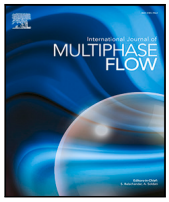
International Journal of Multiphase Flow 181 (2024) 104980

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow



# Generative diffusion models for synthetic trajectories of heavy and light particles in turbulence

Tianyi Li [a], Samuele Tommasi [a], Michele Buzzicotti [a,b,*], Fabio Bonaccorso [a], Luca Biferale [a]

[a] Department of Physics and INFN, University of Rome "Tor Vergata", Via della Ricerca Scientifica 1, Rome, 00133, Italy
[b] Department of Applied Physics and Science Education, Eindhoven University of Technology, MB Eindhoven, 5600, Netherlands

ARTICLE INFO

Dataset link: http://smart-turb.roma2.infn.it, https://github.com/SmartTURB/diffusion-lagr

ABSTRACT

Heavy and light particles are commonly found in many natural phenomena and industrial processes, such as suspensions of bubbles, dust, and droplets in incompressible turbulent flows. Based on a recent machine learning approach using a diffusion model that successfully generated single tracer trajectories in three-dimensional turbulence and passed most statistical benchmarks across time scales, we extend this model to include heavy and light particles. Given the particle type – tracer, light, or heavy – the model can generate synthetic, realistic trajectories with correct fat-tail distributions for acceleration, anomalous power laws, and scale dependent local slope properties. This work paves the way for future exploration of the use of diffusion models to produce high-quality synthetic datasets for different flow configurations, potentially allowing interpolation between different setups and adaptation to new conditions.

## 1. Introduction

The Lagrangian description of turbulence involves tracking the information acquired by individual particles carried by the flow, and provides crucial insights into the physics underlying many natural phenomena and applied processes, such as cloud formation, industrial mixing, pollutant dispersion and quantum fluids (La Porta et al., 2001; Falkovich et al., 2002; Yeung, 2002; Post and Abraham, 2002; Shaw, 2003; Toschi and Bodenschatz, 2009; Xia et al., 2013; Bentkamp et al., 2019; Laussy, 2023). Lagrangian particles can convolve spatial and temporal information over an extensive range of scales. The time scale separation in a turbulent flow is given by $\tau_L/\tau_\eta \propto R_\lambda$, where $\tau_L$ is the largest energy-injection time scale and $\tau_\eta$ is the smallest Kolmogorov time scale. The Taylor microscale Reynolds number, $R_\lambda$, ranges from a few thousand in the laboratory to tens of thousands in atmospheric flows and up to millions in the solar wind (Frisch, 1995; Dhruva et al., 1997; Wrench et al., 2024). Another intriguing feature of Lagrangian turbulence is the strong intermittency of intense fluctuations associated with small-scale vortical structures (Mordant and Lévêque, 2004b; Biferale et al., 2005), which can lead, albeit with low probability, to acceleration events in excess of 50–60 standard deviations in table-top laboratory flows (Voth et al., 2001; Mordant et al., 2004a). Compared to tracers, which exactly follow the local flow, the situation becomes more complicated when the inertial effects of particles are combined with intermittent turbulent properties, which are important in facilitating droplet collisions and the formation of large droplets in clouds (Falkovich et al., 2002; Kostinski and Shaw, 2005). Inertial particles depart from fluid streamlines, resulting in a non-uniform spatial distribution, a phenomenon known as preferential concentration (Toschi and Bodenschatz, 2009). Light particles tend to accumulate in vortical structures, while heavy particles are expelled from these regions (Maxey and Riley, 1983; Balkovsky et al., 2001; Bec, 2003; Chen et al., 2006). Additionally, the response of particles to turbulent events is influenced by inertial filtering effects: heavy particles, due to their larger inertia, tend to filter out smaller-scale vortices and primarily respond to larger-scale structures, while light particles can more readily follow the small-scale turbulent fluctuations. Stochastic modeling of Lagrangian tracer properties is exceptionally challenging due to multi-time dynamics, such as small-scale trapping within vortices for periods exceeding the local eddy turnover time (Wilson and Sawford, 1996; Lamorgese et al., 2007; Minier et al., 2014; Biferale et al., 2005; Toschi et al., 2005). Typical modeling approaches involve proposing a random process in time for the velocity to capture the dynamics at the two spectrum extremes, $\tau_L$ and $\tau_\eta$ (Sawford, 1991; Pope, 2011). Recently, these models have been generalized to be infinitely differentiable with intermittent scaling properties by Viggiano et al. (2020). Multifractal and/or multiplicative models have been used to provide a possible analytical framework, and they can reproduce some non-trivial features of turbulent statistics (Biferale et al., 1998; Arneodo et al., 1998; Chevillard et al., 2019; Sinhuber et al.,

* Corresponding author.
*E-mail address:* michele.buzzicotti@roma2.infn.it (M. Buzzicotti).

https://doi.org/10.1016/j.ijmultiphaseflow.2024.104980
Received 7 June 2024; Received in revised form 22 August 2024; Accepted 22 August 2024
Available online 28 August 2024

2021; Zamansky, 2022; Lübke et al., 2023). Furthermore, stochastic models for the generation of heavy and light particles are even more problematic, having to integrate multi-scale properties and preferential concentration (Friedrich et al., 2022).

To generate turbulent data with the correct multiscale statistics across the full range of dynamics encountered in real turbulent environments, data-driven machine learning methods have been employed due to their powerful expressive capabilities. Generative models, which learn from the underlying distribution of large amounts of training data, are particularly suitable for this task (Buzzicotti, 2023). A notable example is the Generative Adversarial Network (GAN), which has been shown to effectively capture multiscale turbulent properties in the Eulerian framework (Buzzicotti et al., 2021; Yu et al., 2022; Li et al., 2023b,a). Granero-Belinchon (2024) utilized a U-net optimized with carefully designed loss based on multiscale properties to generate one-dimensional stochastic fields. Specifically, in our previous work (Li et al., 2024), we employed a diffusion model (DM) to generate Lagrangian tracers with accurate properties, spanning from large forcing scales, through the intermittent inertial range, to the coupled regime between inertial and dissipative scales (Arnéodo et al., 2008).

Given the previous success of DM in generating tracers with correct statistical properties across time scales, and its surprising ability to generate high-intensity rare events with realistic statistics, we now question the generalizability of the model to different particles properties, i.e. in the case where inertial effects are not negligible. Due to centrifugal/centripetal effects, it is known that heavy particles tend to experience smoother viscous fluctuations, while light particles enhance them (Cencini et al., 2006; Bec et al., 2006; Benzi and Biferale, 2015), making the problem a very important quantitative benchmark for data-driven, equations-blind tools. Specifically, here we show that DMs are able to conditionally generate multiscale Lagrangian trajectories for inertial (heavy/light) particles and tracers at moderate/high Reynolds numbers with unprecedented quantitative agreement with the ground-truth numerical data used for training. This is a step forward towards building a stochastic multiscale model for inertial particles for different Stokes numbers $St$ and added mass coefficients $\beta$ (see next section).

## 2. Materials and methods

### 2.1. Simulations for Lagrangian particles

To generate a dataset of Lagrangian particles, we first performed direct numerical simulations (DNS) of the incompressible Navier–Stokes equations following the approach described in (Biferale et al., 2023):

$$\begin{cases} \partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\nabla p + \nu\Delta\boldsymbol{u} + \boldsymbol{F} \\ \nabla\cdot\boldsymbol{u} = 0. \end{cases} \tag{1}$$

Here $\boldsymbol{u}$ represents the Eulerian velocity field, $\nu$ the viscosity and $\boldsymbol{F}$ the large-scale isotropic forcing. We used a standard pseudo-spectral approach, fully dealiased with the two-thirds rule, within a cubic periodic domain with a resolution of $1024^3$. The resulting Taylor microscale Reynolds number was $R_\lambda \simeq 310$. Details of the simulation can be found in Li et al. (2024).

Once a statistically stationary state was reached for the underlying Eulerian flow, we seeded the flow with particles. The particles were passively advected, assumed to be sufficiently dilute to neglect collisions and not to react back on the flow. The motion of a small spherical particle with radius $a$ and density $\rho_p$ suspended in the fluid with density $\rho_f$ and velocity $\boldsymbol{u}$ can be approximated as (Maxey and Riley, 1983; Biferale et al., 2009):

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{V}(t), \tag{2}$$

$$\dot{\boldsymbol{V}}(t) = \beta D_t\boldsymbol{u}(\boldsymbol{X}(t),t) + \frac{1}{\tau_p}\left(\boldsymbol{u}(\boldsymbol{X}(t),t) - \boldsymbol{V}(t)\right), \tag{3}$$

where, $\boldsymbol{X}(t)$ and $\boldsymbol{V}(t)$ are respectively the particle position and velocity, $\beta = 3\rho_f/(\rho_f + 2\rho_p)$ is the density ratio between the fluid and the particle, $\tau_p = a^2/(3\beta\nu)$ is the particle response time, whose ratio with the Kolmogorov time scale $\tau_\eta$ defines the particle Stokes number, $St = \tau_p/\tau_\eta$.

For the numerical integration of Lagrangian particles, we used a sixth-order B-spline interpolation scheme to obtain the fluid velocity at the particle positions and a second-order Adams–Bashforth time marching scheme for time integration (Van Hinsberg et al., 2012). We tracked $N_p = 327680$ trajectories for each type of particle: heavy ($\beta = 0.01$), tracer ($\beta = 1$) and light ($\beta = 2.5$), over a total time of $T \simeq 1.3\tau_L \simeq 200\tau_\eta$. Both heavy and light particles are integrated with $\tau_p = 0.02$ resulting in a $St = 0.87$. Lagrangian information was recorded every $dt_s \simeq 0.1\tau_\eta$, resulting in each trajectory consisting of $K = 2000$ points (Calascibetta et al., 2023; Li et al., 2024).

### 2.2. Diffusion models for conditional generation

In this section we introduce the DMs used in this work to generate Lagrangian trajectories of different particles. The DM framework consists of two main processes: the forward and the backward process. The forward process operates as a Markov chain, incrementally adding Gaussian noise to the training data until the original signal is reduced to pure noise. In contrast, as shown in Fig. 1, the backward process starts with pure Gaussian noise and uses a learned neural network to gradually denoise and generate information, eventually producing realistic trajectory samples. In our notation we represent each trajectory as $\mathcal{V} = \{V_i(t_k) | t_k \in [0,T]; i = x,y,z\}$, where $k = 1,\dots,K$ are the discretized sampling times of each trajectory. The distribution of the ground-truth trajectories derived from the DNS is denoted by $q(\mathcal{V}|c)$, where $c$ indicates the type of particles: tracers, heavy particles or light particles. The *forward* diffusion process consists of $N$ Markovian noising steps, starting from any of the trajectories generated by the DNS, $\mathcal{V}_0 = \mathcal{V}$. Each step, $n = 1,\dots,N$, is defined as

$$q(\mathcal{V}_n|\mathcal{V}_{n-1}) \to \mathcal{V}_n \sim \mathcal{N}(\sqrt{1-\beta_n}\mathcal{V}_{n-1}, \beta_n\boldsymbol{I}), \tag{4}$$

which means that $\mathcal{V}_n$ samples from a Gaussian distribution with mean $\sqrt{1-\beta_n}\mathcal{V}_{n-1}$ and variance $\beta_n\boldsymbol{I}$. We can formally express the forward process as

$$q(\mathcal{V}_{1:N}|\mathcal{V}_0) := \prod_{n=1}^{N} q(\mathcal{V}_n|\mathcal{V}_{n-1}), \tag{5}$$

where the notation $\mathcal{V}_{1:N}$ denotes the entire sequence of noisy trajectories $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_N$ obtained from a specific $\mathcal{V}_0$ taken from the training set. The variance schedule $\beta_1,\dots,\beta_N$ is predefined, with a large $N$ to allow a continuous transition to the pure Gaussian state, $\mathcal{V}_N \sim \mathcal{N}(0,\boldsymbol{I})$. Further details of the variance schedule can be found in Appendix A.

The *backward* process reverses the above procedure using a neural network to provide $p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n, c)$ for each step. Here the network uses the particle type $c$ as an additional input to condition the generation on the specific inertial properties of the trajectories we want to generate. Details of the network architecture are given in Appendix A. Therefore, starting with Gaussian noise drawn from $p(\mathcal{V}_N) = \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$, it is possible to conditionally generate new trajectories based on the desired type of particles with

$$p_\theta(\mathcal{V}_{0:N}|c) = p(\mathcal{V}_N)\prod_{n=1}^{N} p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n, c). \tag{6}$$

In the continuous diffusion limit, achieved by our choice of variance schedule and number of diffusion steps, the backward step $p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n, c)$ retains the same Gaussian functional form as the forward step. Therefore, the neural network is designed to predict the mean $\mu_\theta(\mathcal{V}_n, n, c)$ and standard deviation $\Sigma_\theta(\mathcal{V}_n, n, c)$ of the transition probability (Feller, 2015; Sohl-Dickstein et al., 2015):

$$p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n, c) \to \mathcal{V}_{n-1} \sim \mathcal{N}(\mu_\theta(\mathcal{V}_n, n, c), \Sigma_\theta(\mathcal{V}_n, n, c)). \tag{7}$$

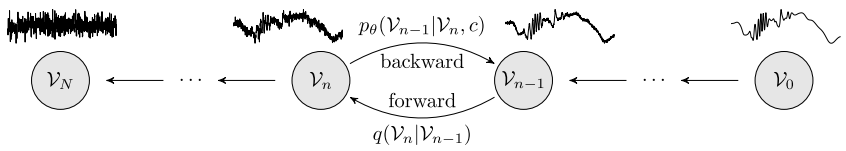



**Fig. 1.** Illustration of the forward and backward diffusion Markov processes. The forward process (right to left) introduces noise progressively over $N$ steps. In contrast, the backward process (left to right), implemented by a neural network, generates the trajectory step by step starting from pure Gaussian noise.

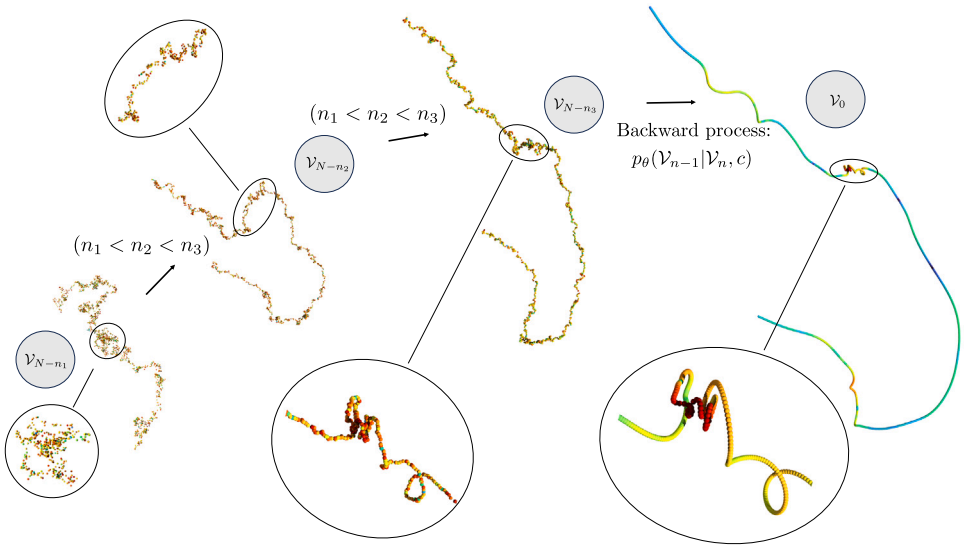



**Fig. 2.** Visualization of a typical tracer trajectory generation process at four different backward steps (from left to right). At each step of the generation process, the inset shows a zoomed view of the region in which a small-scale vortex structure is being generated.

The neural network is trained to minimize an upper bound of the negative log-likelihood,

$$\mathbb{E}_{q(\mathcal{V}_0|c)}[-\log(p_\theta(\mathcal{V}_0|c))]. \tag{8}$$

A detailed derivation of the loss function can be found in Appendix B. In Fig. 2 we illustrate an example of a tracer trajectory that is being gradually generated along the backward process. We can see that the backward diffusion starts to reconstruct the large-scale features of the particular trajectory in the early steps, up to the generation of the small-scale intense fluctuations and the smooth regions in the final steps. This sequence is primarily determined by the methodology. In the forward process, noise is progressively added, first disrupting small-scale correlations and then progressively affecting larger-scale correlations. In essence, information is removed scale by scale, from small to large scales. The backward process, trained to reverse the forward process, reconstructs the signal from large to small scales.

In this work we have considered two different types of DM, specifically we call DM-1c the diffusion model that is trained to generate a single velocity component along the particle trajectory, while we will call DM-3c the model that is trained to generate the three particle velocity components simultaneously. Both models use the same network architecture, except for the very first layer, which is adapted to the different number of channels. From the generation of the latter, it is possible to reconstruct the three-dimensional structure of the Lagrangian trajectory by time integration of the particle velocity.

## 3. Results

As a first result, in Fig. 3 we visually compare three-dimensional (3D) trajectories obtained from DNS with those generated by DM-3c for the three particle types considered in this work, heavy, tracer, and light. This first result is useful to show qualitatively that DM can reproduce the complex topological-vortical structures expected in the real trajectories with different inertia. From this figure, we can see that light particles, in both the DNS and DM examples, experience intense acceleration events (red-colored regions) much more frequently than the tracers, while heavy particles have a much smoother dynamic, reflecting DM's ability to correctly model the particle nature of being trapped in or escaping vortex filaments. Fig. 4 shows the three velocity components as a function of time for typical trajectories of different particles obtained from DNS and DM-3c. This comparison further demonstrates the consistency between DNS and DM for particles with different properties. The increasingly obvious and intense vortex-trapping events from heavy to tracer to light particles reflect their sampling of different regions of the turbulent field and the effects of inertial filtering, as particles with different inertia respond differently to turbulent structures.

In order to have a first comparison over the whole generated and the training dataset, in Fig. 5 we show the probability density function (PDF) of a generic component of the acceleration along the particle trajectories. The instantaneous particle acceleration is calculated as $a_i(t) = \lim_{\tau\to 0} \delta_\tau V_i/\tau$, approximated with a time resolution of $0.1\tau_\eta$. We can see that there is for all cases a very close agreement between the ground-truth DNS distributions and those from DMs over the whole range of fluctuations, and up to the extreme fluctuations, 60–70 times the standard deviation, observed for tracers and light particles. For a more quantitative comparison, we now study the statistical properties of high-order two-point time correlations by introducing the so-called Lagrangian structure functions, defined as

$$S_\tau^{(p)} = \langle [V_i(t+\tau) - V_i(t)]^p \rangle, \tag{9}$$

where on the l.h.s. we have removed the dependency on the component $i = x, y, z$ assuming isotropy. Fig. 6 (top row) shows the Lagrangian structure functions of order $p = 2, 4, 6$ for the DNS training data, the data set generated by DM-1c, which generates individual velocity components, and DM-3c, which generates all three velocity components

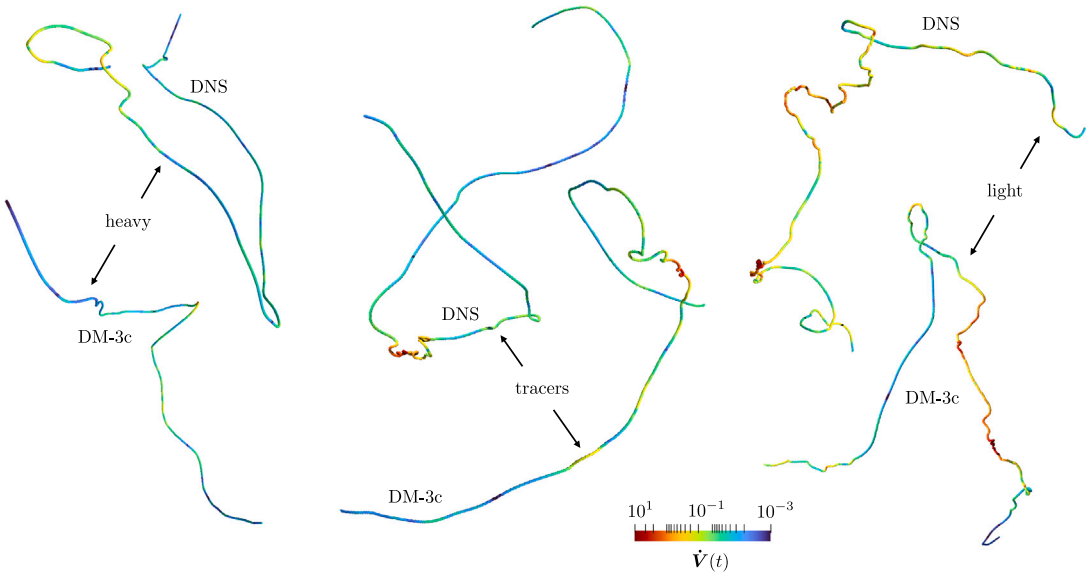



**Fig. 3.** Examples of 3D trajectories generated from DNS and DM-3c from left to right respectively for heavy, tracers, and light particles. The colors are proportional to the local acceleration experienced by the particles along the trajectories, in particular, red indicates intense acceleration and blue indicates low acceleration regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

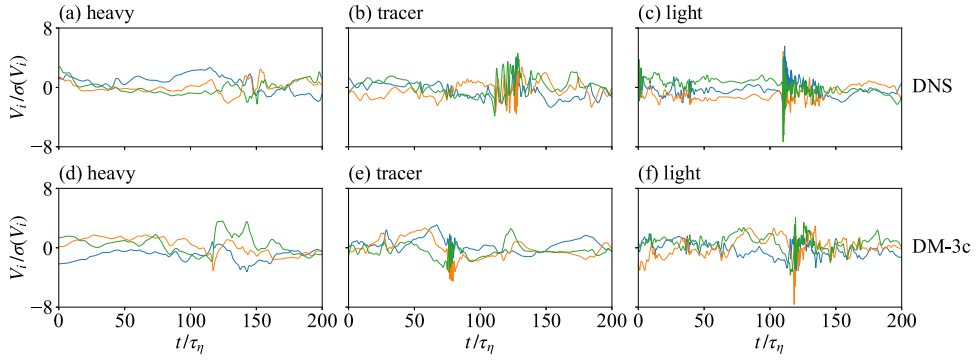



**Fig. 4.** Examples of different velocity components $i = x, y, z$ normalized by the standard deviation $\sigma$ as a function of time for trajectories from DNS (top) and DM-3c (bottom). (a)(d) heavy particles, (b)(e) tracers, (c)(f) light particles.

simultaneously. In the bottom row of the same figure, we show a comparison of the generalized flatness,

$$F_\tau^{(p)} = S_\tau^{(p)}/[S_\tau^{(2)}]^{p/2}, \tag{10}$$

up to $p = 8$ obtained for the same datasets discussed above. Given the sensitivity to the rare fluctuations in the data of such high-order observables and their extension over more than two decades of dynamical scales, it is remarkable how accurately the DM reproduces the correct ground truth statistics while distinguishing the different particle phenomena. We note that for $\tau < \tau_\eta$, DM-1c slightly outperforms DM-3c, likely due to the additional challenge of DM-3c in accounting for correlations between velocity components.

Finally, we discuss the most rigorous multiscale statistical test: comparing the scale-by-scale exponents obtained from the logarithmic derivatives of the structure functions in extended self-similarity (ESS) (Arnéodo et al., 2008), namely computed as

$$\zeta(p,\tau) = \frac{d\ \log S_\tau^{(p)}}{d\ \log S_\tau^{(2)}}. \tag{11}$$

To our knowledge, DM is the first method to successfully generate synthetic 3D Lagrangian tracer trajectories that reproduce this observable across all time scales (Li et al., 2024). In Fig. 7 we show the ESS local exponent for $p = 4$, again comparing DNS, DM-1c and DM-3c for the three particle types. These results allow us to conclude that DMs can correctly capture the multiscale properties of the structure-function scaling exponents even in the presence of different inertial properties. In particular, we can see how the model is able to correctly reproduce the different vortex trapping dynamics, which is strongly enhanced for light particles and depleted for heavy ones compared to the tracers, which is reflected in the intensification of the intermittency level and the depth of the viscous bottleneck around the range $\tau \sim \tau_\eta$ while decreasing the particle inertia, from heavy to tracer to light.

## 4. Conclusions

We have generalized a data-driven diffusion model, originally successful in generating single-particle tracers, to accommodate particle with different inertia: tracers, heavy and light particles. By incorporating data from different particle inertia, the model has adapted to new conditions while maintaining its effectiveness. It reproduces most statistical benchmarks across time scales, including the fat-tail distribution for acceleration, the anomalous power law, and the increased intermittency around the dissipative scale for tracers and light particles. Note that the original model showed a strong ability to generate unseen extreme events (Li et al., 2024); future work will involve collecting more statistics to check for similar capabilities in the current model.

In future research, the generalizability of the DM model can be further tested by including a more diverse set of data configurations

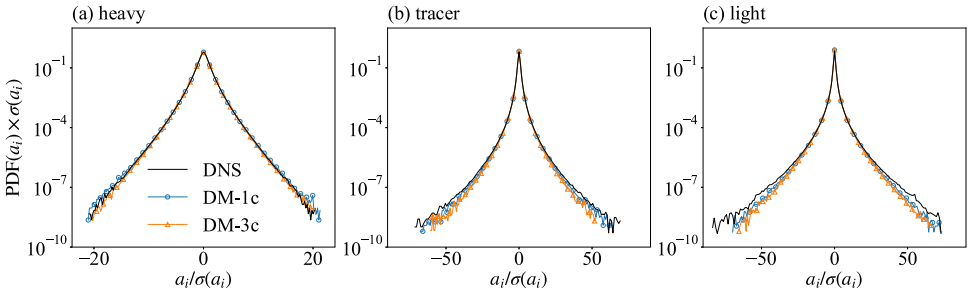



**Fig. 5.** Standardized PDFs of one generic component of the acceleration, $a_i$, for ground-truth DNS data (black line) and synthetically generated data from DM-1c (blue line with circles) and DM-3c (orange line with triangles) for (a) heavy particles, (b) tracers, and (c) light particles. The statistics for DMs are based on the same amount of data as those for DNS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

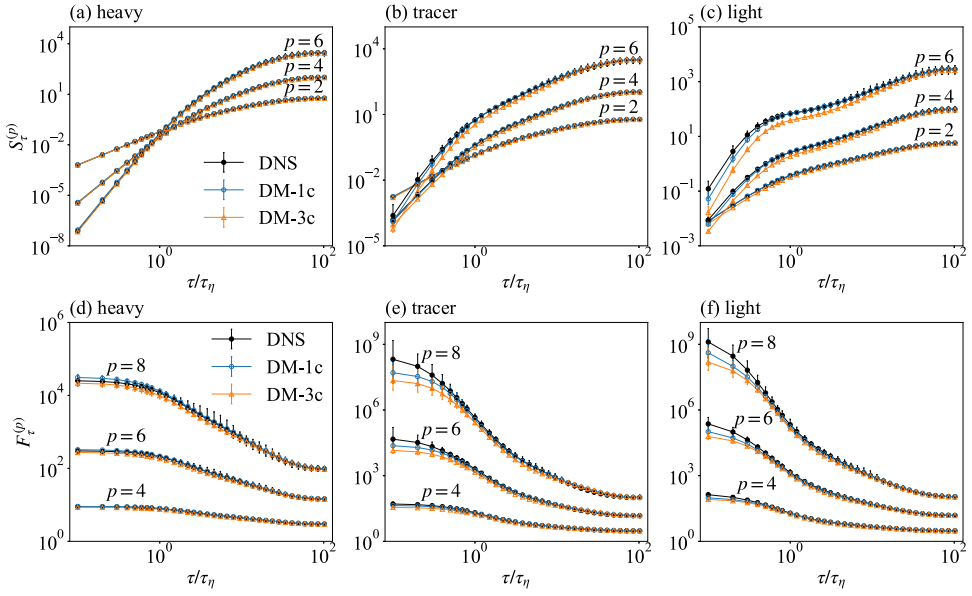



**Fig. 6.** Log–log plots comparing Lagrangian structure functions, $S_\tau^{(p)}$ for $p = 2, 4, 6$, and generalized flatness, $F_\tau^{(p)}$ for $p = 4, 6, 8$, between DNS and DMs for different particle types: (a)(d) heavy particles, (b)(e) tracers, and (c)(f) light particles. The color scheme and symbols are organized as in Fig. 5. The error bars indicate the range of values obtained for each measure by dividing the dataset used for the statistics into ten different independent batches per velocity component. This resulted in 10 batches for DM-1c and 30 batches for DNS and DM-3c.

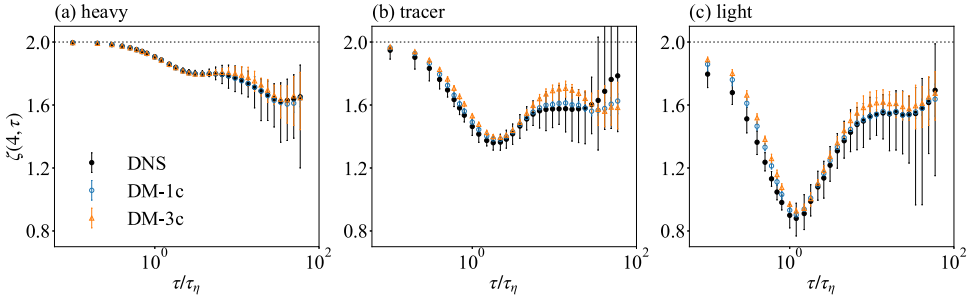



**Fig. 7.** Comparison of 4th-order logarithmic local slope $\zeta(4,\tau)$ between the ground-truth DNS and DMs on a lin–log scale for (a) heavy particles, (b) tracers, and (c) light particles. The dotted horizontal lines represent the non-intermittent dimensional scaling, $S_\tau^{(4)} \propto [S_\tau^{(2)}]^2$. The color scheme and symbols are organized as in Fig. 5, while the statistics and the error bars are derived in the same way as in Fig. 6.

in the training process. This will allow us to evaluate the interpolation and extrapolation capabilities of the model to unseen values of physical parameters during training, such as density ratios, Stokes numbers, and Reynolds numbers, to fully explore the potential of the model. Advanced network architectures, such as transformers (Vaswani et al., 2017), could be used to replace the current U-net to better handle the scaling capability required for larger and more diverse datasets. Our ultimate goal is to provide high-quality, high-volume synthetic datasets for downstream applications such as inertial particle classification and data inpainting (Friedrich et al., 2020; Li et al., 2023c; Zheng et al., 2024). Inertial particle classification involves determining the particle type based on observed trajectories, including properties such as density ratio and Stokes number. Data inpainting refers to the interpolation or reconstruction of complete data from partially observed data. By generating these synthetic datasets, we aim to avoid the impractical computational or experimental effort required to generate real Lagrangian trajectories.

## Data and code availability

The Lagrangian trajectories of different inertial particles used in this study, including positions and velocities, can be downloaded from the open access Smart-TURB portal at http://smart-turb.roma2.infn.it (Biferale et al., 2023). The code used to train the DM and generate new trajectories is available at https://github.com/SmartTURB/diffusion-lagr (Sma, 2024).

## CRediT authorship contribution statement

**Tianyi Li:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Samuele Tommasi:** Visualization, Formal analysis, Data curation. **Michele Buzzicotti:** Writing – review & editing, Visualization, Supervision, Methodology, Investigation, Conceptualization. **Fabio Bonaccorso:** Software, Data curation. **Luca Biferale:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The Lagrangian trajectories of different inertial particles used in this study, including positions and velocities, can be downloaded from the open access Smart-TURB portal at http://smart-turb.roma2.infn.it (Biferale et al., 2023). The code used to train the DM and generate new trajectories is available at https://github.com/SmartTURB/diffusion-lagr (Sma, 2024).

## Acknowledgments

We thank Mauro Sbragaglia and Roberto Benzi for useful discussions and collaborations in a early stage of this work. This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme Smart-TURB (Grant Agreement No. 882340), by the MUR-FARE project R2045J8XAW, and by Next Generation EU, Piano Nazionale di Ripresa e Resilienza (PNRR) Fondo per il programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN 2014, 202249Z89M).

## Appendix A. DM architecture and noise schedule

We use a U-net architecture (Ronneberger et al., 2015) as the backbone of the DM, consisting of two main parts: a downsampling stack and an upsampling stack, connected by skip connections as shown in Fig. A.8. The upsampling stack mirrors the downsampling stack, creating a symmetrical structure, with each stack performing four steps of downsampling or upsampling, respectively. This results in five stages from highest to lowest resolution (2000 to 125, each with a downsampling/upsampling rate of 2). The three residual blocks in these stages are configured with channels $[C, C, 2C, 3C, 4C]$, with $C$ set to 128. Multi-head attention (Vaswani et al., 2017) with four heads is implemented after each residual block in the 250 and 125 resolution stages. The intermediate module connecting the encoder and decoder stacks consists of two residual blocks of $4C$ channels, sandwiching a four-head attention. The diffusion step $n$ is specified to the network using transformer sinusoidal position embedding and the particle type is specified using class embedding.

We adopted the optimal noise schedule from previous research to generate Lagrangian tracers with a total of $N = 800$ diffusion steps (Li et al., 2024):

$$\bar{\alpha}_n = \frac{-\tanh(7n/N - 6) + \tanh 1}{-\tanh(-6) + \tanh 1}. \tag{A.1}$$

The variance can be obtained as $\beta_n = 1 - \bar{\alpha}_n/\bar{\alpha}_{n-1}$, which is clipped to be no greater than 0.999 to avoid singularities at the end of the forward diffusion.

The AdamW optimizer (Loshchilov and Hutter, 2017) was used to train the model with a learning rate of $10^{-4}$ over $2.5 \times 10^5$ iterations for DM-1c, and $4.0 \times 10^5$ iterations for DM-3c. The DMs were trained with a batch size of 256 on four NVIDIA A100 GPUs for approximately 25 h (DM-1c) and 40 h (DM-3c). An exponential moving average (EMA) strategy with a decay rate of 0.999 was applied to the model parameters to sample new trajectories.

## Appendix B. Derivation of the training loss function

We introduce an important property of the forward process: it allows closed-form sampling of $\mathcal{V}_n$ at each diffusion step $n$ (Weng, 2021):

$$q(\mathcal{V}_n|\mathcal{V}_0) \to \mathcal{V}_n \sim \mathcal{N}(\sqrt{\bar{\alpha}_n}\mathcal{V}_0, (1-\bar{\alpha}_n)\boldsymbol{I}), \tag{B.1}$$

where we define $\alpha_n := 1 - \beta_n$ and $\bar{\alpha}_n := \Pi_{i=1}^{n}\alpha_i$. In particular, given any initial trajectory $\mathcal{V}_0$, its state after $n$ diffusion steps can be sampled directly as

$$\mathcal{V}_n = \sqrt{\bar{\alpha}_n}\mathcal{V}_0 + \sqrt{1-\bar{\alpha}_n}\epsilon, \tag{B.2}$$

where $\epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

We use the variational bound to optimize the negative log-likelihood in Eq. (8):

$$L := \mathbb{E}_{q(\mathcal{V}_0)}\mathbb{E}_{q(\mathcal{V}_{1:N}|\mathcal{V}_0)}\left[-\log\frac{p_\theta(\mathcal{V}_{(0:N)})}{q(\mathcal{V}_{1:N}|\mathcal{V}_0)}\right] \geq \mathbb{E}_{q(\mathcal{V}_0)}[-\log(p_\theta(\mathcal{V}_0))]. \tag{B.3}$$

From this point on, we omit the condition on particle type $c$ for the sake of simplicity. This objective can be expressed as the sum of the Kullback–Leibler (KL) divergences, denoted as $D_{\mathrm{KL}}(\cdot \parallel \cdot)$, together with an additional entropy term (Sohl-Dickstein et al., 2015; Ho et al., 2020):

$$L = \mathbb{E}_{q(\mathcal{V}_0)}\Bigg[\underbrace{D_{\mathrm{KL}}(p(\mathcal{V}_N|\mathcal{V}_0) \parallel p_\theta(\mathcal{V}_N))}_{L_N} + \sum_{n>1}^{N}\underbrace{D_{\mathrm{KL}}(p(\mathcal{V}_{n-1}|\mathcal{V}_n, \mathcal{V}_0) \parallel p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n))}_{L_{n-1}} \underbrace{-\log p_\theta(\mathcal{V}_0|\mathcal{V}_1)}_{L_0}\Bigg]. \tag{B.4}$$

The first term, $L_N$, is ignored during training because it contains no learnable parameters, since $p_\theta(\mathcal{V}_N)$ is a Gaussian distribution. The second part of the terms, $L_{n-1}$, represents the KL divergence between $p_\theta(\mathcal{V}_{n-1}|\mathcal{V}_n)$ and the posteriors of the forward process conditioned on $\mathcal{V}_0$, which are tractable using Bayes' theorem (Weng, 2021):

$$p(\mathcal{V}_{n-1}|\mathcal{V}_n, \mathcal{V}_0) \to \mathcal{V}_{n-1} \sim \mathcal{N}(\tilde{\mu}(\mathcal{V}_n, \mathcal{V}_0), \tilde{\beta}_n\boldsymbol{I}), \tag{B.5}$$

where

$$\tilde{\mu}_n(\mathcal{V}_n, \mathcal{V}_0) := \frac{\sqrt{\bar{\alpha}_{n-1}}\beta_n}{1-\bar{\alpha}_n}\mathcal{V}_0 + \frac{\sqrt{\alpha_n}(1-\bar{\alpha}_{n-1})}{1-\bar{\alpha}_n}\mathcal{V}_n \tag{B.6}$$

and

$$\tilde{\beta}_n := \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n. \tag{B.7}$$

The KL divergence between the two Gaussians in Eqs. (7) and (B.5) can be expressed as

$$L_{n-1} = \mathbb{E}_{q(\mathcal{V}_0)}\left[\frac{1}{2\sigma_n^2}\|\tilde{\mu}_n(\mathcal{V}_n, \mathcal{V}_0) - \mu_\theta(\mathcal{V}_n, n)\|^2\right], \tag{B.8}$$

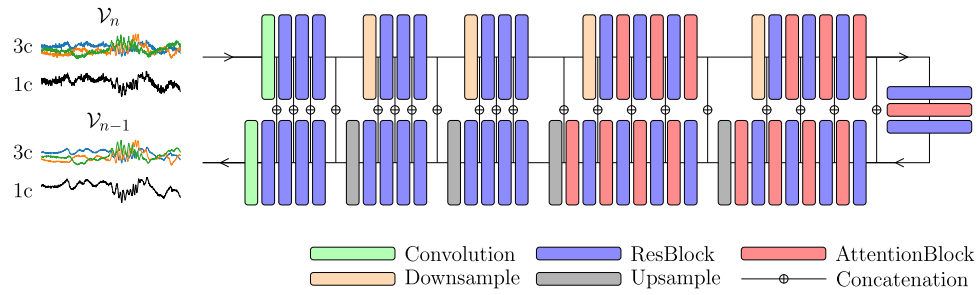



**Fig. A.8.** The U-net architecture that takes a noisy trajectory of a given particle inertia as input at step $n$ and predicts a denoised trajectory at step $n-1$.

given the constant variance $\Sigma_\theta = \sigma_n^2 \boldsymbol{I}$, where $\sigma_n^2$ can be either $\beta_n$ or $\tilde{\beta}_n$ as discussed in Ho et al. (2020) and we use the former in this work. It can be shown that the term $L_0$ takes the same form as in Eq. (B.8) due to the Gaussian form of $p_\theta(\mathcal{V}_0|\mathcal{V}_1)$ in Eq. (7).

We aim to train $\mu_\theta(\mathcal{V}_n, n)$ to predict $\tilde{\mu}(\mathcal{V}_n, \mathcal{V}_0)$, which is given by

$$\tilde{\mu}(\mathcal{V}_n, \mathcal{V}_0) = \frac{1}{\sqrt{\alpha_n}}\left(\mathcal{V}_n - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon\right), \tag{B.9}$$

by substituting Eq. (B.2) into Eq. (B.6). Therefore, given that $\mathcal{V}_n$ is available as input to the model, we can reparameterize to make the network predict the Gaussian noise term $\epsilon$, and the predicted mean is

$$\mu_\theta(\mathcal{V}_n, n) = \frac{1}{\sqrt{\alpha_n}}\left(\mathcal{V}_n - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon_\theta(\mathcal{V}_n, n)\right), \tag{B.10}$$

where $\epsilon_\theta$ is the predicted cumulative noise at step $n$. This reparameterization transforms Eq. (B.8) into

$$L_{n-1} = \mathbb{E}_{q(\mathcal{V}_0),\epsilon}\left[\frac{\beta_n^2}{2\sigma_n^2\alpha_n(1-\bar{\alpha}_n)}\|\epsilon - \epsilon_\theta\left(\mathcal{V}_n(\mathcal{V}_0,\epsilon), n\right)\|^2\right]. \tag{B.11}$$

We further ignore the weighting term and optimize a simplified version of the variational bound:

$$L_{\text{simple}} = \mathbb{E}_{n,q(\mathcal{V}_0),\epsilon}\left[\|\epsilon - \epsilon_\theta\left(\mathcal{V}_n(\mathcal{V}_0,\epsilon), n\right)\|^2\right], \tag{B.12}$$

where $n$ is sampled uniformly from 1 to $N$. In practice, this method improves the sample quality and simplifies the implementation (Ho et al., 2020).

## References


Arneodo, A., Bacry, E., Muzy, J.F., 1998. Random cascades on wavelet dyadic trees. J. Math. Phys. 39, 4142–4164.

Arnéodo, A., Benzi, R., Berg, J., Biferale, L., Bodenschatz, E., Busse, A., Calzavarini, E., Castaing, B., Cencini, M., Chevillard, L., et al., 2008. Universal intermittent properties of particle trajectories in highly turbulent flows. Phys. Rev. Lett. 100, 254504.

Balkovsky, E., Falkovich, G., Fouxon, A., 2001. Intermittent distribution of inertial particles in turbulent flows. Phys. Rev. Lett. 86 (2790).

Bec, J., 2003. Fractal clustering of inertial particles in random flows. Phys. Fluids 15, L81–L84.

Bec, J., Biferale, L., Cencini, M., Lanotte, A.S., Toschi, F., 2006. Effects of vortex filaments on the velocity of tracers and heavy particles in turbulence. Phys. Fluids 18.

Bentkamp, L., Lalescu, C.C., Wilczek, M., 2019. Persistent accelerations disentangle lagrangian turbulence. Nature Commun. 10 (3550).

Benzi, R., Biferale, L., 2015. Homogeneous and isotropic turbulence: A short survey on recent developments. J. Stat. Phys. 161, 1351–1365.

Biferale, L., Boffetta, G., Celani, A., Crisanti, A., Vulpiani, A., 1998. Mimicking a turbulent signal: Sequential multiaffine processes. Phys. Rev. E 57 (R6261).

Biferale, L., Boffetta, G., Celani, A., Lanotte, A., Toschi, F., 2005. Particle trapping in three-dimensional fully developed turbulence. Phys. Fluids 17.

Biferale, L., Bonaccorso, F., Buzzicotti, M., Calascibetta, C., 2023. Turb-lagr. a database of 3d lagrangian trajectories in homogeneous and isotropic turbulence. arXiv preprint arXiv:2303.08662.

Biferale, L., Scagliarini, A., Toschi, F., 2009. Statistics of small scale vortex filaments in turbulence. arXiv preprint arXiv:0908.0205.

Buzzicotti, M., 2023. Data reconstruction for complex flows using ai: Recent progress, obstacles, and perspectives. Europhys. Lett..

Buzzicotti, M., Bonaccorso, F., Di Leoni, P.C., Biferale, L., 2021. Reconstruction of turbulent data with deep generative models for semantic inpainting from turb-rot database. Phys. Rev. Fluids 6, 050503.

Calascibetta, C., Biferale, L., Borra, F., Celani, A., Cencini, M., 2023. Optimal tracking strategies in a turbulent flow. Commun. Phys. 6 (256).

Cencini, M., Bec, J., Biferale, L., Boffetta, G., Celani, A., Lanotte, A., Musacchio, S., Toschi, F., 2006. Dynamics and statistics of heavy particles in turbulent flows. J. Turbul. N36.

Chen, L., Goto, S., Vassilicos, J., 2006. Turbulent clustering of stagnation points and inertial particles. J. Fluid Mech. 553, 143–154.

Chevillard, L., Garban, C., Rhodes, R., Vargas, V., 2019. On a skewed and multifractal unidimensional random field, as a probabilistic representation of kolmogorov's views on turbulence. In: Annales Henri PoincarÉ. Springer, pp. 3693–3741.

Dhruva, B., Tsuji, Y., Sreenivasan, K.R., 1997. Transverse structure functions in high-reynolds-number turbulence. Phys. Rev. E 56 (R4928).

Falkovich, G., Fouxon, A., Stepanov, M., 2002. Acceleration of rain initiation by cloud turbulence. Nature 419, 151–154.

Feller, W., 2015. Retracted chapter: On the theory of stochastic processes, with particular reference to applications. In: Selected Papers I. Springer, pp. 769–798.

Friedrich, J., Gallon, S., Pumir, A., Grauer, R., 2020. Stochastic interpolation of sparsely sampled time series via multipoint fractional brownian bridges. Phys. Rev. Lett. 125, 170602.

Friedrich, J., Viggiano, B., Bourgoin, M., Cal, R.B., Chevillard, L., 2022. Single inertial particle statistics in turbulent flows from lagrangian velocity models. Phys. Rev. Fluids 7, 014303.

Frisch, U., 1995. Turbulence: The Legacy of an Kolmogorov. Cambridge University Press.

Granero-Belinchon, C., 2024. Neural network based generation of a 1-dimensional stochastic field with turbulent velocity statistics. Physica D 458, 133997.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Kostinski, A.B., Shaw, R.A., 2005. Fluctuations and luck in droplet growth by coalescence. Bull. Am. Meteorol. Soc. 86, 235–244.

La Porta, A., Voth, G.A., Crawford, A.M., Alexander, J., Bodenschatz, E., 2001. Fluid particle accelerations in fully developed turbulence. Nature 409, 1017–1019.

Lamorgese, A., Pope, S.B., Yeung, P., Sawford, B.L., 2007. A conditionally cubic-gaussian stochastic lagrangian model for acceleration in isotropic turbulence. J. Fluid Mech. 582, 423–448.

Laussy, F.P., 2023. Shining light on turbulence. Nat. Photonics 17, 381–382.

Li, T., Biferale, L., Bonaccorso, F., Scarpolini, M.A., Buzzicotti, M., 2024. Synthetic lagrangian turbulence by generative diffusion models. Nat. Mach. Intell. 1–11.

Li, T., Buzzicotti, M., Biferale, L., Bonaccorso, F., 2023a. Generative adversarial networks to infer velocity components in rotating turbulent flows. Eur. Phys. J. E 46 (31).

Li, T., Buzzicotti, M., Biferale, L., Bonaccorso, F., Chen, S., Wan, M., 2023b. Multi-scale reconstruction of turbulent rotating flows with proper orthogonal decomposition and generative adversarial networks. J. Fluid Mech. 971 (A3).

Li, T., Lanotte, A.S., Buzzicotti, M., Bonaccorso, F., Biferale, L., 2023c. Multi-scale reconstruction of turbulent rotating flows with generative diffusion models. Atmosphere 15 (60).

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Lübke, J., Friedrich, J., Grauer, R., 2023. Stochastic interpolation of sparsely sampled time series by a superstatistical random process and its synthesis in fourier and wavelet space. J. Phys.: Complex. 4, 015005.

Maxey, M.R., Riley, J.J., 1983. Equation of motion for a small rigid sphere in a nonuniform flow. Phys. Fluids 26, 883–889.

Minier, J.P., Chibbaro, S., Pope, S.B., 2014. Guidelines for the formulation of lagrangian stochastic models for particle simulations of single-phase and dispersed two-phase turbulent flows. Phys. Fluids 26.

Mordant, N., Crawford, A.M., Bodenschatz, E., 2004a. Experimental lagrangian acceleration probability density function measurement. Physica D 193, 245–251.

Mordant, N., Lévêque, J.F., 2004b. Experimental and numerical study of the lagrangian dynamics of high reynolds turbulence. New J. Phys. 6 (116).

Pope, S.B., 2011. Simple models of turbulent flows. Phys. Fluids 23.
Post, S.L., Abraham, J., 2002. Modeling the outcome of drop–drop collisions in diesel sprays. Int. J. Multiph. Flow 28, 997–1019.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October (2015) 5-9, Proceedings, Part III 18. Springer, pp. 234–241.
Sawford, B., 1991. Reynolds number effects in lagrangian stochastic models of turbulent dispersion. Phys. Fluids A 3, 1577–1586.
Shaw, R.A., 2003. Particle-turbulence interactions in atmospheric clouds. Annu. Rev. Fluid Mech. 35, 183–227.
Sinhuber, M., Friedrich, J., Grauer, R., Wilczek, M., 2021. Multi-level stochastic refinement for complex time series and fields: A data-driven approach. New J. Phys. 23, 063063.
2024. Smartturb/diffusion-lagr: stable. http://dx.doi.org/10.5281/zenodo.10563386.
Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. PMLR, pp. 2256–2265.
Toschi, F., Biferale, L., Boffetta, G., Celani, A., Devenish, B., Lanotte, A., 2005. Acceleration and vortex filaments in turbulence. J. Turbul. (N15).
Toschi, F., Bodenschatz, E., 2009. Lagrangian properties of particles in turbulence. Annu. Rev. Fluid Mech. 41, 375–404.
Van Hinsberg, M., Thije Boonkkamp, J., Toschi, F., Clercx, H., 2012. On the efficiency and accuracy of interpolation methods for spectral codes. SIAM J. Sci. Comput. 34, B479–B498.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.
Viggiano, B., Friedrich, J., Volk, R., Bourgoin, M., Cal, R.B., Chevillard, L., 2020. Modelling lagrangian velocity and acceleration in turbulent flows as infinitely differentiable stochastic processes. J. Fluid Mech. 900 (A27).
Voth, G.A., Porta, A.L., Crawford, A.M., Bodenschatz, E., Ward, C., Alexander, J., 2001. A silicon strip detector system for high resolution particle tracking in turbulence. Rev. Sci. Instrum. 72, 4348–4353.
Weng, L., 2021. What are diffusion models? URL https://lilianweng.github.io/posts/2021-07-11-diffusion-models/, lilianweng.github.io.
Wilson, J.D., Sawford, B.L., 1996. Review of lagrangian stochastic models for trajectories in the turbulent atmosphere. Bound.-Layer Meteorol. 78, 191–210.
Wrench, D., Parashar, T.N., Oughton, S., de Lange, K., Frean, M., 2024. What is the reynolds number of the solar wind? Astrophys. J. 961, 182.
Xia, H., Francois, N., Punzmann, H., Shats, M., 2013. Lagrangian scale of particle dispersion in turbulence. Nat. Commun. 4 (2013).
Yeung, P., 2002. Lagrangian investigations of turbulence. Annu. Rev. Fluid Mech. 34, 115–142.
Yu, L., Yousif, M.Z., Zhang, M., Hoyas, S., Vinuesa, R., Lim, H.C., 2022. Three-dimensional esrgan for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning. Phys. Fluids 34.
Zamansky, R., 2022. Acceleration scaling and stochastic dynamics of a fluid particle in turbulence. Phys. Rev. Fluids 7, 084608.
Zheng, Q., Li, T., Ma, B., Fu, L., Li, X., 2024. High-fidelity reconstruction of large-area damaged turbulent fields with a physically constrained generative adversarial network. Phys. Rev. Fluids 9, 024608.