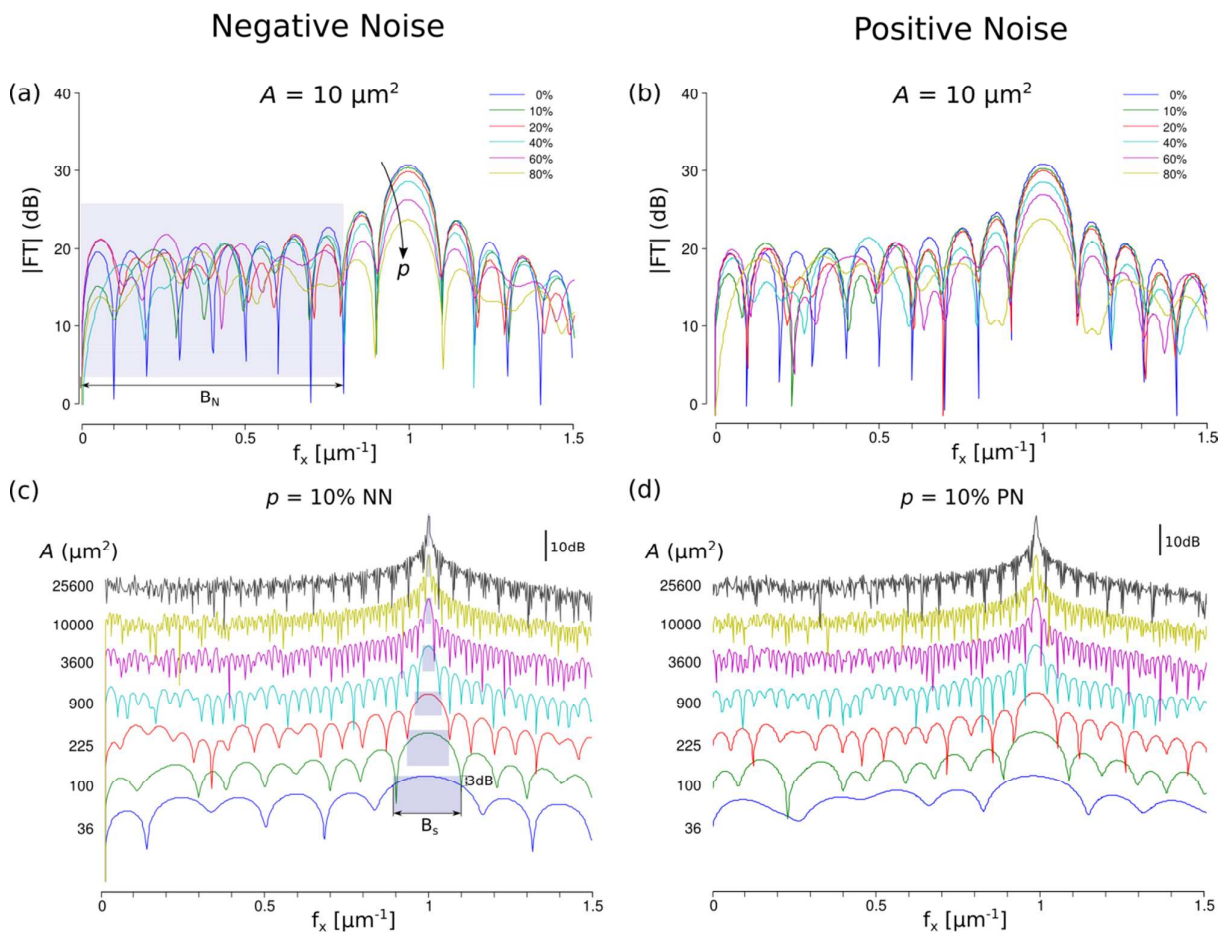


617 **SUPPLEMENTARY MATERIALS**

618

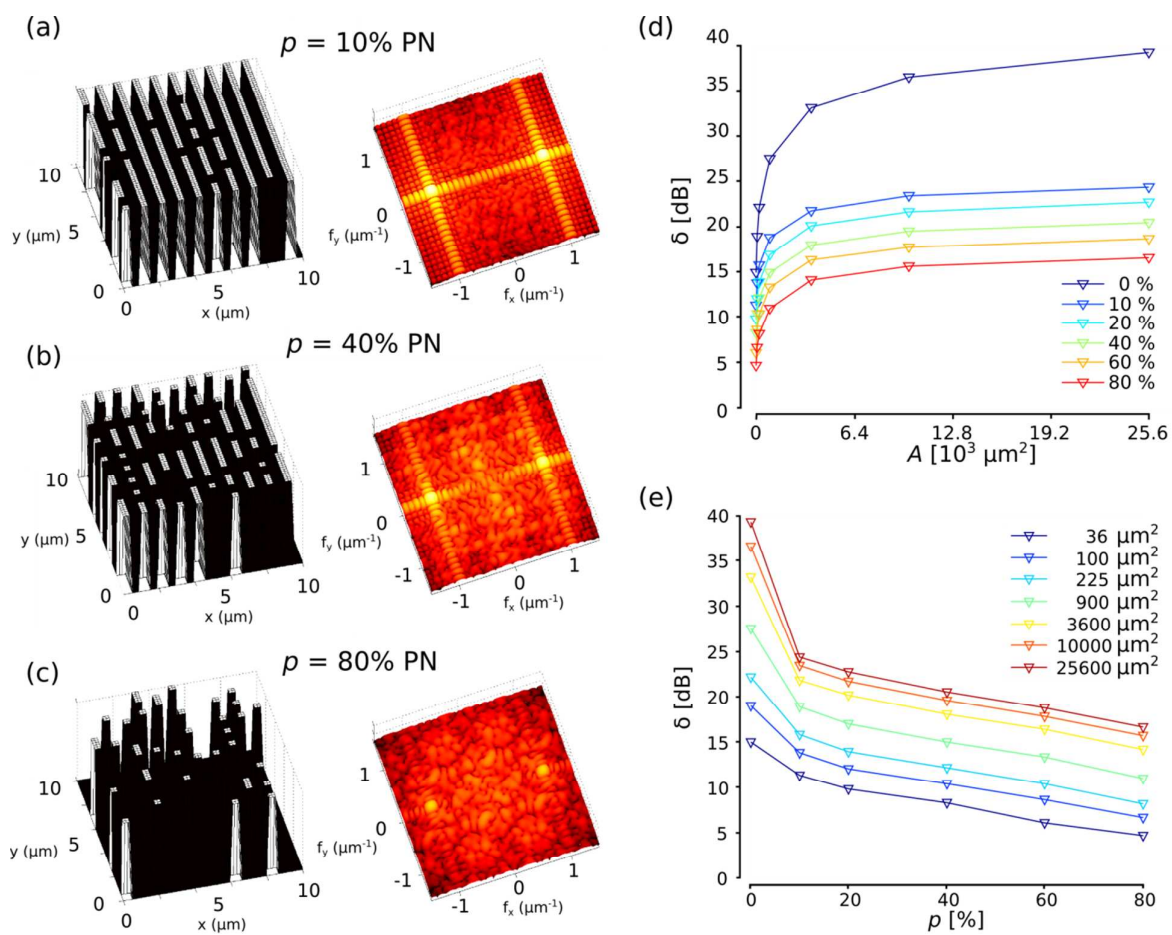


Supplementary Figure 1: FT spectra at $f_y = 0$ for different values of p (a,b) and A (c,d). (a) and (c) are calculated in case of NN. (b) and (d) are calculated in case of PN. The blue boxes in (a) and (c) represent B_n and B_s .

619

620

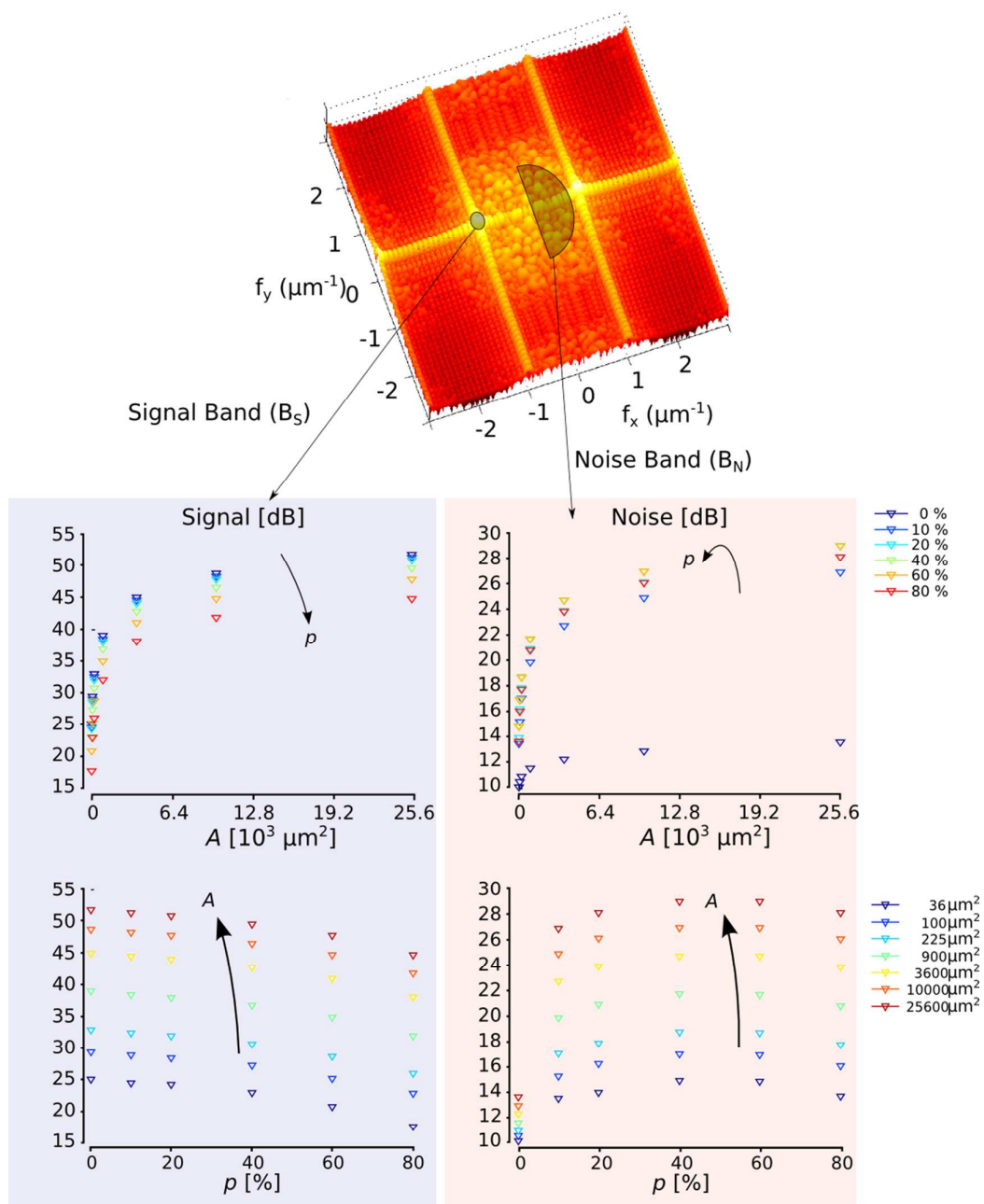
621



Supplementary Figure 2: Directionality quantification for NGs with PN. (a-c) $10 \mu\text{m} \times 10 \mu\text{m}$ 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).

622

623

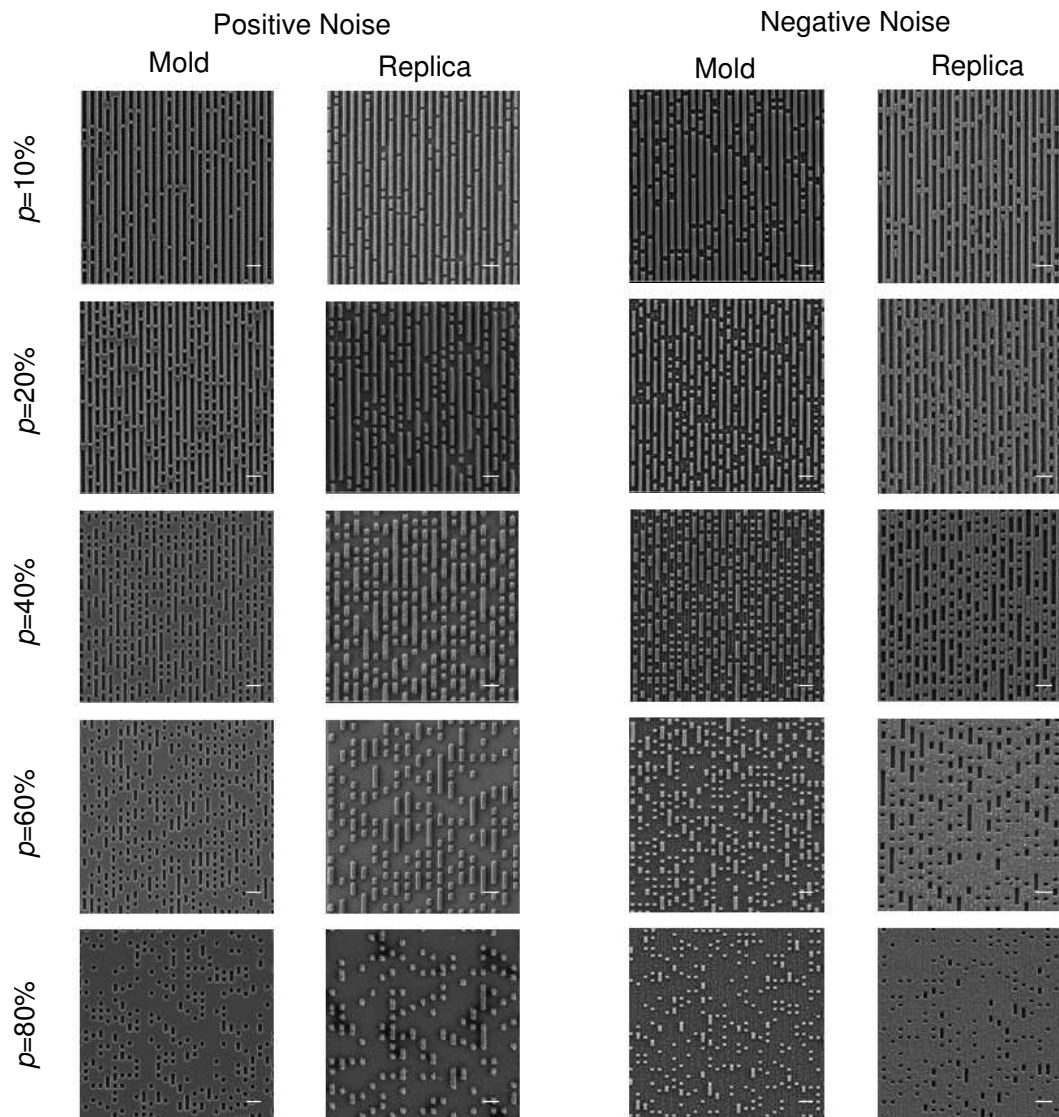


Supplementary Figure 3: Calculation of Signal (left column) and Noise (right column) as defined in Eq. 1 for different values of on A and p .

624

625

626



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 .

627

628

629 ***MATLAB scripts***

630 -----

631 %% noise generator

632 close all,clear all,clc

633

634 dimX=10;

635 dimY=10;

636

637 % period

638 T=1;

639 % duty cycle

640 DC=0.5;

641 %number of elements

642 nX=dimX/T;

643 nY=2*dimY;

644

645 % threshold

646 soglia=0.80;

647

648 terr=zeros(nY,nX);

649

650 nome=['noise',num2str((1-soglia)*100)];

651 app=zeros(nX,nY);

652 app=uint(app);

653 terr=app.MantBits;

654 clear app,

655 terr(:,(1:2:nX))=1;

656

657 imagesc((0:dimX/nX:dim),(0:dimY/nY:dim),terr),colormap copper,

658 xlabel('\mum'),ylabel('\mum'),

659

```
660 %% NEG noise generation
661 dist=rand(nX,nX/2);
662 T=0.90; %non ideality threshold
663 dist(dist>=soglia)=1;
664 dist(dist<soglia)=0;
665 figure,imagesc(dist),colormap copper,
666
667 terr(:,(1:2:nX))=not(dist);
668 imagesc((0:dimX/nX:dim),(0:dimY/nY:dim),terr),colormap copper,
669 xlabel('\mum'),ylabel('\mum'),
670 vertici=5;
671 x=[0 dimX dimX 0 0];
672 y=[0 0 dimY dimY 0];
673 DXF_poly(fid,x,y,vertici,5,1),
674
675
676 dx=dimX/nX*DC;
677 dy=0.5;
678
679 for I=1:nX
680     non_id=find((dist(:,I)));
681
682     x=[(I-1) (I-1)+DC (I-1)+DC (I-1) (I-1)]*T;
683
684     aperto=false;
685     for J=1:length(non_id)-1
686         if not(aperto)
687             yi=non_id(J);
688         end
689         if non_id(J)+1==non_id(J+1)
690             aperto=true;
```

```
691     else
692         yf=non_id(J);
693         y=[yi yi yf+1 yf+1 yi]*dy;
694         DXF_poly(fid,x,y,vertici,7,1),
695         aperto=false;
696     end
697
698 end
699
700     x=[(I-1)*T+(T-dx) I*T I*T (I-1)*T+(T-dx) (I-1)*T+(T-dx)];
701     y=[0 0 dimY dimY 0];
702     DXF_poly(fid,x,y,vertici,7,1),
703 end
704 DXF_end(fid),
705 fclose(fid),
706
707 %% POS noise generation
708 dist=rand(nX,nX/2);
709 dist(dist>=soglia)=1;
710 dist(dist<soglia)
711
712 [fid,err]=DXF_start([nome,'_pos.dxf'],10^scal);
713
714 dx=dimX/nX*DC; dy=0.5;
715 for I=1:nX
716     non_id=[0;find((dist(:,I)));dimX*2+1];
717     if length(non_id)>2
718         if non_id(2)==1
719             non_id(1)=[];
720         end
721         if non_id(end-1)==nY
```



```
722     non_id(end)=[];
723     end
724 end
725
726 x=[(I-1)*T+(T-dx) I*T I*T (I-1)*T+(T-dx) (I-1)*T+(T-dx)];
727 for J=1:length(non_id)-1
728     y=[non_id(J) non_id(J) non_id(J+1)-1 non_id(J+1)-1 non_id(J)]*dy;
729     if non_id(J)+1~=non_id(J+1)
730         DXF_poly(fid,x,y,vertici,7,1),
731     end
732 end
733
734 end
735
736 DXF_end(fid),
737 -----
738
739 dxf script for Matlab (DXF_start , DXF_poly , DXF_end) were available at:
740 http://school.mech.uwa.edu.au/~jamest/matlab-dxf/Default.html
741
742
743
```