Supplementary Information

Inverse design of organic light-emitting diode structure based on deep neural networks

Sanmun Kim[1], Jeong Min Shin[1], Jaeho Lee[2], Chanhyung Park[1], Songju Lee[1], Juho Park[1], Dongjin Seo[1,3], Sehong Park[2], Chan Y. Park[3], Min Seok Jang[1,]*

[1] School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea

[2] OC Optical technology task, LG Display, Seoul, 07796, Republic of Korea

[3] KC Machine Learning Lab, Seoul, 06181, Republic of Korea

* jang.minseok@kaist.ac.kr

S1. Simulation structure for Chance-Prock-Silbey (CPS) model

For the CPS calculation, the organic layer was divided into three sections, where the layer in the middle is lossless. The lossless layer has a thickness of 2 nm, and the dipole is positioned at the centre of the lossless layer (Figure S1). The lossless region was introduced since our in-house CPS code cannot simulate a dipole inside a lossy medium.

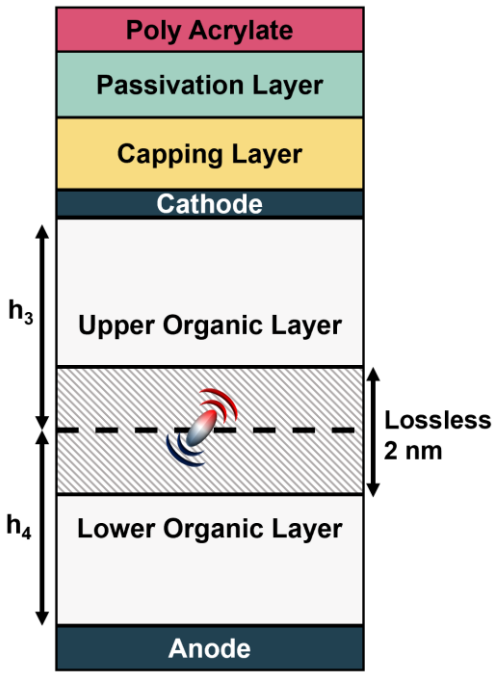

Figure S1. Schematic of the simulated OLED structure. A lossless layer is introduced inside the organic layer.

S2. Dataset size and neural network complexity

RMSE of the neural network prediction is monitored while varying the training dataset size and network complexity. When the number of training sample is varied as shown in Figure S2a, an abrupt drop in the inference error is observed between the network trained with 75,000 samples and the network trained with 100,000 samples.

Neural network complexity was varied by changing the number of hidden layers (Figure S2b), and changing the number of nodes in each layer (Figure S2c). RMSE monotonically decreases as the network complexity increases. A further decrease in RMSE was expected for a neural network with higher complexity, but the network complexity is no longer expanded for the training stability.

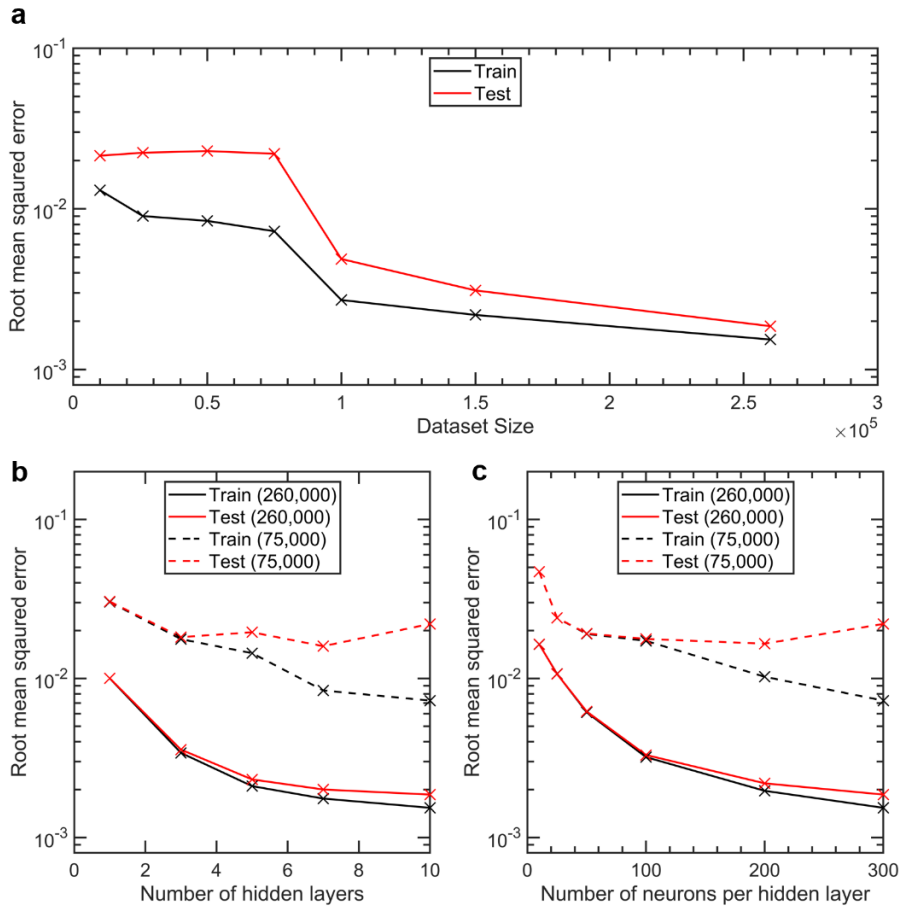

Figure S2. (a) RMSE of the prediction from neural networks trained with different number of training samples. RMSE values for different number of hidden layers (b), and different number of nodes in each layer for a ten-layer-deep network (c). The number inside the bracket indicates the training sample size.

S3. Training neural networks

The forward neural network is trained with a dataset consisting of 260,000 randomly sampled structure-LEE pairs. The datatype of the input and output is a double-precision vector of length six and eighty-one, respectively. The training is done with a batch size of 4,096 for 5,000 epochs with an ADAM optimizer of varied learning rate (Figure S3). The dataset was split into size ratio of 8:1:1 and were assigned to training, validation, and test dataset. Root mean squared error is used as a measure for training loss. The train and test RMSE value of the trained neural network are 1.54×10^{-3} and 1.86×10^{-3}. In the case of the inverse neural network, the network weights diverged during the training process if the mean squared error was taken as a measure for error. Thus, the mean absolute error is chosen as the measure during the training process. The inverse neural network is also trained with an ADAM optimizer for 4,000 epochs and the training is done with the same dataset used for the forward network. The test set RMSE value of the inverse neural network is 4.42×10^{-3}.

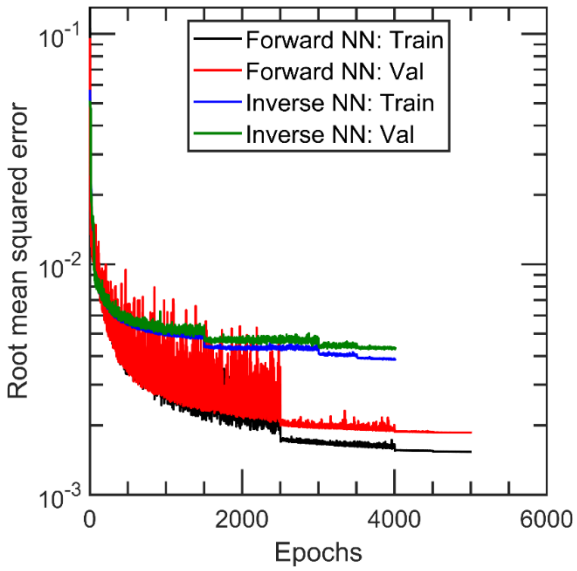

Figure S3. Training curves of our neural network. RMSE curve of the inverse neural network lies above the forward network. The points of jump discontinuity occurred when the learning rate was reduced.

S4. Extrapolation with Forward Neural Networks

In order to test the extrapolation capability of the spectrum prediction network, we plotted a contour of RMSE values for different combinations of h_3 and h_4 outside the trained regime. For comparison, n_1, n_2, h_1, and h_2 are fixed to 1.75, 1.7, 1000 nm, and 130 nm, respectively. The contour (Figure S4) shows that the neural network fails to predict the LEE curve once the inputs leave the trained range. This error contour implies that the forward neural network has not learned the physics from the training process but only learned to predict the LEE curve by interpolating from the preexisting dataset.

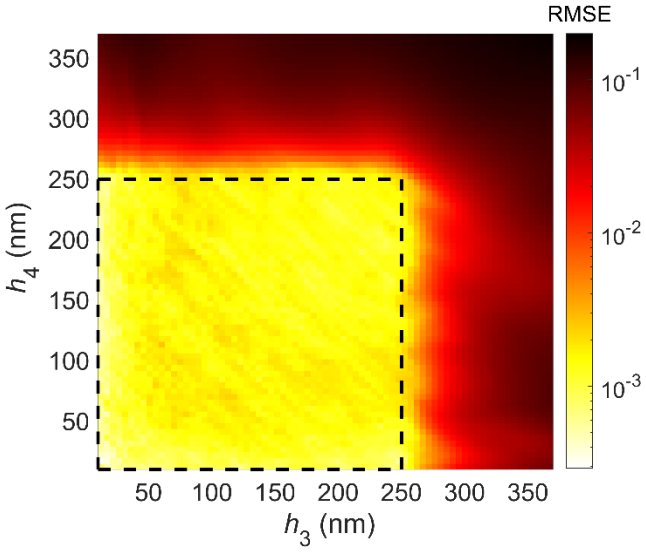

Figure S4. Contour or RMSE values as a function of h_3 and h_4. The red dashed line denotes the boundary of the trained range of the neural network.

S5. Neural network-based optimizations and one-time computational cost

Computation time can be broken down into three parts: time involved in data creation with CPS model, neural network training time, and the inference time. The data creation time is the dominant step in terms of computation time. It takes ~2.5 seconds to calculate a LEE spectrum consisting of 81 wavelength points on a computer configured with two Intel® Xeon Gold® 5220 CPUs (18 cores each). Hence, ~7 days of computation is required to obtain our dataset containing 260,000 structure-LEE spectra pairs. An angular LEE is obtained as a byproduct during the calculation of LEE spectrum, so this 7-day computation was indeed a one-time computation cost.

Neural network training time is heavily dependent on the neural network complexity and the data size for training. In our case with 600,000~900,000 trainable weights and 260,000 data pairs, the training process takes less than an hour on a computer consisting of a single Xeon processor with NVIDIA® Tesla® V100 GPU accelerator. It should be noted that the identical forward network is used for every optimization task except for the RGB subpixel optimization task which required predicting angular LEEs. Hence, the neural network training time can also be considered as a one-time computational cost.

A typical inference time involved in a single optimization task is in order of minutes, and thus the computational cost of inferencing can be neglected compared to the computational cost required by the previous steps.

To sum up, the computation time of an optimization process involving a neural network is dominated by the sample creation time. Hence, the optimization involving a neural network performs superior in terms of speed if the total number of candidates considered is greater than the number of samples required for network training. Also, the supremacy becomes more evident if multiple optimization tasks can be carried out on a single network,

as done in this work. The total number of candidates considered in the optimization tasks are: ~1.5×10^5, 8×10^5, 6×10^6, and 6×10^6 for the inverse design of a single realistic LEE spectrum, the inverse design of a single flat LEE spectrum, the maximization of LEE at a given wavelength, and the minimization of the white color shift, respectively. The number of candidates could be reduced by employing a fine-tuned genetic algorithm or by the application of other optimization methods, but it should be noted that the immense search empowered by neural network assisted genetic algorithm greatly reduces the possibility of optimization terminating at local optima.

S6. One-to-many mapping in inverse neural network

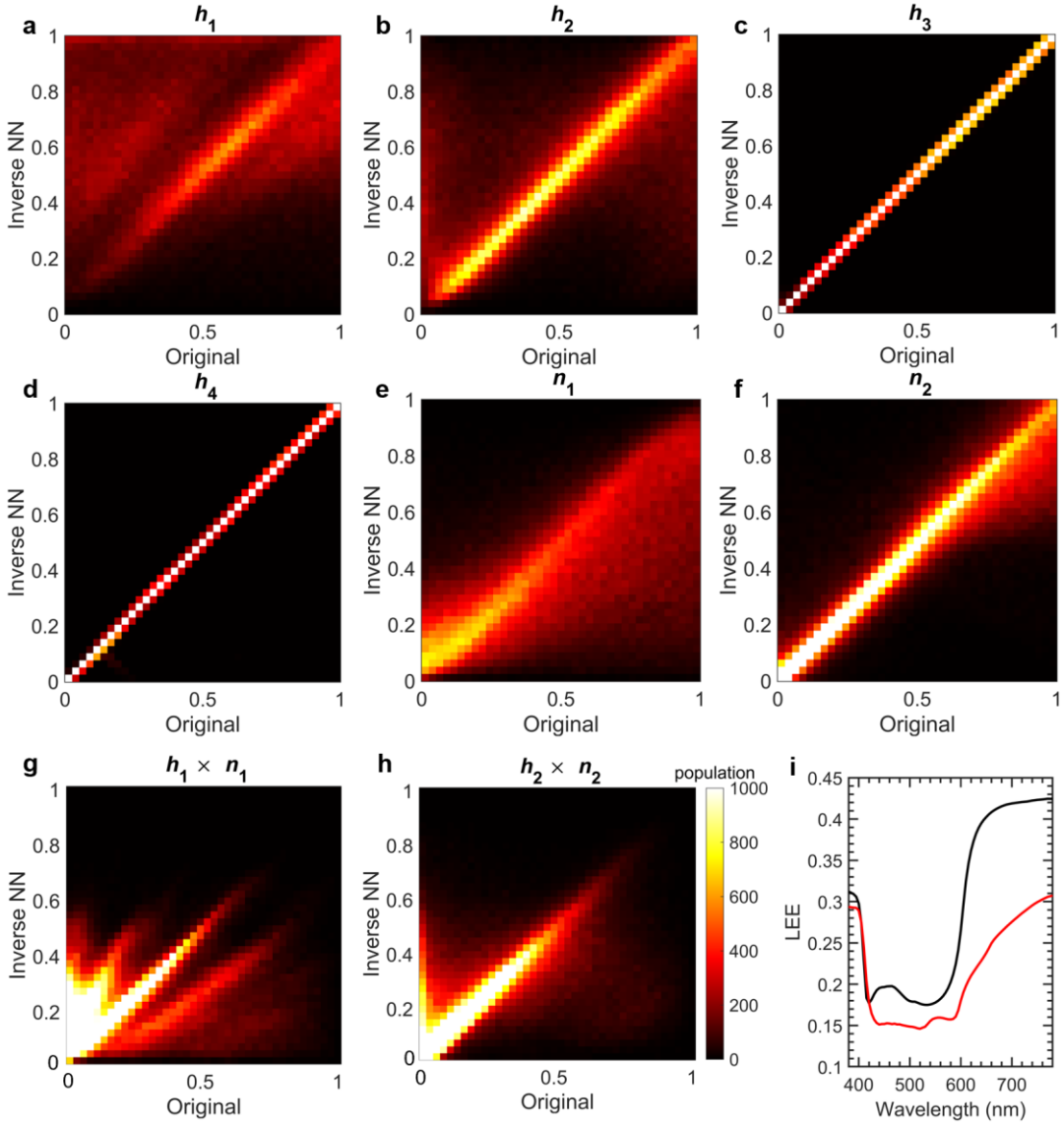

Figure S6. One-to-many mapping in inverse neural network. (a-h) Histograms showing the distribution of 260,000 inverse neural network's normalized outputs. The *x*-axis of the histogram is the normalized original design parameter used to generate the input LEE spectra and the *y*-axis is the normalized parameter obtained from the inverse network. If the output design parameters preserves the original values, only the diagonal element should have non-zero values as shown in h_3 and h_4. Histograms for the passivation layer (h_1, n_1) does not exhibit linear symmetry, which implies that mapping to the latent space (layer of design parameters) of the network is biased. (i) LEE curves of two structures that have the same h_3, h_4 values but differs in other values. h_3 and h_4 may be the dominant design parameters but design parameters for the passivation and capping layer also affect LEE.

S7. Relative error plot for Figure 3b

Relative error is defined as RMSE divided by the target flat value. It is expected that RMSE itself has an effect that is proportional to the value of target value. To account for this, Figure S7 was plotted with the same data in Figure 3b.

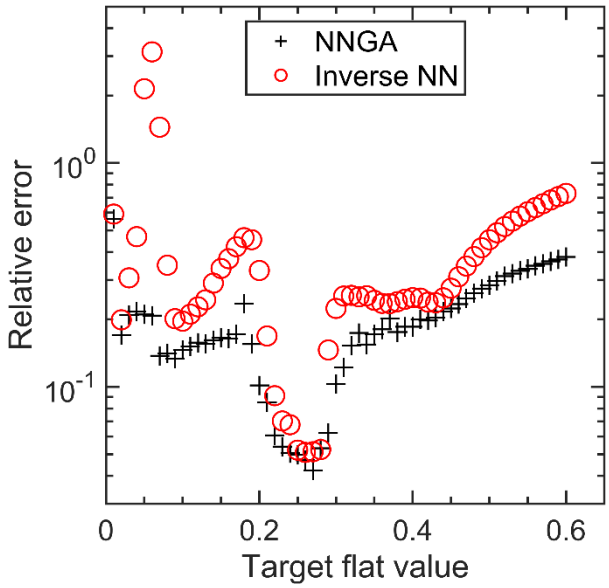

Figure S7. Relative error plotted as a function of the target flat value.

S8. Neural network assisted Genetic Algorithm (NNGA) for the inverse design of LEE

NNGA for the inverse design runs in the following order (Figure S8). In each generation, the individuals, each consisting of six design parameters, are sorted according to their figure of merit, calculated with the neural network. The individuals with high figure of merit are succeeded to the next generation. Along with the individuals' identical copy, their mutated and crossed over versions fill the next generation. Mutation adds uniform random noise to the normalized structure parameters. Cross-over occurs between the individuals with high figure of merit and the average value is taken. The remaining portion of the next generation is filled with randomly created individuals. Structure parameters that have values outside [0, 1] were reset to 0 and 1 to assure the accuracy of the output LEE.

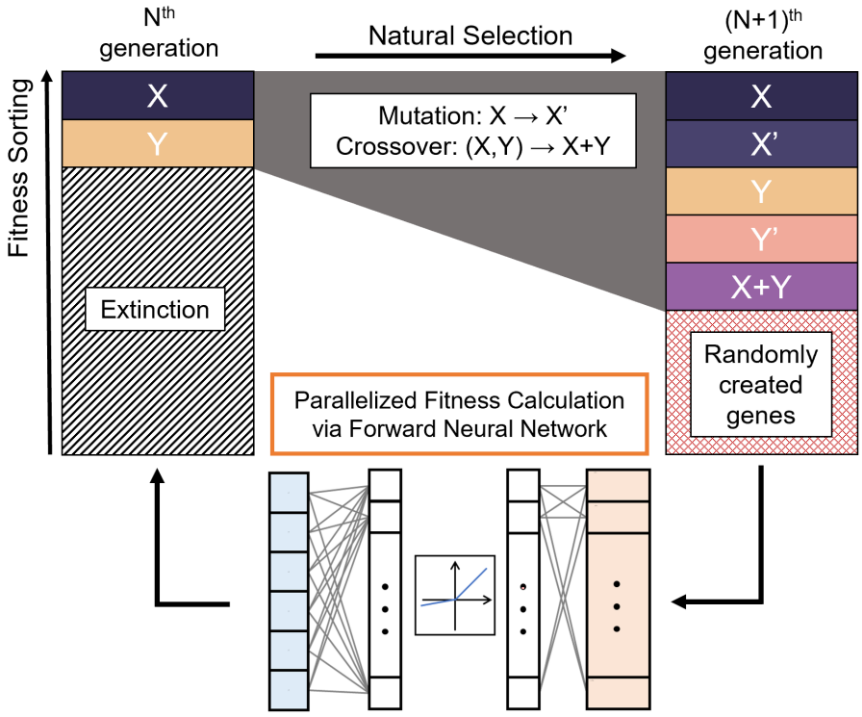

Figure S8. A schematic diagram of NNGA. A parallelized neural network is employed as a LEE spectra calculator.

S9. Origin of the jump discontinuity in Figure 4b

Multiple jumps are observed in the curve representing the optimized values of h_1 in Figure 4b. To analyze how the optical property differs between the structures, we plotted the angular LEE curves for h_1 = 856 nm and h_1 = 1266 nm which were values before and after the jump at λ = 495 nm (Figure S9a). The angular LEE for h_1 = 856 nm shows five peaks whereas the case of h_1 = 1266 nm exhibit six peaks, implying that the optimization result jumps between two different mode configurations with the number of modes differing by one from each other. Similar jumps are also observed at longer wavelengths. For instance, at λ = 665 nm, the NNGA-optimized h_1 value abruptly changes from 1150 nm to 603 nm, and the resulting angular LEE spectra exhibit different mode for the two cases as shown in Figure S9b.

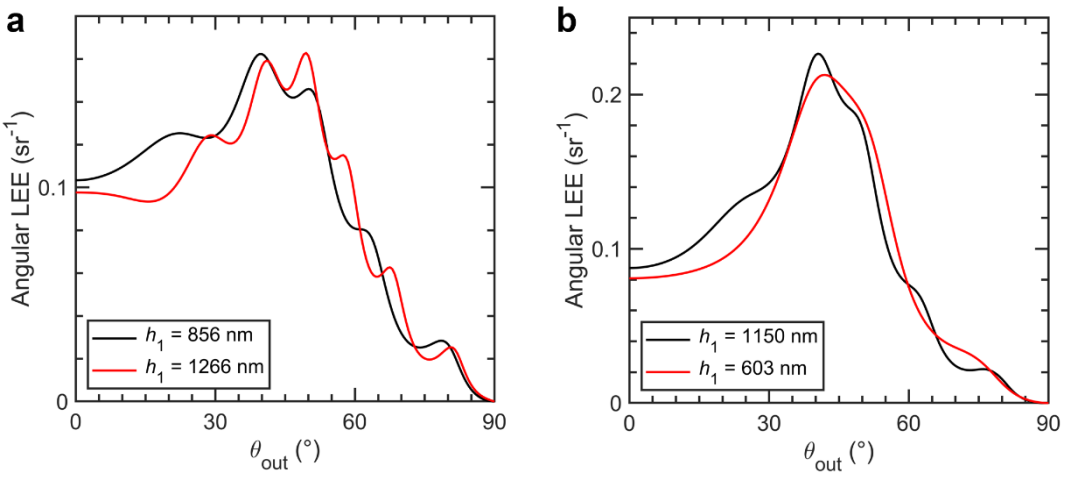

Figure S9. Plots of angular LEE distribution from structures corresponding to jumps at (a) λ = 495 nm and (b) λ = 665 nm of Figure 4b. See Figure 4b of the main article for more detail on the optimized values of LEE and structure parameters.

S10. Angular LEE and spectra of subpixels

Angular LEE is the LEE corresponding to a specific k-vector. In other words, LEE is obtained by integrating the angular LEE values between 0°–90°. Generally, an angular LEE spectrum resembles a Lorentzian curve with the peak blue shifting with the increasing viewing angle. Five new neural networks are designed and trained to predict the angular LEE at 0°, 15°, 30°, 45° and 60° viewing angle. The previously used forward network design was adopted again. The trained networks have test RMSE error of 1.44×10^{-3} sr^{-1}, 1.34×10^{-3} sr^{-1}, 1.08×10^{-3} sr^{-1}, 1.01×10^{-3} sr^{-1}, and 1.20×10^{-4} sr^{-1}.

S11. NNGA for minimizing the white point deviation

The actual NNGA runs in the following order. In the first-generation gene pool, the structure parameters (n_1, n_2, h_1, h_2, h_{3R}, h_{3G}, h_{3B}, h_{4R}, h_{4G}, h_{4B}) are randomly generated. Neural networks predict angular LEEs at 0°, 15°, 30°, 45° and 60° for each individual in the gene pool. The XYZ coordinates (CIE 1931) of the RGB subpixels are obtained by multiplying the corresponding emitter spectrum to LEE and taking the inner product with the color matching function. The participating ratio (A, B, C) between the RGB subpixels are obtained from the following equation.

$$\begin{bmatrix} R_X & G_X & B_X \\ R_Y & G_Y & B_Y \\ R_Z & G_Z & B_Z \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} W_X \\ W_Y \\ W_Z \end{bmatrix} = \begin{bmatrix} 0.95 \\ 1 \\ 1.09 \end{bmatrix}$$

Here, all entries are defined when the observer is directly above the device (0 degrees). Since A, B, and C cannot be negative, individuals with negative (A, B, C) entries are removed from the gene pool. Also, individuals having $CIEy = B_Y/(B_X+B_Y+B_Z) > 0.15$ are removed from the gene pool to secure the color purity of the blue subpixel. The fitness of the individuals are calculated by mapping XYZ coordinates at each viewing angle to CIE 1976 color space and measuring the distance from D65 illuminant, i.e. $\Delta_{max} = \mathrm{Max}(\Delta_{15}, \Delta_{30}, \Delta_{45}, \Delta_{60})$, where $\Delta_\theta = [(u'_\theta - u'_{\mathrm{D65}})^2 + (v'_\theta - v'_{\mathrm{D65}})^2]^{1/2}$.
Natural selection is performed based on the Δ_{max}, reducing angular color shift over generations. The optimization, NNGA, automatically shuts down after meeting a pre-defined convergence condition: no reduction in Δ_{max} for more than ten generations.

S12. Removing one-to-many mapping problem through introduction of angular LEE spectra

As investigated in the main text, one-to-many mapping problem is predominant in the inverse design of photonic devices. Since the need of tool for the inverse design of OLED was first suggested for industrial reasons where inferring the actual device structure based on their spectrum is essential, circumventing the inverse scattering issue through tandem network or NNGA which leads to a completely different design parameters does not solve the problem. It has been expected that use of multiple spectra collected from different observing angle would narrow down the number of possible designs but has not been. In this section, we investigate the inverse design of OLED with multiple angular LEE spectra as the design target. Since each spectrum contains 81 wavelength points, using angular LEE spectrum for N different angles would require 81×N nodes at the input of neural network, thus making the training process difficult. Also, a neural network trained to admit multiple spectra as inputs would lack the flexibility when an additional information (e.g. a spectrum at different angle) is given. Due to these reasons, NNGA is employed to tackle the inverse design problem with multiple spectra.

We define the loss function of NNGA as a simple sum of mean squared error between the design target and the output from the forward neural network predicting the total LEE (divided by 2π, the solid angle of emission in steradians) and angular LEE at viewing angle of 0°, 15°, 30°, 45°, 60°. By running multiple iterations, we were able to obtain the results shown in Figure S12 and Table S12. As it can be seen in Figure S12a, the inverse design with NNGA resulted in angular LEE spectra almost identical to the target angular LEE spectra. Table S12 summarizes the removal of the inverse scattering issue in determining (n_1, n_2, h_3, h_4) when five or six spectra are incorporated for identifying a single structure. However, a

relatively large error with an average of (13 nm, 12 nm) still resides in finding the values of (h_1, h_2). In the case of h_2, the source of error can be inferred from Figure 6b. As the difference between the refractive indices of the passivation layer and capping layer decreases, the cavity effect arising at their interface drops until it becomes physically impossible to inverse-design the target structure without an error at $n_1 = n_2$. On the other hand, if the difference between n_1 and n_2 of the target structure is large, it becomes easier for NNGA to identify the value of h_2.

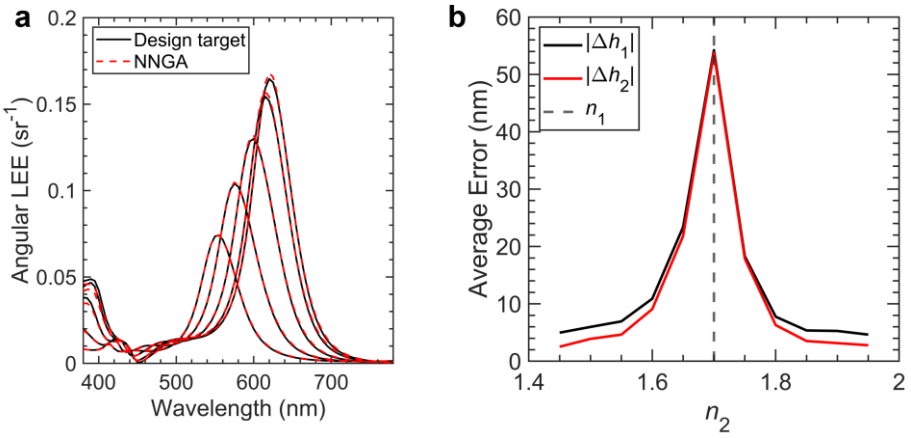

Figure S12 Inverse design with angular LEE spectra. (a) An example of inverse-designed angular LEE. The black line denotes the angular LEE curve from device corresponding to (h_1, n_1, h_2, n_2, h_3, h_4) = (509 nm, 1.88, 74 nm, 1.45, 79 nm, 225 nm) at viewing angle of 0°, 15°, 30°, 45°, 60° with peak height decreasing with the viewing angle. The red dashed line is the inverse-designed angular LEE curves at (h_1, n_1, h_2, n_2, h_3, h_4) = (532 nm, 1.84, 63 nm, 1.44, 80 nm, 225 nm). (b) Average discrepancy between the target height and inverse-designed height, $\Delta h_2 = h_{2,target} - h_{2,NNGA}$, plotted as a function of n_2 when n_1 is fixed to 1.9.

Table S12 : Average error between the target design parameters and the inverse-designed parameters, depending on the type of spectra used for indentification.

Design Parameters	total LEE	Angular LEEs + total LEE	Design range
h_1	284.8 nm	13 nm	500 nm – 1500 nm
n_1	0.05	0.006	1.6 – 1.9
h_2	36.7 nm	12 nm	10 nm – 250 nm
n_2	0.05	0.015	1.4 – 2.0
h_3	1.13 nm	0.3 nm	50 nm – 250 nm
h_4	0.63 nm	0.2 nm	50 nm – 250 nm