Dipolar pathways in dipolar EPR spectroscopy

- Supplementary Information -

Luis Fábregas-Ibáñez [1], Maxx H. Tessmer [2], Gunnar Jeschke [1], and Stefan Stoll [2]

[1] ETH Zurich, Laboratory of Physical Chemistry, Vladimir-Prelog-Weg 2, 8093 Zurich, Switzerland
[2] University of Washington, Department of Chemistry, Seattle, WA 98195, USA

This supplementary information and all associated datasets can be downloaded from the following Zenodo repository:

https://doi.org/10.5281/zenodo.5516807

All Python scripts were written and run with DeerLab v0.14.0 in Python 3.8-3.9

TABLE OF CONTENTS:

Auxiliary functions

These Python function are used in and shared by (most of) the analyses of the experimental datasets.

```python
import numpy as np
import deerlab as dl
import matplotlib.pyplot as plt
%matplotlib inline
%config InlineBackend.figure_format = 'svg'

# Define colors
blue = '#648fffff'
yellow = '#ffb000ff'
magenta = '#dc267fff'
violet  = '#785ef0ff'
emerald = '#28ac9f'

# Define color scheme for DEER pathways
colors_4pdeer = [blue, magenta, violet, emerald]
colors_5pdeer = [blue, yellow, violet, magenta, emerald]



# ---------------------------------------------------------------------------------
def outliers_mask(Vexp,Vi,factor=5,method='gradient'):
    """
    Create a mask to isolate outlier datapoints arising from moving pump pulse echos
    """
    if method=='gradient':
        mask = abs(np.gradient(Vexp))>dl.noiselevel(Vi)**0.5/factor
    else:
        std = factor*dl.noiselevel(Vi)
        mask = ((Vi>std) | (Vi<-std))
        # Remove also 4 points around those to ensure all influence of the crossing
        iskip = -1
        for i in range(len(t)):
            if i<iskip:
                continue
            if mask[i]:
                mask[i-3:i]=True
                if i<len(t)-3:
                    mask[i+1:i+3]=True
                    iskip=i+3
    return mask
# ---------------------------------------------------------------------------------

# ---------------------------------------------------------------------------------
def get_experimental_taus(descriptor,experiment='4pdeer'):
    """
    Extract the pulse delays used by the 4-pulse and 5-pulse DEER experiments
    from the PulseSPEL script contained in the BES3T formatted descriptor files.
    """
    def getdelay(variable):
        """Get the value assigned to a variable in the PulseSPEL definition."""
        # Extract the PulseSPEL script from the descriptor
        PulseSPEL = descriptor['DSL']['ftEpr']['PlsSPELGlbTxt']
        # Read script line by line
        for line in PulseSPEL.split('\\n'):
            # Only analyze the line if the sought variable is in it
            if variable in line:
                # Split words
                words = line.split(' ')
                # Get value assigned to variable
                index = [i for i,x in enumerate(words) if x==variable][0]
                index += 1
                while words[index]=='' or words[index]=='=':
```

```
        # 4-pulse DEER: deadtime=d3, tau1=d1, tau2=d2
        return getdelay('d3'),getdelay('d1'),getdelay('d2')
    elif experiment == '5pdeer':
        # 5-pulse DEER: deadtime=d3, tau1=d1, tau2=d2, tau3=d11
        return getdelay('d3'),getdelay('d1'),getdelay('d2'),getdelay('d11')
# ------------------------------------------------------------------------------------

# ------------------------------------------------------------------------------------
def display_pathway_analysis(Pfit,ts,Vs,Vfits,masks,lams_list,reftimes_list,concs,ex
    """
    Display the results of the multi-pathway analysis. Shows the experimental datase
    contributions from the individual dipolar pathways.
    """
    ndatasets = int(len(Vs))
    n = 0

    # Color schemes of the dipolar pathways, as shown in the manuscript
    if experiment=='4pdeer':
        colors = colors_4pdeer
    elif experiment=='5pdeer':
        colors = colors_5pdeer

    for Vfit,t,Vexp,mask in zip(Vfits,ts,Vs,masks):

        # Plotting logic
        n+=1
        if n==1:
            m = 1 # Start plotting on third row
        else:
            m += 1 # Move to next subplot

        # Plot the experimental datasets and the fit
        plt.subplot(3,ndatasets,m)
        plt.plot(t[mask],Vexp[mask],'.',color='grey',label='data')
        plt.plot(t[~mask],Vexp[~mask],'.',color='grey',alpha=0.2,label='data')
        plt.xlabel('t [μs]')
        plt.ylabel('V(t)')
        plt.plot(t,Vfit,'b',label='fit',linewidth=2)
        plt.ylim([0.98*np.min(Vfit),1.02*np.max(Vfit)])
        if n==1:
            # Add legend only to first subplot
            plt.legend(frameon=False)
        if n<len(Vs)+1:
            # Add titles only to top row
            plt.title(f'Dataset #{n}')

        lams = lams_list[n-1]
        reftimes = reftimes_list[n-1]
        conc = concs[n-1]

        # Get the fitted unmodulated contribution
        Lam0 = 1-np.sum(lams)
        # Get the globally fitted distance distribution
        Pfit = Pfit/np.trapz(Pfit,r)
        # Compute the fitted intermolecular contribution
        pathways = np.array([[lams[i-1],reftimes[i-1]] if i>0 else [Lam0] for i in r
        Vinter = dl.dipolarbackground(t,pathways,lambda t,lam: dl.bg_hom3d(t,conc,la
    plt.tight_layout()
    if saveas is not None:
        plt.savefig(saveas)
    plt.show()
# ------------------------------------------------------------------------------------
```

1. Experiments on MBP

1.1 Sample prepration

Cysteine mutations were introduced to pETM11 plasmid encoding histidine tagged maltodextrin binding protein (MalE) as previously described (Liu and Naismith) using Phusion high fidelity polymerase (New England Biolabs). Site directed mutation products were transformed into NEB Turbo Escherichia coli (New England Biolabs), purified using the QIAprep kit (Qiagen) and verified by double strand sequencing (Genewiz). Validated double cysteine mutants were transformed into chemically competent BL21 (DE3) E. coli (New England Biolabs). Protein expression was performed in LB Broth (Invitrogen) supplemented with 50 µg/ml kanamycin (Gibco). At approximately OD600 = 0.5 the protein expression was induced with 0.5 mM Isopropyl β-d-1-thiogalactopyranoside (IPTG) and allowed to express for 3 hours at 37 C. After expression, cells were centrifuged and stored at -80 °C until purification. Cell pellets were resuspended in 25 ml lysis buffer (20 mM Tris, 150 mM NaCl, 20 mM imidazole, 0.5 mM Phenylmethylsulfonyl fluoride, pH 7.5). Cells were lysed on ice using a Thermo scientific 550 sonic dismembrator. Lysates were clarified by centrifugation (25,000 X g, 45 min, 4 C) and supernatants were applied to 2.5 ml HisPur Ni-NTA resin (Thermo Fischer Scientific). The resin was washed 3 times with 20 mL wash buffer (20 mM Tris, 150 mM NaCl, 30 mM imidazole, pH 7.5) and eluted with 15 mL Elution buffer (20 mM Tris, 150 mM NaCl, 150 mM imidazole, pH 7.5). A fraction of the eluant was spin labeled by incubating overnight with tenfold excess 1-Oxyl-2,2,5,5-tetramethylpyrroline-3- methyl methanethiosulfonate (MTSL, Santa Cruz Biotechnology). Spin label protein was buffer exchanged by repeated concentration and dilutions using an Amicon Ultra-4 centrifugal filter (30 kDacutoff) and 20 mM Tris, 150 mM NaCl, 20% glycerol buffer at pH 7.5. Final EPR samples were diluted to 50 µM protein in the same buffer made with deuterium oxide and D8 Glycerol (Cambridge isotope laboratories). The EPR samples were shipped overseas. Finally, 40 µL of solution were filled into a 2.95 mm (o.d) EPR tubes.

1.2 Equipment

The pulsed EPR experiments were performed at Q-band on a commercial spectrometer (Bruker ElexSysII

E580) equipped with a 200 W travelling wave tube (TWT) amplifier and a homebuilt resonator suitable for 3 mm o.d. sample tubes. A helium flow cryostat (ER 4118 CF, Oxford Instruments) was used to adjust and stabilize the measurement temperature at 50 K.

1.3 Pulse configuration

The following configurations of the probe and pump pulses were used for the acquisition of the different 4-pulse and 5-pulse DEER datasets:

Dataset	**Probe Freq.**	**Pump Freq.**	$(\pi/2)_{probe}$	$(\pi)_{probe}$	$(\pi/2)_{pump}$
#1	33.490 GHz	33.590 GHz	12 ns	26 ns	20 ns
#2	33.515 GHz	33.565 GHz	12 ns	28 ns	22 ns
#3	33.520 GHz	33.560 GHz	12 ns	20 ns	12 ns
#4	33.525 GHz	33.555 GHz	16 ns	24 ns	12 ns
#5	33.530 GHz	33.550 GHz	12 ns	12 ns	12 ns

For each dataset, the same pulses were used for both the 4-pulse and 5-pulse DEER experiments.

1.4 4-pulse DEER experiments

Using the pulses defined above the 4-pulse DEER experiment was performed according to the sequence:

$(\pi/2)_{probe}$ - $[\tau_1]$ - $(\pi)_{probe}$-$[t]$ - $(\pi)_{pump}$ - $[\tau_1 + \tau_2 - t]$ - $(\pi)_{probe}$-$[\tau_2]$-echo

with τ_1 set to 400 ns and τ_2 set to 3000 ns. The dipolar time t was swept in 8ns steps and a (+x,-x) phase cycle was used.

1.5 Analysis of 4-pulse DEER datasets

```python
# ========================================================
# LOADING & PRE-PROCESSING
# ========================================================

# Files containing the 4-pulse DEER data
path = './data/MBP_50K_Qband/'
files = [
    "MBP_50K_4pDEER_100MHz_offset.DTA",
    "MBP_50K_4pDEER_50MHz_offset.DTA",
    "MBP_50K_4pDEER_40MHz_offset.DTA",
    "MBP_50K_4pDEER_30MHz_offset.DTA",
    "MBP_50K_4pDEER_20MHz_offset.DTA",
]

# Preallocate containers
Vs,ts,masks,tau1s,tau2s = [],[],[],[],[]
for n,file in enumerate(files):

    # Load the file
    t,Vexp,descriptor = dl.deerload(path+file, full_output=True)

    # Get the experimental delays
    t0,tau1,tau2 = get_experimental_taus(descriptor,experiment='4pdeer')
    tau1s.append(tau1)
    tau2s.append(tau2)
    t = t + t0

   # Normalize the dataset
    Vexp /= np.max(Vexp)

    # Pre-processing
    Vexp,Vi,_ = dl.correctphase(Vexp,full_output=True)

    if n>0:
        # Determine points affected by crossing echoes
        mask = ~outliers_mask(Vexp,Vi,factor=6)
    else:
        # Use the full signal
        mask = Vexp<1e99

    # Add data to list
    Vs.append(Vexp)
    ts.append(t)
    masks.append(mask)
```

```
# Distance vector
r = np.linspace(2,6,150)

Vmodels = []
for t,tau1,tau2 in zip(ts,tau1s,tau2s):
    # Construct information on the pathways-refocusing times based on experiment
    experimentInfo = dl.ex_4pdeer(tau1,tau2)
    # Construct the dipolar signal model
    Vmodel = dl.dipolarmodel(t,r,npathways=4,experiment=experimentInfo)
    # The amplitudes of pathways #3 and #4 must be equal
    Vmodel = dl.link(Vmodel,lam23 = ['lam2','lam3'])
    # Freeze the refocusing times at the theoretical values
    Vmodels.append(Vmodel)

# Create a global model
Vglobal = dl.merge(*Vmodels,addweights=True)
# Make the spin concentration and distance distribution global to all submodels
Vglobal = dl.link(Vglobal,
                    conc=['conc_1','conc_2','conc_3','conc_4','conc_5'],
                    reftime1 = ['reftime1_1','reftime1_2','reftime1_3','reftime1_4',
                    reftime2 = ['reftime2_1','reftime2_2','reftime2_3','reftime2_4',
                    reftime3 = ['reftime3_1','reftime3_2','reftime3_3','reftime3_4',
                    reftime4 = ['reftime4_1','reftime4_2','reftime4_3','reftime4_4',
                    P=['P_1','P_2','P_3','P_4','P_5'])
Vglobal.conc.set(lb=40, ub=500, par0=80)

 # Include edges in the regularization operator
L = dl.regoperator(r,2,includeedges=True)

# Fit all experiments datasets to the global model
results = dl.fit(Vglobal,Vs,regop=L, mask=masks, bootstrap=500, regparamrange=[1e-3,

lams_list = [[getattr(results,f'lam1_{n+1}'), getattr(results,f'lam23_{n+1}'), getat
reftimes_list = [[getattr(results,f'reftime1'), getattr(results,f'reftime2'), getatt
concs = [results.conc for n in range(len(Vs))]
```

Table S1 : Fit results and fitted model parameters of the 4-pulse DEER MBP datasets

```
print(results)
```

```
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced 𝛘2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.011        0.836      0.009   -3956.847
   #2         0.005        1.245      0.006   -4388.628
   #3         0.006        1.819      0.008   -4097.428
   #4         0.005        1.119      0.005   -4434.813
   #5         0.016        1.858      0.021   -3277.485
========= ============= ============ ======= ===========
Model parameters:
=========== ========= ========================= ======= ==============================
=========
 Parameter   Value     95%-Confidence interval   Units   Description
=========== ========= ========================= ======= ==============================
=========
 lam1_1      0.345     (0.334,0.358)                     Amplitude of pathway #1
 reftime1    0.409     (0.407,0.412)               μs    Refocusing time of pathway #
1
 lam23_1     0.032     (0.026,0.038)                     Amplitude of pathway #2
 reftime2    3.394     (3.364,3.436)              μs     Refocusing time of pathway #
2
 reftime3    -0.031    (-0.077,0.026)             μs     Refocusing time of pathway #
3
 lam4_1      0.007     (0.002,0.010)                     Amplitude of pathway #4
 reftime4    2.954     (2.944,2.962)              μs     Refocusing time of pathway #
4
 conc        120.801   (98.727,142.782)           μM     Spin concentration
 weight_1    0.972     (0.953,0.990)             None    Weighting factor
 lam1_2      0.185     (0.179,0.191)                     Amplitude of pathway #1
 lam23_2     0.031     (0.027,0.036)                     Amplitude of pathway #2
 lam4_2      0.008     (0.005,0.010)                     Amplitude of pathway #4
 weight_2    0.982     (0.961,1.007)             None    Weighting factor
 lam1_3      0.145     (0.142,0.148)                     Amplitude of pathway #1
 lam23_3     0.039     (0.036,0.042)                     Amplitude of pathway #2
 lam4_3      0.054     (0.050,0.058)                     Amplitude of pathway #4
 weight_3    1.045     (1.021,1.070)             None    Weighting factor
 lam1_4      0.081     (0.079,0.083)                     Amplitude of pathway #1
 lam23_4     0.027     (0.025,0.029)                     Amplitude of pathway #2
 lam4_4      0.035     (0.034,0.037)                     Amplitude of pathway #4
 weight_4    0.990     (0.968,1.018)             None    Weighting factor
 lam1_5      0.065     (0.056,0.074)                     Amplitude of pathway #1
 lam23_5     0.068     (0.061,0.074)                     Amplitude of pathway #2
 lam4_5      0.181     (0.174,0.192)                     Amplitude of pathway #4
 weight_5    1.004     (0.989,1.021)             None    Weighting factor
 P           ...       (...,...)                 None    Non-parametric distance dist
ribution
=========== ========= ========================= ======= =============================
---------
```

Figure S1 : Results of the global analysis of the MBP 4-pulse DEER measurements

```
plt.figure(figsize=[20,12])
display_pathway_analysis(results.P,ts,Vs,results.model,masks,lams_list,reftimes_list
```

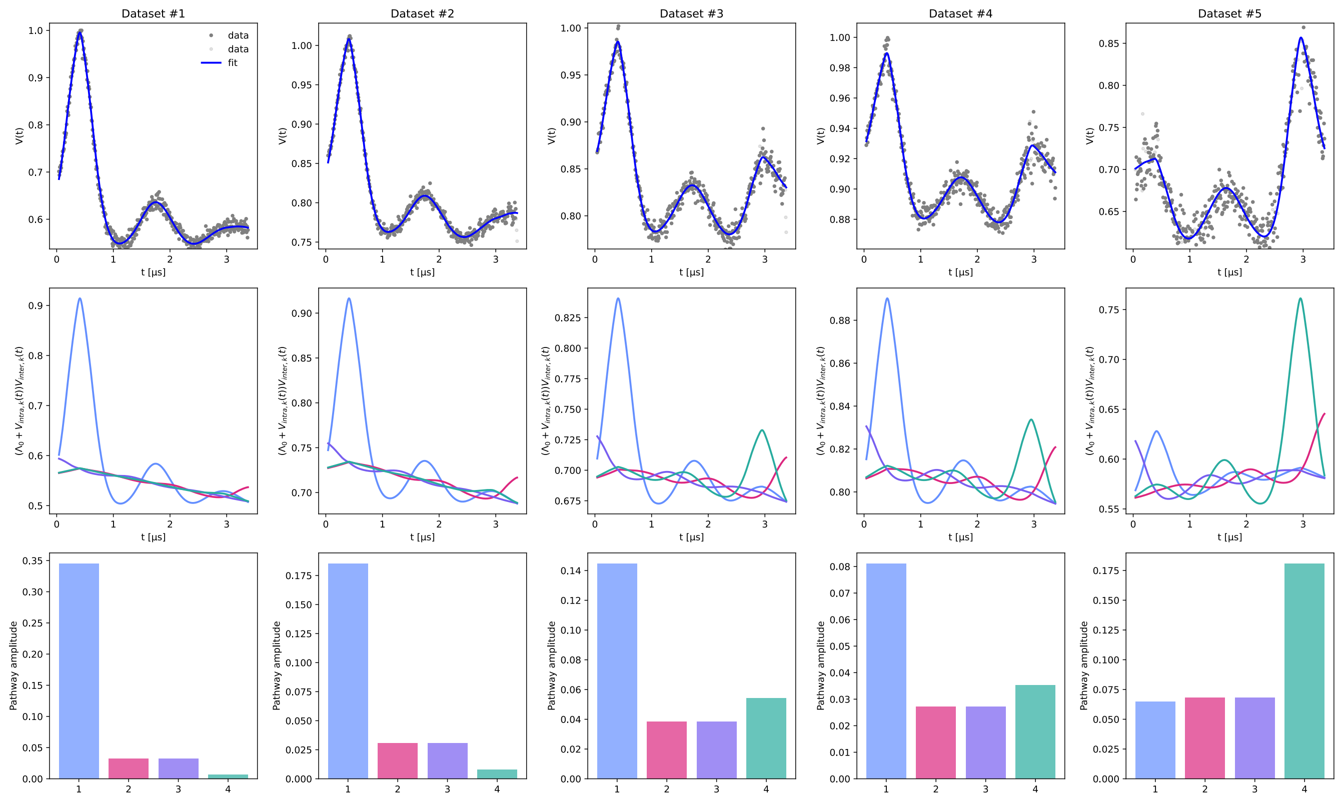
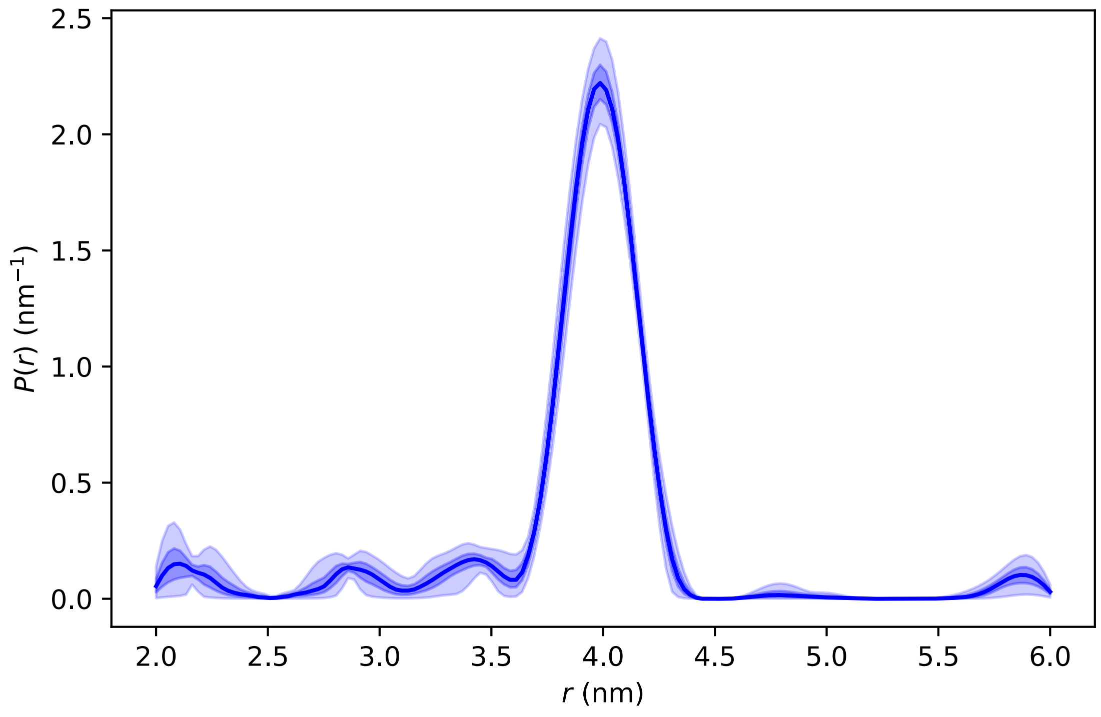


Figure S2 : Globally fitted (non-parametric) distance distribution

```
Pfit = results.P
scale = np.trapz(Pfit,r)
Pfit = Pfit/scale
# Plot the fitted global distance distribution
plt.plot(r,Pfit,color='b')
# Plot the uncertainty of the distance distribution
Pci95 = results.PUncert.ci(95)/scale
Pci50 = results.PUncert.ci(50)/scale
plt.fill_between(r,Pci50[:,0],Pci50[:,1],alpha=0.3,color='b')
plt.fill_between(r,Pci95[:,0],Pci95[:,1],alpha=0.2,color='b')
plt.xlabel('$r$ (nm)')
plt.ylabel('$P(r)$ (nm$^{-1}$)')
plt.tight_layout()
plt.show()
```

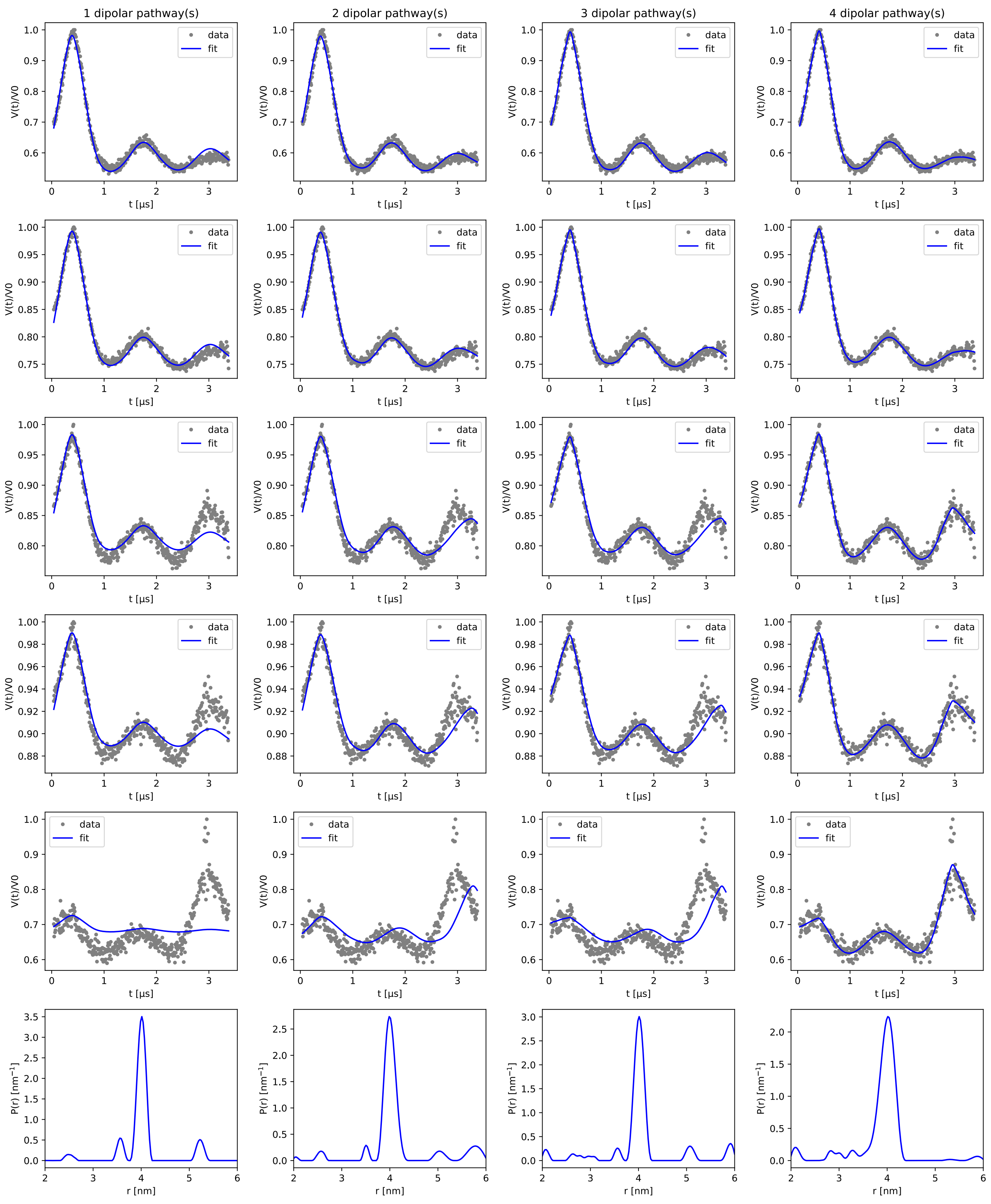


1.6. Effects of insufficient dipolar pathways in the 4-pulse DEER model

This script exemplifies the effects in the fit of the data and in the resulting distance distribution when modelling the 4-pulse DEER signals with an insufficient number of dipolar pathways.

Figure S3 : Effects of insufficient dipolar pathways in the 4-pulse DEER model on the MBP dataset analysis

```python
maxpaths = [1,2,3,4]
plt.figure(figsize=[15,18])
for n,npathways in enumerate(maxpaths):
    # Interspin vector distances
    r = np.linspace(2,6,150)

    Vmodels = []
    for t,tau1,tau2 in zip(ts,tau1s,tau2s):
        # Construct information on the pathways-refocusing times based on experiment
        experimentInfo = dl.ex_4pdeer(tau1,tau2)
        # Construct the dipolar signal model
        Vmodel = dl.dipolarmodel(t,r,npathways=npathways,experiment=experimentInfo)

        Vmodels.append(Vmodel)

    # Create a global model
    Vglobal = dl.merge(*Vmodels,addweights=True)
    # Make the spin concentration and distance distribution global to all submodels
    Vglobal = dl.link(Vglobal,
                        conc=['conc_1','conc_2','conc_3','conc_4','conc_5'],
                        P=['P_1','P_2','P_3','P_4','P_4','P_5'])
    if npathways==1:
        Vglobal = dl.link(Vglobal,**{f'reftime': [f'reftime_1',f'reftime_2',f'reftim
    else:
        for i in range(npathways):
            Vglobal = dl.link(Vglobal,**{f'reftime{i+1}': [f'reftime{i+1}_1',f'refti

    Vglobal.conc.set(lb=40, ub=500, par0=80)

    # Include edges in the regularization operator
    L = dl.regoperator(r,2,includeedges=True)

    # Fit all experiments datasets to the global model
    npathsresult = dl.fit(Vglobal,Vs,regop=L, mask=masks, regparamrange=[1e-3,1e0], 

    Pfit = npathsresult.P/np.trapz(npathsresult.P,r)

    for j in range(5):
        plt.subplot(6,len(maxpaths),n+1+j*len(maxpaths))
        plt.plot(ts[j][masks[j]],Vs[j][masks[j]],'.',color='gray')
        plt.plot(ts[j],npathsresult.model[j],'b')
        plt.legend(['data','fit'])
        plt.xlabel('t [μs]')
        plt.ylabel('V(t)/V0')
        if j==0:
            plt.title(f'{n+1} dipolar pathway(s)')

    # Plot the globally fitted non-parametric distance distribution
    plt.subplot(6,len(maxpaths),n+1+5*len(maxpaths))
    plt.plot(r,Pfit,'b')
    plt.xlabel('r [nm]')
    plt.ylabel('P(r) [nm$^{-1}$]')
    plt.xlim([min(r),max(r)])

plt.tight_layout()
plt.show()
```



1.7 5-pulse DEER experiments

Using the pulses defined above the 4-pulse DEER experiment was performed according to the sequence:

$(\pi/2)_{probe}$ - $[\tau_1]$ - $(\pi)_{probe}$-$[\tau_1 + \tau_2 - t]$- $(\pi)_{pump}$ - $[t]$ - $(\pi)_{probe}$-$[\tau_3]$ - $(\pi)_{pump}$- $[\tau_2 - \tau_3]$-echo

with τ_1 set to 1800 ns, τ_2 set to 2300 ns and τ_3 set to 200 ns. The dipolar time t was swept in 8ns steps and a (+x,-x) phase cycle was used.

1.8 Analysis of 5-pulse DEER datasets

```python
# ========================================================
# LOADING & PRE-PROCESSING
# ========================================================

# Files containing the 5-pulse DEER data
path = './data/MBP_50K_Qband/'
files = [
    "MBP_50K_5pDEER_100MHz_offset.DTA",
    "MBP_50K_5pDEER_50MHz_offset.DTA",
    "MBP_50K_5pDEER_40MHz_offset.DTA",
    "MBP_50K_5pDEER_30MHz_offset.DTA",
    "MBP_50K_5pDEER_20MHz_offset.DTA",
]

# Preallocate containers
Vs,ts,masks,tau1s,tau2s,tau3s = [],[],[],[],[],[]
for n,file in enumerate(files):

    # Load the file
    t,Vexp,descriptor = dl.deerload(path+file, full_output=True)

    t0,tau1,tau2,tau3 = get_experimental_taus(descriptor,experiment='5pdeer')
    tau1s.append(tau1)
    tau2s.append(tau2)
    tau3s.append(tau3)
    # The experiment was performed backwards, adjust time vector
    t = tau1+tau2 - t - t0

    # Pre-processing
    Vexp /= np.max(Vexp)
    Vexp,Vi,_ = dl.correctphase(Vexp,full_output=True)

    if n>1:
        # Determine points affected by crossing echoes
        mask = ~outliers_mask(Vexp,Vi,factor=15)
    else:
        # Use the full signal
        mask = Vexp<1e99


    # Add data to list
    Vs.append(Vexp)
    ts.append(t)
    masks.append(mask)
```

```
# Distance vector
r = np.linspace(2,6,150)

Vmodels = []
for t,tau1,tau2,tau3 in zip(ts,tau1s,tau2s,tau3s):
    # Construct information on the pathways-refocusing times based on experiment
    experimentInfo = dl.ex_5pdeer(tau1,tau2,tau3)
    # Construct the dipolar signal model
    Vmodel = dl.dipolarmodel(t,r,npathways=5,experiment=experimentInfo)
    # The amplitudes of pathways #3 and #4 must be equal
    Vmodel = dl.link(Vmodel,lam34 = ['lam3','lam4'])
    # Freeze the refocusing times at the theoretical values
    Vmodels.append(Vmodel)

# Create a global model
Vglobal = dl.merge(*Vmodels,addweights=True)
# Make the spin concentration and distance distribution global to all submodels
Vglobal = dl.link(Vglobal,
                    conc=['conc_1','conc_2','conc_3','conc_4','conc_5'],
                    reftime1 = ['reftime1_1','reftime1_2','reftime1_3','reftime1_4',
                    reftime2 = ['reftime2_1','reftime2_2','reftime2_3','reftime2_4',
                    reftime3 = ['reftime3_1','reftime3_2','reftime3_3','reftime3_4',
                    reftime4 = ['reftime4_1','reftime4_2','reftime4_3','reftime4_4',
                    reftime5 = ['reftime5_1','reftime5_2','reftime5_3','reftime5_4',
                    P=['P_1','P_2','P_3','P_4','P_4','P_5'])
Vglobal.conc.set(lb=40, ub=500, par0=80)

 # Include edges in the regularization operator
L = dl.regoperator(r,2,includeedges=True)

# Fit all experiments datasets to the global model
results = dl.fit(Vglobal,Vs,bootstrap=500,mask=masks,regop=L,nonlin_tol=1e-3,regparar

lams_list = [[getattr(results,f'lam1_{n+1}'), getattr(results,f'lam2_{n+1}'), getatt
reftimes_list = [[getattr(results,f'reftime1'), getattr(results,f'reftime2'), getatt
concs = [results.conc for n in range(len(Vs))]
```

Table S2 : Fit results and fitted model parameters of the 5-pulse DEER MBP datasets

```
print(results)
```

```
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced 𝛘2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.010         1.004      0.010   -4782.523
   #2         0.004         3.190      0.007   -5138.812
   #3         0.006         9.348      0.018   -4115.142
   #4         0.005         7.151      0.013   -4490.279
   #5         0.006         95.828     0.053   -3041.511
========= ============= ============ ======= ===========
Model parameters:
=========== ========= ========================= ======= =============================
=========
 Parameter   Value     95%-Confidence interval   Units   Description
=========== ========= ========================= ======= =============================
=========
 lam1_1      0.252     (0.239,0.268)                     Amplitude of pathway #1
 reftime1    0.178     (0.171,0.184)              μs     Refocusing time of pathway #
1
```

```
 lam2_1       0.107      (0.073,0.147)                 Amplitude of pathway #2
 reftime2     2.274      (2.253,2.291)          µs     Refocusing time of pathway #
2
 lam34_1      0.043      (0.031,0.056)                 Amplitude of pathway #3
 reftime3     2.117      (2.007,2.195)          µs     Refocusing time of pathway #
3
 reftime4     1.902      (1.900,1.909)          µs     Refocusing time of pathway #
4
 lam5_1       0.035      (0.028,0.044)                 Amplitude of pathway #5
 reftime5     3.944      (3.888,3.985)          µs     Refocusing time of pathway #
5
 conc         131.307    (72.158,183.185)       µM     Spin concentration
 weight_1     1.416      (1.367,1.476)         None    Weighting factor
 lam1_2       0.128      (0.119,0.139)                 Amplitude of pathway #1
 lam2_2       0.063      (0.038,0.091)                 Amplitude of pathway #2
 lam34_2      0.033      (0.023,0.042)                 Amplitude of pathway #3
 lam5_2       0.026      (0.022,0.030)                 Amplitude of pathway #5
 weight_2     1.261      (1.209,1.319)         None    Weighting factor
 lam1_3       0.094      (0.085,0.103)                 Amplitude of pathway #1
 lam2_3       0.061      (0.022,0.109)                 Amplitude of pathway #2
 lam34_3      0.050      (0.035,0.065)                 Amplitude of pathway #3
 lam5_3       0.050      (0.046,0.056)                 Amplitude of pathway #5
 weight_3     1.181      (1.126,1.227)         None    Weighting factor
 lam1_4       0.040      (0.035,0.046)                 Amplitude of pathway #1
 lam2_4       0.042      (0.022,0.065)                 Amplitude of pathway #2
 lam34_4      0.028      (0.020,0.035)                 Amplitude of pathway #3
 lam5_4       0.035      (0.028,0.042)                 Amplitude of pathway #5
 weight_4     1.125      (1.085,1.176)         None    Weighting factor
 lam1_5       0.065      (0.045,0.080)                 Amplitude of pathway #1
 lam2_5       0.051      (0.006,0.103)                 Amplitude of pathway #2
 lam34_5      0.062      (0.042,0.082)                 Amplitude of pathway #3
 lam5_5       0.119      (0.089,0.147)                 Amplitude of pathway #5
 weight_5     0.827      (0.768,0.892)         None    Weighting factor
 P            ...        (...,...)             None    Non-parametric distance dist
ribution
=========== ========= ========================= ======= =============================
=========
```

Figure S4 : Results of the global analysis of the MBP 5-pulse DEER measurements

```
plt.figure(figsize=[20,12])
display_pathway_analysis(results.P,ts,Vs,results.model,masks,lams_list,reftimes_list
```

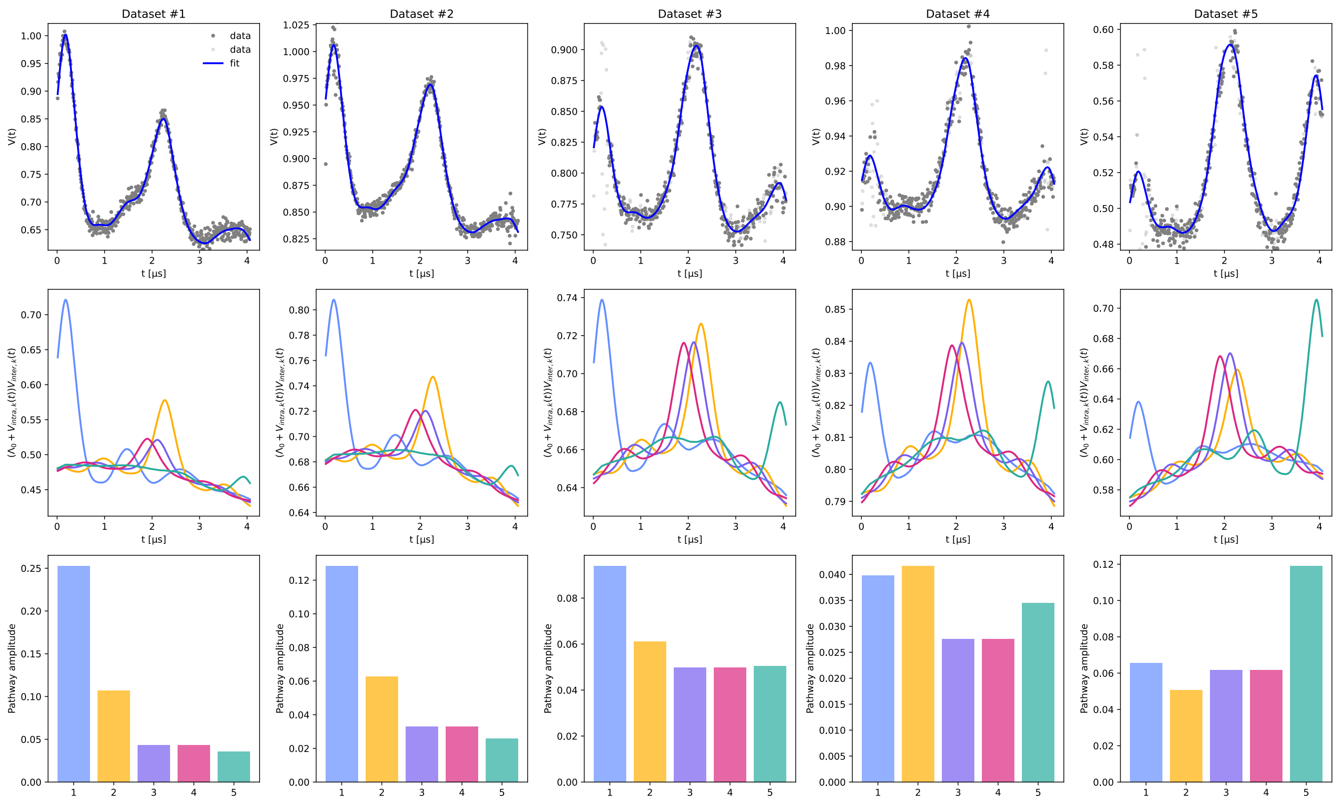
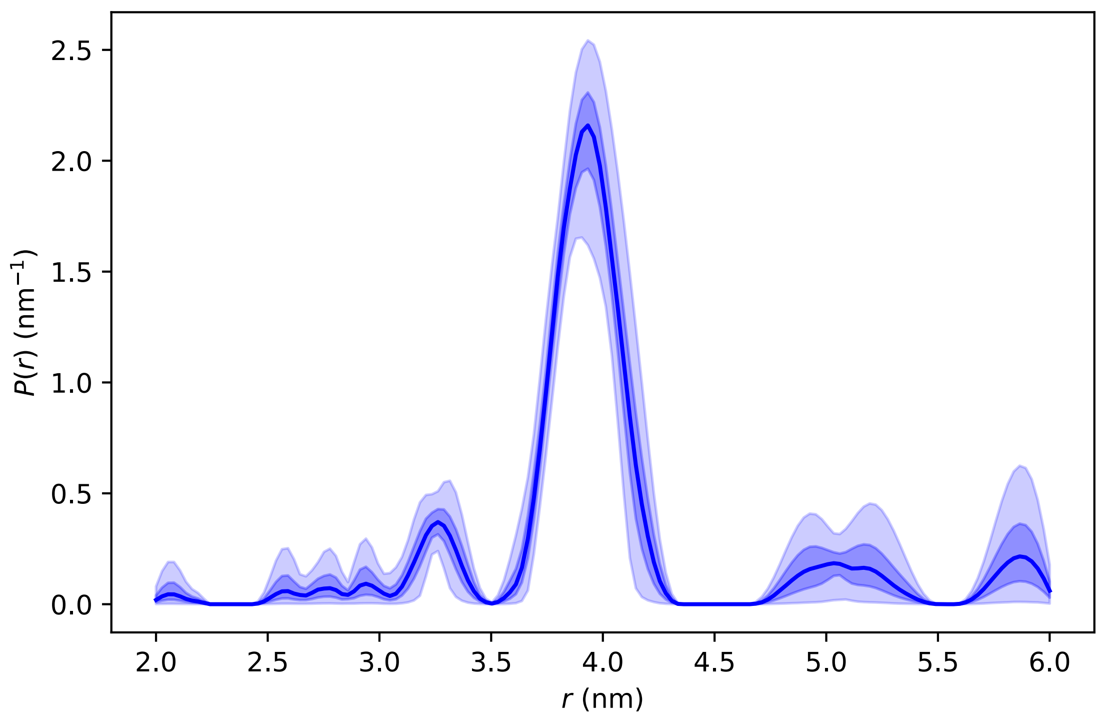


Figure S5 : Globally fitted (non-parametric) distance distribution

```
Pfit = results.P
scale = np.trapz(Pfit,r)
Pfit = Pfit/scale
# Plot the fitted global distance distribution
plt.plot(r,Pfit,color='b')
# Plot the uncertainty of the distance distribution
Pci95 = results.PUncert.ci(95)/scale
Pci50 = results.PUncert.ci(50)/scale
plt.fill_between(r,Pci50[:,0],Pci50[:,1],alpha=0.3,color='b')
plt.fill_between(r,Pci95[:,0],Pci95[:,1],alpha=0.2,color='b')
plt.xlabel('$r$ (nm)')
plt.ylabel('$P(r)$ (nm$^{-1}$)')
plt.tight_layout()
plt.show()
```

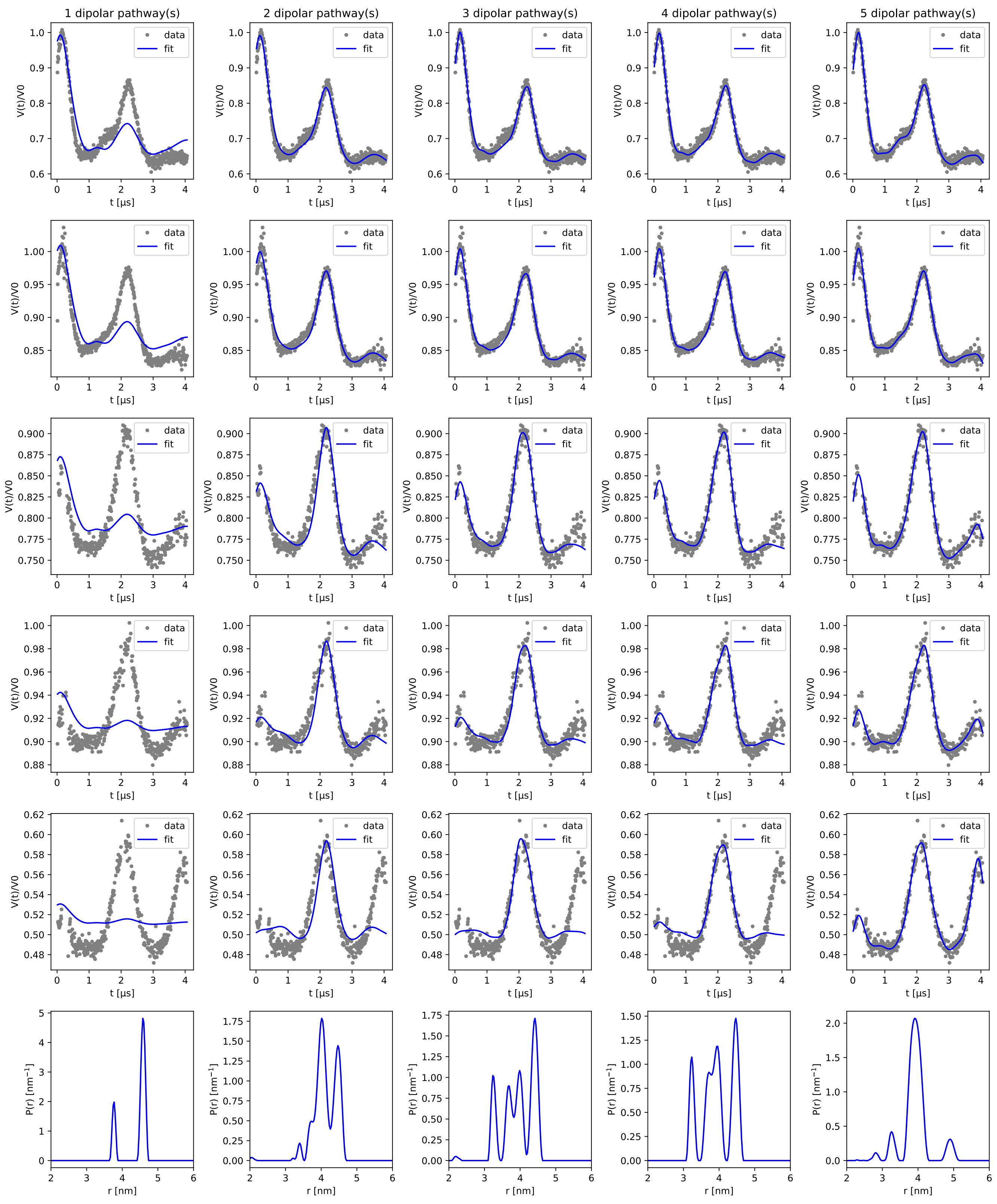


1.9. Effects of insufficient dipolar pathways in the 5-pulse DEER model

This script exemplifies the effects in the fit of the data and in the resulting distance distribution when modelling the 5-pulse DEER signals with an insufficient number of dipolar pathways.

Figure S6 : Effects of insufficient dipolar pathways in the 5-pulse DEER model on the MBP dataset analysis

```
maxpaths = [1,2,3,4,5]
plt.figure(figsize=[15,18])
for n,npathways in enumerate(maxpaths):
    # Interspin vector distances
    r = np.linspace(2,6,150)

    Vmodels = []
    for t,tau1,tau2,tau3 in zip(ts,tau1s,tau2s,tau3s):
        # Construct information on the pathways-refocusing times based on experiment
        experimentInfo = dl.ex_5pdeer(tau1,tau2,tau3)
        # Construct the dipolar signal model
        Vmodel = dl.dipolarmodel(t,r,npathways=npathways,experiment=experimentInfo)

        Vmodels.append(Vmodel)

    # Create a global model
    Vglobal = dl.merge(*Vmodels,addweights=True)
    # Make the spin concentration and distance distribution global to all submodels
    Vglobal = dl.link(Vglobal,
                        conc=['conc_1','conc_2','conc_3','conc_4','conc_5'],
                        P=['P_1','P_2','P_3','P_4','P_4','P_5'])
    if npathways==1:
        Vglobal = dl.link(Vglobal,**{f'reftime': [f'reftime_1',f'reftime_2',f'reftime
    else:
        for i in range(npathways):
            Vglobal = dl.link(Vglobal,**{f'reftime{i+1}': [f'reftime{i+1}_1',f'reftin

    Vglobal.conc.set(lb=40, ub=500, par0=80)

    # Include edges in the regularization operator
    L = dl.regoperator(r,2,includeedges=True)

    # Fit all experiments datasets to the global model
    npathsresult = dl.fit(Vglobal,Vs,regop=L, mask=masks, regparamrange=[1e-3,1e0], r

    Pfit = npathsresult.P/np.trapz(npathsresult.P,r)

    for j in range(5):
        plt.subplot(6,len(maxpaths),n+1+j*len(maxpaths))
        plt.plot(ts[j][masks[j]],Vs[j][masks[j]],'.',color='gray')
        plt.plot(ts[j],npathsresult.model[j],'b')
        plt.legend(['data','fit'])
        plt.xlabel('t [μs]')
        plt.ylabel('V(t)/V0')
        if j==0:
            plt.title(f'{n+1} dipolar pathway(s)')

    # Plot the globally fitted non-parametric distance distribution
    plt.subplot(6,len(maxpaths),n+1+5*len(maxpaths))
    plt.plot(r,Pfit,'b')
    plt.xlabel('r [nm]')
    plt.ylabel('P(r) [nm$^{-1}$]')
    plt.xlim([min(r),max(r)])

plt.tight_layout()
plt.show()
```

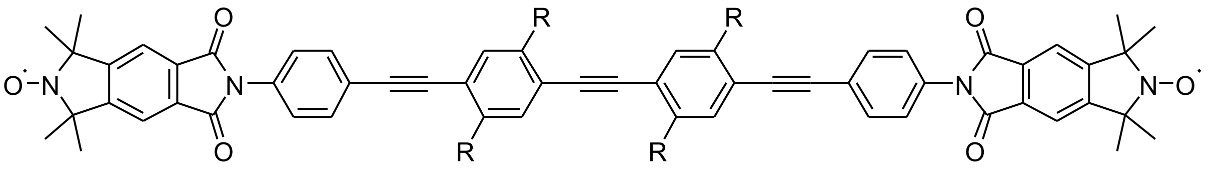


2. Experiments on an oligePPE

2.1. Sample

A pre-existing 90 μM solution of a rigid oligo(p-phenyleneethynylene) (oligoPPE) bi-radical in deuterated o-terphenyl (dOTP) was used (see structure below). Its synthesis and preparation is reported elsewhere (see JACS, 2010, 132, 10107–10117, DOI: 10.1021/ja102983b).

```
from IPython.display import SVG
SVG(filename='.\graphics\oligoPPE.svg')
```



2.2 Equipment

The pulsed EPR experiments were performed at Q-band on a commercial spectrometer (Bruker ElexSysII E580) equipped with a 200 W travelling wave tube (TWT) amplifier and a homebuilt resonator suitable for 3 mm o.d. sample tubes. A helium flow cryostat (ER 4118 CF, Oxford Instruments) was used to adjust and stabilize the measurement temperature at 50K.

2.3 Configuration

Using the pulses defined above the 4-pulse DEER experiment was performed according to the sequence:

$(\pi/2)_{probe}$ - $[\tau_1]$ - $(\pi)_{probe}$-$[t]$ - $(\pi)_{pump}$ - $[\tau_1 + \tau 2 - t]$ - $(\pi)_{probe}$-$[\tau_2]$-echo

The dipolar time t was swept in 8 ns steps and a (+x,-x) phase cycle was used with the following setups:

Dataset	τ_1	τ_2	**Probe Freq.**	**Pump Freq.**	$(\pi/2)_{probe}$	$(\pi)_{probe}$	$(\pi/2)_{pump}$
#1	750 ns	7000 ns	34.41 GHz	34.51 MHz	16 ns	16 ns	16 ns
#2	750 ns	7000 ns	34.47 GHz	34.51 MHz	6 ns	12 ns	16 ns
#3	600 ns	3000 ns	34.41 GHz	34.51 MHz	16 ns	16 ns	16 ns
#4	600 ns	3000 ns	34.47 GHz	34.51 MHz	6 ns	12 ns	16 ns

2.4. Analysis of the 4-pulse DEER datasets

The signals were then analyzed using DeerLab 0.13.2 in Python. After loading the data, the dipolar traces phases were optimized using the `deerlab.correctphase()` function. In some datasets, the presence of moving echoes led to the appearance of spikes in the signal. To remove their influence from the analysis they must be omitted. As the spikes appear in the imaginary part of the signal as well, we removed any point whose imaginary value exceeded four times the estimated standard deviation of the noise (estimated via the `deerlab.noiselevel()` function). As mentioned in the main text, we modelled the experimental signals to account for all four modulated dipolar pathways of the 4-pulse DEER experiments, orientation selection effects modeled using a phenomenological model based on a 4-point cubic spline, and for the distance distribution modeled as a unimodal Gaussian distribution centered about approx. 4 nm. The model was fitted to the individual data using the function `deerlab.fitparamodel()` from DeerLab in Python.

To avoid issues related to some numerical properties of the model, the parameters of the orientational distribution model were determined separately via a Monte-Carlo estimation from 10000 samples. The values were then fixed, while all other parameters were optimized.

The following script was used for the full analysis of the 4-pulse DEER experimental datasets:

```python
from scipy.interpolate import make_interp_spline

# Path to folder containing experimental data
path = './data/MSA236_4pDEER_50K_Qband/'
files = [
    'MSA236_biradical_4pdeer_50K_long_100MHz.DTA',
    'MSA236_biradical_4pdeer_50K_long_40MHz.DTA',
]

# Pre-allocate containers
Vs, ts, Vsmasked, tsmasked, tau1s, tau2s, masks = [],[],[],[],[],[],[]

# Loading and pre-processing
Ndat = len(files)
for n,file in enumerate(files):
    # Load the experimental data
    t,Vexp,descriptor = dl.deerload(path+file, full_output=True)

    t0,tau1,tau2 = get_experimental_taus(descriptor,experiment='4pdeer')
    tau1s.append(tau1)
    tau2s.append(tau2)

    # Account for dead-time
    t += t0
    # Phase correction
    Vexp = Vexp/max(Vexp)
    Vexp,Vi,_ = dl.correctphase(Vexp,full_output=True)

    # Determine points affected by crossing
    if n>0:
        mask = ~outliers_mask(Vexp,Vi,factor=20)
    else:
        mask = np.ones_like(Vexp).astype(bool)

    # Remove spikes due to crossing echoes
    Vmasked = Vexp[mask]
    tmasked = t[mask]

    # Add to the list of pre-processed signals
    masks.append(mask)
    Vs.append(Vexp)
    ts.append(t)
```

```python
# Interspin vector distances
r = np.linspace(3.1,4.1,80)

# Orientation selection distributions
def Pθ_fcn(θ,θspline,Pθspline):
    # Model P(θ) as a 4-point spline with zero-derivative at the edges
    Pθ = make_interp_spline(θspline, Pθspline,bc_type="clamped")
    return 1-Pθ(θ)

# Spline parameters (determined via Monte-Carlo estimation)
θAspline = [0, 0.4, 1, np.pi/2]
PθAspline = [-0.1955177, 0.41876526, -0.55722565, -0.93222681]
PθA = lambda θ: Pθ_fcn(θ, θAspline, PθAspline)

# Spline parameters (determined via Monte-Carlo estimation)
θBspline = [0, 0.4, 0.8, np.pi/2]
PθBspline = [ 0.0744823, -0.38123474, -0.45722565, 0.09777319]
PθB = lambda θ: Pθ_fcn(θ, θBspline, PθBspline)

Pθs = [PθA,PθB]

Vmodels = []
for t,tau1,tau2,Pθ in zip(ts,tau1s,tau2s,Pθs):
    # Construct information on the pathways-refocusing times based on experiment
    experimentInfo = dl.ex_4pdeer(tau1,tau2)
    # Construct the dipolar signal model
    Vmodel = dl.dipolarmodel(t, r, Pmodel=dl.dd_wormchain, npathways=4,
                                experiment=experimentInfo, orisel=Pθ)
    # The amplitudes of pathways #3 and #4 must be equal
    Vmodel = dl.link(Vmodel,lam23 = ['lam2','lam3'])
    # Freeze the refocusing times at the theoretical values
    Vmodels.append(Vmodel)

# Create a global model
Vglobal = dl.merge(*Vmodels)
# Make the spin concentration and distance distribution global to all submodels
Vglobal = dl.link(Vglobal,
                    conc=['conc_1','conc_2'],
                    reftime1 = ['reftime1_1','reftime1_2'],
                    reftime2 = ['reftime2_1','reftime2_2'],
                    reftime3 = ['reftime3_1','reftime3_2'],
                    reftime4 = ['reftime4_1','reftime4_2'],
                    contour=['contour_1','contour_2'],
                    persistence=['persistence_1','persistence_2'])
Vglobal.reftime1.freeze(Vglobal.reftime1.par0)
Vglobal.reftime2.freeze(Vglobal.reftime2.par0)
Vglobal.reftime3.freeze(Vglobal.reftime3.par0)
Vglobal.reftime4.freeze(Vglobal.reftime4.par0)
Vglobal.conc.set(lb=10, ub=500, par0=80)
Vglobal.contour.set(lb=3.00, ub=5.00, par0=4.00)
Vglobal.persistence.set(lb=0.001, ub=100, par0=19)

# Fit all experiments datasets to the global model
results = dl.fit(Vglobal,Vs,mask=masks,bootstrap=500,reg=False)

Vfits = results.model
Pfit = dl.dd_wormchain(r,contour=results.contour,persistence=results.persistence)

lams_list = [[getattr(results,f'lam1_{n+1}'), getattr(results,f'lam23_{n+1}'), getat
reftimes_list = [[getattr(results,f'reftime1'), getattr(results,f'reftime2'), getatt
concs = [results.conc for n in range(len(Vs))]
```

Table S3: Fit results and fitted model parameters of the 4-pulse DEER oligoPPE datasets

```
print(results)
```

```
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced 𝛘2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.010         1.489      0.012   -8525.817
   #2         0.005         4.810      0.011   -8605.515
========= ============= ============ ======= ===========
Model parameters:
============= ========= ========================= ======= ===========================
====
 Parameter     Value     95%-Confidence interval   Units   Description
============= ========= ========================= ======= ===========================
====
 lam1_1        0.345     (0.344,0.346)                     Amplitude of pathway #1
 reftime1      0.750     (frozen)                   μs     Refocusing time of pathway
#1
 lam23_1       0.008     (0.006,0.010)                     Amplitude of pathway #2
 reftime2      7.750     (frozen)                   μs     Refocusing time of pathway
#2
 reftime3      0.000     (frozen)                   μs     Refocusing time of pathway
#3
 lam4_1        0.001     (0.000,0.002)                     Amplitude of pathway #4
 reftime4      7.000     (frozen)                   μs     Refocusing time of pathway
#4
 conc          118.747   (117.827,119.893)          μM     Spin concentration
 contour       4.091     (4.087,4.097)              nm     Contour length
 persistence   20.177    (19.566,20.555)            nm     Persistence length
 lam1_2        0.180     (0.175,0.182)                     Amplitude of pathway #1
 lam23_2       0.069     (0.066,0.073)                     Amplitude of pathway #2
 lam4_2        0.048     (0.045,0.050)                     Amplitude of pathway #4
 scale_1       2.143     (2.126,2.163)              None   None
 scale_2       1.315     (1.286,1.340)              None   None
============= ========= ========================= ======= ===========================
====
```

Figure S7 : Results of the global analysis of the oligoPPE 4-pulse DEER measurements

```
plt.figure(figsize=[10,8])
display_pathway_analysis(Pfit,ts,Vs,Vfits,masks,lams_list,reftimes_list,concs,experi
```

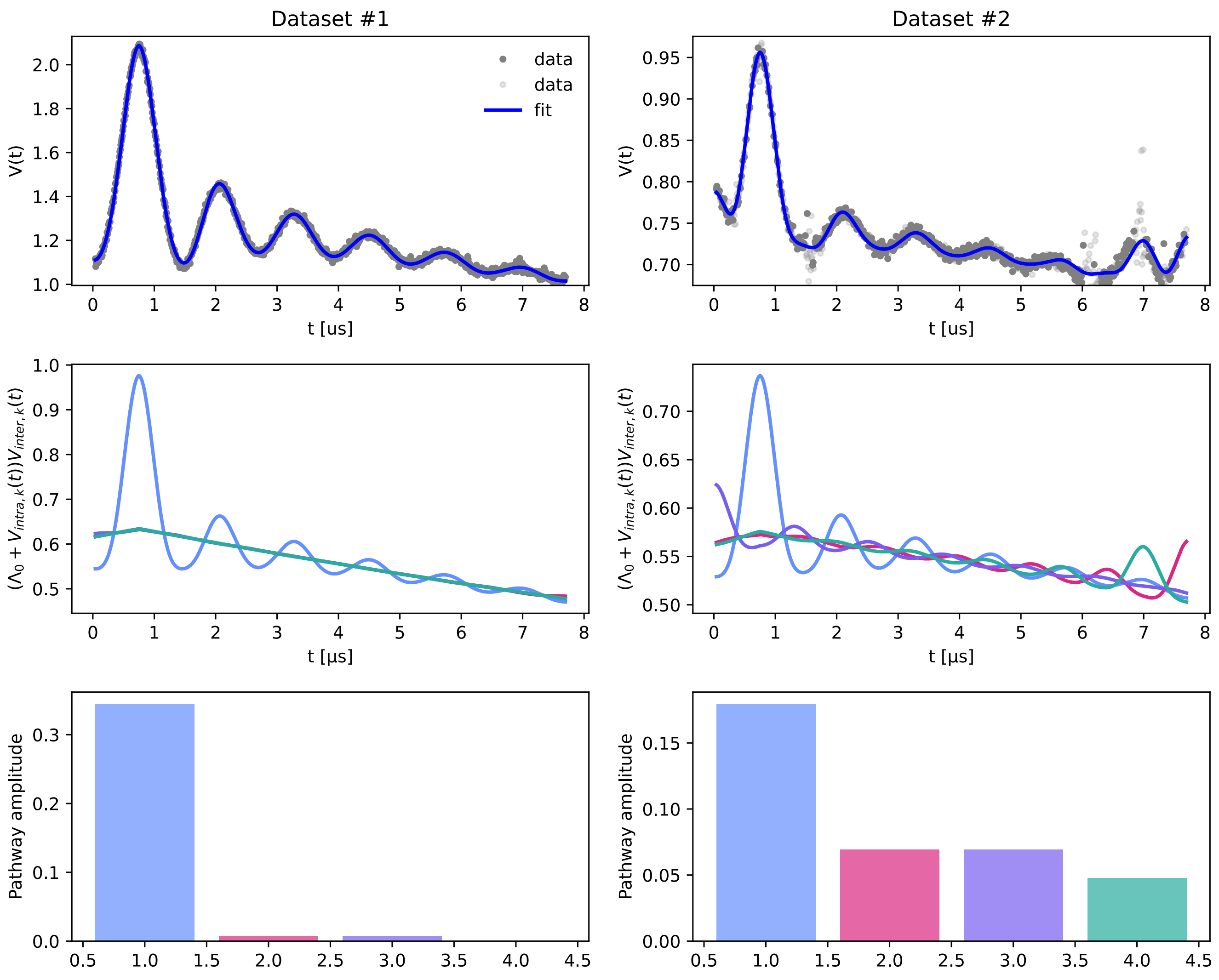


Figure S8 : Fitted orientation selection probability distributions

```python
# Plot the orientational distributions
for n in range(2):
    θ = np.linspace(0,np.pi/2,500)
    P = Pθs[n](θ)
    P = P/np.trapz(P,θ)
    plt.plot(θ,P)

plt.ylabel('P(θ)  [rad$^{-1}$]')
plt.xlabel('Inter-spin vector orientation θ [rad]')
plt.legend(['Dataset #1','Dataset #2'])
plt.tight_layout()
plt.show()
```

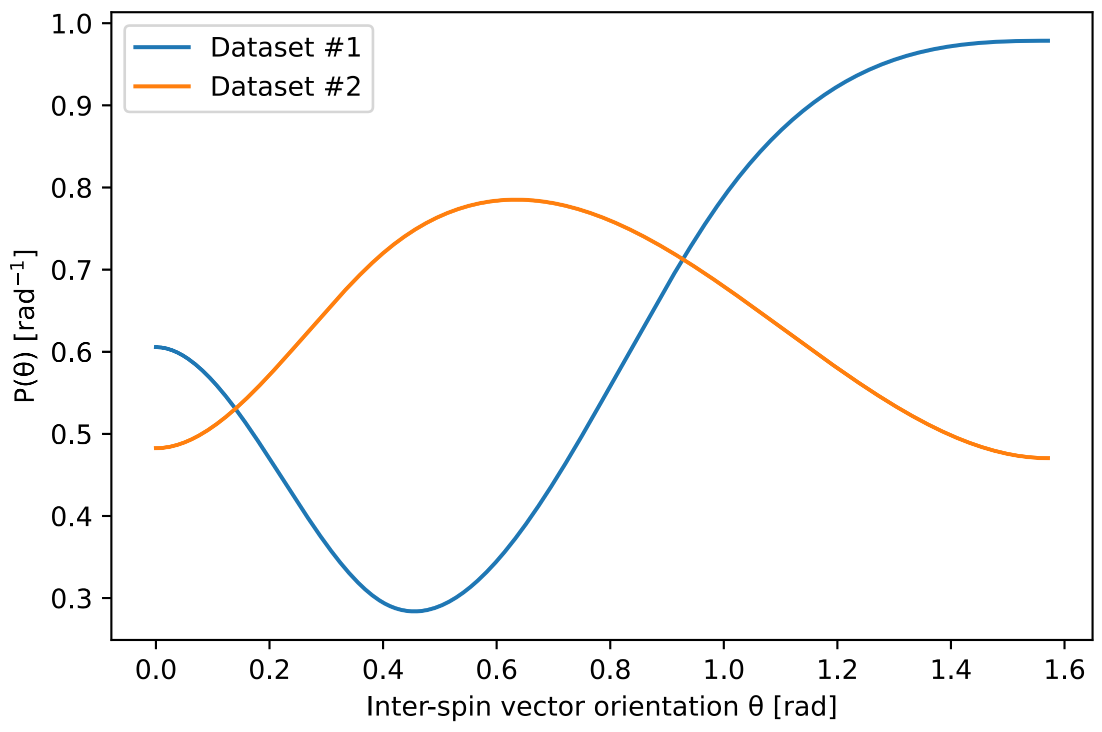
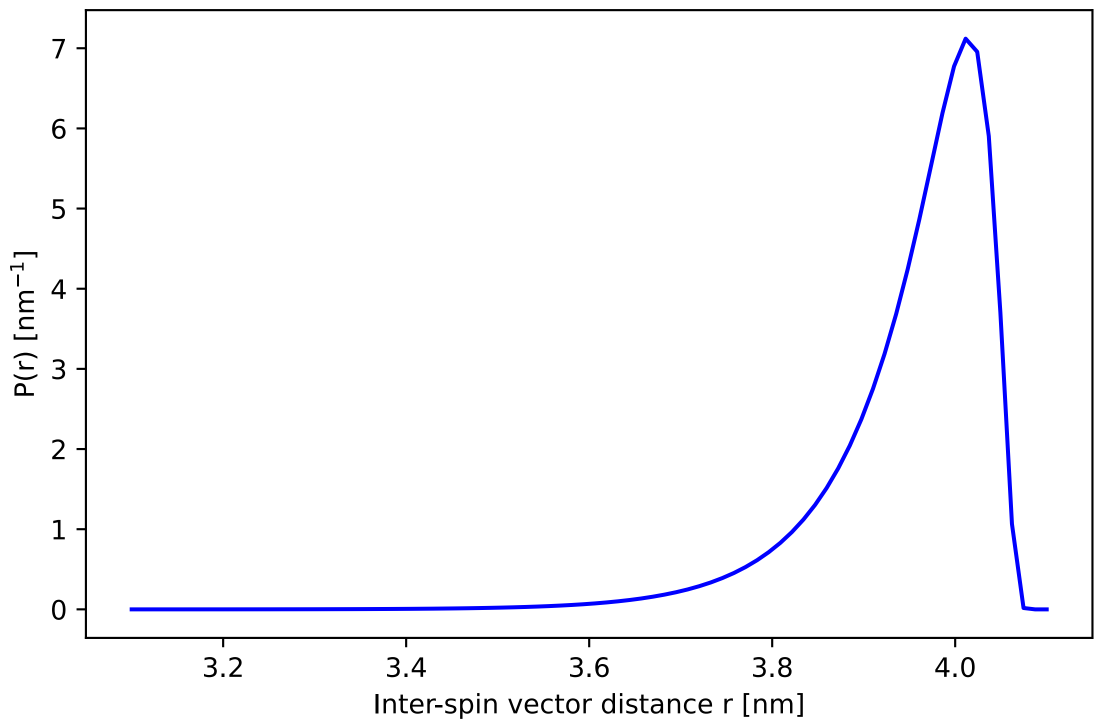


Figure S9 : Globally fitted (parametric) distance distribution

```
# Plot the distance distributions
Pr = results.evaluate(dl.dd_wormchain,r)

plt.plot(r,Pr,'b')

plt.ylabel('P(r) [nm$^{-1}$]')
plt.xlabel('Inter-spin vector distance r [nm]')
plt.tight_layout()
plt.show()
```

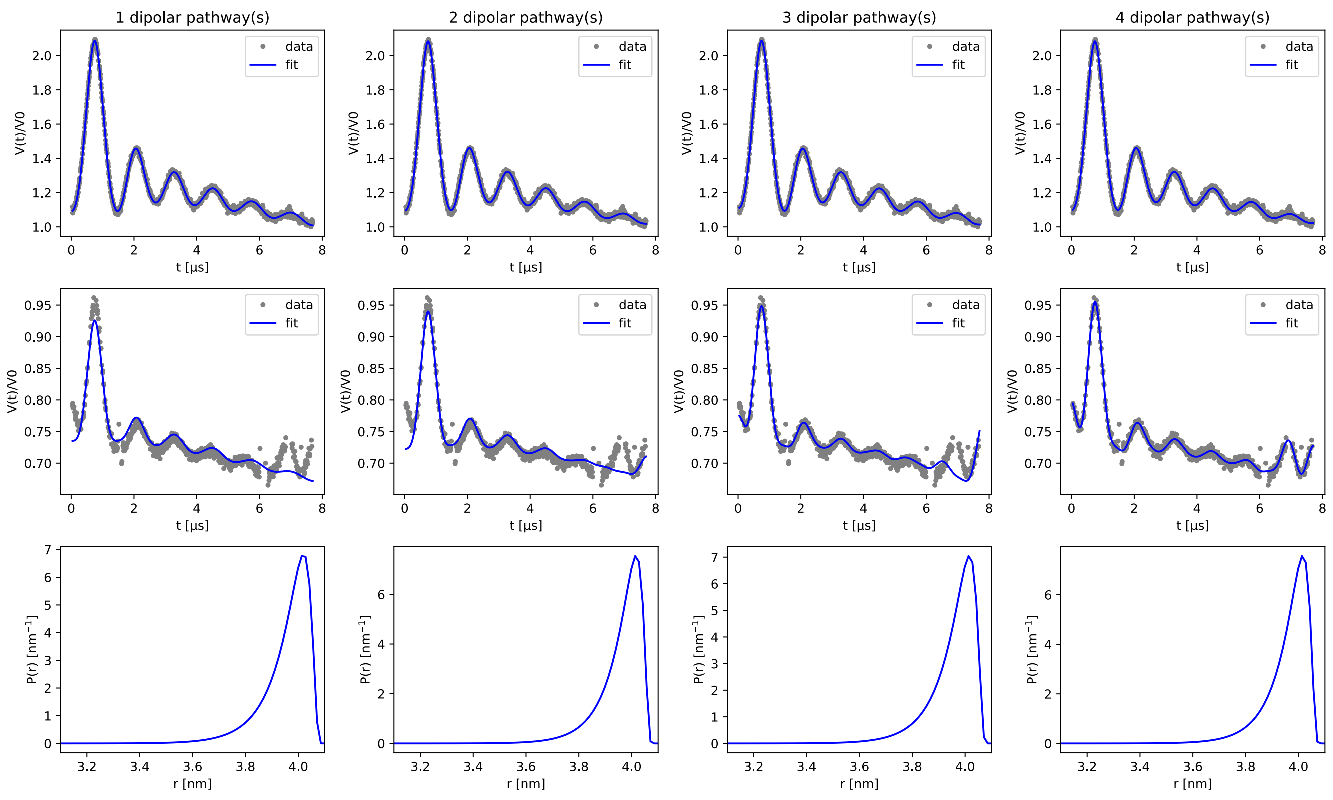


2.5. Effects of insufficient dipolar pathways in the model

This script exemplifies the effects in the fit of the data and in the resulting distance distribution when modelling a 4-pulse DEER signal with an insufficient number of dipolar pathways.

Figure S10 : Effects of insufficient dipolar pathways in the model

```
maxpaths = [1,2,3,4]
plt.figure(figsize=[15,9])
for n,npathways in enumerate(maxpaths):
    # Interspin vector distances
    r = np.linspace(3.1,4.1,70)

    Vmodels = []
    for t,tau1,tau2,P0 in zip(ts,tau1s,tau2s,P0s):
        # Construct information on the pathways-refocusing times based on experiment
        experimentInfo = dl.ex_4pdeer(tau1,tau2)
        # Construct the dipolar signal model
        Vmodel = dl.dipolarmodel(t, r, Pmodel=dl.dd_wormchain, npathways=npathways,
                                        experiment=experimentInfo, orisel=P0)
        # Freeze the refocusing times at the theoretical values
        Vmodels.append(Vmodel)

    # Create a global model
    Vglobal = dl.merge(*Vmodels)
    # Make the spin concentration and distance distribution global to all submodels
    Vglobal = dl.link(Vglobal,
                    conc=['conc_1','conc_2'],
                    contour=['contour_1','contour_2'],
                    persistence=['persistence_1','persistence_2'])
    Vglobal.conc.set(lb=10, ub=500, par0=80)
    Vglobal.contour.set(lb=3.00, ub=5.00, par0=4.00)
    Vglobal.persistence.set(lb=0.001, ub=100, par0=19)

    if npathways==1:
        Vglobal = dl.link(Vglobal,**{f'reftime': [f'reftime_1',f'reftime_2']})
    else:
        for i in range(npathways):
            Vglobal = dl.link(Vglobal,**{f'reftime{i+1}': [f'reftime{i+1}_1',f'refti

    # Fit all experiments datasets to the global model
    npathsresult = dl.fit(Vglobal, Vs, mask=masks, reg=False)

    Pfit = dl.dd_wormchain(r,contour=npathsresult.contour,persistence=npathsresult.p

    for j in range(2):
        plt.subplot(3,len(maxpaths),n+1+j*len(maxpaths))
        plt.plot(ts[j][masks[j]],Vs[j][masks[j]],'.',color='gray')
        plt.plot(ts[j],npathsresult.model[j],'b')
        plt.legend(['data','fit'])
        plt.xlabel('t [μs]')
        plt.ylabel('V(t)/V0')
        if j==0:
            plt.title(f'{n+1} dipolar pathway(s)')

    # Plot the globally fitted non-parametric distance distribution
    plt.subplot(3,len(maxpaths),n+1+2*len(maxpaths))
    plt.plot(r,Pfit,'b')
    plt.xlabel('r [nm]')
    plt.ylabel('P(r) [nm$^{-1}$]')
    plt.xlim([min(r),max(r)])

plt.tight_layout()
plt.show()
```



3. Experiments on a WALP23 mutant

3.1 About

The 5-pulse DEER experiments were performed on a A7R1/W22R1 mutant of the membrane-inserting peptide WALP23 by Breitgoff et al. All information concerning sample prepration and experiments were reported by Breitgoff et al. in a past publication (Phys. Chem. Chem. Phys., 2017,19, 15766-15779).

3.2 Analysis of the 5-pulse DEER datasets

The signals were then analyzed using DeerLab 0.13.2 in Python. After loading the data, the dipolar traces phases were optimized using the `deerlab.correctphase()` function. As mentioned in the main text, we modelled the experimental signals to account for four modulated dipolar pathways of the 5-pulse DEER experiments. To account for the relatively long pump pulses we allowed the refocusing times to vary up to 100 ns from their theoretical values. The distance distribution was modelled as a non-parametric distribution obtained via Tikhonov regularization with a fixed regularization parameter value of α=0.05. The nonlinear parameters and the distance distribution were fitted to the data via separable non-linear least-squares using the function `deerlab.snlls()` from DeerLab in Python.

The following script was used for the full analysis of the 4-pulse DEER experimental datasets:

```python
# ==========================================================
# LOADING & PREPROCESSING
# ==========================================================

# Path to folder containing experimental data
path = r'data\WALP_5pDEER_50K_Qband'

files = ['WALP_7_22_5pDEER_HS16_long.DTA',
         'WALP_7_22_5pDEER_HS16_short.DTA']

# Pulse sequence timings for all experiments
τ1s = [2.0, 1.5] # μs
τ2s = [2.0, 1.5] # μs
τ3s = [0.7, 0.2] # μs
deadtimes = [0.1, 0.1] # μs
# Pre-allocate containers
Vs, ts = [],[]

# Loading and pre-processing
Ndat = len(files)

# Load the first experimental datasets
t,Vexp = dl.deerload(os.path.join(path,files[0]))
# Remove the signal drop at the edge of the dataset
Vexp = Vexp[50:-25]
t = t[50:-25] - np.min(t)
# Add to list of datasets
ts.append(t)
Vs.append(Vexp)

# Load the second experimental datasets
t,Vexp = dl.deerload(os.path.join(path,files[1]))
# Extract just the first dataset
Vexp = Vexp[:,0]
t = t[0] - np.min(t[0])
# Add to list of datasets
ts.append(t)
Vs.append(Vexp)

# Phase correction
Vs = [dl.correctphase(V) for V in Vs]
# Normalization (aesthetic)
Vs = [V/np.max(V) for V in Vs]
# Deadtime shift
ts = [t + deadtime for t,deadtime in zip(ts,deadtimes)]
```

```python
# Interspin vector distances
r = np.linspace(1,4.5,90)

Vmodels = []
for t,τ1,τ2,τ3 in zip(ts,τ1s,τ2s,τ3s):
    # Construct information on the pathways-refocusing times based on experiment
    experimentInfo = dl.ex_5pdeer(τ1,τ2,τ3)
    # Construct the dipolar signal model
    Vmodel = dl.dipolarmodel(t,r,npathways=5,experiment=experimentInfo)
    # The amplitudes of pathways #3 and #4 must be equal
    Vmodel = dl.link(Vmodel,lam34 = ['lam3','lam4'])

    Vmodels.append(Vmodel)

# Create a global model
globalmodel = dl.merge(*Vmodels)

# Make distance distribution global to all submodels
globalmodel = dl.link(globalmodel,  P=['P_1','P_2'])

# Include edges in the regularization operator
L = dl.regoperator(r,2,includeedges=False)

# Fit all experiments datasets to the global model
results = dl.fit(globalmodel,Vs,bootstrap=500,regop=L,regparamrange=[1e-2,1e1])
```

Table S4 : Fit results and fitted model parameters of the 5-pulse DEER WALP datasets

```python
print(results)
```

```
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced 𝛘2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.001        4.395      0.003   -5085.304
   #2         0.004        1.270      0.004   -4023.850
========= ============= ============ ======= ===========
Model parameters:
============ ========= ========================= ======= ============================
==========
 Parameter    Value     95%-Confidence interval   Units   Description
============ ========= ========================= ======= ============================
==========
 lam1_1       0.275     (0.272,0.277)                     Amplitude of pathway #1
 reftime1_1   0.648     (0.648,0.649)               μs     Refocusing time of pathway
#1
 lam2_1       0.073     (0.071,0.075)                     Amplitude of pathway #2
 reftime2_1   2.005     (2.003,2.007)               μs     Refocusing time of pathway
#2
 lam34_1      0.023     (0.022,0.025)                     Amplitude of pathway #3
 reftime3_1   1.399     (1.396,1.400)               μs     Refocusing time of pathway
#3
 reftime4_1   2.713     (2.708,2.718)               μs     Refocusing time of pathway
#4
 lam5_1       0.006     (0.004,0.008)                     Amplitude of pathway #5
 reftime5_1   3.394     (3.377,3.400)               μs     Refocusing time of pathway
#5
 conc_1       288.469   (284.800,292.564)           μM     Spin concentration
 lam1_2       0.289     (0.287,0.292)                     Amplitude of pathway #1
 reftime1_2   0.166     (0.165,0.166)               μs     Refocusing time of pathway
```

```
#1
 lam2_2        0.125     (0.122,0.127)                    Amplitude of pathway #2
 reftime2_2    1.457     (1.456,1.459)            μs      Refocusing time of pathway
#2
 lam34_2       0.007     (0.006,0.009)                    Amplitude of pathway #3
 reftime3_2    1.230     (1.204,1.251)            μs      Refocusing time of pathway
#3
 reftime4_2    1.645     (1.610,1.673)            μs      Refocusing time of pathway
#4
 lam5_2        0.004     (0.002,0.007)                    Amplitude of pathway #5
 reftime5_2    2.782     (2.734,2.838)            μs      Refocusing time of pathway
#5
 conc_2        254.787   (248.855,261.347)        μM      Spin concentration
 P             ...       (...,...)               None     Non-parametric distance dis
tribution
============ ========= ========================= ======= ============================
----------
```

Figure S11 : Results of the global analysis of the WALP23 5-pulse DEER measurements

```
lams_list = [
    [results.lam1_1,results.lam2_1,results.lam34_1,results.lam34_1,results.lam5_1],
    [results.lam1_2,results.lam2_2,results.lam34_2,results.lam34_2,results.lam5_2],
    ]
reftimes_list = [
    [results.reftime1_1,results.reftime2_1,results.reftime3_1,results.reftime4_1,resu
    [results.reftime1_2,results.reftime2_2,results.reftime3_2,results.reftime4_2,resu
    ]
concs = [results.conc_1,results.conc_2]

masks = [np.ones_like(V).astype(bool) for V in Vs]

plt.figure(figsize=[9,6])
display_pathway_analysis(results.P,ts,Vs,results.model,masks,lams_list,reftimes_list
```

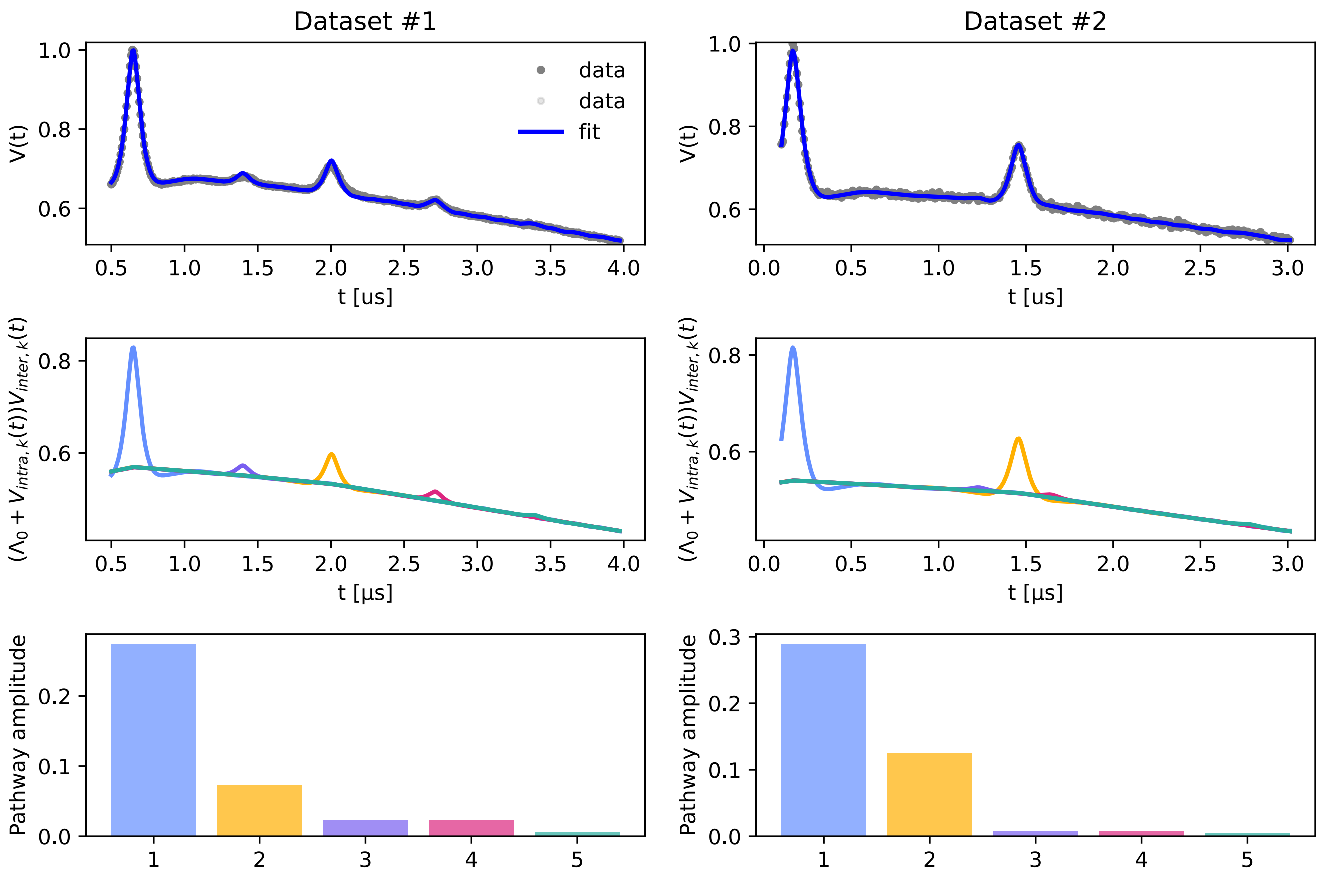
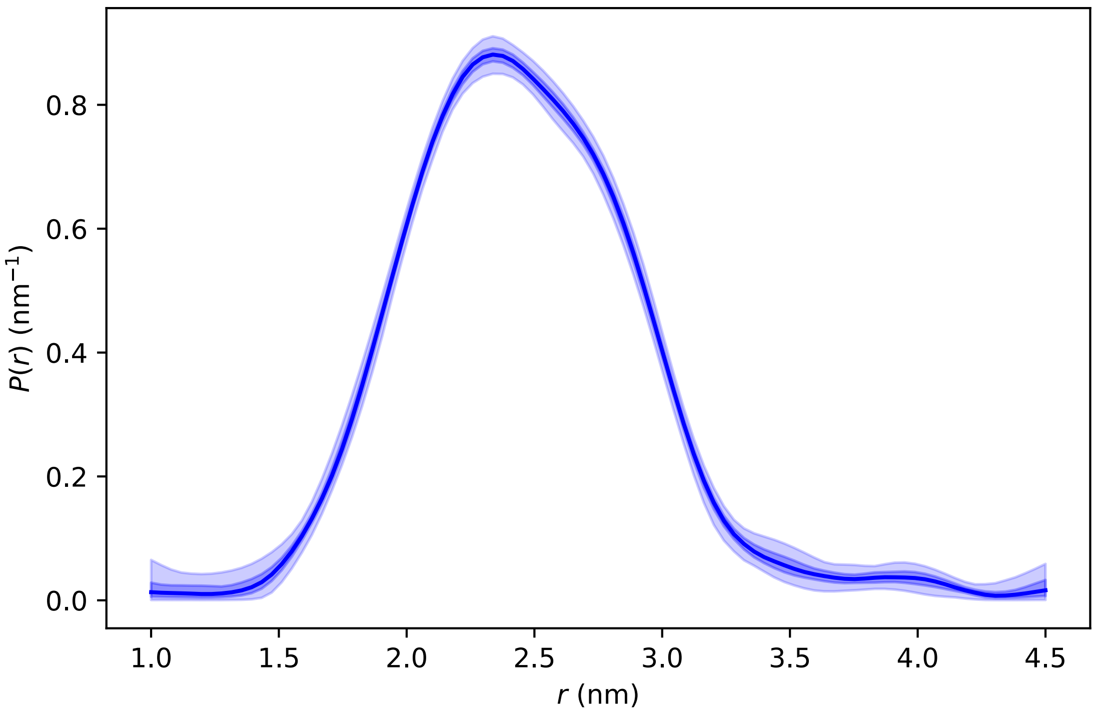


Figure S12: Globally fitted (non-parametric) distance distribution

```
Pfit = results.P
scale = np.trapz(Pfit,r)
Pfit = Pfit/scale
# Plot the fitted global distance distribution
plt.plot(r,Pfit,color='b')
# Plot the uncertainty of the distance distribution
Pci95 = results.PUncert.ci(95)/scale
Pci50 = results.PUncert.ci(50)/scale
plt.fill_between(r,Pci50[:,0],Pci50[:,1],alpha=0.3,color='b')
plt.fill_between(r,Pci95[:,0],Pci95[:,1],alpha=0.2,color='b')
plt.xlabel('$r$ (nm)')
plt.ylabel('$P(r)$ (nm$^{-1}$)')
plt.tight_layout()
plt.show()
```

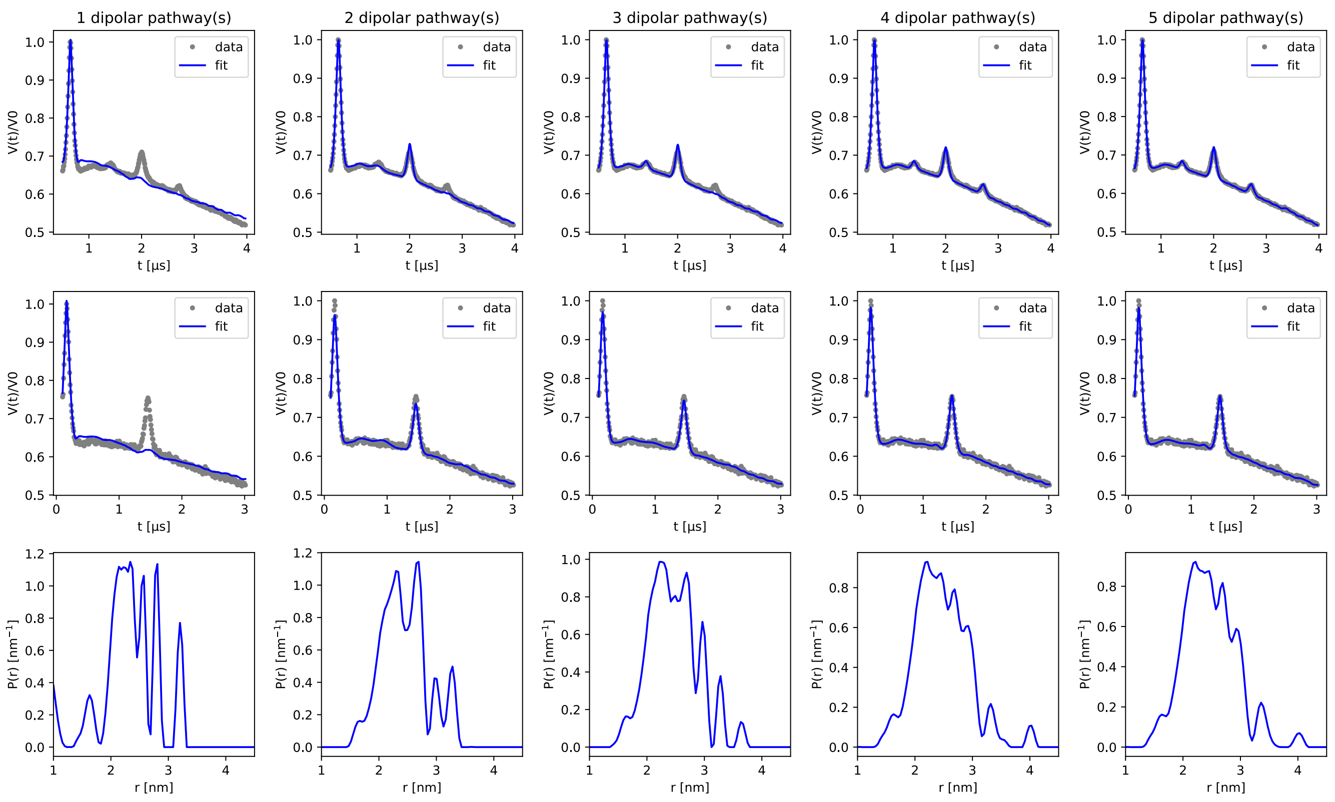


3.3. Effects of insufficient dipolar pathways in the model

This script exemplifies the effects in the fit of the data and in the resulting distance distribution when modelling a 5-pulse DEER signal with an insufficient number of dipolar pathways.

Figure S13 : Effects of insufficient dipolar pathways in the 5-pulse DEER model on the WALP dataset analysis

```
maxpaths = [1,2,3,4,5]
plt.figure(figsize=[15,9])
for n,npathways in enumerate(maxpaths):
    # Interspin vector distances
    r = np.linspace(1,4.5,90)

    Vmodels = []
    for t,τ1,τ2,τ3 in zip(ts,τ1s,τ2s,τ3s):
        # Construct information on the pathways-refocusing times based on experiment
        experimentInfo = dl.ex_5pdeer(τ1,τ2,τ3)
        # Construct the dipolar signal model
        Vmodel = dl.dipolarmodel(t,r,npathways=npathways,experiment=experimentInfo)

        Vmodels.append(Vmodel)

    # Create a global model
    globalmodel = dl.merge(*Vmodels)

    # Make distance distribution global to all submodels
    globalmodel = dl.link(globalmodel,P=['P_1','P_2'])

    # Include edges in the regularization operator
    L = dl.regoperator(np.arange(len(r)),2,includeedges=False)

    # Fit all experiments datasets to the global model

    npathsresult = dl.fit(globalmodel,Vs,regop=L,regparamrange=[1e-0,1e1])

    Pfit = npathsresult.P/np.trapz(npathsresult.P,r)

    plt.subplot(3,len(maxpaths),n+1)
    plt.plot(ts[0],Vs[0],'.',color='gray')
    plt.plot(ts[0],npathsresult.model[0],'b')
    plt.legend(['data','fit'])
    plt.xlabel('t [µs]')
    plt.ylabel('V(t)/V0')
    plt.title(f'{n+1} dipolar pathway(s)')

    plt.subplot(3,len(maxpaths),n+1+len(maxpaths))
    plt.plot(ts[1],Vs[1],'.',color='gray')
    plt.plot(ts[1],npathsresult.model[1],'b')
    plt.legend(['data','fit'])
    plt.xlabel('t [µs]')
    plt.ylabel('V(t)/V0')

    # Plot the globally fitted non-parametric distance distribution
    plt.subplot(3,len(maxpaths),n+1+2*len(maxpaths))
    plt.plot(r,Pfit,'b')
    plt.xlabel('r [nm]')
    plt.ylabel('P(r) [nm$^{-1}$]')
    plt.xlim([min(r),max(r)])

plt.tight_layout()
plt.show()
```



4. Experiments on TEMPOL

4.1. Equipment

The pulsed EPR experiments were performed at Q-band on a home-built spectrometer based on a Keysight Arbitrary Waveform Generator (AWG) with a high-power Q-band extension and equipped with TWT amplifier with 200 W nominal power. A helium flow cryostat (ER 4118 CF, Oxford Instruments) was used to adjust and stabilize the measurement temperature at 50 K.

A home-built box cavity resonator suitable for 1.6 mm o.d. sample tubes was used operating at about 34.5 GHz with a bandwidth in the range of 150-350 MHz. The resonator profile at the experimental conditions is:

4.2. Reverse 5-pulse DEER measurements

Using the pulses defined above the 5-pulse DEER experiment was performed according to the sequence:

$(\pi/2)_{probe}$ - $[\tau_1]$ - $(\pi)_{probe}$-$[t]$ - $(\pi)_{pump}$ - $[\tau_1+\tau_2-t]$ - $(\pi)_{probe}$ - $[\tau_3]$ - $(\pi)_{pump}$ - $[\tau_2-\tau_3]$ -echo

with τ_1 = 3200 ns, τ_2 = 3500 ns, and τ_3 = 1500 ns. The dipolar time t was swept from 500 ns to 5700 ns in 16 ns steps.

All probe pulses were realized with rectangular pulses with settings:

Pulse	Type	Frequency	Amplitude	Length
$(\pi)_{probe}$	rectangular	34.485 GHz	3.38 MHz	32 ns

Pulse	Type	Frequency	Amplitude	Length
$(\pi/2)_{probe}$	rectangular	34.485 GHz	3.38 MHz	16 ns

As described in the main text different pump pulse configurations were used for the same probe-pulse configuration:

	Pulse	Type	Frequency	Amplitude	Length
#1	$(\pi)_{pump}$	rectangular	34.4 GHz	2.8 MHz	8 ns
#2	$(\pi)_{pump}$	rectangular	34.4 GHz	4.7 MHz	32 ns
#3	$(\pi)_{pump}$	rectangular	34.4 GHz	14.1 MHz	8 ns
#4	$(\pi)_{pump}$	rectangular	34.4 GHz	14.1 MHz	32 ns
#5	$(\pi)_{pump}$	chirp	34.25-34.55 GHz	15.0 MHz	100 ns

*The chirp pulses were generated with a 10 ns rise time.

The following phase cycle was used:

Pulse																											
$(\pi/2)_{probe}$	x	x	x	x	-x	-x	-x	-x	x	x	x	x	-x	-x	-x	-x	x	x	x	x	-x	-x	-x	-x	x	x	x
$(\pi)_{pump,1}$	x	x	x	x	x	x	x	x	y	y	y	y	y	y	y	y	-x	-x	-x	-x	-x	-x	-x	-x	-y	-y	-y
$(\pi)_{pump,2}$	x	y	-x	-y	x	y	-x	-y	x	y	-x	-y	x	y	-x	-y	x	y	-x	-y	x	y	-x	-y	x	y	-x
Det. sign	+	+	+	+	-	-	-	-	+	+	+	+	-	-	-	-	+	+	+	+	-	-	-	-	+	+	+

4.3. Analysis of the 5-pulse DEER datasets

Prior to any analysis, all measured echoes were down converted from the local oscillator frequency (33 GHz), phase corrected and integrated over the detection window using home-written MATLAB scripts. The resulting complex-valued dipolar signals and their corresponding dipolar time axes were then exported as BES3T-formatted files for better portability. The signals were then analyzed using DeerLab 0.14.dev in Python. After loading the data, the dipolar traces phases were optimized using the `deerlab.correctphase()` function.

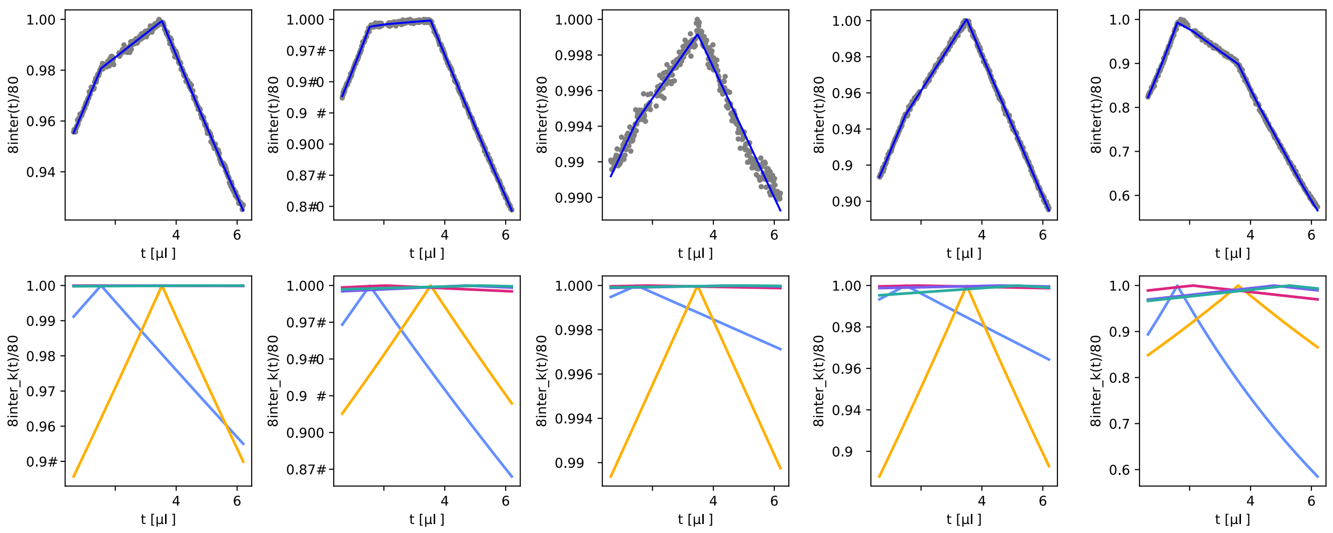
The intermolecular contributions were modeled using the `deerlab.bg_hom3d()` parametric function.

Figure S14 : Results of the analysis of the TEMPOL 5-pulse DEER measurements

```python
import warnings
warnings.filterwarnings("ignore")

# Path to folder containing experimental data
path = './data/TEMPOL_5pDEER_50K_Qband/'

# Experimental data files
files = [
        'TEMPOL_5pDEERr_rect_32ns_low_amp.txt',
        'TEMPOL_5pDEERr_rect_32ns_high_amp.txt',
        'TEMPOL_5pDEERr_rect_8ns_low_amp.txt',
        'TEMPOL_5pDEERr_rect_8ns_high_amp.txt',
        'TEMPOL_5pDEERr_chirp.txt'
]
ovls = [10,17,21,21,35]
# 5-pulse DEER sequence timings
τ1 = 3.2 # µs
τ2 = 3.5 # µs
τ3 = 1.5 # µs

# Spin concentration of the TEMPOL solution
spinconc = 400 # µM

# Refocusing times of the dipolar pathways
tref = np.array([
        τ3,         # Pathway #1
        τ2,         # Pathway #2
        τ2-τ3,      # Pathway #3
        τ1+τ3,      # Pathway #4
        τ1+τ2-τ3]) # Pathway #5

# Initialize the figure
plt.figure(figsize=[14,8])

# Loop over all experiments
for i,filename in enumerate(files):

        # Load the experimental data
        _,Vr,Vi = np.loadtxt(path+filename,unpack=True)

        # Construct the complex-valued experimental signal
        V = Vr + 1j*Vi
        # Phase correction of the experimental signal
        V = dl.correctphase(V)

        # Construct dipolar time-vector based on the experiment settings
        N = len(V)  # number of points
        dt = 0.016 # µs, step size
        t0 = 0.500 # µs, initial delay
        t = τ1 + τ2 - np.linspace(t0,t0+dt*N,N)

        # Flip the signal (due to error in the spectrometer storage)
        mask = np.ones_like(V).astype(bool)
        mask[np.arange(255,265,1)] = False

        t = t[mask]
        V = V[mask]

        # Rescale the signal w.r.t. its maximum
        V /= max(V)

        # Model of the experimental data
```

```
                        V *= dl.bg_hom3d(t-tref[n]-tshift,spinconc,λs[n])
                return V
        Vmodel = dl.Model(Vintermolecular)
        Vmodel.lam1.set(lb=0,ub=1,par0=0.1, description='Amplitude of pathway #1')
        Vmodel.lam2.set(lb=0,ub=1,par0=0.1, description='Amplitude of pathway #2')
        Vmodel.lam34.set(lb=0,ub=1,par0=0.1, description='Amplitude of pathway #3')
        Vmodel.lam5.set(lb=0,ub=1,par0=0.1, description='Amplitude of pathway #4')
        Vmodel.tshift.set(lb=-0.1,ub=0.1,par0=0, description='Time shift', units='µs

        result = dl.fit(Vmodel,V,multistart=5,bootstrap=500)

        print(f'===========================================================')
        print(f'Dataset #{i+1}')
        print(f'===========================================================')
        print(result)
        print(f'')
        print(f'')

        # Loop over all dipolar pathways
        λs = [result.lam1,result.lam2,result.lam34,result.lam34,result.lam5]

        Vinterk = []
        for n in range(Npathways):
                # Intermolecular contribution is a product over all pathway contribu
                Vinterk.append(dl.bg_hom3d(t-tref[n]-result.tshift,spinconc,λs[n]))

        # Plot the data and the fits
        plt.subplot(3,5,i+1)
        plt.plot(t,V,'.',color='grey')
        plt.plot(t,result.model,'b')
        plt.xlabel('t [µs]')
        plt.ylabel('Vinter(t)/V0')

        plt.subplot(3,5,i+6)
        for Vinter,color in zip(Vinterk,colors_5pdeer):
                plt.plot(t,Vinter,color=color,linewidth=2)
        plt.xlabel('t [µs]')
        plt.ylabel('Vinter_k(t)/V0')
plt.tight_layout()
plt.savefig('TEMPOL_analysis_results.svg')
plt.show()
```

```
===========================================================
Dataset #1
===========================================================
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced 𝛘2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.001         1.440      0.001   -4835.654
========= ============= ============ ======= ===========
Model parameters:
=========== ======= ========================= ======= =========================
 Parameter   Value   95%-Confidence interval   Units   Description
=========== ======= ========================= ======= =========================
 lam1        0.025   (0.024,0.025)             None    Amplitude of pathway #1
 lam2        0.048   (0.048,0.049)             None    Amplitude of pathway #2
 lam34       0.000   (0.000,0.001)             None    Amplitude of pathway #3
 lam5        0.000   (0.000,0.001)             None    Amplitude of pathway #4
 tshift      0.033   (0.024,0.043)              μs     Time shift
 scale       1.019   (1.019,1.020)             None    Scaling factor
=========== ======= ========================= ======= =========================


==========================================================
Dataset #2
==========================================================
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced χ2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.001        0.884      0.001   -4736.826
========= ============= ============ ======= ===========
Model parameters:
=========== ======= ========================= ======= =========================
 Parameter   Value   95%-Confidence interval   Units   Description
=========== ======= ========================= ======= =========================
 lam1        0.075   (0.073,0.077)             None    Amplitude of pathway #1
 lam2        0.079   (0.078,0.080)             None    Amplitude of pathway #2
 lam34       0.002   (0.001,0.004)             None    Amplitude of pathway #3
 lam5        0.001   (0.000,0.003)             None    Amplitude of pathway #4
 tshift      0.036   (0.030,0.043)              μs     Time shift
 scale       1.064   (1.063,1.066)             None    Scaling factor
=========== ======= ========================= ======= =========================


==========================================================
Dataset #3
==========================================================
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced χ2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.000        2.301      0.000   -5206.140
========= ============= ============ ======= ===========
Model parameters:
=========== ======== ========================= ======= =========================
 Parameter   Value    95%-Confidence interval   Units   Description
=========== ======== ========================= ======= =========================
 lam1        0.002    (0.001,0.002)             None    Amplitude of pathway #1
 lam2        0.008    (0.007,0.008)             None    Amplitude of pathway #2
 lam34       0.000    (0.000,0.000)             None    Amplitude of pathway #3
 lam5        0.000    (0.000,0.000)             None    Amplitude of pathway #4
 tshift      -0.021   (-0.056,0.031)             μs     Time shift
 scale       1.000    (1.000,1.001)             None    Scaling factor
=========== ======== ========================= ======= =========================


==========================================================
Dataset #4
==========================================================
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced χ2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.001        0.931      0.001   -4833.112
========= ============= ============ ======= ===========
Model parameters:
=========== ======== ========================= ======= =========================
 Parameter   Value    95%-Confidence interval   Units   Description
=========== ======== ========================= ======= =========================
 lam1        0.019    (0.018,0.021)             None    Amplitude of pathway #1
 lam2        0.085    (0.084,0.085)             None    Amplitude of pathway #2
 lam34       0.001    (0.000,0.002)             None    Amplitude of pathway #3
 lam5        0.003    (0.001,0.004)             None    Amplitude of pathway #4
 tshift      -0.002   (-0.007,0.002)             μs     Time shift
 scale       1.019    (1.017,1.020)             None    Scaling factor
=========== ======== ========================= ======= =========================


==========================================================
Dataset #5
==========================================================
Goodness-of-fit:
========= ============= ============ ======= ===========
 Dataset   Noise level   Reduced χ2   RMSD       AIC
========= ============= ============ ======= ===========
   #1         0.003         1.620      0.004   -3815.139
========= ============= ============ ======= ===========
Model parameters:
=========== ======= ========================= ======= =========================
 Parameter   Value   95%-Confidence interval   Units   Description
=========== ======= ========================= ======= =========================
 lam1        0.292   (0.284,0.298)             None    Amplitude of pathway #1
 lam2        0.139   (0.136,0.142)             None    Amplitude of pathway #2
 lam34       0.019   (0.012,0.025)             None    Amplitude of pathway #3
 lam5        0.018   (0.010,0.025)             None    Amplitude of pathway #4
 tshift      0.097   (0.091,0.100)              μs     Time shift
 scale       1.172   (1.162,1.179)             None    Scaling factor
=========== ======= ========================= ======= =========================
```


8inter(t)/80
8inter_k(t)/80
t [μl]