Learning from Pixel-Level Noisy Label: A New Perspective for Light Field Saliency Detection (Supplementary)

We provide more implementation detail and experimental results in this supplementary material. In detail, we provide more ConvLSTM module details in Section A, we present more ablation studies on DUT-LF [10] dataset to thoroughly analyze our proposed modules in Section B, we report elicitation details of cross-scene noise penalty loss in Section C, and we present additional qualitative and quantitative comparisons in Section D.

A. More Architecture Details

To further illustrate the proposed network, we show more details of the ConvLSTM module [12]. The features extracted from focal slices and all-focus central view image in m-th layers (denoted as $g_m$) are fed into a ConvLSTM structure in our architecture to gradually refine the abundant information for accurately identifying the salient objects. The procedure is defined as:

$$
i_t = \sigma(w_{xi} * g_m + w_{hi} * H_{t-1} + w_{ci} \otimes C_{t-1} + b_i)$$

$$f_t = \sigma(w_{xf} * g_m + w_{hf} * H_{t-1} + w_{cf} \otimes C_{t-1} + b_f)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tanh(w_{xc} * g_m + w_{hc} * H_{t-1} + b_c)$$

$$o_t = \sigma(w_{xo} * g_m + w_{ho} * H_{t-1} + w_{co} \otimes C_t + b_o)$$

$$H_t = o_t \otimes \tanh(C_t)$$

(1)

where $\otimes$ denotes pixel-wise multiplication and $\sigma(\cdot)$ is soft-max function. Memory cell $C_t$ stores previous information. All $*$, $w_*$ and $b_*$ represent convolution operator and convolution parameters to be learned. The memory cell $C_t$, the gates $i_t$, $f_t$, $o_t$ and hidden state $H_t$ are 3D tensors.

As shown in Fig.2 of our main paper, the weighted focal slices features for $m$ layers are regarded as a sequence of inputs corresponding to consecutive time steps, feeding into ConvLSTM modules to gradually refine their spatial information (Fig. 1(a)). Then, the updated features $F'_m$ and $R'_m$ are further input to ConvLSTM modules to summarize information (Fig. 1(b) and Fig. 1(c)).

B. Ablation Study

To explore the optimal hyperparameters in the proposed cross-scene noise penalty loss and evaluate influence of different supervision information, we conduct additional ablation studies reported as following.

B.1. Hyperparameters of loss function

In this part, we present parameters details of cross-scene noise penalty loss $L_t$. Based on $m_i$ pairs of cross-scene samples, $L_t$ for pixel $(u, v)$ is defined as:

$$L_t(s^{(u,v)}_t, \hat{y}^{(u,v)}_t) = L(s^{(u,v)}_t, \hat{y}^{(u,v)}_t)$$

$$- \frac{\alpha}{m_i} \sum_{n,n'=2}^{m_i} l(s^{(u,v)}_n, \hat{y}^{(u,v)}_{n'})$$

(2)

The second term is linearly combined with two hyperparameters $\alpha$ and $m_i$. For $\alpha$, we define a dynamic hyperparameter recursive process as $\alpha_{t+1} = \alpha_t + c/m_i$, where $c$ is the maximal value of $\alpha$ and $m_i$ denotes the maximal number of training iterations for each sample ($m_i = 30$ in our experiments). We report the affect of various value of $c$ on saliency detection performance in Table 1. It can be seen that performance achieves optimal when $c = 0.30$.

<table>
<thead>
<tr>
<th>$c$</th>
<th>0</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
<th>0.30</th>
<th>0.35</th>
<th>0.40</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$</td>
<td>0.751</td>
<td>0.766</td>
<td>0.789</td>
<td>0.794</td>
<td>0.811</td>
<td><strong>0.813</strong></td>
<td>0.802</td>
<td>0.793</td>
</tr>
<tr>
<td>$M_m$</td>
<td>0.168</td>
<td>0.144</td>
<td>0.112</td>
<td>0.114</td>
<td>0.102</td>
<td><strong>0.091</strong></td>
<td>0.052</td>
<td>0.026</td>
</tr>
</tbody>
</table>

Table 1: Experimental results of our model trained with different settings of the hyperparameter $c$ (the maximal value of $\alpha$) in Eq.(1) on saliency detection performance while keeping other settings unchanged.

In addition, we conduct experiments on our framework keeping other settings unchanged with different $m_i$ values from 2 to 5. As shown in Table 2, we can see that more samples can consistently boost the performance. However, the increasing number of samples will consume a lot of computing resources. We set $m_i = 4$ to achieve the balance of performance and training time in our experiment.
Table 2: Model performances with regard to different number of correlation samples in Eq.(1) while keeping other settings unchanged.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Metrics</th>
<th>DUT-LF</th>
<th>HUFT</th>
<th>LFSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaselineDSR</td>
<td>$F^\uparrow$</td>
<td>0.641</td>
<td>0.485</td>
<td>0.594</td>
</tr>
<tr>
<td></td>
<td>$M^\downarrow$</td>
<td>0.102</td>
<td>0.121</td>
<td>0.094</td>
</tr>
<tr>
<td>DSR</td>
<td>$F^\uparrow$</td>
<td>0.812</td>
<td>0.640</td>
<td>0.812</td>
</tr>
<tr>
<td></td>
<td>$M^\downarrow$</td>
<td>0.104</td>
<td>0.084</td>
<td>0.066</td>
</tr>
<tr>
<td>BaselineGT</td>
<td>$F^\uparrow$</td>
<td>0.842</td>
<td>0.762</td>
<td>0.835</td>
</tr>
<tr>
<td></td>
<td>$M^\downarrow$</td>
<td>0.012</td>
<td>0.084</td>
<td>0.066</td>
</tr>
<tr>
<td>GT</td>
<td>$F^\uparrow$</td>
<td>0.851</td>
<td>0.760</td>
<td>0.844</td>
</tr>
<tr>
<td></td>
<td>$M^\downarrow$</td>
<td>0.007</td>
<td>0.010</td>
<td>0.047</td>
</tr>
</tbody>
</table>

Table 3: Results on the generalization capability of our proposed method.

<table>
<thead>
<tr>
<th>$m_i$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F^\uparrow$</td>
<td>0.797</td>
<td>0.801</td>
<td>0.813</td>
<td>0.815</td>
</tr>
<tr>
<td>$M^\downarrow$</td>
<td>0.096</td>
<td>0.090</td>
<td>0.091</td>
<td>0.090</td>
</tr>
</tbody>
</table>

B.2. Learn from different noisy label generator

In the main paper, we use conventional unsupervised method RBD [20] to generate noisy labels. We further conduct an experiment with noisy labels generated by conventional method DSR [4]. We build a concise baseline that only contains separate features extraction branches for focal slices and all-focus central view image in the main paper. In this section, we treat the saliency maps generated by DSR method as supervision, and train the model using the setting of baseline. The results are reported in Table 3, we can find a huge gap between DSR and BaselineDSR, indicating the generalization capability of our method. Our method can handle noisy pixels even the noise generation mechanism is different. It is reasonable since that we correlate noisy pixels across whole dataset. In our experiment, we control the actual portion of noisy pixels by using different noisy labelling, instead adding noise to clean data.

B.3. Learn from ground truth labels

The ground truth can be treated as a special case of noisy label. We directly conduct experiments on ground truth using the setting of baseline. Then, we train our proposed model using ground truth (denoted as GT in Table 3). The performance improvement indicates the necessary of noise handling even training model on the ground truth.

C. Elicitation details of loss function

We define pixel $(u, v)$ in noisy labels and ground truth as $\hat{Y}^{(u,v)}$ and $Y^{(u,v)}$, then the error rate can be denoted as:

$e_{+1} = p(\hat{Y}^{(u,v)} = -1 \mid Y^{(u,v)} = +1)$

$e_{-1} = p(\hat{Y}^{(u,v)} = +1 \mid Y^{(u,v)} = -1)$

(3)

Similar to $e_{+1}$ and $e_{-1}$, we define the error rates for noisy labels and the initial noisy saliency maps generated by our method:

$p(\hat{y}^{(u,v)} = -1 \mid y^{(u,v)} = +1) = e_{+1}$

$p(\hat{y}^{(u,v)} = +1 \mid y^{(u,v)} = -1) = e_{-1}$

$p(s^{(u,v)} = -1 \mid y^{(u,v)} = +1) = e_{+1}^s$

$p(s^{(u,v)} = +1 \mid y^{(u,v)} = -1) = e_{-1}^s$

(4)

(5)

where $\hat{y}^{(u,v)}$ and $y^{(u,v)}$ represent pixel $(u, v)$ in noisy label and ground truth respectively.

Consider a binary segmentation case (salient object and background): $p(y^{(u,v)} = -1) = 0.4$, $p(y^{(u,v)} = +1) = 0.6$, the noise in the labels are $e_{-1} = 0.3$, $e_{+1} = 0.4$ and $e_{-1}^s = 0.2$, $e_{+1}^s = 0.3$.

Firstly, we compute the marginals of $s^{(u,v)}$ and $\hat{y}^{(u,v)}$:

$p(s^{(u,v)} = +1) = 1 - p(s^{(u,v)} = -1) = 0.5$  \hspace{0.5cm} (7)

for noisy labels:

$p(\hat{y}^{(u,v)} = -1)$

$= p(s^{(u,v)} = -1 \mid y^{(u,v)} = -1)p(y^{(u,v)} = -1)$

$p(\hat{y}^{(u,v)} = +1 \mid y^{(u,v)} = +1)$

$= (1 - e_{+1}^s) \cdot 0.4 + e_{+1}^s \cdot 0.6 = 0.52$  \hspace{0.5cm} (8)

and

$p(\hat{y}^{(u,v)} = +1) = 1 - p(\hat{y}^{(u,v)} = -1) = 0.48$  \hspace{0.5cm} (9)

for the joint distribution,

$p(s^{(u,v)} = -1, y^{(u,v)} = -1)$

$= p(s^{(u,v)} = -1, y^{(u,v)} = -1 \mid y^{(u,v)} = -1)p(y^{(u,v)} = -1)$

$p(s^{(u,v)} = +1, y^{(u,v)} = -1 \mid y^{(u,v)} = +1)p(y^{(u,v)} = +1)$

$= (1 - e_{-1}^s)(1 - e_{-1}) \cdot 0.4 + e_{+1}^s \cdot e_{+1} \cdot 0.6 = 0.296$  \hspace{0.5cm} (10)

$p(s^{(u,v)} = -1, \hat{y}^{(u,v)} = +1)$

$= p(s^{(u,v)} = -1) - p(s^{(u,v)} = -1, y^{(u,v)} = -1)$

$= 0.264$

further,

$p(s^{(u,v)} = +1, \hat{y}^{(u,v)} = +1)$

$= p(s^{(u,v)} = +1) - p(s^{(u,v)} = +1, y^{(u,v)} = -1)$

$= 0.224$  \hspace{0.5cm} (11)

$p(s^{(u,v)} = +1, y^{(u,v)} = +1)$

$= p(s^{(u,v)} = +1)$

$= 0.216$
With above, the entries in $\Delta_{a,b}$ can be computed easily, for instance

$$\Delta_{1,1} = p(s_{u,v} = -1, y_{u,v} = -1) - p(s_{u,v} = -1) \cdot p(y_{u,v} = -1) = 0.296 - 0.5 \cdot 0.52 = 0.036$$

Then we have

$$\begin{bmatrix} 0.036 & -0.036 \\ -0.036 & 0.036 \end{bmatrix} \Rightarrow \text{Sgn}(\Delta) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The above implies that for $\Delta_{a,b}, \forall a,b = \{1,2\}$, the marginal correlation is positive while for off-diagonal entries, they are negatively correlated.

CA [9] [1] requires each pixel in the predicted salient map to perform multiple tasks: compute the correlation with its corresponding noisy label and exploit the correlation between predictions of other scenes and unpaired noisy labels as the penalty to current scene. Ultimately the scoring function for each task, is defined as follows:

$$S(s_{u,v}, \hat{y}_{u,v}) = \Omega(s_{u,v}, \hat{y}_{u,v}) - \Omega(s_{2u,v}, \hat{y}_{2v})$$

Next, we have the cross scene penalty loss defined as Eq.(2). The terms in cross scene penalty loss is simulating the marginal correlation probability in $\Delta$.

**D. Experiment Results**

**D.1. Quantitative Results**

We extend the quantitative studies as a supplement to the main paper. Firstly, we present results of additional 10 methods in Table 4, including 4 fully supervised RGB methods (C2S [5], DSS [3], DHS [6], UCF [17]), 4 supervised RGB-D methods (CPFP [18], DP [8], PDNet [19], CTMF [2]) and 2 conventional unsupervised methods (DSR [4], DILF [14]). Results of competing methods are generated by authorized codes or directly provided by authors. Our model consistently achieves higher scores on all datasets across two evaluation metrics. Secondly, we compare our method with the state-of-the-art methods on three benchmark datasets and the PR curves are shown in Figure 2. Compared to the state-of-the-art fully supervised RGB and RGB-D methods, our method achieves significant advantages with a relatively small training set DUT-LF. It can be seen we still achieves competitive performance when compared with a number of fully supervised light field methods.

**D.2. Qualitative Comparison**

To further prove the superior of our method, we visualize results for our method and others. As shown in Figure 3, our results have a significant improvement for challenging scenes compared with fully supervised RGB, RGB-D methods and unsupervised RGB method. We still show a competitive performance compared with fully supervised light field method. With our proposed method and the abundant cues of light field data, our model has better noise invariant

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>RGB</th>
<th>RGB-D</th>
<th>Light field</th>
<th>Conventional Model</th>
<th>Noisy label Model</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUT-LF</td>
<td>F ↓</td>
<td>0.791</td>
<td>0.723</td>
<td>0.801</td>
<td>0.868</td>
<td>0.825</td>
<td>0.813</td>
</tr>
<tr>
<td>HFUT</td>
<td>M ↓</td>
<td>0.084</td>
<td>0.128</td>
<td>0.090</td>
<td>0.107</td>
<td>0.100</td>
<td>0.091</td>
</tr>
<tr>
<td>LFSD</td>
<td>M ↓</td>
<td>0.112</td>
<td>0.138</td>
<td>0.129</td>
<td>0.144</td>
<td>0.103</td>
<td>0.108</td>
</tr>
</tbody>
</table>

Table 4: Additional quantitative comparisons between our method and competing methods on three light field datasets.

![Figure 2: The PR curves of our method and other methods on three light field datasets, including fully supervised RGB, RGB-D and light field models, conventional models and unsupervised RGB methods.](image)
capability for salient object detection learning from pixellevel noisy labels.

References


[13] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion:


