Supporting Information for:

Quantitative Prediction of Oil–Water Interfacial Tension in Surfactant Systems Using Dissipative Particle Dynamics

Authors: Rachel Hendrikse,[a],* Carlos Amador[b], Matthew Davies[b] and Mark R. Wilson[a]
[a] *Department of Chemistry, Durham University, Lower Mountjoy, Stockton Road, Durham, DH1 3LE, United Kingdom.*
[b] *Procter and Gamble, Newcastle Innovation Centre, Whitley Road, Newcastle upon Tyne, NE12 9BZ, UK.*
∗ Author for correspondence. E-mail: rachel.hendrikse@durham.ac.uk

1 Relation between molecule length and self-interaction

In the main article, we discuss that the self-interaction needs to be adjusted to account for pressure and density changes that result from bonding beads together. This is also discussed extensively in our previous work.[5] Fig. S1 shows the relation between the number of bonded beads and the interaction parameter required to produce the same pressure as unbonded beads ($P = 23.7$), using the parameters specified in this work.

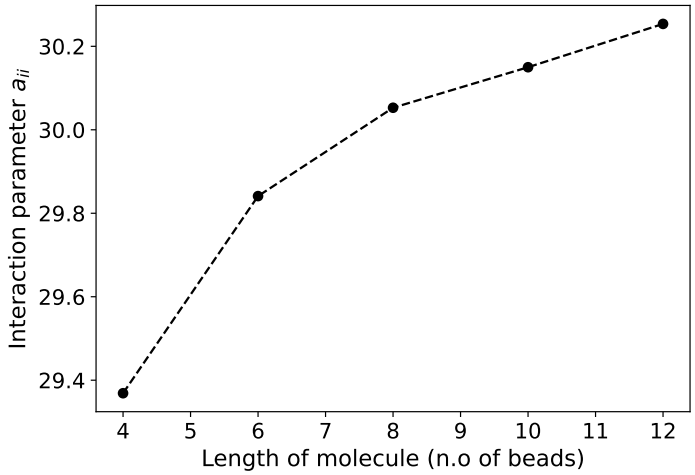

Fig. S1: Relation between the self-interaction parameter and the length of DPD molecules used in this work.

2 DDAO head group interaction with dodecane

In the main article, it was specified that the DDAO head group interaction with dodecane is set to reproduce the same excess chemical potential of a single water molecule in dodecane. This is a simplification for obtaining this interaction, where we expect that the head-water interaction is more important for the properties we study in this article. However, here we provide extra calculations supporting the value for the head-dodecane interaction value used for this interaction.

There is a lack of experimental data available in the literature providing water-dodecane partition coefficients that could otherwise be used to define the DDAO head-oil interaction. However, a value for the octanol-water partition coefficient of N,N-Dimethylpentylamine N-oxide is reported as $\log P = -0.55$,[4] implying that N,N-Dimethylpentylamine N-oxide is more favourable in water than octanol. By using the excess chemical potential values calculated for the hydrocarbon tail in water from the main article, leads us to conclude that the oil/head interaction should have $\Delta\mu_{\mathrm{ex}}^{\mathrm{TMAO}}(\mathrm{TMAO} \rightarrow \mathrm{Octanol}) = 11.6083 k_B T$. However, it is typical that octanol-water partition coefficients are slightly larger than water-dodecane partition coefficients,[9] so it is likely that the actual excess chemical potential of the

head group in the dodecane phase should be slightly lower. This would be in agreement with the value used in the main article.

3 Automated parameterisation script

```
import numpy as np
from scipy.optimize import minimize

def uex(A):
    if A<70:
        return 2.8247792635296888e-05*A*A*A-0.006391263343923913*A*A+0.6285768677015161*A
    else:
        return 0.12443454*A+13.84332426

def OverLapVolumeNew(BL,List):
    V3=(2*np.pi/3)*(2+BL)*(1-BL)**2
    V2=(np.pi/12)*(4*1+BL)*((2*1-BL)**2)
    V1=(4*np.pi/3)
    MuList=[]
    for g in range(len(List)):
        if g==0:
            Con=(V1-V2)*uex(List[g])+(V2-V3)*0.5*uex(List[g]+List[g+1])+(1/3)*V3*uex(List[g]+List[g+1]+List[g+2])
        elif g==(len(List)-1):
            Con=(V1-V2)*uex(List[g])+(V2-V3)*0.5*uex(List[g]+List[g-1])+(1/3)*V3*uex(List[g]+List[g-1]+List[g-2])
        elif g==1:
            Con=(V1-V2*2+V3)*uex(List[g])+(V2-V3)*0.5*uex(List[g]+List[g-1])+(V2-2*V3)*0.5*uex(List[g]+List[g+1])+\
                    (1/3)*V3*uex(List[g]+List[g-1]+List[g+1])+(1/3)*V3*uex(List[g]+List[g+2]+List[g+1])
        elif g==(len(List)-2):
            Con=(V1-V2*2+V3)*uex(List[g])+0.5*(V2-V3)*uex(List[g]+List[g+1])+0.5*(V2-2*V3)*uex(List[g]+List[g-1])+\
                    (1/3)*V3*uex(List[g]+List[g-1]+List[g+1])+(1/3)*V3*uex(List[g]+List[g-2]+List[g-1])
        else:
            Con=(V1-2*V2+V3)*uex(List[g])+0.5*(V2-2*V3)*uex(List[g]+List[g+1])+0.5*(V2-2*V3)*uex(List[g]+List[g-1])\
                    +(1/3)*V3*(uex(List[g]+List[g-2]+List[g-1]) + uex(List[g]+List[g-1]+List[g+1]) + \
                    uex(List[g]+List[g+1]+List[g+2]))
        MuList.append(Con)
    return (3/(4*np.pi))*np.array(MuList)

def uexd(A):
    if A<70:
        return 2.7923251201648353e-05*A*A*A-0.006264534737997436*A*A+0.6294923486350743*A
    else:
        return 0.13535057*A+14.12938424

def OverLapVolumeNewD(BL,List):
    V3=(2*np.pi/3)*(2+BL)*(1-BL)**2
    V2=(np.pi/12)*(4*1+BL)*((2*1-BL)**2)
    V1=(4*np.pi/3)
    MuList=[]
    for g in range(len(List)):
        if g==0:
            Con=(V1-V2)*uexd(List[g])+(V2-V3)*0.5*uexd(List[g]+List[g+1])+(1/3)*V3*uexd(List[g]+List[g+1]+List[g+2])
        elif g==(len(List)-1):
            Con=(V1-V2)*uexd(List[g])+(V2-V3)*0.5*uexd(List[g]+List[g-1])+(1/3)*V3*uexd(List[g]+List[g-1]+List[g-2])
        elif g==1:
            Con=(V1-V2*2+V3)*uexd(List[g])+(V2-V3)*0.5*uexd(List[g]+List[g-1])+(V2-2*V3)*0.5*uexd(List[g]+List[g+1])+\
                    (1/3)*V3*uexd(List[g]+List[g-1]+List[g+1])+(1/3)*V3*uexd(List[g]+List[g+2]+List[g+1])
        elif g==(len(List)-2):
            Con=(V1-V2*2+V3)*uexd(List[g])+0.5*(V2-V3)*uexd(List[g]+List[g+1])+0.5*(V2-2*V3)*uexd(List[g]+List[g-1])+\
                    (1/3)*V3*uexd(List[g]+List[g-1]+List[g+1])+(1/3)*V3*uexd(List[g]+List[g-2]+List[g-1])
        else:
            Con=(V1-2*V2+V3)*uexd(List[g])+0.5*(V2-2*V3)*uexd(List[g]+List[g+1])+\
                    0.5*(V2-2*V3)*uexd(List[g]+List[g-1])+(1/3)*V3*(uexd(List[g]+List[g-2]+List[g-1]) + \
                    uexd(List[g]+List[g-1]+List[g+1]) + uexd(List[g]+List[g+1]+List[g+2]))
        MuList.append(Con)
    return (3/(4*np.pi))*np.array(MuList)

NumberEO=8 # Replace with the number of head group beads (EO/EOT)
NumberC3=4 # Replace with the number of tail beads  (C3/C3T)
Solvent='Water' # Replace with either 'Water' or 'Dodecane'
Total=NumberEO+NumberC3
if Total==4:
    AIJ=29.4
elif Total==6:
```

```
    AIJ=29.8
elif Total==8:
    AIJ=30.1
elif Total==10:
    AIJ=30.1
elif Total==12:
    AIJ=30.3

Self=[AIJ]*Total
InWa=[-4.96]
for h in range(0,NumberEO-1):
    InWa.append(0.64)
for h in range(0,NumberC3-1):
    InWa.append(4.115)
InWa.append(5.445)

InDo=[4.1]
for h in range(0,NumberEO-1):
    InDo.append(1.74)
for h in range(0,NumberC3):
    InDo.append(0)

def calculate_chemical_potential(aij_list):
    if Solvent=='Water':
        return np.array(OverLapVolumeNew(0.6,aij_list)-OverLapVolumeNew(0.6,Self))
    elif Solvent=='Dodecane':
        return np.array(OverLapVolumeNewD(0.6,aij_list)-OverLapVolumeNew(0.6,Self))

if Solvent=='Water':
    target_sum_chunk1 = sum(InWa[0:2])
    target_sum_chunk2 = sum(InWa[2:NumberEO])
    target_sum_chunk3 = sum(InWa[NumberEO:(len(InWa)-1)])
    target_sum_chunk4 = InWa[-1]

elif Solvent=='Dodecane':
    target_sum_chunk1 = sum(InDo[0:2])
    target_sum_chunk2 = sum(InDo[2:NumberEO])
    target_sum_chunk3 = sum(InDo[NumberEO:(len(InDo)-1)])
    target_sum_chunk4 = InDo[-1]

n=NumberEO-1

def objective_function(aij_list):
    calculated_potentials = calculate_chemical_potential(aij_list)
    sum_chunk1 = np.sum(calculated_potentials[:2])
    sum_chunk2 = np.sum(calculated_potentials[2:n+1])
    sum_chunk3 = np.sum(calculated_potentials[n+1:(len(calculated_potentials)-1)])
    sum_chunk4 = calculated_potentials[-1]
    error_chunk1 = (sum_chunk1 - target_sum_chunk1) ** 2
    error_chunk2 = (sum_chunk2 - target_sum_chunk2) ** 2
    error_chunk3 = (sum_chunk3 - target_sum_chunk3) ** 2
    error_chunk4 = (sum_chunk4 - target_sum_chunk4) ** 2
    return error_chunk1 + error_chunk2 + error_chunk3 + error_chunk4

def create_equality_constraints(n, m):
    return [{'type': 'eq', 'fun': lambda aij, i=i: aij[i] - aij[i+1]} for i in range(n, m)]

constraints = []
constraints += create_equality_constraints(1,NumberEO-1)
constraints += create_equality_constraints(NumberEO,NumberEO+NumberC3-2)

initial_aij = np.random.rand(len(InWa))

result = minimize(objective_function, initial_aij,  method='SLSQP', constraints=constraints)

if result.success:
    optimized_aij = result.x
    calculated_potentials = calculate_chemical_potential(optimized_aij)
    sum_chunk1 = np.sum(calculated_potentials[:2])
    sum_chunk2 = np.sum(calculated_potentials[2:n+1])
```

```
    sum_chunk3 = np.sum(calculated_potentials[n+1:(len(calculated_potentials)-1)])
    sum_chunk4 = calculated_potentials[-1]
    print("Optimization successful!")
    print("Optimized aij values:", optimized_aij)
    print("Calculated chemical potentials:", calculated_potentials)
    print(f"Sum of chunk 1 (first 2 beads): {sum_chunk1} (Target: {target_sum_chunk1})")
    print(f"Sum of chunk 2 (3 to {n+1}): {sum_chunk2} (Target: {target_sum_chunk2})")
    print(f"Sum of chunk 3 ({n+2} to end-1): {sum_chunk3} (Target: {target_sum_chunk3})")
    print(f"Sum of chunk 4 (final bead): {sum_chunk4} (Target: {target_sum_chunk4})")
else:
    print("Optimization failed:", result.message)
print(f'Self-interaction: {AIJ}')
```

4 a_{ij} parameters

Here we provide all of the a_{ij} parameters used for the various surfactants simulated in the main article.

4.1 C_6E_2

Surfactant coarse-graining: C3T-C3-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	29.4	-	-	-	-	-
C3	29.4	29.4	-	-	-	-
EO	39.2	39.2	29.4	-	-	-
EOT	52.6	52.6	29.4	29.4	-	-
W	57.4	79.3	32.7	5.3	25.0	-
Do	29.8	20.7	39.2	52.8	198.2	29.4

4.2 C_6E_4

Surfactant coarse-graining: C3T-C3-EO-EO-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	29.8	-	-	-	-	-
C3	29.8	29.8	-	-	-	-
EO	41.2	41.2	29.8	-	-	-
EOT	46.3	46.3	29.8	29.8	-	-
W	58.9	78.7	27.5	14.8	25.9	-
Do	29.9	21.7	41.2	46.3	198.2	29.4

4.3 C_6E_6

Surfactant coarse-graining: C3T-C3-EO-EO-EO-EO-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	30.1	-	-	-	-	-
C3	30.1	30.1	-	-	-	-
EO	39.8	39.8	30.1	-	-	-
EOT	48.3	48.3	30.1	30.1	-	-
W	59.8	76.6	31.5	12.5	25.0	-
Do	30.1	22.7	39.8	48.3	198.2	29.4

4.4 $C_{12}E_2$

Surfactant coarse-graining: C3T-C3-C3-C3-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	29.8	-	-	-	-	-
C3	29.8	29.8	-	-	-	-
EO	41.8	41.8	29.8	-	-	-
EOT	50.7	50.7	29.8	29.8	-	-
W	59.7	65.3	27.7	9.3	25.0	-
Do	29.7	25.8	41.8	50.7	198.2	29.4

4.5 $C_{12}E_4$

Surfactant coarse-graining: C3T-C3-C3-C3-EO-EO-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	30.1	-	-	-	-	-
C3	30.1	30.1	-	-	-	-
EO	40.9	40.9	30.1	-	-	-
EOT	47.3	47.3	30.1	30.1	-	-
W	60.5	64.8	29.6	13.7	25.0	-
Do	29.9	26.4	40.9	47.3	198.2	29.4

4.6 $C_{12}E_6$

Surfactant coarse-graining: C3T-C3-C3-C3-EO-EO-EO-EO-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	30.1	-	-	-	-	-
C3	30.1	30.1	-	-	-	-
EO	39.5	39.5	30.1	-	-	-
EOT	48.6	48.6	30.1	30.1	-	-
W	60.7	64.4	32.3	12.0	25.0	-
Do	29.9	26.6	39.5	48.6	198.2	29.4

4.7 $C_{12}E_8$

Surfactant coarse-graining: C3T-C3-C3-C3-EO-EO-EO-EO-EO-EO-EO-EOT.

	C3T	C3	EO	EOT	W	Do
C3T	30.3	-	-	-	-	-
C3	30.3	30.3	-	-	-	-
EO	39.4	39.4	30.3	-	-	-
EOT	49.2	49.2	30.3	30.3	-	-
W	61.6	64.5	33.2	11.7	25.0	-
Do	30.0	26.8	39.4	49.2	198.2	29.4

4.8 Sodium dodecyl sulfate (SDS)

Surfactant coarse-graining: C3T-C3-C3-C3-$[SO_4]^{-1}$ $[Na]^{+1}$

	C3T	C3	SO4 Head	Na Ion	W	Do
C3T	29.7	-	-	-	-	-
C3	29.7	29.7	-	-	-	-
SO4 Head	78.0	78.0	29.7	-	-	-
Na Ion	60.0	64.0	22.8	25.0	-	-
W	60.0	64.0	22.8	25.0	25.0	-
Do	29.5	28.4	78.0	198.2	198.2	29.4

4.9 Dodecyldimethylamine oxide (DDAO)

Surfactant coarse-graining: C3T-C3-C3-C3-AO

	C3T	C3	AO Head	W	Do
C3T	29.7	-	-	-	-
C3	29.7	29.7	-	-	-
AO Head	78.0	78.0	29.7	-	-
W	60.0	64.0	26.2	25.0	-
Do	29.5	28.4	78.0	198.2	29.4

4.10 DDAO/SDS mixing parameters

(AO Head) - (SO4 Head): $a_{ij} = 29.7$
(AO Head) - (Na Ion): $a_{ij} = 26.2$

5 Bond and angle parameters

5.1 Angle Distributions

As stated in the main article, numerous studies are available in literature,[1–3,7,8] in which the authors produce a coarse-grain model via mapping to atomistic simulations. Therefore, we utilise atomistic simulations available in the existing literature to aid in testing our choice for angle parameter K_A. We choose to use $K_A = 5$ for all angles, allowing us to more easily utilise results from our previous works[5] than if we had chosen a different value (e.g. altering K_A alters the pressure and therefore equilibrium density of the fluid).

The bonding potential between successive C3 units can be studied using the results of atomistic simulations of nonane, provided in An *et al.*[1] We vary the strength of angle parameter K_A in DPD simulations of nonane to investigate the effect on the angle distribution, which is shown in Fig. S2. We find that our choice of $K_A = 5$ provides a reasonably good match to the angle distribution obtained from the atomistic simulations. However, a slightly closer match could be achieved by using $K_A = 3$.

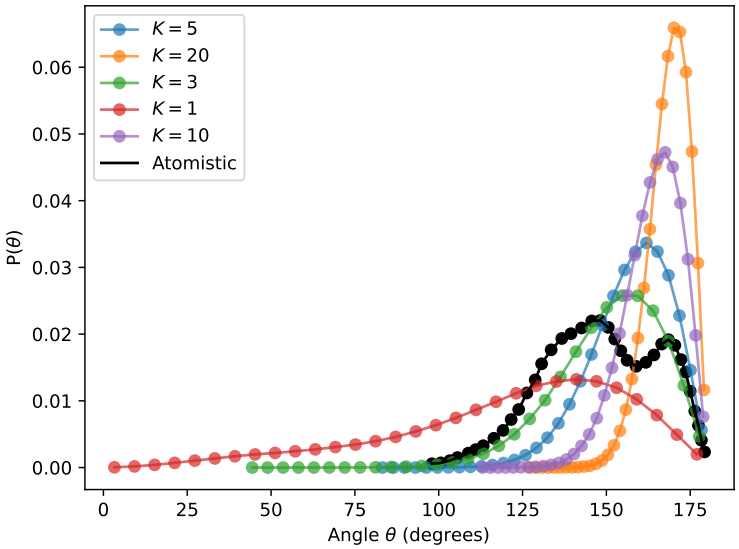

Fig. S2: The angle distribution in DPD simulations (3 bonded beads) with varying angle parameter K_A, compared with that from atomistic simulations of nonane. Atomistic data taken from An *et al.*[1]

For the PEG chains, all-atom simulations using CHARM c33b2 provided in Lee *et al.*[7] provide the expected angle distribution between PEO units. Similarly to the previous case, we vary the angle parameter K_A in simulations of 6 bonded PEG beads to obtain the angle distributions, as shown in Fig. S3. Once again, we find $K_A = 5$ provides a reasonably good match to the atomistic simulations. However, a slightly closer match is achieved using $K_A = 10$.

For each case, the mean average angle is shown in Table S.1, where we see that there is minimal difference in the average angle between $K_A = 3$ and $K_A = 5$ for nonane, and likewise between $K_A = 5$ and $K_A = 10$ for PEG. Therefore, we suspect that the exact choice of angle has only minimal impact on the simulation results, which will now be tested in section 5.2.

5.2 Impact of K_A on IFT

As stated, we choose to use $K_A = 5$ for all angles. We show here the impact that this choice has on the IFT results.

We initially test this choice by calculating the IFT between pure water and pure dodecane. We stress here that we still use our previous work[5] to calculate the self-interaction of the dodecane, which was based on using $K_A = 5$. Therefore, there is a possibility that the density of the dodecane is slightly different to what is expected (where the target density is $\rho R_C^3 = 3$). Fig S4 shows the IFT is only slightly related to the choice of K_A, and once $2 \leq K_A$ there is a negligible difference in the IFT calculated.

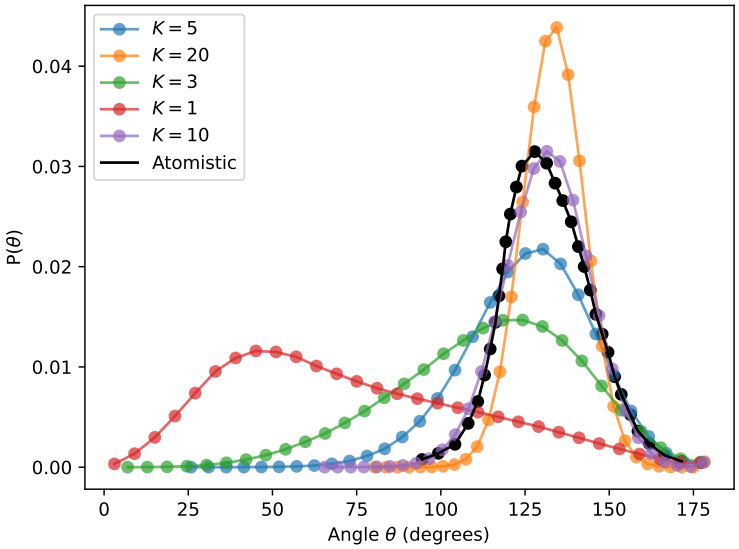

Fig. S3: The angle distribution in DPD simulations (6 bonded beads) with varying angle parameter K_A, compared with that from atomistic simulations of PEG. Atomistic data taken from Lee *et al.*[7] .

	K_A	Average angle (degrees)
PEG	1	73.3
	3	114.0
	5	126.1
	10	131.6
	20	133.4
	Atomistic	131.0
Nonane	1	121.7
	3	151.0
	5	157.4
	10	164.0
	20	168.6
	Atomistic	150.6

Table S.1: Average angles from distributions shown in Figs. S2 and S3

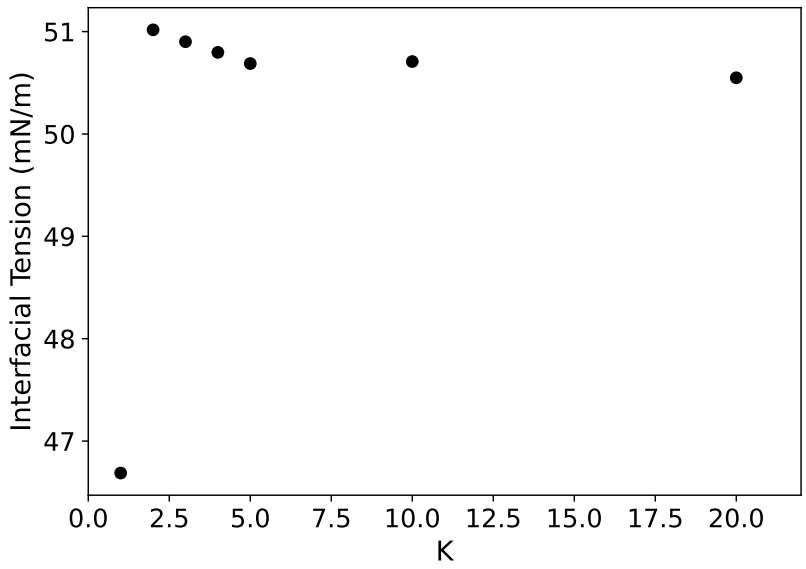

Fig. S4: IFT of water/dodecane systems for varying K_A.

Now we turn our attention to systems containing surfactants. We consider that it is possible that, in these cases, the choice for angle strength has more impact on the results, as it may influence the flexibility and packing of molecules at the interface. In section 5.1, we showed that more exact matches to the angle distributions obtained from atomistic simulations for the surfactants might be $K_A = 3$ for the hydrocarbon bonds and $K_A = 10$ for the PEG bonds. Here we test the impact of using these choices on a selection of simulation cases. We simulate surfactants C_6E_6, $C_{12}E_4$ and $C_{12}E_8$ and the effects of using the new angle parameters are shown in Tables S.2-S.4. We simulate a variety of initial surface concentrations to try and capture if there is any difference in the IFT at a given surface concentration, as well as if there is any change in the maximum packing expected. We see that there is generally minimal effect beyond the expected fluctuation between different simulation runs. There might be a slight increase in the surface concentration at the CMC with the altered parameters, but it is not a noteworthy change.

Initial surface concentration (molecules/nm^2)	Final surface concentration (molecules/nm^2)		IFT (mN/m)	
	Original	Updated	Original	Updated
1.16	1.16	1.16	20.8	21.3
1.74	1.73	1.74	2.4	3.2
2.90	1.71	1.74	3.3	3.4

Table S.2: C_6E_6

Initial surface concentration (molecules/nm^2)	Final surface concentration (molecules/nm^2)		IFT (mN/m)	
	Original	Updated	Original	Updated
1.45	1.45	1.45	13.4	13.5
2.61	2.60	2.61	-0.22	-0.28
3.49	1.86	1.93	0.30	-0.58

Table S.3: $C_{12}E_4$

Initial surface concentration (molecules/nm^2)	Final surface concentration (molecules/nm^2)		IFT (mN/m)	
	Original	Updated	Original	Updated
1.16	1.16	1.16	13.5	14.4
3.49	1.62	1.68	-0.45	-0.89

Table S.4: $C_{12}E_8$

6 The thickness of surfactant layers

6.1 Density profiles for surfactants at different surface coverages.

In figure S5, we provide examples of how the density of $C_{12}E_4$ surfactants can vary across the simulation box at different surface coverages. This is discussed more fully in section 5.3 of the main manuscript.

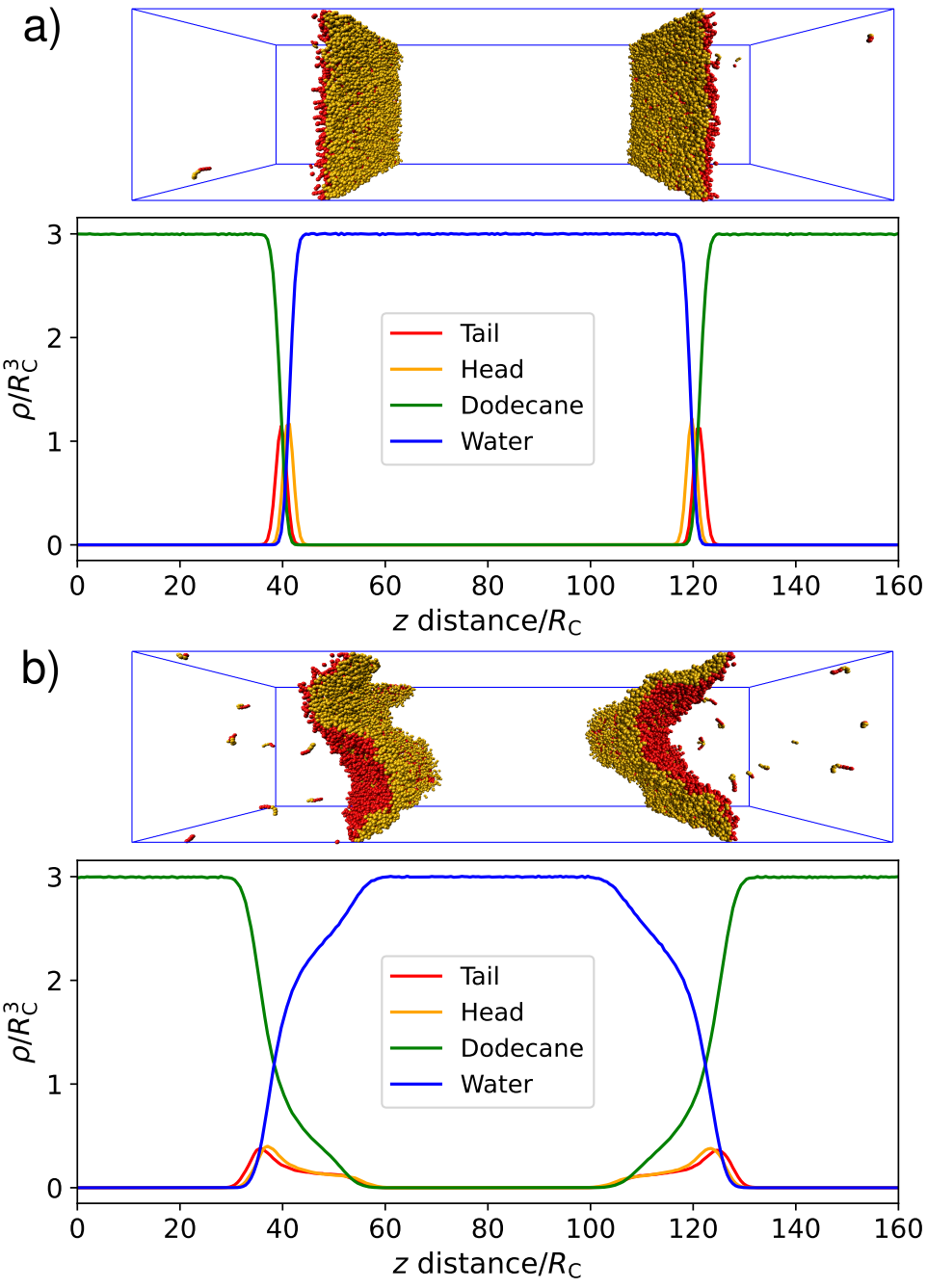

Fig. S5: Example density profiles for $C_{12}E_4$ surfactants at two different surface coverages a) 1.7 and b) 2.6 molecules/nm^2. Also shown are visual representations of the beads in the simulation box at the corresponding surface densities (created using VMD[6]). Beads are coloured according to type, where tail beads are red and head groups are orange.

6.2 Additional thickness calculations

Fig S6 shows all thickness values that were calculable from our simulations (i.e. cases where a Gaussian fit is possible). Note that the larger values are omitted from the figure in the main article as they distort the plot, and these values are also known to correspond to overpacked, unphysical layers.

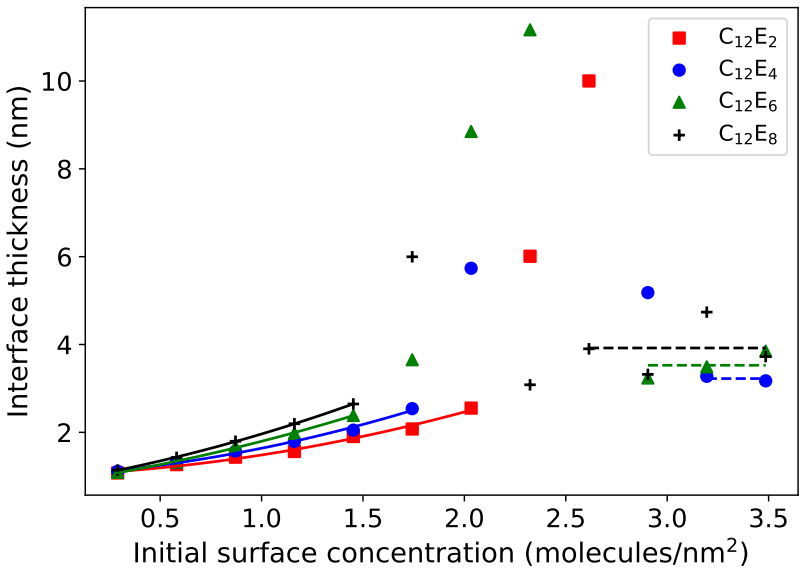

Fig. S6: All thickness values for $C_{12}E_j$ surfactant layers at the interface.

6.3 Polynomial fits

$$\text{layer thickness} = a\Gamma^2 + b\Gamma + c \tag{1}$$

where Γ is the surface concentration.

Surfactant	a	b	c
C_6E_2	0.16362684	-0.05608295	0.92544635
C_6E_4	0.15825607	0.29135147	0.73599205
C_6E_6	0.20449081	0.48197747	0.62722646
$C_{12}E_2$	0.26869938	0.17300848	1.04106116
$C_{12}E_4$	0.35810031	0.20978747	1.0529062
$C_{12}E_6$	0.21477936	0.7443947	0.84552478
$C_{12}E_8$	0.30009277	0.77790307	0.88476727

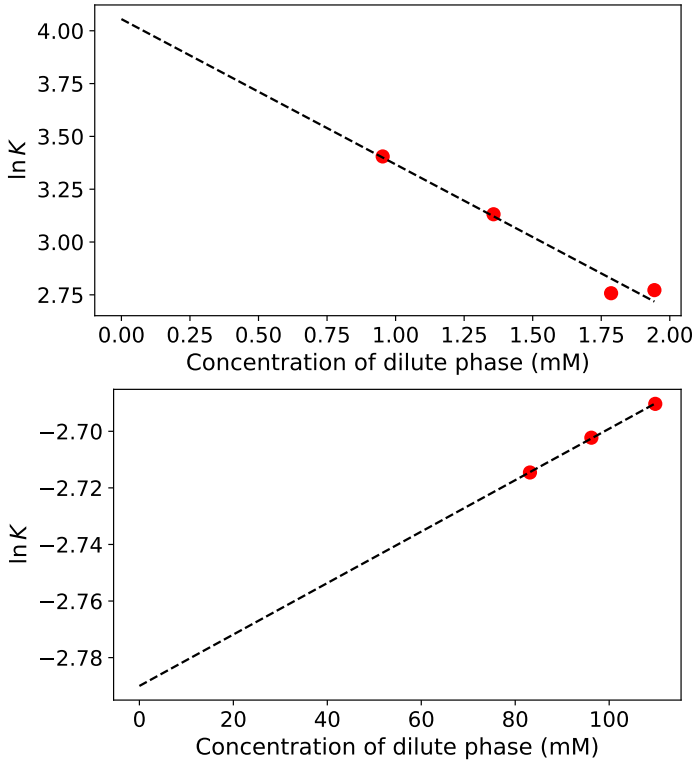

Fig. S7: Extrapolation of partitioning $\ln K$ at infinite dilution for $C_{12}E_6$ (top) and C_6E_4 (bottom) surfactants.

7 Partition coefficient calculation

7.1 a_{ij} values

As stated in the main article, we increase the self-interactions between the surfactants to $a_{ij} = 100$ to calculate a value for the partition coefficient. We perform this calculation on surfactant types C_6E_4, and $C_{12}E_6$. Tables S.5 and S.6 show the set of interaction parameters used in these calculations.

	C3T	C3	EO	EOT	W	Do
C3T	100	-	-	-	-	-
C3	100	100	-	-	-	-
EO	100	100	100	-	-	-
EOT	100	100	100	100	-	-
W	58.9	78.7	27.5	14.8	25.9	-
Do	29.9	21.7	41.2	46.3	198.2	29.4

Table S.5: C_6E_4

	C3T	C3	EO	EOT	W	Do
C3T	100	-	-	-	-	-
C3	100	100	-	-	-	-
EO	100	100	100	-	-	-
EOT	100	100	100	100	-	-
W	60.7	64.4	32.3	12.0	25.0	-
Do	29.9	26.6	39.5	48.6	198.2	29.4

Table S.6: $C_{12}E_6$

7.2 Determining K by extrapolation to infinite dilution

We calculate the partitioning value K at infinite dilution by varying the number of surfactants placed at the interface of the box and extrapolating to infinite dilution to calculate K. Fig. S7 shows this for two example surfactants $C_{12}E_6$ and C_6E_4 (where the final $\ln K$ values are given in the main article).

8 Relationship between IFT and simulation box size

We show here that the IFT calculated at a 'final', equilibrated surface coverage is independent of the box size. The box size dependence is only related to the IFT calculated when there is surfactant removal from the surface in the form of micelles.

Table S.7 shows the IFT calculated for two different box sizes, for various surfactants, at three different surface coverages. The surface coverages presented are the 'initialised' surface coverage values. For $\Gamma = 0.58$ and $\Gamma = 1.16$ molecules/nm^2, this surface concentration is below the maximum packing value for all surfactants. However, for $\Gamma = 1.74$ molecules/nm^2, we start to see differences between the two box sizes due to differences related to whether there is surfactant removal from the surface or not. However, when there is no removal ($\Gamma = 0.58$ and $\Gamma = 1.16$ molecules/nm^2) there is a negligible difference between the two box sizes used.

		IFT (mN/m)		
Surfactant	Box Size	$\Gamma = 0.58$ (molecules/nm^2)	$\Gamma = 1.16$ (molecules/nm^2)	$\Gamma = 1.74$ (molecules/nm^2)
$C_{12}E_2$	$60R_C \times 20R_C \times 20R_C$	43.8	29.5	10.7
	$160R_C \times 40R_C \times 40R_C$	43.4	29.4	11.1
$C_{12}E_4$	$60R_C \times 20R_C \times 20R_C$	40.6	23.2	2.98
	$160R_C \times 40R_C \times 40R_C$	40.3	23.1	3.38
$C_{12}E_6$	$60R_C \times 20R_C \times 20R_C$	36.7	17.8	-1.63
	$160R_C \times 40R_C \times 40R_C$	36.5	17.8	-0.621
$C_{12}E_8$	$60R_C \times 20R_C \times 20R_C$	32.8	13.5	-4.00
	$160R_C \times 40R_C \times 40R_C$	32.6	13.5	-1.07

Table S.7: Calculated interfacial tension (IFT) for different C_iE_j surfactants, using two different box sizes, at three initialised surface concentrations Γ.

9 DDAO micelle formation

In the main article, we highlighted that DDAO prefers to move into the oil phase, as opposed to the water phase. Here, we show that this is not because DDAO is incapable of forming micelles, and is because the excess chemical potential is lower in the oil phase. We simulate DDAO in pure water (box dimensions $20R_C \times 20R_C \times 20R_C$), and the position of the DDAO molecules after equilibration is shown in Fig. S8.

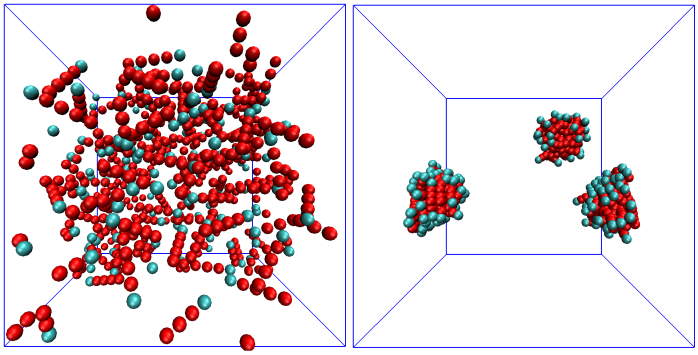

Fig. S8: DDAO molecules in a cubic box with periodic boundary conditions both before (left) and post (right) equilibration. Note that water is excluded for clarity.

10 Effect of altering head-water interaction on DDAO

In the main article, we discussed how altering the head-water interaction for DDAO (reducing the a_{ij} parameter to be the same as that used for SDS-water) does not lead to a greater reduction in IFT per molecule. However it does lead to an increase in the maximum packing of surfactants at the interface, and therefore an overall lower minimum IFT.

These simulations were conducted in a box with dimensions $160R_C \times 40R_C \times 40R_C$, though we did not perform any finite size studies for this case. The results of these simulations, compared to those originally presented in the main article, are shown in Fig. S9

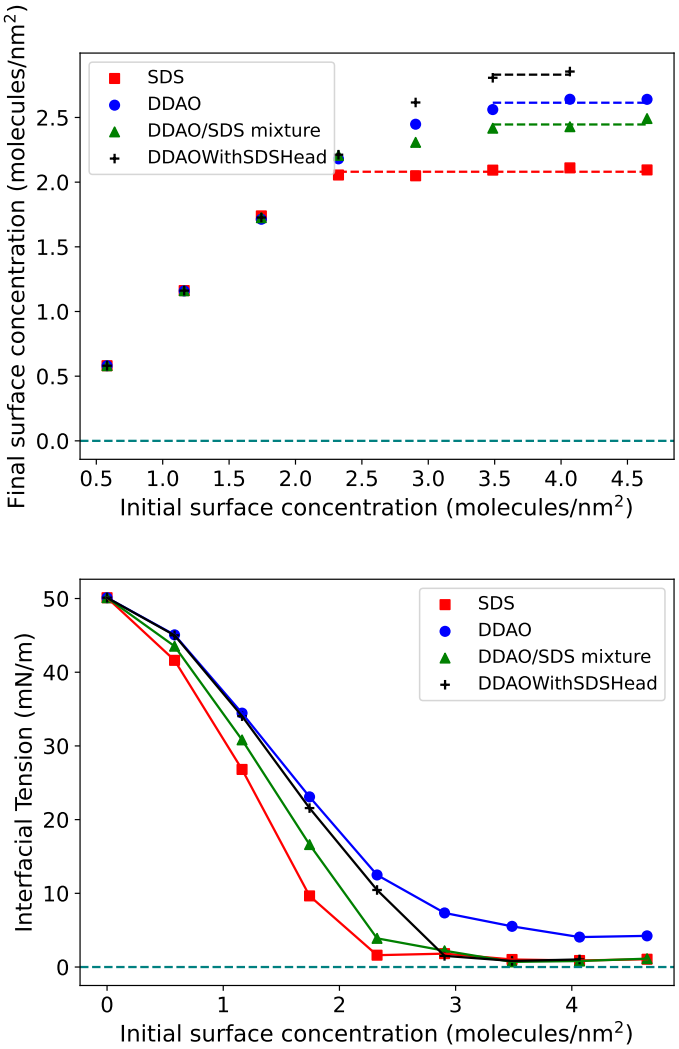

Fig. S9: Results for charged surfactants in test box size of $160R_C \times 40R_C \times 40R_C$. Figs show the final surface concentration Γ (top) and IFT (bottom) as a function of the initial surface concentration Γ_0 for SDS, DDAO and a 50:50 mixture of SDS and DDAO. We also show a case where we have tested simulating DDAO with the same head-water a_{ij} parameters as SDS.

References

[1] Y. An, K. K. Bejagam, and S. A. Deshmukh. Development of new transferable coarse-grained models of hydrocarbons. *The Journal of Physical Chemistry B*, 122(28):7143–7153, 2018. PMID: 29928806.

[2] R. L. Anderson, D. S. D. Gunn, T. Taddese, E. Lavagnini, P. B. Warren, and D. J. Bray. Phase behavior of alkyl ethoxylate surfactants in a dissipative particle dynamics model. *The Journal of Physical Chemistry B*, 127(7):1674–1687, 2023. PMID: 36786752.

[3] D. J. Bray, R. L. Anderson, P. B. Warren, and K. Lewtas. Wax formation in linear and branched alkanes with dissipative particle dynamics. *Journal of Chemical Theory and Computation*, 16(11):7109–7122, 2020. PMID: 32857939.

[4] G. Caron, G. Ermondi, D. Boschi, P.-A. Carrupt, R. Fruttero, B. Testa, and A. Gasco. Structure-property relationships in the basicity and lipophilicity of arylalkylamine oxides. *Helvetica Chimica Acta*, 82(10):1630–1639, 1999.

[5] R. L. Hendrikse, C. Amador, and M. R. Wilson. Dissipative particle dynamics parametrisation using infinite dilution activity coefficients: the impact of bonding. *Phys. Chem. Chem. Phys.*, 27:1554–1566, 2025.

[6] W. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[7] H. Lee, A. H. de Vries, S.-J. Marrink, and R. W. Pastor. A coarse-grained model for polyethylene oxide and polyethylene glycol: Conformation and hydrodynamics. *The Journal of Physical Chemistry B*, 113(40):13186–13194, 2009. PMID: 19754083.

[8] S. O. Nielsen, C. F. Lopez, G. Srinivas, and M. L. Klein. A coarse grain model for n-alkanes parameterized from surface tension data. *The Journal of Chemical Physics*, 119(14):7043–7049, 10 2003.

[9] N. E. Tayar, R.-S. Tsai, B. Testa, P.-A. Carrupt, and A. Leo. Partitioning of solutes in different solvent systems: The contribution of hydrogen-bonding capacity and polarity. *Journal of pharmaceutical sciences*, 80(6):590–598, 1991.