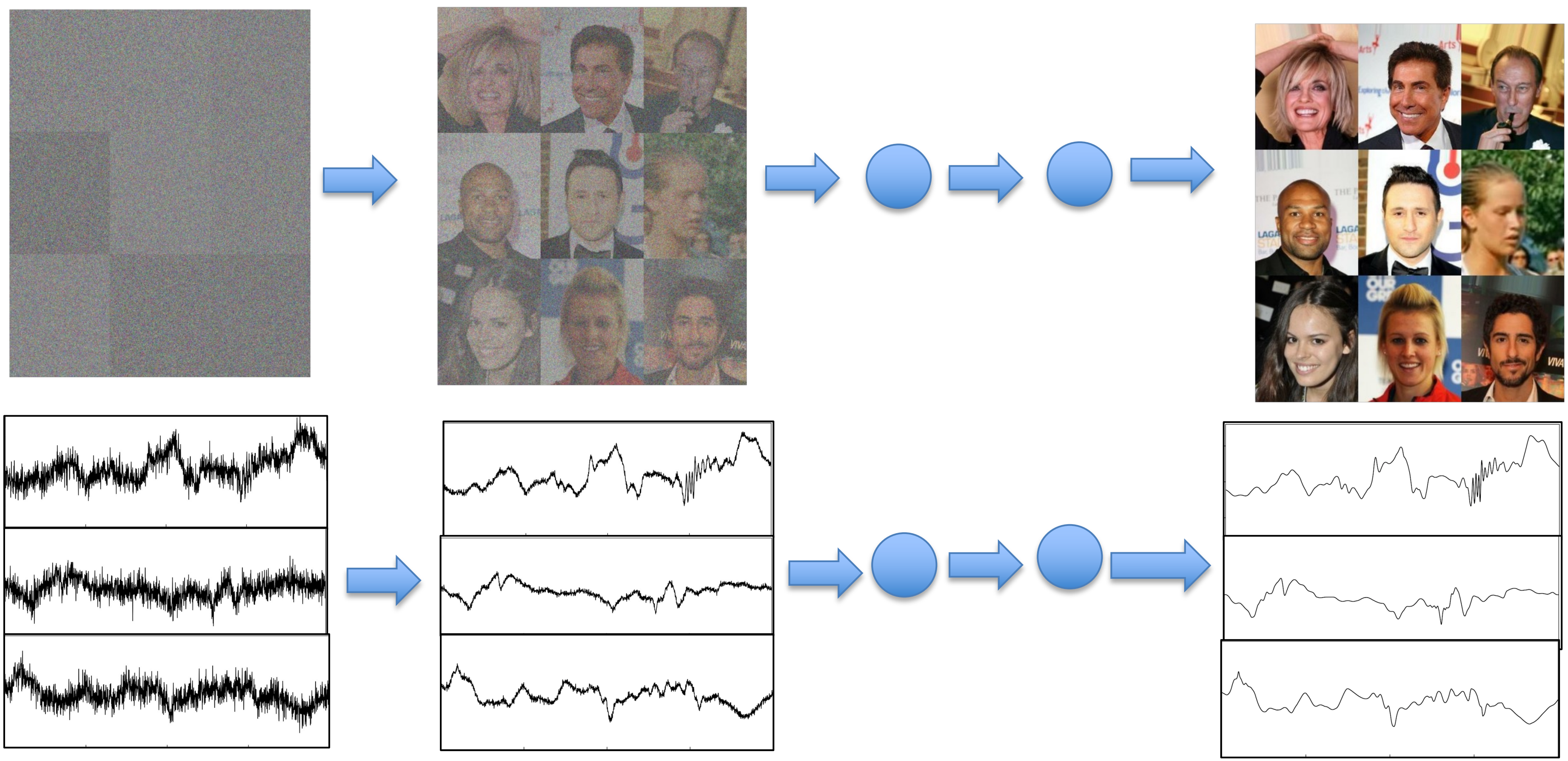
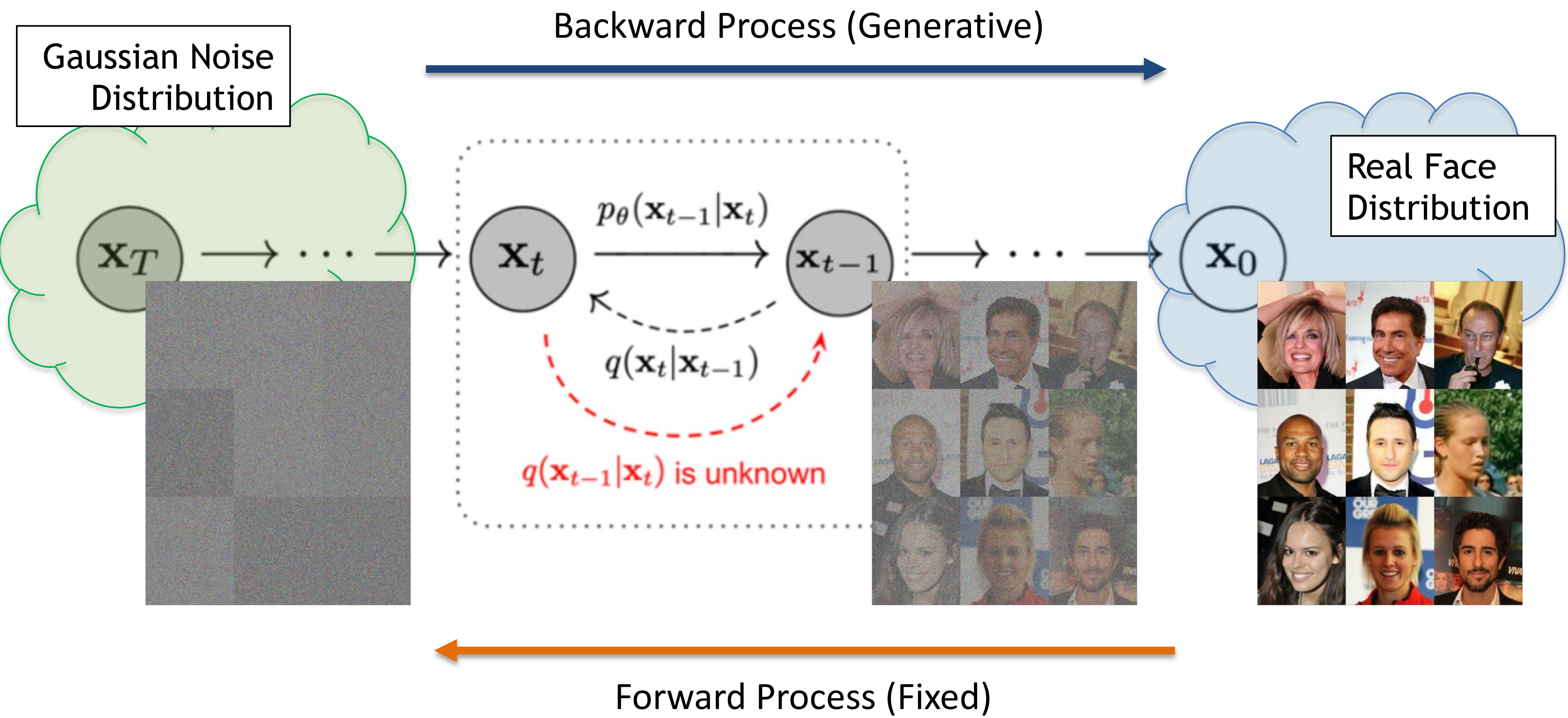
Complex Fluids and Complex Flows Group
Dept. Physics & INFN - University of Rome 'Tor Vergata'
michele.buzzicotti@roma2.infn.it
https://mbuzzico.github.io/



Synthetic Lagrangian Turbulence by Generative Diffusion Models

CREDITS: T. LI, L. BIFERALE, F. BONACCORSO, M. SCARPOLINI

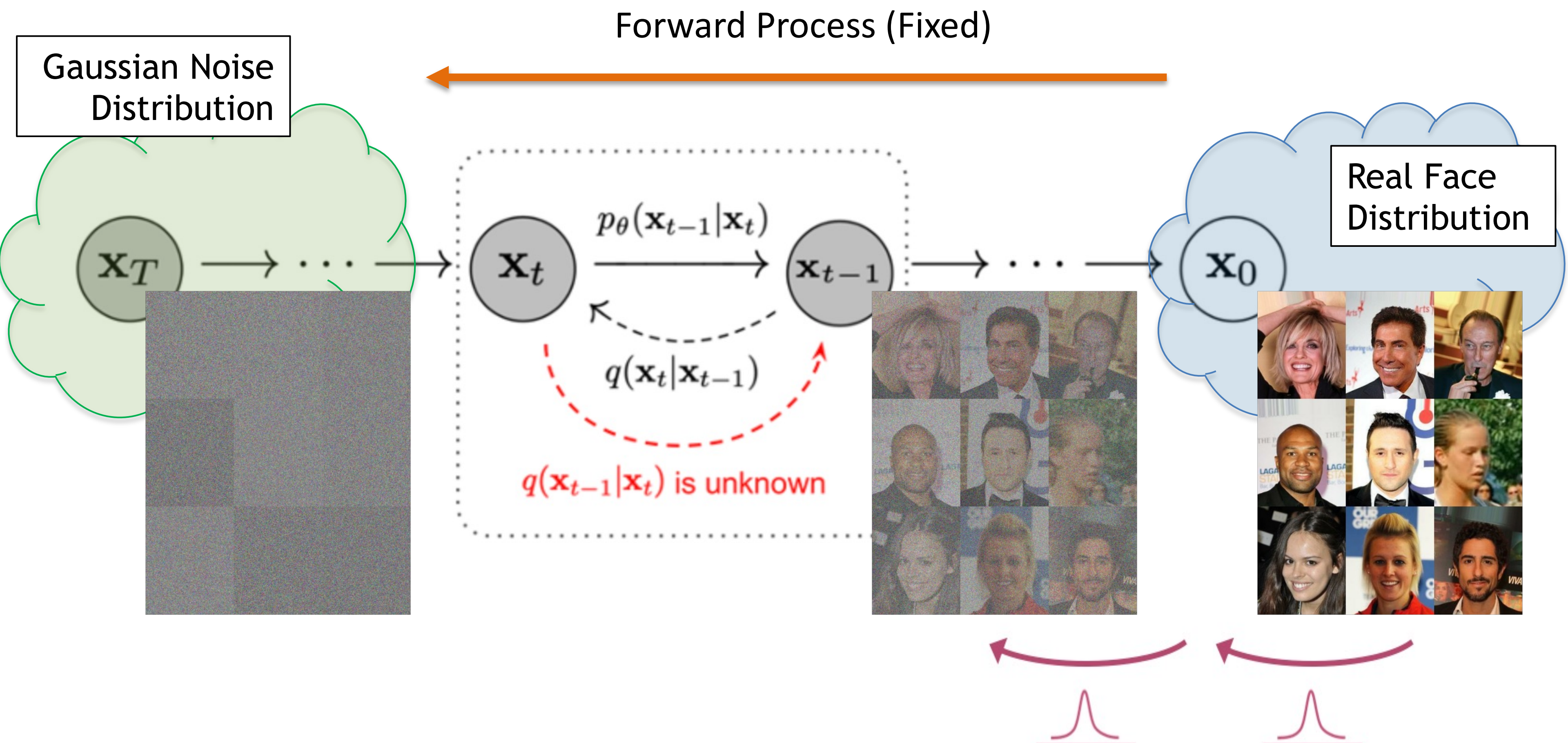
Denoising Diffusion Models



Learning to generate by denoising

Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

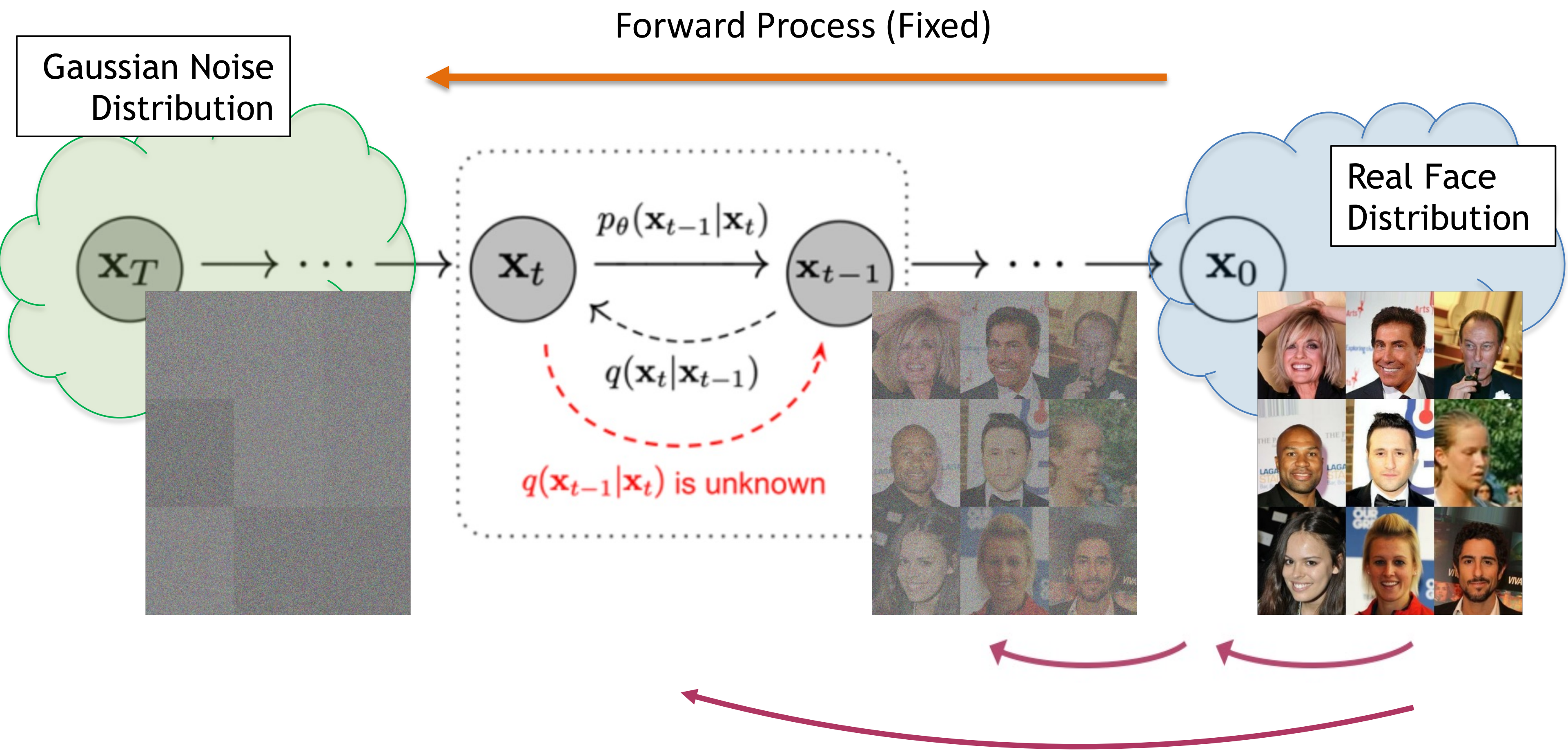
Forward Diffusion Process



The formal definition of the forward process in T steps:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Forward Diffusion Process

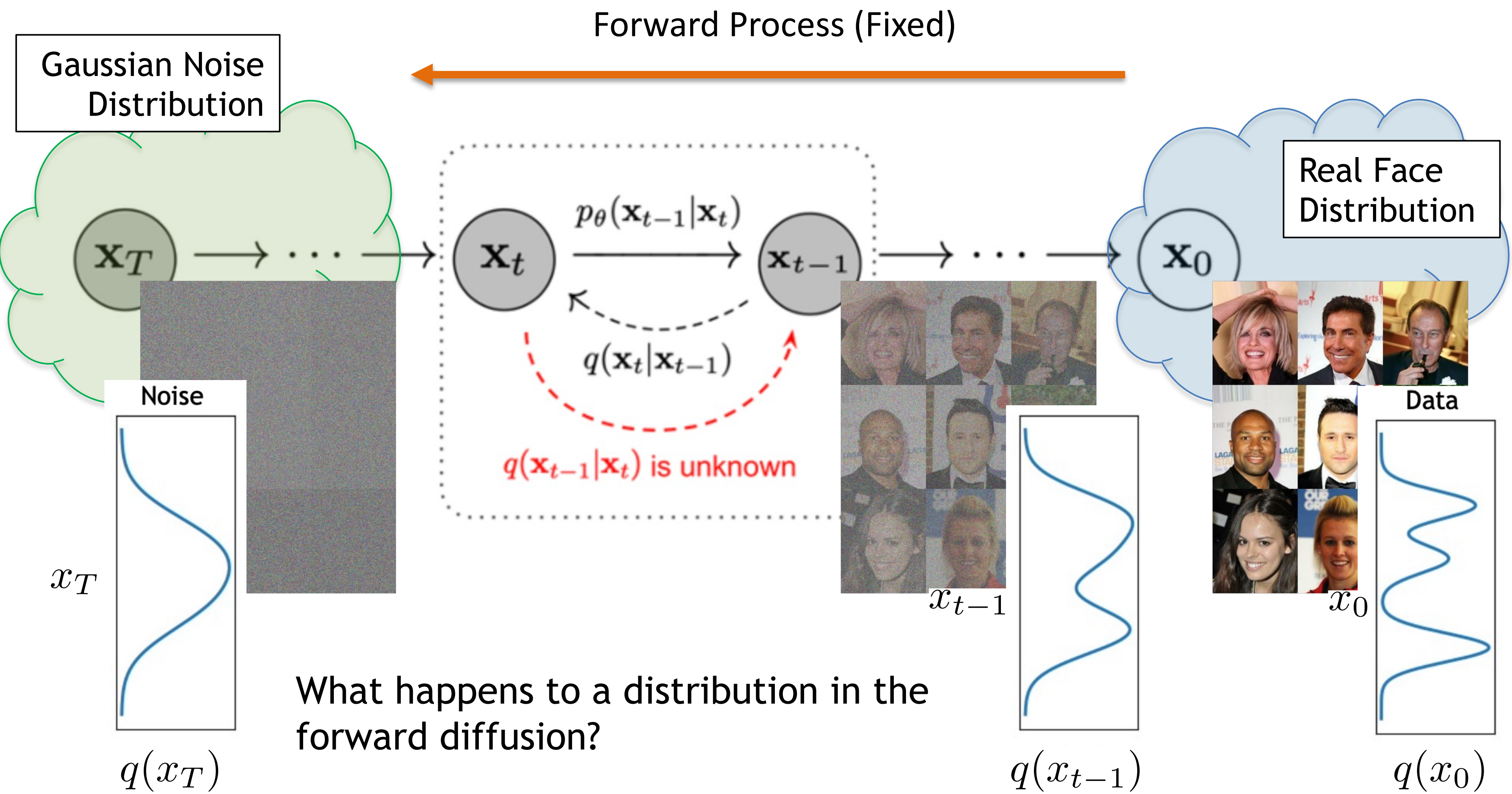


Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\;\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\;\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \to 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

Forward Diffusion Process

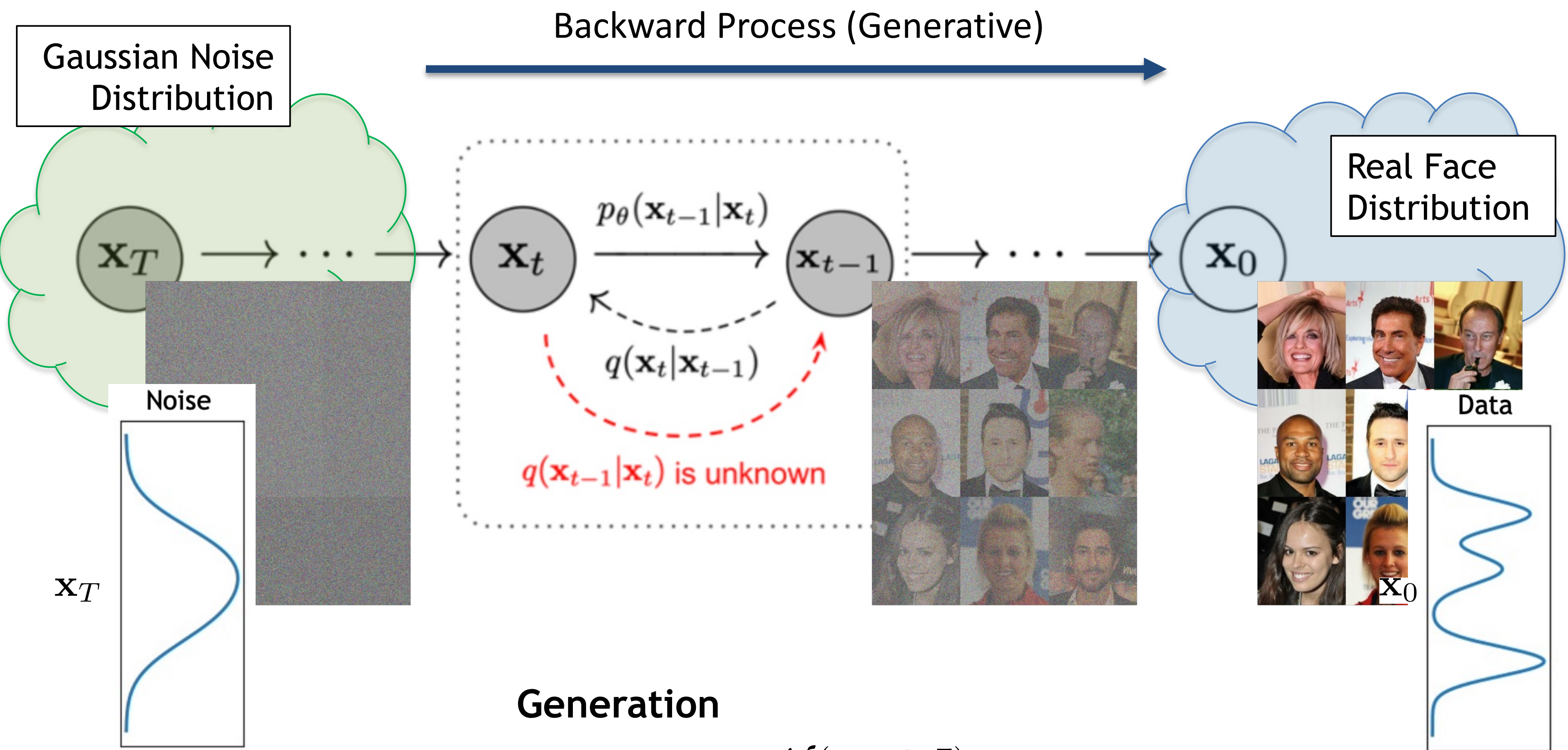


What happens to a distribution in the forward diffusion?

$$\underbrace{q(\mathbf{x}_t)}_{\text{Diffused data dist.}} = \int \underbrace{q(\mathbf{x}_0, \mathbf{x}_t)}_{\text{Joint dist.}} \, d\mathbf{x}_0 = \int \underbrace{q(\mathbf{x}_0)}_{\text{Input data dist.}} \, \underbrace{q(\mathbf{x}_t|\mathbf{x}_0)}_{\text{Diffusion kernel}} \, d\mathbf{x}_0$$

Gaussian Convolution

Backward Diffusion Process



$q(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

$q(\mathbf{x}_0)$

Generation

Sample: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

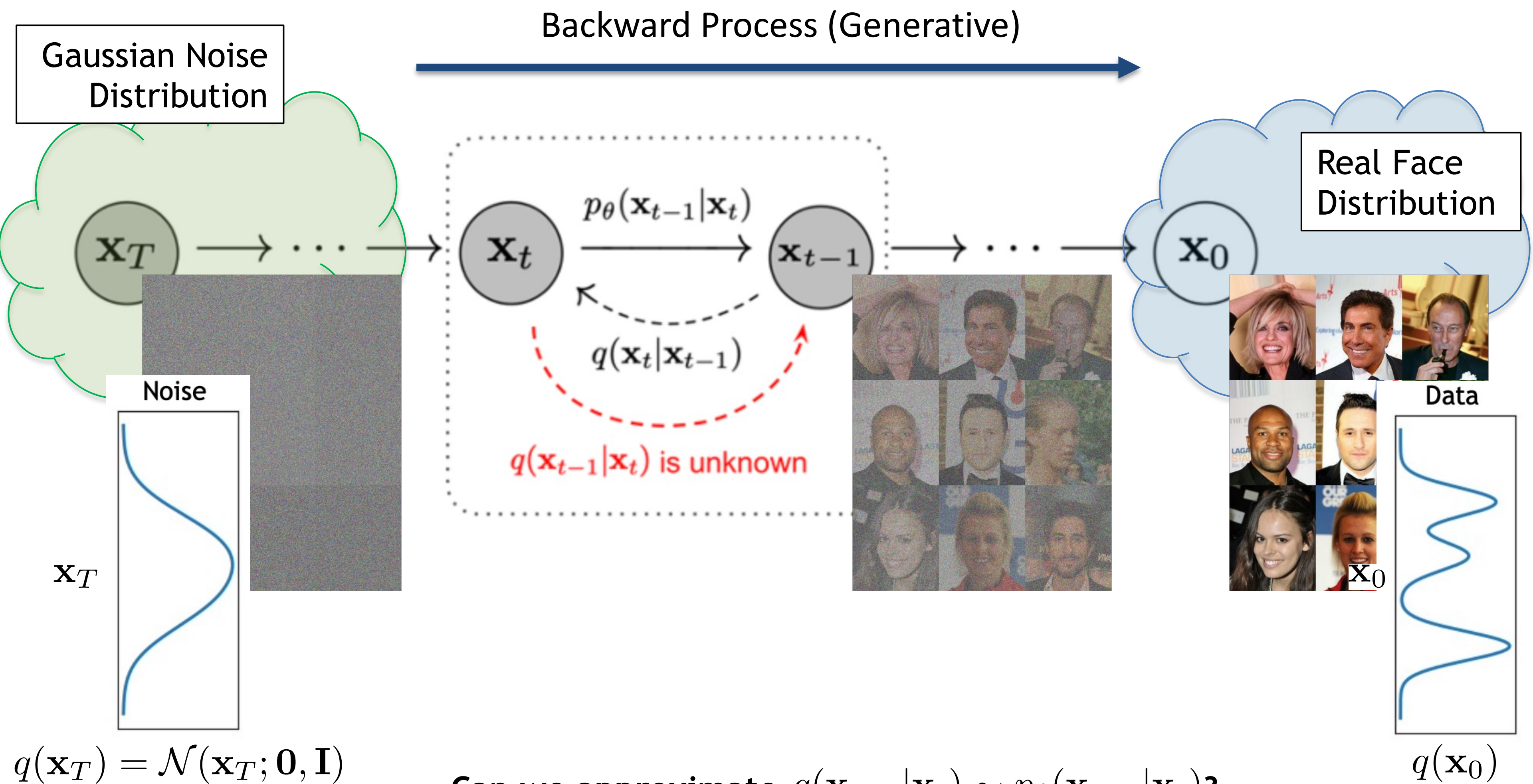
Iterate: $\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, untill $t = 1$

True Denoise

Intractable!

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}) \frac{q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)}$$

Approx. Backward Diffusion Process

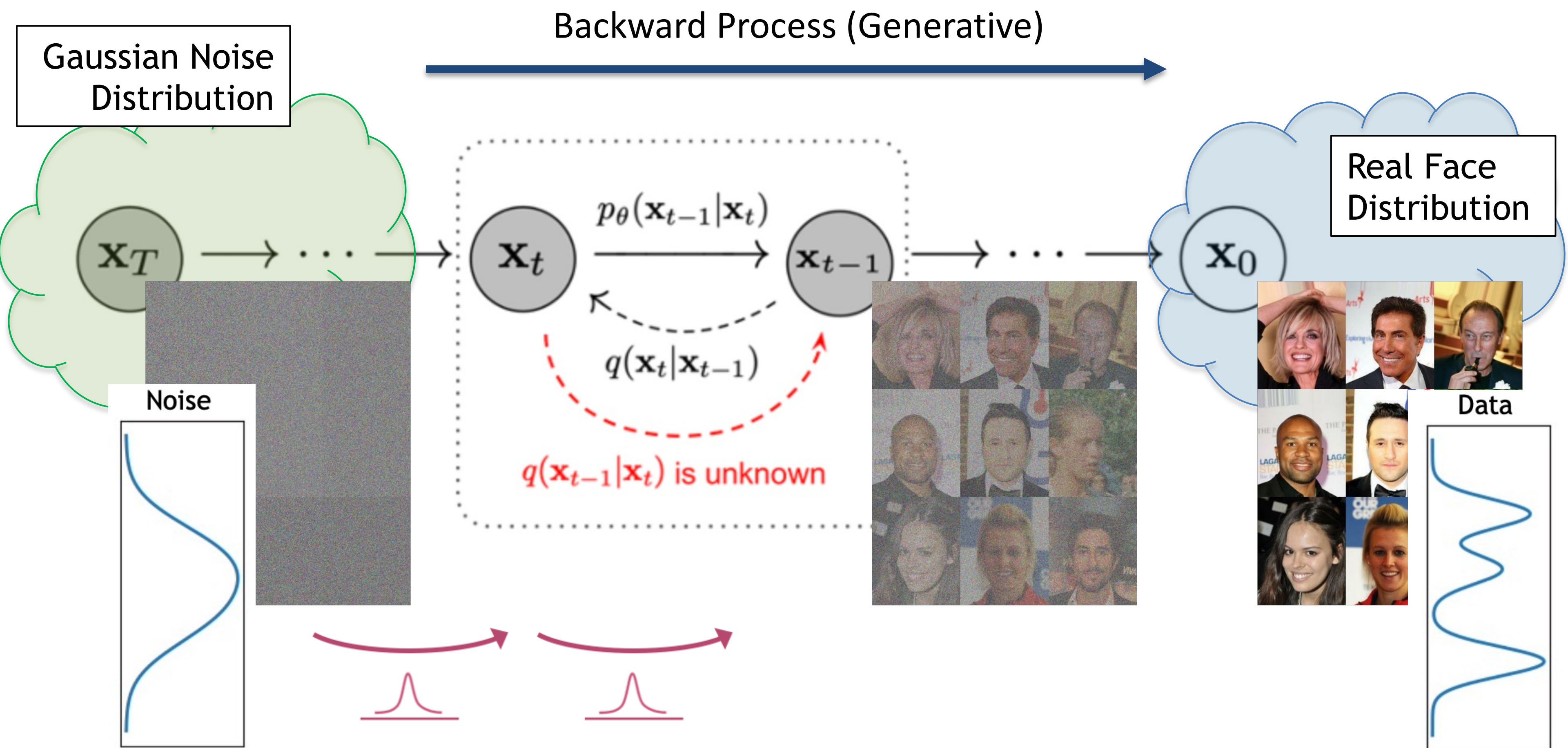


Can we approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$**?**

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Yes, we can use a Normal distribution if β_t is small in each forward diffusion step.

Approx. Backward Diffusion Process



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{}, \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Trainable network (U-net, Denoising Autoencoder)

Learning the Backward Diffusion Process

Variational upper bound

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)]$$

During training, the optimization involves **minimizing the cross entropy** between the ground truth distribution and the likelihood of the generated data

$$\mathbb{E}_{q(\mathbf{x}_0)}[..] = \int [..] q(\mathbf{x}_0) d\mathbf{x}_0 \qquad p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) d\mathbf{x}_{1:T}$$

For training, we can form **variational upper bound** that is commonly used for training variational autoencoders:

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] =: L$$

Sohl-Dickstein et al. ICML 2015 (Appendix B for all details) show that:

$$L = \mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1))}_{L_0}\right]$$

Learning the Backward Diffusion Process

Variational upper bound

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)]$$

During training, the optimization involves **minimizing the cross entropy** between the ground truth distribution and the likelihood of the generated data

$$\mathbb{E}_{q(\mathbf{x}_0)}[..] = \int [..] q(\mathbf{x}_0) d\mathbf{x}_0$$

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) d\mathbf{x}_{1:T}$$

For training, we can form **variational upper bound** that is commonly used for training variational autoencoders:

$$\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_{q(\mathbf{x}_0) q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] =: L$$

Sohl-Dickstein et al. ICML 2015 (Appendix B for all details) show that:

$$L = \mathbb{E}_q \left[\cancel{\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T}} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

$$D_{KL}(p(x)||q(x)) = \int_{-\infty}^{+\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

KULLBACK-LEIBLER Divergence

Learning the Backward Diffusion Process

Variational upper bound

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is the tractable posterior distribution:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}.$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp\left(-\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{2\beta_t}\right) \cdot \exp\left(-\frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{2(1-\bar{\alpha}_{t-1})}\right) \cdot \exp\left(\frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{2(1-\bar{\alpha}_t)}\right)$$

$$\boxed{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_n\mathbf{I})}$$

$$\text{where } \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \text{ and } \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Sohl-Dickstein et al. ICML 2015 (Appendix B for all details) show that:

$$L = \mathbb{E}_q\left[\cancel{\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T}} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right]$$

$$D_{KL}(p(x)||q(x)) = \int_{-\infty}^{+\infty} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx$$

KULLBACK-LEIBLER Divergence

Learning the Backward Diffusion Process

Variational upper bound

We need to evaluate the KL-Divergence between two Gaussians!

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}) \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$$

We end up with the following Loss function:

$$\boxed{L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} ||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2 \right]}$$

Sohl-Dickstein et al. ICML 2015 (Appendix B for all details) show that:

$$L = \mathbb{E}_q \left[\cancel{\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T}} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

$$D_{KL}(p(x)||q(x)) = \int_{-\infty}^{+\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

KULLBACK-LEIBLER Divergence

Learning the Backward Diffusion Process

Variational upper bound

$$\boxed{L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2\right]}$$

Recalling the properties of the forward process:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$$

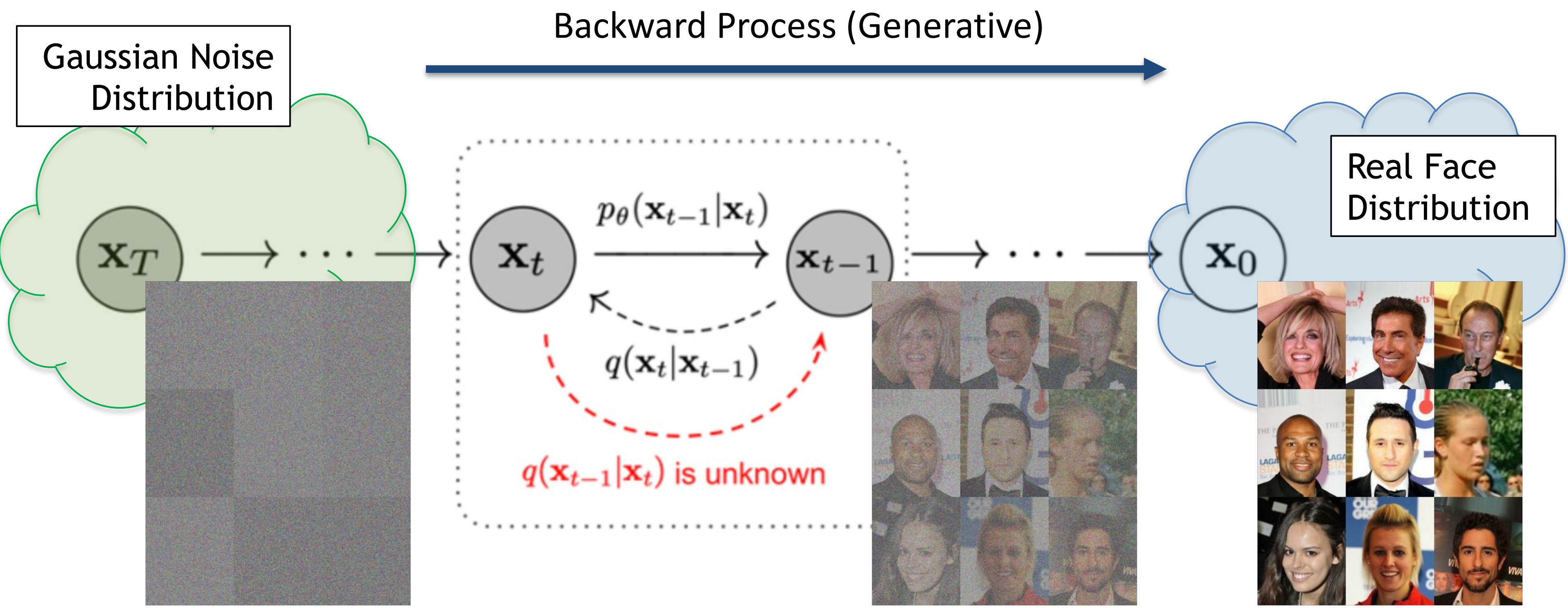
Ho et al. NeurIPS 2020 observed that:

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) \qquad \mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\right)$$

They propose to **reparametrize the denoising model** using a *noise-prediction* network:

$$\boxed{L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t}, t)||^2\right]}$$

Recap [Main steps]



1) Backward step approximation → $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$

2) Optimization of the cross entropy → $\mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)]$

3) Variational Upper Bound → $\mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$

4) Reparametrization of the Loss → $L_{t-1} \propto ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2$

In Action

(Model Training and Sample Generation)

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boxed{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}, t) \right\|^2$
until converged

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\, \epsilon$$

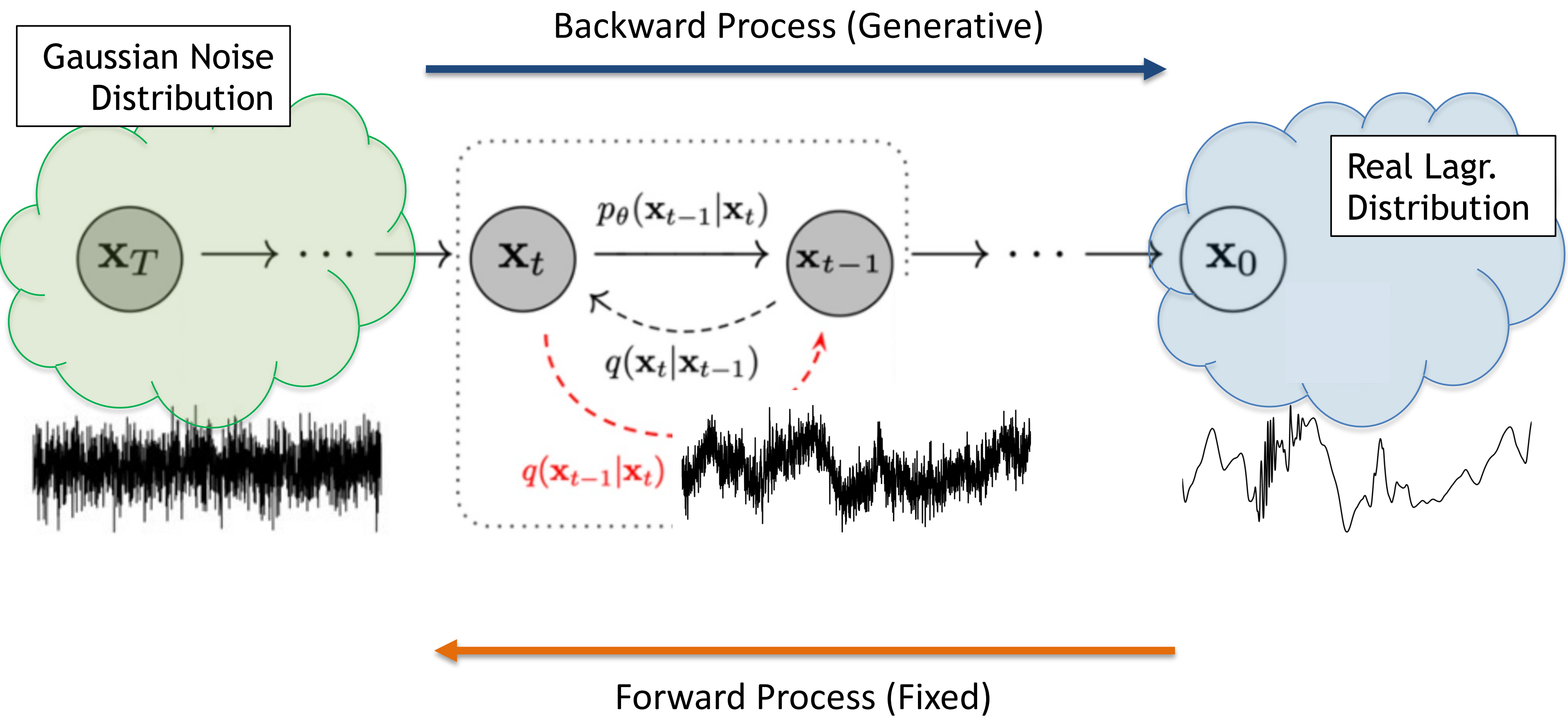
Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)} + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\right)$$

Diffusion Models for Lagrangian Turbulence



Learning to generate by denoising

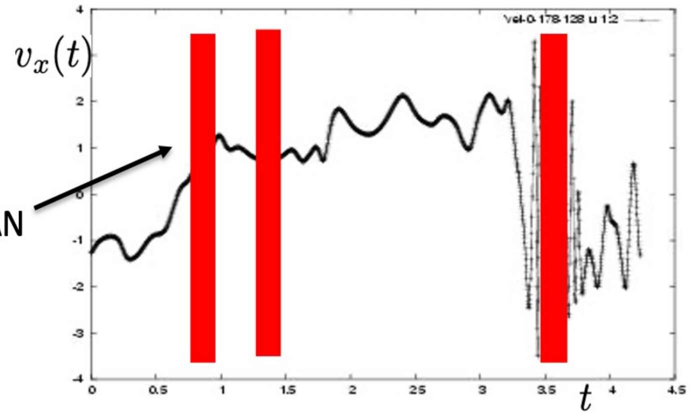
T. Li, L. Biferale, F. Bonaccorso, M. Scarpolini and M. Buzzicotti.
Synthetic Lagrangian Turbulence by Generative Diffusion Models.
arXiv:2307.08529 (2023) – Submitted to Nature Machine Intelligence

STOCHASTIC MODELS FOR LAGRANGIAN TURBULENCE: WHY?

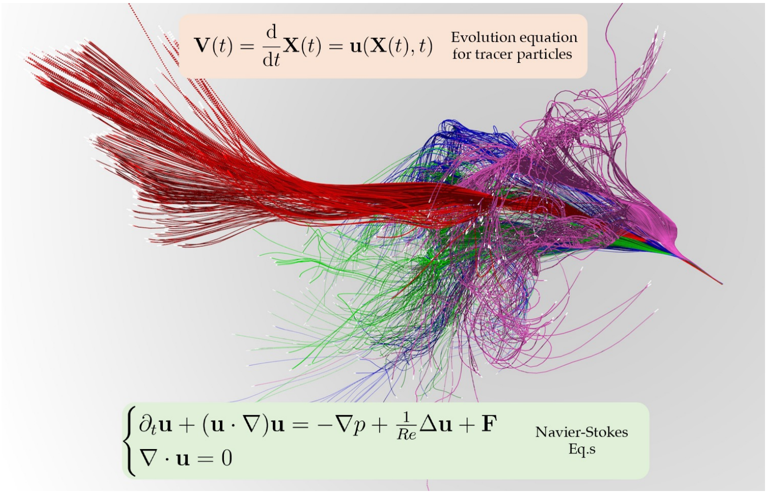
GENERATION OF LARGE SYNTHETIC DATA-BASE FOR
(I) RANKING OF PHYSICS FEATURES
(II) TESTING DOWNSTREAM APPLICATIONS/MODELS

DATA ASSIMILATION/INPAINTING FROM MISSING FIELD/EXPERIMENTAL OBSERVATION

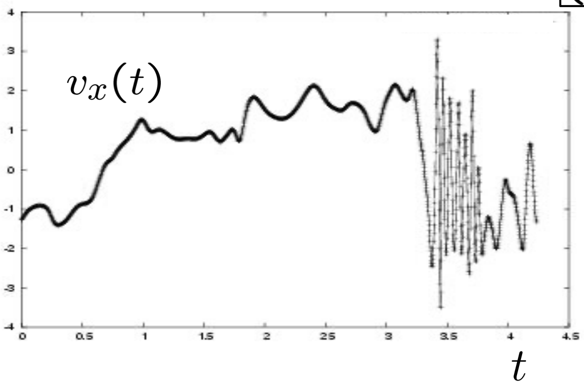
LAGRANGIAN

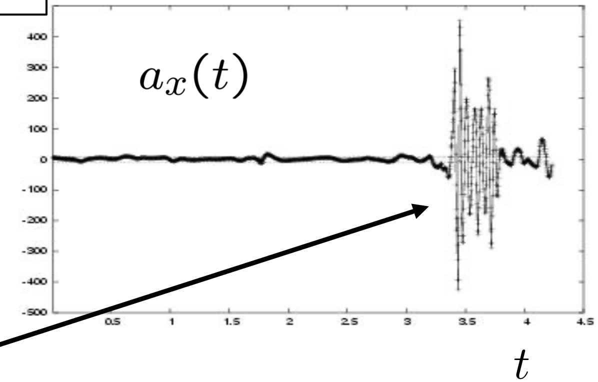


CLASSIFICATION/INFERRAL OF MISSING/INTERNAL PROPERTIES:
(I) INERTIA
(II) SHAPE
(III) ACTIVE DEGREES OF FREEDOM
(IV) ....

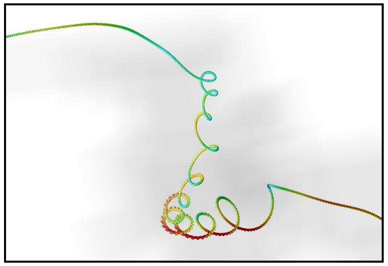


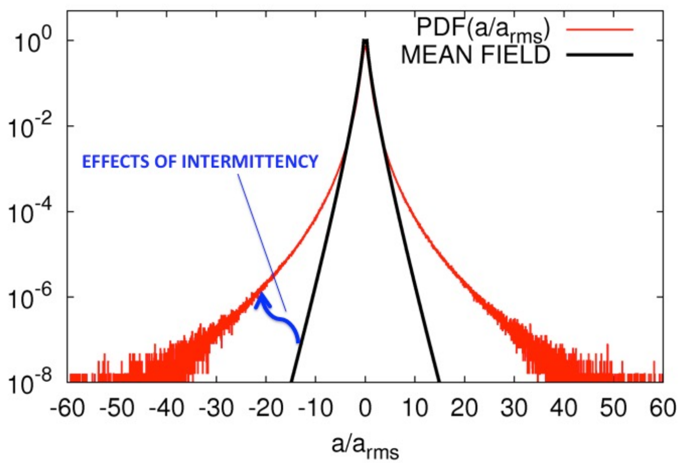
$$\begin{cases} \mathbf{a} = \partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nu\Delta\mathbf{u} + \mathbf{f} \\ \nabla\cdot\mathbf{u} = 0 \end{cases}$$





EXTREME EVENTS



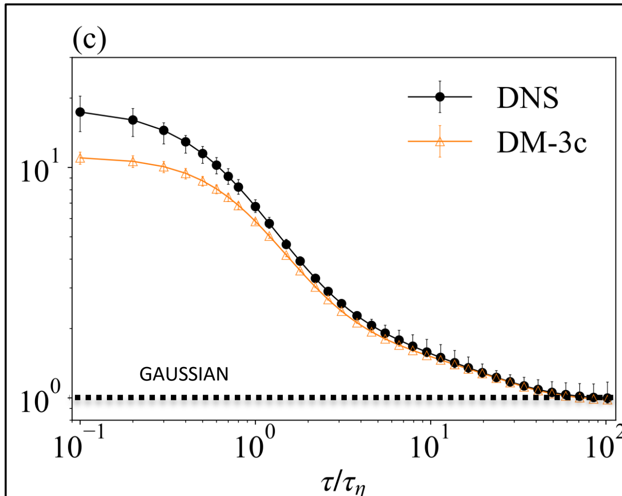


ACCELERATION PROBABILITY DISTRIBUTION FUNCTION (PDF) AT RE ~ 10^5 [Bi04]
COMPARED WITH THE PREDICTION FROM MEAN FIELD (KOLMOGOROV THEORY)

La Porta, G.A. Voth, A.M. Crawford, J. Alexander et al. Fluid particle accelerations in fully developed turbulence. Nature, 409(6823), 1017 (2001)

N. Mordant, P. Metz, O. Michel and J.F. Pinton. Measurement of Lagrangian velocity in fully developed turbulence. Phys. Rev. Lett. 87(21), 214501 (2001)

F. Toschi and E. Bodenschatz. Lagrangian Properties of Particles in Turbulence. Annu. Rev. Fluid Mech. 41, 375 (2009)



$$\delta_\tau V_i(t) = V_i(t+\tau) - V_i(t),$$
$$S_\tau^{(p)} = \langle(\delta_\tau V_i)^p\rangle$$
$$F_\tau^{(p)} = S_\tau^{(p)}/[S_\tau^{(2)}]^{p/2}$$

L Biferale, G Boffetta, A Celani, A Lanotte, F Toschi. Particle trapping in three-dimensional fully developed turbulence Physics of Fluids 17 (2), 021701 (2005)

~30 years of modeling experience

Sawford, B. L. Reynolds number effects in Lagrangian stochastic models of turbulent dispersion. Phys. Fluids A: Fluid Dyn. 3, 1577–1586 (1991).

Wilson, J. D. & Sawford, B. L. Review of lagrangian stochastic models for trajectories in the turbulent atmosphere. Boundary-layer meteorology 78, 191–210 (1996).

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

Arneodo, A., Bacry, E. & Muzy, J.-F. Random cascades on wavelet dyadic trees. Journal of Mathematical Physics 39, 4142–4164 (1998).

Lamorgese, A., Pope, S. B., Yeung, P. & Sawford, B. L. A conditionally cubic-gaussian stochastic lagrangian model for acceleration in isotropic turbulence. Journal of Fluid Mechanics 582, 423-448 (2007).

Arnéodo, A. et al. Universal intermittent properties of particle trajectories in highly turbulent flows. Physical Review Letters 100, 254504 (2008)

Pope, S. B. Simple models of turbulent flows. Physics of Fluids 23, 011301 (2011).

Minier, J.-P., Chibbaro, S. & Pope, S. B. Guidelines for the formulation of lagrangian stochastic models for particle simulations of single-phase and dispersed two-phase turbulent flows. Physics of Fluids 26, 113303 (2014).

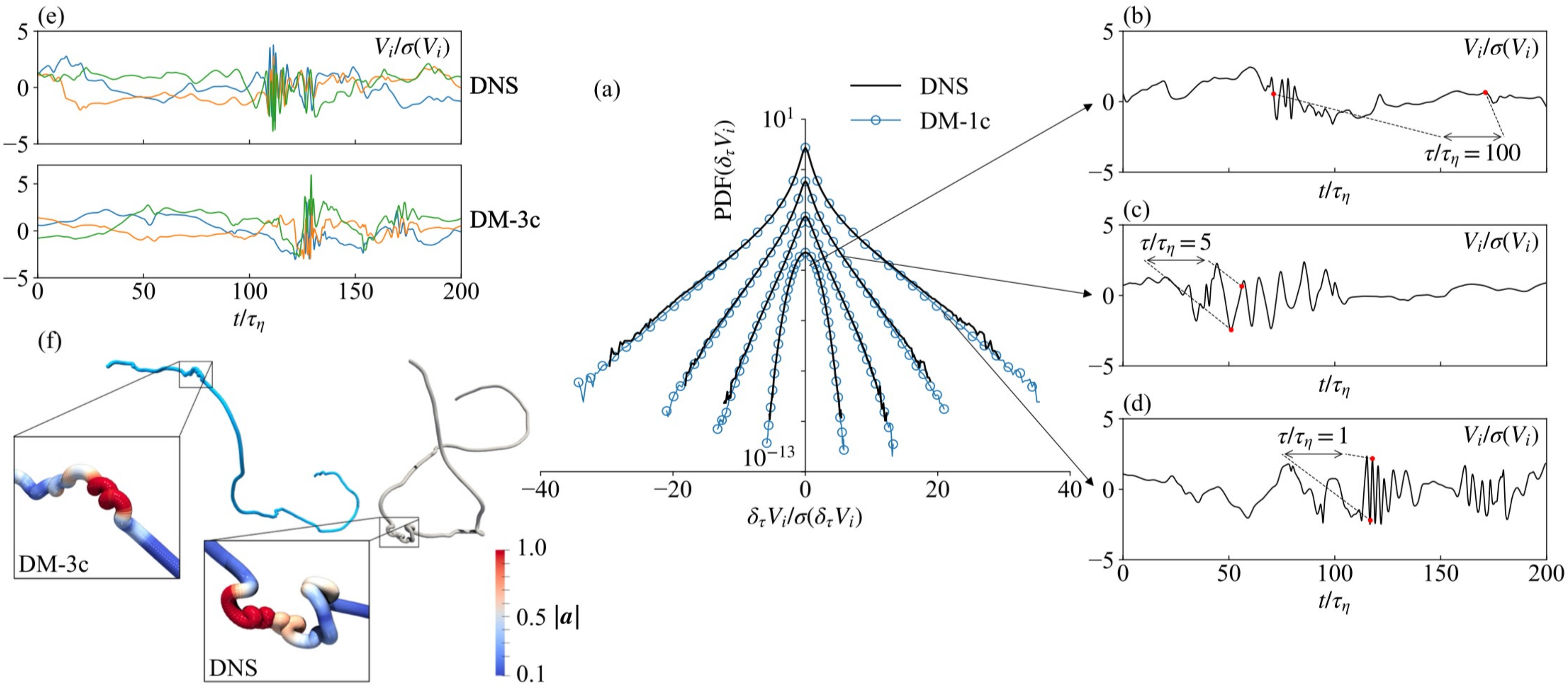
Chevillard, L., Garban, C., Rhodes, R. & Vargas, V. On a skewed and multifractal unidimensional random field, as a probabilistic representation of kolmogorov’s views on turbulence. In Annales Henri Poincaré, vol. 20, 3693–3741 (Springer, 2019).

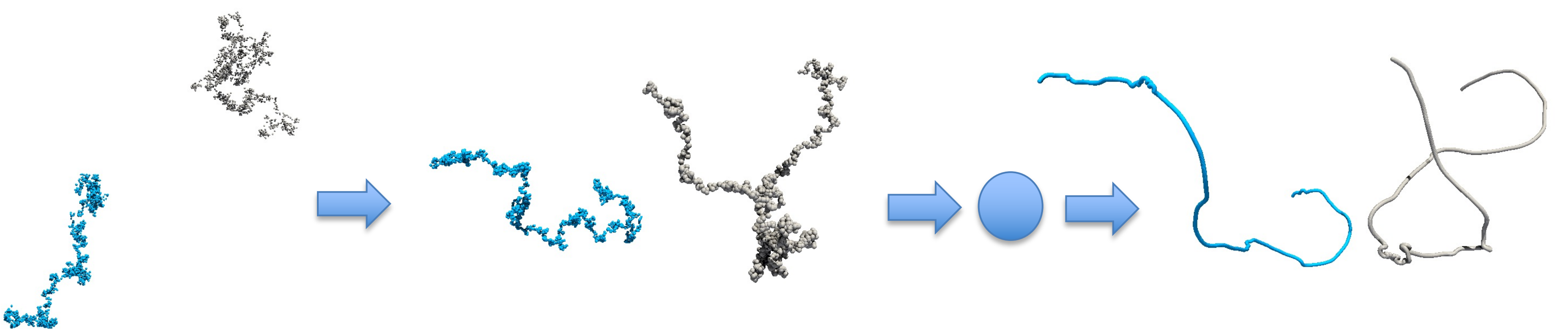
Viggiano, B. et al. Modelling lagrangian velocity and acceleration in turbulent flows as infinitely differentiable stochastic processes. Journal of Fluid Mechanics 900, A27 (2020).

Sinhuber, M., Friedrich, J., Grauer, R. & Wilczek, M. Multi-level stochastic refinement for complex time series and fields: a data-driven approach. New Journal of Physics 23, 063063 (2021).

Zamansky, R. Acceleration scaling and stochastic dynamics of a fluid particle in turbulence. Physical Review Fluids 7, 084608 (2022).

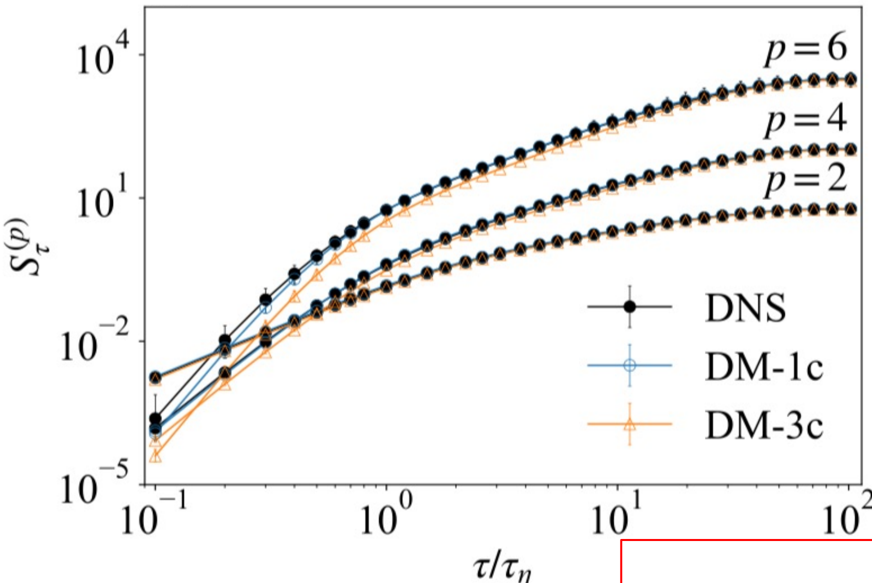
$$\delta_\tau V_i(t) = V_i(t+\tau) - V_i(t),$$





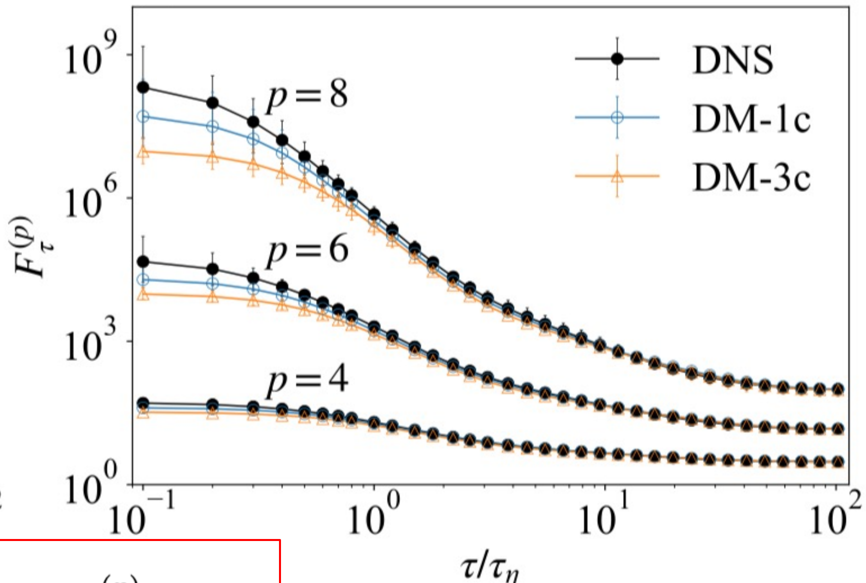
LAGRANGIAN STRUCTURE FUNCTIONS

$$S_\tau^{(p)} = \langle (\delta_\tau V_i)^p \rangle$$

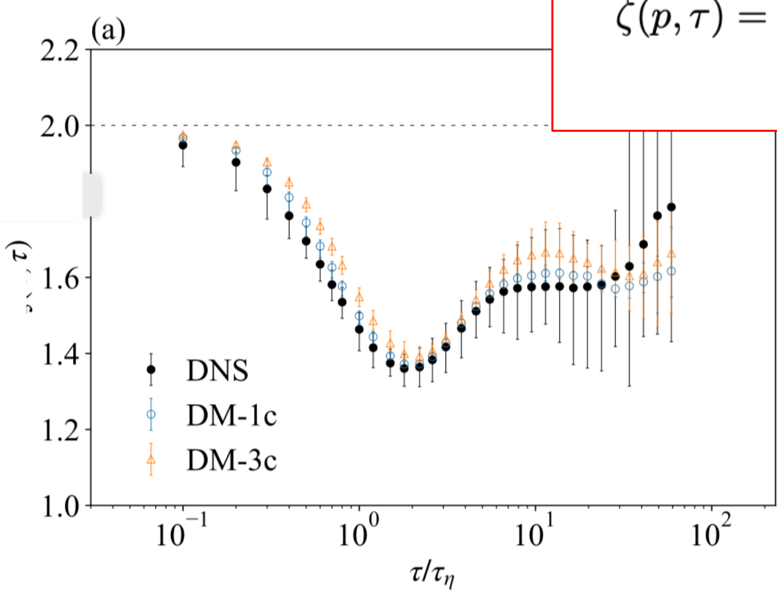


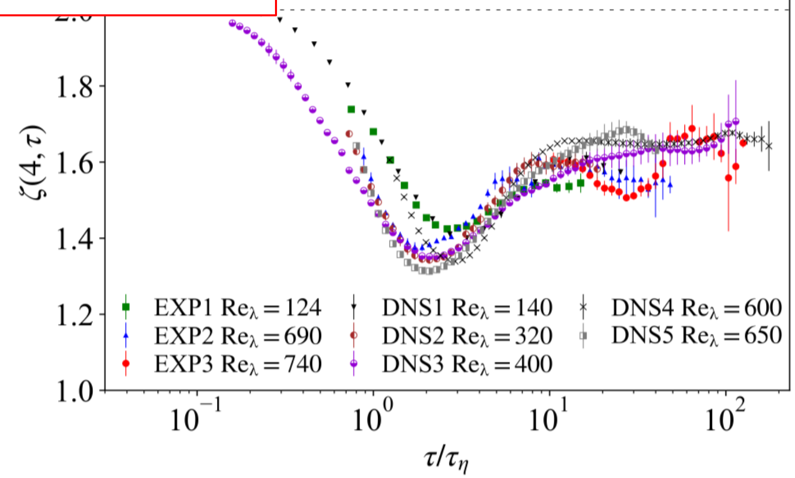
GENERALIZED FLATNESS

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

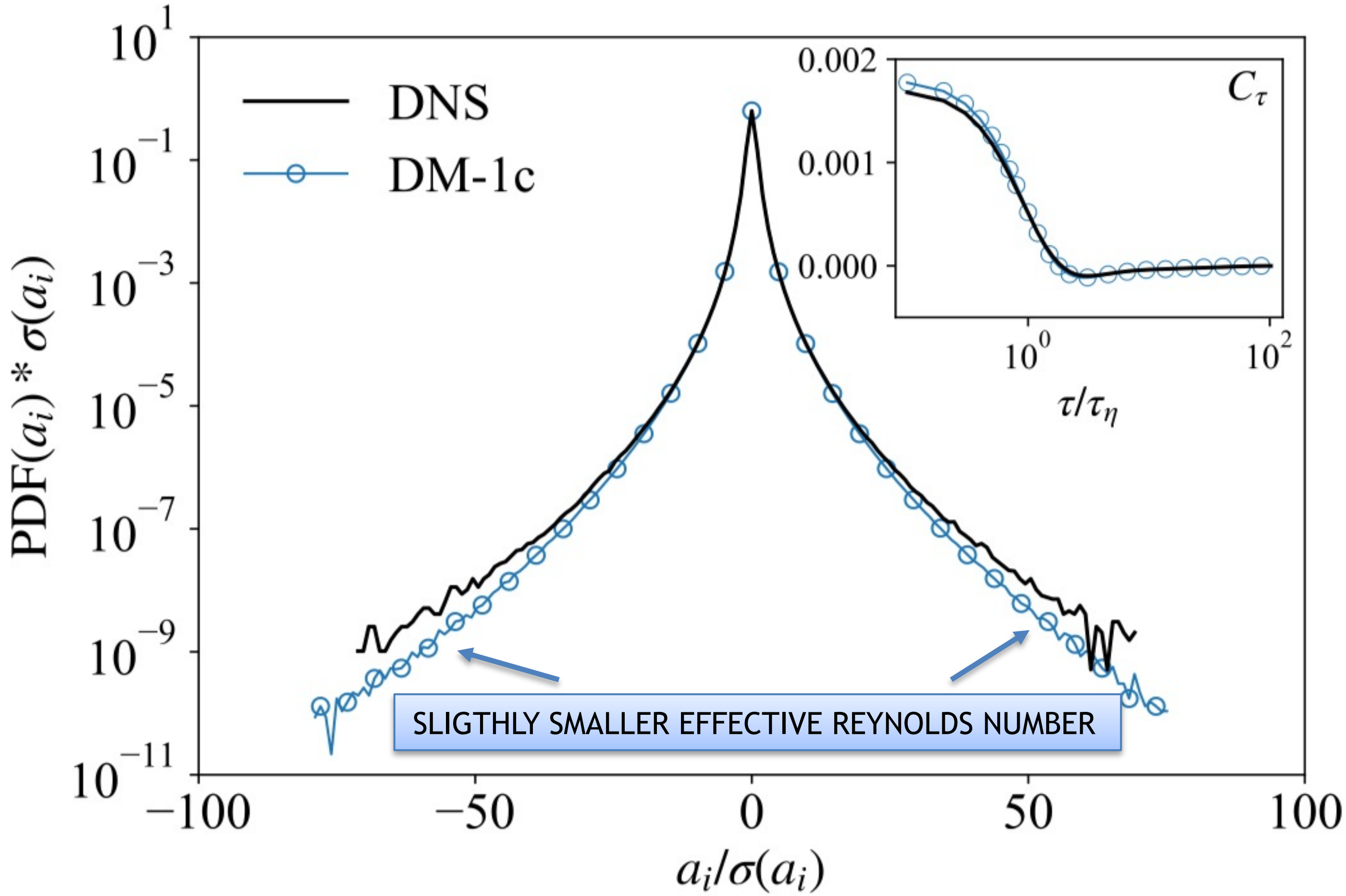


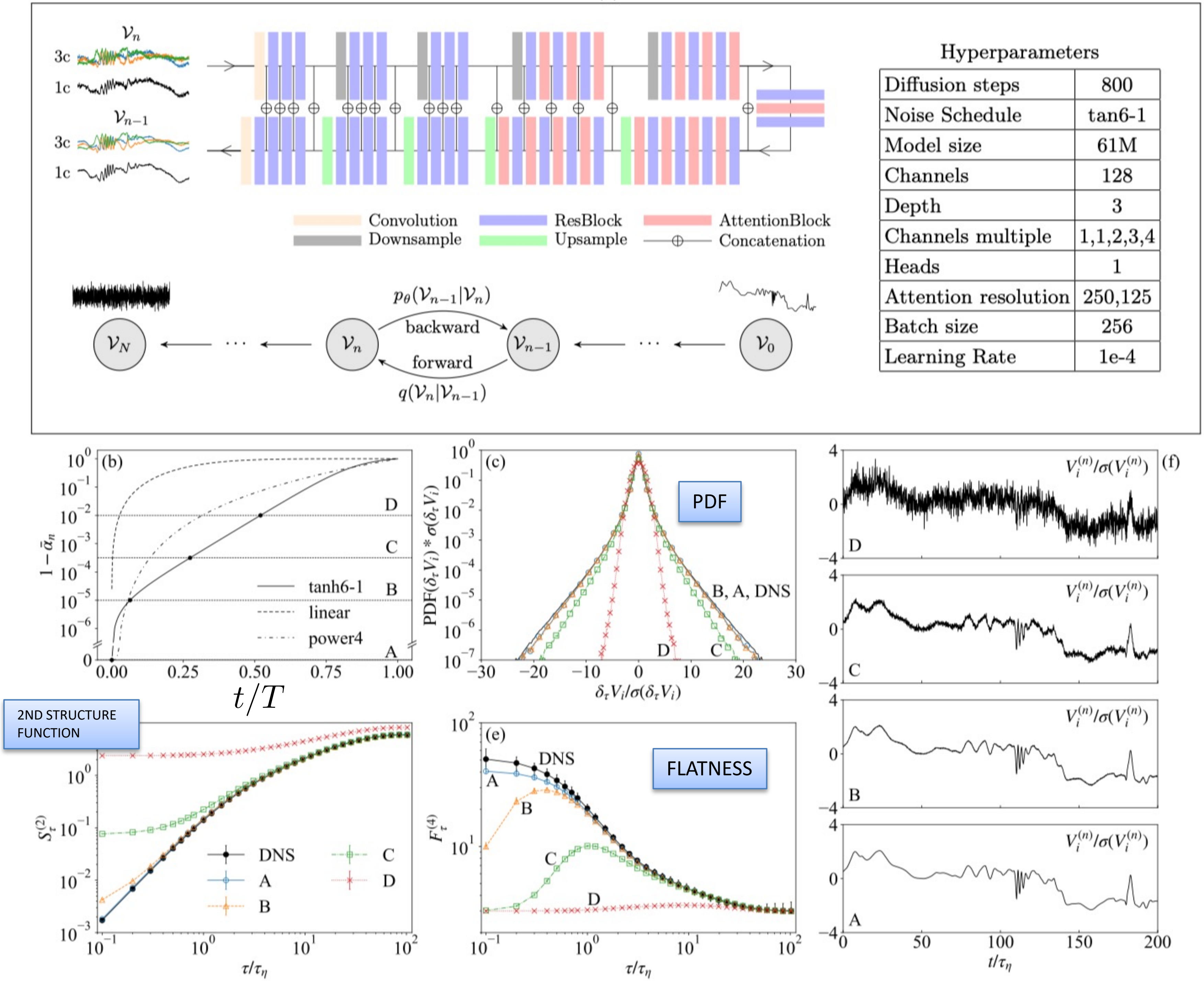
$$\zeta(p,\tau) = \frac{d\ \log S_\tau^{(p)}}{d\ \log S_\tau^{(2)}}.$$





ACCELERATION PDF





Hyperparameters	
Diffusion steps	800
Noise Schedule	tan6-1
Model size	61M
Channels	128
Depth	3
Channels multiple	1,1,2,3,4
Heads	1
Attention resolution	250,125
Batch size	256
Learning Rate	1e-4

WHAT WE HAVE:

- QUICK STOCHASTIC TOOL TO GENERATE REALISTIC 3D TRAJECTORIES OF TRACERS IN HOMOGENEOUS AND ISOTROPIC TURBULENCE, EASY TO GENERALISE FOR DIFFERENT APPLICATIONS

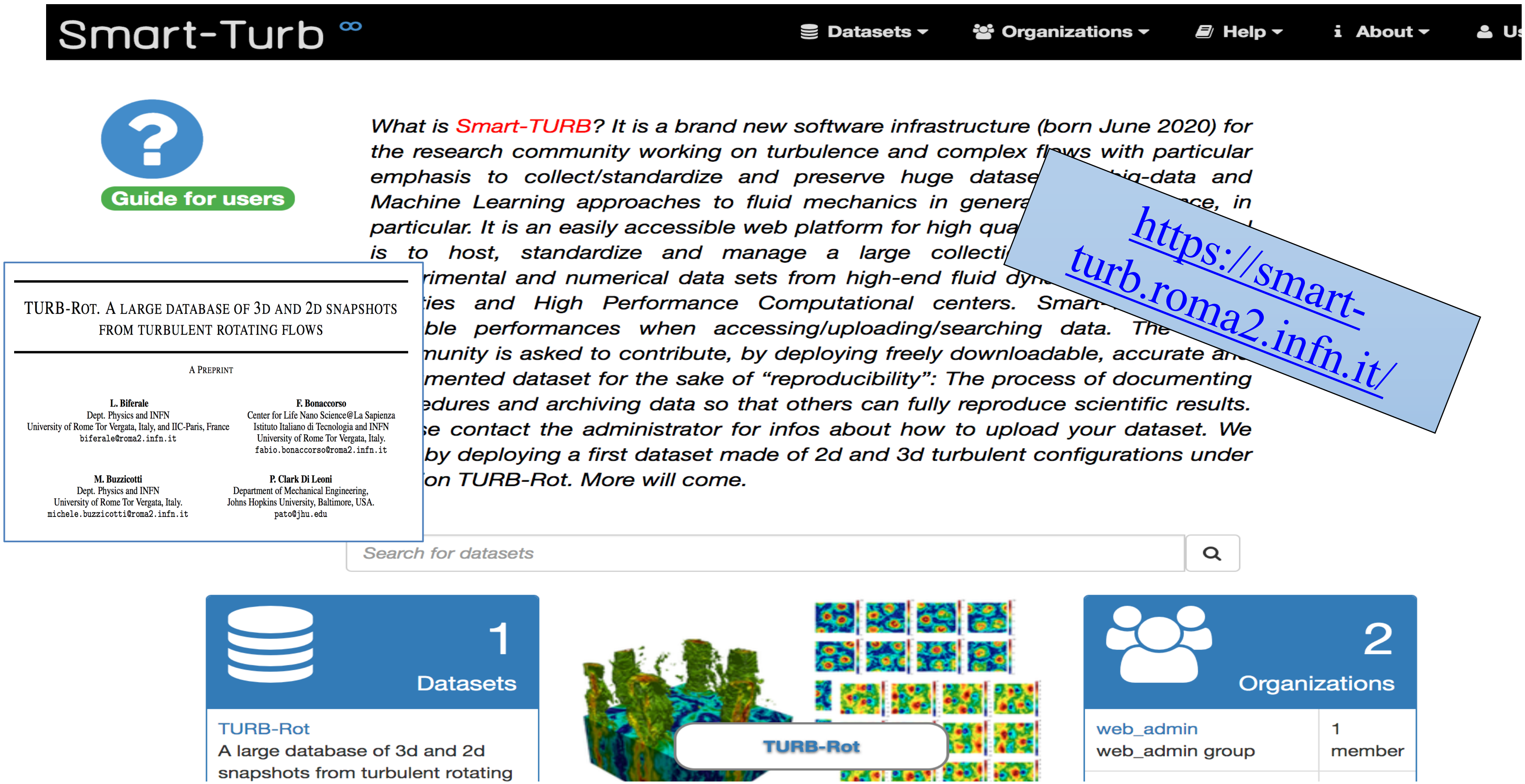
- IMPRESSIVE QUANTITATIVE AGREEMENT WITH MULTI-SCALE STATISTICAL PROPERTIES

WHAT WE MISS:

- UNDERSTADING OF ROBUSTNESS IN GENERALISING OUT-OF-SAMPLE: EXTREME EVENTS, DIFFERENT REYNOLDS NUMBERS, DIFFERENT PARTICLES' PROPERTIES

- UNDERSTANDING SCALING PROPERTIES FOR TIME-TO-SOLUTION AT CHANGING IN-SAMPLE PROPERTIES, I.E. AT CHANGING DIMENSION OF THE TRAINING DATASET, SETS OF HYPER-PARAMETERS, CNN ARCHITECTURES: GAN, DM, TRANSFORMERS

-WHAT-IF QUESTIONS: EXPLICABILITY OF THE GENERATED DATA, FEATURES RANKINGS, PHYSICS DISCOVERY


Smart-Turb
Datasets
Organizations
Help
About
Guide for users
What is Smart-TURB? It is a brand new software infrastructure (born June 2020) for the research community working on turbulence and complex flows with particular emphasis to collect/standardize and preserve huge datase
ig-data and Machine Learning approaches to fluid mechanics in genera
ce, in particular. It is an easily accessible web platform for high qua
is to host, standardize and manage a large collecti
rimental and numerical data sets from high-end fluid dyn
ies and High Performance Computational centers. Smart-
ble performances when accessing/uploading/searching data. The
munity is asked to contribute, by deploying freely downloadable, accurate an
mented dataset for the sake of "reproducibility": The process of documenting
edures and archiving data so that others can fully reproduce scientific results.
se contact the administrator for infos about how to upload your dataset. We
by deploying a first dataset made of 2d and 3d turbulent configurations under
on TURB-Rot. More will come.
https://smart-turb.roma2.infn.it/
TURB-ROT. A LARGE DATABASE OF 3D AND 2D SNAPSHOTS FROM TURBULENT ROTATING FLOWS
A PREPRINT
L. Biferale
Dept. Physics and INFN
University of Rome Tor Vergata, Italy, and IIC-Paris, France
biferale@roma2.infn.it
F. Bonaccorso
Center for Life Nano Science@La Sapienza
Istituto Italiano di Tecnologia and INFN
University of Rome Tor Vergata, Italy.
fabio.bonaccorso@roma2.infn.it
M. Buzzicotti
Dept. Physics and INFN
University of Rome Tor Vergata, Italy.
michele.buzzicotti@roma2.infn.it
P. Clark Di Leoni
Department of Mechanical Engineering,
Johns Hopkins University, Baltimore, USA.
pato@jhu.edu
Search for datasets
1
Datasets
TURB-Rot
A large database of 3d and 2d snapshots from turbulent rotating
TURB-Rot
2
Organizations
web_admin
web_admin group
1
member