# Supplementary Information: Algorithm for optimal de-noising of Raman spectra

## 1 MLE-SG MATLAB Code

These two function files are written for MATLAB software and can be copied and pasted into .m files. In order for them to run correctly the two files must be saved into the same folder as each other and the data that they are required to process. The code for this is presented in the following sections, as well as the commands to call the code from the MATLAB workspace.

### 1.1 Workspace Function Call

```
% Command to call the MLESG algorithm from the
% MATLAB workspace
% Please note that this function call uses the algorithms
% inbuilt parameters to process the spectra
m = calculatem(y,wavNo,pkLoc,gSig);
xe = MLESGcore(y, m);
figure, plot(wavNo, y, 'r')
hold
plot(wavNo, xe, 'g')
```

### 1.2 Early Stopping Procedure

Please note that the last 4 input parameters are to enable users to perform additional optimisation of the algorithms performance. If these inputs are not specified, then the algorithm will estimate the SNR automatically and use the inbuilt parameters specified for that particular SNR.

```
% y is the raw data vector, which is a function of sample
% index i
% wavNo is the axis for y in cm^-1
% m is the iterations vector, which is a function of sample
% index i
% pkLoc is a vector containing a list of peak locations in
% cm^-1
% gSig is the standard deviation of the Gaussian used in
% Eq.14 in paper, 10 is recommended
% SNR is the SNR of the signal itself
% minm and maxm are respectively the min and max numbers
% of iterations
% mu is the background signal vector (mean dark current +
% read noise)
function m = calculatem(y,wavNo,pkLoc,gSig,SNR,minm,maxm,mu)
% Auto-fit Gaussian curves of iterations to peak regions
% defined by the user
% Scale gSig to the wavenumber axis
gSig = gSig*(wavNo(2)-wavNo(1));
if nargin == 7
    mu=0;
end
if nargin == 6
    mu=0;
    SNR=0;
end

if nargin == 5
    mu=0;
    if SNR<=200
        p = [-0.000000000000020511637148964194179,
        0.0000000000174567386068190030 79,
        -0.0000000610837035038146155 5,
        0.000011326343084825261969,
        -0.0011967668932480550 99,
        0.072278932710701945807,
        -2.3738144356931862866,
        36.543962848162728108];

        minm = round(polyval(p,SNR));

        p2 = [0.00000004966205329646179 4872,
        -0.000045735657830153646274,
        0.013867582814278392803,
        -1.7581457324626106331,
        88.680082559339525284];

        maxm = round(polyval(p2,SNR));
    else
        minm = 2; maxm = 5;
    end
end

if nargin == 4
    % If SNR value is not given then estimate from raw
    mu=0;
    temp = sgolayfilt(y-mu,3,9); noise = y - mu - temp;
    clear temp;
    SNR = (max(y-mu)/(std(noise))); clear noise;
    if SNR<=200
        p = [ -0.000000000000020511637148964194179,
        0.000000000017456738606819003079,
        -0.00000006108370350381461555,
        0.000011326343084825261969,
```

```
        -0.001196766893248055099,
        0.072278932710701945807,
        -2.3738144356931862866,
        36.543962848162728108];

        minm = round(polyval(p,SNR));

        p2 = [0.000000049662053296461794872,
        -0.000045735657830153646274,
        0.013867582814278392803,
        -1.7581457324626106331,
        88.680082559339525284];

        maxm = round(polyval(p2,SNR));
    else
        minm = 2; maxm = 5;
    end
end

x=y-mu; clear y; clear mu;
N = length(wavNo);
G = zeros(length(pkLoc),N);

for i=1:1:length(pkLoc)
    G(i,:) = 1 -
    exp(-(power(wavNo-pkLoc(i),2))/(2*power(gSig,2)));
end

% Calculating minimum number of iterations along vertical
m=min(G);
% Scaling with respect to minimum and maximum values m
% can have
m=m*(maxm - minm) + minm;
end
```

### 1.3 Core MLESG Algorithm

Please note that the input parameters lambda, p, v, q, and mu are to enable users to perform additional optimisation of the algorithms performance. If these inputs are not specified, then the algorithm will utilise the inbuilt parameters available to it.

```
% y is the raw data vector, which is a function of sample
% index i
% m is the iterations vector, which is a function of sample
% index i
% lambda and p, default values of 1.8 and 0.4
% v and q are SG papramaters and have default values of 5
% and 7
% mu is the background signal vector (mean dark current +
% read noise)

function xe = MLESGcore(y, m, v, q, lambda, p, mu)

if nargin == 2
    v=5;q=7;lambda=1.8;p=0.4;mu=0;
elseif nargin == 4
    lambda=1.8;p=0.4;mu=0;
elseif nargin == 6
    mu=0;
end

%Ensure m contains integer values
m=round(m);

% Determine first guess for the algorithm
x=y-mu; clear y; clear mu;
N = length(x);

% An estimate is required for the search window in which to
% calculate the MLE; this estimate is based on the
% approximate value for the standard deviation of the noise
% terms as calculated here.
% The range is equal to 6 times this value.
temp = sgolayfilt(x,3,9); noise = x - temp; sigma = std(noise);
clear temp; clear noise;

% Apply MLE
for j = 1 : 1 : max(m)

    if(j <= min(m))
        v =5;q =7;
    elseif (j>max(m)-round(max(m)/5))
        q = q+4; lambda=lambda*10;
    else
        v=3;q =5;
    end

    % First guess
    if j==1
        xe=x;
    end
    % Note this is denoted as x’ in the paper and represents
    % the smoothed signal used within the MLE estimation
    xdash = sgolayfilt(xe, v, q);

    % Apply MLE to each sample index
    for i = 1 : 1 : N
        % Only apply MLE if iteration number for this sample
        % index is greater than the current iteration number
        if m(i) >= j
            % This is the vector of possible xe values we apply
            % MLE over
```

```
            MLErange =
            (xe(i)-3*sigma:6*sigma/100:xe(i)+3*sigma);
            % Minimise the likelihood function (lFunc)
            % This will be assigned the likelihood estimations
            % for each value of x in the range of values
            % MLErange
            leMLErange = zeros(size(MLErange));
            for k = 1 : 1 : length(MLErange)
                limit1 = lambda*
                (power(abs(MLErange(k)-xdash(i)),p));
                limit2 = power(x(i) -
                MLErange(k),2)/(2*power(sigma,2));
                % Calculate likelihood estimation (le)
                % for each value in range for sample i
                leMLErange(k) = limit1 + limit2;
            end
            [a, b] = min(leMLErange);
            % MLE for sample i in iteration j
            xe(i) = MLErange(b);
        end

    end
end
end
```

## 2 Early Stopping Optimisation

The following section provides details of the optimisation process of the proposed algorithm in terms of Signal to Noise Ratio (SNR), as defined in the main body of the paper. These optimisation processes take place after the optimisation of the input parameters $\lambda$ and $p$. In this case the focus is placed on the Savitsky-Golay (SG) input parameters and the number of iterations required by the algorithm to achieve the maximum SNR for both the whole spectrum and a peak of interest. The peak was chosen as a region of interest due to it's narrow width and relatively low intensity which renders it a challenge to preserve effectively. Optimisation processes were performed on artificial plastic datasets with a range of different input SNRs. Results shown are a mean of 100 individual spectra which were denoised using the algorithm. These graphs are where the optimum values for $m_{max}$ and $m_{min}$ were derived for three sets of SG input parameters. For aesthetic purposes the number of iterations is limited to 50 in the graphs, since for the majority of datasets the optimal number of iterations is under 50. However, for the lower SNR datasets the number of iterations was increased until the increase in SNR for the de-noised spectra had stabilised or reached it's maximum, the final decision for $m_{max}$ is provided in the caption for these images. Please note that the iteration curves provided in Fig. 4, of the paper, were derived from the curves associated with SG(5,7), which are illustrated in the following figures.

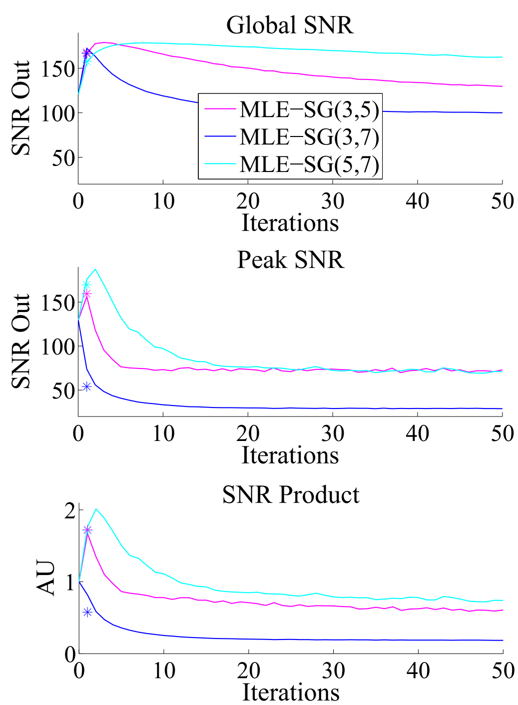


**Fig. 1** Initial SNR: 120

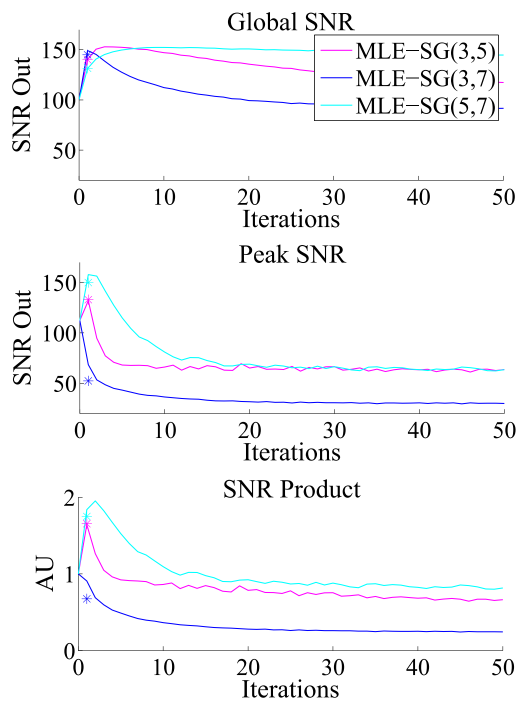


**Fig. 2** Initial SNR: 100

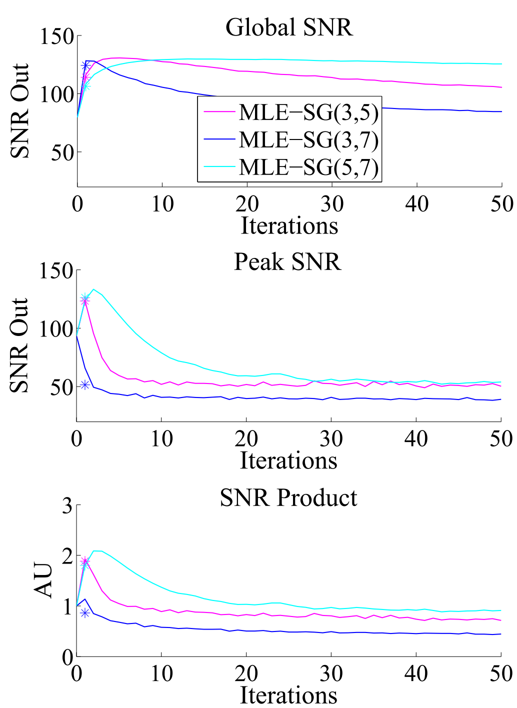


**Fig. 3** Initial SNR: 80

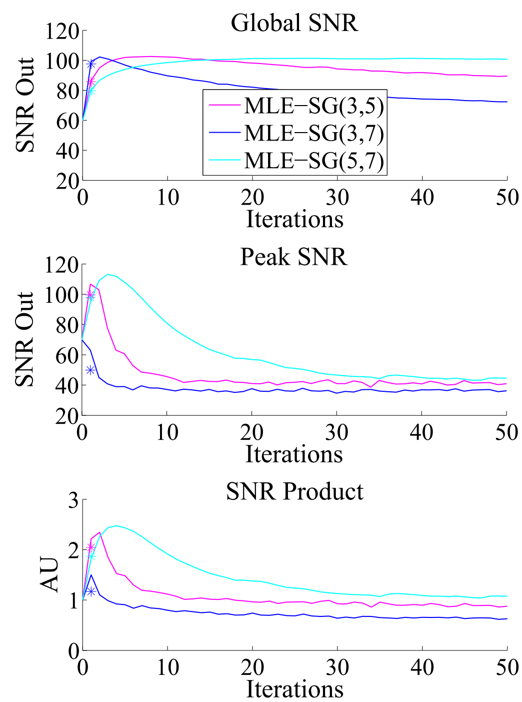


**Fig. 4** Initial SNR: 60

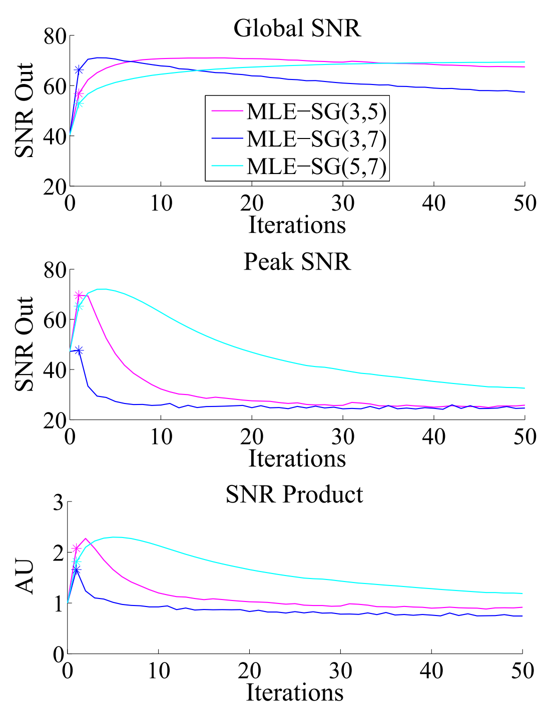


**Fig. 5** Initial SNR: 40

Optimal number for iterations for an initial SNR of 40 is 54.

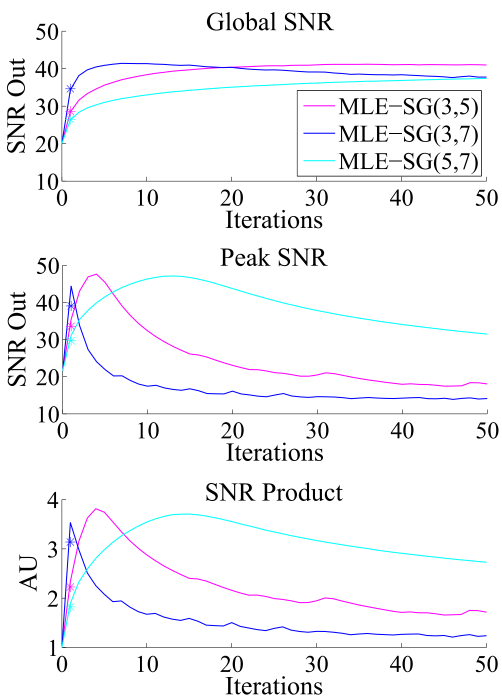


**Fig. 6** Initial SNR: 20

Optimal number for iterations for an initial SNR of 20 is 100.

## 3 Tables of Quantitative Values

In this section the quantitative results correspond to Fig. 7(a), Fig. 7(b), and Fig. 8 in the main body of the text are presented. These results clearly demonstrate enhancement oin SNR achieved by the proposed denoising algorithm compared to the two alternative methods examined in the paper. The value of the SNR is presented for each of the denoising methods. The metrics of Global SNR, Peak SNR, and the SNR product have all been calculated separately using the mathematical definitions provided in Equations 10, 11, and 12 respectively in the main body of the paper. Each value given in the tables is the average value calculated over a dataset of 100 similarly noised spectra.

| | Raw Data | MLE-SG Data | SG(3,7) Data | Perfect Smoother |
|---|---|---|---|---|
| Global SNR | 20.05 | 47.08 | 34.77 | 40.67 |
| | 30.07 | 64.90 | 50.58 | 55.10 |
| | 40.09 | 83.72 | 66.60 | 69.53 |
| | 49.79 | 101.34 | 81.83 | 83.73 |
| | 59.70 | 118.81 | 96.23 | 97.72 |
| | 72.25 | 140.88 | 114.46 | 115.93 |
| | 79.66 | 149.58 | 122.72 | 125.12 |
| | 91.67 | 165.90 | 136.53 | 140.61 |
| | 99.58 | 174.35 | 145.16 | 150.90 |
| | 109.11 | 190.49 | 156.62 | 164.38 |
| | 120.42 | 208.94 | 166.89 | 179.08 |
| | 130.82 | 221.94 | 175.39 | 191.92 |
| | 139.78 | 233.17 | 181.59 | 203.28 |
| Peak SNR | 21.65 | 40.47 | 35.41 | 15.84 |
| | 33.74 | 55.19 | 43.10 | 23.55 |
| | 44.99 | 70.12 | 47.04 | 31.64 |
| | 60.96 | 90.47 | 48.78 | 38.22 |
| | 72.50 | 101.86 | 49.73 | 45.31 |
| | 78.18 | 115.15 | 50.74 | 61.23 |
| | 97.53 | 128.54 | 51.24 | 65.94 |
| | 97.71 | 140.68 | 52.05 | 73.99 |
| | 104.36 | 146.95 | 51.40 | 75.90 |
| | 118.70 | 162.36 | 52.08 | 90.27 |
| | 130.30 | 180.20 | 52.35 | 114.81 |
| | 143.08 | 188.77 | 52.87 | 123.18 |
| | 153.78 | 198.30 | 53.22 | 127.84 |
| SNR Product | 1 | 4.50 | 2.93 | 1.58 |
| | 1 | 3.48 | 2.24 | 1.38 |
| | 1 | 3.26 | 1.84 | 1.31 |
| | 1 | 3.16 | 1.42 | 1.14 |
| | 1 | 2.95 | 1.20 | 1.11 |
| | 1 | 2.94 | 1.09 | 1.32 |
| | 1 | 2.61 | 0.88 | 1.15 |
| | 1 | 2.65 | 0.84 | 1.23 |
| | 1 | 2.53 | 0.76 | 1.15 |
| | 1 | 2.47 | 0.67 | 1.21 |
| | 1 | 2.44 | 0.59 | 1.38 |
| | 1 | 2.28 | 0.52 | 1.32 |
| | 1 | 2.19 | 0.48 | 1.29 |

**Table 1** A table of the values of the SNR calculated based on the simulated plastic datasets for the three denoising methods using the three SNR metrics described in the main body of the paper. Corresponds to Fig. 7(a).

| | Raw Data | MLE-SG Data | SG(3,7) Data | Perfect Smoother |
|---|---|---|---|---|
| Global SNR | 25.41 | 54.51 | 43.15 | 49.21 |
| | 39.55 | 76.73 | 65.46 | 69.03 |
| | 50.48 | 93.99 | 82.28 | 84.24 |
| | 61.56 | 110.40 | 98.04 | 99.69 |
| | 70.44 | 122.73 | 110.66 | 111.68 |
| | 78.38 | 132.27 | 121.47 | 122.19 |
| | 86.51 | 143.70 | 132.74 | 134.21 |
| | 94.42 | 151.43 | 141.50 | 144.86 |
| | 102.95 | 162.12 | 152.46 | 155.56 |
| | 110.52 | 175.34 | 161.24 | 165.56 |
| | 117.26 | 184.77 | 167.35 | 174.31 |
| | 122.74 | 191.63 | 173.49 | 181.21 |
| | 128.74 | 200.17 | 180.24 | 190.06 |
| | 132.74 | 206.34 | 183.28 | 194.39 |
| | 140.96 | 217.59 | 191.57 | 206.19 |
| Peak SNR | 24.84 | 49.73 | 41.92 | 20.12 |
| | 33.12 | 65.13 | 46.58 | 29.60 |
| | 42.79 | 67.25 | 45.22 | 36.08 |
| | 50.36 | 83.42 | 50.80 | 47.80 |
| | 59.41 | 88.45 | 55.51 | 56.23 |
| | 61.90 | 99.63 | 55.88 | 61.16 |
| | 72.48 | 100.52 | 55.92 | 67.09 |
| | 73.44 | 108.90 | 55.59 | 73.11 |
| | 77.63 | 107.43 | 55.73 | 82.97 |
| | 87.85 | 127.07 | 58.63 | 94.73 |
| | 91.44 | 132.82 | 55.92 | 102.23 |
| | 93.55 | 133.88 | 57.81 | 94.35 |
| | 99.21 | 146.91 | 60.09 | 118.80 |
| | 99.38 | 135.94 | 58.57 | 112.49 |
| | 118.04 | 168.48 | 58.74 | 124.92 |
| SNR Product | 1 | 4.26 | 2.88 | 1.63 |
| | 1 | 3.93 | 2.43 | 1.64 |
| | 1 | 2.99 | 1.78 | 1.46 |
| | 1 | 2.94 | 1.67 | 1.59 |
| | 1 | 2.68 | 1.63 | 1.65 |
| | 1 | 2.77 | 1.47 | 1.61 |
| | 1 | 2.33 | 1.26 | 1.52 |
| | 1 | 2.43 | 1.21 | 1.63 |
| | 1 | 2.17 | 1.12 | 1.67 |
| | 1 | 2.32 | 1.02 | 1.66 |
| | 1 | 2.32 | 0.93 | 1.76 |
| | 1 | 2.24 | 0.93 | 1.55 |
| | 1 | 2.31 | 0.90 | 1.81 |
| | 1 | 2.14 | 0.86 | 1.73 |
| | 1 | 2.23 | 0.73 | 1.62 |

**Table 2** A table of the SNR values calculated based on the experimental plastic datasets for the three denoising methods using the three SNR metrics described in the main body of the paper. Corresponds to Fig. 7(b).

| | Raw Data | MLE-SG Data | SG(3,7) Data | Perfect Smoother |
|---|---|---|---|---|
| Global SNR | 19.03 | 42.58 | 32.71 | 41.43 |
| | 31.98 | 63.07 | 53.91 | 60.69 |
| | 38.25 | 74.17 | 63.73 | 69.85 |
| | 47.75 | 87.80 | 79.01 | 83.78 |
| | 56.39 | 95.29 | 91.15 | 94.42 |
| | 68.42 | 118.58 | 107.84 | 109.28 |
| | 79.83 | 136.76 | 122.15 | 123.16 |
| | 87.58 | 148.70 | 130.91 | 132.74 |
| | 95.33 | 160.07 | 139.45 | 141.97 |
| | 106.43 | 171.97 | 150.45 | 155.34 |
| | 113.45 | 171.34 | 156.24 | 162.44 |
| | 125.73 | 188.91 | 166.63 | 177.33 |
| | 134.51 | 201.01 | 173.19 | 186.50 |
| | 146.21 | 213.07 | 180.95 | 202.04 |
| | 153.76 | 216.01 | 184.38 | 207.15 |
| | 163.31 | 235.19 | 190.65 | 218.16 |
| | 171.43 | 245.32 | 195.91 | 228.00 |
| | 183.99 | 255.30 | 201.73 | 240.17 |
| | 190.86 | 261.84 | 203.76 | 245.94 |
| Peak SNR | 23.37 | 32.65 | 27.79 | 14.79 |
| | 34.81 | 49.66 | 32.12 | 18.93 |
| | 42.35 | 60.27 | 33.80 | 21.92 |
| | 54.10 | 67.40 | 33.47 | 22.89 |
| | 61.01 | 82.75 | 34.55 | 25.94 |
| | 78.23 | 97.36 | 34.60 | 29.25 |
| | 88.43 | 107.57 | 35.53 | 33.77 |
| | 100.03 | 107.78 | 35.35 | 37.77 |
| | 111.43 | 119.77 | 35.20 | 41.96 |
| | 119.32 | 133.85 | 35.30 | 45.27 |
| | 131.53 | 139.37 | 35.24 | 49.09 |
| | 130.34 | 147.12 | 35.29 | 60.04 |
| | 150.08 | 162.68 | 35.51 | 60.96 |
| | 166.51 | 176.44 | 35.43 | 66.66 |
| | 173.27 | 184.73 | 35.41 | 72.62 |
| | 191.48 | 196.19 | 35.30 | 72.27 |
| | 187.99 | 195.95 | 35.56 | 78.38 |
| | 211.02 | 218.31 | 35.72 | 79.74 |
| | 207.38 | 222.73 | 35.47 | 86.14 |
| SNR Product | 1 | 3.35 | 2.25 | 1.56 |
| | 1 | 2.93 | 1.65 | 1.11 |
| | 1 | 2.87 | 1.44 | 1.03 |
| | 1 | 2.39 | 1.08 | 0.79 |
| | 1 | 2.34 | 0.98 | 0.76 |
| | 1 | 2.21 | 0.75 | 0.65 |
| | 1 | 2.18 | 0.66 | 0.63 |
| | 1 | 1.93 | 0.58 | 0.63 |
| | 1 | 1.89 | 0.50 | 0.60 |
| | 1 | 1.91 | 0.45 | 0.59 |
| | 1 | 1.66 | 0.41 | 0.58 |
| | 1 | 1.69 | 0.38 | 0.67 |
| | 1 | 1.65 | 0.33 | 0.61 |
| | 1 | 1.57 | 0.29 | 0.60 |
| | 1 | 1.53 | 0.27 | 0.63 |
| | 1 | 1.52 | 0.24 | 0.56 |
| | 1 | 1.52 | 0.23 | 0.60 |
| | 1 | 1.46 | 0.20 | 0.53 |
| | 1 | 1.47 | 0.20 | 0.57 |

**Table 3** A table of the SNR values calculated based on the simulated T24 datasets for the three denoising methods using the three SNR metrics described in the main body of the paper. Corresponds to Fig. 8