Supplementary Material for Large-scale Single-pixel Imaging and Sensing

Lintao Peng[a,], **Siyu Xie**[a,], **Hui Lu**[a], **Liheng Bian**[a,b,,c,*]
[a]MIIT Key Laboratory of Complex-field Intelligent Sensing, Beijing Institute of Technology, Beijing, China, 100081
[b]Guangdong Province Key Laboratory of Intelligent Detection in Complex Environment of Aerospace, Land and Sea, Beijing Institute of Technology, Zhuhai, China, 519088
[c]Yangtze Delta Region Academy of Beijing Institute of Technology (Jiaxing), Jiaxing, China, 314019

This is the supplementary material for *Large-scale single-pixel imaging and sensing*. In the supplementary material, we provide more experiment results for large-scale single-pixel imaging (SPI), image-free single-pixel segmentation, and image-free single-pixel object detection experiments.

1 Supplementary Material for large-scale SPI

Comparison with the conventional SPI methods at different sampling rates. We compared our SPIS with the conventional SPI methods at different sampling rates. Fig. 1 shows the visualization results. The methods involved in the comparison include ReconNet[15], DCAN,[22] GAN,[23] and the combination of ReconNet and BM3D. From Fig. 1, We can see that the reconstructed images by our method outperform the other methods in terms of both fine details and overall quality. Even at 3% sampling rate, our SPIS technique is able to reconstruct a clear image. This validates that the Transformer-based architecture can better model global features and extract high-dimensional semantic features that are effective for reconstruction.

Comparison with different sampling patterns. To validate that the reported small-size optimized pattern is more suitable for SPI tasks, we compared the imaging performance of the small-size optimized pattern with other patterns on the large-scale SPI task. Fig. 2 presents the visual comparison results. The illumination patterns involved in the comparison include Hardmard

Fig 1: Experiment comparison between SPIS and the conventional SPI methods at different sampling rates.

pattern,[22] Random pattern,[20] and full-size optimized pattern.[24] Results in Fig. 2 show that our small-size pattern maintains stable performance even at an extremely low sampling rate. The performance of Hadamard and Random patterns did not perform as well as the optimized pattern due to information loss. The optimized full-size patterns did not perform well because they can only acquire 1-dimensional (1D) measurements, and cannot retain the position information of each pixel. In summary, the optimized small-size pattern is more suitable for large-scale SPI.

Noise interference experiment. To demonstrate the robustness of SPIS to noise interference,

Fig 2: Experiment comparison between small-size optimized pattern and other patterns at different sampling rates.

we added different levels of Gaussian noise to the measurements at different sampling rates, and fed them into the SPIS encoder for SPI reconstruction. We use numpy and opencv (cv2) libraries in Python to add Gaussian noise to the measurements. First, we need to read the measurements as

an array, then calculate the corresponding noise intensity to be added based on the target signal-to-noise ratio (SNR, in dB), and finally add the Gaussian noise to the measurements. As seen in Fig. 3, the reported SPIS method still performs well with measurement noise.

Fig 3: Experiment results of noise robustness. We validate the robustness of SPIS to noise interference on the SPI task by adding different levels of Gaussian noise to the measurements at different sampling rates.

Quantitative results of different SPI methods at different sampling rates. We show the quantitative imaging results of different SPI methods at different sampling rates in Tab. 1. The methods involved in the comparison include SPIS using Hadamard, Random and optimized full-

Table 1: The quantitative large-scale single-pixel imaging results. "SR" stands for sampling rate. We marked the highest score for each column in bold.

Method	SR=3%		SR=5%		SR=7%		SR=10%		SR=15%	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Hadamard	15.82	0.34	16.61	0.41	17.19	0.44	17.74	0.49	18.27	0.53
Random	17.23	0.36	18.25	0.38	18.93	0.41	19.42	0.46	21.27	0.51
Optimized	19.16	0.61	19.76	0.63	20.02	0.63	20.89	0.66	21.53	0.69
DCAN	18.67	0.51	18.86	0.54	19.13	0.59	19.53	0.62	20.13	0.63
GAN	19.22	0.58	19.85	0.61	20.19	0.65	20.61	0.68	20.94	0.72
ReconNet	19.89	0.62	20.01	0.67	20.90	0.72	21.12	0.75	22.41	0.75
W/O Uncertainty	22.92	0.82	23.14	0.83	23.25	0.84	23.81	0.85	24.36	0.85
Step1	22.47	0.78	22.77	0.79	23.39	0.82	23.71	0.83	24.15	0.84
Step2 (Ours)	**24.13**	**0.83**	**24.64**	**0.86**	**25.33**	**0.86**	**25.69**	**0.88**	**26.17**	**0.89**

size patterns, the existing deep learning-based single-pixel imaging methods (DCAN, GAN and ReconNet), the SPIS without uncertainty-driven loss function, the SPIS only completing the first training stage, and the SPIS completing two stages of training. As shown in Tab. 1, our SPIS method achieves the highest PSNR and SSIM at different sampling rates.

Noise mitigation techniques used in real-world experiments. For the SPI task, it has been shown that normalization of the photodetector signals leads to an improvement in the signal-to-noise ratio (SNR) and the overall image reconstruction quality.[20] In this work, to improve SNR, we binarized the illuminating patterns and performed normalization to signals by maintaining an equal black/white ratio in each illumination pattern.

2 Supplementary Material for image-free segmentation

Noise interference experiment. To demonstrate the robustness of SPIS to noise interference on the image-free single-pixel segmentation task, we added different levels of noise to the measurements at different sampling rates, and fed them into the SPIS decoder for image-free segmentation. The experiment results are presented in Tab. 2 and Fig. 4. As shown in Fig. 4, even at 1% sampling

rate and 30dB noise interference, the SPIS technique was still able to segment the cells. Tab. 2 shows that even at a sampling rate of 0.1% and a noise level of 30dB, our method still achieved a Dice of 0.78 for the segmented images, validating that SPIS is robust to noise interference.

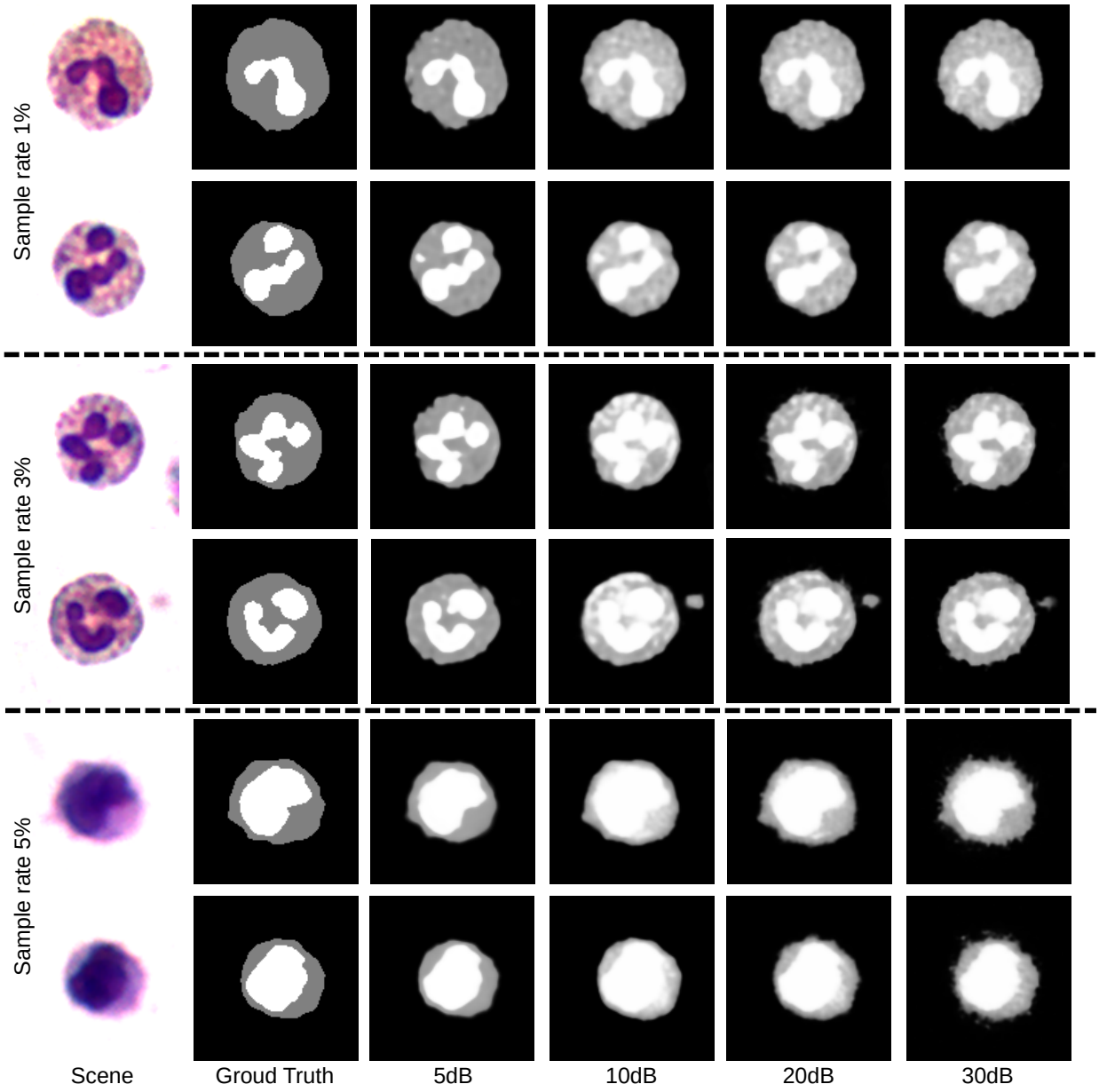


Fig 4: Noise robustness experiment of SPIS on image-free single-pixel segmentation. We validate the robustness of SPIS to noise interference on image-free single-pixel segmentation tasks by adding different levels of noise to the measurements at different sampling rates.

Comparison with different illumination patterns. We compared the segmentation perfor-

Table 2: The quantitative results of SPIS on image-free single-pixel segmentation at different sampling rates with different levels of measurement noise.

Noise level	SR=0.1%		SR=3%		SR=5%		SR=7%		SR=10%		SR=15%	
	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU
0dB	0.84	0.72	0.82	0.70	0.82	0.70	0.88	0.79	0.89	0.80	0.92	0.85
5dB	0.83	0.71	0.82	0.70	0.82	0.70	0.86	0.75	0.87	0.77	0.91	0.85
10dB	0.83	0.68	0.82	0.69	0.82	0.69	0.78	0.73	0.76	0.75	0.90	0.83
15dB	0.81	0.67	0.81	0.67	0.82	0.69	0.71	0.70	0.65	0.71	0.89	0.80
20dB	0.79	0.66	0.79	0.67	0.81	0.69	0.68	0.68	0.63	0.69	0.88	0.78
30dB	0.78	0.63	0.78	0.65	0.77	0.64	0.64	0.66	0.62	0.67	0.87	0.76

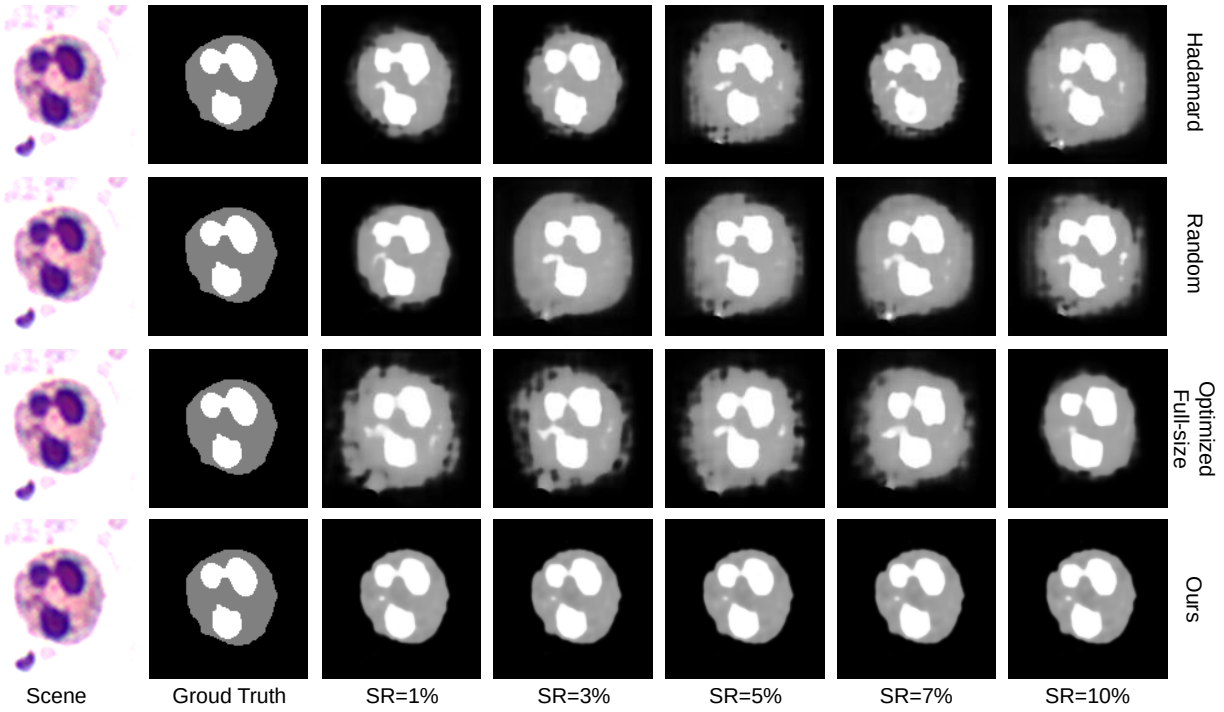


Fig 5: Comparison between small-size optimized pattern and other patterns at different sampling rates on the image-free single-pixel segmentation task.

mance of the small-size optimized pattern with other patterns on the image-free single-pixel segmentation task. The patterns methods involved in the comparison include Hardmard pattern,[22] Random pattern[20] and full-size optimized pattern.[24] Results in Fig. 5 show that the small-size optimized pattern can maintain a stable performance even at an extremely low sampling rate. The Hadamard and Random patterns did not perform as well as the optimized pattern due to information loss. The optimized full-size pattern did not perform well because it can only acquire 1D

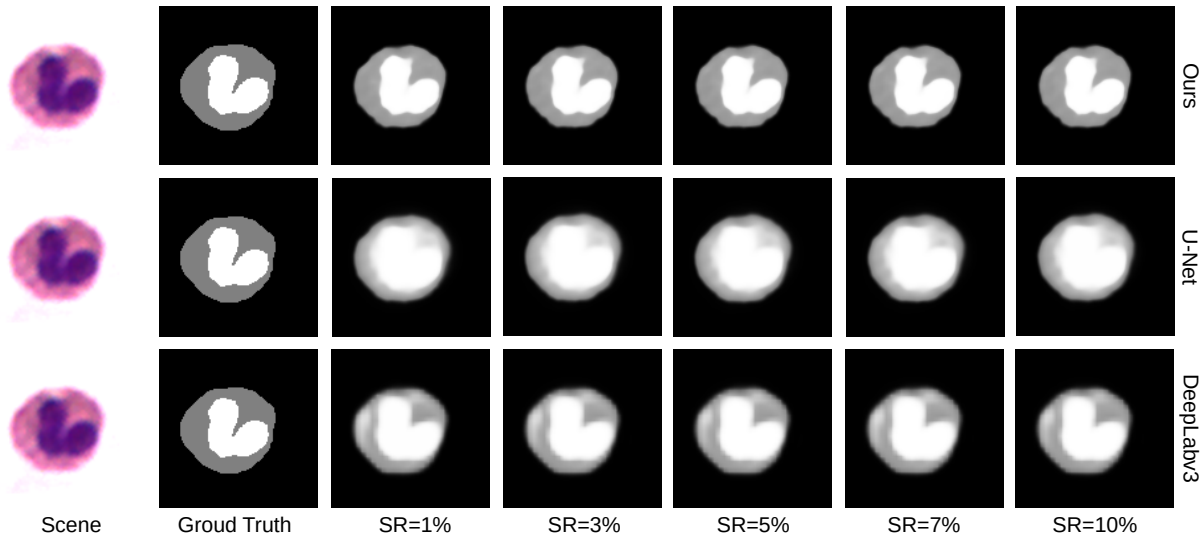


Fig 6: Comparison between SPIS and the existing deep-learning-based image-free single-pixel segmentation methods at different sampling rates.

measurements and cannot retain the position information of each pixel. In summary, the small-size optimized pattern is more suitable for the image-free single-pixel segmentation task.

Comparison with deep-learning-based image-free segmentation methods. We also compared SPIS with the existing deep-learning-based image-free segmentation methods at different sampling rates. The methods involved in the comparison include SPS[25] and DeepLabv3.[17] The SPS method uses full-size optimized patterns to sample the scene, and uses a convolutional network to preliminarily process the measurements and output intermediate results. Then, it inputs the intermediate results into a UNet to perform segmentation. We added an upsampling residual convolution layer (consistent with the upsampling residual convolutional layer structure in SPIS's image reconstruction decoder) to increase the output resolution to 256×256 pixels (originally it was 128×128), and the other settings remain the same as the original publication. The DeepLabv3 method is implemented by using the DeepLabv3 network[17] to replace Unet in SPS. From Fig. 6, We can see that the results output by our method outperform other methods in terms of both fine details and overall quality. Even at 1% sampling rate, our SPIS technique can segment the cells.

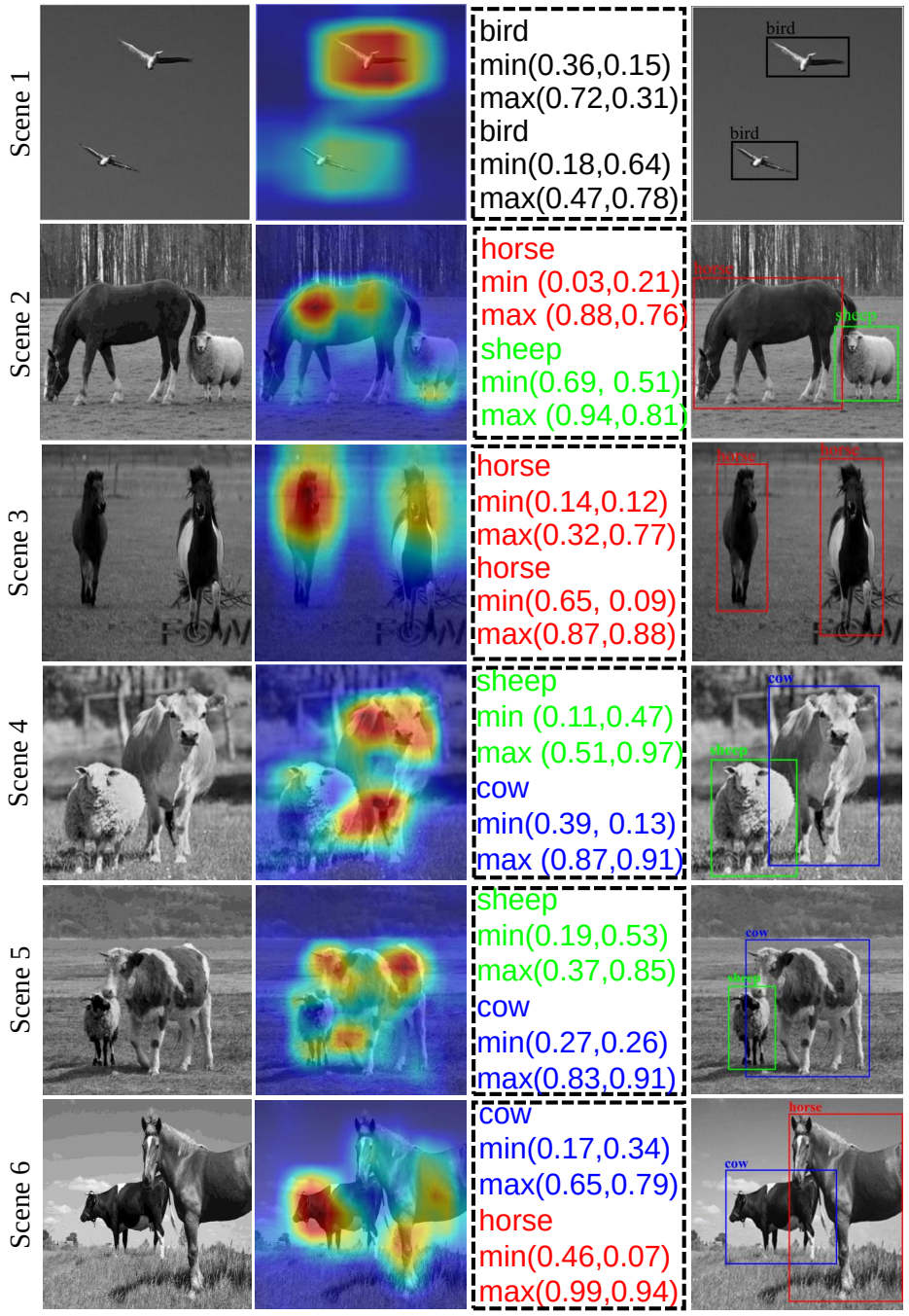


Fig 7: Visualization results of image-free single-pixel object detection. The "min" and "max" represent the relative coordinates of the upper left corner and lower right corner of the target bounding box, respectively. To better demonstrate the detection results, we visualized the output of SPIS on the input scene.

3 Supplementary Material for image-free object detection

Experiment results of image-free single-pixel object detection. Figure 7 shows the detection results corresponding to several real scenes printed on films. Among them, the attention heat map

validates that the Transformer-based SPIS can reinforce the network's attention to the targets in the scene. We can see that the SPIS technique maintains a high detection accuracy on different classes of objects, which validates the effectiveness of our proposed technique.

The derivation of the UDL loss function. In the first training stage, we used the L_1 loss function as

$$Loss_{L_1}(I_{RHQ}, I_{HQ}) = E_{I_{RHQ}, I_{HQ}}[||I_{RHQ} - I_{HQ}||]. \tag{1}$$

Among them, I_{RHQ} stands for the high-quality image reconstructed by the network, I_{HQ} stands for the ground truth.

In the second training stage, the loss function consists of the regression loss L_{reg}, confidence loss L_{con}, and classification loss L_{cls}.

$$\begin{aligned}
L_{\text{reg}} &= \sum_{i=0}^{K*K} \sum_{j=0}^{M} I_{ij}^{obj} \left(2 - w_i * h_i\right) \left[\left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2 + \left(w_i - w_i'\right)^2 + \left(h_i - h_i'\right)^2\right] \\
L_{con} &= -\sum_{i=0}^{K*K} \sum_{j=0}^{M} I_{ij}^{obj} \left[C_i' \log\left(C_i\right) + \left(1 - C_i'\right) \log\left(1 - C_i\right)\right] - \\
&\gamma_{noobj} \sum_{i=0}^{K*K} \sum_{j=0}^{M} I_{ij}^{noobj} \left[C_i' \log\left(C_i\right) + \left(1 - C_i'\right) \log\left(1 - C_i\right)\right] \\
L_{cls} &= -\sum_{i=0}^{K*K} I_{ij}^{obj} \sum_{c \in classes} \left[p_i'(c) \log\left(p_i(c)\right) + \left(1 - p_i'(c)\right) \log\left(1 - p_i(c)\right)\right]
\end{aligned} \tag{2}$$

where $K * K$ represents that the scene is divided into $K * K$ grids by the network, and each grid produces M candidate boxes. Each candidate box will get the corresponding bounding box after the network processing, finally forming $K * K * M$ bounding boxes. If there is no target in the bounding box, only the confidence loss of the box is calculated.

In the regression loss L_{reg}, w_i and h_i represent the length and width of the object's bounding box, x_i and y_i denote the true coordinates of the object, $w_i^{'}$ and $h_i^{'}$ stand for the length and width

of the predict object's bounding box, and $x_i^{'}$ and $y_i^{'}$ represent the predicted object coordinates. I_i^{obj} denotes if the object appears in cell i, and I_{ij}^{obj} denotes that the j_{th} bounding box predictor in cell i is responsible for that prediction.

In the confidence loss L_{con}, C_i represents the category confidence in cell i. γ_{noobj} is a hyper-parameter to suppress the loss of confidence prediction for boxes that do not contain objects, and prevent the confidence from being too close to 0 when cell i does not contain any object.

The classification loss L_{cls} uses the cross entropy function to calculate the loss. When the j_{th} anchor box of the i_{th} grid is responsible for an object, then the bounding box generated by the anchor box will calculate the classification loss function.

The final loss function is the linear summation of the above three kinds of loss functions

$$Loss = \alpha L_{reg} + \beta L_{con} + \mu L_{cls} \tag{3}$$

Among them, α, β, and μ are hyperparameters that aim to keep the three sub-loss functions in the same order of magnitude.

4 Detailed Structure of the Encoder

The encoder module consists of several convolutional blocks with a kernel size of 32×32 and a stride of 32, which are used to simulate the sampling process of SPI. The trained encoding module $g \in \mathbb{R}^{k \times k \times n}$ (k is pattern size set to be 32) is extracted and used as the optimized small-size pattern in practice. The patterns in all three applications are found by first decoding the original image, then switching to a different decoding task and fine-tuning the patterns in the convolutional layers. Assuming that the scene is $s \in \mathbb{R}^{H \times W \times C_{in}}$ (H, W and C_{in} are the height, width and channel

number, respectively), we use the pattern g to scan and sample the scene s to obtain the coupled 2D measurements $F_m \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C}$ (C represents the number of channels of the acquired 2D coupled measurements), and the above process can be characterized as

$$F_m = f_{k*k}(s * g). \tag{4}$$

We implement small-size pattern sampling by embedding non-overlapping small-size patterns in multiple zero-initialized full-size patterns, and quickly switching between full-size patterns. For example, when the sampling rate is 3% and the sampling resolution is 1024×1024, we need 31 32×32 small-size optimized patterns. Each 32×32 pattern needs to be scanned and sampled 1024 times on the 1024×1024 target scene, and finally produces a total of 31 32×32 2D measurements. To achieve this goal, we embed each 32×32 pattern in 1024 1024×1024 zero-initialized patterns without overlapping, so that we get a total of 31744 (31×1024) 1024×1024 locally valid patterns. These patterns are then fed into the DMD to sample the target scene and obtain 31744 1D measurements, which are sequentially reshaped into 31 32×32 2D measurements. This sampling method combines the advantages of compressed sensing and point scanning imaging.[34] Compared with the conventional full-size pattern sampling method, the small-size sampling approach can retain the position information of the target and improve sampling efficiency. Compared with the point-scanning system, our method samples a larger portion of the scene at once, thus reducing sampling times and increasing sampling speed. Moreover, to improve SNR, we binarized the illuminating patterns and performed normalization to signals by maintaining an equal white/black ratio in each illuminating pattern.

The high-dimensional semantic feature extraction module of the encoder consists of several

Transformer layers. Transformer is a basic deep-learning network structure.[26] Thanks to its excellent contextual information capturing and global feature modeling capabilities, the Global Vision Transformer[26] has achieved great success in the image processing field recently. Our Transformer-based encoder can guide the network to focus on the regions with interesting targets, so as to extract high-dimensional semantic features that are effective for imaging and sensing.[26] Since the Transformer only deals with 1D sequence information, we transform the coupled 2D measurements into 1D sequence information by linear projection. In order to retain location information, we introduce learnable position encoding (PE) and fuse them with the feature sequence by direct summation. The computational process is denoted as

$$F_0 = W * F_m + PE. \tag{5}$$

Among them, W represents the linear projection operation, and $F_0 \in \mathbb{R}^{\frac{HW}{1024} \times C}$ represents the output feature sequence. Then we input F_0 into the Transformer block which contains four Transformer layers. Each Transformer layer contains a standard multi-headed attention block (MHA)[26] and a feedforward network (FFN) module, where the FFN consists of a normalization layer and a fully connected layer. The output of the l_{th} layer in the Transformer block can be calculated as

$$F_l^{'} = MHA(LN(F_{l-1})) + F_{l-1}, \quad F_l = FFN(LN(F_l^{'})) + F_l^{'}, \tag{6}$$

where LN represents the normalization layer, F_l represents the output of the l_{th} layer in the Transformer block. The feature sequence output by the Transformer block is $F_L \in \mathbb{R}^{\frac{HW}{1024} \times C}$, which is reshaped to a feature map of $F_e \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C}$ after feature remapping, and sent to the decoder for

subsequent processing.

5 Detailed Structure of the Decoder

When SPIS is applied for large-scale SPI or image-free single-pixel segmentation, the decoder consists of a multi-scale upsampling pyramid network with residual connections. Each up-sampling block is composed of an up-sampling function, a 3×3 convolutional layer, an activation function, and a residual connection. The rearranged layer is used to rearrange the number of channels after upsampling.

The feature map inputed into the decoder is $F_e \in R^{(\frac{H}{32} \times \frac{W}{32} \times C)}$. In the decoder, the height H and width W of the feature map are doubled and the channel number C is halved after being processed by each up-sampling block. After processing by 5 upsampling blocks, we can obtain $F_{up} \in R^{(H \times W \times \frac{C}{32})}$. The channel number C is determined by the resolution of the reconstructed image. When the reconstruction resolution is 1024×1024, C is set to 256. Finally, we use 1×1 convolution to change the number of channels of the feature map F_{up} to 1, and obtain the final reconstructed image $I_R \in R^{H \times W}$. The above process is calculated as

$$F_{up_i} = \sum_{i=1}^{M} H_i^{up}(F_{up_{i-1}}) + F_{up_{i-1}}, \quad I_R = Con_{1\times1}(F_{up}). \tag{7}$$

Among them, F_{up_i} represents the feature map output after the i_{th} upsampling block, H_i^{up} is the i_{th} up-sampling block, M denotes the total number of up-sampling blocks in the decoder.

When SPIS is applied for image-free object detection, the decoder consists of an MSAN module, BMFP module and predict head. The MSAN and BMFP modules are constructed by stacking multi-scale LC-blocks. The LC-block combines local-window (7×7 in this work) self-attention

and channel-wise convolution in a parallel design to model cross-window connections, expanding its receptive field and capturing contextual information. At this point, the complexity of self-attention calculation changes from $O(n^2)$ to $O(n)$. It also provides spatial and channel-wise information interaction, and enables cross-window and cross-dimensional feature complementarity of the decoder. For an input feature F_{i-1}, it first passes through a $1 * 1$ convolution, then split evenly into two feature map groups X_1 and X_2. We formulate such a process as

$$X_1; X_2 = Split(Conv_{1\times1}(F_{i-1})). \tag{8}$$

Next, X_1 and X_2 are separately fed into a local-window Transformer block and a channel-wise convolution block, giving rise to

$$Y_1; Y_2 = Transformer(X_1); Conv(X_2). \tag{9}$$

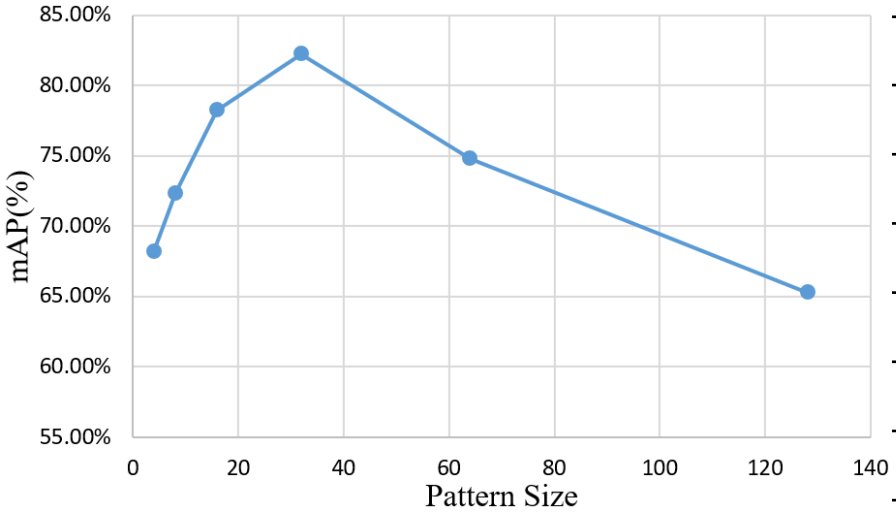
Finally, Y_1 and Y_2 are concatenated as the input of a $1 * 1$ convolution which has a residual connection with the input X. As such, the final output of the i_{th} LC-block is given by

$$F_i = Conv_{1*1}(Concat(Y_1, Y_2)) + F_{i-1}. \tag{10}$$

The feature F_e output from the encoder is first fed into the MSAN module for feature extraction, and the extracted features are termed feature layers. In the backbone part, we acquire three effective feature layers. Then, the three effective feature layers are fed into the BMFP module for bi-directional multi-scale feature fusion. In BMFP, we upsample and downsample the features simultaneously and perform feature fusion to fully fuse the feature information at different scales.

After the MSAN and BMFP processing, we obtain three enhanced effective feature layers. They are then fed into the predict head module for the final object detection. We divide the predict head into two parts to implement classification and regression separately, and finally integrate them when making predictions.

6 The pattern size selection experiment.



Pattern Size	mAP(%)
4	68.24%
8	72.32%
16	78.20%
32	82.20%
64	74.80%
128	65.25%

Fig 8: **The experiment results of image-free object detection under different pattern sizes.** We can see that the pattern size of 32×32 produces the highest detection accuracy.

In real-world experiments, the size of illumination patterns will affect the performance of SPI and SPIS. Theoretically, the smaller the pattern size, the better the imaging and image-free sensing performance. This is because a smaller pattern size can retain more detailed location information and capture rich local features. However, as the pattern size becomes smaller, the luminous flux becomes smaller, which will reduce the signal-to-noise ratio, thus reducing the performance of imaging and image-free sensing.

To select the most suitable pattern size, we studied the image-free object detection performance under different pattern sizes at a fixed sampling rate of 5%. We tried 6 different pattern sizes

(including 4×4, 8×8, 16×16, 32×32, 64×64, 128×128), and the object detection results are shown in Fig. 8). We can see that the pattern size of 32×32 pixels produced the best performance. However, we anticipate that in different practical application scenarios, 32×32 may not be the most appropriate pattern size. For example, in a low-light environment, a larger pattern size can bring more light flux. This means that the optimal pattern size depends on specific lighting and noise conditions. How to choose the most appropriate pattern size according to different applications is one of future research directions.