Supplementary Information For: Electronic Dynamics in Natural Iron Pyrite Studied by Broadband Transient Reflection Spectroscopy

Shayne A. Sorenson, Joel G. Patrow, and Jahan M. Dawlaty*

Department of Chemistry, University of Southern California

E-mail: dawlaty@usc.edu

*To whom correspondence should be addressed

Crystal Characterization

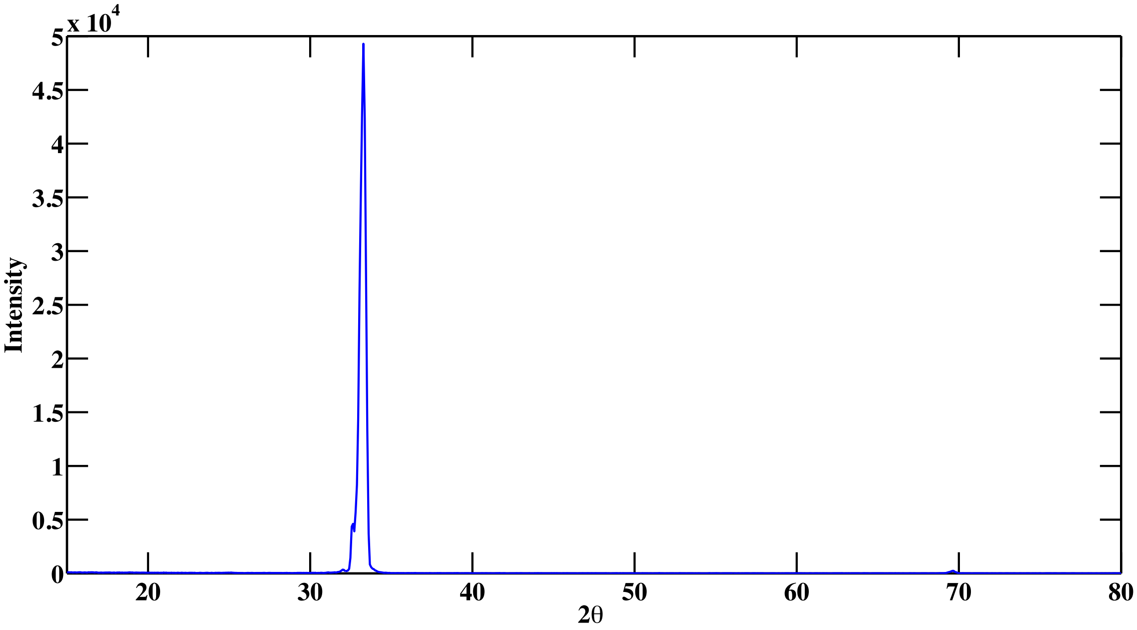

Figure S1: X-Ray Diffraction was first carried out on the same face of the single crystal as was used for the transient reflection tests. There is one prominent peak with its maximum at $2\theta = 33.3°$, which corresponds to a d-spacing of 2.689 Å. Using the lattice parameter from Liu of $a = 5.417$ Å, this corresponds to $\mathrm{h}^2 + \mathrm{k}^2 + l^2 = 4$ and is assigned to be from the {200} plane.[1]

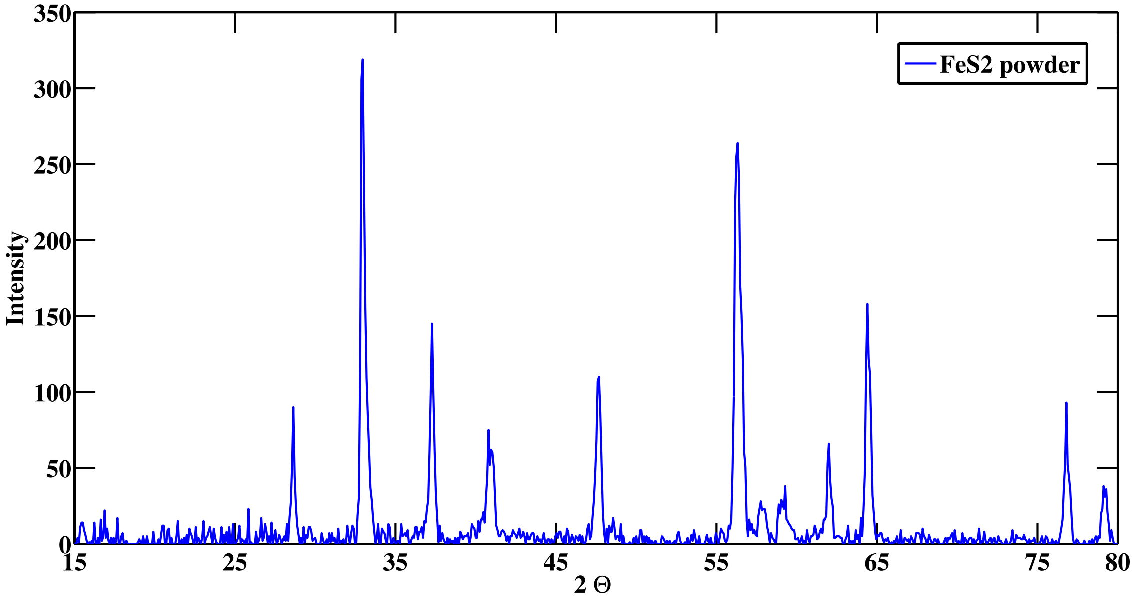

Figure S2: Powder diffraction was also performed using a small portion of the same crystal ground by mortar and pestle. All of the peaks are in agreement with literature values.[1]

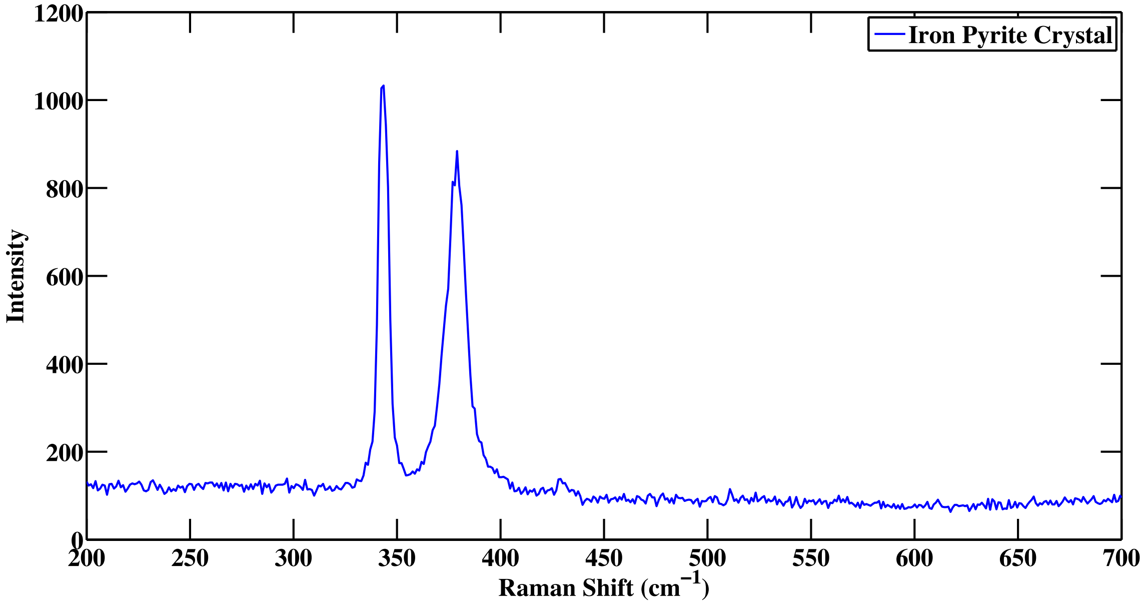

Figure S3: The Raman spectrum was obtained using a 532 nm excitation laser with no polarizer on the scattered light. The peak maxima are at 343.5, 379 and 433 cm^{-1} and are in excellent agreement with literature values for phase pure iron pyrite.[2] No marcasite peaks are detected.

Transformation Derivation Details

Including the effects of the incidence angle in the derivation is quite straight forward. First, note that the full expression for reflectance depends on polarization, such that there are two expressions (maintaining the assumption that one of the media is air):

$$R_S = \left| \frac{A - (n + i\kappa)\, B}{A + (n + i\kappa)\, B} \right|^2 , \tag{1}$$

and

$$R_P = \left| \frac{B - (n + i\kappa)\, A}{B + (n + i\kappa)\, A} \right|^2 . \tag{2}$$

Here, $A = \cos\theta_i$, $B = \sqrt{1 - \left(\frac{1}{n}\sin\theta_i\right)}$ and θ_i is the angle with respect to the surface normal of the incident beam. Expressing the change in reflectance as a Taylor series expansion can still be done, giving:

$$\frac{\Delta R}{R} = \frac{1}{R}\left[\left(\frac{\partial R}{\partial \kappa}\right)_n \Delta\kappa + \left(\frac{\partial R}{\partial n}\right)_\kappa \Delta n\right]. \tag{3}$$

Where now $\left(\frac{\partial R}{\partial \kappa}\right)_n$ and $\left(\frac{\partial R}{\partial n}\right)_\kappa$ have a different, slightly more complicated form when the angle dependent factors are included. Whether to include the factors dependent on incidence angle or not depends on the system studied (ranges of n and κ and the magnitude of the measured transients $\Delta R/R$) as well as the incidence angles used; either way, however, the necessary derivatives can be computed analytically and the overall transformation remains tractable. For the measurements reported throughout this publication, the incidence angle of the probe beam was about 3.6°, and the polarization was parallel with the plane of incidence (P). To assess the validity of neglecting the deviation from normal incidence we compute the necessary multiplicative factors as follows:

$$\left(\frac{1}{R}\frac{\partial R}{\partial \kappa}\right)(n,\kappa;theta) = \frac{8n\kappa B^3 A}{\left(A^2+n^2B^2+\kappa^2B^2\right)^2 - 4n^2B^2A^2} \tag{4}$$

and

$$\left(\frac{1}{R}\frac{\partial R}{\partial n}\right)(n,\kappa;theta) = \frac{4B^3 A\left(n^2-\kappa^2-\frac{A^2}{B^2}\right)}{\left(A^2+n^2B^2+\kappa^2B^2\right)^2 - 4n^2B^2A^2}. \tag{5}$$

Using these expressions in the transformation derived in the text, allows for any incidence angle without approximation. Each of these factors in eqs. 4 and 5 are parameterized by the incidence angle θ via A and B. A simple, albeit brute force, test of validity of our neglect of incidence angle dependence would be to calculate these quantities throughout the range of n and κ values relevant to pyrite at both $\theta = 0$ and $\theta = 0.0625$ and examine the difference. Figure S4 shows this difference as a percentage of the actual value of the factor (i.e. $\left[\left(\frac{1}{R}\frac{\partial R}{\partial \kappa}\right)(n,\kappa;0) - \left(\frac{1}{R}\frac{\partial R}{\partial \kappa}\right)(n,\kappa;0.0625)\right] / \left(\frac{1}{R}\frac{\partial R}{\partial \kappa}\right)(n,\kappa;0.0625)$). For our experiment, it is clear that the corrections to the terms in the transformation arising from the small deviation from normal incidence are three orders of magnitude smaller than the terms themselves and therefore safely neglected.

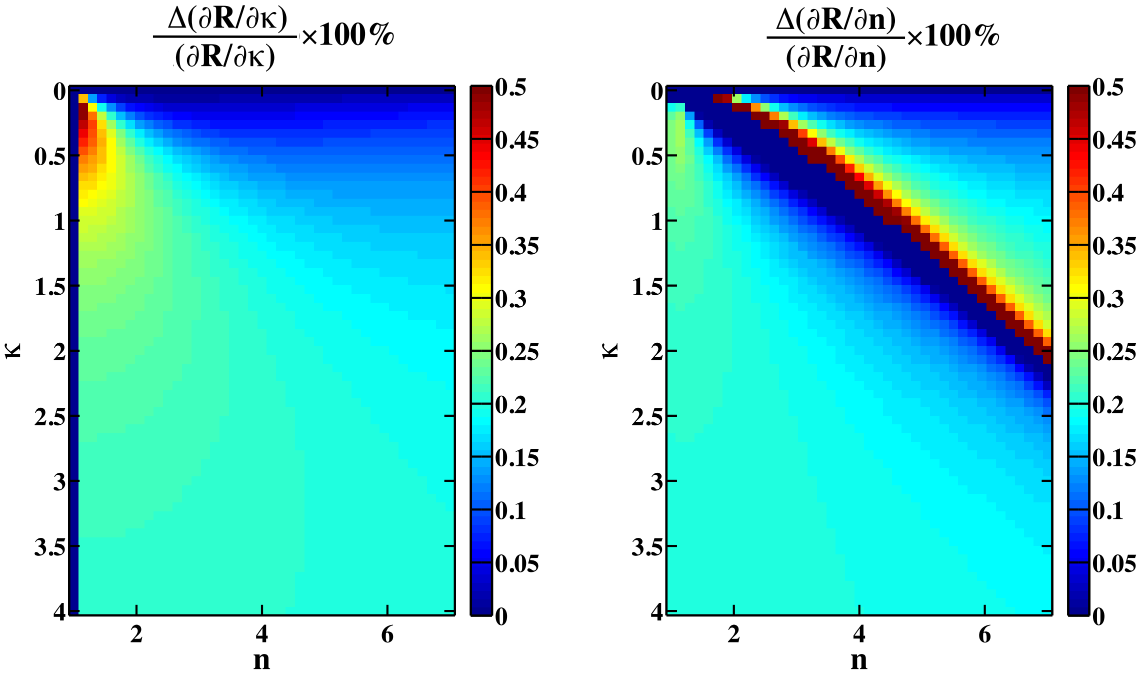

Figure S4: The difference between the multiplicative terms involved in the transformation ($\left(\frac{1}{R}\frac{\partial R}{\partial \kappa}\right)$ on the left and $\left(\frac{1}{R}\frac{\partial R}{\partial n}\right)$ on the right), calculated at $\theta = 0$ and $\theta = 0.0625$. The differences are divided by the factors calculated at $\theta = 0.0625$ to inform us of their relative contribution to the transformation and expressed as a percentage. In each case we see that the relative correction introduced by the deviation from normal incidence is on the order of 10^{-3} and can be safely neglected.

Fourier Analysis

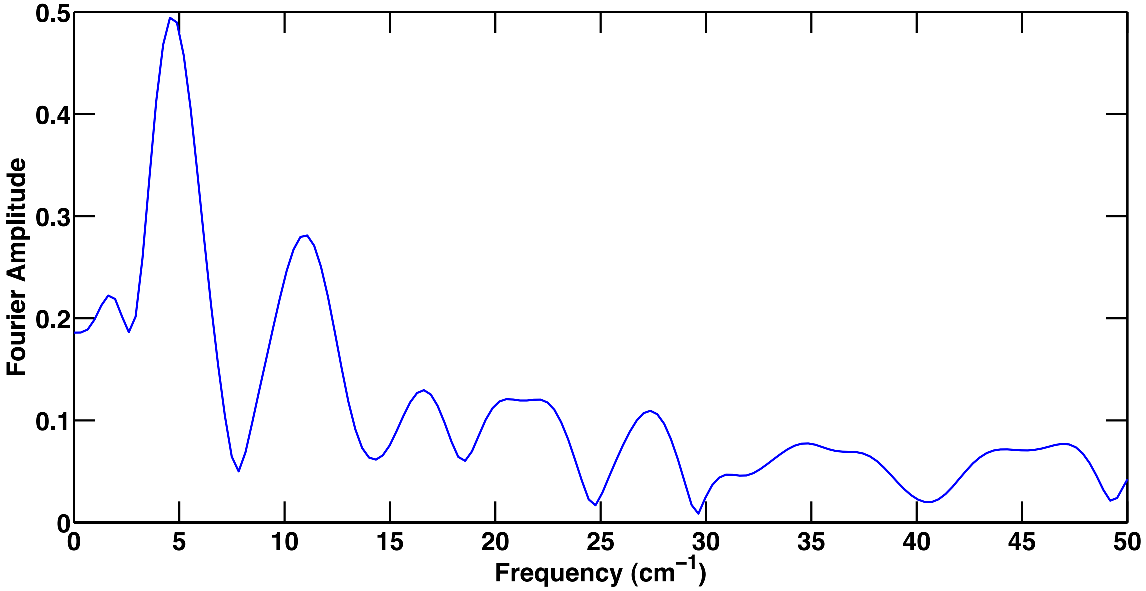

Figure S5: The Fourier amplitude of the 2.64 eV pump data at a single probe wavelength (1.43 eV) is shown as a representative slice of the oscillations present in the low energy probe region. The peak at 4.5 cm^{-1} corresponds to a period of 7.3 ps, and has been measured in pyrite before.[3] The peak at 11.08 cm^{-1} is present only in the 2.64 eV pump data and may be a noise component of the pump laser itself.

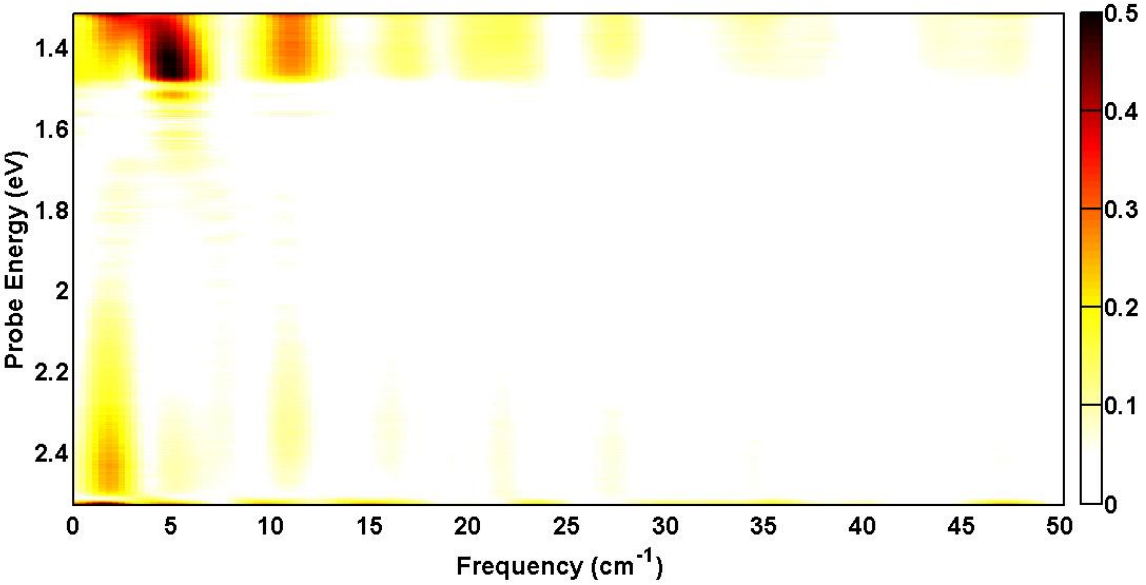

Figure S6: Fourier transform of the full two dimensional 2.64 eV pump data. Fourier analysis can be applied to every probe wavelength. Doing so allows us to see dispersion of the acoustic phonon - a slight change in the phonon frequency as a function of probe wavelength. Attempts at extracting the complex index of refraction from the fitted frequency and damping constant of the acoustic phonon, based on Kasami's *et. al* analysis were unsuccessful. We believe that our signal to noise ratio is not good enough to determine the damping constant to the needed precision.

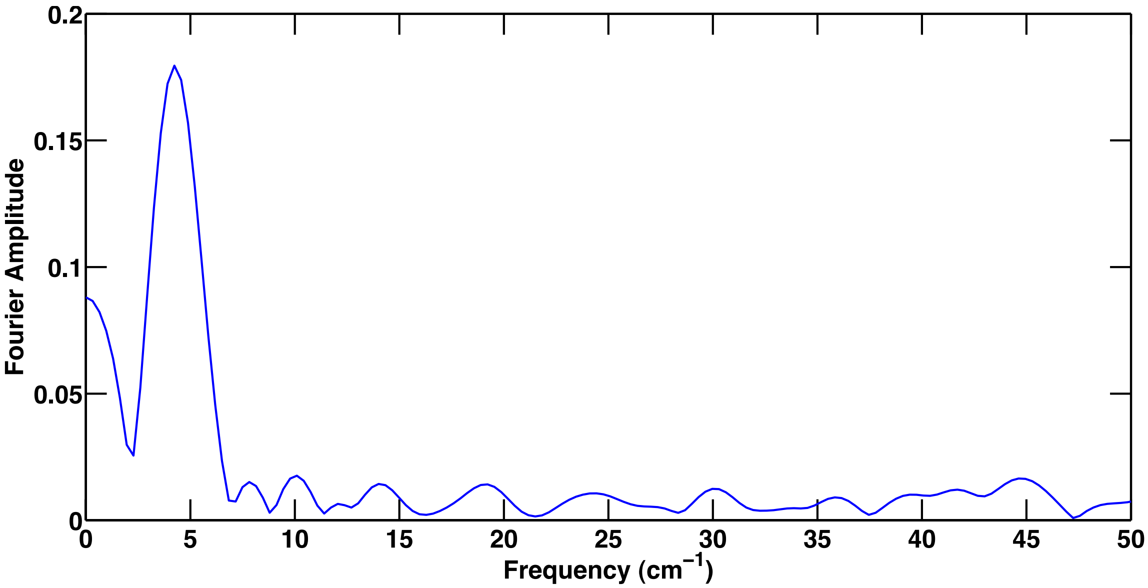

Figure S7: In the case of the 1.55 eV pump data, only the feature around 4.5 cm^{-1} appears.

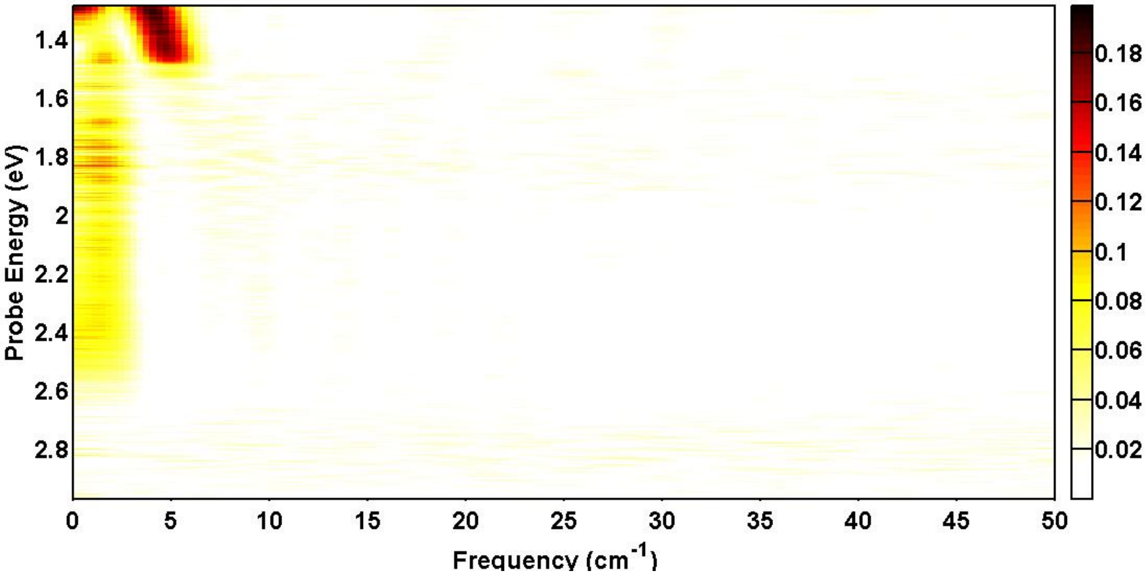

Figure S8: The full Fourier transform of the 1.55 eV pump data.

Details on the Transformation from Reflection to Absorption

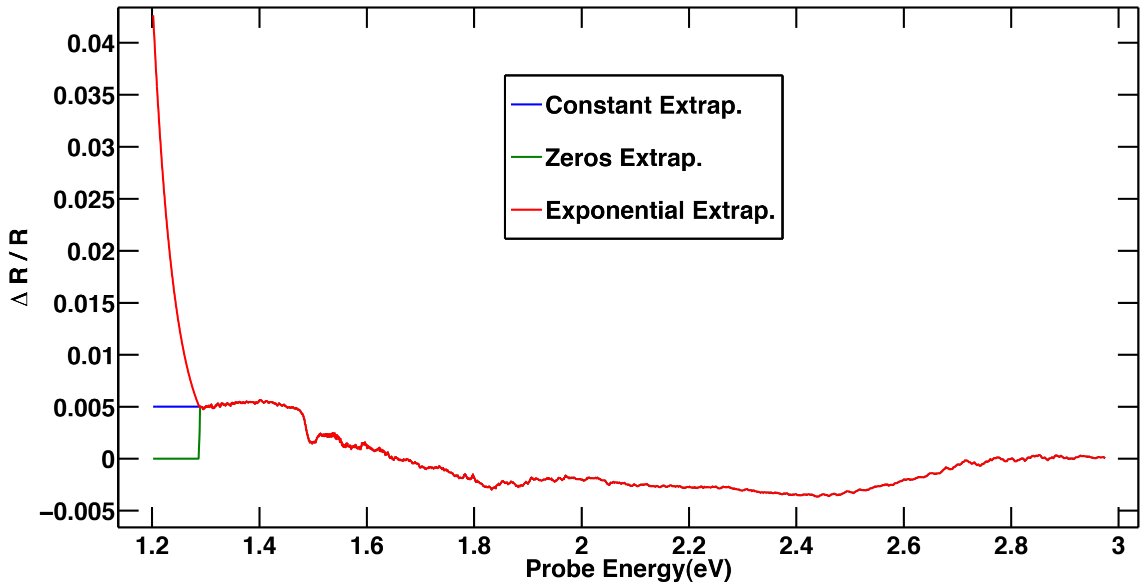

Figure S9: The measured transient reflection spectrum in the case of 1.55 eV pump and at a delay of 2 ps. The low energy end of the transient spectrum was extrapolated using three different schemes. The blue shows the constant extrapolation, where the spectrum extended at a constant value found by averaging 10 points at the edge of the measured spectrum. The green trace shows the zero extrapolation, where all extrapolated points were zero. The red shows an exponential extrapolation, where the data increased exponentially starting from the value at the edge of the measured spectrum - the goal here was to simulate a worst case scenario. Each of these was then transformed according to the procedure described in the text and the results are shown in the next figure.

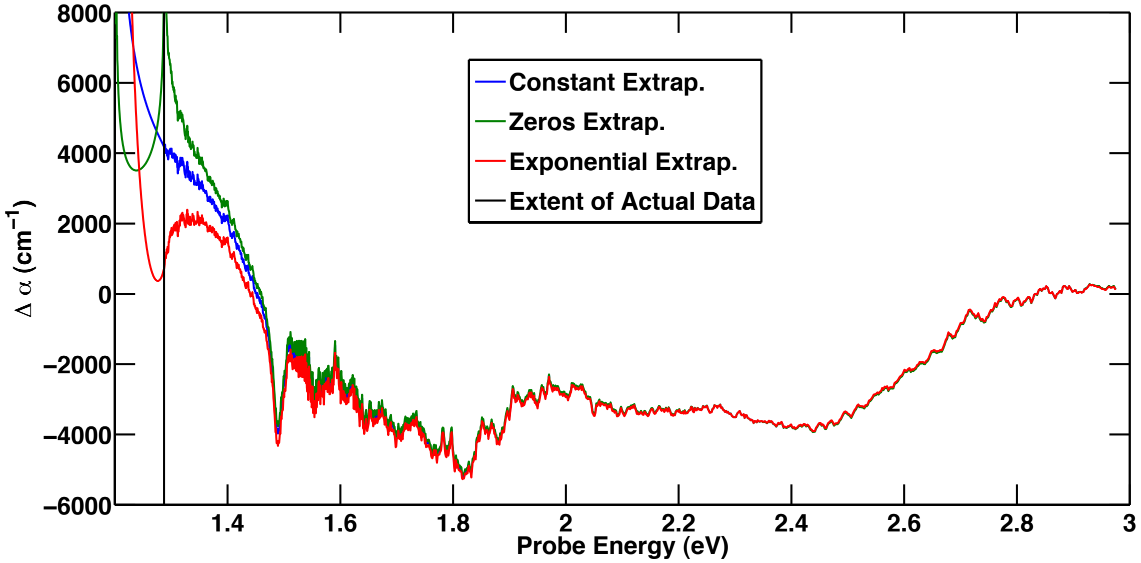

Figure S10: The results of the transformation of the various extrapolations. The vertical black line shows the limit of the experimentally measured data. You can see that the values of the transient reflection spectrum outside of our measurement window affect the spectrum quantitavely as far as 0.2 eV from the edge in the so-called worst case scenario of the exponential extrapolation. Qualitatively however, the transformed spectrum of the worst case scenario is fairly good as close as 0.1 eV from the edge. Notice that a similar situation exists at the high energy end of the probe. Though it looks like the transient spectrum smoothly goes to zero this is actually an artifact of the probe intensity dying there - in reality we have terminated the transient spectrum around 2.7 eV and so our spectra should only be interpreted between 1.35 and 2.6 eV.

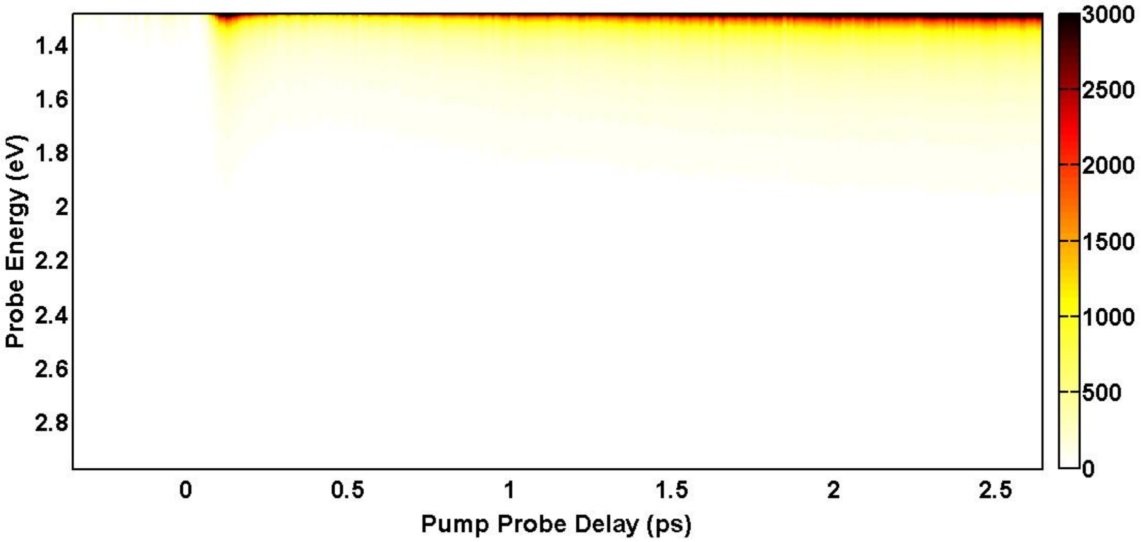

Figure S11: The same transformation procedure using the three extrapolations was performed at every probe wavelength and the the sum of the squares of all the possible differences between the resulting transformed spectra was computed for each point of the two dimensional data. For example, if the two dimensional data for each extrapolation was transformed to absorption and stored as a matrix (R^{expo}, R^{zero} and R^{cons}). The matrix shown here is: $dev_{ij} = (R_{ij}^{expo} - R_{ij}^{zero})^2 + (R_{ij}^{zero} - R_{ij}^{cons})^2 + (R_{ij}^{cons} - R_{ij}^{expo})^2$. As you can see the differences introduced by the extrapolations outside of our experimental window are limited mainly to within 0.05-0.1 eV from the edge.

Fluence Dependence

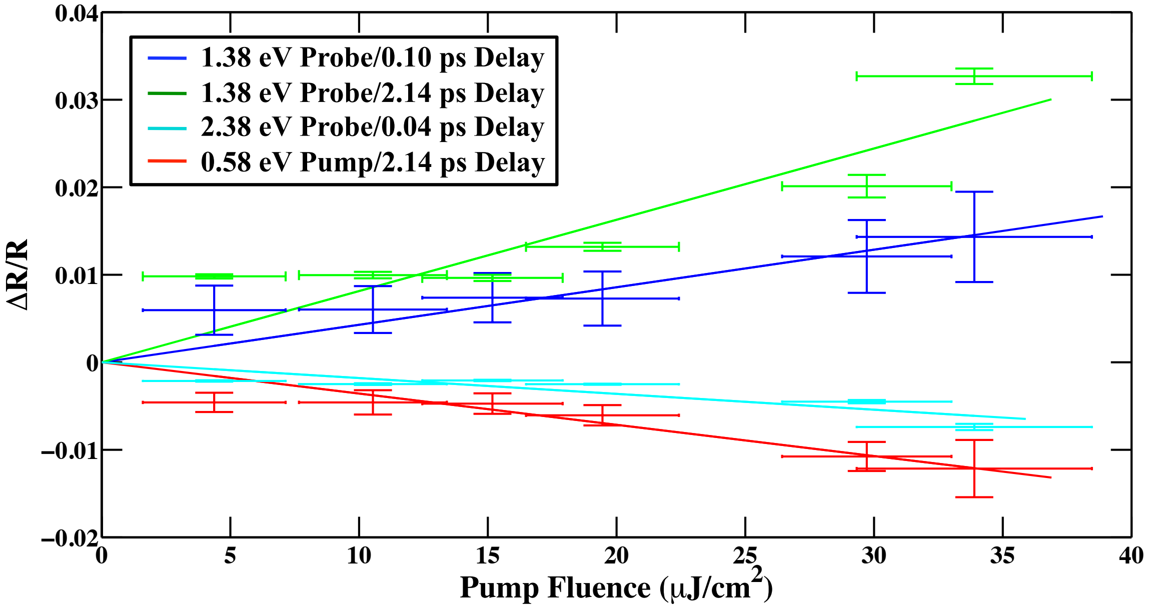

Figure S12: Pump fluence dependence of transient reflection intensity for the 2.64 eV pumped data at various probe energies and delays. Compared to the other pump energies these curves appear much noisier and are fit poorly by lines passing through zero. We suspect that this is because the pump itself is quite noisy having been generated through a series of nonlinear processes. For this reason we have shown the SVD components in the manuscript to demonstrate that throughout the range of pump fluences used in our measurements the dynamics are consistent.

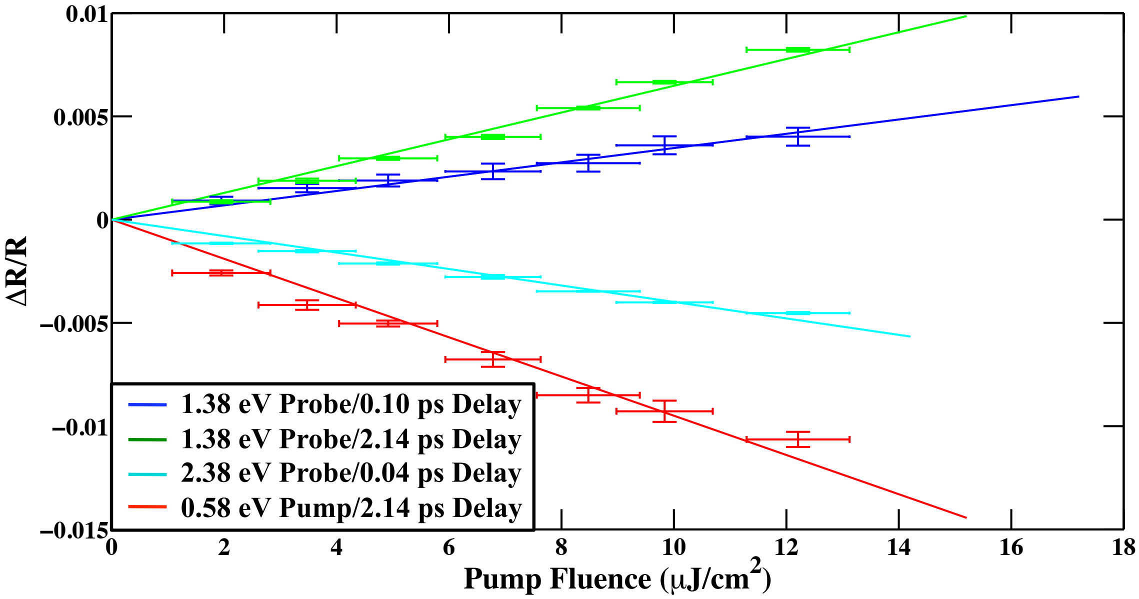

Figure S13: Pump fluence dependence of transient reflection intensity for the 1.55 eV pumped data at the same probe energies and delays as above. In all cases the trend is linear throughout the range.

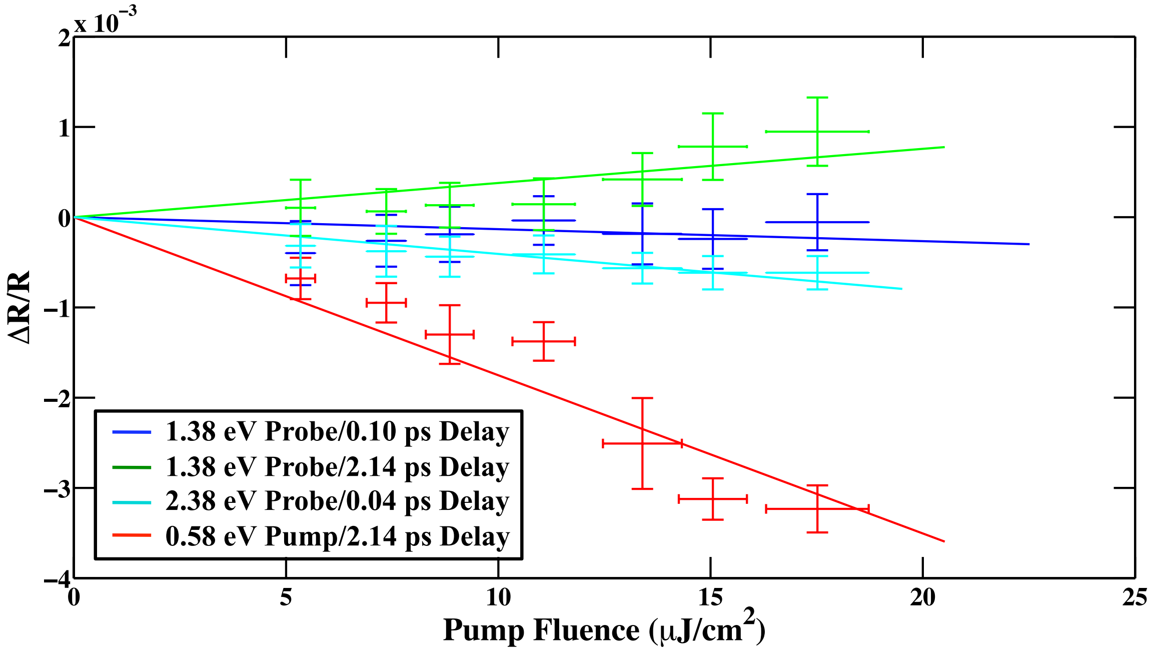

Figure S14: Same as above for the 0.89 eV pumped data. Where signal is appreciable, the trend appears to be linear throughout this range of fluences.

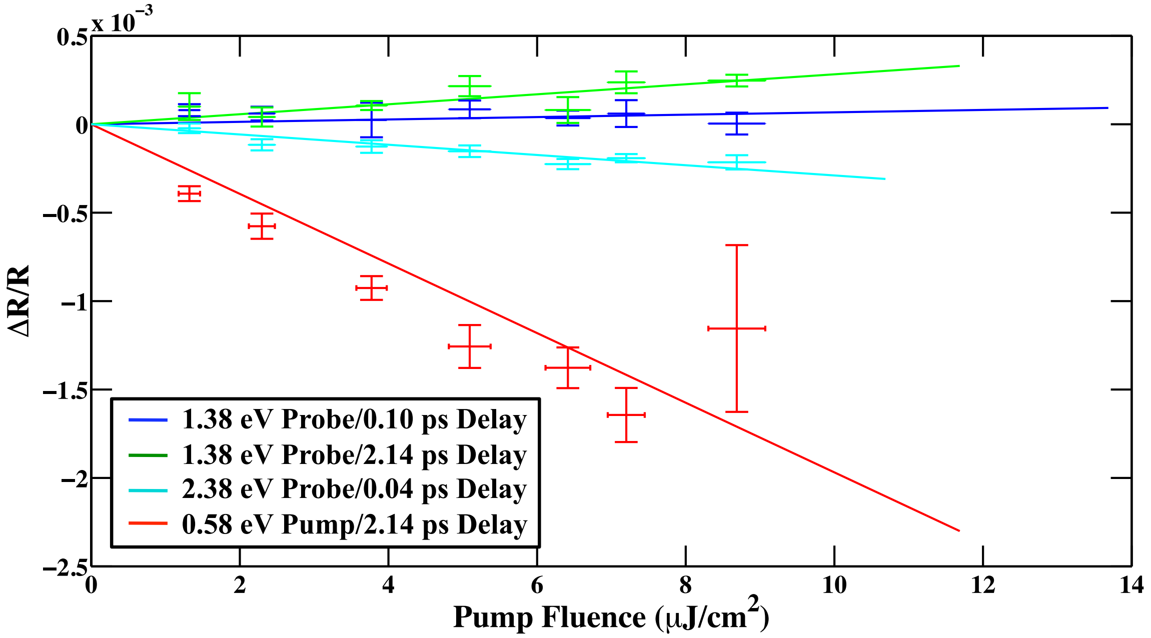

Figure S15: Same as above for the 0.58 eV pumped data. Again where signal is appreciable, the trend appears to be linear throughout this range of fluences.

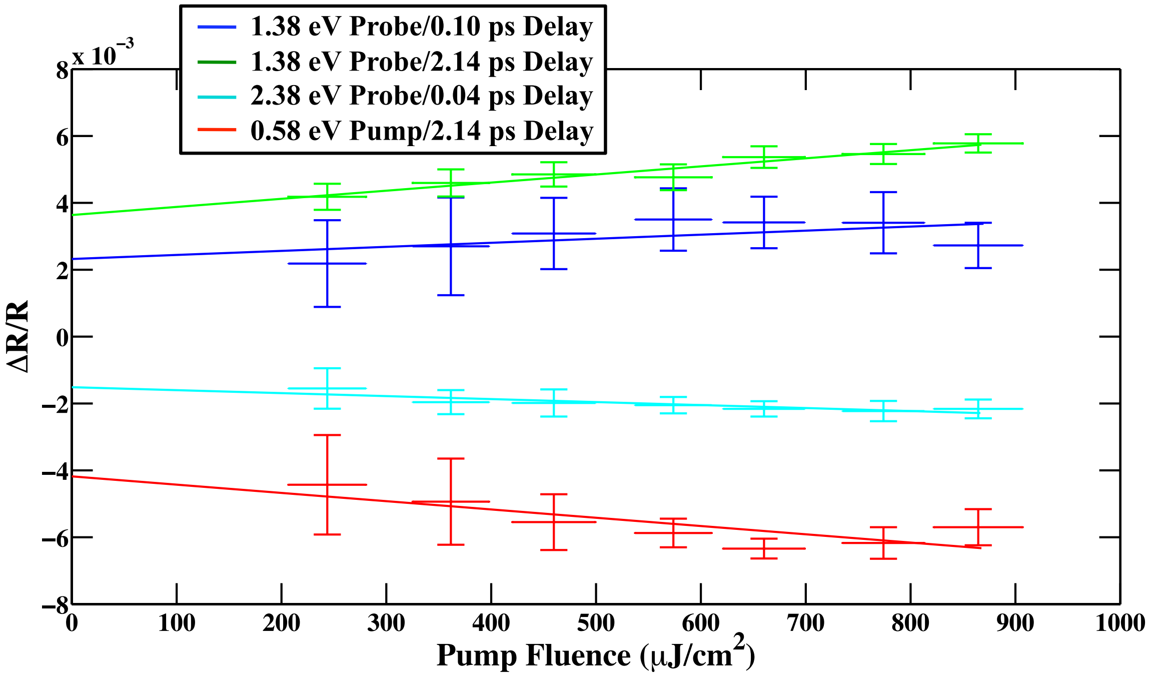

Figure S16: Where our white light fluence was quite large we found it prudent to also conduct probe fluence dependence of our signal. These dependencies are shown at the same probe wavelengths and delays as above. Only a very slight linear dependence is observed showing that the probe does not interact nonlinearly with the sample to any large degree.

Additional Transient Reflection Data

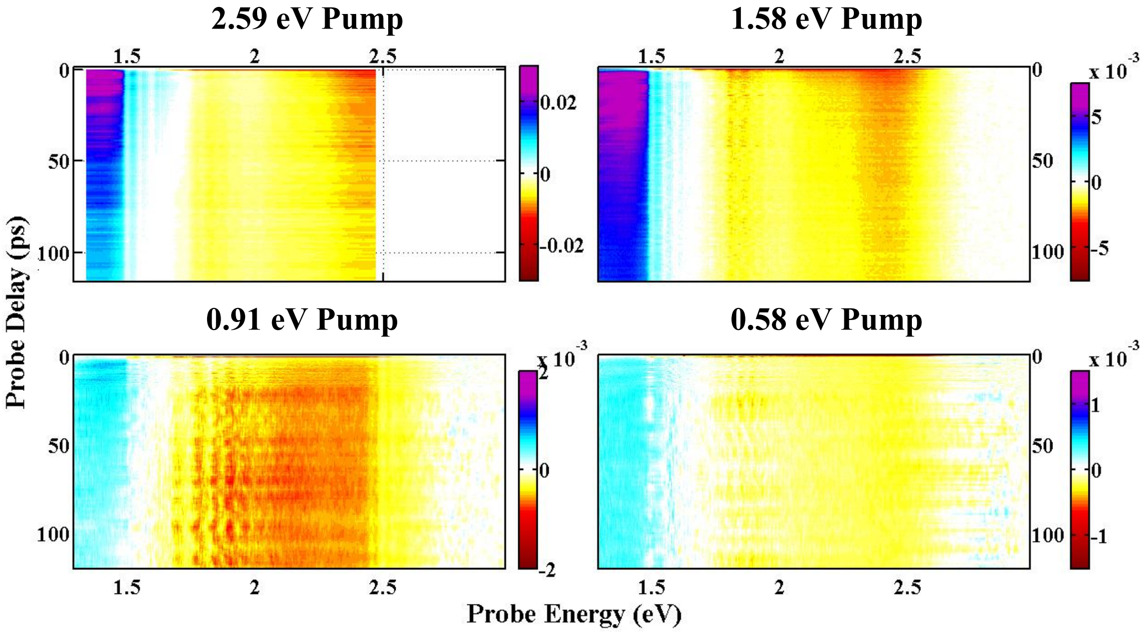

Figure S17: Two dimensional summary of the transient reflection data for each of the pump energies. The top row from left to right shows the 2.59 eV pump and the 1.58 eV pump data, while the bottom row shows the 0.91 eV pump data on the left and the 0.58 pump data on the right.

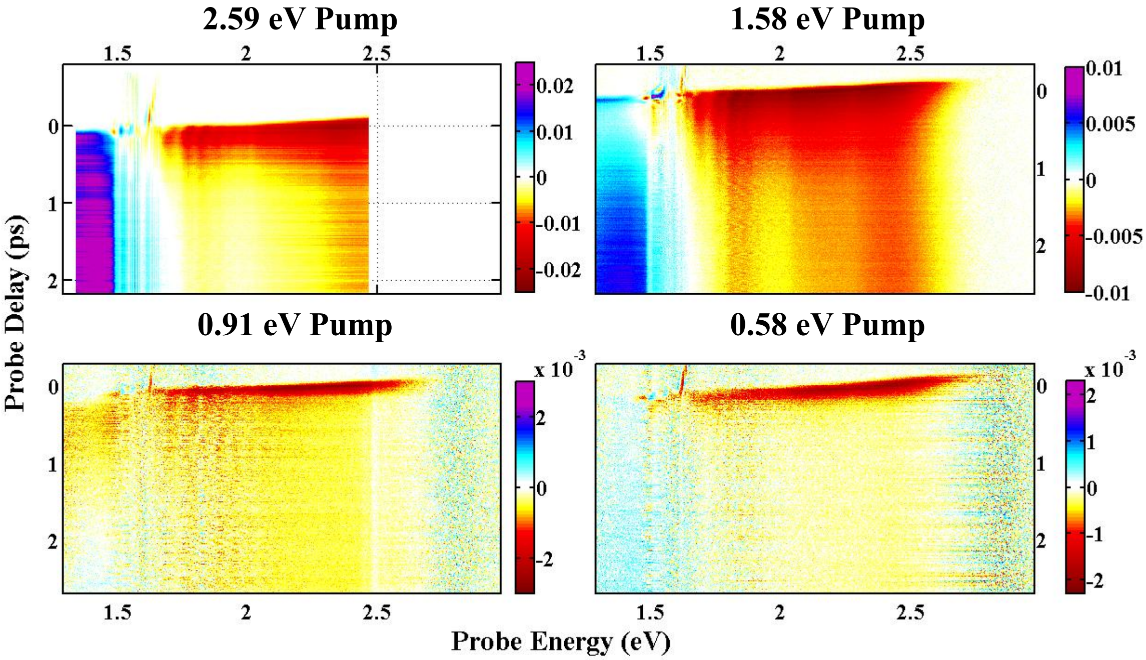

Figure S18: Transient reflection measurements were also done with each of the pump wavelengths near time zero with a much smaller time step (10 fs). Here the chirp of the white light can be seen as well as the temporal distortion of the white light near its seed energy (1.58 eV).

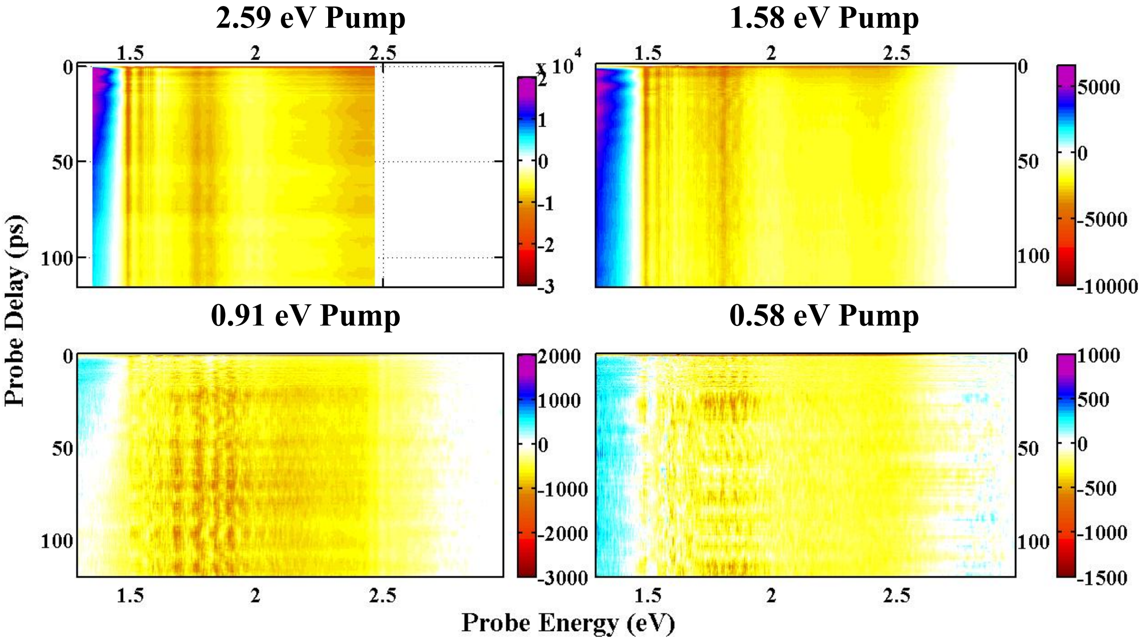

Figure S19: Two dimensional summary of the transient absorption data obtained from transforming the data in Fig. S18 according to the method presented in the Data Analysis section of the manuscript.

Data Transformation Code

```
% 'data' is a two-dimensional matrix containing the transient reflection
% data and having dimension (# of probe energies) x (# of delays).  Some of
% the steps involved in this transformation are quite computationally
% demanding (especially creation of the hilbert transform operator).  To
% illustrate this:  on a macbook from ca. 2007 this file took ~3 minutes to
% run with length(probeWL) = 3236.

data = delR;
probeWL = lambda;
delay = t;




% As explained in the text the transformation is non-local and can be
% improved slightly by extrapolating the edges of the data.  'extrapLength'
% indicates the number of data points you wish to add to EACH edge.  Enter
% '0' if no extrapolation is desired.

extrapLength = 20;


% The transformation should be done in units of energy.

data = flipud(data);
probeEnergy = fliplr(1240./probeWL);
```

```
% The steady state complex material response must be known and entered here
% as the complex dielectric function (epsilon).

ep1 = epreal;
ep2 = epimag;
epEnergy = epeV;



% The transformation relies upon a matrix inversion.  A particularly
% convenient method to accomplish this is to first perform a singular value
% decomposition on the matrix to be inverted.  If said matrix is
% ill-conditioned (i.e. nearly singular) the component vectors
% corresponding to singular values below the specified threshold can be
% neglected.  In our case there was no problem with the condition of the
% matrix, however, in principle one should always check.

SVDthresh = 0.0000001;


% If the inversion has been performed before the matrices U, S, and V
% should be kept on the workspace and 'makehill' should be set to 'no'.
% This will save time you the time and computational power necessary to
% perform some of the more demanding and time consuming calculations.

makehill = 'yes';
```

```
% A new wavelength axis is created that includes the number of extrapolated
% points indicated by extrapLength on each side.

delta = probeWL(2) - probeWL(1);
leftExtrap = probeWL(1)-delta*extrapLength:delta:probeWL(1)-delta;
rightExtrap = probeWL(end)+delta:delta:probeWL(end)+delta*extrapLength;
extrapWL = [leftExtrap probeWL rightExtrap];
```

```
% The wavelength axis is converted to energy since the transformation
% should be done in the energy (frequency really) domain.

extrapEnergy = fliplr(1240./extrapWL);
```

```
% The data is extrapolated either as constant values or as zeros.  Switch
% which lines are commented to toggle between the two.  If constant values
% are chosen, 'n' indicates the number of data points to consider at the
% edge in determining the constant value to use for the extrapolation.

n = 3;

data = [kron(ones(extrapLength,1),(1/n)*sum(data(1:n,:),1)); data]; %constants
%data = [zeros(extrapLength,size(data,2)); data]; %zeros
data = flipud(data);

data = [kron(ones(extrapLength,1),(1/n)*sum(data(1:n,:),1)); data]; %constants
%data = [zeros(extrapLength,size(data,2)); data]; %zeros
data = flipud(data);



% The complex material response needs to have the same energy axis as the
% data.  This is done by interpolating the known data onto the newly
% created energy axis.  Switch the commented out code if data was
% originally entered as n and k.

ep1 = interp1(epEnergy,ep1,extrapEnergy,'pchip');
ep2 = interp1(epEnergy,ep2,extrapEnergy,'pchip');



% Index of refraction and extinction coefficient are calculated from the
% digitized susceptibilities if necessary.

n = sqrt((ep1+sqrt(ep1.^2+ep2.^2))/2);
k = sqrt((sqrt(ep1.^2+ep2.^2)-ep1)/2);

figure (1); plot(extrapEnergy,[k; n])


% If calculations are to be repeated for various sets of data on the same
% axis then U, S, and V should be left on the workspace and repetition
% of the following computationally intensive steps can be avoided


if strcmp(makehill,’yes’)

    % A hilbert matrix of the correct dimension is constructed if indicated.
    % This process is computationally intensive and should not be done if the
    % necessary inversion matrices are already on your workspace.

    hill = zeros(length(extrapEnergy),length(extrapEnergy));
    j = 0:1:length(extrapEnergy)-1;
    dumb = 1./j;

    for i = 1:length(extrapEnergy)
        dumb2 = circshift(dumb,[0 i-1]);
        hill(i,:) = dumb2;
    end
    hill = -1/pi*(triu(hill,1) - tril(hill',-1));


    % The various terms involved in the transform are calculated here

    A = (1 + k.^2).^2 + 2.*(-1 + k.^2).*n.^2 + n.^4;
    T1 = diag( (8*k.*n)./ A);
    T2 = diag( (-4*(1 + k.^2 - n.^2)) ./ A );
    B = T1 + T2 * hill;


    % As mentioned above, an especially efficient and convenient way to
    % invert a matrix is using the SVD method

    [U,S,V] = svd(B);

end


% The matrix inversion can be performed using the more straightforward
% method, mitigation of the problems associated with inversion of
% ill-conditioned matrices will not be an option
```

```
%deltaKappa = B\data;



% The inversion is performed using the SVD method (this allows for
% rejecting nearly singular values in the case that the transformation
% operator is ill-conditioned - as mentioned this was not a problem for us
% and the ’SVDthresh’ was set to be lower than the smallest sing. value.)
ind = S < SVDthresh;
Sinv = 1./S;
Sinv(ind) = 0;


% Deltakappa is the actual quantity calculated here.  It is converted to
% Deltaalpha using the relationship alpha = kappa*4*pi/lambda

deltaKappa = V*Sinv*U’*data;
deltaAlpha = deltaKappa*4*pi./kron(ones(1,size(data,2)),(1240E-7./extrapEnergy’));



% The extrapolated data is put out to pasture as its purpose has now been
% served.

deltaKappa = deltaKappa(21:end-20,:);
deltaAlpha = deltaAlpha(21:end-20,:);
```

```
figure (2); imagesc(probeEnergy,delay,deltaAlpha')



% Everybody likes a clean workspace


clear extrapLength ep1 ep2 epEnergy SVDthresh makehill A T1 T2 B ind Sinv
clear dumb dumb2 i j delta leftExtrap rightExtrap extrapEnergy delay
clear extrapWL probeWL
```

References

(1) Liu, S.; Li, M.; Li, S.; Li, H.; Yan, L. Synthesis and Adsorption/Photocatalysis Performance of Pyrite FeS_2. *Appl. Surf. Sci.* **2013**, *268*, 213–217.

(2) Vogt, H.; Chattopadhyay, T.; Stolz, H. J. Complete First-Order Raman Spectra of the Pyrite Structure Compounds FeS_2, MnS_2 And SiP_2. *J. Phys. Chem. Solids* **1983**, *44*, 869–873.

(3) Kasami, M.; Mishina, T.; Yamamoto, S.; Nakahara, J. Coherent Longitudinal Acoustic Phonons in Pyrite. *J. Lumin.* **2004**, *108*, 291–295.