# ON LEARNING QUASI-LAGRANGIAN TURBULENCE

**Artur P. Toshev**[1], **Teodor Kalinov**[1], **Nicholas Gao**[2], **Stephan Günnemann**[2], **Nikolaus A. Adams**[1,3]
[1] Department of Engineering Physics and Computation, TUM, Germany
[2] Department of Computer Science & Munich Data Science Institute, TUM, Germany
[3] Munich Institute of Integrated Materials, Energy and Process Engineering, TUM, Germany
*Email address:* artur.toshev@tum.de

## ABSTRACT

Lagrangian, or particle-based, fluid mechanics methods are the dominant numerical tool for simulating complex boundaries, solid-fluid interactions, and multiphase flows. While their counterpart, the Eulerian framework, has seen significant progress in learning turbulence closures – such as large eddy simulation (LES) modeling – turbulence modeling in the Lagrangian framework has been far less successful. In this paper, we first explain why preserving the correct energy spectrum, crucial for analyzing turbulence, is fundamentally impossible in a fully Lagrangian description. This limitation necessitates using quasi-Lagrangian schemes – methods that adjust the evolution of fluid particle positions beyond their physical velocity to improve accuracy. However, manually designing such corrections is challenging, motivating data-driven approaches. To this end, we are the first to investigate machine-learned quasi-Lagrangian fluid dynamics surrogates. Our experiments are on a new quasi-Lagrangian 2D turbulent Kolmogorov dataset, where velocities from a high-fidelity direct numerical simulation (DNS) solver are spectrally interpolated onto fluid particles, interleaved with particle relaxations to achieve weakly compressible fluid dynamics. We compare six machine-learning parametrizations for evolving the positions and velocities of particles. Our results show that learning simple unconstrained correction terms yields coarse-grained simulations that align well with the reference high-fidelity simulation.

## 1 INTRODUCTION

Differential equations are essential in engineering and physics, governing the spatial and temporal behavior of dynamical systems. Although their theoretical foundations are well established, practical applications often face a major challenge: complexity. In most cases, closed-form analytical solutions are unattainable, making numerical approximations indispensable. Techniques such as finite element and finite volume methods have become standard tools for addressing this challenge. Numerical approaches for solving partial differential equations (PDEs) can broadly be divided into two main categories. Eulerian methods (typically grid- or mesh-based) discretize continuous space using fixed spatial points, while Lagrangian methods – particularly particle-based – rely on discrete material points, which move according to the velocity field. Each of these frameworks possesses distinct properties suited to different types of PDEs or applications. However, purely Lagrangian approaches have some intrinsic limitations, which have led to the development of quasi-Lagrangian methods – through techniques such as particle shifting – where particles move with a slightly different velocity than the physical velocity (Oger et al., 2016).

Turbulent flow simulations present a particularly difficult challenge for Lagrangian methods. A common strategy for modeling turbulence is large eddy simulation (LES), which resolves the large-scale features of turbulent flows while modeling the smaller and unresolved subgrid-scale eddies (Garnier et al., 2009). By resolving only the most energetic structures, LES strikes a balance between accuracy and computational efficiency, making it a powerful tool for studying complex fluid dynamics in engineering and atmospheric sciences. The effectiveness of LES is often evaluated through spectral analysis which examines the distribution of turbulent kinetic energy across different length scales, ensuring proper resolution of large eddies.

Smoothed Particle Hydrodynamics (SPH), introduced independently by Lucy (1977) and Gingold & Monaghan (1977), is among the most popular Lagrangian discretization schemes. SPH is a mesh-free method that uses radial kernel functions to interpolate physical quantities over neighboring particles, similar in concept to stencils in grid-based methods. While SPH incurs a higher computational cost compared to Eulerian methods due to frequent updates of the connectivity graph, its mesh-free nature makes it especially suitable for simulating systems with large deformations. For incompressible fluids, SPH often relies on a weak compressibility assumption, permitting density deviations of up to $\sim 1\%$ (Monaghan, 2005). Originally developed for astrophysics, SPH has since been applied to various applications, including free surface flows (Marrone et al., 2011; Violeau & Rogers, 2016), complex boundaries interactions (Adami et al., 2012b), multi-phase flows (Hu & Adams, 2007), and fluid-structure interactions (Antoci et al., 2007).

Despite its versatility, turbulence modeling within SPH remains challenging. Adami et al. (2012a) identified two primary difficulties: (1) in inviscid flows, SPH's energy-conserving nature prevents energy dissipation, leading to unphysical thermal fluctuations at high wave numbers (Monaghan, 2002); and (2) in viscous flows, the pressure-term discretization in the conservative form violates gauge invariance, introducing numerical dissipation which overly dampens the flow. Interestingly, SPH interpolation shares conceptual similarities with LES filtering – for instance, SPH introduces an implicit viscosity – that could be advantageous for an SPH turbulence model (Ellero et al., 2010; Shi et al., 2012).

Although several classical turbulence modeling approaches have been proposed for SPH, including Reynolds-averages Navier Stokes (RANS) (Violeau & Issa, 2007; Wang et al., 2025) as well as LES-like approximations (Gotoh et al., 2004; Dalrymple & Rogers, 2006; Adami et al., 2012a; Di Mascio et al., 2017; Antuono et al., 2021), they generally lag behind the performance of state-of-the-art grid-based methods. Recently, neural networks have been employed to learn Lagrangian LES models (Tian et al., 2023; Woodward et al., 2023), but these approaches rely on passive tracer particles, which are fundamentally different from SPH particles, as we outline in the next section.

In this work, we introduce a new quasi-Lagrangian framework for learning weakly compressible SPH-LES turbulence. Our key contributions are:

- Rationale for using a quasi-Lagrangian over a Lagrangian approach to SPH-LES modeling;
- Approach to constructing high-quality quasi-Lagrangian datasets, which is used to generate a 2D turbulent Kolmogorov flow dataset by interpolating velocities from a spectral direct numerical simulation (DNS) solver to SPH particles;
- Comparison of six machine-learning parametrizations for evolving the physical and shifting velocities, including purely data-driven as well as hybrid approaches.

## 2 Quasi-Lagrangian Methods

**Shifting schemes in SPH.** Standard SPH (Monaghan, 2005) suffers from tensile instabilities caused by negative pressure, leading to particle clustering or clumping (Price, 2012; Sun et al., 2018; Lyu et al., 2022). We refer to quasi-Lagrangian extensions of SPH that explicitly target improving the particle distribution as shifting schemes, to which we count Adami et al. (2013); Zhang et al. (2017a); Sun et al. (2019); Toshev et al. (2024b), while other schemes implicitly also improve the particle distribution even if that is not how they were originally motivated (Marrone et al., 2011; Zhang et al., 2017b). A more generic framework than the quasi-Lagrangian description is given by the arbitrary Lagrangian-Eulerian (ALE) methods (Hirt et al., 1974; Oger et al., 2016; Jacob et al., 2021; Rastelli et al., 2023), which treat the evolution of the physical velocity field $\mathbf{u}$ and the particle shifting velocity $\mathbf{v}$ separately. This body of literature hints at the inherent limitations of fully Lagrangian methods and that we can deliberately design more general transport schemes. One can compare the two velocities $\mathbf{u}$ and $\mathbf{v}$ with the Reynolds and Favre averaging in compressible turbulence (Gatski & Bonnet, 2013), where $\mathbf{u}$ and Reynolds averaging relate to the filtered Eulerian velocity field, while $\mathbf{v}$ and Favre averaging both depend on the local density.

**Passive tracers versus SPH particles.** While ALE methods blend the transition between Lagrangian and Eulerian viewpoints, it is worth noting that the solutions derived from each framework are qualitatively different. To illustrate this, we use the two-dimensional Taylor-Green vortex

(TGV) (Taylor & Green, 1937), initialized with the following velocity field:

$$\mathbf{u}(x, y) = \begin{bmatrix} -\cos(2\pi x)\sin(2\pi y) \\ \sin(2\pi x)\cos(2\pi y) \end{bmatrix}. \tag{1}$$

In the absence of viscosity and perturbations, this flow remains steady in the Eulerian frame. However, a purely Lagrangian simulation initialized on a Cartesian grid leads to undesired coherent structures and significantly overestimated dissipation (Adami et al., 2013). This discrepancy highlights a key challenge: although TGV is a benchmark case for SPH model validation, standard SPH struggles with it, while quasi-Lagrangian methods such as Adami et al. (2013) show significant improvement.

If we simply advect particles using the analytically defined, incompressible Eulerian velocity field, we obtain very different behavior than what we would expect from a weakly-compressible field, see Figure 5. Particles advected by the instantaneous point-wise velocity field are called *tracer particles*. They are fundamentally different from SPH particles as tracer particles do not interact with each other and lack a mass or volume. In Figure 5, we see that if we treat tracer particles as SPH particles for the purpose of computing their density, i.e., summation of mass over neighborhoods, we violate the 1% weak-compressibility density condition by more than one order of magnitude. Notably, while recent works (Tian et al., 2023; Woodward et al., 2023) have proposed learning Lagrangian LES models using tracer particles, our work is the first to address data-driven weakly-compressible SPH turbulence modeling.

**Our shifting scheme.** We propose a novel shifting strategy for generating high-quality weakly-compressible SPH reference trajectories, ingraining the 1% compressibility threshold (Monaghan, 2005). Our procedure consists of first initializing (a) a high-fidelity DNS solver and (b) an initial relaxed particle distribution, and then alternating between the following two steps (see Figure 1):

1. **Spectral evolution and interpolation:** We evolve the DNS solution for one time step by solving the Navier Stokes equations in the Eulerian description (upper half of Figure 1). High-frequency components of the velocity field are then removed via spectral filtering (zeroing large wavenumbers). We then interpolate $\mathbf{u}$ at the particle positions by inverse transforming the velocities from spectral space using the discrete Fourier Transform (DFT) ("Interpolate" in Figure 1). Note that we cannot use the FFT, which requires regular grids in both the real and complex domains. Particles are then advected as passive tracers using the interpolated velocity: $\mathbf{x}'_{t+1} = \mathbf{x}_t + \Delta t \cdot \mathbf{u}_t$ in Figure 1.
2. **Relaxation step:** To satisfy the weak-compressibility constraint, we relax the particle positions through two SPH simulation steps. Particles are initialized with zero velocity and no viscosity, thus, the relaxation is driven by pressure gradients. We also incorporate a transport velocity formulation (TVF) (Adami et al., 2013) to improve particle distribution. For an analysis of SPH relaxations, we refer to Litvinov et al. (2015); Toshev et al. (2024b); Fan et al. (2024).

To generate high-fidelity reference data, we use a spectral DNS solver based on Mortensen & Langtangen (2016) (see Canuto et al. (1988)). We simulate DNS dynamics on a $256^2$ grid and spectrally filter to $64^2$. In this way, we reduce the number of spatial discretization points by $16\times$ and could also use $4\times$ larger time steps to maintain the same CFL number (Courant et al., 1928).

**Evaluating the energy spectrum.** To evaluate the energy spectrum, we must compute the Fourier transform of the velocity field. While this is straightforward on regular grids, additional care is needed when working with irregular discretization. Building on results by Shi et al. (2013), who extensively compared algorithms for evaluating the energy spectrum in weakly-compressible SPH, we employ a mean least squares (MLS) quadratic interpolation to project particle velocities onto a uniform Cartesian grid. For comparison, Figure 4 illustrates the difference between DFT interpolation and our MLS-based method.

## 3 Learning quasi-Lagrangian updates

Our datasets consist of particle evolution trajectories with $N$ steps $\{(\mathbf{x}_i, \mathbf{u}_i)_{i=1:N}\}$, for which we define the shifting velocity as the difference in positions $\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$. In this setup, the physical

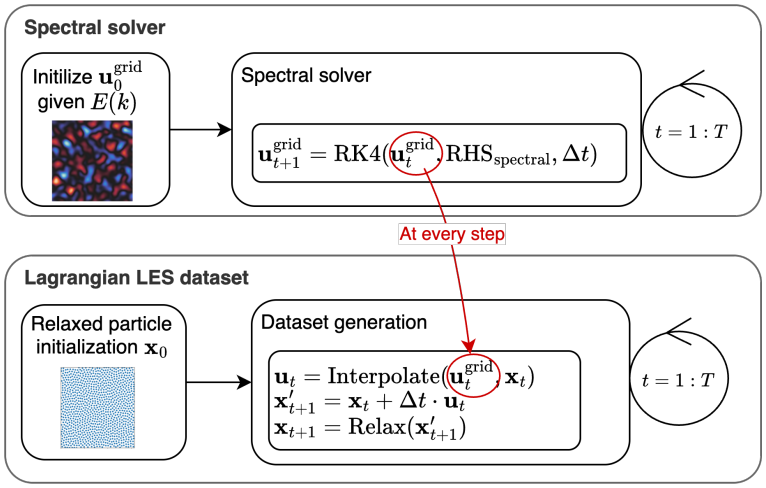



Figure 1: Dataset generation procedure.

velocity $\mathbf{u}$ is used to evaluate the energy spectrum, and the shifting velocity $\mathbf{v}$ evolves the positions of particles. We propose to use a graph neural network (GNN) to jointly learn the physical velocity $\mathbf{u}$ and shifting velocity $\mathbf{v}$, which is further simplified to learning accelerations $\mathbf{a}^u$ and $\mathbf{a}^v$, respectively. However, there are various ways of defining these accelerations, from which we focus on six specific parametrizations described below and illustrated in Figure 2:

$$\mathbf{a}_0^u, \mathbf{a}_0^v = GNN(\mathbf{u}_0, \mathbf{v}_0) \tag{2}$$

1. *Simple-Base:* This formulation is the simplest one conceptually, as it evolves the physical and shifting velocities completely independently, i.e., $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{a}_0^u,\ \mathbf{v}_1 = \mathbf{v}_0 + \mathbf{a}_0^v$.

2. *TVF-Base:* Following the transport velocity formulation (TVF) (Adami et al., 2013), we define the shifting velocity $\mathbf{v}_1 = \mathbf{u}_1 + \mathbf{a}_0^v$ as a modified version of the physical velocity $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{a}_0^u$. This introduces a direct coupling between the two velocity fields, which is not present in the Simple-Base variant.

3. *Simple-*$\mathbf{u}$*:* This formulation is conceptually similar to TVF-Base as it also defines a coupling, but here, the two learning targets depend on each other only during rollout, i.e., $\mathbf{v}_1 = \mathbf{u}_0 + \mathbf{a}_0^v,\ \mathbf{u}_1 = \mathbf{u}_0 + \mathbf{a}_0^u$.

4. *Simple-*$\mathbf{u}$*-Closure:* Adding terms to the right-hand side of the Navier-Stokes equations to compensate for unresolved scales is commonly known as closure modeling. We adopt such an approach by learning the part of the physical acceleration $\mathbf{a}^u$ not captured by the regular SPH acceleration approximation $\mathbf{a}^{\text{SPH},u}$, i.e., $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{a}_0^{\text{SPH},u} + \mathbf{a}_0^u$. For the shifting velocity, we stick to the Simple-Base design $\mathbf{v}_1 = \mathbf{v}_0 + \mathbf{a}_0^v$. This approach closely resembles hybrid machine learning methods (Toshev et al., 2023) like Solver-in-the-Loop (Um et al., 2020; Toshev et al., 2024c) and JAX-CFD (Kochkov et al., 2021).

5. *TVF-Rlx:* During dataset generation, we use SPH to relax the particle distribution. Here, we explore what would happen if we leverage this exact same relaxation $\mathbf{a}^{\text{Relax}}$ to have to learn only the evolution of the physical velocity field $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{a}_0^u$. If $\mathbf{a}_0^u$ is perfectly learned, the relaxation acceleration $\mathbf{a}_0^{\text{Relax}}$ is guaranteed to be exact.

6. *TVF-Rlx-Closure:* This setting extends the previous one by learning a part of the shifting acceleration, combined with a regular particle relaxation term $\mathbf{a}^{\text{Relax}}$, i.e., $\mathbf{v}_1 = \mathbf{u}_1 + \mathbf{a}_0^{\text{Relax}} + \mathbf{a}_0^v$. Note that $\mathbf{a}_0^v = 0$ if we use the exact same step size as for dataset generation, but $\mathbf{a}_0^v \neq 0$ if we do temporal coarsening, i.e., learning to simulate with larger time steps.

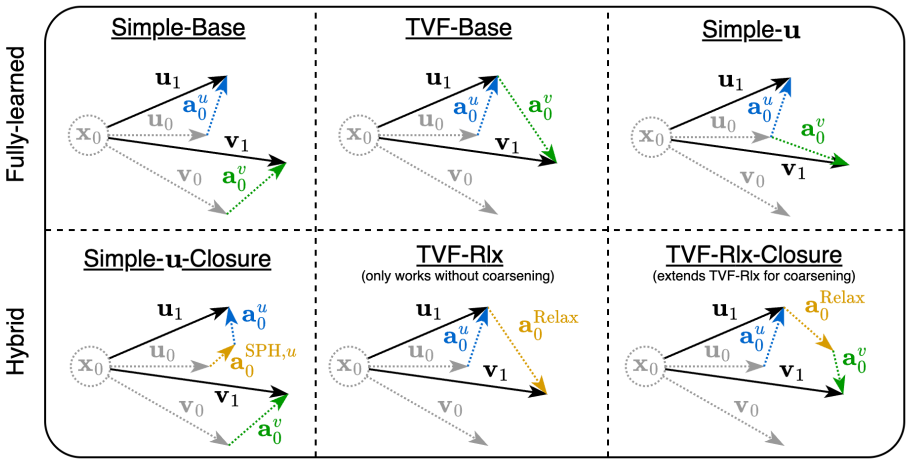



Figure 2: Learning problem variants.

## 4 EXPERIMENTS

Our experiments are based on the GNS architecture (Sanchez-Gonzalez et al., 2020) and the LagrangeBench benchmarking setup (Toshev et al., 2024a). The architecture is a model with 10 message-passing steps, a latent size of 128, 5 historic velocities $\mathbf{v}$, and one hidden MLP layer resulting in 1.2M parameters. Optimization is conducted for 500k steps with a batch size of one using Adam (Kingma, 2014) with weight decay and exponentially decaying learning rate from 1e-3 to 1e-5, similar to Sanchez-Gonzalez et al. (2020).

**Datasets.** We use the 2D decaying turbulent Kolmogorov flow with Reynolds number 1000 described in Kochkov et al. (2021). Using the codebase from this previous work, we generate 20 initial conditions (10/5/5 for training/validation/testing) and simulate each for $t_{\text{final}} = 10$ with 20k integration steps as described in Section 2. The resolution of the spectral method is $256^2$, and the number of particles is $64^2$. This full dataset is denoted as *Kolm2D-1*, where "1" refers to every *one* time step. We also have a dataset consisting of every 10th step denoted *Kolm2D-10*.

**Metrics.** We use the average joint velocity MSE over 20 rollout steps denoted $\text{MSE}_{20}^{\mathbf{u},\mathbf{v}}$, i.e. averaged over both $\mathbf{u}$ and $\mathbf{v}$, and kinetic energy $\text{MSE}_{20}^{Ekin}$ as defined in Toshev et al. (2024a). In addition, we evaluate the energy spectrum, as described in Section 2.

**Comparing parametrizations.** Our experiments are summarized in Table 1, while various visualizations can be found in Appendix C. Overall, we find that the Simple-Base variant outperforms the others on both short and long rollouts, i.e., 20 steps loss (see Table 1) and 1000-step rollouts (see Appendix C), respectively. From the fully learned top half of Figure 2, we surprisingly find that, while classical quasi-Lagrangian modeling couples the physical velocity $\mathbf{u}$ and shifting velocity $\mathbf{v}$ similarly to our TVF-Base and Simple-$\mathbf{u}$ approaches, our best results are achieved when we learn the two separately, i.e., Simple-Base. Even more surprisingly, while hybrid approaches like our Simple-$\mathbf{u}$-Closure (lower half of Figure 2) have shown good results in the Eulerian description (Um et al., 2020; Kochkov et al., 2021), the 20-step velocity loss of our closure models, i.e., Simple-$\mathbf{u}$-Closure and TVF-Rlx-Closure, was more than two orders of magnitude worse than the fully-learned alternatives and we omit reporting it in the table. Lastly, the variant TVF-Rlx performed competitively or slightly worse than the fully learned ones.

**Comparison with SPH.** As discussed in the introduction, standard SPH is not well-suited for simulating turbulent flows. As a baseline for comparison, we use the transport velocity formulation of SPH (Adami et al., 2013), denoted as "SPH" in Figure 3, which has shown significant improvements for simulating turbulent dynamics (Adami et al., 2012a). However, SPH is still too dissipative (see drop of kinetic energy of left side of Figure 3) and also cannot reproduce the reference spectrum by

Table 1: Performance comparison.

| Variant | Kolm2D-1 | | Kolm2D-10 | |
|---|---|---|---|---|
| | $\text{MSE}_{20}^{\mathbf{u},\mathbf{v}}$ | $\text{MSE}_{20}^{Ekin}$ | $\text{MSE}_{20}^{\mathbf{u},\mathbf{v}}$ | $\text{MSE}_{20}^{Ekin}$ |
| *Simple-Base* | 1.2e-4 | 1.3e-1 | 2.1e-3 | 0.95 |
| *TVF-Base* | 1.4e-4 | 3.4e-1 | 2.4e-3 | 2.6 |
| *Simple-*$\mathbf{u}$ | 1.3e-4 | 2.8e-1 | 2.5e-3 | 2.6 |
| *TVF-Rlx* | 1.6e-4 | 3.7e-1 | 4.1e-3 | 3.3 |

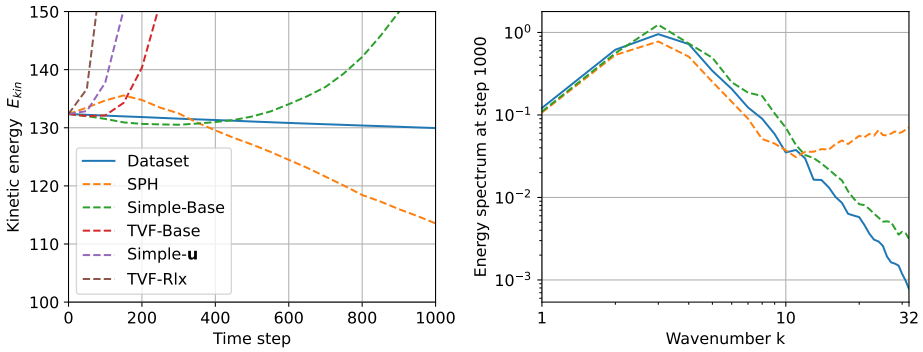



Figure 3: Kinetic energy evolution (left) and energy spectra at step 1000 (right) on Kolm2D-1. Plots are based on the first test trajectory. Some variants are missing in the right plot because MLS interpolation fails on domains with void regions. "Dataset" denotes the ground truth data, and "SPH" denotes the baseline classical numerical method. While SPH over-dissipates kinetic energy, in particular, through unwanted high wave number dynamics, the learned methods increase kinetic energy,i.e., become unstable, while better capturing the spectrum.

introducing high-frequency oscillations (contains too much energy at high wave numbers $k$ on the right side of Figure 3). According to Figure 3, the Simple-Base parametrization is the only one that does not blow up and also matches the energy spectrum better than SPH.

**NeuralSPH.** In a final ablation, we add the NeuralSPH approach of Toshev et al. (2024b) to our inference routine. The idea of this approach is to relax the particle configuration only during inference, promising improved particle distribution and more physical rollouts. Our results with 10 NeuralSPH relaxation steps are visualized in Appendix C, where we do see a significant improvement for the Simple-Base and TVF-Base approaches. In the case of the Simple-Base variant, this improvement in dynamics $\mathbf{v}$ interestingly also improves the velocity field $\mathbf{u}$, showing that our quasi-Lagrangian setting benefits from NeuralSPH, although NeuralSPH has been developed for fully Lagrangian data.

## 5 Conclusion and Future Work

In this work, we introduced a learned quasi-Lagrangian framework for turbulence modeling in particle-based fluid simulations, addressing the fundamental limitations of fully Lagrangian approaches. By constructing a high-fidelity quasi-Lagrangian dataset and evaluating multiple update schemes, we demonstrated that learned shifting velocities can improve energy spectra while maintaining weak compressibility constraints. Our experiments suggest that decoupling the evolution of physical and shifting velocities yields the most promising results, surpassing traditional SPH-based approaches. Future work includes testing this framework on three-dimensional turbulence, exploring alternative data-driven closure models for further accuracy improvements, and further improving the stability in long-term rollouts. Additionally, investigating how our approach generalizes to other flow fields and different Reynolds numbers remains an important avenue for research.

## ACKNOWLEDGEMENTS

This work has been funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

## REFERENCES

Stefan Adami, Xiangyu Hu, and Nikolaus Adams. Simulating three-dimensional turbulence with sph. Technical report, Lehrstuhl für Aerodynamik, 2012a.

Stefan Adami, Xiangyu Hu, and Nikolaus A Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012b.

Stefan Adami, XY Hu, and Nikolaus A Adams. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241:292–307, 2013.

Carla Antoci, Mario Gallati, and Stefano Sibilla. Numerical simulation of fluid–structure interaction by sph. *Computers & structures*, 85(11-14):879–890, 2007.

M Antuono, S Marrone, A Di Mascio, and A Colagrossi. Smoothed particle hydrodynamics method from a large eddy simulation perspective. generalization to a quasi-lagrangian model. *Physics of Fluids*, 33(1), 2021.

Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. *Spectral Methods in Fluid Dynamics*. Springer, Berlin, Heidelberg, 1988.

Richard Courant, Kurt Friedrichs, and Hans Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen*, 100(1):32–74, 1928.

Robert A Dalrymple and Benedict D Rogers. Numerical modeling of water waves with the sph method. *Coastal engineering*, 53(2-3):141–147, 2006.

A Di Mascio, M Antuono, A Colagrossi, and S Marrone. Smoothed particle hydrodynamics method from a large eddy simulation perspective. *Physics of Fluids*, 29(3), 2017.

Marco Ellero, Pep Español, and Nikolaus A Adams. Implicit atomistic viscosities in smoothed particle hydrodynamics. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82 (4):046702, 2010.

Yu Fan, Xiaoliang Li, Shuoguo Zhang, Xiangyu Hu, and Nikolaus A Adams. Analysis of the particle relaxation method for generating uniform particle distributions in smoothed particle hydrodynamics. *arXiv preprint arXiv:2403.00623*, 2024.

E. Garnier, N. Adams, and P. Sagaut. *Large Eddy Simulation for Compressible Flows*. Scientific Computation. Springer Netherlands, Dordrecht, 2009. ISBN 978-90-481-2818-1 978-90-481-2819-8.

Thomas B Gatski and Jean-Paul Bonnet. *Compressibility, turbulence and high speed flow*. Academic Press, 2013.

Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

Hitoshi Gotoh, Songdong Shao, and Tetsu Memita. Sph-les model for numerical investigation of wave interaction with partially immersed breakwater. *Coastal Engineering Journal*, 46(1):39–63, 2004.

Cyrill W Hirt, Anthony A Amsden, and JL Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of computational physics*, 14(3):227–253, 1974.

XY Hu and Nikolaus A Adams. An incompressible multi-phase sph method. *Journal of computational physics*, 227(1):264–278, 2007.

Bruno Jacob, Brian Drawert, Tau-Mu Yi, and Linda Petzold. An arbitrary lagrangian eulerian smoothed particle hydrodynamics (ale-sph) method with a boundary volume fraction formulation for fluid-structure interaction. *Engineering analysis with boundary elements*, 128:274–289, 2021.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

Sergey Litvinov, XY Hu, and Nikolaus A Adams. Towards consistence and convergence of conservative sph approximations. *Journal of Computational Physics*, 301:394–401, 2015.

Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal, vol. 82, Dec. 1977, p. 1013-1024.*, 82:1013–1024, 1977.

Hong-Guan Lyu, Pengnan Sun, Andrea Colagrossi, and A-Man Zhang. Towards sph simulations of cavitating flows with an eosb cavitation model. *Acta Mechanica Sinica*, 39, 07 2022. doi: 10.1007/s10409-022-22158-x.

Salvatore Marrone, Matteo Antuono, A Colagrossi, G Colicchio, D Le Touzé, and G Graziani. $\delta$-sph model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1526–1542, 2011.

Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

Joseph J Monaghan. Sph compressible turbulence. *Monthly Notices of the Royal Astronomical Society*, 335(3):843–852, 2002.

Mikael Mortensen and Hans Petter Langtangen. High performance python for direct numerical simulations of turbulent flows. *Computer Physics Communications*, 203:53–65, 2016.

Guillaume Oger, Salvatore Marrone, David Le Touzé, and Matthieu De Leffe. Sph accuracy improvement through the combination of a quasi-lagrangian shifting transport velocity and consistent ale formalisms. *Journal of Computational Physics*, 313:76–98, 2016.

Daniel J Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.

P Rastelli, R Vacondio, and JC Marongiu. An arbitrarily lagrangian–eulerian sph scheme with implicit iterative particle shifting procedure. *Computer Methods in Applied Mechanics and Engineering*, 414:116159, 2023.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Yilei Shi, Marco Ellero, and Nikolaus A Adams. Analysis of intermittency in under-resolved smoothed-particle-hydrodynamics direct numerical simulations of forced compressible turbulence. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 85(3):036708, 2012.

Yilei Shi, Xiao Xiang Zhu, Marco Ellero, and Nikolaus A Adams. Analysis of interpolation schemes for the accurate estimation of energy spectrum in lagrangian methods. *Computers & Fluids*, 82: 122–131, 2013.

PN Sun, Andrea Colagrossi, Salvatore Marrone, Matteo Antuono, and AM Zhang. Multi-resolution delta-plus-sph with tensile instability control: Towards high reynolds number flows. *Computer Physics Communications*, 224:63–80, 2018.

PN Sun, A Colagrossi, S Marrone, M Antuono, and A-M Zhang. A consistent approach to particle shifting in the $\delta$-plus-sph model. *Computer Methods in Applied Mechanics and Engineering*, 348: 912–934, 2019.

Geoffrey Ingram Taylor and Albert Edward Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 158(895):499–521, 1937.

Yifeng Tian, Michael Woodward, Mikhail Stepanov, Chris Fryer, Criston Hyett, Daniel Livescu, and Michael Chertkov. Lagrangian large eddy simulations via physics-informed machine learning. *Proceedings of the National Academy of Sciences*, 120(34):e2213638120, 2023.

Artur Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. *Advances in Neural Information Processing Systems*, 36, 2024a.

Artur P Toshev, Ludger Paehler, Andrea Panizza, and Nikolaus A Adams. On the relationships between graph neural networks for the simulation of physical systems and classical numerical methods. *arXiv preprint arXiv:2304.00146*, 2023.

Artur P Toshev, Jonas A Erbesdobler, Nikolaus A Adams, and Johannes Brandstetter. Neural sph: Improved neural modeling of lagrangian fluid dynamics. *arXiv preprint arXiv:2402.06275*, 2024b.

Artur P Toshev, Harish Ramachandran, Jonas A Erbesdobler, Gianluca Galletti, Johannes Brandstetter, and Nikolaus A Adams. Jax-sph: A differentiable smoothed particle hydrodynamics framework. *arXiv preprint arXiv:2403.04750*, 2024c.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 33:6111–6122, 2020.

Damien Violeau and Reza Issa. Numerical modelling of complex turbulent free-surface flows with the sph method: an overview. *International Journal for Numerical Methods in Fluids*, 53(2): 277–304, 2007.

Damien Violeau and Benedict D Rogers. Smoothed particle hydrodynamics (sph) for free-surface flows: past, present and future. *Journal of Hydraulic Research*, 54(1):1–26, 2016.

Feng Wang, Zhongguo Sun, and Xiangyu Hu. A weakly compressible sph method for rans simulation of wall-bounded turbulent flows. *arXiv preprint arXiv:2501.18397*, 2025.

Michael Woodward, Yifeng Tian, Criston Hyett, Chris Fryer, Mikhail Stepanov, Daniel Livescu, and Michael Chertkov. Physics-informed machine learning with smoothed particle hydrodynamics: Hierarchy of reduced lagrangian models of turbulence. *Physical Review Fluids*, 8(5):054602, 2023.

Chi Zhang, Xiangyu Y Hu, and Nikolaus A Adams. A generalized transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 337:216–232, 2017a.

Chi Zhang, XY Hu, and Nikolaus A Adams. A weakly compressible sph method based on a low-dissipation riemann solver. *Journal of Computational Physics*, 335:605–620, 2017b.

## A EVALUATING THE SPECTRUM

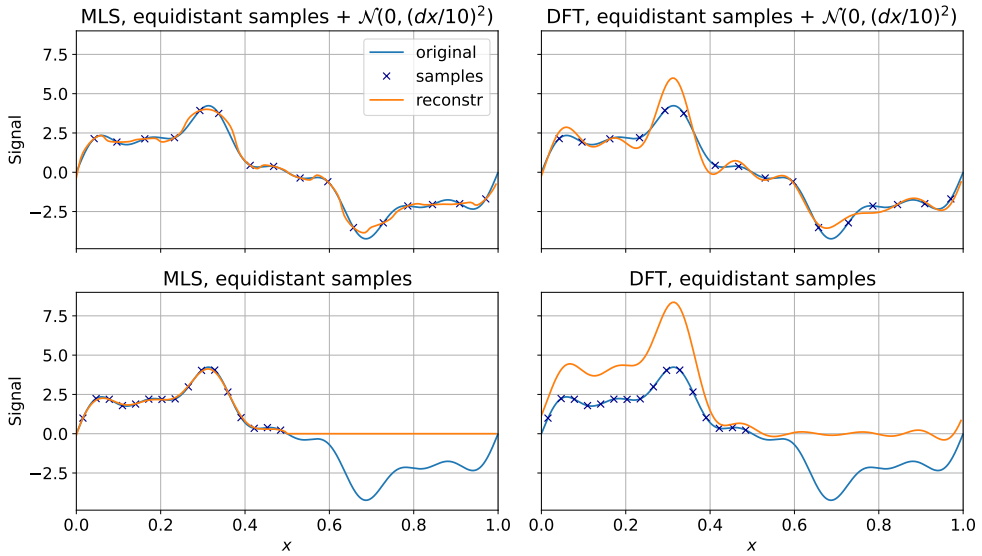



Figure 4: Comparison of quadratic mean least squares (MLS) interpolation (left) and discrete Fourier transform (DFT) interpolation (right) on the example of a scalar signal sampled equidistantly corrupted with additive noise (top) or equidistantly, but not covering the full domain (bottom). The upper half is representative of Lagrangian particle dynamics and shows the sensitivity of DFT to the nonregularity of the samples, while the lower half is an edge case for qualitative comparison when a region is sparsely populated.

## B TRACER PARTICLES

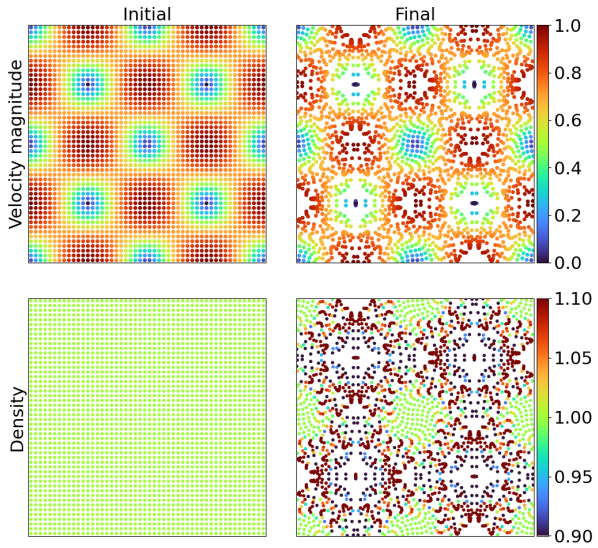



Figure 5: Taylor-Green case simulated for 1s in a $[0, 1]^2$ domain with $50^2$ passive tracer particles. Particles are initially on a Cartesian grid and are then advected along the analytical solution. At the end of the simulation, particles are distributed unequally over the domain, violating weak compressibility if we treat them as SPH particle to estimate their density.

# C ADDITIONAL VISUALIZATIONS

## C.1 KOLM2D-1

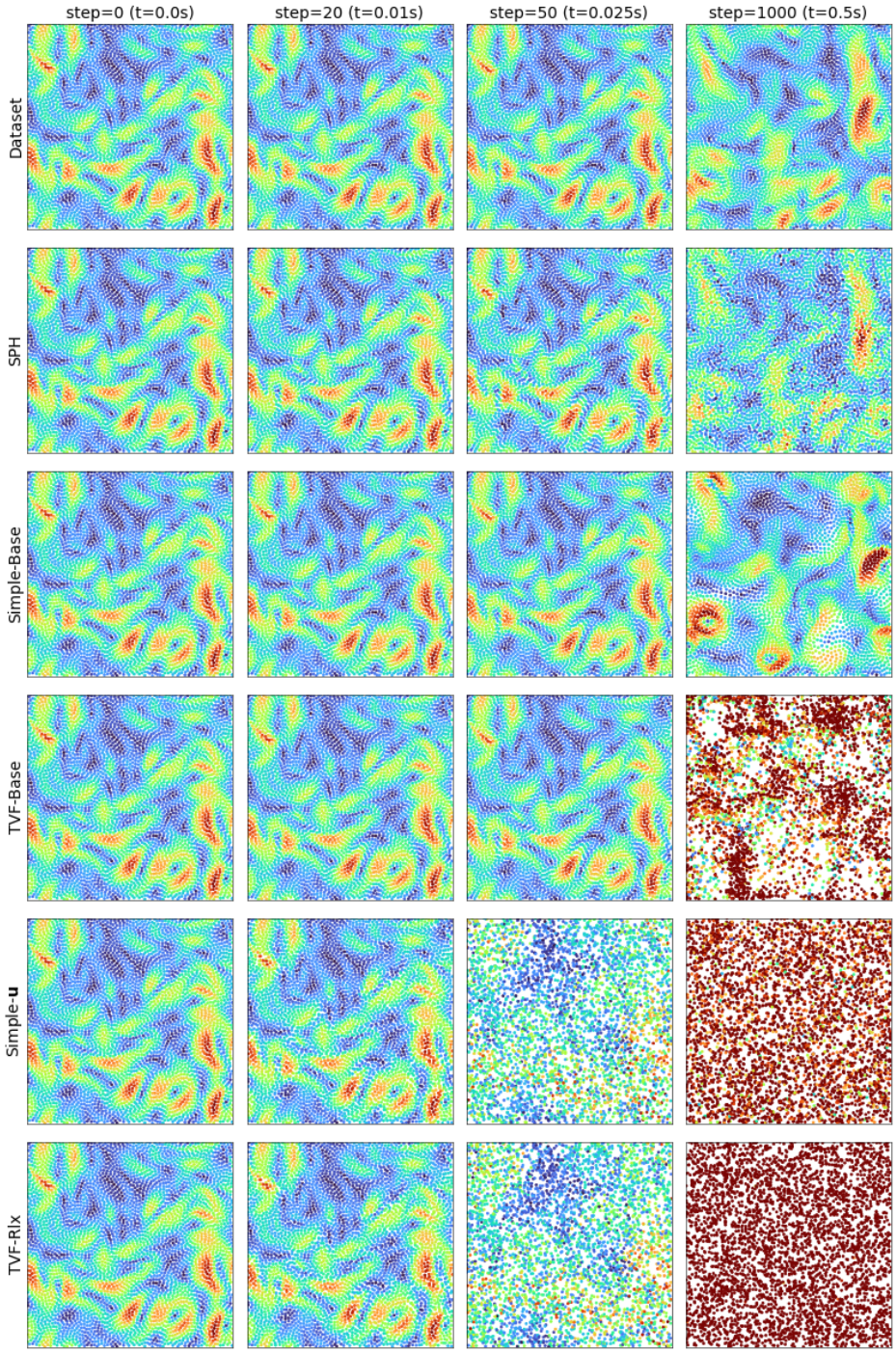



Figure 6: Velocity field of first test trajectory from Kolm2D-1.

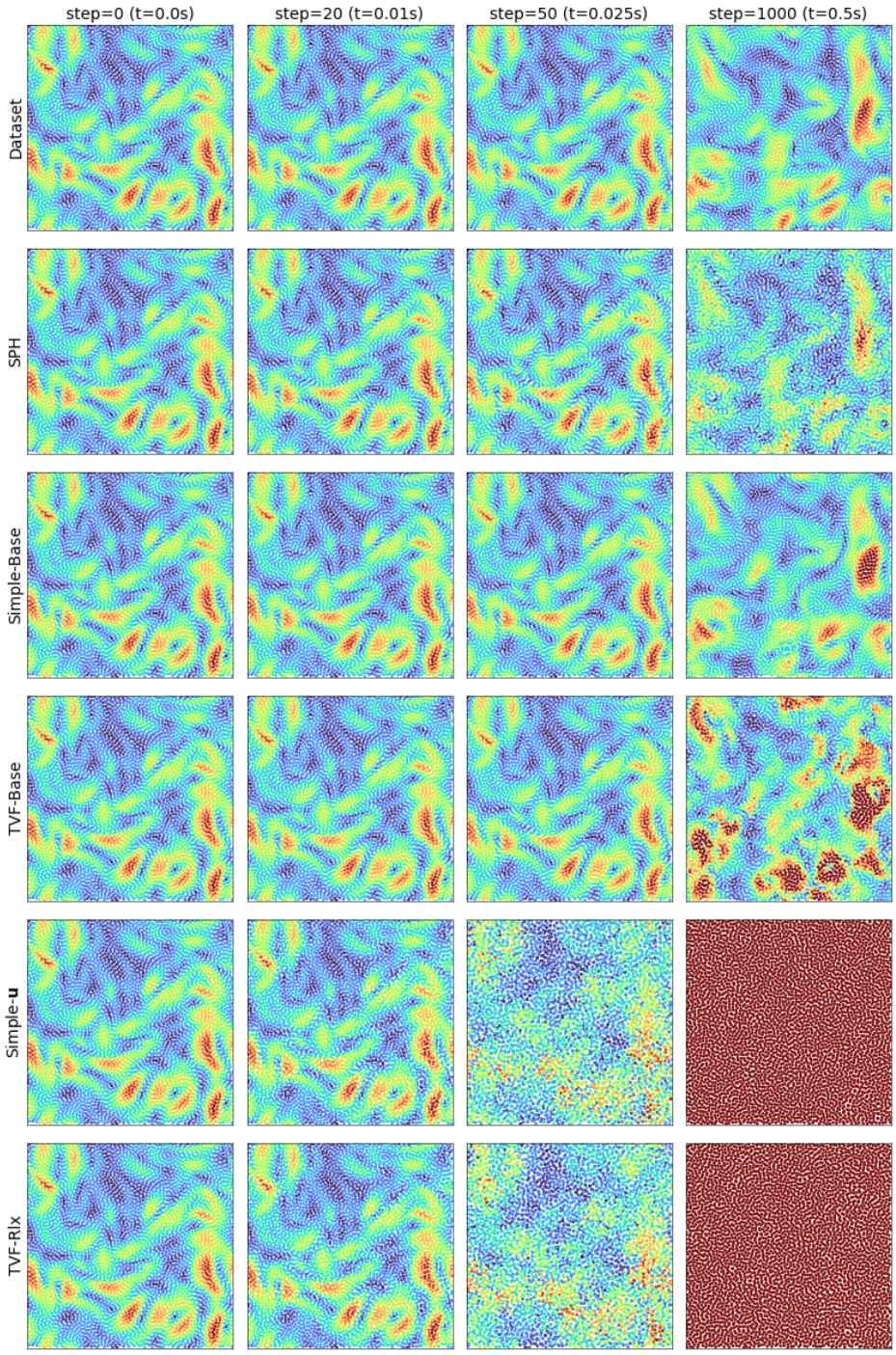
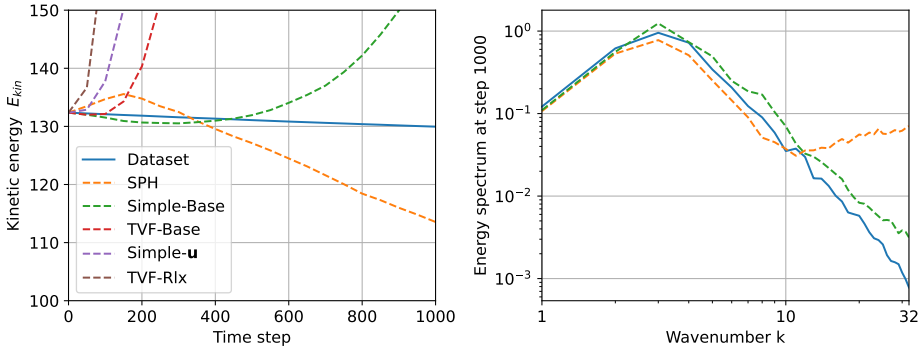
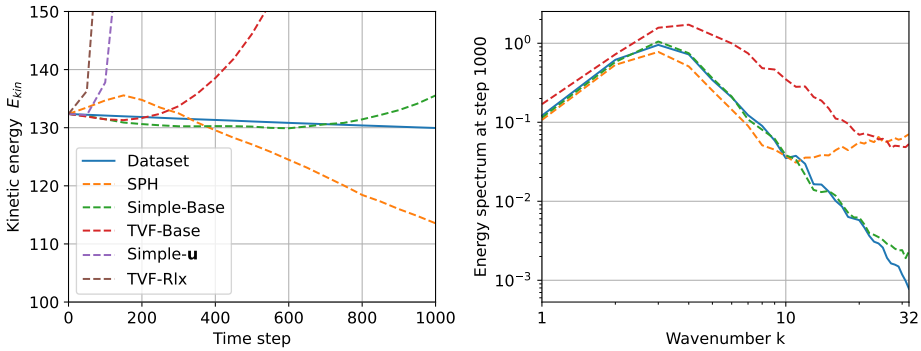



Figure 7: Velocity field of first test trajectory from Kolm2D-1 with NeuralSPH




(a) Without NeuralSPH




(b) With NeuralSPH

Figure 8: Kinetic energy and energy spectra on Kolm2D-1.

### C.2 KOLM2D-10

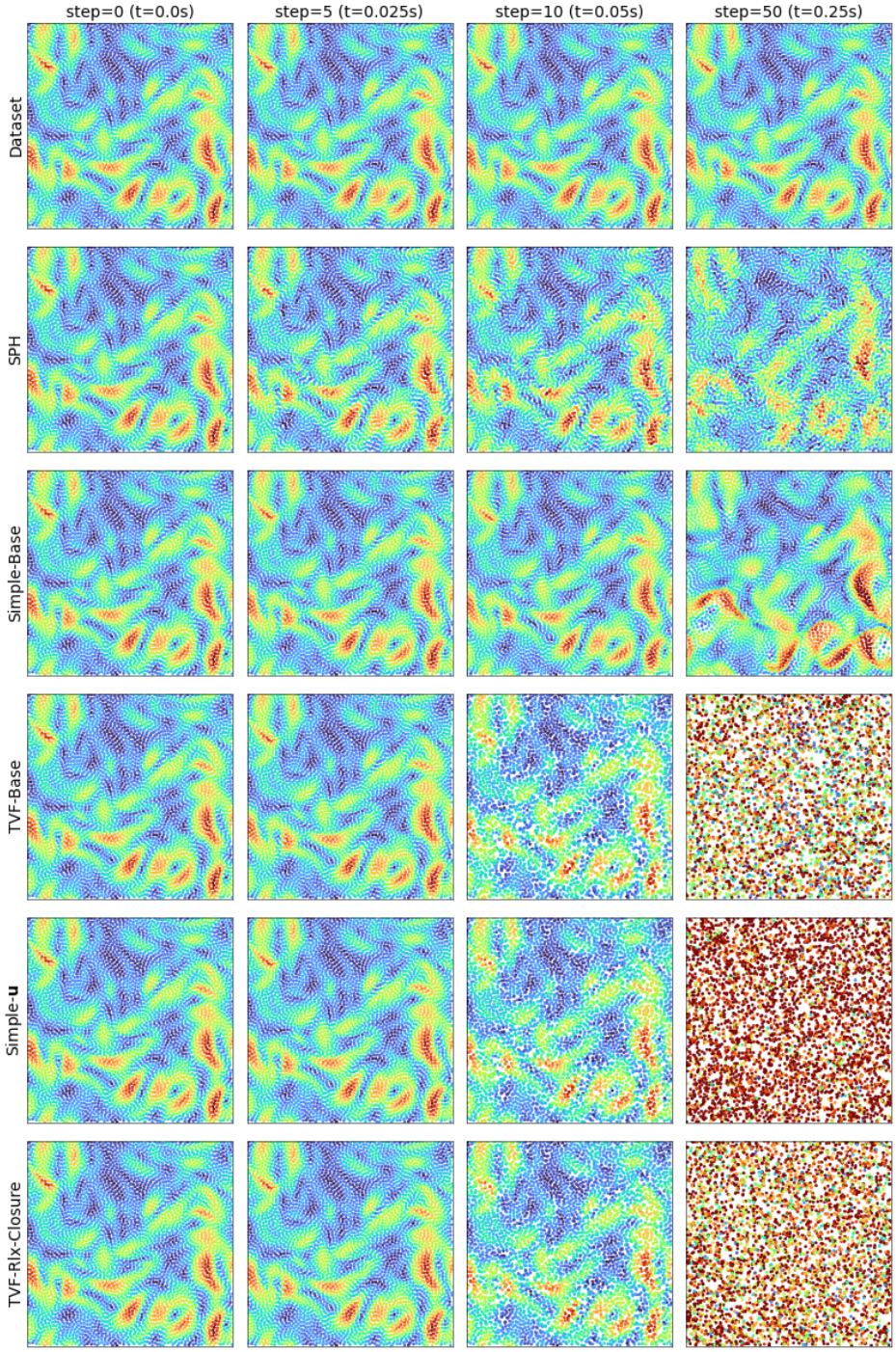



Figure 9: Velocity field of first test trajectory from Kolm2D-10.

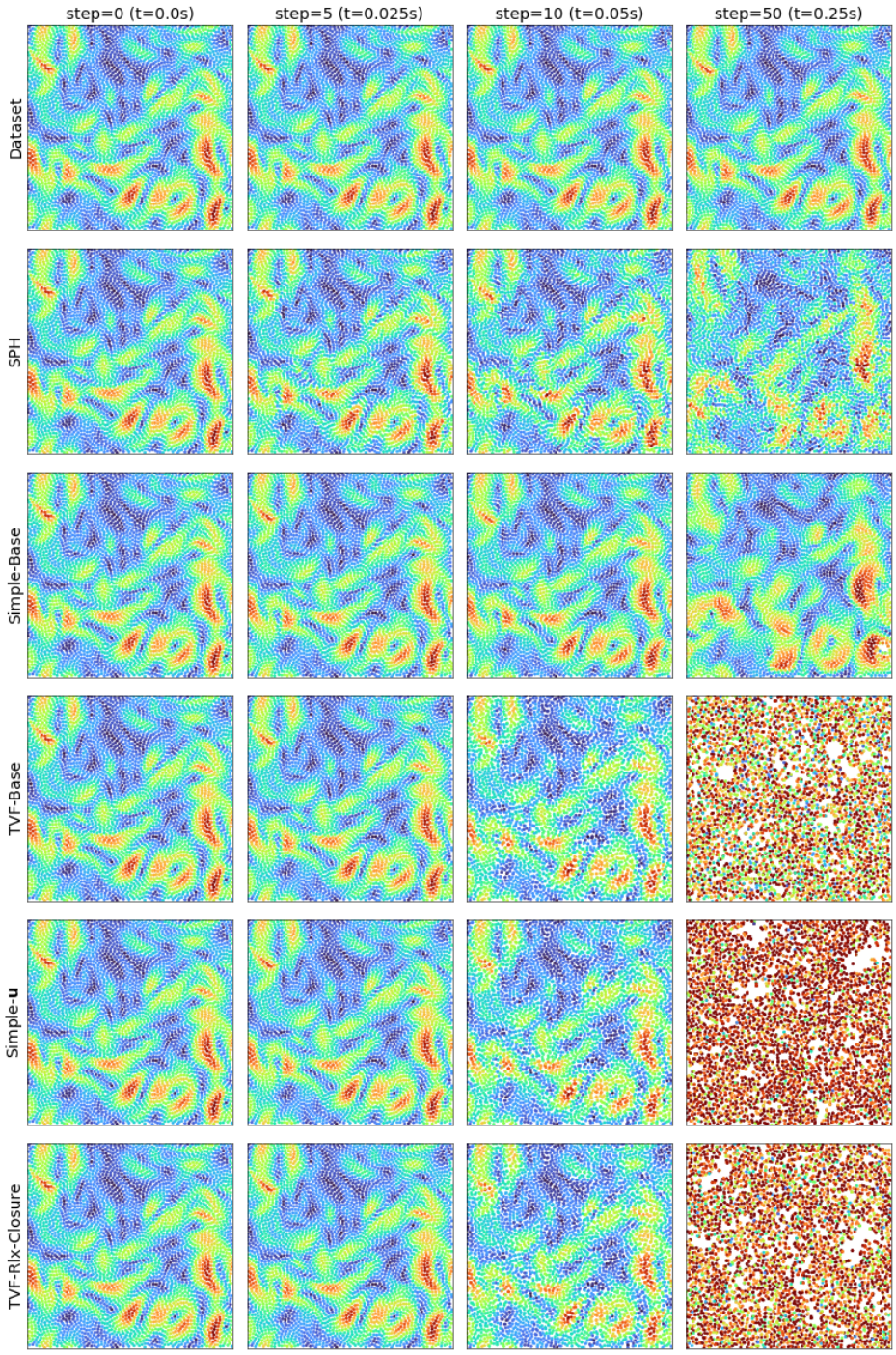
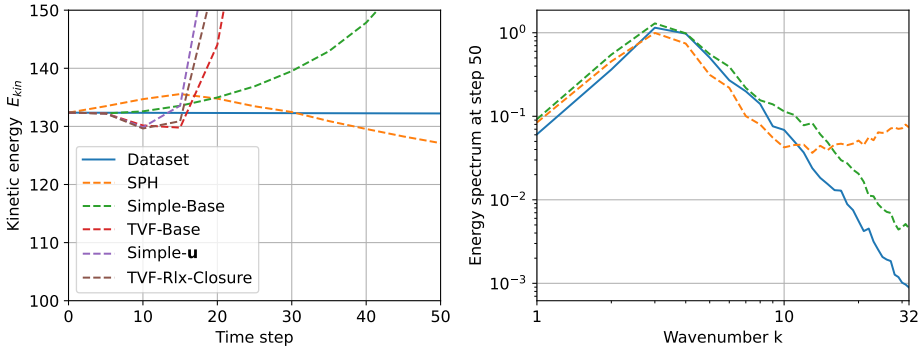
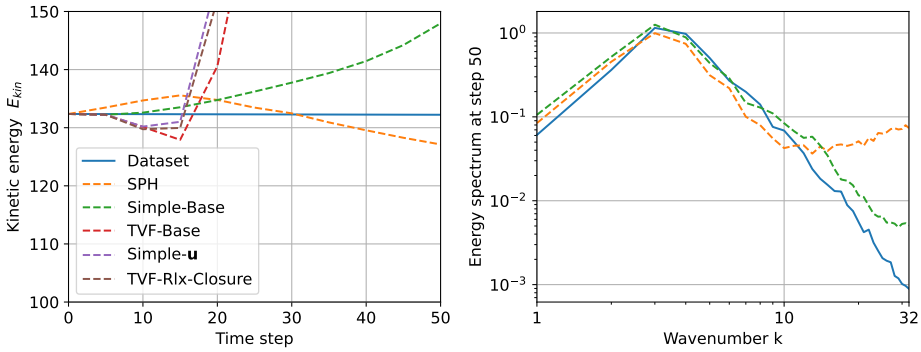



Figure 10: Velocity field of first test trajectory from Kolm2D-10 with NeuralSPH




(a) Without NeuralSPH




(b) With NeuralSPH

Figure 11: Kinetic energy and energy spectra on Kolm2D-10.