Supplemental Material on "Non-Gaussian statistics of concentration fluctuations in free liquid diffusion"

Marco Bussoletti and Mirko Gallo
Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, via Eudossiana 18, 00184 Rome, Italy

Amir Jafari
Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD, USA, 21218

Gregory L. Eyink
Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD, USA, 21218 and Department of Physics and Astronomy, The Johns Hopkins University, Baltimore, MD, USA, 21218

A. SHORT-TIME, LARGE-r ASYMPTOTIC SOLUTION FOR TRIPLE CUMULANT

We first describe the variables used to represent $\mathcal{C}_{123}$, in terms of the geometry of the triangle with vertices $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. We take as reference the triangle with sides r_{12}, r_{13}, r_{23} and interior angles $\psi_{12}, \psi_{13}, \psi_{23}$ located in the first quadrant of the xy-plane with vertex 3 at the origin and side $\mathbf{r}_{13}$ along the x-axis. See Figure 1. Applying rotation $R(\alpha, \beta, \gamma) = R_z(\alpha)R_x(\beta)R_z(\gamma)$ around point 3 and then translating point 3 by (X, Y, Z), the general position and orientation of the triangle as a rigid body in space is specified by 9 variables: 3 side-lengths r_{12}, r_{13}, r_{23}, 3 Euler angles α, β, γ, and 3 coordinates (X, Y, Z) of vertex 3. The number of independent variables is reduced from 9 to 6 by using translational symmetry along the planar interface (eliminating X, Y) and rotational symmetry around the gradient direction (eliminating α). Transforming Cartesian variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ to these new variables, therefore, the 3rd order cumulant $\mathcal{C}_{123}$ can be written entirely in terms of $r_{12}, r_{23}, r_{13}, \beta, \gamma, Z$. The shape variables are subject to the triangle inequalities $r_{pq} \leq r_{ik} + r_{jk}$ for i, j, k any cyclic permutation of $1, 2, 3$. It is thus more convenient to use unconstrained variables $r_{13}, r_{23}, \psi_{12}$ to specify triangle shape and recover the others from the trigonometric laws of cosines and sines:

$$r_{12} = (r_{13}^2 + r_{23}^2 - 2r_{13}r_{23}\cos(\psi_{12}))^{1/2}; \quad \psi_{23} = \arcsin(r_{23}\sin(\psi_{12})/r_{12}); \quad \psi_{13} = \arcsin(r_{13}\sin(\psi_{12})/r_{12}) \tag{A.1}$$

Triangle orientation is specified by rotating coordinate vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$, by $R_x(\beta)R_z(\gamma)$ into a new orthogonal system

$$\mathbf{l} = (\cos\gamma, \cos\beta\sin\gamma, \sin\beta\sin\gamma)^\top; \quad \mathbf{m} = (-\sin\gamma, \cos\beta\cos\gamma, \sin\beta\cos\gamma)^\top; \quad \mathbf{n} = (0, -\sin\beta, \cos\beta)^\top, \tag{A.2}$$

which then gives the triangle side vectors $\mathbf{r}_{13}$, $\mathbf{r}_{23}$, $\mathbf{r}_{12}$, in general position:

$$\mathbf{r}_{13} = r_{13}\mathbf{l}; \quad \mathbf{r}_{23} = r_{23}(\cos\psi_{12}\mathbf{l} + \sin\psi_{12}\mathbf{m}); \quad \mathbf{r}_{12} = r_{12}(\cos\psi_{23}\mathbf{l} - \sin\psi_{23}\mathbf{m}). \tag{A.3}$$

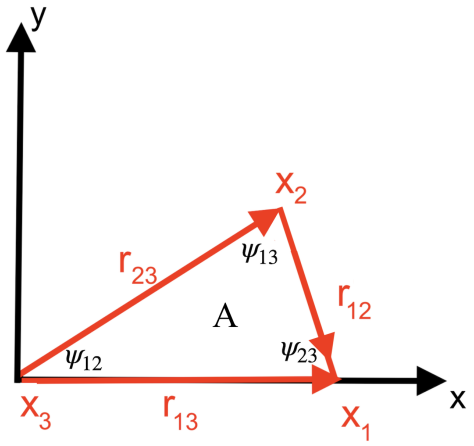

FIG. 1 Illustration of the triangle formed by vertices $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, in reference position in the horizontal plane.

The variables used to represent the second cumulant $\mathcal{C}_{pq}$ for the two points $\mathbf{x}_p$, $\mathbf{x}_q$ are the radial distance r_{pq}, the polar angle θ_{pq}, and the mean height $Z_{pq} = \frac{1}{2}(z_p + z_q)$. The forcing terms in equation (4) of the Main Text for $\mathcal{C}_{123}$ involve the gradients of the second cumulants and thus we must relate its variables to those that we have adopted for $\mathcal{C}_{123}$. The $z-$coordinates of the triangle side vectors can be represented also by the polar angles relative to vector $\hat{\mathbf{z}}$

$$z_{13} = r_{13}\cos\theta_{13};\ z_{23} = r_{23}\cos\theta_{23};\ z_{12} = r_{12}\cos\theta_{12}, \tag{A.4}$$

so that comparing with equation (A.3) yields:

$$\cos\theta_{13} = \sin\beta\sin\gamma; \quad \cos\theta_{23} = \sin\beta\sin(\gamma+\psi_{12}); \quad \cos\theta_{12} = \sin\beta\sin(\gamma-\psi_{23}); \tag{A.5}$$

Recalling $Z = z_3$, the mean vertical heights of the triangle legs are then given by

$$Z_{13} = Z + \frac{1}{2}z_{13}; \quad Z_{23} = Z + \frac{1}{2}z_{23}; \quad Z_{12} = Z + \frac{1}{2}(z_{13}+z_{23}) \tag{A.6}$$

The six forcing or source terms that appear on the right side of equation (4) of the Main Text for $\mathcal{C}_{123}$ can be obtained by straightforward but lengthy computations. We give only the final results:

$$F_{123} = \frac{k_BT}{4\pi\eta r_{12}}\bar{c}'(z_1)\Bigg\{(\cos\theta_{23} - \cos\psi_{13}\cos\theta_{12})\frac{\partial\mathcal{C}_{23}}{\partial r_{23}} + \frac{\sin^2\theta_{23} + \cos\theta_{12}(\cos\theta_{12} + \cos\psi_{13}\cos\theta_{23})}{r_{23}}\frac{\partial\mathcal{C}_{23}}{\partial(\cos\theta_{23})} + \frac{1}{2}(1+\cos^2\theta_{12})\frac{\partial\mathcal{C}_{23}}{\partial Z_{23}}\Bigg\} \tag{A.7}$$

$$F_{213} = \frac{k_BT}{4\pi\eta r_{12}}\bar{c}'(z_2)\Bigg\{(\cos\theta_{13} + \cos\psi_{23}\cos\theta_{12})\frac{\partial\mathcal{C}_{13}}{\partial r_{13}} + \frac{\sin^2\theta_{13} + \cos\theta_{12}(\cos\theta_{12} - \cos\psi_{23}\cos\theta_{13})}{r_{13}}\frac{\partial\mathcal{C}_{13}}{\partial(\cos\theta_{13})} + \frac{1}{2}(1+\cos^2\theta_{12})\frac{\partial\mathcal{C}_{13}}{\partial Z_{13}}\Bigg\} \tag{A.8}$$

$$F_{132} = \frac{k_BT}{4\pi\eta r_{13}}\bar{c}'(z_1)\Bigg\{-(\cos\theta_{23} + \cos\psi_{12}\cos\theta_{13})\frac{\partial\mathcal{C}_{23}}{\partial r_{23}} - \frac{\sin^2\theta_{23} + \cos\theta_{13}(\cos\theta_{13} - \cos\psi_{12}\cos\theta_{23})}{r_{23}}\frac{\partial\mathcal{C}_{23}}{\partial(\cos\theta_{23})} + \frac{1}{2}(1+\cos^2\theta_{13})\frac{\partial\mathcal{C}_{23}}{\partial Z_{23}}\Bigg\} \tag{A.9}$$

$$F_{312} = \frac{k_BT}{4\pi\eta r_{12}}\bar{c}'(z_3)\Bigg\{(\cos\theta_{12} + \cos\psi_{23}\cos\theta_{13})\frac{\partial\mathcal{C}_{12}}{\partial r_{12}} + \frac{\sin^2\theta_{12} + \cos\theta_{13}(\cos\theta_{13} - \cos\psi_{23}\cos\theta_{23})}{r_{12}}\frac{\partial\mathcal{C}_{12}}{\partial(\cos\theta_{12})} + \frac{1}{2}(1+\cos^2\theta_{13})\frac{\partial\mathcal{C}_{12}}{\partial Z_{12}}\Bigg\} \tag{A.10}$$

$$F_{231} = \frac{k_BT}{4\pi\eta r_{23}}\bar{c}'(z_2)\Bigg\{-(\cos\theta_{13} + \cos\psi_{12}\cos\theta_{23})\frac{\partial\mathcal{C}_{13}}{\partial r_{13}} - \frac{\sin^2\theta_{13} + \cos\theta_{23}(\cos\theta_{23} - \cos\psi_{12}\cos\theta_{13})}{r_{13}}\frac{\partial\mathcal{C}_{13}}{\partial(\cos\theta_{13})} + \frac{1}{2}(1+\cos^2\theta_{23})\frac{\partial\mathcal{C}_{13}}{\partial Z_{13}}\Bigg\} \tag{A.11}$$

$$F_{321} = \frac{k_BT}{4\pi\eta r_{23}}\bar{c}'(z_3)\Bigg\{(-\cos\theta_{12} + \cos\psi_{13}\cos\theta_{23})\frac{\partial\mathcal{C}_{12}}{\partial r_{12}} - \frac{\sin^2\theta_{12} + \cos\theta_{23}(\cos\theta_{23} + \cos\psi_{13}\cos\theta_{12})}{r_{12}}\frac{\partial\mathcal{C}_{12}}{\partial(\cos\theta_{12})} + \frac{1}{2}(1+\cos^2\theta_{23})\frac{\partial\mathcal{C}_{12}}{\partial Z_{12}}\Bigg\} \tag{A.12}$$

In order to calculate the leading-order asymptotic approximation to $\mathcal{C}_{123}$ for short times and large-r, we must substitute into the forcing terms the corresponding derivatives of the mean, $\nabla\bar{c}(z,t) = c_0\exp(-z^2/4D(t+\tau))/\sqrt{4\pi D(t+\tau)}$, and of the second cumulants. The latter are obtained from equation (6) in the Main Text to be

$$\frac{\partial \mathcal{C}_{pq}}{\partial r_{pq}} = -\frac{\mathcal{C}_{pq}}{r_{pq}} - \frac{3c_0^2}{16\pi}\frac{\sigma}{Z_{pq}^2+\frac{1}{4}z_{pq}^2}(1+\cos^2\theta_{pq})\cos^2\theta_{pq}\left\{\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D(t+\tau)}\right)-\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D\tau}\right)\right\} \quad \text{(A.13)}$$

$$\frac{\partial \mathcal{C}_{pq}}{\partial(\cos\theta_{pq})} = -\frac{2\cos\theta_{pq}}{1+\cos^2\theta_{pq}}\mathcal{C}_{pq} - \frac{3c_0^2}{16\pi}\frac{\sigma r_{pq}}{Z_{pq}^2+\frac{1}{4}z_{pq}^2}(1+\cos^2\theta_{pq})\cos\theta_{pq}\left\{\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D(t+\tau)}\right)-\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D\tau}\right)\right\} \quad \text{(A.14)}$$

$$\frac{\partial \mathcal{C}_{pq}}{\partial Z_{pq}} = -\frac{3c_0^2}{4\pi}\frac{\sigma}{r_{pq}}(1+\cos^2\theta_{pq})\frac{Z_{pq}}{Z_{pq}^2+\frac{1}{4}z_{pq}^2}\left\{\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D(t+\tau)}\right)-\exp\left(-\frac{Z_{pq}^2+\frac{1}{4}z_{pq}^2}{2D\tau}\right)\right\} \quad \text{(A.15)}$$

Finally, we must integrate in time the six forcing terms and sum them to obtain the asymptotic result for $\mathcal{C}_{123}$. The time-integration of the products of the mean gradient and of the derivatives of the second cumulants in (A.13)-(A.15) leads, in addition to elementary integrals, also to two less trivial integrals. The first can be evaluated by the change of variables $u = 1/(s+\tau)$ as follows

$$\int_0^t \frac{e^{-c/(s+\tau)}}{\sqrt{s+\tau}}\,ds = c^{1/2}\left[\Gamma\left(-\frac{1}{2},\frac{c}{t+\tau}\right)-\Gamma\left(-\frac{1}{2},\frac{c}{\tau}\right)\right] \quad \text{(A.16)}$$

where $\Gamma(\alpha,z)$ is the incomplete Gamma function [1, (8.2.2)]. Thus, one gets the first integral

$$\begin{aligned} I_1 &= \int_0^t \frac{e^{-a/(s+\tau)}}{\sqrt{s+\tau}}\left[e^{-b/(s+\tau)}-e^{-b/\tau}\right]ds \\ &= (a+b)^{1/2}\left[\Gamma\left(-\frac{1}{2},\frac{a+b}{t+\tau}\right)-\Gamma\left(-\frac{1}{2},\frac{a+b}{\tau}\right)\right] - a^{1/2}e^{-\frac{b}{\tau}}\left[\Gamma\left(-\frac{1}{2},\frac{a}{t+\tau}\right)-\Gamma\left(-\frac{1}{2},\frac{a}{\tau}\right)\right]. \end{aligned} \quad \text{(A.17)}$$

By using the expression for the incomplete Gamma function in terms of the Kummer confluent hypergeometric function U [1, (8.5.3)]:

$$\Gamma\left(-\frac{1}{2},z\right) = e^{-z}U\left(\frac{3}{2},\frac{3}{2};z\right), \quad \text{(A.18)}$$

one thus gets also

$$\begin{aligned} I_1 &= \int_0^t \frac{e^{-a/(s+\tau)}}{\sqrt{s+\tau}}\left[e^{-b/(s+\tau)}-e^{-b/\tau}\right]ds \\ &= (a+b)^{1/2}\left[e^{-\frac{a+b}{t+\tau}}U\left(\frac{3}{2},\frac{3}{2};\frac{a+b}{t+\tau}\right)-e^{-\frac{a+b}{\tau}}U\left(\frac{3}{2},\frac{3}{2};\frac{a+b}{\tau}\right)\right] - a^{1/2}e^{-\frac{b}{\tau}}\left[e^{-\frac{a}{t+\tau}}U\left(\frac{3}{2},\frac{3}{2};\frac{a}{t+\tau}\right)-e^{-\frac{a}{\tau}}U\left(\frac{3}{2},\frac{3}{2};\frac{a}{\tau}\right)\right]. \end{aligned} \quad \text{(A.19)}$$

The second time integral that appears is

$$I_2 = \int_0^t \frac{e^{-a/(s+\tau)}}{\sqrt{s+\tau}}\left[E_1\left(\frac{b}{s+\tau}\right)-E_1\left(\frac{b}{\tau}\right)\right]ds \quad \text{(A.20)}$$

and we have found no exact analytical expression for it. Instead we shall evaluate I_2 by numerical quadrature.

For short times t, the analytical results (A.17),(A.19) for I_1 and the integrand in the definition (A.20) of I_2 are all subject to cancellations from subtracting nearly equal terms. In that case we can obtain analytical results from series expansion. For I_1 we can use the series representation of the incomplete Gamma function [1, (8.7.3)] and the binomial expansion to obtain

$$c^\alpha\left[\Gamma\left(-\alpha,\frac{c}{t+\tau}\right)-\Gamma\left(-\alpha,\frac{c}{\tau}\right)\right] = \tau^\alpha\sum_{n=0}^{\infty}\frac{(-1)^n}{n!}\left(\frac{c}{\tau}\right)^n\sum_{k=1}^{\infty}(-1)^{k-1}\frac{(n-\alpha+1)^{k+1}}{k!}\left(\frac{t}{\tau}\right)^k$$

(A.21)

We apply this for $\alpha = 1/2$ with $c = a + b$, a. For the integrand of I_2 we use [1, (6.6.2)] to obtain

$$E_1\left(\frac{b}{s+\tau}\right) - E_1\left(\frac{b}{\tau}\right) = \ln\left(1+\frac{s}{\tau}\right) - \sum_{k=1}^{\infty}\frac{1}{k}\left(\frac{-b}{s+\tau}\right)^k \sum_{j=1}^{k}\frac{1}{j!(k-j)!}\left(\frac{s}{\tau}\right)^j \tag{A.22}$$

When $\frac{a+b}{\tau}$ is sufficiently large, above some threshold Θ, then the expression (A.19) for I_1 is accurate. In our numerical work, we took $\Theta = 10$. However, for $\frac{a+b}{\tau} < \Theta$ we must use the expansion (A.21) for $\alpha = 1/2$ with $c = a + b$, a. There is still a difficulty, because the factor $(a+b)^n - a^n e^{-b/\tau}$ in the double summation is subject to loss of significance error when b is small. Thus, we rewrite this term as

$$(a+b)^n - a^n e^{-b/\tau} = 2a^n e^{-b/2\tau}\sinh\left(\frac{b}{2\tau}\right) + \sum_{m=1}^{n}\binom{n}{m}a^{n-m}b^m \tag{A.23}$$

to avoid the cancellation of significant digits. For $\frac{b}{\tau}$ sufficiently large, above some threshold Θ, then the expression for I_2 in (A.20) for the integrand is accurate. We took $\Theta = 1$. Otherwise, we define the integral by the expansion (A.22). In either case, the time-integral I_2 is evaluated using the 2nd-order composite trapezoid rule. A script in `Matlab` is appended below, which carries out all of the numerical evaluations discussed above.

Finally, we note that great simplifications occur in the large-τ limit. To leader-order accuracy, only terms $\propto t/\tau$ need be retained in the summations (A.21)-(A.22). Note, in particular, that the mean concentration gradient becomes t-independent to leading order: $\nabla\bar{c} \sim \nabla c_0 \sim c_0/\sqrt{4\pi D\tau}$. The time-integrals of the forcing terms thus simplify and reduce to time-integrals of only the derivatives of the second cumulants. These are easily evaluated for large-τ to be

$$\int_0^t \frac{\partial\mathcal{C}}{\partial r_{pq}}(r_{pq},\cos\theta_{pq},Z_{pq},s)\,ds \sim -\frac{3}{4}|\nabla c_0|^2\frac{\sigma}{r_{pq}^2}(1+\cos^2\theta_{pq})Dt^2 \tag{A.24}$$

$$\int_0^t \frac{\partial\mathcal{C}}{\partial(\cos\theta_{pq})}(r_{pq},\cos\theta_{pq},Z_{pq},s)\,ds \sim \frac{3}{2}|\nabla c_0|^2\frac{\sigma}{r_{pq}}\cos\theta_{pq}Dt^2 \tag{A.25}$$

$$\int_0^t \frac{\partial\mathcal{C}}{\partial Z_{pq}}(r_{pq},\cos\theta_{pq},Z_{pq},s)\,ds \sim -3\pi|\nabla c_0|^4\frac{\sigma}{r_{pq}}(1+\cos^2\theta_{pq})Z_{pq}\frac{Dt^2}{c_0^2} \tag{A.26}$$

The latter derivative is thus sub-leading compared with the other two and may be neglected.

To give some intuition about the complex formulas, we plot in Figure 2 the three-point skewness, $\mathcal{S}_{123} = \mathcal{C}_{123}/(\mathcal{C}_{12}\mathcal{C}_{13}\mathcal{C}_{23})^{1/2}$ in the limit of vanishing mean concentration gradient, for varying orientation of the triangle (left panel) and for varying triangle shape (right panel). It is notable that the configuration studied in the main text does not have maximum possible skewness.

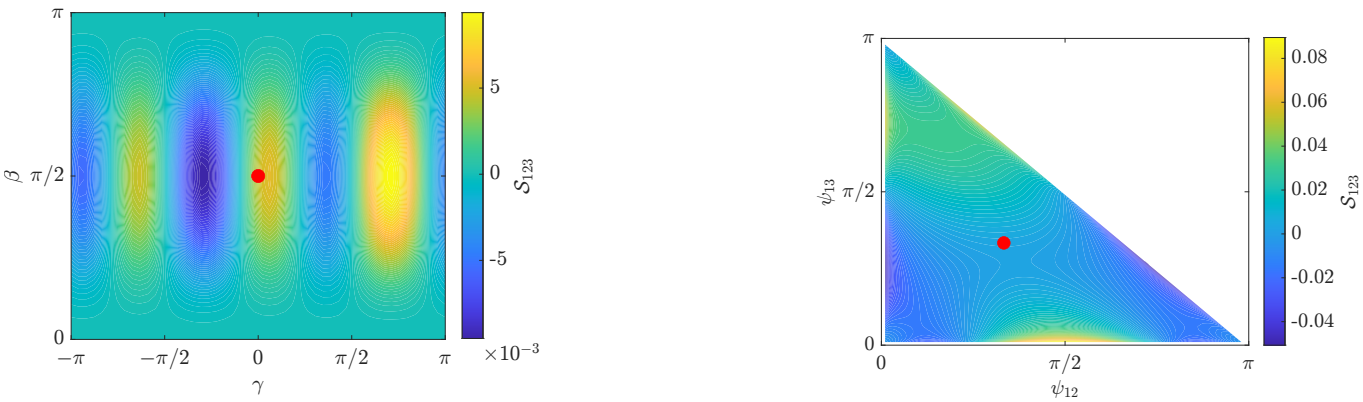

FIG. 2 Dependence of the asymptotic three-point skewness, $\mathcal{S}_{123} = \mathcal{C}_{123}/(\mathcal{C}_{12}\mathcal{C}_{13}\mathcal{C}_{23})^{1/2}$, on triangle orientation (specified by Euler angles, β and γ) and on triangle shape (specified by internal angles, ψ_{12} and ψ_{13}, subject to the restriction $0 \leq \psi_{13} + \psi_{12} \leq \pi$). In the *left* panel the triangle is equilateral with side lengths $r^* = 50$, while the *right* panel refers to a vertical triangle, $\beta = \pi/2$ and $\gamma = 0$, with arithmetic mean of side lengths = 50. All data refer to $t^* = 100$ and red dots represent the configuration studied in the Main Text.

```
clear all

% INPUTS

% geometry

r13=1e4;
r23=1e4;
ps12=3*pi/2;

% orientation

% alp=0, gam from 0 to 2*pi, bet from 0 to pi
alp=0;
bet=pi/2;
gam=pi/3;

% vertical location
z3=-5e3;

% time interval and time step

T=1;
dt=0.005;
t=0:dt:T;

% gradient parameter

tau=1e13;


% OUTPUTS

% derived quantities for triangle

r12=sqrt(r13^2+r23^2-2*r13*r23*cos(ps12));
ps23=asin(r23*sin(ps12)/r12);
ps13=asin(r13*sin(ps12)/r12);

costh13=sin(bet)*sin(gam);
costh23=sin(bet)*sin(gam+ps12);
costh12=sin(bet)*sin(gam-ps23);


lh=[cos(alp)*cos(gam)-sin(alp)*cos(bet)*sin(gam); ...
    sin(alp)*cos(gam)+cos(alp)*cos(bet)*sin(gam); ...
    sin(bet)*sin(gam)];

mh=[-cos(alp)*sin(gam)-sin(alp)*cos(bet)*cos(gam); ...
    -sin(alp)*sin(gam)+cos(alp)*cos(bet)*cos(gam); ...
    sin(bet)*cos(gam)];

nh=[sin(bet)*sin(alp); -sin(bet)*cos(alp); cos(bet)];

rv13=r13*lh;
rv23=r23*(cos(ps12)*lh+sin(ps12)*mh);
rv12=r12*(cos(ps23)*lh-sin(ps23)*mh);

z1=z3+rv13(3); z2=z3+rv23(3);
Z12=(z1+z2)/2; Z23=(z2+z3)/2; Z13=(z1+z3)/2;

% FIRST MOMENTS

C1=meanf(z1,tau,t);
C2=meanf(z2,tau,t);
C3=meanf(z3,tau,t);

% SECOND MOMENTS

```

```
C13=covf(r13,costh13,Z13,tau,t);
C23=covf(r23,costh23,Z23,tau,t);
C12=covf(r12,costh12,Z12,tau,t);

% THIRD MOMENT

btaumax=0;


% source term 123
A=costh23-cos(ps13)*costh12;
B=(1-costh23^2+costh12*(costh12+cos(ps13)*costh23))/r23;
C=(1+costh12^2)/2;

a=z1^2/2;
z23=r23*costh23;
b=Z23^2+z23^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh23^2)/r23^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh23^2)*costh23^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh23/r23/4/pi)*I2;
dC123= dC123 - B*(3*r23*(1+costh23^2)*costh23/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh23^2)*Z23/b/r23/4/pi)*I1;
dC123=dC123/r12;

C123=dC123;

% source term 213
A=costh13+cos(ps23)*costh12;
B=(1-costh13^2+costh12*(costh12-cos(ps23)*costh13))/r13;
C=(1+costh12^2)/2;

a=z2^2/2;
z13=r13*costh13;
b=Z13^2+z13^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh13^2)/r13^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh13^2)*costh13^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh13/r13/4/pi)*I2;
dC123= dC123 - B*(3*r13*(1+costh13^2)*costh13/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh13^2)*Z13/b/r13/4/pi)*I1;
dC123=dC123/r12;

C123=C123+dC123;

% source term 132
A=-(costh23+cos(ps12)*costh13);
B=-(1-costh23^2+costh13*(costh13-cos(ps12)*costh23))/r23;
C=(1+costh13^2)/2;

a=z1^2/2;
z23=r23*costh23;
b=Z23^2+z23^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh23^2)/r23^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh23^2)*costh23^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh23/r23/4/pi)*I2;
dC123= dC123 - B*(3*r23*(1+costh23^2)*costh23/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh23^2)*Z23/b/r23/4/pi)*I1;
dC123=dC123/r13;

C123=C123+dC123;

% source term 312
```

```
A=costh12+cos(ps23)*costh13;
B=(1-costh12^2+costh13*(costh13-cos(ps23)*costh23))/r12;
C=(1+costh13^2)/2;

a=z3^2/2;
z12=r12*costh12;
b=Z12^2+z12^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh12^2)/r12^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh12^2)*costh12^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh12/r12/4/pi)*I2;
dC123= dC123 - B*(3*r12*(1+costh12^2)*costh12/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh12^2)*Z12/b/r12/4/pi)*I1;
dC123=dC123/r13;

C123=C123+dC123;

% source term 231
A=-(costh13+cos(ps12)*costh23);
B=-(1-costh13^2+costh23*(costh23-cos(ps12)*costh13))/r13;
C=(1+costh23^2)/2;

a=z2^2/2;
z13=r13*costh13;
b=Z13^2+z13^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh13^2)/r13^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh13^2)*costh13^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh13/r13/4/pi)*I2;
dC123= dC123 - B*(3*r13*(1+costh13^2)*costh13/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh13^2)*Z13/b/r13/4/pi)*I1;
dC123=dC123/r23;

C123=C123+dC123;

% source term 321
A=-costh12+cos(ps13)*costh23;
B=-(1-costh12^2+costh23*(costh23+cos(ps13)*costh12))/r12;
C=(1+costh23^2)/2;

a=z3^2/2;
z12=r12*costh12;
b=Z12^2+z12^2/4; btaumax=max(b/tau,btaumax);
I1=int1(a,b,tau,t);
I2=int2(a,b,tau,dt,T);

dC123=-A*(3*(1+costh12^2)/r12^2/8/pi)*I2;
dC123= dC123 -A*(3*(1+costh12^2)*costh12^2/b/16/pi)*I1;
dC123= dC123 + B*(3*costh12/r12/4/pi)*I2;
dC123= dC123 - B*(3*r12*(1+costh12^2)*costh12/b/16/pi)*I1;
dC123= dC123 -C*(3*(1+costh12^2)*Z12/b/r12/4/pi)*I1;
dC123=dC123/r23;

C123=C123+dC123;


C123=C123*.75/sqrt(2*pi);

figure
plot(t,C123)
title('triple cumulant')

figure
N123=C123./sqrt(abs(C13.*C23.*C12));
plot(t,N123)
title('normalized triple cumulant')
```

```
210
211 btaumax=btaumax
212
213
214 function y=int1(a,b,tau,t)
215
216 if a+b≤10*tau
217
218 y=zeros(size(t));
219 nn=0;
220 next=ones(size(t));
221 fac=1;
222 while norm(next,'inf')>eps*norm(y,'inf')
223 next=t/tau;
224 dnext=next;
225 kk=2;
226 while norm(dnext,'inf')>eps*norm(next,'inf')
227 dnext=-(nn+kk-1.5)*dnext.*(t/tau)/kk;
228 next=next+dnext;
229 kk=kk+1;
230 end
231 mult=a^nn*2*exp(-b/tau/2)*sinh(b/tau/2);
232 if nn≥1
233 for mm=1:nn
234 mult=mult+nchoosek(nn,mm)*a^(nn-mm)*b^mm;
235 end
236 end
237 next=fac*next*mult;
238 y=y+next;
239 nn=nn+1;
240 fac=-fac/nn/tau;
241 end
242 y=sqrt(tau)*y;
243
244 else
245 y=sqrt(a+b)*(exp(-(a+b)./(t+tau)).*kummerU(1.5,1.5,(a+b)./(t+tau))...
246                -exp(-(a+b)/tau)*kummerU(1.5,1.5,(a+b)/tau));
247 y=y-sqrt(a)*exp(-b/tau)*(exp(-a./(t+tau)).*kummerU(1.5,1.5,a./(t+tau))...
248                -exp(-a/tau)*kummerU(1.5,1.5,a/tau));
249 end
250
251 end
252
253 function y=int2(a,b,tau,dt,tf)
254
255 s=0:dt:tf;
256
257 if b≤5*tau
258 diff=log(1+s/tau);
259 next=1;
260 kk=1;
261 while norm(next,'inf')>eps*norm(diff,'inf')
262 jj=1:kk;
263 S=ones(kk,1)*s;
264 JJ=jj.'*ones(size(s));
265 nsum=sum((S/tau).^JJ./factorial(JJ)./factorial(kk-JJ),1);
266 next=nsum.*(-b./(s+tau)).^kk/kk;
267 diff=diff-next;
268 kk=kk+1;
269 end
270 f=exp(-a./(s+tau)).*diff./sqrt(s+tau);
271
272 else
273 f=exp(-a./(s+tau)).*(expint(b./(s+tau))-expint(b/tau))./sqrt(s+tau);
274 end
275 y=cumtrapz(s,f);
276
277 end
278
279 function y=meanf(z,tau,t)
```

```
280
281  y=(1+erf(z./sqrt(2*(t+tau))))/2;
282
283  end
284
285  function y=covf(r,costh,Z,tau,t)
286
287  z=r*costh;
288  b=Z^2+z^2/4;
289
290  if b<= tau
291  diff=log(1+t/tau);
292  next=ones(size(t));
293  kk=1;
294  while norm(next,'inf')>eps*norm(diff,'inf')
295  jj=1:kk;
296  T=ones(kk,1)*t;
297  JJ=jj.'*ones(size(t));
298  nsum=sum((T/tau).^JJ./factorial(JJ)./factorial(kk-JJ),1);
299  next=nsum.*(-b./(t+tau)).^kk/kk;
300  diff=diff-next;
301  kk=kk+1;
302  end
303  y=3*(1+costh^2)*diff/r/(8*pi);
304
305  else
306  y=3*(1+costh^2)*(expint(b./(t+tau))-expint(b/tau))/r/(8*pi);
307  end
308
309  end
```

B. LAGRANGIAN MONTE CARLO NUMERICAL METHOD AND HPC IMPLEMENTATION

We exploit a Lagrangian numerical scheme that was developed to calculate statistical correlation functions of a passive scalar advected by the Kraichnan model of a turbulent velocity field [2–4]. To calculate the Pth moment function of the concentration field $\langle c(\mathbf{x}_1,t)\cdot\cdots\cdot c(\mathbf{x}_P,t)\rangle$, the DFV model is first solved for $c(\mathbf{x}_p,t)$ at the points $\mathbf{x}_p$, $p=1,...,P$ in one realization of the thermal velocity $\mathbf{w}$. This is accomplished by setting $c(\mathbf{x}_p,t)=c_0(\boldsymbol{\xi}_p(t))$ where $d\boldsymbol{\xi}_p/dt=\mathbf{w}_p$, $\boldsymbol{\xi}_p(0)=\mathbf{x}_p$, $p=1,...,P$ with $\boldsymbol{\mathcal{W}}=(\mathbf{w}_1,...,\mathbf{w}_P)$ a $3P$-dimensional Gaussian white-noise with covariance

$$\langle w_{pi}(t)w_{qj}(t')\rangle_{\boldsymbol{\Xi}}=R_{pq}(\boldsymbol{\xi}_p(t),\boldsymbol{\xi}_q(t))\delta(t-t')\,, \tag{B.1}$$

for fixed $\boldsymbol{\Xi}=(\boldsymbol{\xi}_1,...,\boldsymbol{\xi}_P)$ and for $p,q=1,...,P$ and $i,j=1,...,d$ in space dimension d. The evolution of the P stochastic Lagrangian tracers, $\boldsymbol{\xi}_p(t)$, $p=1,...,P$ is implemented with an Euler-Maruyama discretization

$$\boldsymbol{\xi}_p(t+\Delta t)=\boldsymbol{\xi}_p(t)+\tilde{\mathbf{w}}_p\sqrt{\Delta t},\quad p=1,...,P, \tag{B.2}$$

where $\tilde{\boldsymbol{\mathcal{W}}}_P=(\tilde{\mathbf{w}}_1,...,\tilde{\mathbf{w}}_P)$ is a normal random variable having mean zero and $Pd\times Pd$ covariance matrix

$$R_{pi,qj}=R_{pq}(\boldsymbol{\xi}_p(t),\boldsymbol{\xi}_q(t))\,, \tag{B.3}$$

for $p,q=1,...,P$ and $i,j=1,...,d$ in d space dimensions. We can generate an independent sample of the Pd-random vector by writing it as $\tilde{\boldsymbol{\mathcal{W}}}_P=\mathbf{L}_P\tilde{\mathbf{N}}_P$, where $\mathbf{L}_P$ is the lower triangular Cholesky factor of the positive-definite covariance matrix (B.3) and $\tilde{\mathbf{N}}_P$ is a Pd-dimensional standard normal vector. Since our focus is on the triple cumulant, we thus take $P=3$ and evolve a triplet $\boldsymbol{\Xi}=(\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\boldsymbol{\xi}_3)$ of stochastic Lagrangian tracers in each realization $\mathbf{w}$ of the thermal velocity field, obtaining thereby $c(\mathbf{x}_p,t)=c_0(\boldsymbol{\xi}_p(t))$ for $p=1,2,3$.

As in the earlier work on turbulent advection [2–4], the ensemble average that defines the triple cumulant is then approximated by an empirical average

$$C(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,t)\simeq\frac{1}{N}\sum_{n=1}^{N}c'^{(n)}(\mathbf{x}_1,t)c'^{(n)}(\mathbf{x}_2,t)c'^{(n)}(\mathbf{x}_3,t) \tag{B.4}$$

over a large number N of independent samples, where the concentration fluctuation field is defined by $c'^{(n)}(\mathbf{x}_p,t)=c_0(\boldsymbol{\xi}_p^{(n)}(t))-\bar{c}(\mathbf{x}_p,t)$ and $\boldsymbol{\xi}_p^{(n)}(t)$, $n=,1,...,N$ are independent solutions of (B.2). The convergence of the sample average (B.4) as $N\to\infty$ is slow, with typical Monte Carlo errors $\propto 1/\sqrt{N}$. Nevertheless, the scheme yielded accurate results for turbulent advection [2–4] with a number of samples at most $N\sim 10^7$. Unfortunately, our problem differs in two crucial respects. First, the covariance for the Kraichnan model of a turbulent velocity field had covariance increasing with r, whereas the velocity corresponding to thermal noise has covariance proportional to the Oseen tensor

$$R_{ij}(\mathbf{r})=\frac{k_BT}{4\pi\eta r}(\delta_{ij}+\hat{r}_i\hat{r}_j),\quad r\gtrsim\sigma \tag{B.5}$$

and thus decreasing with r. The consequence is that the elements in the $Pd\times Pd$ matrix (B.3) off-diagonal in p,q, which are entirely responsible for producing correlations between particles, grow smaller in time for our problem as particles separate. The second key difference is that the turbulence studies [2–4] considered a statistical steady-state for the passive scalar with large-scale injection, whereas we study the problem of free diffusion for which the concentration correlations decay in time. This decaying magnitude makes it more difficult to achieve acceptable relative error. Both of these differences require that N must be very much larger in our problem. Fortunately, the calculation of the sample average (B.4) is trivially parallelizable, with only the requirement to choose independent seeds for pseudorandom number generators on different processors.

We have developed and implemented a Monte Carlo algorithm based on a regularized version of (B.5) that accounts for the filtering of wavenumbers $\gtrsim 1/\sigma$. Specifically, we implement the exponentially decaying kernel proposed in [5], Appendix D, that also reproduces the expected Stokes-Einstein macroscopic diffusivity $D=k_BT/6\pi\eta\sigma$. For the sake of better performance, the numerical solver implements the regularization from [5] in (B.3) when $r_{pq}:=|\boldsymbol{\xi}_p-\boldsymbol{\xi}_q|\le 10\sigma$ and employs (B.5) otherwise. Moreover, we exploit an adaptive time-stepping method for the Euler-Maruyama time-discretization in (B.2) such that $\Delta t=f\rho^2/2D$, where $\rho=\min_{p,q}|\boldsymbol{\xi}_p-\boldsymbol{\xi}_q|$ and $0<f\le 1$ is a parameter controlling what fraction of the distance between the nearest two Lagrangian tracers can be covered on average with one time-step. We use $f=0.2$ in our results – tests with smaller and larger fractions confirm the time-scheme is converging. In addition, the results of the Main Text related to the long-time behavior are obtained imposing a minimum of 10 time-steps between consecutive data points to ensure smooth plots.

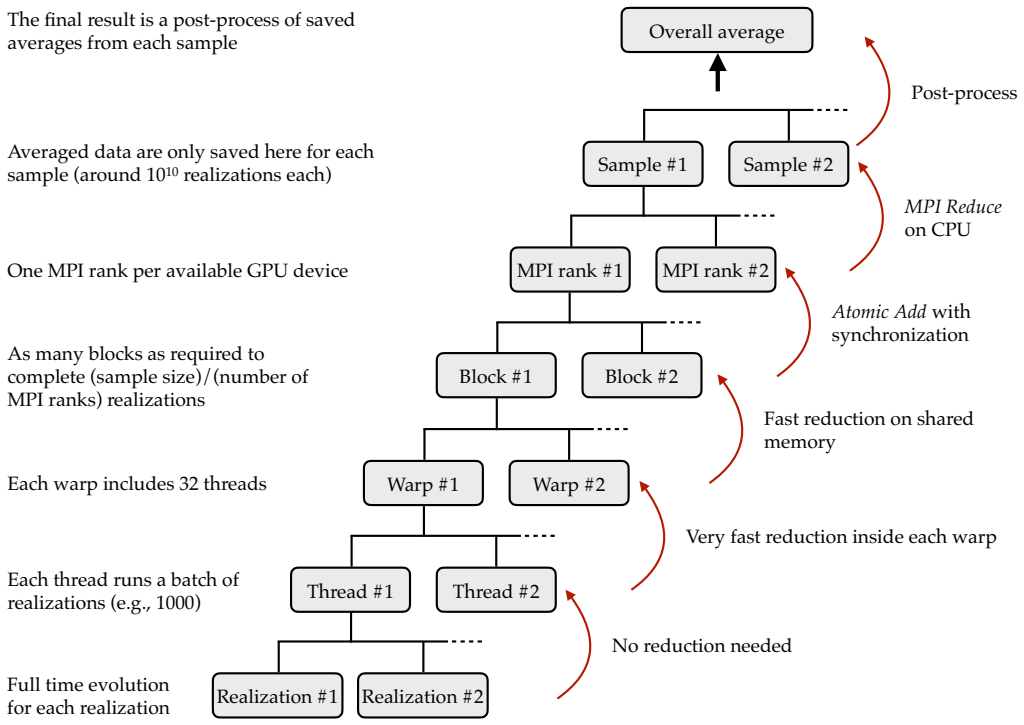

FIG. 3 Scheme illustrating the parallelization strategy and data management. Very large samples of realizations are independently run by several MPI processes and GPU devices each. The final result is obtained through a post-processing.

Although conceptually simple, the Lagrangian Monte Carlo algorithm requires careful implementation to efficiently support an extremely high number of realizations. Given the algorithm's inherently parallel structure, GPU architectures are the natural choice. To this end, we have developed a CUDA+MPI implementation that minimizes communication – the real bottleneck in reduction-heavy algorithms – by hierarchically distributing realizations and reductions across GPU devices, GPU blocks, warps, and, finally, batches in each thread. Figure 3 shows the parallelization scheme of our GPU code.

In order to manage the enormous number of realizations contributing to the average and to allow running independent Monte Carlo campaigns, we split the computation into smaller samples and compute the overall statistical observables through a post-process. This requires storing some data. Saving the trajectory of each realization for just one of the curves in the Main Text would require $\sim 300\,\mathrm{PB}$. We drastically reduce the amount of stored data to $\sim 50\,\mathrm{MB}$ by saving only the statistically meaningful observables (*i.e.* the cumulants up to third order) at the desired times for very large batches of realizations (called samples in Figure 3). Since each data file now contains the averages over a smaller sample, this data management also allows the central limit theorem to be exploited to compute the Monte Carlo error as the standard deviation of the sample means divided by the square root of the number of samples. This is sensibly faster than computing the standard deviation from each single realization.

Every time the code is launched, it iteratively computes new samples independently. Each sample is distributed over several nodes and GPU devices through standard MPI parallelization. The workload is further distributed over different GPU blocks, each containing several warps of 32 threads. Each thread then runs a batch of 1000 realizations, making the full time evolution of the particles trajectories the innermost loop. In order to compute the desired averages up to the level of samples, many reductions are needed. Proceeding upwards in the scheme of Figure 3, we reduce the partial sums computed by each thread to an intermediate sum over threads of the same warp (this is very efficient). Then, the first thread of each warp inside the same block reduces the sum again exploiting the GPU's shared memory, therefore no synchronization across blocks is needed up to here. The last reduction inside the GPU device is the `atomicAdd`, which requires blocks to synchronize in order to copy the partial sum to the CPU host. Finally, `MPI_Reduce` is used to collect the partial sums computed by each device and to store the desired averages for each sample at the penultimate level in Figure 3

The CUDA environment also allows using very efficient random number generators. We use `Philox_4x32_10` from the cuRAND library with a different seed for each MPI rank (or GPU) generated from the Linux file `/dev/urandom`

and a different state for each thread. Moreover, the `curand_normal2_double` function within the cuRAND library is exploited to halve the cost of producing two random numbers with double precision. Beside managing the large amount of realizations, the implementation of the Monte Carlo algorithm also requires to deal with the loss of significance, typical of averages over very large samples. We mitigate this issue by using double precision and implementing an extended version of the Kahan–Babuška compensated summation provided by Neumaier [6, 7] in all summations and reductions up to the block level. Distributing the summations over many levels of parallelization also helps.

In our computation, the loss of significance is even worsened by the asymptotically decaying concentration profile, $\bar{c}^*(z^*, t^*)$. Indeed, when the particles are outside the interface region, the concentration is almost homogeneous and the small deviations from the asymptotic value might be lost during the evaluation of the concentration fluctuation, $c_0^*(\boldsymbol{\xi}^*) - \bar{c}^*(\mathbf{x}^*, t^*)$. First of all, we shift the profile to be $\in [-0.5, 0.5]$ to balance the number of summations of positive and negative quantities. We use the Heaviside function for $t^* + \tau^* = 0$ and a combination of error function and complementary error function otherwise. In particular, if $z^* \gtrsim \sqrt{2(t^* + \tau^*)}$ or $z^* \lesssim -\sqrt{2(t^* + \tau^*)}$, then $\bar{c}^*(z^*, t^*) = 0.5 - \epsilon$ or $\bar{c}^*(z^*, t^*) = -0.5 + \epsilon$, respectively, with ϵ a small positive deviation. Therefore, we may have loss of significance due to the leading order terms being much larger than the small deviations. To avoid this, we exploit $\mathrm{erf}(z^*) = 1 - \mathrm{erfc}(z^*)$, $\mathrm{erf}(-z^*) = -\mathrm{erf}(z^*)$ in the formula for $\bar{c}^*(z^*, t^*)$, thus separating the leading order term from the remainder ($\epsilon = \frac{1}{2}\mathrm{erfc}|z^*|$). Therefore, the concentration fluctuation is computed by separately subtracting the leading order terms and the remainders, similar to the compensated summations algorithm mentioned above.

Overall, the Monte Carlo code has proven capable to yield up to $N \sim 10^{15}$ independent realizations. Each curve in the insets of Figure 1 and Figure 2 of the Main Text required $\sim 10^{14}$ realizations, whereas those in the main plots (long-time behavior) were obtained with $\sim 3 \times 10^{13}$ realizations. All these data were produced on the Leonardo Tier-0 cluster hosted by CINECA by performing a total of ~ 76 EFLOP in double precision, corresponding to ~ 7700 GPU-hours. Despite the very large computation, the computational cost was as low as ~ 1 GPU-hour per 10^{12} realizations. Indeed, this efficient implementation allowed to reach a performance of 5.8 TFLOP/s per GPU (which is close to the cluster's peak performance) and run simulations at a rate of ~ 130 TFLOP/s.

C. SUPPLEMENTAL NUMERICAL DATA

We provide here some further numerical data to supplement that presented in the Main Text and End Matter.

Skewness for $r^* = 50$, including $\tau^* = 10^4$

We elected in Fig. 2 of the Main Text to plot numerical results for the three-point skewness only at the three largest values $\tau^* = 10^6$, 10^8, 10^{10}, in order to show better their convergence as $\tau^* \to \infty$. For completeness, we plot here in Fig. 4 the entire set of results including also $\tau^* = 10^4$. As expected from the sign of the triple cumulant in Fig. 1 of the Main Text for the corresponding value of τ^*, the three-point skewness for $\tau^* = 10^4$ is negative over the entire time range. A simple physical explanation for this sign is lacking and also for its reversal with increasing τ^*. Nevertheless, just as for the other three choices of τ^*, the three-point skewness appears to level off and approach a "quasi-equilibrium" value for $t^* \to \infty$. An extremely slow decrease to zero might also be suggested by the plotted results and cannot be ruled out. Perhaps the most significant observation from Fig. 4 is that the maximum amplitude of the skewness is two times larger (about 0.02) for $\tau^* = 10^4$ than it is for the higher τ^* values. This has the interesting implication for experiment that sharper interfaces with larger initial concentration gradients may have greater non-Gaussian signatures of the concentration fluctuations. In physical units, taking water at ambient temperature as the reference fluid, the times we have considered in Fig. 4 are only up to 20 ms (as indicated by the labels on the upper abscissa). Nonetheless, taking the mean free path as 2.5 $\mathring{A}$, this value actually corresponds to $\sim 5 \times 10^{10}$ mean free times of the water molecules and is thus quite long on microscopic time scales. Moreover, the mean concentration gradient at the origin at this final time $t^* = 10^4$ when compared with the initial gradient has dropped by the factor 29.3%, 0.496%, 0.005%, 5×10^{-5}% for the four cases $\tau^* = 10^4, 10^6, 10^8, 10^{10}$, respectively. Thus, this time is sufficient to observe macroscopic diffusion of the mean concentration, especially for the sharper interfaces.

Cumulants for $r^* = 100$, short-time regime

In the Main Text and in the End Matter we presented results only for three points forming an equilateral triangle with side-length $r^* = 50$, but comparable results are obtained for other values of r^* and for other triangle shapes and orientations. As representative of this fact, we show here numerical results for an equilateral triangle of the same orientation as the earlier case ($\gamma = 0, \beta = \pi/2$) but instead with $r^* = 100$. These results were computed only for short times $t^* \leq 100$ where they can be directly compared with the asymptotic analytical results derived in SM,§A.

We present first in Figure 5 the results for the three-point skewness with $r^* = 100$, which can be observed to be qualitatively quite similar to those for $r^* = 50$ in Fig. 4. The agreement between numerical Monte Carlo and analytical results is even better than for the smaller r^*, as expected since the asymptotic calculation assumes large r^* and small t^*. The main result of increasing r^* is that the magnitudes of the three-point skewness have become smaller, in agreement with analytical predictions.

The input results for calculating the skewness at $r^* = 100$ are presented in the remaining plots, Figure 6 for the triple cumulant $\mathcal{C}_{123}$ and Figure 7 for the combined second cumulants $(\mathcal{C}_{13}\mathcal{C}_{23}\mathcal{C}_{12})^{1/2}$, both plotted versus t^*. An interesting feature observed in Fig. 6 is that the Monte Carlo errors are greatest for the case $\tau^* = 10^5$ at which the triple cumulant has switched from negative to positive. The same feature is also observed for the results at $r^* = 50$ in Fig. 1 of the Main Text. The increase in Monte Carlo error for the same sample size implies an increase in the variance, which is tempting to attribute to the variation as realizations switch between triple products with positive and negative values. Note in Fig. 7 that the Monte Carlo error are consistently much smaller for the second cumulants than they are for the triple cumulants, just as was found for the results at $r^* = 50$ in the End Matter, Fig.3.

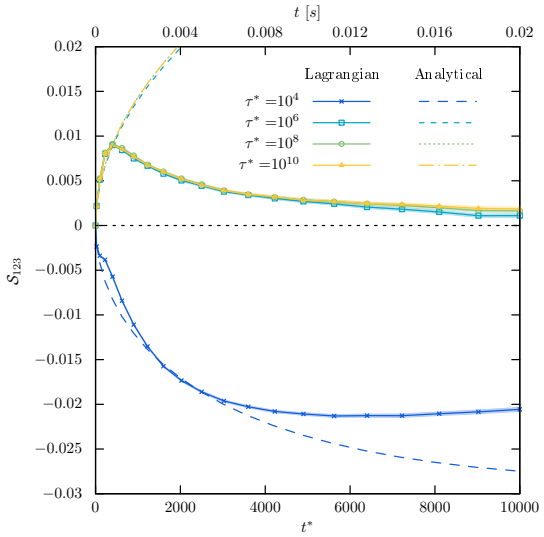

FIG. 4 Three-point skewness for the vertical equilateral triangle with $r^* = 50$ and vanishing concentration gradient, $\tau^* = 10^4$, 10^6, 10^8, and 10^{10}. This is an extension of Figure 2 of the Main Text. Solid lines with symbols and shades represent numerical results while dashed lines the analytical prediction.

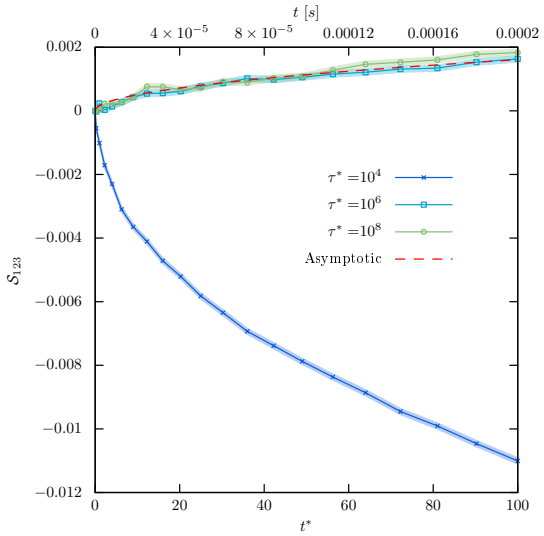

FIG. 5 Three-point skewness for the vertical equilateral triangle with $r^* = 100$ and vanishing concentration gradient, $\tau^* = 10^4$, 10^6, and 10^8. Solid lines with symbols and shades represent numerical results while the red dashed line is the asymptotic analytical prediction in the limit of vanishing gradient.

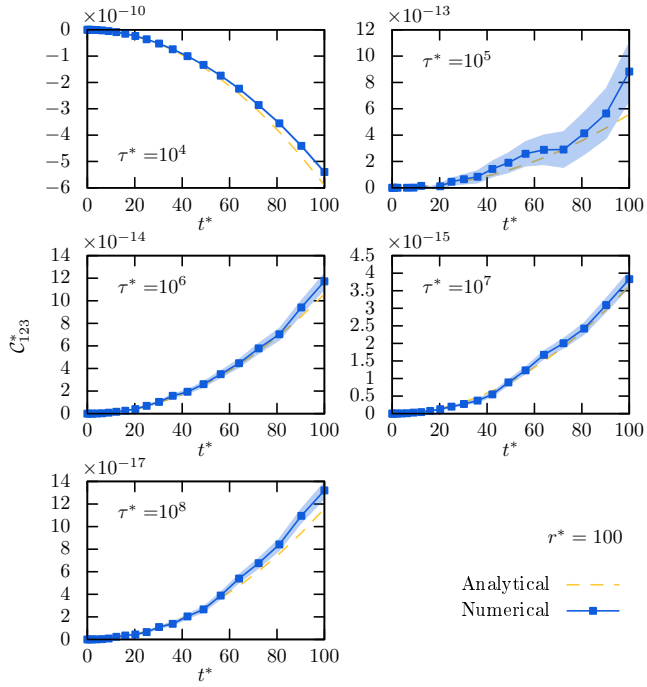

FIG. 6 Triple cumulant for the vertical equilateral triangle with $r^* = 100$ and vanishing concentration gradient, $\tau^* = 10^4$, 10^5, 10^6, 10^7, and 10^8. Solid lines with symbols and shades represent numerical results while dashed lines the analytical prediction.

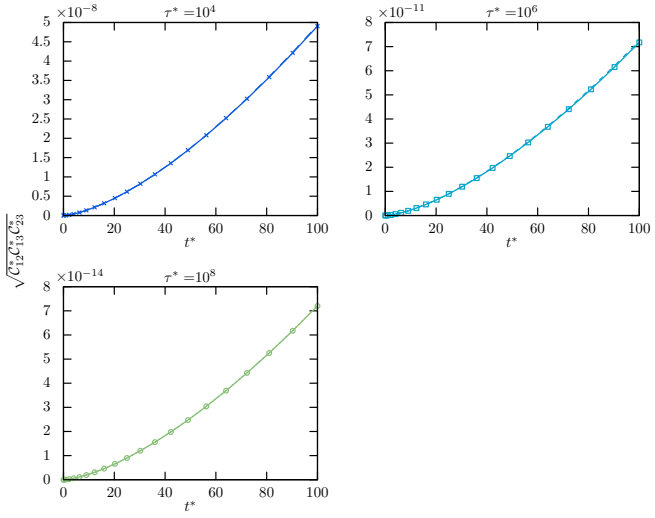

FIG. 7 Average second cumulant for the vertical equilateral triangle with $r^* = 100$ and vanishing concentration gradient, $\tau^* = 10^4$, 10^6, and 10^8. Solid lines with symbols and shades represent numerical results while dashed lines the analytical prediction. The Monte Carlo error is too small to be visible at this scale.

D. CONCISE COMPARISON OF LIQUID DIFFUSION AND TURBULENT ADVECTION

Our results deepen the analogy between high-Schmidt liquid diffusion and turbulent advection of a passive scalar, which was first proposed by Donev, Fai and Vanden-Eijnden [8]. The latter paper first pointed out that a macroscopic diffusivity renormalized by thermal velocity fluctuations closely resembles an "eddy diffusivity" in a turbulent flow and can be explained by a similar cascade mechanism. This correspondence is not perfect. For example, in a turbulent scalar cascade there is a "dissipative anomaly" so that the effective dissipation rate at high Reynolds numbers becomes completely independent of the microscale parameters such as the bare diffusivity D_0. By contrast, the Stokes-Einstein relation $D = k_B T/6\pi\eta\sigma$ for macroscopic diffusivity involves the radius σ of the solute particles and the limit $\sigma \to 0$ cannot be taken. Thus, as observed already by [8], some small "bare diffusivity" D_0 is required at the finest scale σ where a hydrodynamic approximation is admissible. These differences stem from the fact that the energy spectrum of turbulent velocity fluctuations is peaked at the lowest wavenumbers, so that the turbulent velocities are spatially Hölder continuous, whereas thermal velocity fluctuations have a spectrum peaked at the highest wavenumber and their Hölder exponent is formally negative. We find that advection by such a rough thermal velocity produces non-Gaussian scalar statistics just as does turbulent advection, but all of our derived scaling is consistent with naive dimensional analysis. Thus, there is no signature in our results of "zero modes" [9] and associated anomalous scaling of the higher-order scalar statistics, which is a principal feature of turbulent-advected scalars [2–4].

[1] DLMF, *NIST Digital Library of Mathematical Functions*, `https://dlmf.nist.gov/`, Release 1.2.4 of 2025-03-15, F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[2] U. Frisch, A. Mazzino, and M. Vergassola, Intermittency in passive scalar advection, Physical review letters **80**, 5532 (1998).

[3] U. Frisch, A. Mazzino, A. Noullez, and M. Vergassola, Lagrangian method for multiple correlations in passive scalar advection, Physics of Fluids **11**, 2178 (1999).

[4] O. Gat, I. Procaccia, and R. Zeitak, Anomalous scaling in passive scalar advection: Monte carlo lagrangian trajectories, Physical review letters **80**, 5536 (1998).

[5] G. Eyink and A. Jafari, High schmidt-number turbulent advection and giant concentration fluctuations, Physical Review Research **4**, 023246 (2022).

[6] W. Kahan, Pracniques: further remarks on reducing truncation errors, Communications of the ACM **8**, 40 (1965).

[7] A. Neumaier, Rundungsfehleranalyse einiger verfahren zur summation endlicher summen, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **54**, 39 (1974).

[8] A. Donev, T. G. Fai, and E. Vanden-Eijnden, A reversible mesoscopic model of diffusion in liquids: from giant fluctuations to fick's law, Journal of Statistical Mechanics: Theory and Experiment **2014**, P04004 (2014).

[9] D. Bernard, K. Gawędzki, and A. Kupiainen, Slow modes in passive advection, Journal of Statistical Physics **90**, 519 (1998).