(\*The following program is based on a global optimization algorithm, using reverse fitting of the experimentally obtained PA power-dependence curve data to determine the true PA response curve of prepared nanoparticles. The software used is ***1stOpt 10.0*** from 7D-soft high technology Inc.\*)

//Algorithm=UGO1; //This comment indicates that a Universal Global Optimization algorithm will be used for the fitting process.

//QuickReg = 2;

Parameter a1=[-50,-20],a2=[-400,-300],a3=[-20,0],b1=[5,15],b2=[100,160],b3=[2,20],con1=[0,1]; //Parameters need to be determined using the following mathematical model and experimental results

Variable x, y; //x represents the power density in kW cm-2; The y represents the logarithmic form of fluorescence counts

Constant areaofspot=2.826E-9, con2=171394,

ConstStr p1=(a2-a1)/(b1-b2), //Constant strings used to simply the formula.

p2=(a3-a2)/(b2-b3),

pow=log(100000\*x\*areaofspot\*(1E+9)\*exp(-0.000020989278\*(r^2))/con2), //A model used to extract the intensity of a certain position in a gaussian beam spot when the overall intensity is given.

realpowerdependence=if(pow<p1,a1+b1\*pow,if(pow>=p2,a3+b3\*pow,a2+b2\*pow)); //A simplified piecewise function model is used to approximate the S-shaped trend of classical photon avalanche as closely as possible.

Function y=log(int(con1\*(10^realpowerdependence)\*6.28\*r,r=0,2000)); //The mathematical model to be fitted.

2<p2<3;

2<p1<3;

p1<p2;

Data; //Raw data for photon avalanche response of prepared nanocrystals showing the overall PA curve under a gaussian beam excitation; x represents the power density in kW cm-2; The Y-axis represents the logarithmic form of fluorescence counts.

// x y

534.65525 4.32222

543.24325 4.39794

551.3705 4.47712

556.4055 4.50515

561.5925 4.54407

564.224 4.60206

570.41325 4.65321

572.1185 4.69897

584.50175 4.81291

589.38 4.8451

591.47 4.90309

593.42 5.0499

595.59775 5.20412

597.3 5.353

598.96075 5.47712

600.154 5.584

601.352 5.7031

602.445 5.79758

603.426 5.92

604.532 6.0436

608.42275 6.25836

612.06125 6.39244

617.01075 6.45332

627.38475 6.56348

650.218 6.70672

658.35 6.75051

673.28875 6.81657

686.603 6.87674

704.63875 6.96095

723.05925 7.05154

741.40375 7.12767

759.981 7.20041

773.68475 7.25091

790.172 7.33385

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(\*Based on the intrinsic response curve of PA nanoparticles obtained from the above program, the following program uses the Monte Carlo method to simulate the process of Gaussian beam scanning for super-resolution imaging of PA nanoparticles. The software used is ***Mathematica 12.0*** from Wolfram Research. \*)

ClearAll["Global`\*"]

wavelength=1064; //The wavelength of the pumping laser.

NA=1.45; //The Numerical aperture of the objective lens used.

areaofspot=0.2826\*10^(-8)\*(wavelength/1064)^2;(\*cm^2\*)//The size of the beam spot.

powerdensity=655.1;(\*kW/cm2\*)//A certain power density selected for simulation.

power=powerdensity\*areaofspot\*10^9;(\*uW, real inputpower\*)

laserprofileorigin[r\_]:=(2\*BesselJ[1,2\*Pi\*NA\*r/wavelength]/(2\*Pi\*NA\*r/wavelength))^2; //A mathematical model used to fit the intensity distribution of a laser beam.

data=Table[{i,laserprofileorigin[i]},{i,1,5000,1}];(\*get data of laser profile\*)//Get the intensity of light at given postion in the focal spot.

gaussianModel=a Exp[-(x^2)/(2 b^2)];

fit=NonlinearModelFit[data,gaussianModel,{a,b},x];

laserprofile[x\_]:=fit[x]/fit[0];(\*the profile was fitted with real gaussian distribution\*)

constant=Integrate[laserprofileorigin[r]\*2Pi\*r,{r,0,50000000}];

powerdensityprofile[rr\_]:=100000\*power\*laserprofile[rr]/constant;(\*uW/nm^2 to kW/cm^2\*)

a1=-32.586; // Parameters obtained using the 1stopt program above.

a2=-412.245;

a3=-17.017;

b1=11.259;

b2=138.539;

b3=7;

con1=0.0976386019;

p1=(a2-a1)/(b1-b2);

p2=(a3-a2)/(b2-b3);

p1

p2

f[x\_]:=Piecewise[{{a1+b1\*x,x<=p1},{a2+b2\*x,p1<x<=p2},{a3+b3\*x,x>=p2}}]; // The model used to show the S curve PA response.

Plot[f[x],{x,2.5,4}];

powerdependentdatareal=Table[{10^x,10^f[x]},{x,1,4,0.01}];

ListLogLogPlot[powerdependentdatareal]

powerdependentdatafunction=Interpolation[powerdependentdatareal,InterpolationOrder->1];

nonlinearemissionfunction[r\_]:=powerdependentdatafunction[powerdensityprofile[r]];

NIntegrate[powerdensityprofile[r]\*2Pi\*r/100000,{r,0,500}];power/areaofspot;

Integrate[(powerdensityprofile[r]/100000)\*2\*Pi\*r,{r,0,3000}];

centerofparticle1={55,-33}; // Set the location of nanoparticle1.

centerofparticle2={-66,45};// Set the location of nanoparticle2.

particlediameter=27;(\*nm\*)// Set the size of nanoparticles.

points1=200; // Number of emitters in particle; The higher this value, the finer the image will be.

plotrange={{-150,150},{-150,150}}; // The image size.

region1=RegionUnion[Disk[centerofparticle1,particlediameter/2],Disk[centerofparticle2,particlediameter/2]]; //Define the region containing two nanocrystals.

pts1=RandomPoint[region1,points1]; //Generate random points in the defined region.

Graphics[Point[pts1],Frame->True,AxesOrigin->{0,0},Axes->True,PlotRange->plotrange]

region2=Rectangle[{plotrange[[1,1]],plotrange[[2,1]]},{plotrange[[1,2]],plotrange[[2,2]]}]; //Define the region will be scanned.

points2=10000;(\*Scanning points\*) //Define number of points will be scanned in the imaging region.

pts2=RandomPoint[region2,points2]; //Define the locations of points will be scanned in the imaging region..

Graphics[Point[pts2],Frame->True,AxesOrigin->{0,0},Axes->True];

intensitymapping=Table[{pts2[[j]],Total[Table[nonlinearemissionfunction[RegionDistance[Point[pts2[[j]]],pts1[[i]]]],{i,1,points1}]]}//Flatten,{j,1,points2}];//Get the simulated emission intensity at each scanning point.

customColorFunction[z\_] := Blend[{{0, RGBColor[26/255, 40/255, 71/255]}, {0.7, RGBColor[33/255, 166/255, 117/255]}, {1, RGBColor[255/255, 241/255, 67/255]}}, z];

ListDensityPlot[intensitymapping, PlotRange -> Full, ColorFunction -> customColorFunction, ColorFunctionScaling -> True] //Plot the simulated super-resolution imaging.