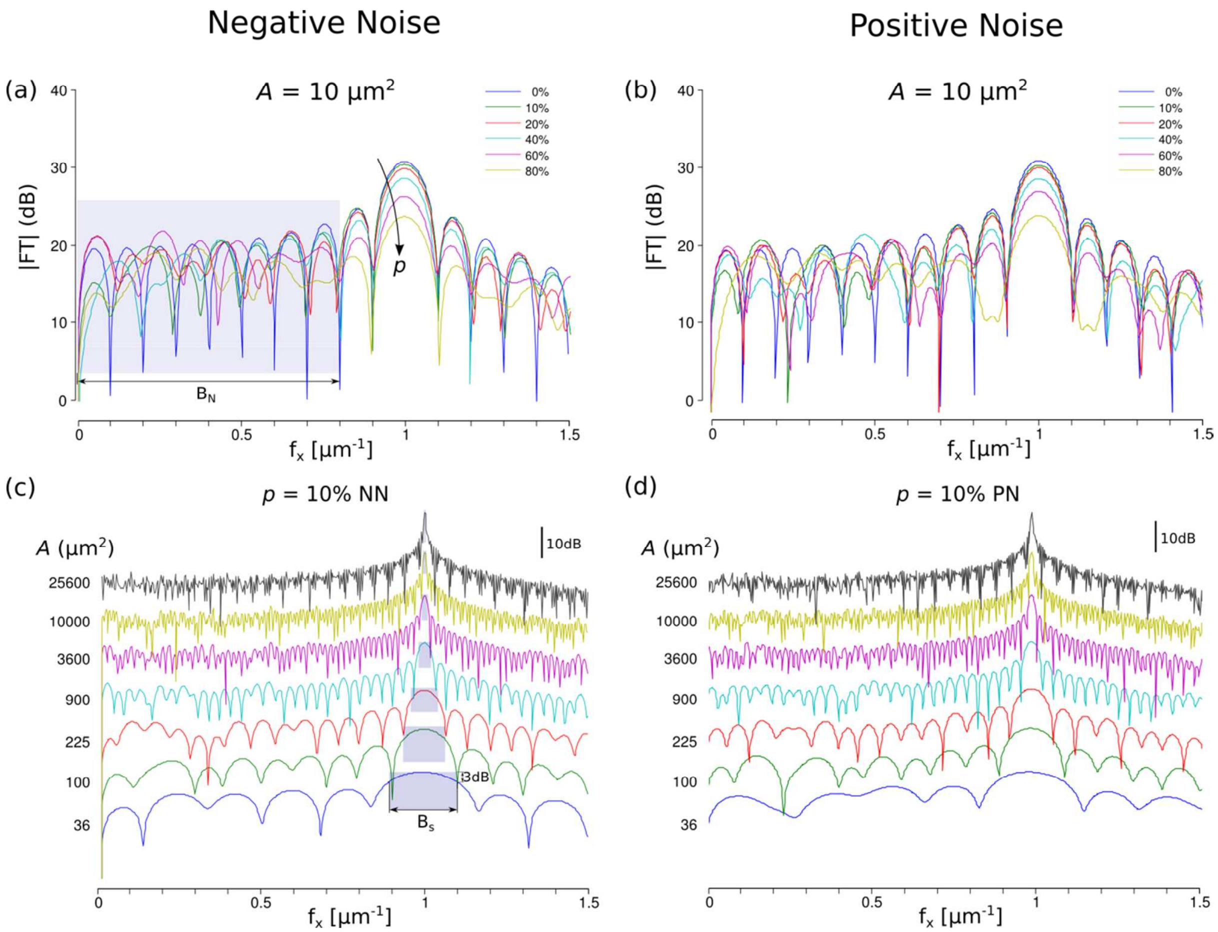
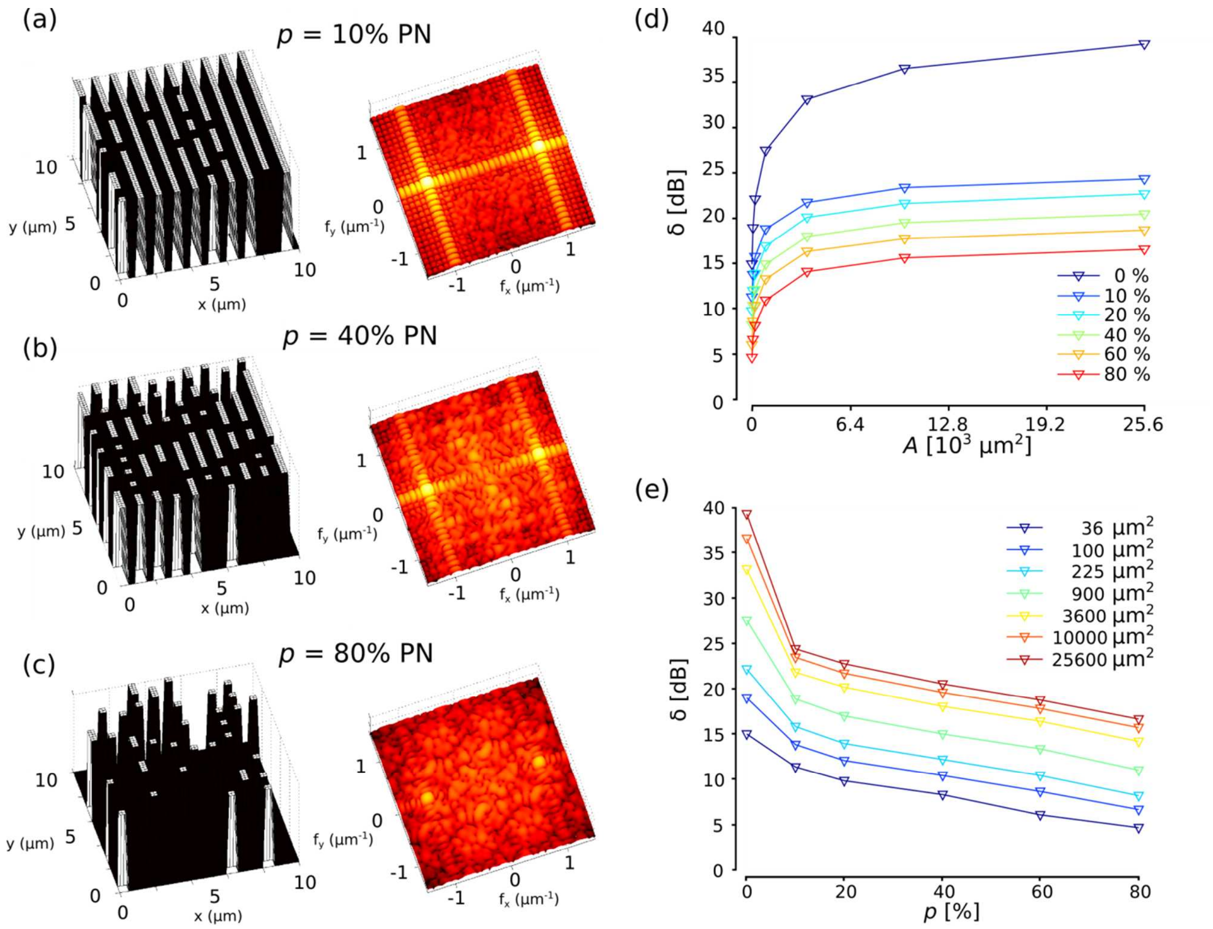
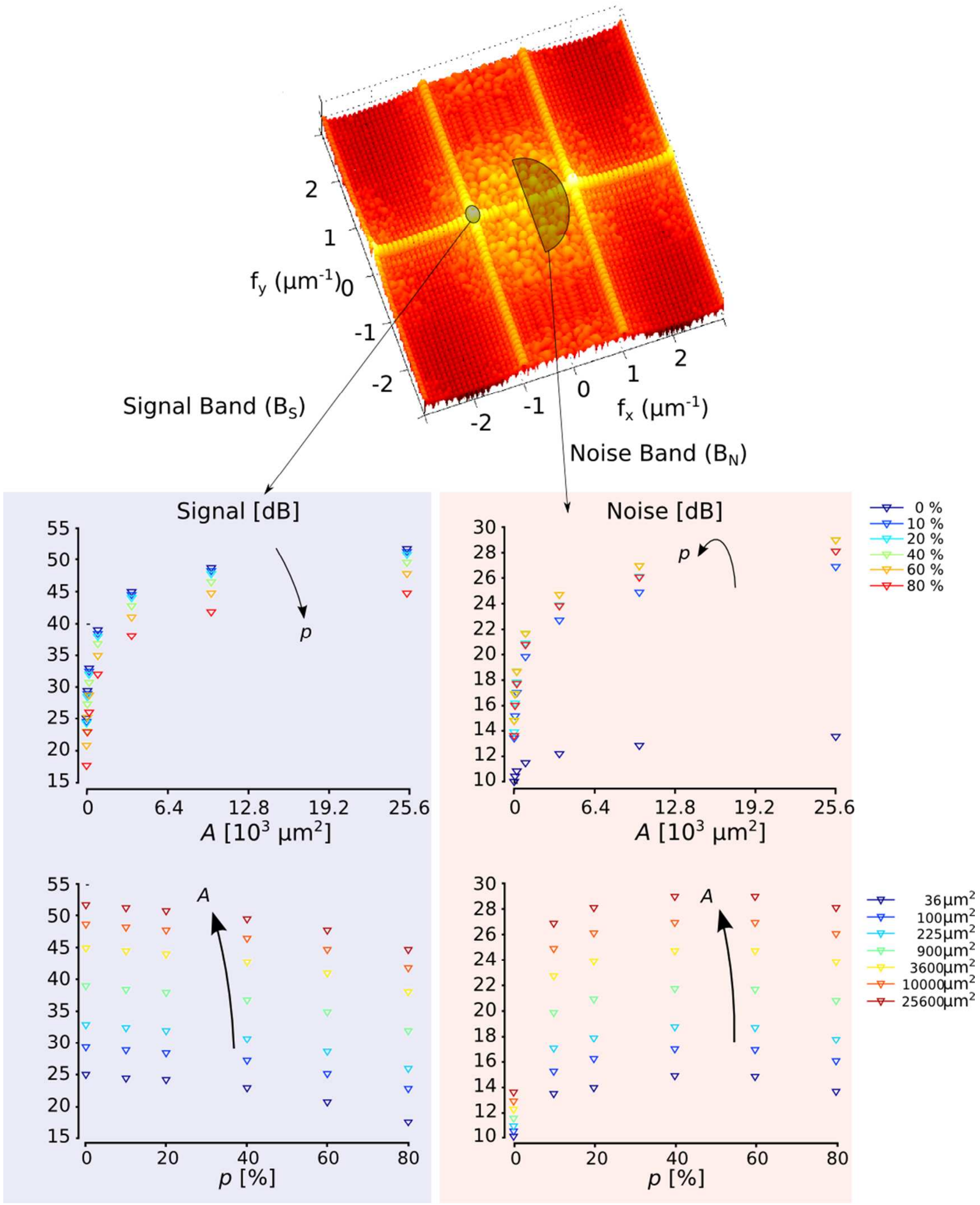
SUPPLEMENTARY MATERIALS



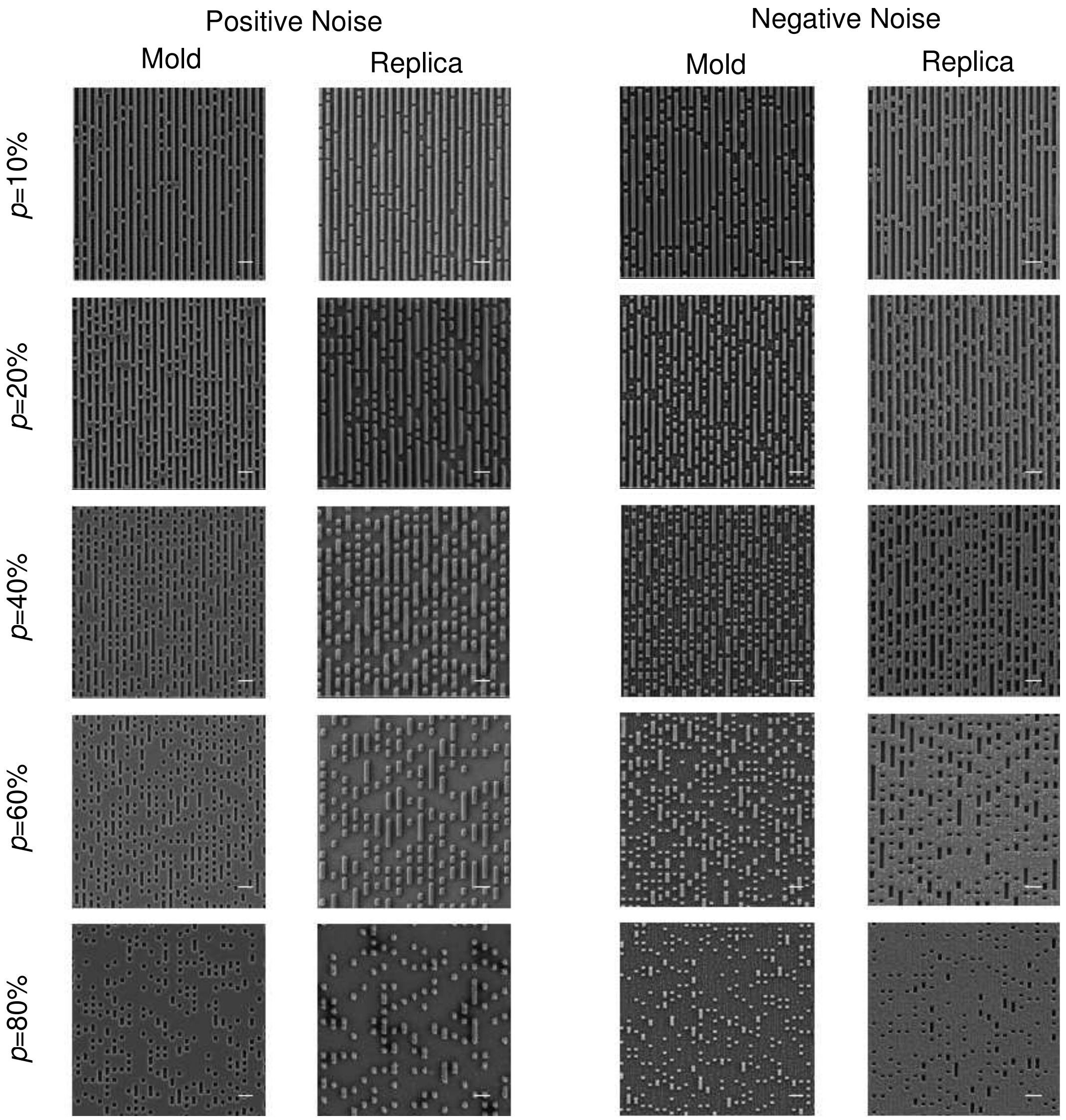
Supplementary Figure 1: FT spectra at $f_y = 0$ for different values of *p* (a,b) and *A* (c,d). (a) and (c) are calculated in case of NN. (b) and (d) are calculated in case of PN. The blue boxes in (a) and (c) represent B_n and B_s.



Supplementary Figure 2: Directionality quantification for NGs with PN. (a-c) 10 µm x 10 µm area of noisy NGs and relative FTs for *p* = 10%, 40% and 80%. (d,e) δ as a function of *A* and *p*. As expected, no difference is found with respect to NGs with NN (see Fig. 3 and Fig. 4).



Supplementary Figure 3: Calculation of Signal (left column) and Noise (right column) as defined in Eq. 1 for different values of on *A* and *p*.



Supplementary Figure 4: SEM images of the noisy NGs. In case of high values of *p*, NN replicas are characterized by scattered nano-posts on an overall flat surface, while PN leads to surfaces with scattered nanoposts. Scale bars = 2 µm.

MATLAB scripts

```
-----------------------------------------------------------------------------------------------
%% noise generator
close all,clear all,clc

dimX=10;
dimY=10;

% period
T=1;
% duty cycle
DC=0.5;
%number of elements
nX=dimX/T;
nY=2*dimY;

% threshold
soglia=0.80;

terr=zeros(nY,nX);

nome=['noise',num2str((1-soglia)*100)];
app=zeros(nX,nY);
app=uint(app);
terr=app.MantBits;
clear app,
terr(:,(1:2:nX))=1;

imagesc((0:dimX/nX:dim),(0:dimY/nY:dim),terr),colormap copper,
xlabel('\mum'),ylabel('\mum'),

%% NEG noise generation
dist=rand(nX,nX/2);
T=0.90; %non ideality threshold
dist(dist>=soglia)=1;
dist(dist<soglia)=0;
figure,imagesc(dist),colormap copper,

terr(:,(1:2:nX))=not(dist);
imagesc((0:dimX/nX:dim),(0:dimY/nY:dim),terr),colormap copper,
xlabel('\mum'),ylabel('\mum'),
vertici=5;
x=[0 dimX dimX 0 0];
y=[0 0 dimY dimY 0];
DXF_poly(fid,x,y,vertici,5,1),


dx=dimX/nX*DC;
dy=0.5;

for I=1:nX
    non_id=find((dist(:,I)));

    x=[(I-1) (I-1)+DC (I-1)+DC (I-1) (I-1)]*T;

    aperto=false;
    for J=1:length(non_id)-1
        if not(aperto)
            yi=non_id(J);
        end
        if non_id(J)+1==non_id(J+1)
            aperto=true;
        else
            yf=non_id(J);
            y=[yi yi yf+1 yf+1 yi]*dy;
            DXF_poly(fid,x,y,vertici,7,1),
            aperto=false;
        end

    end

    x=[(I-1)*T+(T-dx) I*T I*T (I-1)*T+(T-dx) (I-1)*T+(T-dx)];
    y=[0 0 dimY dimY 0];
    DXF_poly(fid,x,y,vertici,7,1),
end
DXF_end(fid),
fclose(fid),

%% POS noise generation
dist=rand(nX,nX/2);
dist(dist>=soglia)=1;
dist(dist<soglia)

[fid,err]=DXF_start([nome,'_pos.dxf'],10^scal);

dx=dimX/nX*DC; dy=0.5;
for I=1:nX
    non_id=[0;find((dist(:,I)));dimX*2+1];
    if length(non_id)>2
        if non_id(2)==1
            non_id(1)=[];
        end
        if non_id(end-1)==nY
            non_id(end)=[];
        end
    end

    x=[(I-1)*T+(T-dx) I*T I*T (I-1)*T+(T-dx) (I-1)*T+(T-dx)];
    for J=1:length(non_id)-1
        y=[non_id(J) non_id(J) non_id(J+1)-1 non_id(J+1)-1 non_id(J)]*dy;
        if non_id(J)+1~=non_id(J+1)
            DXF_poly(fid,x,y,vertici,7,1),
        end
    end

end

DXF_end(fid),
```

----

dxf script for Matlab (DXF_start , DXF_poly , DXF_end) were available at:

http://school.mech.uwa.edu.au/~jamest/matlab-dxf/Default.html