nash equilibria might be reached if the initial state $\boldsymbol{z}_i(0)$ is set to different interior points of Δ. Empirically, we find that $\forall i \in \mathcal{I}, \boldsymbol{z}_i(0) = (0.5, 0.5)$ is a good initialization leading to plausible saliency detection. We analyze the sensitivity of the saliency detection results to different initial state $\boldsymbol{z}_i(0)$. We denote the initial state used in Section III as V^{half}, and the other four initial states as V^{bd}, V^{pos}, V^{obj}, and V^{prior}. Each of them is a $2 \times N$ matrix, where the i-th column vector (denoted as $\boldsymbol{v}_i^{bd}$, $\boldsymbol{v}_i^{pos}$, $\boldsymbol{v}_i^{obj}$, $\boldsymbol{v}_i^{prior}$, respectively) corresponds to the mixed strategy of superpixel i. Variables $\boldsymbol{v}_i^{bd}$, $\boldsymbol{v}_i^{pos}$, $\boldsymbol{v}_i^{obj}$ and $\boldsymbol{v}_i^{prior}$ are set as follows:

$$\boldsymbol{v}_i^{bd,1} = \begin{cases} 0.4 & \text{if } i \in \mathcal{B} \\ 0.5 & \text{otherwise} \end{cases}, \quad \boldsymbol{v}_i^{bd,0} = 1 - \boldsymbol{v}_i^{bd,1}. \tag{22}$$

$$\boldsymbol{v}_i^{pos,1} = N \cdot pos_i(1), \quad \boldsymbol{v}_i^{pos,0} = N \cdot pos_i(0). \tag{23}$$

$$\boldsymbol{v}_i^{obj,1} = N \cdot obj_i(1), \quad \boldsymbol{v}_i^{obj,0} = N \cdot obj_i(0). \tag{24}$$

$$\boldsymbol{v}_i^{prior,1} = prior_i, \quad \boldsymbol{v}_i^{prior,0} = 1 - prior_i, \tag{25}$$

where $prior_i$ is the saliency value of superpixel i computed by another saliency detection method. In this experiment, we use MR [5] model to compute $prior_i, \forall i \in \mathcal{I}$. We try four different initial states to test whether inducing prior knowledge into the initial state leads to a better saliency detection result. We show the quantitative comparison of the five different results in terms of F-measure curves and PR curves in Figure 5-F. It can be seen that the initial state V^{half} without any prior knowledge, which is adopted in the paper, leads to the best saliency detection.

VI. Summary and Conclusion

We formulate a Saliency Game among superpixels and propose an iterative random walk to combine deep feature and a color feature for a better saliency detection result. Extensive experiments over four benchmark datasets demonstrate that the proposed algorithm achieves favorable performance against state-of-the-art methods. The sensitivity analysis shows the robustness of the proposed method to parameter changes. Different from most previous methods that formulate saliency detection as minimizing one single energy function, the game-theoretic approach can be regarded as maximizing many competing objective functions simultaneously. The game equilibrium automatically provides a trade-off. This seems very natural for attention modeling and saliency detection, as also features and objects compete to capture our attention. In addition, compared with CNN based saliency detection methods which need to be trained on images with pixel-level masks as ground truth, the proposed method extracts features from a pre-trained CNN and combines them with color features in an unsupervised manner. This provides an efficient complement to CNNs that does on par with models that need labeled training data. Hopefully, our approach will encourage future models that can utilize both labeled and unlabeled data.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels," Ecole Polytechn. Fedrale de Lausanne, Lausanne, Switzerland, Tech. Rep. 149300, 2010.

[2] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.

[3] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[4] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2083–2090.

[5] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[6] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing Markov chain," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1665–1672.

[7] Y. Kong, L. Wang, X. Liu, H. Lu, and R. Xiang, "Pattern mining saliency," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 583–598.

[8] Q. Wang, W. Zheng, and R. Piramuthu, "Grab: Visual saliency via novel graph model and background priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 535–543.

[9] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3183–3192.

[10] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1265–1274.

[11] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5455–5463.
[12] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 660–668.
[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[14] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2814–2821.
[15] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1884–1892.
[16] H. Pan, B. Wang, and H. Jiang. (2015). "Deep learning for object saliency detection and image segmentation." [Online]. Available: https://arxiv.org/abs/1505.01173
[17] A. Torsello, S. R. Bulo, and M. Pelillo, "Grouping with asymmetric affinities: A game-theoretic perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 292–299.
[18] A. Albarelli, S. R. Bulò, A. Torsello, and M. Pelillo, "Matching as a non-cooperative game," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1319–1326.
[19] A. Chakraborty and J. S. Duncan, "Game-theoretic integration for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 1, pp. 12–30, Jan. 1999.
[20] A. Erdem and M. Pelillo, "Graph transduction as a noncooperative game," *Neural Comput.*, vol. 24, no. 3, pp. 700–723, 2012.
[21] D. A. Miller and S. W. Zucker, "Copositive-plus lemke algorithm solves polymatrix games," *Oper. Res. Lett.*, vol. 10, no. 5, pp. 285–290, 1991.
[22] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. TPAMI-5, no. 3, pp. 267–287, May 1983.
[23] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 725–739.
[24] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1337–1342.
[26] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Math. Biosci.*, vol. 40, nos. 1–2, pp. 145–156, 1978.
[27] J. W. Weibull, *Eolutionary Game Theory*. Cambridge, MA, USA: MIT Press, 1999.
[28] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2997–3004.
[29] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.
[30] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.
[31] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.
[32] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 49–56.
[33] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 FPS," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1404–1412.
[34] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2976–2983.
[35] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2334–2342.
[36] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 853–860.
[37] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.
[38] T. Wang, L. Zhang, H. Lu, C. Sun, and J. Qi, "Kernelized subspace ranking for saliency detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 450–466.
[39] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1597–1604.
[40] J. T. Howson, Jr., "Equilibria of polymatrix games," *Manage. Sci.*, vol. 18, no. 5, pp. 312–318, 1972.
[41] A. Deligkas, J. Fearnley, T. P. Igwe, and R. Savani, "An empirical study on computing equilibria in polymatrix games," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 186–195.

Yu Zeng received the B.S. degree in electronic and information engineering from the Dalian University of Technology, Dalian, China, in 2017, where she is currently pursuing the master's degree under the supervision of Prof. H. Lu.

Mengyang Feng received the B.E. degree in electrical and information engineering from the Dalian University of Technology in 2015, where he is currently pursuing the Ph.D. degree under the supervision of Prof. H. Lu.

Huchuan Lu received the Ph.D. degree in system engineering and the M.S. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 1998, respectively. He joined the Faculty of the School of Information and Communication Engineering, DUT, in 1998, where he is currently a Full Professor. He is a member of ACM and an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS.

Gang Yang received the M.S. degree and the Ph.D. degree in measurement and control technology and instrumentation from Northeastern University (NEU), Shenyang, China, in 1999 and 2007, respectively. He joined the Faculty of the College of Information Science and Engineering, NEU, in 1999, where he is currently an Associate Professor. His current research interests include computer vision and pattern recognition with focus on visual tracking, saliency detection, and segmentation.

Ali Borji received the B.S. degree in computer engineering from the Petroleum University of Technology, Tehran, Iran, the M.S. degree in computer engineering from Shiraz University, Shiraz, Iran, and the Ph.D. degree in cognitive neurosciences from the Institute for Studies in Fundamental Sciences, Tehran. He spent four years as a Post-Doctoral Scholar with the iLab, University of Southern California from 2010 to 2014. He is currently an Assistant Professor with the University of Central Florida.