

Compression, simulation, and synthesis of turbulent flows with tensor trains

Stefano Pisoni^{1,2,*}, Raghavendra D. Peddinti¹, Siddhartha E. Guzman¹, Egor Tiunov¹, and Leandro Aolita¹

¹Quantum Research Center, *Technology Innovation Institute*, Abu Dhabi, United Arab Emirates

²Institute for Quantum Inspired and Quantum Optimization, *Hamburg University of Technology*, Hamburg, Germany



(Received 10 July 2025; accepted 11 May 2026; published 4 June 2026)

Numerical simulations of turbulent fluids are paramount to real-life applications, from predicting and modeling flows to diagnostic purposes in engineering. However, they are also computationally challenging due to their intrinsically nonlinear dynamics, which require a very high spatial resolution to accurately describe them. A promising idea is to represent flows on a discrete mesh using tensor trains (TTs), featuring a convenient scaling of the number of parameters with the mesh size. However, it is unclear how the compression power of TTs is affected by the complexity of the flows, as measured by the Reynolds number. In fact, no comprehensive analysis of how the TT representation affects the turbulent properties has yet been carried out. We fill this gap by analyzing TTs as an ansatz to compress, simulate, and generate three-dimensional (3D) snapshots with turbulentlike features. Specifically, we first investigate the effect of TT compression on key turbulence signatures, such as the energy spectrum, the probability distribution function of velocity increments, and flatness. Second, we extend the two-dimensional TT solver introduced by Peddinti *et al.* [Commun. Phys. 7, 135 (2024)] to a 3D cubic domain with periodic boundary conditions. We use it to simulate the incompressible Navier-Stokes dynamics at $Re_\lambda = 315$ for a total of 9–10 Kolmogorov turnover times, showcasing the numerical stability of the TT solver in fully developed turbulent regimes. Third, we develop a TT algorithm to synthesize artificial snapshots that exhibit turbulentlike features, with a logarithmic cost in the mesh size. Our analysis demonstrates the ability of the TT representation to capture the characteristic features of turbulence. This offers a powerful quantum-inspired toolkit for the computational treatment of turbulent flows.

DOI: [10.1103/v8c9-9ccm](https://doi.org/10.1103/v8c9-9ccm)

I. INTRODUCTION

Understanding turbulence is a long-standing open problem in classical physics. Computational fluid dynamics (CFD) plays a key role in that quest, as it aims at numerically simulating turbulent flows. However, in standard numerical approaches, such as direct numerical simulations (DNS), accuracy is often hindered by the large mesh sizes required to capture the multiscale properties of turbulent flows [1]. This is the CFD incarnation of the infamous curse of dimensionality that affects many computational problems. In this regard, tensor networks (TNs) have emerged as a computational tool that has proven crucial in broadening the set of problems that can be tackled with numerical techniques, from machine learning models [2] to the simulation of many quantum systems [3,4]. TNs enable efficient data compression of high-dimensional objects while still preserving their fundamental features and correlations. The best-known example is the celebrated matrix product state (MPS) [3,4], also referred to as tensor train (TT) [5,6]. This TN was originally proposed for one-dimensional (1D) quantum systems [7]. However, its versatility and ease of

manipulation have allowed it to adapt well to diverse scenarios [8,9], recently including fluid simulations [10–17].

Indeed, the seminal work [10] opened a research program aimed at simulating the time evolution of a fluid field within the compressed TT representation. To this end, significant efforts have been put into translating standard CFD practices to TT native methods, from the treatment of non-trivial boundary conditions, including immersed bodies with complex geometries, to numerical schemes for the efficient retrieval of the solution from its TT representation [11,13,14]. These TT-based CFD solvers have reported promising results, indicating computational advantages in both memory and run-time. The general intuition is that the scale separation induced by the Kolmogorov energy cascade mechanism [18], whereby energy is transferred locally from one spatial scale to the next smaller one, should render TT representations efficient, in analogy with local interactions in 1D quantum systems. However, the computational resources required by these algorithms for addressing turbulent flow scenarios have not yet been systematically investigated. Specifically, the investigation of the key statistical metrics characterizing turbulence within the context of TT solvers remains absent. This is particularly unsettling, since turbulent flows precisely define the regime in which the computational advantages potentially offered by TTs are needed the most.

With this motivation, we conduct a comprehensive analysis to benchmark the TT encoding for turbulent flows. Throughout the work, we consider incompressible fluids in a periodic cubic domain and examine the statistical properties

*Contact author: stefano.pisoni@tii.ae

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

of turbulent velocity fields—such as the energy spectrum, flatness, and probability distribution function (PDF) of velocity increments—as our main metrics. We divide our paper into three aspects: first, we consider *single snapshot compression*. We encode turbulent snapshots on a 3D mesh of $1024^3 = 2^{30}$ pixels into TTs of $N = 30$ tensors (i.e., 30-qubit MPSs). We check the statistical metrics of the resulting TTs when varying the bond dimension χ (i.e., the maximal tensor size), which controls the compression rate of the TT encoding. The snapshots are taken from a reference DNS dataset [19], with $\text{Re}_\lambda = 315$. We show that the TT with $\chi = 1000$, which retains the 2.2% of the total number of parameters, already reproduces the statistical metrics under consideration with small deviations from the ground truth. We also find that the ordering of the matrices in the TT is a crucial choice for this task, with the *interleaved* one outperforming the *stacked* one. To our knowledge, this constitutes the most rigorous analysis of TT compression of a fully developed turbulent flow. Second, we present a *TT-based 3D fluid solver* whose memory and run-time scale polylogarithmically with the mesh size. This is an extension of the solver presented in Ref. [11]. We initialize the flow on a snapshot taken from the dataset mentioned above [19]. The time evolution is performed with $\chi = 100$ (retaining only the 0.03% of the total parameters), in a homogeneous and isotropic setting with $\text{Re}_\lambda = 315$, a regime that has never been tested before using TT-based solvers. The physical evolution time covers 9–10 Kolmogorov turnover times. Third, we develop an efficient TT algorithm to generate *synthetic snapshots* that exhibit some features of turbulent flows. The run-time of this algorithm scales polynomially with the bond dimension and linearly with the number of spatial scales. It relies on an interpolation scheme for TT-encoded fields, which we introduce as a technical contribution and that is potentially interesting beyond the current scope [20].

II. PRELIMINARIES

A. TT formalism

We introduce the TT formalism—also known as matrix product state [5,7]—directly applied to the encoding of the velocity vector field $\mathbf{v}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x}))$. Each velocity component is a scalar field discretized on a 1024^3 grid and is represented as an individual TT. Then, the 3D domain constitutes a mesh of 2^N points, specified by $N = N_x + N_y + N_z = 30$ bits, which corresponds to the total number of tensors in the TT. For instance, the velocity component $u(x, y, z)$ (the same holds true for v and w) is given by the vector of elements $u_{\mathbf{i}, \mathbf{j}, \mathbf{k}} := u(x_i, y_j, z_k)$, where the binary strings $\mathbf{i} := (i_1, i_2, \dots, i_{N_x})$, $\mathbf{j} := (j_1, j_2, \dots, j_{N_y})$, and $\mathbf{k} := (k_1, k_2, \dots, k_{N_z})$ index the discretized coordinates x_i , y_j , and z_k , respectively.

Then, each $u_{\mathbf{i}, \mathbf{j}, \mathbf{k}}$ is a product of N matrices:

$$u_{\mathbf{i}, \mathbf{j}, \mathbf{k}} = A_1^{(i_1)} A_2^{(i_2)} \dots A_{N_x}^{(i_{N_x})} B_1^{(j_1)} B_2^{(j_2)} \dots B_{N_y}^{(j_{N_y})} C_1^{(k_1)} C_2^{(k_2)} \dots C_{N_z}^{(k_{N_z})}. \quad (1)$$

This is the TT (or MPS) representation. When the indices i_l , j_l , and k_l —referred to as *physical indices*—are binary valued like in our case, i.e., $i_l, j_l, k_l \in \{0, 1\}$, Eq. (1) is sometimes

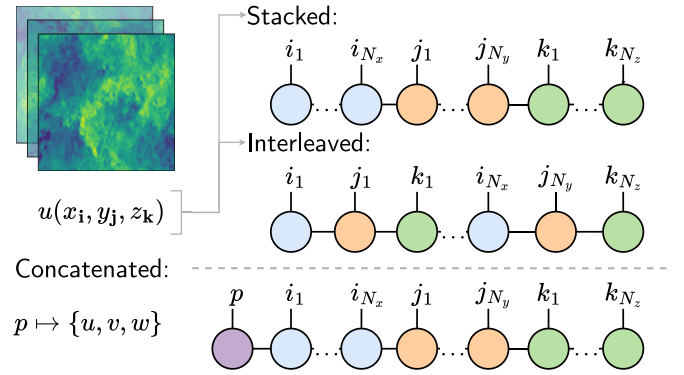


FIG. 1. Different TT encodings of a velocity field. The discretized velocity component $u(x_i, y_j, z_k)$ can be decomposed into two different types of TTs: stacked and interleaved. In both cases, the TT consists of a 1D chain of tensors connected over their virtual (horizontal) indices. Each tensor has a physical (vertical) index labeled by a bit i_l , j_l , or k_l in the binary representation of $(\mathbf{i}, \mathbf{j}, \mathbf{k})$. These binary indices naturally define a notion of spatial scales, with (i_m, j_m, k_m) signifying the m th subdivision of the dyadic grid. The maximal cardinality over all virtual indices is called the *bond dimension* of the TT, which captures the amount of interscale correlations. The stacked and interleaved encodings differ in the ordering of the binary indices, as described in Eqs. (1) and (2). The three components of the velocity field (u, v, w) are encoded into three individual TTs. These can, in turn, be represented by a single TT using an additional tensor with a three-dimensional physical index p , defined as the concatenated TT representation of the full velocity field. An example of this is shown at the bottom for the stacked encoding.

referred to as quantics TT (QTT). Then, the bond dimension χ is defined as the maximum dimension over all $2N$ matrices used. Importantly, the total number of parameters is at most $2N\chi^2$. Hence, the TT representation provides an exponential compression of the 2^N -dimensional vector when χ is constant or scales polynomially with N .

Equation (1) is often referred to as the *stacked encoding* because the matrices are ordered according to the physical dimensions of the 3D grid. However, other arrangements are possible. In particular, we will also consider the following arrangement:

$$u_{\mathbf{i}, \mathbf{j}, \mathbf{k}} = A_1^{(i_1)} B_1^{(j_1)} C_1^{(k_1)} A_2^{(i_2)} B_2^{(j_2)} C_2^{(k_2)} \dots A_{N_x}^{(i_{N_x})} B_{N_y}^{(j_{N_y})} C_{N_z}^{(k_{N_z})}. \quad (2)$$

This is known as the *interleaved encoding*. Figure 1 depicts diagrammatically the two encodings as well as the additional *concatenated* encoding, which is used in Appendix A 1.

We remark that the binary discretization of the domain naturally defines a notion of spatial scales. For example, in a 1D domain, each TT tensor $A_l^{(i_l)}$ labels the discretization at the scale l in a dyadic fashion. Therefore, the chosen arrangement of tensors defines the correlation structure across the spatial scales of the domain, and finding the optimal arrangement amounts to minimizing interscale correlations, leading to a TT encoding with the fewest number of parameters. From a quantum information point of view, the correlations embedded in the state, defined in terms of the entanglement entropy, determine the matrix dimensions in the TT representation. In particular, having local interactions in 1D quantum systems was shown to be a necessary condition for the corresponding

ground states to be well approximated by a TT with low χ [4,21]. This suggests that turbulent signals, which exhibit stronger couplings for neighboring scales as we will see in Sec. II B, might also be well captured by TT with low χ 's.

B. Turbulence

Here, we introduce the key concepts in the study of turbulent fields. First, we recap the theory of homogeneous and isotropic turbulence (HIT) [22] and clarify why this setting is well suited for the TT framework. We will later use these results to benchmark our numerical investigations in Sec. III. Second, we present the key challenges with the current approaches to turbulence simulation.

1. Isotropic and homogeneous turbulence

The word turbulence usually refers to a chaotic and multiscale behavior of fluid flows in space and time. The complexity of turbulent flows stems from the Navier-Stokes equations:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v}, \quad (3a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (3b)$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$ and $p = p(\mathbf{x}, t)$ are, respectively, the velocity and pressure fields at position $\mathbf{x}$ and time t , ρ is the density, and ν the kinematic viscosity. Equations (3a) and (3b) follow, respectively, from momentum and mass conservation [1]. In this work, only 3D incompressible fluids are considered, a condition enforced by Eq. (3b). The emergence of a turbulent phase is associated with a scalar parameter, the Reynolds number (Re), which describes the degree of turbulence in a flow. Specifically, denoting by v_0 the characteristic scale of velocity fluctuations and by L the scale characterizing energy input, $\text{Re} := v_0 L / \nu$.

Given the chaotic nature of turbulence, a statistical approach has been developed through the years [23,24]. In particular, being a system strongly out of equilibrium, new tools have been developed, starting from the observation that the energy dissipation $\epsilon = \nu \langle (\nabla \mathbf{v})^2 \rangle$ is independent of the Reynolds number, where $\langle \cdots \rangle$ means an average in space and time. Specifically, $\epsilon \sim \text{const}$ as $\text{Re} \rightarrow \infty$.

In 1941 [18], Kolmogorov clearly highlighted this fundamental feature of turbulence and showed that, in the HIT setting, two separated ranges of scales emerge: the inertial range and the dissipative range, separated by the Kolmogorov scale $\eta = (\nu^3 / \epsilon)^{1/4}$. Specifically, introducing the longitudinal velocity fluctuations $\delta v(r) := (\mathbf{v}(\mathbf{x} + \mathbf{r}) - \mathbf{v}(\mathbf{x})) \cdot \mathbf{r} / r$, it was shown that in the inertial range ($r \gg \eta$) these are solely controlled by ϵ and r , while in the dissipative range ($r \sim \eta$) the dissipative forces become important. Importantly, the induced scale separation implies that what happens at the tiny scales must be independent of what happens at the large scales, suggesting that the interactions among them decay with increasing scale separation. This was indeed confirmed by analyzing the nonlinear advection term in momentum space [25–27]. This local interaction among scales is what ultimately motivates our study because it corresponds to a local correlation among the tensors in the TT representation, which reflects in an efficient encoding as outlined in Sec. II B.

Moreover, by assuming that the statistical properties of turbulent flows are scale invariant in the inertial range and using only dimensional arguments, one can further conclude that

$$S_p(r) = \langle \delta v(r)^p \rangle \sim \epsilon^{p/3} r^{p/3}. \quad (4)$$

Equation (4) gives rise to the turbulent kinetic energy spectrum, whose Fourier representation in the inertial range follows the celebrated power law:

$$E(k) \propto \epsilon^{2/3} k^{-5/3}, \quad (5)$$

where $k = |\vec{k}|$ is the wave number magnitude. This result will be used in Sec. III as a benchmark for our numerical results. We remark that Eq. (4) requires scale invariance to hold, which is an assumption of Kolmogorov theory. Therefore, deviations from Eqs. (4) and (5) might be observed, giving rise to intermittent phenomena. Intermittency is another fundamental feature of turbulence that is associated with regions of high vorticity in the flow. One standard way to evaluate it is via the so-called flatness, or generalized kurtosis, defined as

$$\Gamma_p(r) \equiv S_p(r) / S_2(r)^{p/2}. \quad (6)$$

Note that Eq. (4) predicts $\Gamma_p(r) \sim \text{const}$ in the inertial range. However, $\Gamma_p(r)$ is observed to increase for $r \rightarrow \eta$, both in DNS and in laboratory experiments [28]. Therefore, we also look at the flatness as an additional metric to benchmark our numerical results in Sec. III. Finally, we remark that the intermittent nature of a turbulent signal can also be inferred from the PDF of the longitudinal velocity increments. In fact, the emergence of a non-Gaussian statistics, characterized by fat tails signaling the occurrence of extreme events, marks the turbulent phase of a fluid flow [22]. We also use this metric to benchmark our numerical results in Sec. III.

2. Turbulence simulation

Simulating turbulent flows has always been a major challenge in computational physics. Ideally, one would simulate any flow regime using DNS, a mesh-based method that evolves the discretized version of Eq. (3) in time without additional modeling. However, for turbulent flows, the required mesh size for an accurate DNS solution increases with the Reynolds number. Specifically, the minimal number of spatial scales M to be resolved is given by $M \sim \frac{3}{4} \log_2 \text{Re}$. In fact, as discussed in Sec. II B, turbulent fluctuations span a large range of scales, from the Kolmogorov scale η associated with the tiniest whirls of turbulence to the integral scale L where energy is injected into the system. This is referred to as the inertial range, and it is well known that $\frac{\eta}{L} \sim \text{Re}^{-3/4}$. This scale separation is what ultimately hinders DNS of turbulent flows. From the TT perspective, since each tensor in the TT represents a scale, at least M tensors are needed per spatial dimension.

III. RESULTS

In this section, we present the numerical results of this work. First, in Sec. III A, we compress turbulent snapshots and look at how the key statistical measures (introduced in Sec. II B) are reproduced when varying the number of parameters in the TT representation, controlled by χ . Next, in

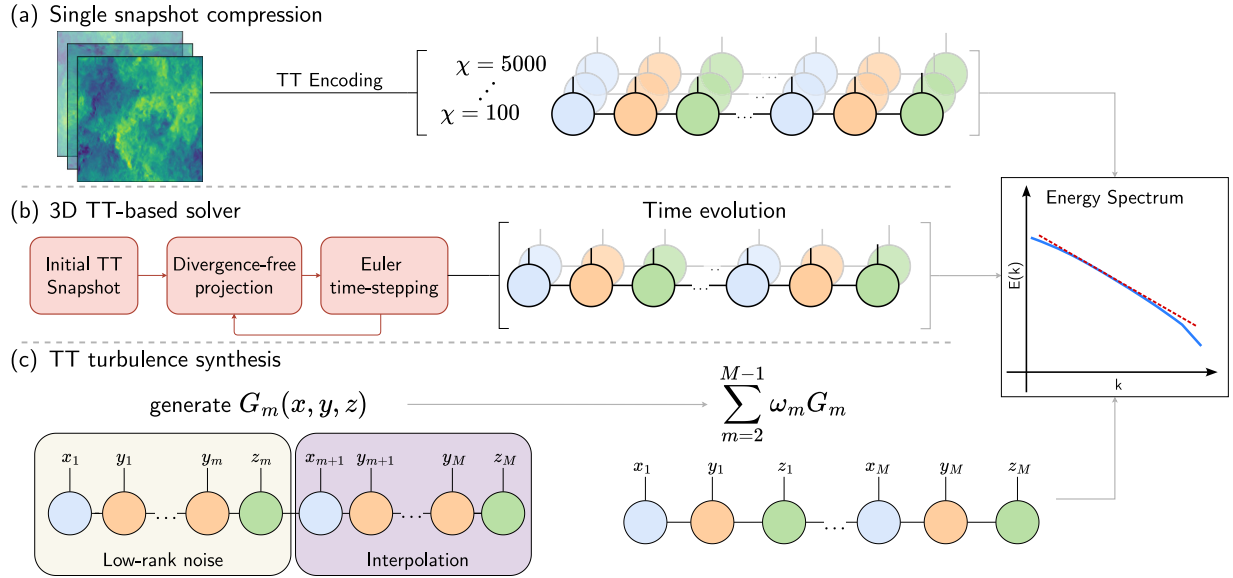


FIG. 2. Schematic representation of the three investigated aspects of turbulence. We numerically investigate three different settings to analyze the TT encoding of turbulent flows. (a) Single snapshot compression. Here, we encode the velocity field at a given time, called a *snapshot*, into its corresponding TT representation. We use the well-known turbulence DNS dataset [19] and compare key statistical metrics of the compressed TT snapshots with those of the original dataset for increasing bond dimensions χ . We perform this analysis for both stacked and interleaved encoding. The results are reported in Sec. III A. (b) 3D TT-based solver. Using the TT snapshot obtained from the DNS solution as the initial condition, we simulate the time evolution of the flow completely within the TT representation. We first project onto the divergence-free manifold of the velocity field and perform the time stepping using an explicit Euler scheme. We compute the energy spectrum of the time series of solutions obtained. We report the results for the interleaved encoding in Sec. III B. (c) TT snapshot synthesis. Here, we construct a TT field that exhibits some key turbulent features: the divergence-free condition, the Kolmogorov energy spectrum, and intermittency, quantified by the flatness. The algorithm generates random low-rank TTs ($\chi = 10$) at each spatial scale m and then interpolates them to the desired resolution M . The final snapshot is the summation of these TTs weighted by the appropriate weights ω_m . We compute and verify these properties for an ensemble of 20 snapshots. In this instance, we restrict ourselves to the interleaved encoding. The detailed explanation with results is outlined in Sec. III C.

Sec. III B, we use turbulent snapshots to initialize and benchmark the time evolution performed with our 3D TT-based solver. Finally, in Sec. III C we present an efficient algorithm to synthesize snapshots with turbulentlike features in the TT representation.

For the first two subsections, we use the DNS of isotropic turbulence generated in Minotauro at the Barcelona Supercomputing Center [19] as the reference dataset. The flow features a Reynolds number at the Taylor microscale λ (often used in turbulence; see Ref. [1]) of $Re_\lambda = 315$ on a periodic cubic domain of $(2\pi)^3$, which is mapped to a discretized computational domain of $(1024)^3$. The code is available in Ref. [29].

A. Single snapshot compression

In this section, we analyze the extent to which the TT representation is able to compress turbulent snapshots. To this end, we consider the following statistical metrics: the energy spectrum [Eq. (5)], the flatness [Eq. (6)], and the PDF of the longitudinal velocity increments. For all of these metrics, we compare the TT-compressed snapshots with the original snapshots across various bond dimensions and two encodings: stacked and interleaved. In Appendix A 1, we additionally present the energy spectrum for the concatenated encoding.

The results for the energy spectra are reported in Fig. 3(a). As expected, we observe that the energy spectra reproduced by the truncated snapshots get closer to the ground truth as the

bond dimension increases. Specifically, at roughly $\chi = 1000$, the spectrum already covers the entire flat region of the inertial range for both the encodings. Smaller bond dimensions also reproduce good portions of the spectrum, with the high-frequency part better captured by the *interleaved* encoding. We note that $\chi = 1000$ retains only the 2.2% of the total number of parameters, achieving a remarkable compression. We further comment that the resolved wave number for $\chi = 1000$ is around $k = 70$, where the reconstructed spectrum detaches from the original one.

In Fig. 3(b), we report the flatness of the truncated snapshots for the interleaved encoding and for various bond dimensions. The inset shows both the encodings at $\chi = 1000$. The figure highlights the effect of the interleaved TT compression on the intermittent nature of turbulence. To further characterize the intermittency of the compressed snapshots for the two different encodings, we report in Figs. 3(c) and 3(d) the PDF of the longitudinal velocity increments. From our analysis, we conclude that the two proposed encodings behave quite differently, with the interleaved one capturing the higher-order statistics much better than the stacked one. In fact, the stacked encoding tends to smooth out the velocity field dramatically, especially at lower values of χ , while the interleaved one allows non-Gaussian statistics to occur. However, we also note that the interleaved encoding seems to introduce artificial discontinuities or fluctuations at very small scales when a low bond dimension ($\chi < 500$) is employed, as

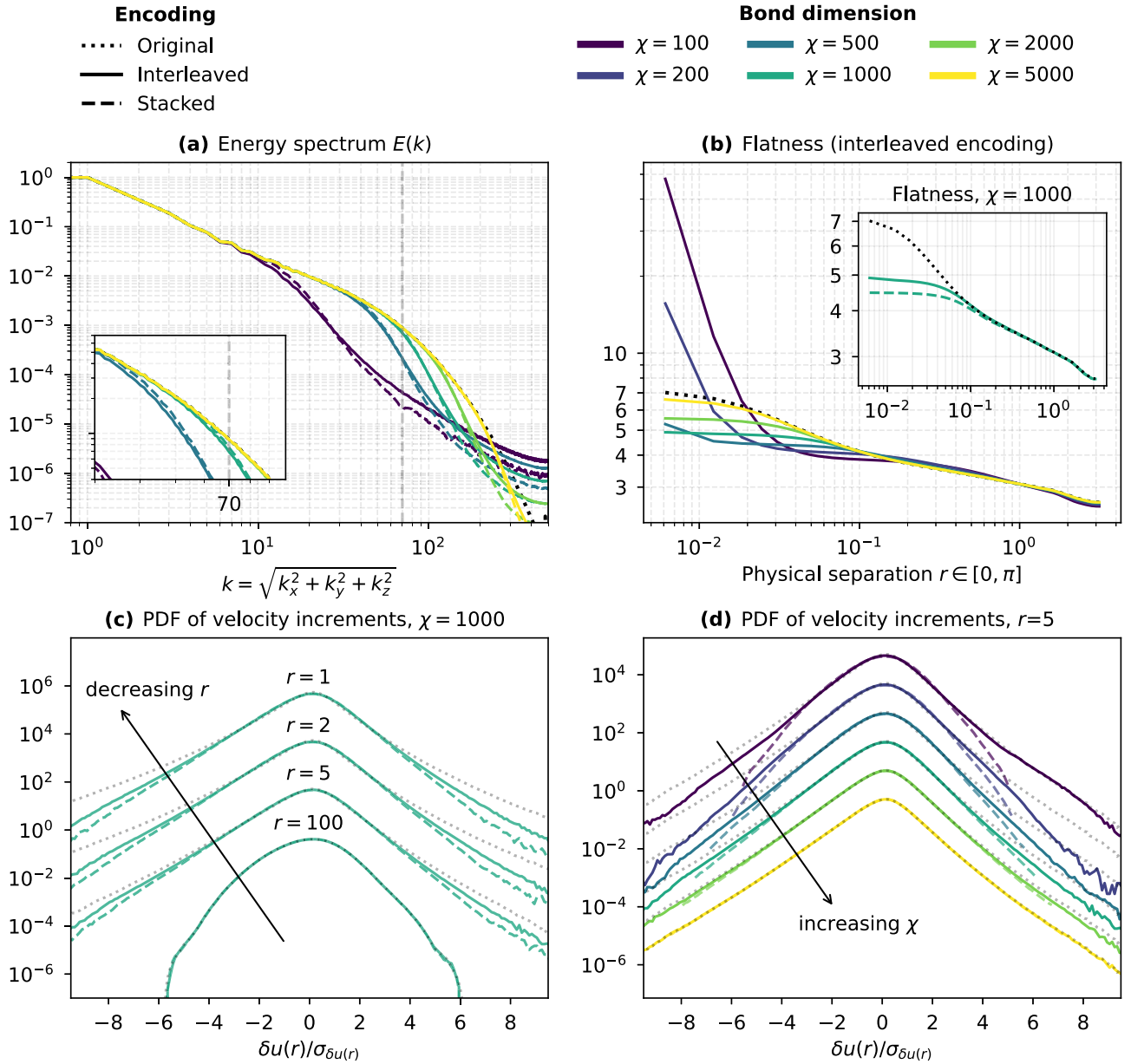


FIG. 3. Single snapshot compression: Turbulent statistical metrics for the compressed TT snapshots. The snapshots have been compressed and projected to satisfy the divergence-free condition. This is achieved through an iterative scheme that alternates between the compression and projection steps, for a total of ten iterations. We end with a compression step, such that the resulting TTs have exactly the reported χ . (a) Kinetic turbulent energy spectrum as a function of the wave number magnitude for various values of χ and the two possible encodings. The plots are on a log-log scale, and the approximately linear region corresponds to the inertial range, with a power-law decay given by Eq. (5). We note that $\chi = 1000$ already reproduces entirely the inertial range for both the encodings, with the compressed spectra detaching from the original one (dotted line) around $k = 70$. The inset shows an enlarged version of the main plot around that region. $\chi = 1000$ corresponds to a TT with only 2.2% of the total number of parameters required for its dense-vector representation. (b) The main plot reports the flatness, as per Eq. (6), for the interleaved encoding and various bond dimensions. The inset shows the flatness for both the encodings at $\chi = 1000$. We see that for very small separation distances the curves deviate from the ground truth, either by overshooting (for low χ 's) or by undershooting (for large χ 's). We explain this behavior in the main text. (c) PDF of the longitudinal velocity increments with $\chi = 1000$ for various separation distances. Here, we report the separation distances in terms of computational unit cells for clarity. The curves are shifted for the sake of clarity. We note that the interleaved encoding captures the non-Gaussian statistics that emerge for small separations better than the stacked one. (d) PDF of the velocity increments for a given separation distance ($r = 5$) and various bond dimensions. We again observe that the interleaved encoding captures the non-Gaussian statistics better than the stacked one, especially for moderate bond dimensions. We also observe, as expected, that we converge to the ground-truth statistics as χ increases.

marked by the overshooting of the two curves in Fig. 3(b), well above the ground-truth values.

This fact deserves a more detailed comment. We understand that there exists a fundamental trade-off between bond-dimension truncation and divergence-free projection. In fact, a TT compression introduces discontinuities that cannot be balanced out by just enforcing a global divergence-free condition on the three TTs representing the three vector components of the velocity field. Then, at low bond dimensions and small separation distances, this effect is stronger and causes the overshooting of the flatness observed in Fig. 3(b).

B. 3D TT-based solver

In this section, we aim at describing and validating the 3D TT-based solver, which is an extension of the two-dimensional (2D) solver introduced in Ref. [11]. The solver is based on a finite-difference approach and fundamental TT truncation and contraction schemes that feature internal routines, such as DMRG-type algorithms for solving linear systems when necessary. After discretization, we solve Eqs. (3a) and (3b) by a standard Chorin's projection scheme, where we first evolve the solution with explicit Euler time stepping and subsequently correct the solution to satisfy the divergence-free condition. This last step involves solving a Poisson-like equation for pressure, which dominates the solver's algorithmic complexity. In Appendix A 2, we provide details about the 3D solver.

The simulations are performed in an empty 3D cubic domain with periodic boundary conditions, a maximum fixed bond dimension $\chi = 100$, and $N = 30$. All the simulation parameters are equivalent to those of the dataset considered [19], except for the time step and the external forcing term that we do not include in Eq. (3a). In fact, our simulations use an explicit Euler time-stepping scheme, which is constrained by standard convergence criteria. In particular, we need to satisfy the Courant-Friedrichs-Lewy (CFL) condition, which requires that $\frac{\delta v \Delta t}{\Delta x} \leq 1$, where δv is the maximal absolute value of velocity fluctuations. For this reason, we set the time step to be 10 times smaller than that used in the reference dataset, achieving $\text{CFL} = 0.16$. To initialize the evolution, we use the compressed TT representation of a snapshot from the turbulent dataset [19] as the initial condition. During the evolution, we check whether the typical behavior of $E(k)$, expected at $\chi = 100$, is well reproduced. The incompressible flow setting allows us to further check the numerical stability of the solver by looking at the total divergence of the vector field, $\nabla \cdot \mathbf{v}$, over time. Therefore, we also compute the divergence and look at its L^∞ -norm as an additional sanity check. Given that the analysis for the single snapshot compression in Sec. III A found the *interleaved* encoding to be the more appropriate one, we only report the time evolution for this encoding in what follows. The results are described and summarized in Fig. 4. We note the drastic compression achieved by these simulations: Only the 0.03% of the total number of parameters is used, compared to the reference dataset. We observe that the energy spectrum slightly deviates over time from its expected shape at $\chi = 100$. This might be due to the limited χ used for the simulations as well as the absence of a forcing term in Eq. (3). However, from a numerical point of view, the

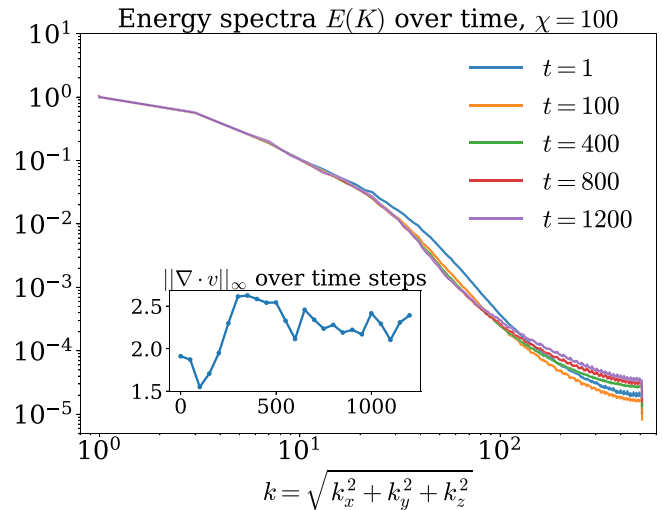


FIG. 4. 3D TT-based solver: $E(k)$ and divergence L^∞ -norm during the 3D TT time evolution. We show the energy spectrum $E(k)$ as a function of the wave number k (main figure) and the L^∞ -norm of the divergence (inset) over time. We performed a total of 1200 computational time steps, with the following parameters: kinematic viscosity $\nu = 0.00067$, time step $\Delta t = 0.0002$, number of tensor cores per spatial dimension $N_x = N_y = N_z = 10$, and Reynolds number at the Taylor microscale $\text{Re}_\lambda = 315$. These simulations use only 0.03% of the total number of parameters needed for the full vector representation. The number of time steps simulated is equivalent to 120 time steps in the original dataset presented in the main text [19]. This evolution time roughly corresponds to 9–10 Kolmogorov turnover times. The TTs used have a low bond dimension, $\chi = 100$. The low bond dimension, and the absence of a forcing term in Eq. (3), explains the small deviations from the expected shape for $E(k)$.

TT solver appears to be stable and consistent, as the total divergence does not blow up over time.

We conclude by pointing out that TT-based simulations with higher χ values are needed to compete with state-of-the-art DNS. With this respect, we point out that the limited bond dimension $\chi = 100$ is dictated by the memory consumption of the current implementation of the TT-based solver. Future improvements to the solver might enable larger bond dimensions, opening the way for DNS simulations at mesh sizes beyond the reach of standard CFD approaches. Currently, another limiting factor is the small time step, which is constrained by stability criteria. This limitation can be overcome by replacing the explicit Euler scheme with an implicit method.

C. TT snapshot synthesis

We propose a constructive algorithm to generate 3D snapshots that exhibit turbulentlike features within the TT representation. Specifically, the snapshots satisfy the divergence-free condition [Eq. (3b)], the correct power-law decay in the energy spectrum [Eq. (5)], and nonzero intermittency quantified by the flatness [Eq. (6)]. By constructing synthetic snapshots directly as TTs, our algorithm reduces both computational and memory costs, thereby enabling rapid generation of high-resolution flow fields for real-time applications. This is particularly useful for many applications in which the

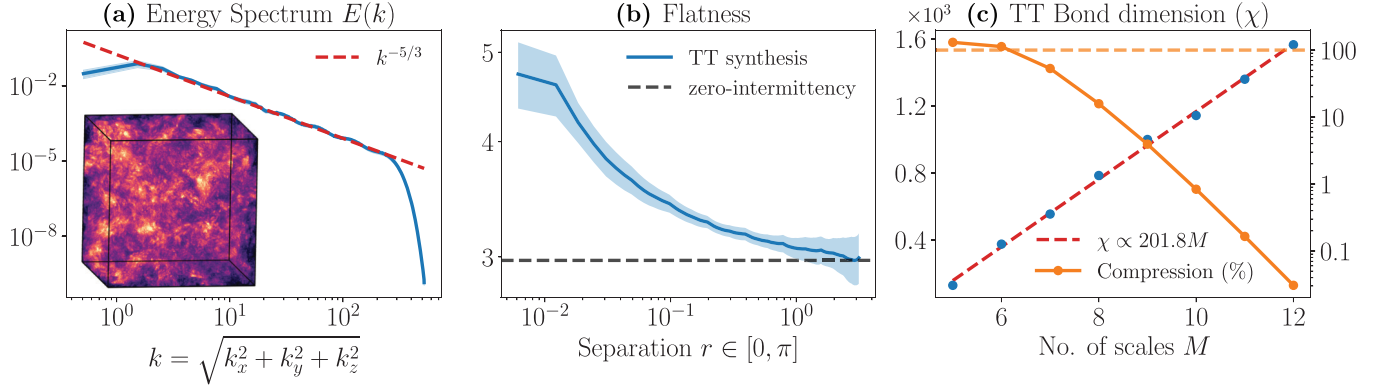


FIG. 5. TT snapshot synthesis: Turbulence metrics for the synthetic generated snapshots with the proposed TT algorithm. Panel (a) shows the energy spectrum $E(k)$ plotted against wave number $k = |\mathbf{k}|$ on a log-log scale, with the dashed red line indicating $k^{-5/3}$ power-law decay. We also plot a 3D snapshot of the velocity magnitude of the synthetic turbulent field. Panel (b) presents the flatness (kurtosis) of velocity increments [Eq. (6)] as a function of physical separation r , on a log-log scale; the deviation from the flat line indicates the presence of intermittency, or non-Gaussianity, in the velocity fluctuations. Panel (c) displays the TT bond dimension χ vs the number of scales M (one-third of the total TT tensors), together with the dashed linear fit $\chi \propto 201.8M$, showing a linear growth. Moreover, we show the compression rates achieved by the TT-generated snapshots with respect to the full vector representations as a function of M . We average over 20 synthetic snapshots with random seeds, and the shaded regions denote one standard deviation.

rapid generation of turbulentlike flows is demanded but often unfeasible due to fundamental limitations of DNS. State-of-the-art methods for generating turbulentlike snapshots include Fourier spectral synthesis, wavelet decomposition, and multiscale cascade models [30–32]. In this work, we follow a multiscale cascade approach, since its construction is well suited for the TT representation in interleaved encoding. A multiscale cascade model begins by generating random fields at each subdivision level (or scale) m of the domain. Then, each of these fields is interpolated up to the desired scale M . The final field is an additive or multiplicative combination of them, weighted by the appropriate amplitudes. To enforce incompressibility of the synthesized velocity field $\mathbf{v}$, we introduce an auxiliary vector potential $\mathbf{A}$ called the stream function, such that $\mathbf{v} = \nabla \times \mathbf{A}$. Hence, we directly generate the derivatives of the stream function via

$$\partial_i A_j(x, y, z) = \sum_{m=2}^{M-1} \omega_m G_{i,j}^m(x, y, z), \quad (7)$$

where the sum starts from the second scale $m = 2$, i.e., $4 \times 4 \times 4$ grid, and terminates at the second-last one $M - 1$. In Eq. (7), the amplitudes ω_m are chosen to reproduce the hierarchical order of the cascade mechanism, as they control the energy injected at each scale m . Specifically, according to Eq. (4), choosing $\omega_m = 2^{-m/3}$ allows us to reproduce Kolmogorov spectrum. The m th term, $G_{i,j}^m$, is built as shown in Fig. 2(C): First, for each j , we initialize a TT with $3m$ tensors with bond dimension χ_G . The entries of each tensor are sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$, where σ is chosen such that the variance of the corresponding random field with $2^m \times 2^m \times 2^m$ entries is ~ 1 . Second, for each i , we obtain a TT of the derivative ∂_i by extending the number of tensors from $3m$ to $3M$. This extension to the final scale M with periodic boundary conditions is done via our TT interpolation algorithm, as described in Ref. [20].

By construction, our approach allows us to control the bond dimension of the synthesized field. Specifically, each generated $G_{i,j}^m$ has a bond dimension scaling linearly with the number of tensors. Moreover, by leveraging the subadditivity of the TT ranks, the full construction in Eq. (7) yields a bond dimension χ that scales linearly with the total number of TT tensors N . Remarkably, this technique [20] reduces the bond dimension of the TT representation of the velocity field by analytically constructing $\partial_i A_j$, rather than computing $\mathbf{A}$ and then applying an approximate derivative TT operator.

To numerically benchmark our synthetic turbulence generator algorithm, the initial noisy TTs are generated with $\chi_G = 10$, and we set $M = 10$. We report the features of our synthetic field in Fig. 5, where we can observe that we are able to reproduce the correct power energy spectrum within the allowed frequency range for our finite lattice. We also compute the flatness of our synthetic field and observe intermittent behavior. Finally, we explicitly show that the bond dimension of the TT representation of the synthetic velocity field scales linearly with M , achieving an exponential memory compression. We remark that this is not a physical turbulent snapshot, but rather a multiscale signal that exhibits turbulentlike features.

IV. CONCLUSIONS

We conducted a detailed investigation of TTs applied to turbulent flows. This consisted of three aspects: turbulent snapshot TT compression, TT-based simulation of turbulent flows, and the generation of 3D TT snapshots. The analysis of compressed turbulent snapshots is, to our knowledge, the most complete benchmark of TT encodings of fully developed turbulent flows. Retaining only 2.2% of the number of parameters, TTs still successfully reproduce the key turbulence features of the original snapshots for a DNS with $\text{Re}_\lambda = 315$ on a 3D mesh with 2^{30} grid points. Interestingly, we find that interleaved encoding outperforms stacked encoding across all

turbulent metrics. As for time evolution, our benchmark on the same turbulence dataset constitutes an unprecedented flow regime for TT methods. Our solver operates entirely within the TT representation with its memory scaling polylogarithmically with the mesh size. Finally, our synthetic snapshot generator demonstrates that it is possible to efficiently construct a low-bond-dimension TT field that reproduces some of the characteristic properties of a turbulent flow. This algorithm features linear scaling of the bond dimension with the number of spatial scales and is equipped with a TT interpolation technique introduced in Ref. [20].

Although the simulated flows considered here do not have any immersed bodies, we note that our 3D solver is fully compatible with the TT mask framework presented in Ref. [11]. In this regard, the TT interpolation scheme provides a way to construct such masks in 3D, as demonstrated in Ref. [20]. Moreover, further improvements to the solver might be possible, such as migrating to implicit time-stepping schemes and optimizing the TT subroutines. This may both speed up the simulations and allow for larger values of χ . Additionally, we remark that other tensor network encodings could also be explored in the context of turbulent flows.

ACKNOWLEDGMENT

The authors acknowledge L. Biferale, C. Calascibetta, and M. Buzzicotti for stimulating discussions.

DATA AVAILABILITY

The data that support the findings of this article are openly available.

APPENDIX: SUPPLEMENTARY MATERIAL

1. Concatenated encoding

In this Appendix, we present and analyze the idea of encoding the vector field components (u, v, w) into a single TT, stitching them together with an additional extra tensor. The physical leg of the extra tensor will then be three dimensional to label the three different components of $\mathbf{v}$. In analogy to the snapshot compression analysis, we fix a maximal bond dimension χ and truncate the augmented TT according to it.

For this comparison, we only consider the energy spectrum $E(k)$ as a metric. We plot the results in Fig. 6. We only plot the concatenated TT built from individual stacked TTs, because we empirically do not observe any difference with respect to the concatenated TT built from individual interleaved ones. However, we compare $E(k)$ against both the stacked and interleaved encodings. We highlight the impressive compression given by the concatenated TT with respect to three individual TTs: Indeed, the upper bound for the total number of parameters in the former case is $2\chi^2(N+1)$, whereas in the latter one is $3(2\chi^2N)$.

For convenience, we plot the results only for $\chi = 2000$. In fact, we observe that the energy spectra differ from each other for smaller χ 's, and are practically equivalent for larger ones. For $\chi = 2000$, $E(k)$ in the inertial range is perfectly matched by the concatenated TT. In terms of compression, the concatenated encoding with $\chi = 2000$ reduces the number

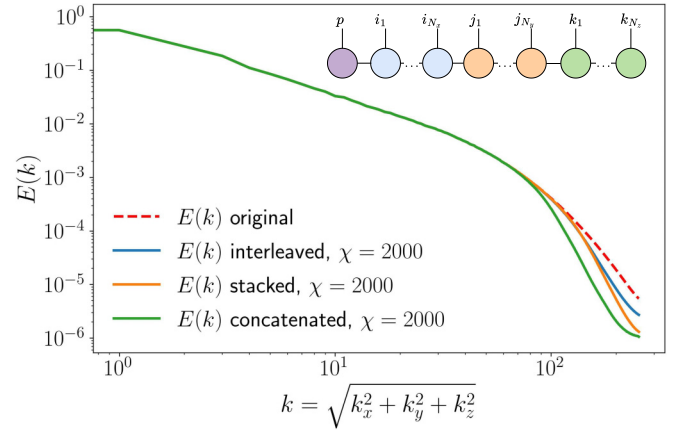


FIG. 6. Concatenated tensor train energy spectrum. We plot the energy spectrum for the three different encodings introduced in Fig. 1 for a fixed bond dimension $\chi = 2000$. Note that the concatenated encoding already encodes the three components of the vector field $\mathbf{v}$ in the extra physical leg p . Therefore, the number of parameters it retains is lower (2.2%) than the number of parameters needed to encode the three vector components with the stacked or interleaved encodings (7%).

of total parameters to 2.6% of the full vector representation. Three individual TTs with the same χ would reduce it to 7%.

This suggests that whenever a high bond dimension is needed to accurately capture the interscale correlations in the fluid, the concatenated encoding might become relevant to further reduce the number of total parameters without sacrificing accuracy with respect to the energy spectrum.

2. TT-based solver

Here, we present the TT-based solver for the Navier-Stokes equations. We use this to simulate the time evolution of the turbulent snapshots from the DNS dataset, as discussed in Sec. III B. Our solver is an extension of the TT framework introduced in Ref. [11], with natural extensions from 2D to 3D. The distinguishing feature is that the time evolution is performed completely within the TT representation. We achieve this by representing the discretized differential operators, appearing in Eq. (3), in their corresponding TT form known as matrix product operators (MPOs). For the finite-difference operators used in this work, the corresponding MPOs are analytically constructed with low bond dimensions [33].

In Table I, we report the asymptotic complexities for several steps of the TT framework. There, we include the complexities for two modes of evaluation of the TT solution along with retrieving the full-resolution solution. Since the memory cost of storing the full-resolution solution grows exponentially with N , these two modes, namely, *pixel sampling* and *coarse graining*, serve as an efficient alternative. These were also introduced in the proposed framework [11], to which we refer for detailed explanations. In this work, we can still afford to compute the full-resolution solution. Hence, these two evaluation modes are not used in this work. However, once again, these evaluation modes will become indispensable at high resolutions when the full resolution is prohibited by the exponential memory requirements. For $N = 30$, the resulting

TABLE I. Asymptotic time complexities of the main TT subroutines of the 3D solver. We report the asymptotic worst-case time complexity of each subroutine of the TT solver. We report the theoretical scalings of the algorithms with respect to the number of TT tensors (N) and the bond dimension (χ).

Algorithmic task	Time complexity
Divergence-free projection	$\mathcal{O}(N \chi^6)$
Euler time stepping	$\mathcal{O}(\chi^6)$
Coarse-grained evaluation	$\mathcal{O}(N \chi^3)$
Pixel sampling (per pixel)	$\mathcal{O}(\chi^3)$
SVD truncation	$\mathcal{O}(N \chi^3)$

vector is still small enough for us to evaluate the full vector from the compressed TT. We then compute the various measures, such as $E(k)$, from the full-resolution solution.

a. Time stepping and numerical stability

For completeness, we discuss the details of the time-stepping scheme, following the exposition in Ref. [11]. We evolve the velocity fields within the divergence-free manifold using Chorin's projection method [34,35]. The spatial discretization is given by the uniform grid and the differentiation is computed using the finite-difference method. For the discretization of time, we implement an explicit Euler time-stepping scheme. Starting from Eq. (3), this results in the following:

$$\frac{\mathbf{v}_{t+\Delta t} - \mathbf{v}_t}{\Delta t} + (\mathbf{v}_t \cdot \nabla) \mathbf{v}_t = -\frac{1}{\rho} \nabla p_{t+\Delta t} + \nu \nabla^2 \mathbf{v}_t, \quad (\text{A1})$$

along with the divergence-free condition:

$$\nabla \cdot \mathbf{v}_{t+\Delta t} = 0. \quad (\text{A2})$$

Next, ignoring the pressure term in Eq. (A1), we define an intermediate velocity field given by

$$\mathbf{v}_{t+\Delta t}^* = \mathbf{v}_t + (-\mathbf{v}_t \cdot \nabla) \mathbf{v}_t + \nu \nabla^2 \mathbf{v}_t \times \Delta t. \quad (\text{A3})$$

However, the intermediate velocity does not satisfy the divergence-free condition. We use the Helmholtz decomposition of $\mathbf{v}_{t+\Delta t}^*$ to define the solenoidal and irrotational components of the vector field. By definition, the solenoidal field has zero divergence at all points, which is indeed our objective.

Next, instead of finding the solenoidal component directly, we determine the irrotational component of the intermediate velocity. As stated in Chorin's projection method [34], this reduces to solving the Poisson equation for the pressure field:

$$\nabla^2 p_{t+\Delta t} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{v}_{t+\Delta t}^*, \quad (\text{A4})$$

which we solve using a DMRG-based linear system solver [36]. Here, we remark that alternatives such as the TT-AMEn solver [37] could be used to further optimize the convergence and stability of the linear solver. We consider this enhancement out of scope for this work and leave it open for future work.

By subtracting the gradient of the pressure field, we find the solenoidal component of the velocity field, which completes

the time evolution for one time step:

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_{t+\Delta t}^* - \frac{\Delta t}{\rho} \nabla p_{t+\Delta t}. \quad (\text{A5})$$

Moreover, we need to choose the time step size (Δt), which affects the stability of the time stepping. For a stable time evolution, we have to satisfy the CFL condition [38]. It states that for $(\frac{U \Delta t}{\Delta x} \leq 1)$, where U is the characteristic velocity, the information about the flow travels slower than the flow itself, ensuring stable numerical integration. We emphasize that this is only a necessary condition, not sufficient for the algorithm's stability. In the 3D simulations reported in Sec. III B, we set $\Delta t = 2 \times 10^{-4}$. Other simulation parameters include $\nu = 6.7 \times 10^{-4}$ and $\text{Re}_\lambda = 315$, in agreement with the dataset [19].

b. Bond-dimension truncation

Each time evolution iteration involves several TT operations, such as elementwise multiplication [nonlinear term in Eq. (3a)], summation (Euler time step), DMRG-type algorithm (projection step), and TT contractions with differential operators. Each of them increases the bond dimension of resulting TT. Hence, we perform TT rounding after each operation to a fixed bond dimension, for an efficient time evolution. We use the SVD-based truncation algorithm and limit the number of allowed singular values to fix the bond dimension.

c. DMRG-type solver for linear systems

As already discussed, in order to project the velocity fields onto the divergence-free manifold, we solve the resulting linear system using a DMRG-based algorithm. This task includes two major components: tensor contractions and solution of the *local* linear systems. For a fixed bond dimension, the time complexity is then estimated as the combined cost of tensor contractions needed to determine the local systems and the solution of the local linear systems. The tensor contractions are optimized in a way that the most information is reused from the previous DMRG subsweep, which scales as $\mathcal{O}(N \chi^4)$. The resulting local systems are of size $4\chi^2 \times 4\chi^2$ and their exact solution scales as $\mathcal{O}(\chi^6)$. Thus, the resulting scaling for the projection step scales as in $\mathcal{O}(N \chi^6)$.

Previous works [10] included variational approaches to tackle the linear system solution with a favorable worst-case complexity $\mathcal{O}(N \chi^4)$. However, as already emphasized in Ref. [36], the exact solution is to be preferred when the bond dimensions are small enough to allow direct computation.

d. Immersed objects and masks

Moreover, in Ref. [11], a great deal was put into the idea of *mask* that allows to simulate fluid flows around complex geometries, enforcing nontrivial boundary conditions by building the TTs associated with the immersed objects themselves. However, in the context of this work, we are not interested in that construction as we look at a periodic cubic empty domain. Despite this, we remark that numerical simulations of turbulent fluids often involve nontrivial boundary conditions to make the turbulent behavior arise earlier in the dynamical evolution, or at lower Reynolds numbers. One famous example is grid turbulence. Therefore, the concept of

mask might become of practical relevance in future works when trying to simulate grid turbulence or, more in general,

flows around complex geometries in 3D for high Reynolds numbers.

-
- [1] S. B. Pope, *Turbulent Flows* (Cambridge University Press, New York, 2000).
 - [2] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Unsupervised generative modeling using matrix product states, *Phys. Rev. X* **8**, 031012 (2018).
 - [3] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, *Adv. Phys.* **57**, 143 (2008).
 - [4] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, *Ann. Phys.* **349**, 117 (2014).
 - [5] I. V. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.* **33**, 2295 (2011).
 - [6] B. N. Khoromskij, $\mathcal{O}(d \log N)$ -quantics approximation of N - d tensors in high-dimensional numerical modeling, *Constr. Approximation* **34**, 257 (2011).
 - [7] S. R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.* **69**, 2863 (1992).
 - [8] J. Latorre, Image compression and entanglement, [arXiv:quant-ph.0510031](https://arxiv.org/abs/quant-ph.0510031).
 - [9] M. Kastoryano and N. Pancotti, A highly efficient tensor network algorithm for multi-asset Fourier options pricing, [arXiv:2203.02804](https://arxiv.org/abs/2203.02804).
 - [10] N. Gourianov, M. Lubasch, S. Dolgov, Q. Y. van den Berg, H. Babaei, P. Givi, M. Kiffner, and D. Jaksch, A quantum-inspired approach to exploit turbulence structures, *Nat. Comput. Sci.* **2**, 30 (2022).
 - [11] R. D. Peddinti, S. Pisoni, A. Marini, P. Lott, H. Argentieri, E. Tiunov, and L. Aolita, Quantum-inspired framework for computational fluid dynamics, *Commun. Phys.* **7**, 135 (2024).
 - [12] R. D. Peddinti, S. Pisoni, E. Tiunov, A. Marini, and L. Aolita, Technical report on a quantum-inspired solver for simulating compressible flows, [arXiv:2506.03833](https://arxiv.org/abs/2506.03833).
 - [13] M. Kiffner and D. Jaksch, Tensor network reduced order models for wall-bounded flows, *Phys. Rev. Fluids* **8**, 124101 (2023).
 - [14] E. Kornev, S. Dolgov, K. Pinto, M. Pflichtsch, M. Perelshtein, and A. Melnikov, Numerical solution of the incompressible Navier-Stokes equations for chemical mixers via quantum-inspired tensor train finite element method, [arXiv:2305.10784](https://arxiv.org/abs/2305.10784).
 - [15] L. Arenstein, M. Mikkelsen, and M. Kastoryano, Fast and flexible quantum-inspired differential equation solvers with data integration, [arXiv:2505.17046](https://arxiv.org/abs/2505.17046).
 - [16] N. Gourianov, P. Givi, D. Jaksch, and S. B. Pope, Tensor networks enable the calculation of turbulence probability distributions, *Sci. Adv.* **11**, eads5990 (2025).
 - [17] N.-L. van Hülst, P. Siegl, P. Over, S. Bengoechea, T. Hashizume, M. G. Cecile, T. Rung, and D. Jaksch, Quantum-inspired tensor-network fractional-step method for incompressible flow in curvilinear coordinates *Comput. Phys. Commun.* **325**, 110169 (2026).
 - [18] A. N. Kolmogorov, The local structure of turbulence in incompressible viscous fluid for very large Reynolds number, in *Dokl. Akad. Nauk. SSSR* **30**, 301 (1941).
 - [19] J. I. Cardesa, A. Vela-Martín, and J. Jiménez, The turbulent cascade in five dimensions, *Science* **357**, 782 (2017).
 - [20] S. E. Guzman, E. Tiunov, and L. Aolita, Local interpolation via low-rank tensor trains, [arXiv:2601.03885](https://arxiv.org/abs/2601.03885).
 - [21] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys.* **326**, 96 (2011).
 - [22] R. Benzi and L. Biferale, Homogeneous and isotropic turbulence: A short survey on recent developments, *J. Stat. Phys.* **161**, 1351 (2015).
 - [23] A. S. Monin and A. M. Yaglom, *Statistical Fluid Mechanics* (Courier Corporation, New York, 2007), Vol. I.
 - [24] A. S. Monin and A. M. Yaglom, *Statistical Fluid Mechanics; Mechanics of Turbulence* (Courier Corporation, New York, 2013), Vol. II.
 - [25] G. L. Eyink, Locality of turbulent cascades, *Physica D* **207**, 91 (2005).
 - [26] R. H. Kraichnan, The structure of isotropic turbulence at very high Reynolds numbers, *J. Fluid Mech.* **5**, 497 (1959).
 - [27] L. Biferale, Shell models of energy cascade in turbulence, *Annu. Rev. Fluid Mech.* **35**, 441 (2003).
 - [28] A. Arnéodo, R. Benzi, J. Berg, L. Biferale, E. Bodenschatz, A. Busse, E. Calzavarini, B. Castaing, M. Cencini, L. Chevillard, *et al.*, Universal intermittent properties of particle trajectories in highly turbulent flows, *Phys. Rev. Lett.* **100**, 254504 (2008).
 - [29] The code is available at: https://github.com/stefanopisoni/TN_Turbulence.
 - [30] T. Kim, N. Thürey, D. James, and M. Gross, Wavelet turbulence for fluid simulation, *ACM Trans. Graph.* **27**, 1 (2008).
 - [31] A. Scotti and C. Meneveau, Fractal model for coarse-grained nonlinear partial differential equations, *Phys. Rev. Lett.* **78**, 867 (1997).
 - [32] A. Scotti and C. Meneveau, A fractal model for large eddy simulation of turbulent flow, *Physica D* **127**, 198 (1999).
 - [33] V. A. Kazeev and B. N. Khoromskij, Low-rank explicit QTT representation of the Laplace operator and its inverse, *SIAM J. Matrix Anal. Appl.* **33**, 742 (2012).
 - [34] A. J. Chorin, The numerical solution of the Navier-Stokes equations for an incompressible fluid, *Bull. Am. Math. Soc.* **73**, 928 (1967).
 - [35] A. J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comput.* **22**, 745 (1968).
 - [36] I. V. Oseledets and S. V. Dolgov, Solution of linear systems and matrix inversion in the TT-format, *SIAM J. Sci. Comput.* **34**, A2718 (2012).
 - [37] S. V. Dolgov and D. V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, *SIAM J. Sci. Comput.* **36**, A2248 (2014).
 - [38] R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Math. Ann.* **100**, 32 (1928).