Super Diffusion for Salient Object Detection

Peng Jiang, *Member, IEEE,* Zhiyi Pan, Changhe Tu, Nuno Vasconcelos, *Fellow, IEEE,* Baoquan Chen, *Senior Member, IEEE,* and Jingliang Peng, *Member, IEEE*

***Abstract*—One major branch of saliency object detection methods are diffusion-based which construct a graph model on a given image and diffuse seed saliency values to the whole graph by a diffusion matrix. While their performance is sensitive to specific feature spaces and scales used for the diffusion matrix definition, little work has been published to systematically promote the robustness and accuracy of salient object detection under the generic mechanism of diffusion.**

In this work, we firstly present a novel view of the working mechanism of the diffusion process based on mathematical analysis, which reveals that the diffusion process is actually computing the similarity of nodes with respect to the seeds based on diffusion maps. Following this analysis, we propose super diffusion, a novel inclusive learning-based framework for salient object detection, which makes the optimum and robust performance by integrating a large pool of feature spaces, scales and even features originally computed for non-diffusion-based salient object detection. A closed-form solution of the optimal parameters for the integration is determined through supervised learning.

At the local level, we propose to promote each individual diffusion before the integration. Our mathematical analysis reveals the close relationship between saliency diffusion and spectral clustering. Based on this, we propose to re-synthesize each individual diffusion matrix from the most discriminative eigenvectors and the constant eigenvector (for saliency normalization).

The proposed framework is implemented and experimented on prevalently used benchmark datasets, consistently leading to state-of-the-art performance.

***Index Terms*—Saliency detection, diffusion, spectral clustering.**

I. Introduction

The aim of saliency detection is to identify the most salient pixels or regions in a digital image which attract humans' first visual attention. Results of saliency detection can be applied to other computer vision tasks such as image resizing, thumbnailing, image segmentation and object detection. Due to its importance, saliency detection has received intensive research attention resulting in many recently proposed algorithms.

In the field of saliency detection, two branches have developed, which are visual saliency detection [1]–[13] and salient object detection [14]–[52]. While the former tries to predict where the human eyes focus on, the latter aims to detect the whole salient object in an image. Saliency in both branches can be computed in a bottom-up fashion using low-level features [1], [2], [4], [5], [8]–[12], [14], [15], [18], [19], [22]–[24], [29], [31]–[40], in a top-down fashion by training with certain samples driven by specific tasks [3], [6], [7], [17], [26]–[28], [30], or in a way of combining both low-level and high-level features [16], [20], [21], [25], [48], [50]. Recently, researchers start to use deep features for saliency detection [13], [41], [42], [46], [47], [51], [52]. In this work, we focus on salient object detection and utilize both high-level training and low-level features.

Salient object detection algorithms usually generate bounding boxes, binary foreground and background segmentation, or saliency maps which indicate the saliency likelihood of each pixel. Over the past several years, contrast-based methods [14], [15], [19], [20] significantly promote the benchmark of salient object detection. However, these methods usually miss small local salient regions or bring some outliers such that the resultant saliency maps tend to be nonuniform. To tackle these problems, diffusion-based methods [22], [23], [28], [32], [36]–[40], [43], [44] use diffusion matrices to propagate saliency information of seeds to the whole salient object. While most of them focus on constructing good graph structures, generating good seed vectors and/or controlling the diffusion process, they have not yet made sufficient effort in analyzing the fundamental working mechanism of the diffusion process and accordingly addressing the inherent problems with the diffusion-based approaches.

The existing diffusion-based methods more or less follow a restricted framework, *i.e.*, a specific diffusion matrix is defined in specific feature space and scale based on a specific graph structure, usually with the seed saliency vector computed according to specific color-space heuristics. As a result, they usually lack in extensibility and robustness. This has motivated our search in this work for an inclusive and extensible diffusion-based framework that incorporates a large pool of feature spaces, scales, and seeds for robust performance. Major contributions of this work reside in the following aspects.

- **Novel interpretation of the diffusion mechanism**. Through eigen-analysis of the diffusion matrix, we find that: 1) the saliency of a node (called focus node) is equal to a weighted sum of all the seed saliency values, with the weights determined by the similarity in diffusion map between the focus node and each seed node, and 2) since the diffusion map is formed by the eigenvectors and eigenvalues of the diffusion matrix, the process of saliency diffusion has a close relationship with spectral clustering. This novel interpretation provides the foundation for the novel framework and methods proposed in this work.

This work was funded by the National Key Research & Development Plan of China (No. 2017YFB1002603), the National Natural Science Foundation of China (NSFC Grants No. 61702301, 61472223, 61872398 and 61602273) and the Fundamental Research Funds of Shandong University.

P. Jiang, Z. Pan, C. Tu and J. Peng are with Shandong University, China. (E-mails:{sdujump, panzhiyi1996, changhe.tu, jingliap}@gmail.com). N. Vasconcelos is with University of California, San Diego, USA. (E-mail: nvasconcelos@ucsd.edu). B. Chen is with Peking University, China. (E-mail: baoquan.chen@gmail.com)

J. Peng is the corresponding author.

- **Super diffusion framework for salient object detection**. We propose an inclusive and extensible framework, named super diffusion, for salient object detection, which computes the optimal diffusion matrix by exploiting a pool of feature spaces, scales and even saliency features originally computed for non-diffusion-based salient object detection. The parameters for the optimal integration are derived in a closed-form solution through supervised learning. This contrasts with traditional diffusion-based methods that define the diffusion matrices and seeds with high specificity, compromising the robustness of performance.
- **Local refinement of saliency diffusion**. We propose to promote each individual saliency diffusion scheme prior to its integration into the overall super diffusion framework. Based on the close relationship between saliency diffusion and spectral clustering, the promotion is achieved by re-synthesizing an individual diffusion matrix from the most discriminative eigenvectors and the constant eigenvector (for saliency normalization). In addition, we propose efficient and effective ways to compute seed vectors based on background and foreground priors.

It should be noted that an initial version of this work was published as a conference paper [38], which has been extended to this journal version mainly in the following aspects: 1) proposal of the super diffusion framework with full-length explanation, 2) significantly extended experiments to evaluate the proposed framework in concrete implementation, and 3) more comprehensive coverage and analysis of related works.

To help the clarity of description, we list in Tab. I all the mathematical notations that are used later in the text.

TABLE I
A LIST OF NOTATIONS DEFINED IN THIS PAPER

Notation	Dimension	Description
N	$[1]$	Number of superpixels
v_i	$[1]$	Mean feature value of the i-th node
$\mathbf{W}$	$[N, N]$	Affinity matrix
$\mathbf{D}$	$[N, N]$	Degree matrix
$\mathbf{P}$	$[N, N]$	Transition matrix
$\mathbf{L}$	$[N, N]$	Laplacian matrix
$\mathbf{L}_{rw}$	$[N, N]$	Normalized Laplacian matrix
$\mathbf{A}^{-1}$	$[N, N]$	Diffusion matrix
$\mathbf{s}$	$[N, 1]$	Seed vector
$\mathbf{y}$	$[N, 1]$	Saliency vector
$\mathbf{u}_l$	$[N, 1]$	The l-th eigenvector of $\mathbf{A}$
$\mathbf{U}$	$[N, N]$	Matrix of eigenvectors
λ_l	$[1]$	The l-th eigenvalue of $\mathbf{A}$
$\mathbf{\Lambda}$	$[N, N]$	Diagonal matrix of eigenvalues
$\tilde{a}(i, j)$	$[1]$	The (i, j)-th element of $\mathbf{A}^{-1}$
$\mathbf{\Psi}_i$	$[N, N]$	Diffusion map for the i-th node
${\mathbf{A}^j}^{-1}$	$[N, N]$	The j-th diffusion matrix
$\mathbf{U}^j, \mathbf{\Lambda}^j$	$[N, N]$	Eigenvector and diagonal eigenvalue matrices for $\mathbf{A}^j$
s^j	$[N, 1]$	The j-th seed vector
${\mathbf{A}_I}^{-1}$	$[N, N]$	Integrated diffusion matrix
$\mathbf{s}_I$	$[N, 1]$	Integrated seed vector
$\mathbf{\Psi}_{I_i}$	$[N, N \times M]$	Integrated diffusion map for the i-th node
$\mathbf{y}_I$	$[N, 1]$	Integrated saliency vector
$\mathbf{y}^{i,j}$	$[N, 1]$	Separate Saliency vector
${\bar{\mathbf{A}}^j}^{-1}$	$[N, N]$	The j-th diffusion matrix prior to normalization
$\bar{\mathbf{y}}^{i,j}$	$[N, 1]$	Separate Saliency vector prior to normalization
${\hat{\mathbf{A}}^j}^{-1}$	$[N, N]$	The j-th diffusion matrix after local refinement
$\hat{\mathbf{y}}^{i,j}$	$[N, 1]$	Separate Saliency vector after local refinement
$\hat{\mathbf{y}}_I$	$[N, 1]$	Final integrated saliency vector
$\mathbf{w}$	$[1, M \times K]$	Weight vector to be learned

II. Related Works

Most diffusion-based salient object detection methods, some of which are given in the references [22], [23], [28], [36]–[40], [43], [44], share the same main formula:

$$\mathbf{y} = \mathbf{A}^{-1}\mathbf{s}, \tag{1}$$

where $\mathbf{A}^{-1}$ is the diffusion matrix (also called ranking matrix or propagation matrix), $\mathbf{s}$ is the seed vector (diffusion seed), and $\mathbf{y}$ is the final saliency vector to be computed. Here $\mathbf{s}$ usually contains preliminary saliency information of a portion of nodes, that is to say, usually $\mathbf{s}$ is not complete and we need to propagate the partial saliency information in $\mathbf{s}$ to the whole salient region based on a graph structure to obtain the final saliency map, $\mathbf{y}$ [28]. The diffusion matrix $\mathbf{A}^{-1}$ is designed to fulfill this task. The existing methods mostly focus on how to construct the graph structure, how to generate the seed and/or how to control the diffusion process. Accordingly, we review them based on their approaches to the three sub-problems, respectively, in the following sub-sections[1].

[1]We have noted that some diffusion-based methods, such as the one given in the reference [49], have different working procedures that can not be expressed as Eq. 1 with a constant diffusion matrix. We do not study them in this section, but still make comparisons with them in Sec. V.

A. Graph Construction

A diffusion-based salient object detection algorithm needs to firstly construct a graph structure on a given image for the definition of diffusion matrix. Specifically, it segments the given image into N superpixels first by an algorithm such as SLIC [53] or ERS [54], and then constructs a graph $G = (V, E)$ with superpixels as nodes v_i, $1 \leq i \leq N$, and undirected links between node pairs (v_i, v_j) as edges e_{ij}, $1 \leq i, j \leq N$, to define the adjacency. Note that superpixels but not pixels are usually used as nodes for efficiency and stability considerations.

Straightforwardly, two nodes are connected by an edge in the graph if they are contiguous in the image. In order to capture relationship between nodes farther on the image, some methods [22], [23], [28], [36], [38], [40], [43] connect a node to not only its directly contiguous neighbors, but also its 2-hop and even up to 5-hop neighbors [37]. Besides, some methods [22], [23], [28], [36], [38]–[40], [44] make a close-loop graph by connecting the nodes at the four borders of the image to each other. As a result, the distance between two nodes close to two different borders will be shortened by a path through borders. There is also a method [36] that connects each node to all the nodes at the four borders to increase the connectivity of the graph, which provides certain robustness to noise.

The weight w_{ij} of the edge e_{ij} which encodes the similarity between linked nodes usually is defined as

$$w_{ij} = e^{-\frac{\|v_i - v_j\|_2}{\sigma^2}}, \tag{2}$$

where v_i and v_j represent the mean feature value of two nodes respectively, and σ is a scale parameter that controls the strength of the weight. All the mentioned diffusion-based methods use the CIE LAB color space as feature space. Finally, the affinity matrix is defined as $\mathbf{W} = [w_{ij}]_{N\times N}$ with w_{ij} computed by Eq. 2 if $i = j$ or edge e_{ij} exists in the graph and assigned 0 otherwise; the degree matrix is defined as $\mathbf{D} = diag\{d_{11}, ..., d_{NN}\}$, where $d_{ii} = \sum_j w_{ij}$.

B. Diffusion Matrix and Seed Computation

Different algorithms derive diffusion matrices and seed vectors in different ways. Some algorithms [22], [36], [40] use inverse Laplacian matrix $\mathbf{L}^{-1}$ as the diffusion matrix. Correspondingly, the formula of saliency diffusion is

$$\mathbf{y} = \mathbf{L}^{-1}\mathbf{s}, \tag{3}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Inverse normalized Laplacian matrix $\mathbf{L}_{rw}^{-1}$ is also used by some algorithms [28], [39], [44] as the diffusion matrix which normalizes weights by degrees of nodes when computing similarity. Correspondingly, the formula of saliency diffusion is

$$\mathbf{y} = \mathbf{L}_{rw}^{-1}\mathbf{s}, \tag{4}$$

where $\mathbf{L}_{rw} = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{W}) = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{W})$. Some algorithms [22], [36], [39], [40] use binary background and foreground indication vectors as seed vectors in two stages, respectively. Seed vector $\mathbf{s}$ is also computed by combining hundreds of saliency features $\mathbf{F}$ with learned weight $\mathbf{w}$ ($\mathbf{s} = \mathbf{F}\mathbf{w}$) [28].

One method [23] works differently by duplicating the superpixels around the image borders as the virtual background absorbing nodes and setting the inner nodes as transient nodes. Then, the entry of seed vector $s_i = 1$ if node v_i is a transient node and $s_i = 0$ otherwise. Correspondingly, the formula of saliency diffusion is

$$\mathbf{y} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{s} = \mathbf{L}_{rw}^{-1}\mathbf{s}, \tag{5}$$

where $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ and $\mathbf{P}$ is called transition matrix. Note that Eq. 5 is derived from but not identical to the original formula in the reference [23] and the derivation process is described in Appendix A. Another method [43] also uses a variant of Laplacian matrix to emphasize consistency between neighbor nodes.

In general, the existing diffusion-based salient object detection methods derive their diffusion matrices from the basic form of Laplacian matrix. As a result, their performance is restricted by the Laplacian matrix that makes the performance sensitive to the scale parameter and the feature space used for the matrix construction.

C. Diffusion Process Control

Applying Eq. 1 for once to complete the salient object detection task may not produce satisfactory results, as the seed saliency information may diffuse to the non-salient region or may not diffuse to the whole salient region. One common way to control the diffusion process is by applying multi-stage diffusion instead of one-stage diffusion. Some algorithms [22], [36], [39], [40] diffuse to estimate a non-saliency map using the background prior, and reverse and threshold the map to get the most salient seed nodes at the first stage; they conduct another pass of diffusion at the second stage with the seed saliency estimated at the first stage.

Another algorithm [39] further divides each pass of diffusion into a sequence of steps that, instead of computing saliency of all nodes at once, estimates saliency of a subset of nodes as selected according to certain rules. Though effective to a certain extent, these approaches lack in theoretical support and may not be robust in general.

To summarize, researchers have devised good ways to construct the graph structures, the diffusion matrices and the seed vectors exploiting effective heuristics and priors. In this work, we step further to explore novel views of the fundamental diffusion mechanism and, accordingly, make the systematic promotion of the diffusion-based salient object detection performance.

III. Diffusion Re-Interpreted

As discussed before, diffusion-based salient objection detection algorithms [22], [23], [28], [36], [37], [39], [40], [43], [44] usually define diffusion matrices by certain forms of the Laplacian matrix, denoted by $\mathbf{A}$, which is real, symmetric and positive semi-definite. As $\mathbf{A}$ is a real symmetric matrix, its eigenvalues and eigenvectors are all real and its eigenvectors are orthogonal to each other. Therefore, $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{\Lambda}$ is a diagonal matrix formed from the eigenvalues λ_l, $l = 1, 2, \ldots, N$, and the columns of $\mathbf{U}$ are the corresponding eigenvectors $\mathbf{u}_l$, $l = 1, 2, \ldots, N$. Accordingly, $\mathbf{A}^{-1} = \left(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\right)^{-1} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^T$, and each element, $\tilde{a}(i,j)$, of $\mathbf{A}^{-1}$ can be expressed as

$$\tilde{a}(i,j) = \sum_{l=1}^{N} \lambda_l^{-1}\mathbf{u}_l(i)\mathbf{u}_l(j), \tag{6}$$

where $\mathbf{u}_l(i)$ denotes the i-th element of $\mathbf{u}_l$. Then, each entry, y_i, of $\mathbf{y}$ is computed as

$$\begin{aligned} y_i &= \sum_{j=1}^{N} \mathbf{s}_j \sum_{l=1}^{N} \lambda_l^{-1}\mathbf{u}_l(i)\mathbf{u}_l(j) \\ &= \sum_{j=1}^{N} \mathbf{s}_j \left\langle \mathbf{\Psi}_i, \mathbf{\Psi}_j \right\rangle, \end{aligned} \tag{7}$$

$$\mathbf{\Psi}_i = [\lambda_1^{-\frac{1}{2}}\mathbf{u}_1(i), ..., \lambda_N^{-\frac{1}{2}}\mathbf{u}_N(i)], \tag{8}$$

where $\langle\cdot,\cdot\rangle$ is the inner product operation. According to the reference [55], $\mathbf{\Psi}_i$ is called diffusion map (diffusion map at time $t = -\frac{1}{2}$ to be more exactly) at the i-th data point (node).

From Eq. 7, we see that y_i (saliency value for the i-th superpixel) is equal to the weighted sum of all the seed saliency values and the weight for each seed is determined by the inner product of diffusion maps defined by Eq. 8. Accordingly, we make a novel interpretation of the working mechanism of diffusion-based salient object detection: the saliency of a node (called focus node) is determined by all the seed saliency values in the form of weighted sum, with each

weight determined by diffusion map similarity (measured by inner product) between the corresponding seed node and the focus node.

According to Eq.s 7 and 8, nodes with similar (*resp*., distinct) diffusion maps tend to obtain similar (*resp*., distinct) saliency values. Therefore, the process of saliency diffusion is closely related to the clustering of the nodes based on their diffusion maps. Further, diffusion maps are derived from the eigenvalues and eigenvectors of the diffusion matrix, *i.e*., we form a matrix by putting the weighted eigenvectors in columns and each row of the matrix gives one node's diffusion map (see Eq. 8). As such, the diffusion-map-based clustering is almost identical in form to the standard spectral clustering of the nodes [56], [57].

According to the references [56]–[58], the spectral clustering performance tends to be sensitive to the scale parameter σ and the feature space used for computing the Laplacian matrix (see Eq. 2), and only a subset of the eigenvectors are the most discriminative while the rest are less discriminative or even cause confusions to the clustering. Due to the close relationship between spectral clustering and saliency diffusion, we foresee that the limitations of the spectral clustering also limit the performance of the saliency diffusion. As such, we address these limitations in this work to fundamentally promote the performance of saliency diffusion.

IV. Super Diffusion

A. Generic Framework

As discussed in Sec. III, the performance of saliency diffusion is sensitive to scale parameter and feature space used for Laplacian matrix definition. Therefore, we are motivated to devise a generic and robust scheme to get rid of the sensitiveness of traditional diffusion-based salient object detection methods to feature space and scale (for diffusion matrix definition) and, further, heuristics (for saliency seed definition). Specifically, we propose a framework that systematically integrates diffusion maps originally derived from various diffusion matrices and seed vectors originally derived by various heuristics and optimizes for the best performance. We call this framework *super diffusion*.

Assume that we have M diffusion matrices, $\mathbf{A}^{j^{-1}}$, $1 \leq j \leq M$, each defined on a specific scale and a specific feature space, and the eigen decompositions of them are $\mathbf{A}^{j^{-1}} = \mathbf{U}^j \mathbf{\Lambda}^{j^{-1}} \mathbf{U}^{j^T}$, $1 \leq j \leq M$. For promoted robustness, we propose to construct an integrated diffusion matrix by

$$\begin{aligned} \mathbf{A}_I^{-1} &= \sum_{j=1}^{M} \alpha_j \mathbf{A}^{j^{-1}} \\ &= \mathbf{U}_I \mathbf{\Lambda}_I{}^{-1} \mathbf{U}_I^T \\ &= \mathbf{U}_I \begin{bmatrix} \alpha_1 \mathbf{\Lambda}^{1^{-1}} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \alpha_M \mathbf{\Lambda}^{M^{-1}} \end{bmatrix} \mathbf{U}_I^T, \end{aligned} \tag{9}$$

where $\mathbf{U}_I = \left[\mathbf{U}^1, \cdots, \mathbf{U}^M\right]$, $\mathbf{\Lambda}_I{}^{-1}$ is a diagonal matrix and $\alpha_j \geq 0$, $1 \leq j \leq M$ are the weights to be determined. Note that the weights are constrained to be nonnegative to ensure that $\mathbf{A}_I^{-1}$ is still positive semi-definite.

For a specific saliency seed, $\mathbf{s}$, the corresponding integrated saliency vector $\mathbf{y}_I$ is computed by $\mathbf{y}_I = \mathbf{A}_I^{-1}\mathbf{s} = \sum_{j=1}^{M} \alpha_j \mathbf{A}^{j^{-1}} \mathbf{s}$. Also referring to Eq. 7, we compute each entry, $\mathbf{y}_{I_i}$, of $\mathbf{y}_I$ as

$$\begin{aligned} \mathbf{y}_{I_i} &= \sum_{j=1}^{M} \alpha_j \sum_{k=1}^{N} \mathbf{s}_k \left\langle \mathbf{\Psi}_i^j, \mathbf{\Psi}_k^j \right\rangle \\ &= \sum_{k=1}^{N} \mathbf{s}_k \left(\sum_{j=1}^{M} \alpha_j \left\langle \mathbf{\Psi}_i^j, \mathbf{\Psi}_k^j \right\rangle \right) \\ &= \sum_{k=1}^{N} \mathbf{s}_k \left\langle \mathbf{\Psi}_{Ii}, \mathbf{\Psi}_{Ik} \right\rangle, \end{aligned} \tag{10}$$

where $\mathbf{\Psi}_i^j$ is the diffusion map associated with $\mathbf{A}^{j^{-1}}$ at the i-th node, and $\mathbf{\Psi}_{Ii}$ is the integrated diffusion map at the i-th node, which is defined as

$$\mathbf{\Psi}_{Ii} = [\alpha_1{}^{\frac{1}{2}} \mathbf{\Psi}_i^1, \alpha_2{}^{\frac{1}{2}} \mathbf{\Psi}_i^2, \cdots, \alpha_M{}^{\frac{1}{2}} \mathbf{\Psi}_i^M]. \tag{11}$$

Further, assume that we have K saliency seeds, $\mathbf{s}^1, \mathbf{s}^2, \cdots, \mathbf{s}^K$, each defined by specific heuristics. For promoted robustness, we construct an integrated seed vector by

$$\mathbf{s}_I = [\mathbf{s}^1, \mathbf{s}^2, \cdots, \mathbf{s}^K][\beta_1, \beta_2, \cdots, \beta_K]^T, \tag{12}$$

with β_k, $1 \leq k \leq K$, being the weights to be determined.

The integrated saliency vector $\mathbf{y}_I$ is finally computed by $\mathbf{y}_I = \mathbf{A}_I^{-1}\mathbf{s}_I$. Referring to Eq.s 9 and 12, we have

$$\begin{aligned} \mathbf{y}_I &= \mathbf{A}_I^{-1}\mathbf{s}_I \\ &= \sum_{i=1}^{M}\sum_{j=1}^{K} \alpha_i\beta_j \mathbf{U}^i \mathbf{\Lambda}^{i^{-1}} \mathbf{U}^{i^T} \mathbf{s}^j \\ &= \sum_{i=1}^{M}\sum_{j=1}^{K} \alpha_i\beta_j \mathbf{A}^{i^{-1}} \mathbf{s}^j \\ &= \sum_{i=1}^{M}\sum_{j=1}^{K} \alpha_i\beta_j \mathbf{y}^{i,j} \\ &= \mathbf{H}\mathbf{w}^T, \end{aligned} \tag{13}$$

where $\mathbf{y}^{i,j} = \mathbf{A}^{i^{-1}}\mathbf{s}^j$ is the separate saliency vector for diffusion matrix $\mathbf{A}^{i^{-1}}$ and saliency seed $\mathbf{s}^j$, $\mathbf{H} = [\mathbf{y}^{1,1}, \cdots, \mathbf{y}^{1,K}, \cdots, \mathbf{y}^{M,1}, \cdots, \mathbf{y}^{M,K}]$ and $\mathbf{w} = [\alpha_1\beta_1, \cdots, \alpha_1\beta_K, \cdots, \alpha_M\beta_1, \cdots, \alpha_M\beta_K]$. With $\mathbf{A}^{i^{-1}}$, $1 \leq i \leq M$, and $\mathbf{s}^j$, $1 \leq j \leq K$, given, the variables of this system are α_i, $1 \leq i \leq M$, and β_j, $1 \leq j \leq K$. In other words, the degree of freedom (DOF) for our solution is $M + K$. In order to increase the room for optimization, we increase the DOF to $M \times K$ by replacing $\mathbf{w}$ in Eq. 13 with $\mathbf{w} = [w_1, w_2, \cdots, w_{M\times K}]$ and solving for w_i, $1 \leq i \leq M \times K$, instead.

We determine the weighting vector, $\mathbf{w}$, by supervised learning from a training set of L samples, with the loss function defined as

$$\begin{aligned} J &= \sum_{i=1}^{L} (\mathbf{y}_I(i) - \mathbf{y}_{gt}(i))^2 \\ &= \sum_{i=1}^{L} (\mathbf{H}(i)\mathbf{w}^T - \mathbf{y}_{gt}(i))^2, \end{aligned} \tag{14}$$

where $\mathbf{y}_I(i)$, $\mathbf{y}_{gt}(i)$ and $\mathbf{H}(i)$ are the computed integrated saliency vector, the ground-truth binary saliency vector and the $\mathbf{H}$ matrix for the i-th training sample, respectively. As J is convex, the optimal $\mathbf{w}$ has a closed-form expression of

$$\mathbf{w} = \frac{\sum_{i=1}^{L} \mathbf{H}(i)^T \mathbf{y}_{gt}(i)}{\sum_{i=1}^{L} \mathbf{H}(i)^T \mathbf{H}(i)}. \tag{15}$$

B. Local Refinement

While the proposed framework in Sec. IV-A promotes the robustness by optimally integrating various diffusion matrices and seeds, each individual diffusion matrix on its own may be optimized as well.

As discussed in Sec. III, only a subset of $\mathbf{A}$'s eigenvectors are the most discriminative. Thus, in order to increase the discriminative power of the diffusion maps associated with each specific $\mathbf{A}^i$, $1 \leq i \leq M$, in Sec. IV-A, we are motivated to keep only the most discriminative while discarding the rest of its eigenvectors. Specifically, we refine each individual $\mathbf{A}^{i^{-1}}$ by re-synthesizing it from $\mathbf{A}^i$'s most discriminative eigenvectors followed by a normalization step, as detailed in the following subsections. We call this process local refinement for short.

In practice, we first refine each individual diffusion matrix, $\mathbf{A}^{i^{-1}}$, and then use the refined diffusion matrices to compute all the saliency values in matrix $\mathbf{H}$ in Eq. 13 and $\mathbf{H}(i)$ in Eq.s 14 and 15. Regarding the choice of $\mathbf{A}^i$, $1 \leq i \leq M$, we use a slightly modified $\mathbf{L}_{rw}$, $\tilde{\mathbf{L}}_{rw}$ (*c.f.* Sec. IV-B1), as the basic form and define a series of diffusion matrices by varying the feature space and scale parameter when computing the edge weights (*c.f.* Eq. 2). Our choice is motivated by the fact that $\mathbf{L}_{rw}$ often leads to better intra-cluster coherency and clustering consistency than $\mathbf{L}$ for spectral clustering [57].

1) Constant Eigenvector: The eigenvalues, λ_l, and eigenvectors, $\mathbf{u}_l$, $1 \leq l \leq N$, of $\mathbf{L}_{rw}$ (the same for $\mathbf{L}$) are ordered such that $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ with $\mathbf{u}_1 = \mathbf{1}$ [56]. We avoid zero eigenvalues by approximately setting $\tilde{\mathbf{L}}_{rw} = \mathbf{D}^{-1}(\mathbf{D} - 0.99W)$ such that $\tilde{\mathbf{L}}_{rw}$ is always invertible [22]. Assuming $\tilde{\lambda}_l$ and $\tilde{\mathbf{u}}_l$, $1 \leq l \leq N$, are the corresponding eigenvalues and eigenvectors of $\tilde{\mathbf{L}}_{rw} = \mathbf{D}^{-1}(\mathbf{D} - 0.99\mathbf{W})$, it can be proven that $\tilde{\mathbf{u}}_l = \mathbf{u}_l$ and $\tilde{\lambda}_l = 0.99\lambda_l + 0.01$. Thus, $0.01 = \tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \ldots \leq \tilde{\lambda}_N$ with $\tilde{\mathbf{u}}_1 = \mathbf{1}$.

The constant eigenvector $\tilde{\mathbf{u}}_1$ contains no discriminative information. Thus, we discard it and re-synthesize the diffusion matrix, as done in our early conference version [38] of this work. But novelly, we reuse the constant eigenvector later in Sec. IV-B4 for normalization of saliency.

2) Eigengap: In each diffusion matrix, except $\mathbf{u}_1$ that is a constant vector, the more $\mathbf{u}_l$ ($l \in [2, N]$) is to the front of the ordered array, the more indicative it usually is for the clustering. For instance, we visualize in Fig. 1 a leading portion (excluding $\mathbf{u}_1$) of the ordered array of eigenvectors for each of four sample images. From Fig. 1, we see that, for each sample image, the first few eigenvectors well indicate node clusters while the later ones often convey less or even confusing information about the clustering. The key is how to determine the exact cutting point before which the eigenvectors should be kept and after which discarded.

In practice, $\mathbf{L}_{rw}$ (the same for $\mathbf{L}$) often exhibits an eigengap, *i.e.*, a few of its eigenvalues before the eigengap are much smaller than the rest. Specifically, we denote the eigengap as r and define it as

$$\begin{aligned} r &= \underset{l}{\operatorname{argmax}} |\Delta\Upsilon_l|, \\ \Delta\Upsilon_l &= \lambda_l - \lambda_{l-1},\ l = 2, \ldots, N. \end{aligned} \tag{16}$$

Usually, Eq. 16 is called eigengap heuristic. According to [57], some leading eigenvectors (except u_1) before the eigengap are usually good cluster indicators which can capture the data cluster information with good accuracy (as observed in Fig. 1), meanwhile the location of the eigengap often indicates the right number of data clusters. Further, the larger the difference between the two successive eigenvalues at the eigengap is, the more important the leading eigenvectors are, since u_l is weighted by $\lambda_l^{-\frac{1}{2}}$ in diffusion map $\mathbf{\Psi}$ (*c.f.* Eq. 8). Ideally, the eigenvalues before the eigengap are close to zero while the rest are much larger, which means that the leading eigenvectors (except u_1) will dominate the behavior of the diffusion map.

With the eigengap identified, we then keep only the eigenvectors prior to the eigengap, which are usually the most discriminative ones for the task of node clustering. It may sometimes happen that $r = 2$ according to Eq. 16, meaning that all the eigenvectors will be filtered out. In this case, we assume the position of the second largest $|\Delta\Upsilon_l|$ as the eigengap.

3) Discriminability: In some cases, an eigenvector may only distinguish a tiny region from the background, *e.g.*, $\mathbf{u}_5$, $\mathbf{u}_6$ in the second row and $\mathbf{u}_6$ in the last row of Fig. 1. Usually, these tiny regions are less likely to be the salient regions we search for. Besides, these tiny regions often have been captured by other leading eigenvectors as well. Therefore, such eigenvectors usually have low discriminability and may even worsen the final results by overemphasizing tiny regions.Therefore, we evaluate the discriminability of eigenvector $\mathbf{u}_l$ by its variance $var(\mathbf{u}_l)$, and filter out eigenvectors with variance values below a threshold, v.

4) Normalization: After the above local refinement operations, each original diffusion matrix $\mathbf{A}^{i^{-1}}$ becomes $\bar{\mathbf{A}}^{i^{-1}} = \bar{\mathbf{U}}^i \bar{\mathbf{\Lambda}}^{i^{-1}} \bar{\mathbf{U}}^{i^T}$ ($1 \leq i \leq M$) in Eq. 13. Immediately, we may compute $\bar{\mathbf{H}}(i)$ of the i-th ($1 \leq i \leq L$) training sample using its refined diffusion matrices to replace $\mathbf{H}(i)$ in Eq. 15 and obtain $\mathbf{w}$. However, this usually is problematic as the saliency vectors computed on different samples and/or by different matrix-seed combinations often exhibit inconsistent ranges of componential values. Therefore, in order to derive

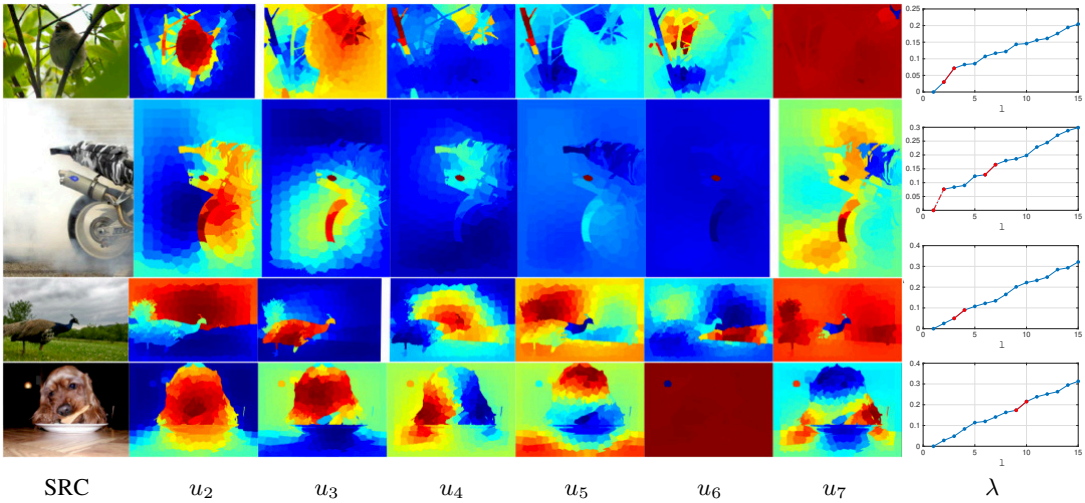

Fig. 1. Visualization of normalized eigenvectors by color coding. Pixels in each node are assigned a single color and nodes with similar values in an eigenvector are colored similarly. The eight columns show the source images (SRC), the corresponding eigenvectors ($\mathbf{u}_2$-$\mathbf{u}_7$) and eigenvalue curves (λ), respectively. We use a white margin between successive eigenvectors to indicate an eigengap (all the eigenvectors, $\mathbf{u}_2$ to $\mathbf{u}_7$, are before the eigengap, if there is no white margin in that row.). Besides, on the eigenvalue curves, we use red solid segments to indicate the final eigengaps and a red dash segment to indicate an initial eigengap of $r = 2$ to be reset.

an optimal $\mathbf{w}$ of generic applicability, we first normalize the saliency vector of each sample computed by each matrix-seed combination, as explained below.

On each specific image sample, for each matrix-seed combination $\left(\bar{\mathbf{A}}^{i^{-1}}, \mathbf{s}^j\right)$, $1 \le i \le M$, $1 \le j \le K$, we need to normalize the saliency vector, $\bar{\mathbf{y}}^{i,j} = \bar{\mathbf{A}}^{i^{-1}}\mathbf{s}^j = \bar{\mathbf{U}}^i\bar{\mathbf{\Lambda}}^{i^{-1}}\bar{\mathbf{U}}^{i^T}\mathbf{s}^j$, to range its componential values to $[0, 1]$. It is commonly known that a vector $\mathbf{x}$ whose componential values extend a range of $[p, q]$ may be normalized by

$$\begin{aligned} \hat{\mathbf{x}} &= b\mathbf{1} + \frac{\mathbf{x}}{a}, \\ b &= \frac{p}{p-q}, \\ a &= -(p-q). \end{aligned} \tag{17}$$

Similarly, we normalize $\bar{\mathbf{y}}^{i,j}$ with a componential value range of $[p, q]$ by

$$\begin{aligned} \hat{\mathbf{y}}^{i,j} &= [\mathbf{u}_1, \bar{\mathbf{U}}^i] \begin{bmatrix} {\lambda'_1}^{-1} & \mathbf{0}^T \\ \mathbf{0} & \frac{1}{\hat{a}}\bar{\mathbf{\Lambda}}^{i^{-1}} \end{bmatrix} [\mathbf{u}_1, \bar{\mathbf{U}}^i]^T\mathbf{s}^j \\ &= {\lambda'_1}^{-1}\mathbf{u}_1\mathbf{u}_1^T\mathbf{s}^j + \frac{1}{\hat{a}}\bar{\mathbf{U}}^i\bar{\mathbf{\Lambda}}^{i^{-1}}\bar{\mathbf{U}}^{i^T}\mathbf{s}^j \\ &= {\lambda'_1}^{-1}\mathbf{u}_1\mathbf{u}_1^T\mathbf{s}^j + \frac{\bar{\mathbf{y}}^{i,j}}{\hat{a}}, \end{aligned} \tag{18}$$

where $\mathbf{u}_1$ is the constant vector and λ'_1 and $\hat{a}$ are scalars to be determined. By comparing Eq.s 17 and 18, we set $\hat{a} = a$ and ${\lambda'_1}^{-1}\mathbf{u}_1\mathbf{u}_1^T\mathbf{s}^j = b\mathbf{1}$ for the normalization. Equivalently, we have $\hat{a} = -(p-q)$ and $\lambda'_1 = \sum_{i=1}^{N}\mathbf{s}^j(i)(p-q)/p$.

In essence, the above normalization process refines $\bar{\mathbf{A}}^{i^{-1}}$ to

$$\begin{aligned} \hat{\mathbf{A}}^{i,j^{-1}} &= \hat{\mathbf{U}}^i\hat{\mathbf{\Lambda}}^{i^{-1}}\hat{\mathbf{U}}^{i^T} \\ &= [\mathbf{u}_1, \bar{\mathbf{U}}^i] \begin{bmatrix} {\lambda'_1}^{-1} & \mathbf{0}^T \\ \mathbf{0} & \frac{1}{\hat{a}}\bar{\mathbf{\Lambda}}^{i^{-1}} \end{bmatrix} [\mathbf{u}_1, \bar{\mathbf{U}}^i]^T, \end{aligned} \tag{19}$$

which is used as the final diffusion matrix for $\mathbf{s}^j$ on the specific image sample.

Using the finally refined diffusion matrices, we compute $\hat{\mathbf{H}}(i)$ for the i-th ($1 \le i \le L$) training sample to replace $\mathbf{H}(i)$ in Eq. 15 and finally obtain the solution of $\mathbf{w}$.

C. Choice of Seeds

We utilize the foreground and background prior to design two kinds of seed vectors and use them as $\mathbf{s}^j$, $1 \le j \le K$ in Eq.s 12 and 13.

Firstly, we assume that nodes closer to the center of image are more salient, and initialize a sequence of Gaussian-filter-like images (with different variances) to compute the first kind of seed vectors, as people usually put salient objects in the central foreground area when taking a photo.

Secondly, we assume that nodes located at the border of image are the least salient, and compute the time that other non-border nodes random walk to them to form the seed vector. Nodes that take more time to reach the border nodes are more salient. Note that the transition matrix of random walk can also be derived from the highly discriminative diffusion matrices, $\bar{\mathbf{A}}^{i^{-1}}$, $1 \le i \le M$, as explained in Appendix A with an in-depth analysis of the working mechanism of this proposed seed vector construction method.

The foreground and background prior leads to not only good accuracy of seed value estimation, but also high time-efficiency as it avoids an extra pass of color-based preliminary saliency search.

D. Implementation Details

When constructing the graph, we have $N = 2000$ superpixels, and, in order to utilize the cross-node correlation in a broader range, we connect not only nodes that are directly adjacent, but also those that are up to 7 hops apart.

Algorithm 1 Super Diffusion (Training)

Require:
(a) A list of training images, $[I_1, \cdots, I_L]$,
(b) A list of scale parameters, $[\sigma_1, \cdots, \sigma_m]$,
(c) A list of feature spaces, $[f_1, \cdots, f_n]$,
(d) A list of seed computing methods, $[c_1, \cdots, c_K]$,

Initialization:
Segment each training image into N superpixels, use the superpixels as nodes, connect border nodes to each other and connect nodes that are up to 7-hop away to construct a graph G.

Local refinement: For each training image in (a),
1: Compute $\mathbf{A}^i = \mathbf{D}^{i^{-1}}(\mathbf{D}^i - 0.99\mathbf{W}^i)$ and its eigenvectors $\mathbf{U}_i$ and eigenvalues $\mathbf{\Lambda}^i$ for each setting i in combination of (b)(c);
2: For each $\mathbf{A}^i$, discard the constant eigenvector, the eigenvectors after the eigengap or with low discriminability to get $\bar{\mathbf{A}}^{i^{-1}}$, $\bar{\mathbf{U}}^i$, $\bar{\mathbf{\Lambda}}^i$ and $\bar{\mathbf{H}}$ by local refinement operations described in Sec.s IV-B1 to IV-B3;
3: For each $\bar{\mathbf{A}}^{i^{-1}}$ and seed computing method, c_j, in (d),
(i) Compute the seed vector, $\mathbf{s}^j$;
(ii) Re-add the constant eigenvector with an updated eigenvalue, and scale $\bar{\mathbf{\Lambda}}^i$ to normalize $\bar{\mathbf{y}}^{i,j}$ by Eq. 18;
(iii) Correspondingly, re-synthesize $\bar{\mathbf{A}}^{i^{-1}}$ to get the final diffusion matrix $\hat{\mathbf{A}}_{i,j}^{-1}$ by Eq. 19;
4: Integrate all $\hat{\mathbf{y}}^{i,j}$ to get $\hat{\mathbf{H}}$.

Global optimization: With $\hat{\mathbf{H}}(i)$, $1 \leq i \leq L$, for all the training images computed,
5: Substitute $\hat{\mathbf{H}}(i)$ for $\mathbf{H}(i)$ in Eq. 15 to compute the optimal weight $\mathbf{w}$.

Ensure: Weight $\mathbf{w}$.

Furthermore, we connect the nodes at the four borders of an image to each other to make a close-loop graph.

The main training steps of the proposed salient object detection algorithm are summarized in Algorithm 1. As for testing, given an input image I, we conduct the superpixel segmentation and graph construction on it and compute its $\hat{\mathbf{H}}$, following the same initialization and local refinement steps in Alg. 1, and apply the learned weight $\mathbf{w}$ to $\hat{\mathbf{H}}$ to obtain the final integrated saliency vector (after local refinement) $\hat{\mathbf{y}}_I = \hat{\mathbf{H}}\mathbf{w}^T$. Finally, we obtain the saliency map S by assigning the value of $\hat{\mathbf{y}}_{I_i}$ to the corresponding node v_i, $1 \leq i \leq N$.

E. Saliency Features as Diffusion Maps

Diffusion-based salient object detection methods [22], [23], [28], [36]–[40], [43], [44] usually rely on raw color features, *e.g.*, they use the mean color vectors of two linked nodes to compute the edge weight (*c.f.* Eq. 2) and, correspondingly, the affinity matrix and the diffusion matrix. However, the raw color features may sometimes not be well indicative of the saliency. As such, more saliency features have been devised and used by non-diffusion-based salient object detection methods. In particular, hundreds of saliency features for the task of salient object detection are effectively integrated in some algorithms [26], [46], [50], [51]. This has motivated us to integrate more saliency features seamlessly into our super diffusion framework.

By our interpretation of the diffusion mechanism (*c.f.* Sec. III), diffusion maps play a key role in saliency computation and nodes with similar (*resp.*, dissimilar) diffusion maps tend to be assigned similar (*resp.*, dissimilar) saliency values. Therefore, good diffusion maps themselves should be discriminative which are similar for nodes of similar factual saliency and dissimilar otherwise. As saliency features discriminate salient from non-salient nodes, we use them to construct discriminative maps at the nodes to imitate the diffusion process. We call them diffusion maps as well for the convenience of description.

We denote the Z saliency features by $\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^Z$ with each $\mathbf{g}^j$, $j \in [1, Z]$, being an N-dimensional vector containing the corresponding feature values of the nodes. Then we construct a diffusion map for node i by

$$\mathbf{\Psi}'_i = [\mathbf{1}_i, \mathbf{g}_i^1, \mathbf{g}_i^2, \cdots, \mathbf{g}_i^Z]. \tag{20}$$

Incorporating $\mathbf{\Psi}'_i$ into Eq. 11, we update $\mathbf{\Psi}_{Ii}$ to

$$\mathbf{\Psi}_{Ii} = [\alpha_1^{\frac{1}{2}}\mathbf{\Psi}_i^1, \alpha_2^{\frac{1}{2}}\mathbf{\Psi}_i^2, \cdots, \alpha_M^{\frac{1}{2}}\mathbf{\Psi}_i^M, \alpha_{M+1}^{\frac{1}{2}}\mathbf{\Psi}_i^{M+1}], \tag{21}$$

with $\mathbf{\Psi}_i^{M+1} = \mathbf{\Psi}'_i$. Correspondingly, we make $\mathbf{A}^{M+1^{-1}} = \mathbf{U}^{M+1}\mathbf{\Lambda}^{M+1^{-1}}\mathbf{U}^{M+1^T}$ with $\mathbf{U}^{M+1} = [\mathbf{1}, \mathbf{g}^1, \mathbf{g}^2, \cdots, \mathbf{g}^Z]$ and $\mathbf{\Lambda}^{M+1^{-1}} = diag\{1, 1, \cdots, 1\}$, and update $\mathbf{A}_I^{-1}$ in Eq. 9 to

$$\begin{aligned}\mathbf{A}_I^{-1} &= \mathbf{U}_I{\mathbf{\Lambda}_I}^{-1}\mathbf{U}_I^T \\ &= \mathbf{U}_I \begin{bmatrix} \alpha_1\mathbf{\Lambda}^{1^{-1}} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \alpha_{M+1}\mathbf{\Lambda}^{M+1^{-1}} \end{bmatrix} \mathbf{U}_I^T,\end{aligned} \tag{22}$$

where $\mathbf{U}_I = \left[\mathbf{U}^1, \cdots, \mathbf{U}^{M+1}\right]$. Further, we update $\mathbf{H}$ and $\mathbf{w}$ by $\mathbf{H} = [\mathbf{y}^{1,1}, \cdots, \mathbf{y}^{1,K}, \cdots, \mathbf{y}^{M+1,1}, \cdots, \mathbf{y}^{M+1,K}]$ and $\mathbf{w} = [w_1, w_2, \cdots, w_{(M+1)\times K}]$ for Eq.s 13, 14 and 15.

Finally, for the training and the testing, the procedures described in the previous sections are still conducted except that the steps in Sec.s IV-B1–IV-B3 are not applied on any $\mathbf{A}^{M+1^{-1}}$ as it is not a common graph-based diffusion matrix. But still, the normalization step in Sec. IV-B4 is conducted for each matrix-seed combination, $\left(\mathbf{A}^{M+1^{-1}}, \mathbf{s}^j\right)$, $1 \leq j \leq K$, on each specific image sample.

V. Experiments and Analysis

A. Datasets and Evaluation Methods

Our experiments are mainly conducted on four datasets: the MSRA10K dataset [15], [33] with $10K$ images, the MSRA-B dataset [26] with $5K$ images (MSRA-B contains many images from MSRA10K), the DUT-OMRON dataset [22] with $5K$ images and the ECSSD dataset [24] with $1K$ images. Each image in these datasets is associated with a human-labeled ground truth.

In order to study the performance of our final super diffusion method, we adopt prevalently used evaluation protocols including precision-recall (PR) curves [14], F-measure score which is a weighted harmonic of precision and recall [14],

mean overlap rate (MOR) score [25], area under ROC curve (AUC) score [28] and mean square error (MSE), as described in Sec. V-E. Among these protocols, F-measure and MOR require firstly thresholding the images, and we set the threshold as twice the mean saliency value over the ground truth set. Further, to analyze how much the local refinement operations benefit our method, we propose to measure the quality of a diffusion matrix by visual saliency promotion and constrained optimal seed efficiency (COSE), as detailed in Sec. V-C and Sec. V-D, respectively. Finally, in Sec. V-F, we give an ablation study of all the global and local refinement operations, to show the effects of different steps in Alg. 1.

B. Experimental Settings

We choose 11 different settings for the scale parameter σ, $\sigma^2 \in [10, 11, \cdots, 20]$, and 3 different color spaces, Lab, RGB and HSV, for the feature space, which leads to $11 \times 3 = 33$ different diffusion matrices, *i.e.*, $M = 33$ for Eq.s 11, 9 and 13. We set $v = 300$ as the threshold to filter out eigenvectors of low discriminability in Sec. IV-B3. For the first kind of seed vectors, we take the Gaussian variance from $\{0.5, 1, 2\}$. We integrate the saliency features of the work [26] into our super diffusion framework (*c.f.* Sec. IV-E). For each dataset, we use a half of the images for training, and the other half for testing and evaluation. In Sec. V-C and Sec. V-D, in order to avoid zero eigenvalues, we approximately set $\tilde{\mathbf{L}}_{rw} = \mathbf{D}^{-1}(\mathbf{D} - 0.99\mathbf{W})$ and $\tilde{\mathbf{L}} = \mathbf{D} - 0.99\mathbf{W}$ when comparing diffusion matrices, as done in the reference [22]. However, our each diffusion matrix $\hat{\mathbf{A}}^j$ is directly re-synthesized from $\mathbf{L}_{rw} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{W})$ by the local refinement.

To comprehensively report the effectiveness of our proposed local refinement operations in Sec. IV-B, in Sec. V-C and Sec. V-D, we design two experiments to compare the diffusion results with and without the local refinement. In Sec. V-E and Sec. V-F, we further demonstrate how much our method gets promoted after global enhancement by the integration of diffusions. We have noted that recently published methods start to incorporate deep features to detect saliency and achieve state-of-the-art performance. As such, we introduce deep features as saliency features into our super diffusion framework, and compare with deep learning (feature) based methods in Sec. V-E and Sec. V-F.

Note that, for each dataset, we derive the optimal weights, $\mathbf{w}$, from the training set and apply the learned model on the test set. This approach is justified as follows. Firstly, samples in the same dataset share (more or less) similar properties such as object scale and center bias, such that the optimal weighting learned from the training set may apply well on the test set. Secondly, the logic of learning the weights from the training set and testing the performance on the test set has been widely used in other learning-based methods [17], [26] and proved to be effective.

C. Promotion of Visual Saliency

Visual saliency detection predicts human fixation locations in an image, which are often indicative of salient objects around. Therefore, we use the detected visual saliency as the seed information, and conduct diffusion on it to detect the salient object region in an image. In other words, we promote a visual saliency detection algorithm by diffusion for the task of salient object detection.

In this experiment, we use the results of nine visual saliency detection methods (*i.e.*, IT [1], AIM [7], GB [2], SR [8], SUN [9], SeR [10], SIM [11], SS [4] and COV [12]) on the MSRA10K dataset as the seed vectors, respectively, and compare the saliency detection results before and after diffusion. For the diffusion, we test three matrices including $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$, which are all computed in Lab feature space with $\sigma^2 = 10$. It's worth noting that $\bar{\mathbf{A}}^{1^{-1}}$ is only one of our locally refined diffusion matrices (without normalization yet) before the integration.

The PR curves of the nine visual saliency detection methods before and after diffusion by $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$ are plotted in Fig. 2(a), (b) and (c), respectively. Remarkably, as shown in Fig. 2, previous visual saliency detection methods which usually can not highlight the whole salient object all get significantly boosted after diffusion with any of $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$. The promotion is so significant that some promoted methods even outperform some state-of-the-art salient objection detection methods, as observed by comparing Fig. 2 and Fig. 4. This means that, with a good diffusion matrix, we can fill the performance gap between two branches of saliency detection methods.

Comparing Fig.s 2(a), 2(b) and 2(c), we observe that $\bar{\mathbf{A}}^{1^{-1}}$ leads to more significant performance promotion and more consistent promoted performance than $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$, demonstrating higher effectiveness and robustness of the refined diffusion matrix, $\bar{\mathbf{A}}^{1^{-1}}$, in visual saliency promotion.

D. Constrained Optimal Seed Efficiency

In this section, we design experiments to demonstrate the effectiveness of the proposed local refinement method as proposed in Sec. IV-B. Firstly, we propose constrained optimal seed saliency (OSE) curves to measure the upper bound of a diffusion matrix' potential best performance by using the ground truth to optimize the best seeds. Secondly, we compare the OSE curves of $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$ to show the effectiveness of the proposed local refinement method.

Given the ground truth $\mathbf{GT}$ and the diffusion matrix $\mathbf{A}^{-1}$, we hope to find the optimal seed vector, $\mathbf{s}$, that minimizes the residual, $\mathbf{res}$, computed by

$$\mathbf{res} = \mathbf{GT} - \mathbf{A}^{-1}\mathbf{s}. \tag{23}$$

Aiming to reduce the number of non-zero values in $\mathbf{s}$, we turn the residual minimization to a sparse recovery problem, to solve which we adapt the algorithm of orthogonal matching pursuit (OMP) [59], as described in Alg. 2.

As shown in Alg. 2, we adapt the residual computation to $\tilde{\mathbf{res}} = \mathbf{GT} - bin(\mathbf{A}^{-1}\mathbf{s})$ in Step 4, where bin is the binarization operation since $\mathbf{GT}$ is binary; we multiply a factor $\mathbf{GT}(j)$ in Step 1 to ensure that the non-zero seed values are selected from only the salient region; we solve the nonnegative least-squares problem in Step 3 to ensure nonnegative elements of $\mathbf{s}$. The adapted OMP will stop when $\|\tilde{\mathbf{res}}\|_2$ is below a threshold,

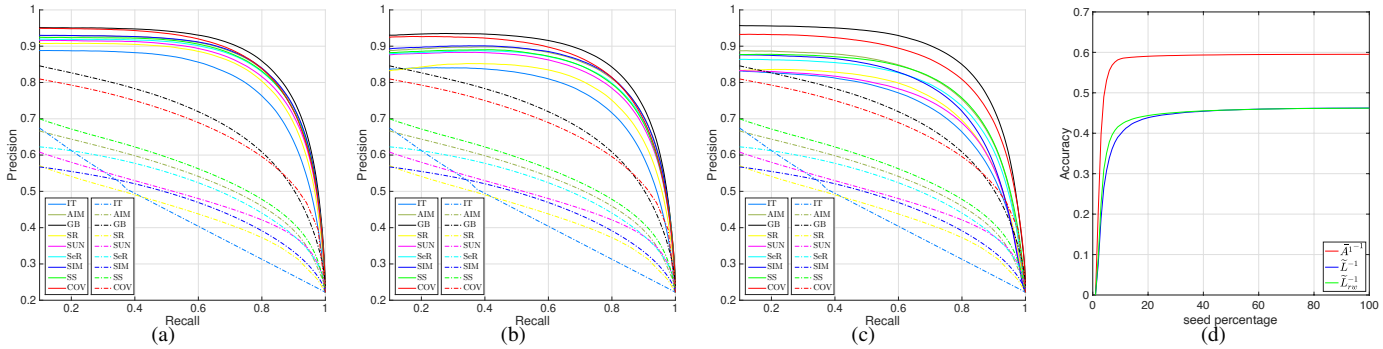

Fig. 2. PR curves of nine visual saliency detection methods before (dash line) and after (solid line) diffusion by (a) $\bar{\mathbf{A}}^{1^{-1}}$, (b) $\tilde{\mathbf{L}}^{-1}$, and (c) $\tilde{\mathbf{L}}_{rw}^{-1}$. The constrained optimal seed efficiency curves for $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$ on the MSRA10K dataset are shown in (d).

c, or the nonnegative seed values at the salient region are all selected, as shown in Step 5. We see that the optimization process in Alg. 2 is constrained, *e.g.*, the seeds are selected from only the salient region, the optimization is conducted in a greedy fashion and so forth. Although the saliency detection performance of these resultant seed vectors provides a good reference for our diffusion matrix evaluation, it should be noted that their optimal performance is constrained but not absolute. Accordingly, we name the measured performance as constrained optimal saliency detection accuracy.

In order to obtain the OSE curve over the full range of nonnegative seed value budget, we set $c = 0$ in Alg. 2 and, at the i-th ($0 \leq i \leq 100$) iteration, we compute and record the pair of nonnegative seed percentage, r_i, and saliency detection accuracy, a_i, according to the following formulae:

$$\begin{aligned} r_i &= \frac{100 \times \|\mathbf{s}\|_0}{\|\mathbf{GT}\|_0}\%, \\ a_i &= \frac{\|\mathbf{GT}\|_2 - \|\mathbf{r\tilde{e}s}\|_2}{\|\mathbf{GT}\|_2}. \end{aligned} \tag{24}$$

Based on these (r_i, a_i) pairs, we can plot the OSE curve of $\mathbf{A}^{-1}$ on an image.

We substitute $\bar{\mathbf{A}}^{1^{-1}}$, $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$ in the last section into Eq. 23 for $\mathbf{A}^{-1}$, respectively. For each diffusion matrix, we plot the average OSE curve over all the images in the MSRA10K dataset, as shown in Fig. 2(d). From Fig. 2(d), we observe that the constrained optimal seed efficiency rises sharply at the beginning and levels off at around the nonnegative seed percentage of 30%, that $\bar{\mathbf{A}}^{1^{-1}}$ exhibits significantly higher average constrained optimal seed efficiency than $\tilde{\mathbf{L}}^{-1}$ and $\tilde{\mathbf{L}}_{rw}^{-1}$, and that there is an inherent performance upper bound for each diffusion matrix while $\bar{\mathbf{A}}^{1^{-1}}$ has the highest one. According to the last observation, it appears that the performance of diffusion-based saliency detection is fundamentally determined by the diffusion matrix, again emphasizing the importance in constructing a good diffusion matrix.

E. Salient Object Detection

We experimentally compare our method with ten other recently proposed ones including GMR [22], MC [23], GP [38], DRFI [26], SS [44], ELE [48], HCA [49], DHS [47], DCL [42] and DSS [52] on salient object detection. When evaluating these methods, we either use the results from the original authors, if available, or run our own implementations. Among these methods, GMR [22], MC [23], GP [38], SS [44] and HCA [49] are the diffusion-based methods that lead to outstanding performance, and DRFI [26] is the approach that integrates hundreds of saliency features and yields top performance on the saliency benchmark study [60]. Deep learning is used in DHS [47], DCL [42] and DSS [52] and results in state-of-the-art performance. Deep features together with diffusion mechanism are used to predict saliency in ELE [48] and HCA [49], which also achieve comparable results as deep learning based methods. To compare with these salient object detection methods fairly, we design several variants of our supper diffusion framework. Ours(N) is our super diffusion method without using other saliency features, Ours(F) is our super diffusion method with saliency features of DRFI [26] integrated and Ours(F_All) is our super diffusion method with deep features of DHS [47], DCL [42] and DSS [52] further integrated upon Ours(F) as additional saliency features.

Among these compared methods, DRFI [26], ELE [48], DHS [47], DCL [42] and DSS [52] are supervised methods which make different partitionings of training and test sets on the ECSSD and DUT-OMRON datasets (We omit MSRA10K

Algorithm 2 Adapted Orthogonal Matching Pursuit

Require: Dictionary($\mathbf{A}_{N\times N}^{-1}$), Signal($\mathbf{GT}_{N\times 1}$) and Stop criterion($c$).
Ensure: Coefficient vector($\mathbf{s}_{N\times 1}$) and Residual($\mathbf{res}$).
Initialization: $\mathbf{res} = \mathbf{GT}$, $Inds = \emptyset$,
$FgInds = \arg_i\{\mathbf{GT}(i) = 1\}$.
Iteration:
1: $ind = \operatorname{argmax}_j\{|\langle \mathbf{res}, \mathbf{A}^{-1}(:,j)\rangle| \cdot \mathbf{GT}(j)\}$, $j \in FgInds$;
2: $Inds = Inds \cup ind$, $FgInds = FgInds \setminus ind$;
3: $\mathbf{s}(Inds) = \operatorname{argmin}_{\tilde{\mathbf{s}}\geq 0}\|\mathbf{GT} - \mathbf{A}^{-1}(:, Inds)\tilde{\mathbf{s}}\|_2$;
4: $\mathbf{r\tilde{e}s} = \mathbf{GT} - bin(\mathbf{A}^{-1}\mathbf{s})$;
5: **if** $\|\mathbf{r\tilde{e}s}\|_2 \geq c \wedge FgInds \neq \emptyset$ **then**
6: **Go to** 1;
7: **end if**

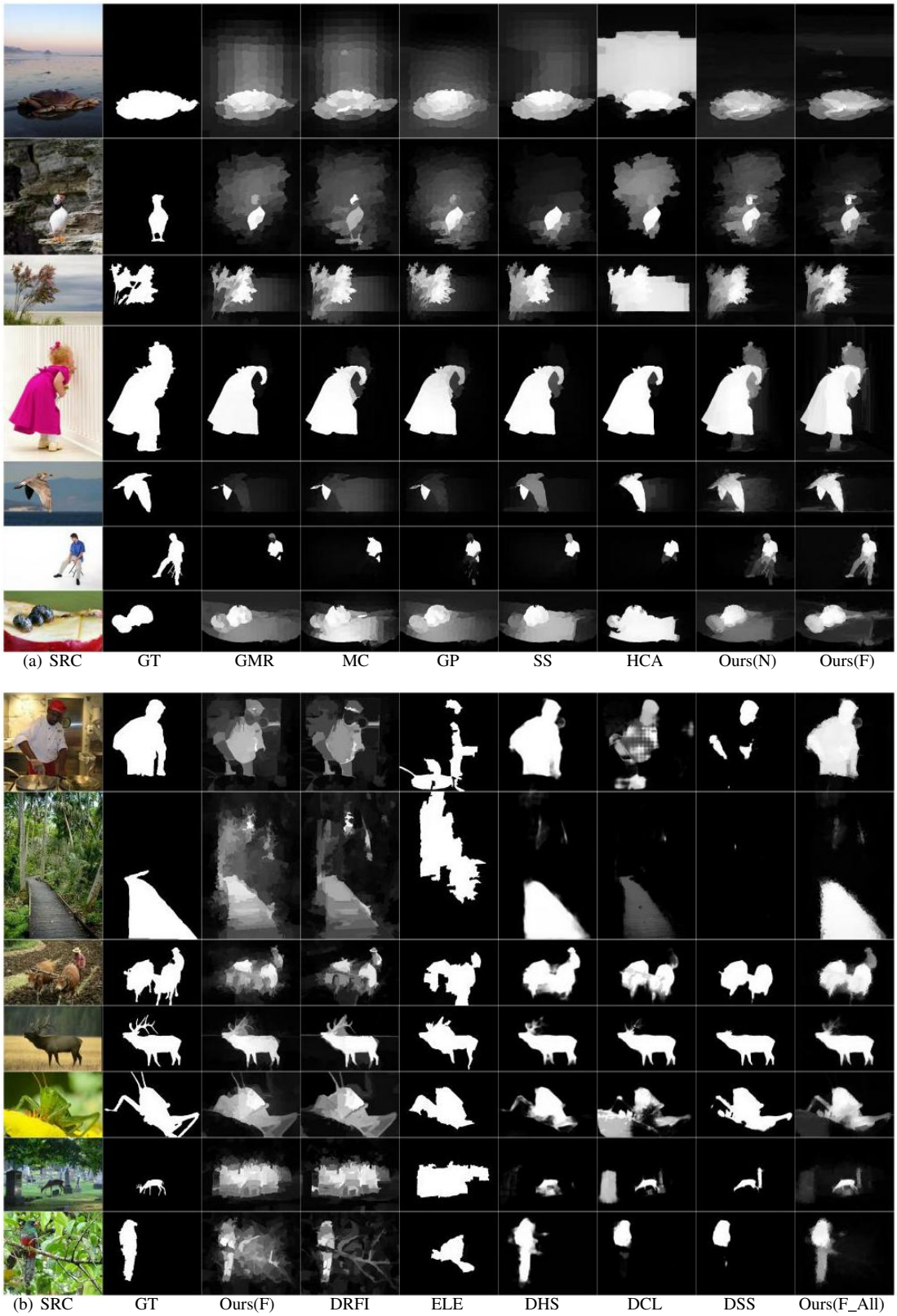

Fig. 3. Visual comparison of our methods with (a) unsupervised methods, and (b) supervised methods. In each subfigure, the source (SRC) and groudtruth (GT) images are shown in the first two columns and the results of various methods in the following.

TABLE II
PERFORMANCE STATISTICS OF COMPARED UNSUPERVISED ALGORITHMS, OURS(N) AND OURS(F) ON THE FOUR PROTOCOLS AND THE FOUR DATASETS. FOR EACH DATASET AND PROTOCOL, THE TOP THREE RESULTS ARE HIGHLIGHTED IN RED, BLUE AND GREEN, RESPECTIVELY. THE ↑/↓ SIGN INDICATES THAT THE VALUE IS POSITIVELY/NEGATIVELY RELATED WITH THE PERFORMANCE.

Dataset	Protocol	GMR	MC	GP	SS	HCA	Our(N)	Ours(F)
MSRA10K	F-measure↑	0.84357	0.844	0.85328	0.86775	0.87183	0.8589	0.89484
	MSE↓	0.072376	0.069237	0.065566	0.057949	0.065261	0.065839	0.048936
	Overlap↑	0.6348	0.64281	0.65582	0.71002	0.71086	0.65321	0.75329
	AUC↑	0.94361	0.95048	0.95909	0.96387	0.9608	0.96675	0.98274
MSRA-B	F-measure↑	0.82389	0.82608	0.82867	0.83402	0.85367	0.84255	0.86699
	MSE↓	0.069554	0.066154	0.066108	0.059501	0.063485	0.064036	0.05264
	Overlap↑	0.63434	0.64612	0.64922	0.69523	0.70141	0.65807	0.71653
	AUC↑	0.94082	0.95043	0.95269	0.95922	0.96175	0.96149	0.97792
ECSSD	F-measure↑	0.74025	0.74216	0.74333	0.75451	0.81434	0.75353	0.80222
	MSE↓	0.10886	0.10233	0.10901	0.098271	0.095293	0.10613	0.08358
	Overlap↑	0.49214	0.50212	0.49355	0.55143	0.60769	0.4937	0.58673
	AUC↑	0.89477	0.91713	0.9088	0.92332	0.94391	0.92334	0.95365
DUT-OMRON	F-measure↑	0.57038	0.56903	0.5635	0.57222	0.59478	0.58126	0.59707
	MSE↓	0.1039	0.089958	0.10262	0.10153	0.12645	0.091031	0.070968
	Overlap↑	0.42166	0.43299	0.42303	0.44397	0.45092	0.44107	0.49561
	AUC↑	0.85499	0.88684	0.8705	0.87806	0.88541	0.88129	0.93619

TABLE III
PERFORMANCE STATISTICS OF SUPERVISED ALGORITHMS, OURS(F) AND OURS(F_ALL) ON THE FOUR PROTOCOLS AND THE THREE DATASETS. FOR EACH DATASET AND PROTOCOL, THE TOP THREE RESULTS ARE HIGHLIGHTED IN RED, BLUE AND GREEN, RESPECTIVELY. THE ↑/↓ SIGN INDICATES THAT THE VALUE IS POSITIVELY/NEGATIVELY RELATED WITH THE PERFORMANCE.

Dataset	Protocol	Ours(F)	DRFI	ELE	DHS	DCL	DSS	Ours(F_All)
MSRA-B	F-measure↑	0.86699	0.87852	0.85291	0.91998	0.90587	0.92693	0.92318
	MSE↓	0.05264	0.051071	0.069124	0.02529	0.033359	0.032193	0.022262
	Overlap↑	0.71653	0.74278	0.71882	0.86798	0.82869	0.85659	0.87347
	AUC↑	0.97792	0.97758	0.89152	0.9849	0.98327	0.96523	0.99346
ECSSD	F-measure↑	0.80168	0.78558	0.78534	0.90041	0.89637	0.92058	0.921
	MSE↓	0.083019	0.084882	0.11995	0.043085	0.0485	0.048624	0.035771
	Overlap↑	0.58881	0.57828	0.59763	0.80212	0.77935	0.80886	0.82341
	AUC↑	0.9531	0.94521	0.82913	0.97327	0.97097	0.9419	0.98778
DUT-OMRON	F-measure↑	0.59331	0.57992	0.61206	0.88482	0.73605	0.78027	0.87846
	MSE↓	0.071134	0.072474	0.12136	0.017641	0.059478	0.060321	0.018004
	Overlap↑	0.49255	0.48024	0.4667	0.82422	0.59005	0.62946	0.81734
	AUC↑	0.93566	0.93353	0.80145	0.9843	0.93492	0.87841	0.99011

because most of them do not provide results on this dataset). Therefore, when comparing with them, we test on each whole dataset for a fair evaluation on a common ground. For all the other unsupervised methods, including GMR [22], MC [23], GP [38], SS [44] and HCA [49], we evaluate their performance on test sets defined by ourselves on the MSRA10K, ECSSD and DUT-OMRON datasets. Since MSRA-B dataset has an official training set and test set partitioning, we use its test set for all the compared methods.

For the compared supervised methods, we plot their PR curves in Fig.s 5(a), 5(b) and 5(c), while the PR curves of the compared unsupervised methods are shown in Fig.s 4(a), 4(b), 4(c) and 4(d). We plot the PR curves of our methods, *i.e.*, Ours(N), Ours(F), and Ours(F_All), in Fig. 4 and Fig. 5 as well for the purpose of comparison. Further, we provide the performance statistics on the four prevalent protocols for most of the methods on the benchmark datasets in Tab. II and Tab. III for performance comparison with unsupervised and supervised methods, respectively. Note that there are other works related to ours, such as BDS [43], CRPSD [46], AM [50] and IMC [51]. We can not compare with them as no code or relevant results are publicly released.

From Fig. 4, Fig. 5, Tab. II and Tab. III, we clearly observe that: 1) Ours(N) outperforms the common diffusion-based methods, 2) after integrating the saliency feature of DRFI [26], Ours(F) yields the top performance compared with most non-deep-learning based methods and even outperforms the diffusion-based method HCA [49] which uses deep features, and 3) after further integrating deep features of DHS [47], DCL [42] and DSS [52] as saliency features, Ours(F_All) yields the top performance, even when compared with the deep learning based methods (*i.e.*, DHS, DCL and DSS). All these observations confirm that the proposed super diffusion framework is capable of systematically integrating various diffusion maps or saliency features and optimizing for the best performance.

For visual comparison, we show in Fig. 3 the salient object detection results by the benchmark methods and our methods. We compare our methods with unsupervised benchmarks in Fig. 3(a) and supervised benchmarks in Fig. 3(b), respectively. From Fig. 3(a), we observe clearly that Ours(N) produces much closer results to the ground truth than the common diffusion based methods, and Ours(F) promotes the performance further with more saliency features integrated. From

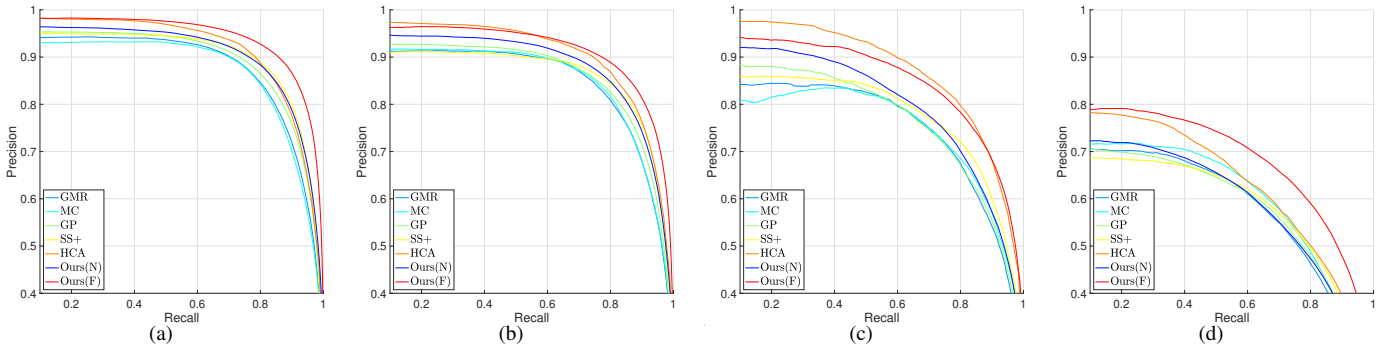

Fig. 4. PR curves for compared unsupervised algorithms, Ours(N) and Ours(F) on (a) the MSRA10K dataset, (b) the MSRA-B dataset, (c) the ECSSD dataset and (d) the DUT-OMRON dataset.

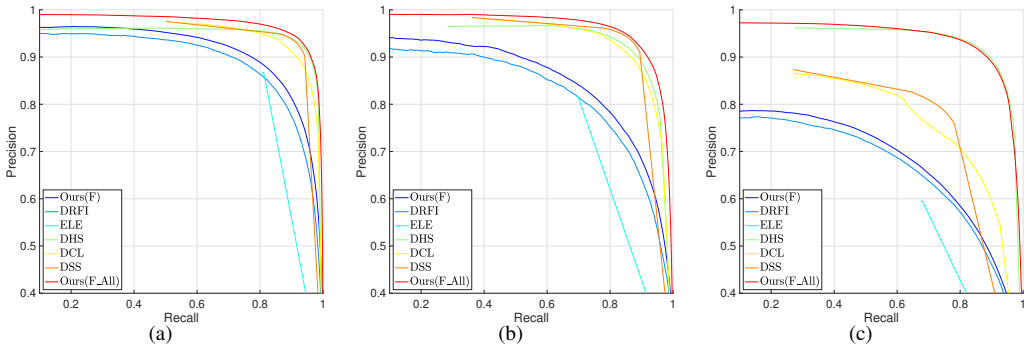

Fig. 5. PR curves for compared supervised algorithms, Ours(F) and Ours(F_All) on (a) the MSRA-B dataset, (b) the ECSSD dataset and (c) the DUT-OMRON dataset.

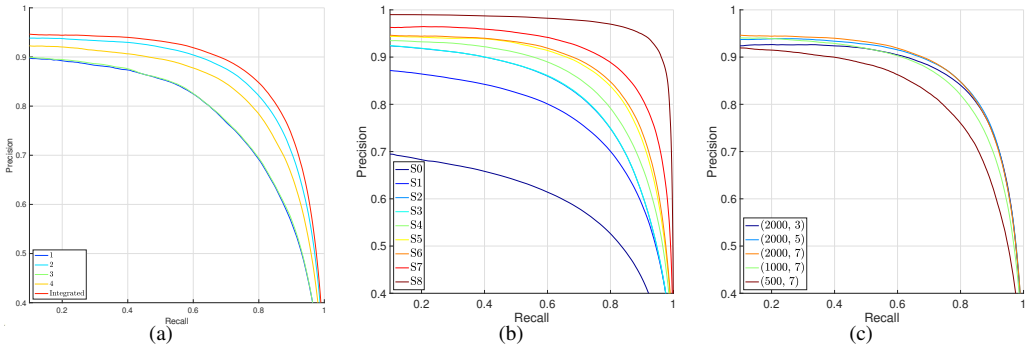

Fig. 6. (a) PR curves of several separate saliency vectors $\hat{\mathbf{y}}^{i,j}$ and the integrated saliency vector $\hat{\mathbf{y}}_I$ on the MSRA-B test dataset, (b) PR curves for building steps of our scheme on the MSRA-B test dataset, (c) PR curves for integrated saliency vector $\hat{\mathbf{y}}_I$ under different graph construction configurations on the MSRA-B test dataset.

Fig. 3(b), we observe that, with deep features further introduced as saliency features, Ours(F_All) achieves comparable performance with DHS and better performance than all the rest. Note that the images used in Fig. 3(b) are among the most challenging ones from the datasets. On these images, we see that none of the methods works perfectly while Ours(F_All) ranks among the top.

F. Effects of Building Steps

In this section, we demonstrate the incremental effects of building steps in the proposed global and local optimization operations (*c.f.* Sec. IV-A, Sec. IV-B, Sec. IV-C and Sec. IV-E), as detailed below.

In order to demonstrate the effectiveness of the optimal integration of saliency vectors, we plot in Fig. 6(a) the PR curves of separate saliency vectors, $\hat{\mathbf{y}}^{i,j}$, and the integrated saliency vector, $\hat{\mathbf{y}}$, as appear in Eq. 13. Because the combinations of different diffusion matrices and seeds produce a large number of saliency vectors, we only plot the PR curves for several selected saliency vectors with a wide range of weights. The experiment is carried out on the MSRA-B test dataset. From the PR curves in Fig. 6(a), we observe large differences among the separate saliency vectors' performance, which confirms the sensitiveness issue of diffusion matrices and seeds. Further, we observe that the PR curve of the integrated saliency vector goes above all the others, demonstrating the effectiveness of the proposed learning-based optimal integration.

Further, we give an ablation study of all the building steps of the proposed super diffusion scheme in Fig. 6(b). For each test image, we may obtain nine PR curves, $S0$ to $S8$. We

start from its $S0$ and progressively obtain $S1$ to $S8$ when the constant eigenvector is discarded, the eigenvectors after the eigengap are filtered out, the discriminability weighting is conducted, diffusion maps derived from multiple color spaces are integrated, diffusion maps derived from multiple scales are integrated, multiple diffusion seeds are integrated, saliency features are imported as diffusion maps and deep features of DHS [47], DCL [42] and DSS [52] are integrated, respectively. Experimenting on the whole MSRA-B test dataset [15], [33], we obtain the average PR curves for $S0$ to $S8$, as plotted in Fig. 6(b). From Fig. 6(b), we observe that all the local and global optimization operations consistently improve the performance, and the introduction of deep features leads to the top performance.

Finally, in Fig. 6(c), we provide the performance comparison of the integrated saliency vector $\hat{\mathbf{y}}_I$ under different graph construction configurations. For each curve, its legend tells the number of superpixels and the maximum topological distance of node connections. For example, $(2000, 7)$ means a graph of 2000 nodes, each being connected to its neighbors up to 7 hops away. From Fig. 6(c), we observe that the accuracy improves with the increase of node number and connection distance. However, the accuracy increase is diminishing. Considering the rapid growth of computing complexity with the increase of node number and connection distance, we use $(2000, 7)$ and do not go beyond for all the previous experiments in this work.

VI. Conclusions

In this work, we have proposed a super diffusion framework that systematically integrates various diffusion matrices, saliency features and seed vectors into a generalized diffusion system for salient object detection. To the best of our knowledge, this is the first framework of this kind ever published. The whole framework is theoretically based on our novel re-interpretation of the working mechanism of diffusion-based salient object detection, *i.e.*, diffusion maps are core functional elements and the diffusion process is closely related to spectral clustering in general. It takes a learning-based approach and provides a closed-form best solution to the global weighting for the integration. At the local level, it refines each diffusion matrix by getting rid of less discriminating eigenvectors, normalizes each specific saliency vector, and even incorporates discriminative saliency features as diffusion maps. As a result, the proposed framework produces a highly robust salient object detection scheme, yielding the state-of-the-art performance.

The proposed super diffusion framework is open and extensible. Besides those employed in this work, it may integrate any other diffusion matrices, saliency features, deep features and/or seed vectors as well into the system specifically trained for any application with specific criterion in saliency object detection.

It should be noted that, though outstanding performance has been reported by existing salient object detection methods (including the proposed one), they mostly experiment on benchmark datasets of narrowly focused images. As real-world images are often produced with wider fields of view, it is important to have salient object detection methods work on those images. This is worth our investigation in the future.

Appendix A

In this appendix, we give the proof of Eq. 5 and clarify the working mechanism of the second kind of seed vectors proposed in Sec. IV-C.

A. Proof of Eq. 5

The approach in the reference [23] duplicates the superpixels around the image borders as virtual background absorbing nodes, and sets the inner nodes as transient nodes, thus constructing an Absorbing Markov Chain. It computes the absorbed time for each node as its saliency value. In Eq.s 1 and 8 of the paper [23], it formulates the transition matrix as

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{25}$$

where the first m nodes are transient nodes and the last $N - m$ nodes are absorbing nodes, $\mathbf{Q} \in [0, 1]^{m \times m}$ contains the transition probabilities between any pair of transient nodes, while $\mathbf{R} \in [0, 1]^{m \times (N-m)}$ contains the probabilities of moving from any transient node to any absorbing node. 0 is the $(N - m) \times m$ zero matrix and $\mathbf{I}$ is the $(N - m) \times (N - m)$ identity matrix. According to Eq. 2 of the paper [23], the absorbed time for m transient nodes is

$$\mathbf{y}^* = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{c}, \tag{26}$$

where $\mathbf{c}$ is a m dimensional column vector all of whose elements are 1.

In our derivation, we extend Eq. 26 to

$$\mathbf{y}^* = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{c} = (\mathbf{Q}^0 + \mathbf{Q}^1 + \mathbf{Q}^2 + \ldots)\mathbf{c}, \tag{27}$$

and compute the n-th power of $\mathbf{P}$ as

$$\mathbf{P}^n = \begin{pmatrix} \mathbf{Q}^n & (\mathbf{Q}^0 + \mathbf{Q}^1 + \ldots + \mathbf{Q}^{n-1})\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}. \tag{28}$$

As the absorbed time for absorbing nodes is 0, we define the absorbed time for all the nodes as $\mathbf{y} = \begin{pmatrix} \mathbf{y}^* \\ \mathbf{0} \end{pmatrix}$. From Eq.s 27, 28 and 25, we have

$$\begin{aligned} \mathbf{y} &= (\mathbf{P}^0 + \mathbf{P}^1 + \mathbf{P}^2 + \ldots)\mathbf{x} = (\mathbf{1} - \mathbf{P})^{-1}\mathbf{x} \\ &= (\mathbf{D}^{-1}(\mathbf{D} - \mathbf{W}))^{-1}\mathbf{x} = \mathbf{L}_{rw}^{-1}\mathbf{x}, \end{aligned} \tag{29}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix}$. This completes the proof of Eq. 5.

Further, based on our re-interpolation of the diffusion (*ref.* Sec. III),

$$\begin{aligned} \mathbf{y}_i &= \sum_{j=1}^{N} \mathbf{x}_j \left\langle \mathbf{\Psi}_{\mathbf{L_{rw\,i}}}, \mathbf{\Psi}_{\mathbf{L_{rw\,j}}} \right\rangle \\ &= \sum_{j=1}^{m} \left\langle \mathbf{\Psi}_{\mathbf{L_{rw\,i}}}, \mathbf{\Psi}_{\mathbf{L_{rw\,j}}} \right\rangle, \end{aligned} \tag{30}$$

meaning that the absorbed time of each node is equal to the sum of the inner products of its diffusion map with those of all the m non-border nodes on the Absorbing Markov Chain.

B. The Second Kind of Seed Vectors

In effect, after connecting all the nodes at the four borders of the image, we have constructed a graph similar to the Absorbing Markov Chain. For every node at the border, it connects with all bn border nodes (including itself) and only bm non-border nodes ($bm \ll bn$), meaning that once a random walk reaches a border node, it will less likely escape from the border node set. Therefore, we may assume that all the non-border nodes are transient nodes and all the border nodes are background absorbing nodes.

Accordingly, we compute the absorbed time of all nodes in an image to form a seed vector of the second kind, That is, following Eq. 30, we have seed $\mathbf{s}^i$

$$\mathbf{s}_j^i = \sum_{k=1}^{d} \left\langle \bar{\mathbf{\Psi}}_j^i, \bar{\mathbf{\Psi}}_k^i \right\rangle, \tag{31}$$

or, equivalently,

$$\mathbf{s}^i = \bar{\mathbf{A}}^{i^{-1}} \mathbf{z}, \tag{32}$$

where $\mathbf{z}_k = 1$ if v_k is a non-border node and $\mathbf{z}_k = 0$ otherwise.

REFERENCES

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE PAMI*, 1998.

[2] J. Harel, C. Koch, and P. Perona., "Graph-based visual saliency," *NIPS*, 2006.

[3] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," *ICCV*, 2009.

[4] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE PAMI*, 2012.

[5] D. Gao and N. Vasconcelos, "Decision-theoretic saliency: Computational principles, biological plausibility, and implications for neurophysiology and psychophysics," *Neural Computation*, 2009.

[6] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," *CVPR*, 2012.

[7] N. D. B. Bruce and J. K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of Vision*, 2009.

[8] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," *CVPR*, 2007.

[9] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of Vision*, 2008.

[10] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision*, 2009.

[11] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," *CVPR*, 2011.

[12] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *Journal of Vision*, 2013.

[13] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2018.

[14] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," *CVPR*, 2009.

[15] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu, "Global contrast based salient region detection," *IEEE PAMI*, 2015.

[16] K. Y. Chang, T. L. Liu, H. T. Chen, and S. H. Lai, "Fusing generic objectness and visual saliency for salient object detection," *ICCV*, 2011.

[17] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum, "Learning to detect a salient object," *IEEE PAMI*, 2011.

[18] Y. Lu, W. Zhang, H. Lu, and X. Y. Xue, "Salient object detection using concavity context," *ICCV*, 2011.

[19] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," *CVPR*, 2012.

[20] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE PAMI*, 2012.

[21] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," *CVPR*, 2012.

[22] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," *CVPR*, 2013.

[23] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing markov chain," *ICCV*, 2013.

[24] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," *CVPR*, 2013.

[25] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by ufo: Uniqueness, focusness and objectness," *ICCV*, 2013.

[26] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 251–268, 2017.

[27] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," *CVPR*, 2013.

[28] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," *CVPR*, 2014.

[29] R. Liu, J. Cao, Z. Lin, and S. Shan, "Adaptive partial differential equation learning for visual saliency detection," *CVPR*, 2014.

[30] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform," *CVPR*, 2014.

[31] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," *CVPR*, 2014.

[32] Z. Ren, Y. Hu, L.-T. Chia, and D. Rajan, "Improved saliency detection based on superpixel clustering and saliency propagation," *ACM Multimedia*, 2010.

[33] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," *ICCV*, 2013.

[34] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," *ICCV*, 2013.

[35] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," *ICCV*, 2013.

[36] Q. Wang, W. Zheng, and R. Piramuthu, "Grab: Visual saliency via novel graph model and background priors," *CVPR*, 2016.

[37] A. Aytekin, E. C. Ozan, S. Kiranyaz, and M. G. Tampere, "Visual saliency by extended quantum cuts," *ICIP*, 2015.

[38] P. Jiang, N. Vasconcelos, and J. Peng, "Generic promotion of diffusion-based salient object detection," *ICCV*, 2015.

[39] C. Gong, D. Tao, W. Liu, S. J. Maybank, M. Fang, K. Fu, and J. Yang, "Saliency propagation from simple to difficult," *CVPR*, 2015.

[40] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. D. Feng, "Robust saliency detection via regularized random walks ranking," *CVPR*, 2015.

[41] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," *ICCV*, 2017.

[42] G. Li and Y. Yu, "Deep contrast learning for salient object detection," *CVPR*, 2016.

[43] Z. Wang, D. Xiang, S. Hou, and F. Wu, "Background-driven salient object detection," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 750–762, 2017.

[44] K. Fu, I. Y.-H. Gu, and J. Yang, "Spectral salient object detection," *Neurocomputing*, vol. 275, pp. 788 – 803, 2018.

[45] Z. Liu, J. Li, L. Ye, G. Sun, and L. Shen, "Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2527–2542, 2017.

[46] Y. Tang and X. Wu, "Saliency detection via combining region-level and pixel-level predictions with cnns," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016.

[47] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, "What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] Y. Qin, M. Feng, H. Lu, and G. W. Cottrell, "Hierarchical cellular automata for visual saliency," *International Journal of Computer Vision*, vol. 126, no. 7, pp. 751–770, Jul 2018.

[50] Y. Xu, X. Hong, F. Porikli, X. Liu, J. Chen, and G. Zhao, "Saliency integration: An arbitrator model," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 98–113, 2019.

[51] J. Zhang, Y. Dai, and F. Porikli, "Deep salient object detection by integrating multi-level cues," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 1–10.

[52] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[53] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE PAMI*, 2012.
[54] M. Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," *CVPR*, 2011.
[55] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, 2006.
[56] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *NIPS*, 2002.
[57] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, 2007.
[58] B. Nadler and M. Galun, "Fundamental limitations of spectral clustering," *NIPS*, 2006.
[59] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, 2007.
[60] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE TIP*, 2015.

Peng Jiang received the B.S. and Ph.D. degrees in computer science and technology from Shandong University, China, in 2010 and 2016, respectively. Currently, He is a research assistant with Shandong University, China. His research spans various areas, including computer vision, image processing, machine learning and related interdisciplinary fields.

Zhiyi Pan received the BS degree from Shandong University, China, in 2018. He is currently pursuing the MS degree in computer science and technology with Shandong University, China. His research interests include computer vision and machine learning.

Changhe Tu received the BSc, MEng, and PhD degrees from Shandong University, China, in 1990, 1993, and 2003, respectively. His research interests are in the areas of computer graphics and visualization. He is a Professor in the School of Computer Science and Technology, Shandong University, China.

Nuno Vasconcelos received his PhD from the Massachusetts Institute of Technology in 2000. From 2000 to 2002, he was a member of the research staff at the Compaq Cambridge Research Laboratory. In 2003, he joined the Department of Electrical and Computer Engineering at the University of California, San Diego, where he is the head of the Statistical Visual Computing Laboratory. His work spans various areas, including computer vision, machine learning, signal processing, and multimedia systems.

Baoquan Chen is a Chair Professor of Peking University, where he is the Executive Director of the Center on Frontiers of Computing Studies. Prior to the current post, he was the Dean of School of Computer Science and Technology at Shandong University. His research interests generally lie in computer graphics, visualization, and human-computer interaction. Chen received an MS in Electronic Engineering from Tsinghua University, Beijing (1994), and a second MS (1997) and then PhD (1999) in Computer Science from the State University of New York at Stony Brook. Chen served as conference co-chair of IEEE Visualization 2005, and as the conference chair of SIGGRAPH Asia 2014.

Jingliang Peng received the PhD degree in electrical engineering from the University of Southern California in 2006, the BS and MS degrees in computer science from Peking University in 1997 and 2000, respectively. Currently, he is a professor at the School of Software, Shandong University, China. His research interest mainly resides in digital geometry processing and digital image/video analysis.