LET'S DO TURBULENCE RIGHT: IN SPACE AND TIME!

Anonymous authors
Paper under double-blind review

ABSTRACT

The three-dimensional Navier–Stokes equations govern fluid turbulence, a chaotic and non-linear physical phenomenon characterized by complex eddies interacting across vast spatial and temporal scales. A numerically accurate modeling of this multi-scale energy cascade requires fully resolving both the fine spatial structures and the continuous temporal evolution of the flow field. In this paper, we leverage massive, time-resolved direct numerical simulations, which exceed 28 TB per run, to assess how numerical choices govern the resolved $(3+1)$-dimensional physics of turbulent flows. Building on this analysis, we study how neural networks represent the recorded turbulent motion. Our results highlight that spatial and temporal assessments can disagree for both numerical records and learned representations. We therefore call on the machine learning community to treat $(3+1)$-dimensional turbulence as a four-dimensional phenomenon.

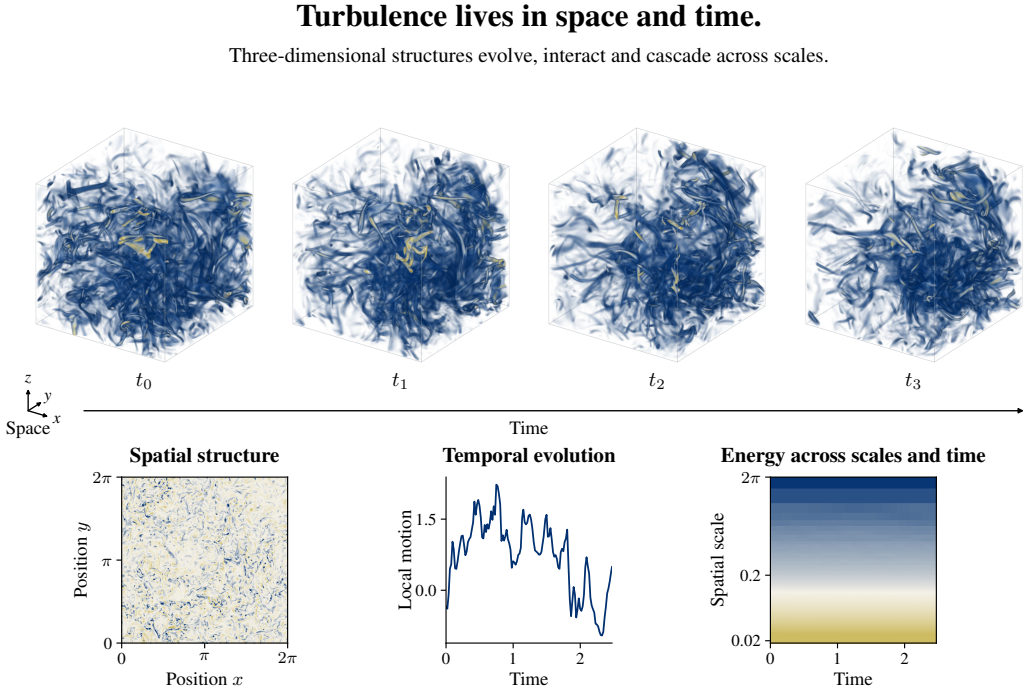


Figure 1: The visualization shows **turbulent** structures from a single 28.9 TB time-resolved **direct numerical simulation**. The evolution of these structures (top) reveals coupled motion across space and time (bottom). How **neural representations** capture these **coupled spatio–temporal dynamics** remains an open question. To address this gap, we first use massive time-resolved simulation records to assess how numerical choices shape turbulent dynamics. We then examine how neural networks represent the recorded motion. Our analyses indicate that spatial and temporal assessments can disagree. *We therefore call on the machine learning community to treat* $(3+1)$*-dimensional turbulence as a four-dimensional phenomenon.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Fluid flows become **turbulent** when inertial effects outweigh viscous resistance, as measured by the Reynolds number. This creates a complex multiscale cascade where energy transfers from large vortices to small eddies before dissipating as heat. Turbulence is a ubiquitous phenomenon, governing transport in engineering flows, the atmosphere, and the oceans (Pope, 2000). Its defining characteristics involve interacting structures across multiple scales in **space and time**, i.e., in $(3+1)$ dimensions (Figure 1).

Resolving this multiscale motion through **direct numerical simulation** (DNS) is computationally demanding. For many industrial applications, simulations instead describe time-averaged or spatially filtered flows, using closure models to account for unresolved motion (Reynolds, 1895; Smagorinsky, 1963). Much of the machine learning literature likewise focuses on selected aspects of turbulence, including closures for reduced simulations (Ling et al., 2016), two-dimensional dynamics (Kochkov et al., 2021), and individual three-dimensional snapshots (Lienen et al., 2024; Shenoy & Frankel, 2026). However, none of these settings address how **neural representations** capture the **coupled spatio-temporal dynamics** of genuine $(3+1)$-dimensional turbulence.

We therefore ask: *How do numerical choices shape the coupled space–time dynamics resolved by DNS, and how do neural representations capture them?*

To address this question, we first assess how numerical modeling and representation shape the resulting dynamics of $(3+1)$-dimensional turbulent structures, drawing on massive, time-resolved direct numerical simulations, which exceed 28 TB per run. Building on this analysis, we study how neural networks represent the recorded turbulent motion.

We summarize our contributions as follows:

1. **Numerical study.** We characterize how resolution, filtering, solver precision, and record quantization affect spatial and temporal behavior, including numerical failures and storage requirements.
2. **Neural network study.** We compare joint space–time representations with independently fitted snapshots, examining how data construction shapes representation quality and when spatial and temporal assessments lead to different conclusions.
3. **Data.** We prepare filtered DNS records at multiple coupled space–time bandwidths from our high-resolution, time-resolved 28.9 TB reference trajectory, with metadata documenting the source simulation and each record's filtering, sampling, and storage.

Overall, we show that spatial and temporal assessments can disagree for both numerical records and learned representations. The difficulty of representing DNS data therefore extends beyond the cost of generating and storing it. *We therefore call on the machine learning community to treat $(3+1)$-dimensional turbulence as a four-dimensional phenomenon.*

2 CHARACTERISTICS OF $(3+1)$-DIMENSIONAL TURBULENCE

The three-dimensional Navier–Stokes equations describe fluid motion across different flow regimes. For an incompressible fluid, the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ obeys (Batchelor, 1967)

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0. \tag{1}$$

In equation 1, the first relation balances fluid acceleration against pressure, viscous, and applied forces; p is pressure per unit density, ν is kinematic viscosity, and $\boldsymbol{f}$ is the applied force. The second equation states that fluid motion preserves volume. The nonlinear transport term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ couples motion at different scales.

This nonlinear transport competes with viscous smoothing. Their relative importance is measured by the Reynolds number $Re = u'L/\nu$, where u' is the root mean square of one velocity component and L is the integral length scale characterizing the large motions. When inertial effects dominate, nonlinear interactions can produce turbulent motion across many scales in space and time. This motion can be characterized using statistical measures from turbulence theory (Pope, 2000). The following measures describe how energy is distributed across spatial scales, how strong and uneven velocity gradients are, and how motion at each scale decorrelates in time (Figure 2).

Energy spectrum. The energy spectrum $E(k)$ describes how kinetic energy is distributed across spatial scales. It groups spatial Fourier modes by their wavenumber magnitude k, with larger k corresponding to smaller scales. In developed turbulence, most energy lies at large scales. An intermediate inertial range can exhibit $E(k) \propto k^{-5/3}$ (Kolmogorov, 1941; Pope, 2000). At the smallest scales, viscosity causes a rapid spectral decay.

Enstrophy. Small-scale motion can carry little energy yet produce strong velocity gradients. Vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ describes local fluid rotation, and enstrophy $\Omega = \langle |\boldsymbol{\omega}|^2 \rangle / 2$ measures its mean squared strength, where brackets denote spatial averages. Spatial derivatives amplify contributions from large wavenumbers. In turbulence, strong small-scale velocity gradients therefore contribute substantially to enstrophy despite the low energy at these scales.

Gradient moments. Gradient moments describe how these velocity gradients are distributed. The *skewness* S measures asymmetry between positive and negative gradients. In developed turbulence, S is typically negative (Ishihara et al., 2009). The *flatness* F measures the relative weight of extreme gradients. In turbulence, F typically exceeds the Gaussian value of 3 (Sreenivasan & Antonia, 1997).

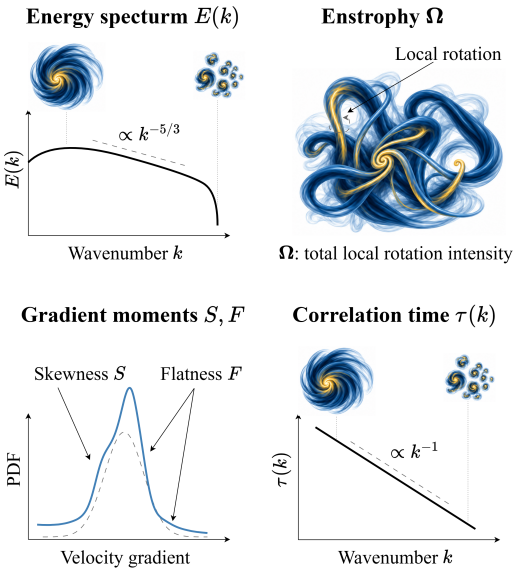


Figure 2: **Turbulence diagnostics (schematic).** The energy spectrum shows how energy is distributed across spatial scales, with small scales carrying little energy but producing strong velocity gradients. These gradients generate local rotation, quantified by enstrophy. Their moments then reveal asymmetry and rare extreme events. Correlation time finally measures how quickly motion at each scale loses memory. Together, these diagnostics describe spatial and temporal dynamics.

Correlation time. Temporal correlations describe how the spatial fluctuations characterized above evolve at fixed coordinates. For each wavenumber shell, the normalized correlation of velocity Fourier coefficients at different time lags measures how quickly these fluctuations lose correlation. Its first crossing of e^{-1} defines the correlation time $\tau_e(k)$. In turbulence, large-scale motion carries smaller structures past fixed positions, a process called sweeping. When sweeping dominates, $\tau_e(k) \propto (u'k)^{-1}$ (Tennekes, 1975; Gorbunova et al., 2021).

3 Direct Numerical Simulation (DNS)

Numerical simulation provides access to the evolving velocity field, allowing the statistics above to be studied under controlled physical conditions. Direct numerical simulation (DNS) evolves the Navier–Stokes equations (1) with spatial grids and time steps fine enough to resolve motion down to the scales at which viscosity dissipates energy (Moin & Mahesh, 1998). Saving states of this field produces a time-resolved velocity record $\boldsymbol{u}^{\mathrm{rec}}$, which we also call a *movie*: a sequence of three-dimensional velocity fields (Figure 1). Its spatial and temporal content can be affected both by how the motion is computed (*numerical modeling*) and by how it is retained (*representation*).

Numerical modeling. Spatial *resolution* determines which scales the solver can resolve. For a cubic grid, let N denote the number of points per spatial direction. Arithmetic *precision* determines the numerical values available during time stepping: rounding enters the evolving state and can affect subsequent motion. We write $\boldsymbol{u}_{N,\sigma}$ for the record produced directly by the simulation, where σ denotes the solver's floating-point format.

Representation. A completed record can be transformed without rerunning the simulation. *Filtering*, denoted $\mathcal{P}_M$, selects the spatial and temporal scales retained in that record. It acts on space and time together, with temporal bandwidth chosen to reflect how rapidly motion at the retained spatial scales evolves. Here M indexes the retained bandwidth and the resampled movie: M points per spatial direction and M times give an M^4 array for each velocity component. *Quantization*, denoted $\mathcal{Q}_q$, rounds recorded velocity values to a floating-point format q. Unlike rounding during evolution, these changes are not fed back into the solver. Thus $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_M \boldsymbol{u}_{N,\sigma}$ retains selected scales of the computed trajectory, while $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{Q}_q \boldsymbol{u}_{N,\sigma}$ contains rounded velocity values.

4 Neural Representation

A DNS record supplies velocity samples at discrete positions and times. A *neural representation* approximates these samples with a continuous function from coordinates to the three velocity components (Figure 3). A standard way to fit its parameters is to minimize mean squared error (MSE), the average squared difference between predicted and recorded values (Sitzmann et al., 2020; Shenoy & Frankel, 2026). Fitting a supplied movie in this way represents recorded motion rather than predicting its future.

Fitting procedures. A movie can be represented as one field evolving in time or as a sequence of spatial fields. In *joint fitting*, one network $f_\theta(\boldsymbol{x}, t) \approx \boldsymbol{u}^{\mathrm{rec}}(\boldsymbol{x}, t)$ takes position and time as inputs, with parameters θ shared across the movie. In *independent fitting*, a separate network $f_{\theta_j}(\boldsymbol{x}) \approx \boldsymbol{u}^{\mathrm{rec}}(\boldsymbol{x}, t_j)$ represents each recorded time t_j. Its parameters θ_j belong only to that snapshot. Both fitting procedures approximate the same velocity samples, but only joint fitting couples their representation through a shared function of space and time.

Network families. The analysis of multiscale turbulence benefits from descriptions that retain information about both scale and spatial localization (Farge, 1992). SIREN parameterizes a continuous field using a fully connected network with sinusoidal activation functions (Sitzmann et al., 2020). Gaussian splatting uses spatially localized Gaussian primitives with learned centers and shapes (Kerbl et al., 2023). An additive velocity-field formulation sums such primitives with learned vector-valued coefficients, giving a generalized Gaussian basis-function network (Poggio & Girosi, 1990). We focus on SIRENs and additive Gaussian fields because they distribute their representational capacity in fundamentally different ways. SIREN uses a single coordinate network across the full domain, whereas the Gaussian field constructs the velocity field from localized, adaptive primitives.

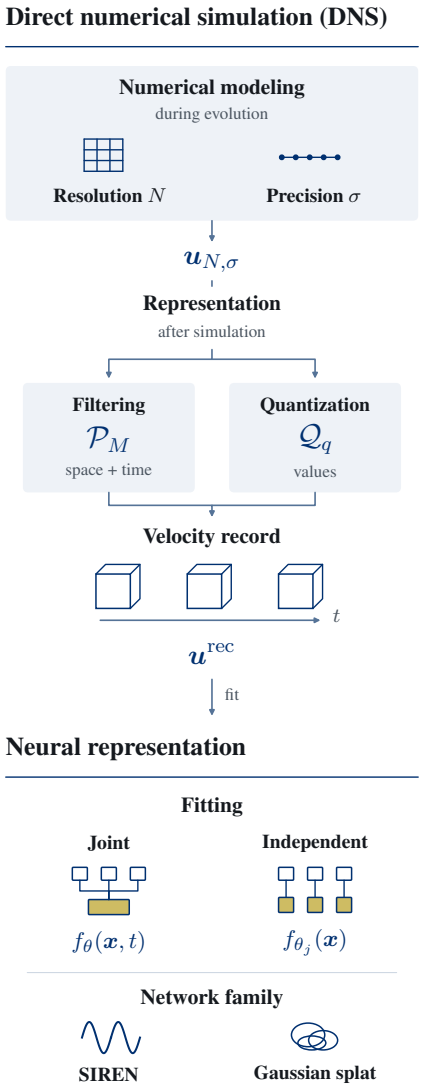


Figure 3: **From DNS to neural representation.** Computation and recording determine the velocity samples supplied to joint or independent neural fitting.

5 EXPERIMENTS

In this paper, we want to understand how numerical choices shape the coupled space–time dynamics present in DNS records, and how neural representations capture the resulting dynamics. To that end, Section 5.1 examines the effects of resolution, solver precision, filtering, and quantization on the recorded flow, and Section 5.2 studies how these records are represented by joint space–time and independent snapshot-wise fits. Applying the same diagnostics from Section 2 to both records and fits separates changes introduced during data construction from those introduced during neural fitting.

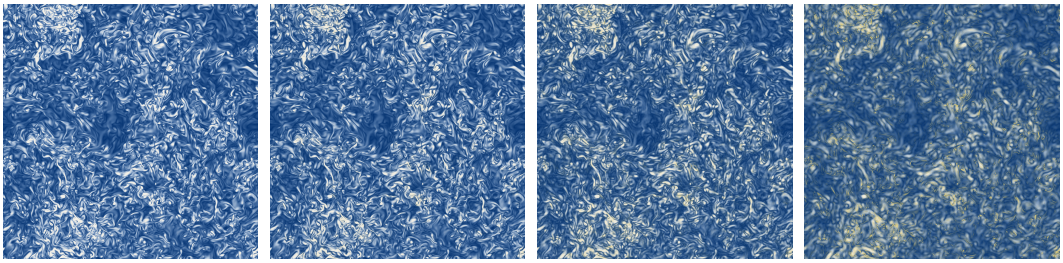


Figure 4: **Space–time filtering.** From left to right, the DNS record $\boldsymbol{u}_{1024,\mathrm{FP32}}$ is followed by its projections $\mathcal{P}_M$ at $M = 512, 256, 128$. Blue shows retained vorticity and yellow highlights the removed part. Each panel shows the same time and plane $z = \pi$ on a common intensity scale, with $0 \le x, y < 2\pi$. Lighter shades reveal stronger vorticity. We will release these filtered records and their construction metadata.

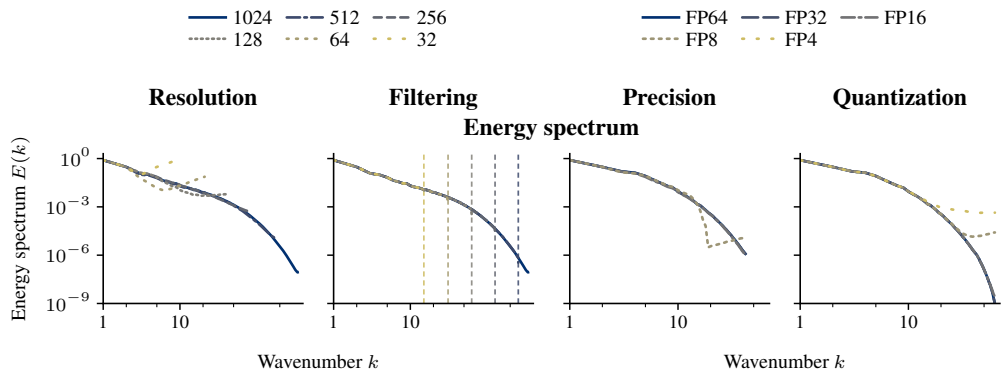
5.1 NUMERICAL MODELING AND REPRESENTATION

We write $\boldsymbol{u}_{N,\sigma}$ for the record produced directly by a simulation with N points per spatial direction and solver format σ. The record analyzed is denoted by $\boldsymbol{u}^{\mathrm{rec}}$. The operator $\mathcal{P}_M$ retains bandwidth M, while $\mathcal{Q}_q$ rounds recorded values to format q.

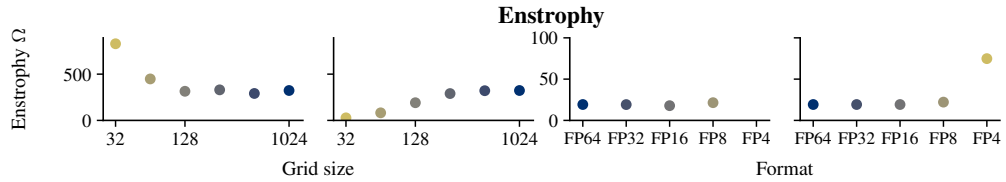
Numerical modeling. For *resolution*, we compare independent simulations $\boldsymbol{u}^{\mathrm{rec}} = \boldsymbol{u}_{N,\mathrm{FP32}}$ at $N = 32, 64, 128, 256, 512, 1024$. We fix viscosity ν to test coarser grids on the high-Reynolds-number flow resolved at $N = 1024$. For *precision*, we use the higher-viscosity, spatially resolved $N = 128$ reference and compare $\boldsymbol{u}^{\mathrm{rec}} = \boldsymbol{u}_{128,\sigma}$ with $\sigma \in \{\mathrm{FP64}, \mathrm{FP32}, \mathrm{FP16}, \mathrm{FP8}, \mathrm{FP4}\}$. Within each comparison, the initial condition, forcing, and recording schedule are fixed, as are the nominal viscosity and time step. FP64 uses native arithmetic and lower formats emulate the same operations with nearest-value rounding after each arithmetic result.

Representation. For *filtering*, we compare $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_M \boldsymbol{u}_{1024,\mathrm{FP32}}$ at $M = 32, 64, 128, 256, 512$. For *quantization*, we compare $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{Q}_q \boldsymbol{u}_{128,\mathrm{FP64}}$, using the same five formats for q as for solver precision. Each uses the corresponding nearest-value rounding rule on the completed FP64 trajectory.

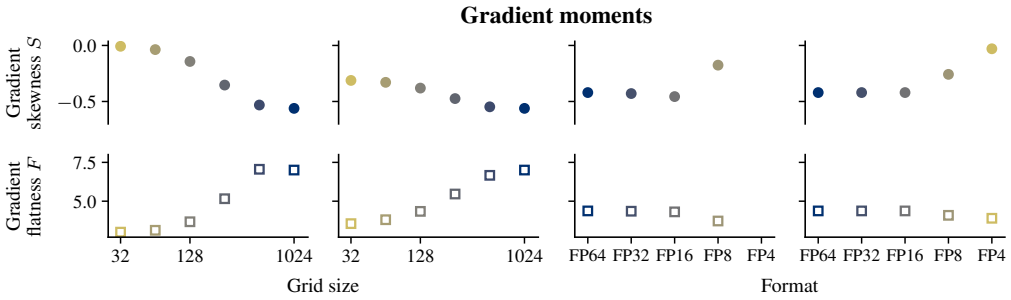
Effects on the recorded motion. Figure 5 shows how the four numerical and representational choices affect both the spatial structure and the temporal evolution of the recorded flow (Appendix D.3, Table 3). The *energy spectrum* shows how energy is distributed across spatial scales, while *enstrophy* captures the strength of the corresponding velocity gradients. Looking at the two together reveals an important difference: filtering removes small-scale gradient content, whereas insufficient resolution and coarse quantization can instead introduce artificial gradient strength. The *gradient moments* provide a more detailed view of these gradients, showing that increased enstrophy does not necessarily recover the asymmetry and heavy tails associated with developed turbulence. The *correlation times* then show how these structures evolve over time. Filtering largely preserves the dynamics of the scales that remain, while insufficient resolution and coarse quantization can alter their evolution. At very low solver precision, the dynamics can even become effectively frozen (Appendix D, Figure 7). Thus, a record may look reasonable according to a spatial diagnostic while still failing to preserve the underlying turbulent motion over time.



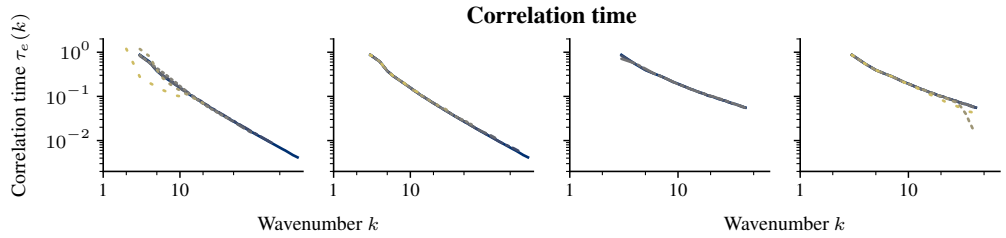
(a) The **energy spectrum** changes with *resolution*: coarse grids accumulate energy near their cutoff, while refinement extends the resolved range. *Filtering* removes fine scales but preserves the retained spectrum. *Precision* has little effect from FP64 to FP16, whereas FP8 distorts the small-scale decay. *Quantization* adds small-scale energy as the record format becomes coarser.



(b) **Enstrophy** increases at coarse *resolution* because energy accumulates near the grid cutoff. *Filtering* removes substantial gradient strength despite retaining most energy. *Precision* changes enstrophy only modestly through FP8. Coarse *quantization* instead raises enstrophy through numerical distortion.



(c) The **gradient moments** change distinctly across the four numerical choices. Increasing *resolution* restores more negative gradient skewness and heavier tails. Stronger *filtering* weakens both signatures. *Precision* preserves them from FP64 to FP16, whereas FP8 weakens them. Coarser *quantization* likewise suppresses gradient asymmetry and heavy tails.



(d) **Correlation times** vary with *resolution* even at shared scales. *Filtering* largely preserves the dynamics of the scales it retains. For *precision*, FP64–FP16 agree, FP8 is frozen, and FP4 produces no evolved record. Coarse *quantization* shortens small-scale correlations.

Figure 5: Numerical modeling and representation of the recorded flow.

5.2 NEURAL REPRESENTATIONS

The numerical study establishes what motion is available in each record. We now ask how neural representations capture that motion. We compare joint fitting, which shares one model across the movie, with independent fitting, which assigns a separate model to each snapshot. Both use SIREN and Gaussian fields (Section 4), with velocity MSE, comparable total parameter counts, and a common stopping rule.

Learning targets. The supplied records follow the four comparisons in Section 5.1. For resolution, precision, and quantization, we apply the same space–time projection to obtain a common observation bandwidth: resolution, $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_{64}\boldsymbol{u}_{N,\mathrm{FP32}}$; precision, $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_{64}\boldsymbol{u}_{128,\sigma}$; quantization, $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_{64}\mathcal{Q}_q\boldsymbol{u}_{128,\mathrm{FP64}}$. For resolution, N ranges over $128, 256, 512, 1024$; for precision, σ ranges over the evolving FP64, FP32, and FP16 records. Quantization uses all five formats for q, with rounding before projection. The filtering comparison varies the observation bandwidth itself: $\boldsymbol{u}^{\mathrm{rec}} = \mathcal{P}_M\boldsymbol{u}_{1024,\mathrm{FP32}}$ for $M = 32, 64, 128$. Each fit is assessed against its supplied movie.

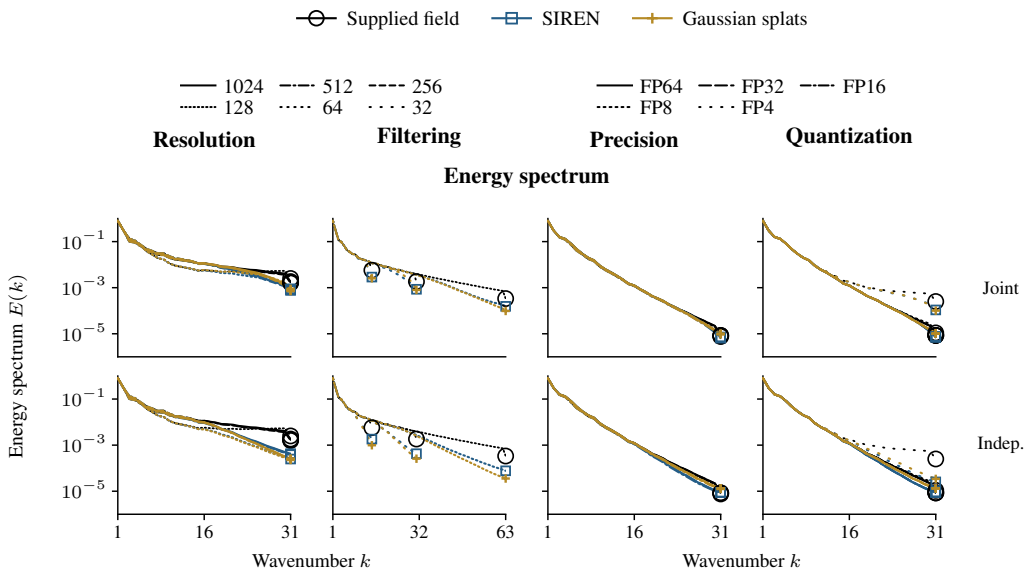
Effects of neural fitting. Figure 6 compares how well the joint and independent fits reproduce both the spatial structure and the temporal evolution of the supplied records (Appendix E.2, Figure 9, Tables 7–10). The *energy spectrum* shows which scales are lost or weakened, while *enstrophy* captures the change in gradient strength. Whether this attenuation is desirable depends on the record: fitting can remove genuine turbulent structure or suppress spurious content introduced by quantization. The *gradient moments* provide a complementary view. Similar enstrophy does not necessarily mean that the underlying gradients are distributed in the same way. SIREN and Gaussian fits differ in skewness and flatness even after matching gradient power across scales (Appendix E.6, Figure 11). The *correlation times* then reveal differences invisible to these spatial diagnostics. Joint fits generally retain fluctuations for longer, whereas independent fits decorrelate more quickly, with neither consistently matching the supplied dynamics better (Appendix E.5, Table 13; Appendix E.6, Figure 10). Taken together, these results show that spatial and temporal fidelity are not interchangeable: comparable frame accuracy can still yield different accuracy of the recorded changes (Appendix E.6, Table 14).

6 DISCUSSION AND KEY TAKEAWAYS

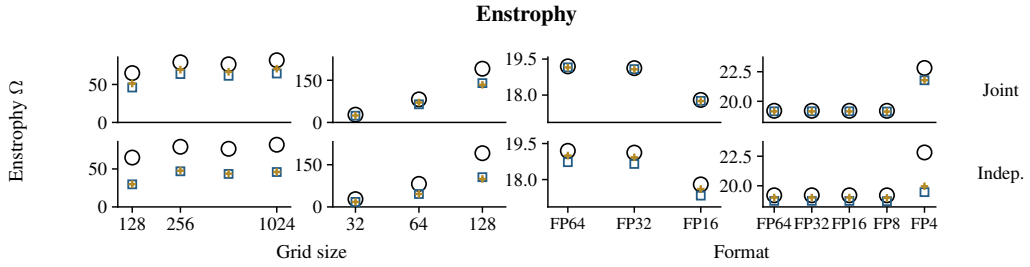
Our results show that spatial fidelity and temporal fidelity are distinct properties of a turbulent representation. This distinction already appears before learning. At fixed physical parameters, insufficient resolution can create excess small-scale energy, distort gradients, and alter correlation times. Space-time filtering instead removes fine scales while largely preserving the temporal behavior of those retained. Low solver precision can preserve a structured field while arresting its evolution, whereas post-hoc quantization can add apparent small-scale content while shortening correlations. Thus, spatial detail alone does not determine whether a record contains faithful turbulent motion.

Neural representations inherit this ambiguity. Both network families attenuate small-scale content, but whether this is beneficial depends on its origin: fitting can remove genuine turbulent gradients from a resolved record, yet suppress numerical distortion in a strongly quantized one. This parallels the mitigation of numerical artifacts observed in neural PDE emulators (Koehler & Thuerey, 2025). Likewise, representations with similar enstrophy can have different gradient statistics, and similar spatial accuracy can coexist with different temporal correlations. The simulation, recording procedure, and learned representation should therefore be viewed as a single information pipeline: each stage determines which spatial and temporal information remains available to the next.

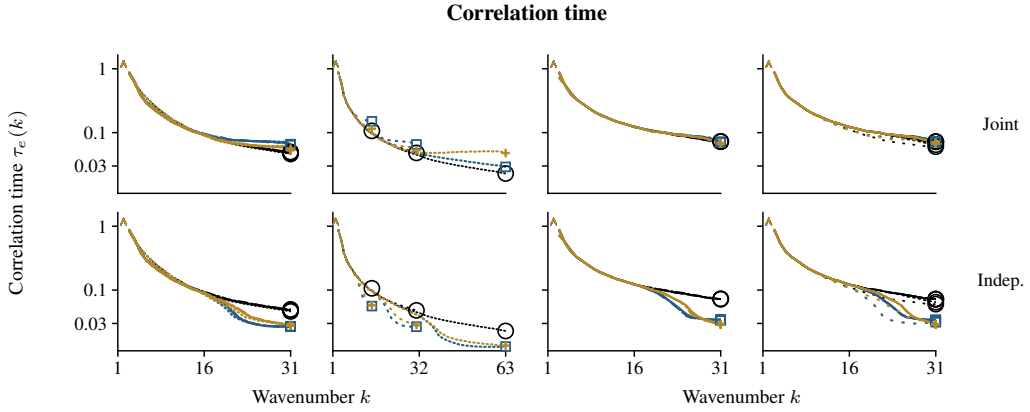
In practice, our results suggest that the architecture and temporal parameter sharing of neural representations should be matched to the gradients and velocity changes relevant to the application. For example, joint Gaussian fits better preserve the record's small-scale temporal correlations than independent fits. These choices become increasingly important as neural representations are used to encode turbulent flows for downstream applications (Du et al., 2024; Guo et al., 2026; Lu & Xiao, 2026). Furthermore, practitioners should account for how numerical and recording choices affect the spatial and temporal information these representations preserve, as this information can shape the turbulent dynamics downstream models learn to reproduce. For instance, independent SIREN fits show a smoothing bias that counteracts excess enstrophy caused by quantization.



(a) The **energy spectrum** retains the energy-containing scales under both *joint* and *independent fitting*. Small-scale energy is generally underestimated, especially by independent fits. Across *resolution* and *filtering*, this removes supplied structure. For coarse *quantization*, it attenuates numerical excess.



(b) **Enstrophy** is generally lower under *independent fitting* than under *joint fitting*. Deficits occur across source *resolutions* and grow with wider *filtering* bandwidths. On the FP4-quantized record, this reduction instead brings enstrophy closer to the unquantized value.



(c) **Correlation times** expose a temporal trade-off. *Independent fits* generally decorrelate too rapidly at small scales, whereas *joint fits* can remain correlated too long. This tendency spans *resolution*, *filtering*, *precision*, and *quantization*.

Figure 6: Joint and independent neural representations of the recorded flow.

7 Related Work

Numerical modeling of turbulence. DNS resolves three-dimensional turbulent motion in space and time, down to the scales at which viscosity dissipates energy (Moin & Mahesh, 1998; Ishihara et al., 2009). Its computational cost motivates a hierarchy of reduced descriptions. As the Reynolds number increases, the separation between large and small scales grows, requiring finer grids and shorter time steps. For many industrial applications, resolving this full range is impractical. Reynolds-averaged Navier–Stokes (RANS) models instead describe the mean flow (Reynolds, 1895), while large-eddy simulation (LES) resolves larger motions and models unresolved scales (Smagorinsky, 1963; Germano et al., 1991). Both require closures because averaging or filtering the nonlinear term in equation 1 introduces stresses that cannot be determined from the retained velocity alone. For velocity components u_i, u_j, the difference $\overline{u_i u_j} - \bar{u}_i \bar{u}_j$ expresses this unresolved contribution, where the bar denotes the chosen average or spatial filter. The closure supplies its effect on the retained motion. A different simplification restricts motion to two spatial dimensions. This reduces the computational burden but also changes the cascade. Without vortex stretching, the inviscid, unforced dynamics conserve both energy and enstrophy. In the classical two-dimensional cascade, energy therefore transfers toward larger scales, while enstrophy transfers toward smaller scales (Boffetta & Ecke, 2012). Two-dimensional turbulence thus offers a simpler setting for studying multiscale motion, with different energy-transfer physics from the three-dimensional flow considered here.

Machine learning for turbulence. Machine learning builds on these reduced approaches, using simulated flows for training and evaluation (Brunton et al., 2020; Duraisamy et al., 2019). What a model learns therefore depends in part on how those flows were computed and recorded. Applications include learned closures for unresolved stresses (Ling et al., 2016) and learned numerical schemes for coarse-grid evolution in two-dimensional turbulence (Kochkov et al., 2021). Super-resolution methods recover spatial detail from coarser observations (Fukami et al., 2019; Fukami & Taira, 2024), and generative approaches synthesize three-dimensional turbulent states (Lienen et al., 2024). Much of this work addresses selected aspects of the full $(3+1)$-dimensional motion resolved by DNS.

Neural representations for turbulence. More recently, methods have begun to generate three-dimensional flow sequences (Du et al., 2024). Neural representations offer one way to encode such sequences. For instance, CoNFiLD encodes flow fields with a conditional neural field (Du et al., 2024). Related neural representations for turbulence have also been explored (Pan et al., 2023; Guo et al., 2026; Shenoy & Frankel, 2026). What these representations preserve, however, depends on their inductive biases: small velocity errors, for example, can coexist with substantial enstrophy loss (Shenoy & Frankel, 2026). These biases act on flows that are themselves shaped by numerical and recording choices, raising the question of how such choices interact with the way space and time are represented and fitted.

8 Conclusion

In this work, we examine how numerical choices shape turbulent motion in space and time, and how neural networks represent the resulting records. This matters because what these representations preserve determines which turbulent dynamics downstream models can learn (Du et al., 2024; Guo et al., 2026; Lu & Xiao, 2026). To the best of our knowledge, this is the first study to trace how choices made during simulation and recording propagate into neural representations of the full $(3+1)$-dimensional motion resolved by DNS. Using massive time-resolved DNS records, we vary resolution, solver precision, filtering, and quantization, and compare joint space–time fitting with independent snapshot fitting. We find that spatial and temporal assessments can disagree for both numerical records and learned representations. The difficulty of representing DNS data therefore extends beyond the cost of generating and storing it. *We therefore call on the machine learning community to treat $(3+1)$-dimensional turbulence as a four-dimensional phenomenon.*

AI USE STATEMENT

In this work, we used generative AI tools for two tasks requiring disclosure: generating synthetic datasets and implementing methods. We used Claude to help set up the code for the simulation environment used to generate the data discussed in Section 5.1. One author reviewed and verified both this code and the resulting simulation record. We also used Codex to implement the neural representations (SIREN and Gaussian splats) and the supporting evaluation code used in Section 5.2. One author reviewed, verified, and tested this code.

We did not use generative AI tools to develop the problem setting or framing; to help develop theoretical models or conceptual frameworks; to formulate mathematical claims; to propose or refine hypotheses; to design or provide feedback on research methodology or experiments; to clean or reformat datasets; or to interpret results. Providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, translation, and qualitative or thematic data analysis were not applicable to this work.

Additionally, we used generative AI tools to identify relevant literature, improve the manuscript's readability, and make cosmetic improvements to figures. We did not use these tools for other tasks for which disclosure is recommended, such as drafting text, suggesting hyperparameters, identifying research gaps, formatting references, or proposing a title.

We have reviewed all AI-assisted work. In addition to the code checks described above, all suggested papers were read by the authors before being cited, and text edits were checked to ensure that the technical content and claims remained unchanged. Figure edits were limited to styling. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix D specifies the physical setup, solver, initial conditions, time stepping, arithmetic formats, recording schedules, filtering, quantization, and diagnostic estimators. Appendix E specifies the learning targets, architectures, parameter budgets, normalization, sampling, optimizer, stopping criterion, initialization, and evaluation procedure.

REFERENCES

G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

Guido Boffetta and Robert E. Ecke. Two-dimensional turbulence. *Annual Review of Fluid Mechanics*, 44:427–451, 2012. doi: 10.1146/annurev-fluid-120710-101240.

Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020. doi: 10.1146/annurev-fluid-010719-060214.

Bita Darvish Rouhani, Nitin Garegrat, Tom Savell, Ankit More, Kyung-Nam Han, Ritchie Zhao, Mathew Hall, Jasmine Klar, Eric Chung, Yuan Yu, Michael Schulte, Ralph Wittig, Ian Bratt, Nigel Stephens, Jelena Milanovic, John Brothers, Pradeep Dubey, Marius Cornea, Alexander Heinecke, Andres Rodriguez, Martin Langhammer, Summer Deng, Maxim Naumov, Paulius Micikevicius, Michael Siu, and Colin Verrilli. OCP microscaling formats (MX) specification. Technical Report Version 1.0, Open Compute Project, September 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15:10416, 2024. doi: 10.1038/s41467-024-54712-1.

Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019. doi: 10.1146/annurev-fluid-010518-040547.

Marie Farge. Wavelet transforms and their applications to turbulence. *Annual Review of Fluid Mechanics*, 24:395–458, 1992. doi: 10.1146/annurev.fl.24.010192.002143.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OmtmcPkkhT.

Kai Fukami and Kunihiko Taira. Single-snapshot machine learning for super-resolution of turbulence. *Journal of Fluid Mechanics*, 1001:A32, 2024. doi: 10.1017/jfm.2024.1136.

Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019. doi: 10.1017/jfm.2019.238.

Massimo Germano, Ugo Piomelli, Parviz Moin, and William H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991. doi: 10.1063/1.857955.

David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991. doi: 10.1145/103162.103163.

Anastasiia Gorbunova, Guillaume Balarac, Léonie Canet, Gregory Eyink, and Vincent Rossetto. Spatio-temporal correlations in three-dimensional homogeneous and isotropic turbulence. *Physics of Fluids*, 33(4):045114, 2021. doi: 10.1063/5.0046677. URL https://arxiv.org/abs/2102.02858.

Junyi Guo, Pan Du, Xiantao Fan, Yahui Li, and Jian-Xun Wang. Conditional neural field for spatial dimension reduction of turbulence data: A comparison study. *Physics of Fluids*, 38:025153, 2026. doi: 10.1063/5.0310238.

IEEE. IEEE standard for floating-point arithmetic. Standard IEEE Std 754-2019, IEEE, 2019. URL https://doi.org/10.1109/IEEESTD.2019.8766229.

Takashi Ishihara, Toshiyuki Gotoh, and Yukio Kaneda. Study of high-Reynolds number isotropic turbulence by direct numerical simulation. *Annual Review of Fluid Mechanics*, 41:165–180, 2009. doi: 10.1146/annurev.fluid.010908.165203.

Ugur Karban, Eduardo Martini, Peter Jordan, Guillaume A. Brès, and Aaron Towne. Solutions to aliasing in time-resolved flow data. *Theoretical and Computational Fluid Dynamics*, 36:887–914, 2022. doi: 10.1007/s00162-022-00630-1.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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. doi: 10.1073/pnas.2101784118.

Felix Koehler and Nils Thuerey. Neural emulator superiority: When machine learning for PDEs surpasses its training data. In *Advances in Neural Information Processing Systems*, volume 38, pp. 165660–165702. Curran Associates, Inc., 2025. doi: 10.52202/085713-5521. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f200119a40846e508954abcd61f5f3fd-Abstract-Conference.html.

Andrey N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Doklady Akademii Nauk SSSR*, 30:299–303, 1941.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3D flow simulation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.01776.

Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016. doi: 10.1017/jfm.2016.615.

Zeyuan Lu and Xiu Xiao. Generation and posterior inference of spatiotemporal turbulent flows using latent neural fields. *Physics of Fluids*, 38:075145, 2026. doi: 10.1063/5.0340524.

Parviz Moin and Krishnan Mahesh. Direct numerical simulation: A tool in turbulence research. *Annual Review of Fluid Mechanics*, 30:539–578, 1998. doi: 10.1146/annurev.fluid.30.1.539.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127.

Shaowu Pan, Steven L. Brunton, and J. Nathan Kutz. Neural Implicit Flow: A mesh-agnostic dimensionality reduction paradigm of spatio-temporal data. *Journal of Machine Learning Research*, 24 (41):1–60, 2023. URL https://www.jmlr.org/papers/v24/22-0365.html.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019. doi: 10.1109/CVPR.2019.00025. URL https://arxiv.org/abs/1901.05103.

Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990. doi: 10.1109/5.58326.

Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

Osborne Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. A*, 186:123–164, 1895. doi: 10.1098/rsta.1895.0004.

Dhanush V. Shenoy and Steven H. Frankel. Gaussian field representations for turbulent flow: Compression, scale separation, and physical fidelity. *Computers & Fluids*, 318:107202, 2026. doi: 10.1016/j.compfluid.2026.107202.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020. URL https://arxiv.org/abs/2006.09661.

Joseph Smagorinsky. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3):99–164, 1963. doi: 10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2.

Katepalli R. Sreenivasan and Robert A. Antonia. The phenomenology of small-scale turbulence. *Annual Review of Fluid Mechanics*, 29:435–472, 1997. doi: 10.1146/annurev.fluid.29.1.435.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020. URL https://arxiv.org/abs/2006.10739.

Hendrik Tennekes. Eulerian and Lagrangian time microscales in isotropic turbulence. *Journal of Fluid Mechanics*, 67(3):561–567, 1975. doi: 10.1017/S0022112075000468.

P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967. doi: 10.1109/TAU.1967.1161901.

Michael Wilczek and Yasuhito Narita. Wave-number–frequency spectrum for turbulence from a random sweeping hypothesis with mean flow. *Physical Review E*, 86:066308, 2012. doi: 10.1103/PhysRevE.86.066308.

A CHARACTERISTICS OF $(3+1)$-DIMENSIONAL TURBULENCE

Turbulence is a state of fluid motion with fluctuations over many spatial scales and time scales. This motion is described by the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$. This field has three components at each position $\boldsymbol{x} = (x_1, x_2, x_3)$ and time t. The four dimensions therefore describe three spatial coordinates and one temporal coordinate. The equations and statistical quantities that follow describe its spatial and temporal structure. We refer to Pope (2000) for a fuller treatment of turbulence theory.

For a fluid of constant density and kinematic viscosity ν, the incompressible Navier–Stokes equations (1) read

$$\underbrace{\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}}_{\text{material acceleration}} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0. \tag{2}$$

Here p is pressure divided by density and $\boldsymbol{f}$ is force per unit mass. The time derivative measures change at a fixed position. The transport term accounts for a fluid parcel moving through a spatially varying field. Their sum is the acceleration experienced by that parcel. Nonlinear transport couples spatial scales, while viscosity smooths velocity gradients. Thus the equation connects **spatial variation and temporal evolution**.

Incompressibility. The condition $\nabla \cdot \boldsymbol{u} = \sum_i \partial u_i / \partial x_i = 0$ means that an infinitesimal fluid volume neither expands nor contracts. Pressure enforces this constraint: taking the divergence of (2) gives

$$\nabla^2 p = -\sum_{i,j} (\partial_i u_j)(\partial_j u_i) + \nabla \cdot \boldsymbol{f}, \qquad \partial_i = \frac{\partial}{\partial x_i}. \tag{3}$$

The root-mean-square divergence $\langle (\nabla \cdot \boldsymbol{u})^2 \rangle^{1/2}$ quantifies departures from this constraint. It vanishes for an incompressible field. Here $\langle a \rangle = |V|^{-1} \int_V a \, d\boldsymbol{x}$ denotes a spatial average over the domain V. An overbar denotes a time average.

Statistical setting and energy balance. *Homogeneity* means that statistics do not depend on position, and *isotropy* that they do not depend on orientation. *Stationarity* means that statistics do not change with the time origin. Consider a zero-mean velocity field in a periodic domain. Its kinetic energy per unit mass is $K = \langle |\boldsymbol{u}|^2 \rangle / 2$. Multiplying (2) by velocity and averaging gives

$$\frac{dK}{dt} = \langle \boldsymbol{u} \cdot \boldsymbol{f} \rangle - \epsilon, \qquad \epsilon = 2\nu \left\langle \sum_{i,j} s_{ij}^2 \right\rangle, \qquad s_{ij} = \tfrac{1}{2}(\partial_j u_i + \partial_i u_j). \tag{4}$$

The symmetric tensor s_{ij} measures local deformation. The quantity ϵ is the rate of viscous energy loss. Nonlinear transport redistributes energy without changing its total in this domain. Forcing supplies energy. In a statistically stationary flow, its mean input balances mean dissipation.

Reynolds number and the range of scales. Let $u' = (2\overline{K}/3)^{1/2}$ be the root-mean-square velocity of one component in isotropic flow, and L the integral length scale, which characterizes the extent of large-scale velocity correlations. Then

$$Re = \frac{u' L}{\nu}, \qquad T_L = \frac{L}{u'}, \qquad \eta = \left(\frac{\nu^3}{\overline{\epsilon}} \right)^{1/4}, \qquad \tau_\eta = \left(\frac{\nu}{\overline{\epsilon}} \right)^{1/2}. \tag{5}$$

The Reynolds number compares inertial transport with viscous smoothing. The turnover time T_L describes the large motions. The Kolmogorov scales η and τ_η characterize viscous dissipation. Large Reynolds numbers permit a broad separation between them. This separation allows motions at many scales to coexist. An alternative measure is the *Taylor-scale Reynolds number* $Re_\lambda = u' \lambda / \nu$, where $\lambda^2 = 15\nu (u')^2 / \overline{\epsilon}$ for isotropic incompressible flow. Both Reynolds numbers express the relative importance of inertia and viscosity, but they use different characteristic length scales.

Energy spectrum. A spatial Fourier expansion writes $\boldsymbol{u}(\boldsymbol{x},t) = \sum_{\boldsymbol{k}} \hat{\boldsymbol{u}}(\boldsymbol{k},t)e^{i\boldsymbol{k}\cdot\boldsymbol{x}}$. A mode of magnitude $k = |\boldsymbol{k}|$ has wavelength $2\pi/k$. Grouping modes in radial shells gives

$$E(k,t) = \frac{1}{2}\sum_{\boldsymbol{k}\in\mathcal{S}_k} |\hat{\boldsymbol{u}}(\boldsymbol{k},t)|^2, \qquad K(t) = \sum_k E(k,t), \tag{6}$$

where $\mathcal{S}_k$ is the shell labeled k and the Fourier normalization preserves spatial mean-square velocity. The continuum spectrum instead satisfies $K = \int_0^\infty E(k)\,dk$. In developed three-dimensional turbulence, energy is supplied at large scales and transferred on average toward smaller scales. Where forcing and viscosity have little direct effect, Kolmogorov theory predicts $E(k) \simeq C_K \bar{\epsilon}^{2/3} k^{-5/3}$, with dimensionless constant C_K (Kolmogorov, 1941; Pope, 2000). This *inertial range* appears only when the large and dissipative scales are sufficiently separated.

Enstrophy. Vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ measures local rotation, and $\Omega = \langle|\boldsymbol{\omega}|^2\rangle/2$ is its enstrophy. For a smooth, periodic, divergence-free field,

$$\Omega = \frac{1}{2}\sum_{\boldsymbol{k}} |\boldsymbol{k}|^2 |\hat{\boldsymbol{u}}(\boldsymbol{k})|^2, \qquad \epsilon = 2\nu\Omega. \tag{7}$$

The factor $|\boldsymbol{k}|^2$ makes fine scales important even when they carry little energy. In continuum notation, $\Omega = \int k^2 E(k)\,dk$. The exact Fourier sum weights each mode by its own wavenumber. For a general periodic velocity field, $\langle|\nabla\boldsymbol{u}|^2\rangle = 2\Omega + \langle(\nabla\cdot\boldsymbol{u})^2\rangle$. The relation $\epsilon = 2\nu\Omega$ therefore relies on incompressibility.

Gradient moments. The longitudinal derivatives $g_i = \partial_i u_i$ have zero spatial mean in a periodic domain. Their normalized third and fourth moments are

$$S_i = \frac{\langle g_i^3\rangle}{\langle g_i^2\rangle^{3/2}}, \qquad F_i = \frac{\langle g_i^4\rangle}{\langle g_i^2\rangle^2}, \qquad S = \tfrac{1}{3}\sum_i S_i, \quad F = \tfrac{1}{3}\sum_i F_i. \tag{8}$$

These moments are first defined through spatial averages. Their stationary values can then be averaged over time. Negative *skewness* indicates a greater contribution from negative gradients to the third moment. *Flatness* above the Gaussian value 3 indicates heavier tails. Developed turbulence typically has negative skewness, often near -0.5, and flatness above 3. The values depend on Reynolds number and retained scales (Ishihara et al., 2009; Sreenivasan & Antonia, 1997).

Correlation time. The preceding quantities describe the spatial field at one time. Temporal correlation describes how long each spatial scale retains its state. Define the shell correlation

$$R(k,\tau) = \mathrm{Re}\sum_{\boldsymbol{k}\in\mathcal{S}_k} \overline{\hat{\boldsymbol{u}}(\boldsymbol{k},t)^*\cdot\hat{\boldsymbol{u}}(\boldsymbol{k},t+\tau)}, \qquad C(k,\tau) = \frac{R(k,\tau)}{R(k,0)}. \tag{9}$$

The star denotes complex conjugation. The overbar averages over the time origin t. For nonzero shell energy, $C(k,0) = 1$. The first crossing of e^{-1} defines the correlation time $\tau_e(k)$. A frozen nonzero field has $C(k,\tau) = 1$ at every lag. An empty shell has no normalized correlation.

These are Eulerian correlations: they describe change at fixed coordinates, not the lifetime of a moving vortex. Large motions can sweep small structures past those coordinates without destroying them. When sweeping dominates, the characteristic time scales as $\tau_e(k) \propto (u'k)^{-1}$ (Tennekes, 1975; Gorbunova et al., 2021).

B DIRECT NUMERICAL SIMULATION (DNS)

Appendix A describes turbulent motion as a velocity field evolving across spatial and temporal scales. Direct numerical simulation (DNS) computes this field from the Navier–Stokes equations without a closure for unresolved motion and resolves the scales of viscous dissipation (Moin & Mahesh, 1998). Section 3 distinguishes the *numerical modeling* that computes this trajectory from the *representation* that retains it as a record.

B.1 NUMERICAL MODELING

From the equations to a solver. The physical problem specifies the domain, boundary conditions, viscosity ν, forcing $\boldsymbol{f}$, and initial velocity $\boldsymbol{u}(\boldsymbol{x}, 0)$. Consider a periodic cube with side length L_0. Opposite faces connect, which permits a Fourier representation of the field. With **spatial resolution** N, each direction contains N grid points and the grid spacing is $\Delta x = L_0/N$. The truncated field $\boldsymbol{u}_N$ uses a finite set of Fourier modes $\mathcal{K}_N$. Projecting (2) onto these modes gives, for $\boldsymbol{k} \neq 0$,

$$\frac{d\hat{\boldsymbol{u}}_N(\boldsymbol{k},t)}{dt} = -\mathsf{P}_{\boldsymbol{k}}\widehat{(\boldsymbol{u}_N \cdot \nabla)\boldsymbol{u}_N}(\boldsymbol{k},t) - \nu|\boldsymbol{k}|^2\hat{\boldsymbol{u}}_N(\boldsymbol{k},t) + \mathsf{P}_{\boldsymbol{k}}\widehat{\boldsymbol{f}}(\boldsymbol{k},t), \qquad \mathsf{P}_{\boldsymbol{k}} = I - \frac{\boldsymbol{k}\boldsymbol{k}^{\mathsf{T}}}{|\boldsymbol{k}|^2}. \quad (10)$$

Here I is the identity matrix. The projection $\mathsf{P}_{\boldsymbol{k}}$ enforces $\boldsymbol{k} \cdot \hat{\boldsymbol{u}}_N = 0$ and thus eliminates pressure from the evolution equation. The zero mode describes the spatial mean and evolves with the mean forcing.

Equation (10) defines a finite system of ordinary differential equations. A *pseudo-spectral* method evaluates derivatives in Fourier space and nonlinear products in physical space. These products generate modes outside $\mathcal{K}_N$. On a finite grid, unresolved modes can reappear at false lower wavenumbers. This effect is *aliasing*. Dealiasing prevents it through padding or a restriction of the retained modes. The dealiasing rule determines $k_{\max}$, the largest retained shell wavenumber. Thus N alone does not specify the resolved band.

Time stepping and arithmetic. A time integrator advances the retained coefficients over intervals Δt. Let U^n collect the numerical state at solver time $n\Delta t$. Runge–Kutta methods approximate evolution through intermediate stages. An integrating factor can apply the viscous decay $e^{-\nu|\boldsymbol{k}|^2\Delta t}$ directly.

The computer evaluates each stage with finite arithmetic. A floating-point format provides a finite set of real numbers $\mathbb{F}_\sigma$. Away from zero, a binary number has the form $\pm m\, 2^e$, with $1 \leq m < 2$. The significant digits in m determine the spacing of nearby values. The allowed exponents determine the range of magnitudes (Goldberg, 1991). Let $r_\sigma(a)$ round a real value a to the nearest available value. Ties, very small values, and values beyond the finite range require specified rules.

With **arithmetic precision** σ, one solver step has the form

$$U^{n+1} = \Phi^{\Delta t}_{N,\sigma}(U^n). \quad (11)$$

The map $\Phi^{\Delta t}_{N,\sigma}$ includes the nonlinear term, forcing, viscous evolution, and the incompressibility projection. Rounding can enter its intermediate stages, transforms, constants, and state. Each rounded result can affect every later step. Rounding only the completed update defines a different map. The meaning of σ therefore includes the format, rounding rules, and the operations to which those rules apply.

From solver states to a record. Repeated application of (11) produces the computed trajectory. Saving selected states produces the record $\boldsymbol{u}_{N,\sigma}$. For J equally spaced snapshots, write $t_j = t_0 + j\Delta t_{\text{snap}}$, with $j = 0, \ldots, J-1$. The snapshot interval Δt_{snap} differs from the solver step Δt. The observation span is $(J-1)\Delta t_{\text{snap}}$. Reproducing the trajectory requires the initial condition, forcing, time integrator, dealiasing rule, and arithmetic rules. Reproducing the record also requires the discarded transient, snapshot times, stored fields, and encoding.

The ratios $k_{\max}\eta$ and $\Delta t/\tau_\eta$ compare the solver's resolution with the dissipative scales defined in Appendix A. The advective Courant number $\Delta t \max_{\boldsymbol{x}} \sum_i |u_i|/\Delta x$ also constrains time-step stability. Snapshot spacing is described by $\Delta t_{\text{snap}}/\tau_\eta$ and, at each shell, $\Delta t_{\text{snap}}/\tau_e(k)$. The observation span in units of T_L describes coverage of large-scale motion. Measured u', $\bar{\epsilon}$, L, and Re_λ characterize the resulting flow.

B.2 REPRESENTATION

A record retains a finite description of the computed trajectory. Its representation determines which scales and velocity values remain. Filtering selects the scales, while quantization selects the values.

Selecting spatial and temporal scales. Resampling without aliasing requires the removal of unresolved frequencies. A spatial cutoff limits the retained wavenumbers. A temporal cutoff limits how rapidly their coefficients change. Both are required when space and time are resampled.

Sweeping provides a physical relation between the two cutoffs. Appendix A describes how large motions sweep smaller structures past fixed positions. The temporal width of a shell then scales approximately as $u'k$ (Wilczek & Narita, 2012). Let $k_0 = 2\pi/L_0$ be the fundamental spatial wavenumber. Let M set the number of samples per spatial direction and in time. The operator $\mathcal{P}_M$ uses the coupled cutoffs

$$k_c(M) = k_0\left(\frac{M}{2} - 1\right), \qquad \omega_c(M) = \alpha u' k_0 \frac{M}{2}, \qquad \Delta t_M = \frac{\pi}{\omega_c(M)}. \tag{12}$$

Here ω is angular temporal frequency. The dimensionless constant α sets the temporal bandwidth for a chosen energy-retention level near the spatial cutoff.

Choosing the temporal interval. The physical domain is periodic in space, but a finite observation interval has no equivalent temporal boundary condition. A discrete Fourier transform repeats each transformed interval periodically. If its endpoint frames differ, their artificial join adds high-frequency content. A four-dimensional FFT therefore imposes a temporal join when it is applied directly to a short movie.

The temporal transform uses the full record before evaluating a shorter interior window. This moves the join away from the selected times without removing all boundary influence. For identical intervals and cutoffs, a four-dimensional FFT and separate spatial and temporal transforms are mathematically equivalent. Their cutoffs remain distinct because $\alpha u'$ converts wavenumber to frequency in (12).

Filtering before resampling. For the full record, define $\omega_\ell = 2\pi\ell/(J\Delta t_{\mathrm{snap}})$, where ℓ indexes signed frequency bins. The temporal Fourier coefficients and the filtered field are

$$\widetilde{\boldsymbol{u}}_{N,\sigma}(\boldsymbol{k}, \omega_\ell) = \frac{1}{J}\sum_{j=0}^{J-1} \hat{\boldsymbol{u}}_{N,\sigma}(\boldsymbol{k}, t_j) e^{-i\omega_\ell (t_j - t_0)},$$

$$(\mathcal{P}_M \boldsymbol{u}_{N,\sigma})(\boldsymbol{x}, t) = \sum_{\substack{\boldsymbol{k}\in\mathcal{K}_N \\ |\boldsymbol{k}|\le k_c(M)}} \sum_{|\omega_\ell| < \omega_c(M)} \widetilde{\boldsymbol{u}}_{N,\sigma}(\boldsymbol{k}, \omega_\ell) e^{i\boldsymbol{k}\cdot\boldsymbol{x} + i\omega_\ell (t - t_0)}. \tag{13}$$

The retained modes lie below the Nyquist limits $\pi M/L_0$ in space and $\pi/\Delta t_M$ in time. Applying the cutoffs before resampling prevents discarded modes from appearing as false larger or slower motion (Karban et al., 2022).

Evaluating (13) at M^3 positions and M times gives an $M \times M \times M \times M \times 3$ array. Its axes represent time, three spatial coordinates, and velocity components. For fixed α, u', and L_0, the window length $T_w = M\Delta t_M = L_0/(\alpha u')$ is independent of M. The half-open window spans $(M-1)\Delta t_M$ between its endpoint samples.

Quantizing the retained values. Quantization applies the rounding map to each retained velocity component:

$$(\mathcal{Q}_q \boldsymbol{u})_i(\boldsymbol{x}, t) = r_q\big(u_i(\boldsymbol{x}, t)\big), \qquad i = 1, 2, 3. \tag{14}$$

The format q specifies saved values independently of solver precision σ. During evolution, rounded values affect later updates. During recording, they do not feed back into the trajectory.

When filtering precedes quantization, the represented record is $\boldsymbol{u}^{\mathrm{rec}}_{N,\sigma,M,q} = \mathcal{Q}_q \mathcal{P}_M \boldsymbol{u}_{N,\sigma}$. The order must be specified because $\mathcal{Q}_q\mathcal{P}_M\boldsymbol{u}_{N,\sigma}$ and $\mathcal{P}_M\mathcal{Q}_q\boldsymbol{u}_{N,\sigma}$ generally differ. Quantization can add variations outside the retained band. Filtering afterward removes their out-of-band part. Quantizing afterward retains the full rounding distortion in the saved values.

C NEURAL REPRESENTATIONS

Appendix B leaves us with a recorded velocity field $\boldsymbol{u}^{\mathrm{rec}}$, sampled at discrete spatial locations and times. A neural representation replaces this collection of samples with a parameterized function. We first describe this approximation, then introduce two ways of sharing parameters across time, and finally define the two representation families used in our experiments.

C.1 FROM VELOCITY SAMPLES TO A FUNCTION

Let $\boldsymbol{u}^{\mathrm{rec}}(\boldsymbol{x}_a, t_j)$ denote the recorded velocity at spatial position $\boldsymbol{x}_a$ and time t_j. The record contains n_x spatial positions and n_t time points. For an M^4 record, $n_x = M^3$ and $n_t = M$. A neural representation maps each space–time coordinate to the three velocity components,

$$f(\boldsymbol{x}_a, t_j) = \big(f_1(\boldsymbol{x}_a, t_j), f_2(\boldsymbol{x}_a, t_j), f_3(\boldsymbol{x}_a, t_j)\big) \approx \boldsymbol{u}^{\mathrm{rec}}(\boldsymbol{x}_a, t_j). \tag{15}$$

Holding t_j fixed and varying $\boldsymbol{x}_a$ reconstructs a three-dimensional snapshot of the flow. Holding $\boldsymbol{x}_a$ fixed and varying t_j instead gives the velocity history at a single spatial location.

The function is determined by a set of learned scalar parameters. For a multilayer network, these are the network weights and biases. For a Gaussian field, they additionally include the locations, shapes, and velocity coefficients of the basis functions. The architecture determines how these parameters are used to map coordinates to velocity values.

Importantly, the full record is available during fitting. The task is therefore one of *reconstruction*, not prediction. Prediction would require estimating states that were not observed during training. Here, evaluating a fitted representation simply returns an approximation to a recorded state; it does not advance an initial condition through the Navier–Stokes equations.

Fitting recorded velocities. Coordinate-based signal representations are commonly fitted by minimizing mean squared error (MSE) (Sitzmann et al., 2020). Giving equal weight to all positions, times, and velocity components gives

$$\mathcal{L}(f) = \frac{1}{3n_x n_t} \sum_{j=0}^{n_t-1} \sum_{a=1}^{n_x} \big\| f(\boldsymbol{x}_a, t_j) - \boldsymbol{u}^{\mathrm{rec}}(\boldsymbol{x}_a, t_j) \big\|_2^2. \tag{16}$$

Training adjusts the representation parameters to reduce this average squared velocity error.

Let $e = f - \boldsymbol{u}^{\mathrm{rec}}$ denote the reconstruction error. On a uniform spatial grid, Parseval's identity gives $\mathcal{L}(f) = (3n_t)^{-1} \sum_j \sum_{\boldsymbol{k}} |\widehat{e}(\boldsymbol{k}, t_j)|^2$ when the spatial Fourier transform is normalized by $1/n_x$. In this sense, MSE measures the total energy of the reconstruction error across spatial modes.

C.2 SHARING PARAMETERS ACROSS TIME

The same samples and the same reconstruction loss can be paired with different ways of distributing parameters across time. Let B denote the total number of stored scalar parameters available for the complete movie, and let $|\theta|$ denote the size of a parameter set. At a fixed numerical precision, B also determines the storage budget.

Joint fitting. Joint fitting uses a single function $f_\theta(\boldsymbol{x}, t)$ that takes both space and time as input. The same parameters are shared across every frame, with $|\theta| \le B$. Changing t changes the represented spatial field, but all recorded times constrain the same parameter set.

Independent fitting. Independent fitting assigns a separate spatial function $f_{\theta_j}(\boldsymbol{x})$ to each recorded time. The total parameter budget is shared across these models, $\sum_j |\theta_j| \le B$. With an approximately uniform allocation, each frame therefore receives about B/n_t parameters. In this case, the frame index selects an entirely separate parameter set rather than entering the function as a coordinate.

These two procedures differ only in how information is shared across the record. Independent fitting shares no parameters between frames. Joint fitting treats the entire movie as a single function of space and time.

C.3 TWO REPRESENTATION FAMILIES

The way parameters are shared across time is separate from the choice of function family. Continuous representations can be built from a variety of architectures, e.g., multilayer decoders (Park et al., 2019), sinusoidal networks (Sitzmann et al., 2020), Fourier-feature networks (Tancik et al., 2020), multiplicative-filter networks (Fathony et al., 2021), multiresolution feature grids (Müller et al., 2022), or Gaussian basis-function networks (Poggio & Girosi, 1990). We focus on SIRENs and additive Gaussian fields because they distribute their representational capacity in fundamentally different ways. SIREN uses a single coordinate network across the full domain, whereas the Gaussian field constructs the velocity field from localized, adaptive primitives.

SIREN. A SIREN alternates affine transformations with sinusoidal activations (Sitzmann et al., 2020). Its hidden layers take the form

$$h_{\ell+1} = \sin(W_\ell h_\ell + b_\ell), \tag{17}$$

followed by a final affine layer that outputs the three velocity components. For joint fitting, the network receives the four coordinates $(\boldsymbol{x}, t)$. Independent models receive only the three spatial coordinates $\boldsymbol{x}$.

Additive Gaussian fields. Gaussian splatting represents a field using learned Gaussian primitives (Kerbl et al., 2023). We adopt the same basic idea for velocity fields and use the additive form

$$f_j^{\mathrm{G}}(\xi) = \sum_{r=1}^{n_{\mathrm{G}}} \mathbf{a}_{rj} \exp\!\left[-\tfrac{1}{2}\|L_{rj}(\xi - \boldsymbol{\mu}_{rj})\|_2^2\right] w_{rj}(\xi - \boldsymbol{\mu}_{rj}), \tag{18}$$

where $\boldsymbol{\mu}_{rj}$ specifies the primitive center, L_{rj} its anisotropic shape, $\mathbf{a}_{rj} \in \mathbb{R}^3$ its velocity coefficient, and w_{rj} a compact taper. The frame index is included here to allow these attributes to vary when required by the fitting procedure. The primitive contributions are summed directly, rather than combined through the opacity compositing used in image rendering. The resulting model can be viewed as a generalized Gaussian basis-function network with a three-component linear output (Poggio & Girosi, 1990).

D EXPERIMENTS: NUMERICAL MODELING AND REPRESENTATION

Section 5.1 considers four ways in which a velocity record can change. Resolution N and arithmetic precision σ affect how the trajectory is computed, whereas filtering $\mathcal{P}_M$ and quantization $\mathcal{Q}_q$ act on the record after the simulation. Here we describe these experiments using the solver and recording notation from Appendix B, and report the measurements and storage requirements for all records in Figure 5.

D.1 NUMERICAL MODELING

Physical problem and spatial discretization. We solve (2) on $[0, 2\pi)^3$ with periodic boundaries and zero mean velocity. Large-scale forcing sustains the turbulent motion. After every full time step, we rescale the Fourier coefficients in shells $\mathrm{round}(|\boldsymbol{k}|) = 1$ and 2 so that their shell energies are 0.8 and 0.3, respectively. The forcing is deterministic and introduces no preferred direction.

On each N^3 grid, the nonlinear term is evaluated in rotational form, $\mathsf{P}_{\boldsymbol{k}}\widehat{\boldsymbol{u} \times \boldsymbol{\omega}}$, which is equivalent to the projected advective term in (10). Products are formed in physical space. We retain modes satisfying $|k_i| \le h_N = \lfloor N/3 \rfloor$ in each direction and apply the divergence-free projection to the nonlinear term. The retained region is therefore a cube in Fourier space. We report $k_{\max} = h_N$, the radius of the largest complete sphere contained in this cube. Corner modes extend beyond this radius and are still included in the scalar statistics.

Time integration. Let $G(U)$ denote the dealiased, projected nonlinear term and let $H = e^{-\nu|\boldsymbol{k}|^2\Delta t/2}$ act independently on each Fourier mode. The third-order integrating-factor Runge–Kutta update is

$$\begin{aligned} U_a &= H^2U^n + \Delta t H^2 G(U^n), \\ U_b &= HU^n + \tfrac{1}{4}\Delta t HG(U^n) + \tfrac{1}{4}\Delta t H^{-1}G(U_a), \\ U_* &= H^2U^n + \tfrac{1}{6}\Delta t H^2G(U^n) + \tfrac{1}{6}\Delta t G(U_a) + \tfrac{2}{3}\Delta t HG(U_b). \end{aligned} \tag{19}$$

We then rescale the forced shells and remove the zero mode to obtain U^{n+1}. Table 1 summarizes the two physical setups. Each comparison uses one initial realization for each setting.

Table 1: Physical and recording setup.

Quantity	Resolution comparison	Precision comparison
Grid size N	$32, 64, 128, 256, 512, 1024$	128
Solver precision σ	FP32	FP64, FP32, FP16, FP8, FP4
Viscosity ν	0.00065	0.010421385668748454
Nominal step Δt	0.00040015018141313906	0.004701911987471931
Steps between snapshots	6	2
Snapshot interval Δt_{snap}	0.0024009010884788345	0.009403823974943862
Scheduled snapshots J	7,562	2,362
First-to-last span $(J-1)\Delta t_{\mathrm{snap}}$	18.15321312999	22.20242840484
Update implementation	Native FP32	Compensated, format-rounded

Resolution. The existing 1024^3 simulation provides the reference trajectory. Its initial random solenoidal field was generated with seed 20240617, with a spectrum proportional to $k^4e^{-2(k/2)^2}$ and initial $u' = 1$. The field was evolved for about eight large-eddy turnover times before recording began. The first archived state defines time zero for the resolution comparison.

For every smaller N, we take that same state and retain the modes satisfying $|k_i| \le h_N$. Because the stored forward FFT is unnormalized, the coefficients are multiplied by $(N/1024)^3$ when transferred to the smaller grid. This accounts for the FFT convention while preserving the physical mode amplitudes. We do not renormalize the energy or discard an additional transient. All simulations

use the same native FP32 update, viscosity, forcing, time step, and snapshot times. Only N and the corresponding discrete operators change.

The measured Reynolds number can therefore differ even though the physical inputs are fixed. The coarse-grid results should be interpreted as outcomes of under-resolved calculations, not as additional fully resolved DNS references.

Precision. The precision comparison begins from a single archived, equilibrated 128^3 field at the higher viscosity listed in Table 1. Every format receives the same initial field after rounding by r_σ. For the formats that evolve successfully, we run $2{,}048$ nominal warm-up steps, reset the recording clock to zero, and then save states at the intervals listed above. The formats themselves are defined in Table 2.

All formats use an increment form of (19) with compensated accumulation. A second array stores the accumulation residual between updates. It uses the same format as the velocity state and is rescaled together with that state during forcing. This implementation is separate from the native FP32 solver used in the resolution study. The precision comparison therefore varies σ within one solver implementation, while the resolution comparison varies N within the other.

Each real addition, subtraction, multiplication, division, and square root is followed by r_σ. Complex multiplication uses four rounded real products and two rounded sums. Fourier transforms use radix-two butterflies with rounding after every butterfly, and the forward transform divides by two at each stage. Reductions use pairwise rounded sums. Wavevectors, time-step coefficients, and forcing targets are rounded as well. Exponential factors and Fourier phase constants are evaluated in binary64 and then rounded. The compensated update uses `expm1` to form small viscous increments. Binary64 is used to emulate the lower-precision formats.

D.2 Representation

Floating-point values and recorded velocities. The same map r is used for arithmetic rounding and for encoding the recorded values. All conversions round to the nearest representable value, with ties to even, and retain subnormal values. FP64, FP32, and FP16 use the standard binary IEEE formats (IEEE, 2019). For FP8 E4M3 and FP4 E2M1, we use finite-value encodings (Darvish Rouhani et al., 2023) and saturate values outside their finite range. These choices are part of the experiment.

Table 2: Floating-point formats used for computation and recording.

Label	Encoding	Bits per value	Largest finite magnitude	Overflow rule
FP64	binary64	64	1.798×10^{308}	IEEE infinity
FP32	binary32	32	3.403×10^{38}	IEEE infinity
FP16	binary16	16	$65{,}504$	IEEE infinity
FP8	E4M3	8	448	Finite saturation
FP4	E2M1	4	6	Finite saturation

At each saved time in the precision experiment, we reconstruct the physical velocity from the evolved Fourier state using a common binary64 inverse transform. Each velocity component is then encoded in the corresponding format σ. The recorded precision experiment therefore includes both the effect of arithmetic during the simulation and the final encoding of the output.

The quantization experiment isolates the second of these effects. We apply $\mathcal{Q}_q$ once to the completed FP64 trajectory using each of the same five formats. FP64 quantization is a bitwise no-op and therefore shares the reference files. The low-precision encoders round directly from binary64, avoiding an intermediate FP32 conversion near rounding boundaries. Every encoded archive was decoded and checked during writing.

Space–time filtering. The filtering experiment applies $\mathcal{P}_M$ from (13) to $\boldsymbol{u}_{1024,\mathrm{FP32}}$ for $M = 32, 64, 128, 256, 512$. The spatial mask retains $|\boldsymbol{k}| \le M/2 - 1$, while the temporal cutoff is $|\omega| < \alpha u' M/2$. We calibrate α once from the reference trajectory and use the same value for every M.

The calibration is chosen to retain **99% of the temporal spectral energy** in a pooled, normalized shell-frequency distribution. This should not be interpreted as retaining 99% of the total DNS energy at every M. Frequency widths are estimated from up to 512 sampled modes per shell, selected with seed 20260751. Welch spectra (Welch, 1967) use Hann windows of 1,024 samples with 50% overlap for shells 1–63, and windows of 512 samples for shells $80, 100, 120, 127$. For the higher shells, we use two 1,024-frame intervals starting at frames 1,024 and 5,514. The width estimate removes the spectral broadening introduced by the window.

Before estimating the widths, we linearly interpolate across $|\omega| < 2\Delta\omega$, where $\Delta\omega$ is the Welch frequency spacing. The pooled distribution contains shells 16–63 together with the four higher shells, with each shell rescaled by its measured width. Its two-sided 99% quantile is $x_* = 2.455$. Taking the largest interpolated width-to-wavenumber ratio at $k = 16, 32, 64, 128$ gives $c_{\mathrm{edge}} = 1.03358067410$, and therefore

$$\alpha = \frac{x_* c_{\mathrm{edge}}}{u'} = 2.31005299598, \qquad u' = 1.09843391443, \qquad \alpha u' = 2.53744055491. \tag{20}$$

For each retained spatial mode, we transform the complete 7,562-frame time series, keep only the selected frequency bins, and evaluate the resulting Fourier series at $t_j = 9.07780701554 + (j - M/2)\Delta t_M$, where $\Delta t_M = \pi/(\alpha u' M/2)$. All five filtered records cover the same half-open interval $[7.83971194075, 10.31590209032)$, with duration $T_w = 2.47619014957$.

Temporal transforms and evaluation use binary64 arithmetic. For $M \le 256$, the construction stores intermediate coefficients in complex64 and outputs three FP32 velocity components at each of the M^4 coordinates. For $M = 512$, the same projection is evaluated using streamed matrix products. Direct Fourier-series checks confirm the numerical evaluation.

D.3 Measurements and numerical outcomes

Common observation window. The unfiltered records are measured over the same nominal time interval as the filtered movies. This interval contains 1,031 native snapshots for the resolution comparison and 263 snapshots for the precision and quantization comparisons.

For these records, spatial statistics are computed from 16 evenly spaced snapshots. The 1024^3 reference uses 81 snapshots for its energy spectrum, 16 for the gradient moments, and all 1,031 recorded dissipation values for enstrophy. For the filtered records, all M frames are used. At $M = 256, 512$, cached gradient moments were checked against the common derivative operator before being reused. All reported means therefore refer to this finite observation window and to a single realization.

Spatial and temporal estimators. Spatial derivatives are evaluated spectrally in binary64 using the actual stored fields. Nyquist derivative multipliers are set to zero, as required for the real trigonometric interpolant. We apply neither smoothing nor a divergence-free projection before computing the diagnostics. Enstrophy is obtained from the curl, while S and F are normalized separately within each direction and frame before averaging.

For the reference simulation, enstrophy is computed as $\bar{\epsilon}/(2\nu)$. A direct Fourier-space check on the 16 frames used for the gradient moments confirms negligible divergence. Spectra are shown only over their complete-shell range, whereas the scalar statistics include all stored modes.

Correlation times use all available samples in the observation window and up to 512 Fourier modes per shell, chosen using seed 20260910. We average the real mode products over all valid time pairs, without subtracting the mean or applying a taper, and normalize by the zero-lag value. The first crossing of e^{-1} is found by linear interpolation, considering lags up to half the record length. Shells without a measured crossing are omitted from the figure. Table 3 reports two representative shells, while the accompanying data include every measured $E(k)$ and $\tau_e(k)$.

Incompressibility. To compare fields with different gradient strengths, we also report

$$d_\nabla = \left[\frac{\overline{\langle (\nabla \cdot \boldsymbol{u})^2 \rangle}}{2\overline{\Omega} + \overline{\langle (\nabla \cdot \boldsymbol{u})^2 \rangle}} \right]^{1/2}. \tag{21}$$

Table 3: Measured records and stored payloads underlying Figure 5.

Record	$\overline{K}$	$\overline{\Omega}$	$\overline{S}$	$\overline{F}$	d_∇	$\tau_e(8)$	$\tau_e(24)$	TB
Resolution								
$N = 1024$	1.795	323.20	-0.562	7.005	2.91×10^{-6}	0.206	0.058	28.954457
$N = 512$	1.734	291.14	-0.532	7.055	2.56×10^{-6}	0.213	0.059	3.608717
$N = 256$	1.824	330.24	-0.354	5.160	2.53×10^{-6}	0.203	0.058	0.456393
$N = 128$	1.771	314.76	-0.143	3.676	2.61×10^{-6}	0.235	0.056	0.056384
$N = 64$	2.235	448.19	-0.037	3.126	5.99×10^{-6}	0.171	—	0.007383
$N = 32$	8.695	828.27	-0.008	3.007	7.39×10^{-4}	0.112	—	0.000880
Filtering								
$M = 512$	1.795	320.99	-0.548	6.666	2.94×10^{-6}	0.206	0.058	0.824634
$M = 256$	1.794	290.12	-0.474	5.462	3.09×10^{-6}	0.207	0.058	0.051540
$M = 128$	1.780	191.02	-0.381	4.344	3.78×10^{-6}	0.207	0.058	0.003221
$M = 64$	1.722	81.89	-0.329	3.807	5.70×10^{-6}	0.207	0.060	0.000201
$M = 32$	1.609	27.60	-0.312	3.555	9.61×10^{-6}	0.209	—	0.000013
Precision								
FP64	1.617	19.25	-0.420	4.374	2.91×10^{-15}	0.245	0.087	0.118883
FP32	1.612	19.17	-0.429	4.355	2.94×10^{-7}	0.244	0.087	0.059442
FP16	1.572	17.84	-0.456	4.313	3.26×10^{-3}	0.246	0.090	0.029721
FP8	1.601	21.47	-0.176	3.724	2.76×10^{-1}	∞	∞	0.014860
FP4	No evolved record							—
Quantization								
FP64	1.617	19.25	-0.420	4.374	2.91×10^{-15}	0.245	0.087	0.118883
FP32	1.617	19.25	-0.420	4.374	2.69×10^{-7}	0.245	0.087	0.059442
FP16	1.617	19.25	-0.420	4.374	2.21×10^{-3}	0.245	0.087	0.029721
FP8	1.616	22.14	-0.258	4.090	2.43×10^{-1}	0.245	0.085	0.014860
FP4	1.631	74.82	-0.030	3.900	4.77×10^{-1}	0.243	0.069	0.007430

By the periodic identity in Appendix A, the denominator is $\overline{\langle|\nabla \boldsymbol{u}|^2\rangle}$. Values close to zero mean that divergence is small relative to the overall velocity gradients. This is a check of one physical constraint; it does not by itself show that the flow is resolved or statistically stationary.

All quantities use the dimensionless units of the simulation. The storage column refers to the full archived record, not only to the common observation window. A dash indicates that the corresponding shell or record is unavailable.

The FP8 precision field is a verified fixed point, so its nonzero shells have $\tau_e = \infty$, although its spatial statistics are still well defined. The FP4 precision simulation cannot initialize because both the nominal time step and the viscosity round to zero, and therefore produces no evolving record. Quantization is different: all five formats yield finite recorded fields.

Flow scales and resolution checks. Table 4 reports the usual flow and sampling scales for the evolving records. We compute $L = (3\pi/4\overline{K})\sum_{k\geq 1}\overline{E}(k)/k$ and use the definitions in (5). When divergence is nonzero, we define the strain-based dissipation consistently as $\overline{\epsilon} = 2\nu[\overline{\Omega} + \overline{\langle(\nabla\cdot\boldsymbol{u})^2\rangle}]$.

The table uses window-averaged quantities. For strongly distorted records, these values should be regarded as diagnostic scale estimates rather than as evidence of developed turbulence. In particular, $k_{\max}\eta < 1$ for $N \leq 512$, so using the same forcing and viscosity does not make the coarse simulations spatially resolved.

Table 4: Flow scales estimated on the observation window.

Record	u'	$\overline{\epsilon}$	L	Re_λ	$k_{\max}\eta$	$\Delta t_{\mathrm{snap}}/\tau_\eta$	$(J-1)\Delta t_{\mathrm{snap}}/T_L$
$N = 1024$	1.094	0.4202	1.393	280.5	1.724	0.061	14.26
$N = 512$	1.075	0.3785	1.426	285.5	0.882	0.058	13.68
$N = 256$	1.103	0.4293	1.379	281.9	0.427	0.062	14.51
$N = 128$	1.087	0.4092	1.411	280.5	0.214	0.060	13.98
$N = 64$	1.221	0.5826	1.096	296.5	0.098	0.072	20.22
$N = 32$	2.408	1.0767	0.487	848.6	0.040	0.098	89.77
$N = 128$, FP64	1.038	0.4011	1.562	64.6	1.721	0.058	14.76
$N = 128$, FP32	1.037	0.3995	1.565	64.5	1.723	0.058	14.71
$N = 128$, FP16	1.024	0.3719	1.590	65.2	1.754	0.056	14.29

All resolution simulations remain finite, although the $N = 32$ run grows strongly. Over the full monitored trajectory, its maximum divergence-to-gradient ratio is 0.02775, exceeding the campaign threshold of 10^{-3}. The smaller value reported in Table 3 refers only to the central observation window. All runs with $N \geq 64$ pass the divergence check.

Differences between grids over this finite window do not establish monotone convergence. FP16 evolution remains finite and time dependent, but develops measurable divergence. For FP8, three update checks leave both the state and the compensation array bitwise unchanged. The full frozen archive was therefore generated by repeating this verified state at the scheduled times.

These results apply to the particular unscaled arithmetic algorithm used here. They do not imply that a different scaling strategy or mixed-precision solver would behave in the same way.

D.4 Storage and reproducibility

What is stored. The resolution archives contain only the retained real-FFT coefficients. Each snapshot is stored as a complex64 array with shape $3 \times (2h_N + 1) \times (2h_N + 1) \times (h_N + 1)$. Since each complex value contains two four-byte real components, the payload of one complete record is $24J(2h_N + 1)^2(h_N + 1)$ bytes. The 1024^3 archive alone occupies 28.954457 decimal TB. This representation avoids storing zeros and redundant Hermitian coefficients without introducing any further quantization of the retained modes.

The precision and quantization archives instead store all three physical velocity components at every grid point. For a format with b bits per value, the payload is $3JN^3b/8$ bytes. FP4 stores two values per byte. These are the actual written payloads for FP16 and FP8 as well, even though the lower formats are emulated using binary64 working arrays. Solver working memory and archived record size are therefore separate quantities.

The tables use $1\,\mathrm{TB} = 10^{12}$ bytes. Records depicted in Figure 5 enter its accounting. Their computed records contain 33.291 TB after excluding the frozen FP8 control and the failed $N = 32$ record. This includes under-resolved calculations and is not a total of fully resolved DNS alone. Including the depicted failure controls and the distinct filtered and quantized records gives 34.298 TB.

Reproduction records. The source manifests record array shapes, coordinate conventions, time steps, viscosities, frame counts, and storage formats. They also identify the initial states and solver versions using SHA-256 checksums. The reference simulation used JAX 0.9.1, while the coarse-grid simulations used JAX 0.11.0.

Table 5: Distinct stored payloads of the records depicted in Figure 5.

Contribution	TB
Resolution records, including the finite $N = 32$ failure	33.084213
Precision records, including the frozen FP8 control	0.222906
Five filtered movies	0.879608
Additional quantized records, excluding the shared FP64 reference	0.111453
All distinct records	**34.298181**
Computed records excluding the frozen FP8 and failed $N = 32$ controls	33.291379

Replaying one snapshot interval of the reference with the pinned solver gives a relative Fourier-coefficient error of 8.2×10^{-9}. This confirms short-time agreement, but not bitwise reproducibility across different backends. The filtering manifests additionally store the cutoffs, sample times, retained frequency bins, and construction checks.

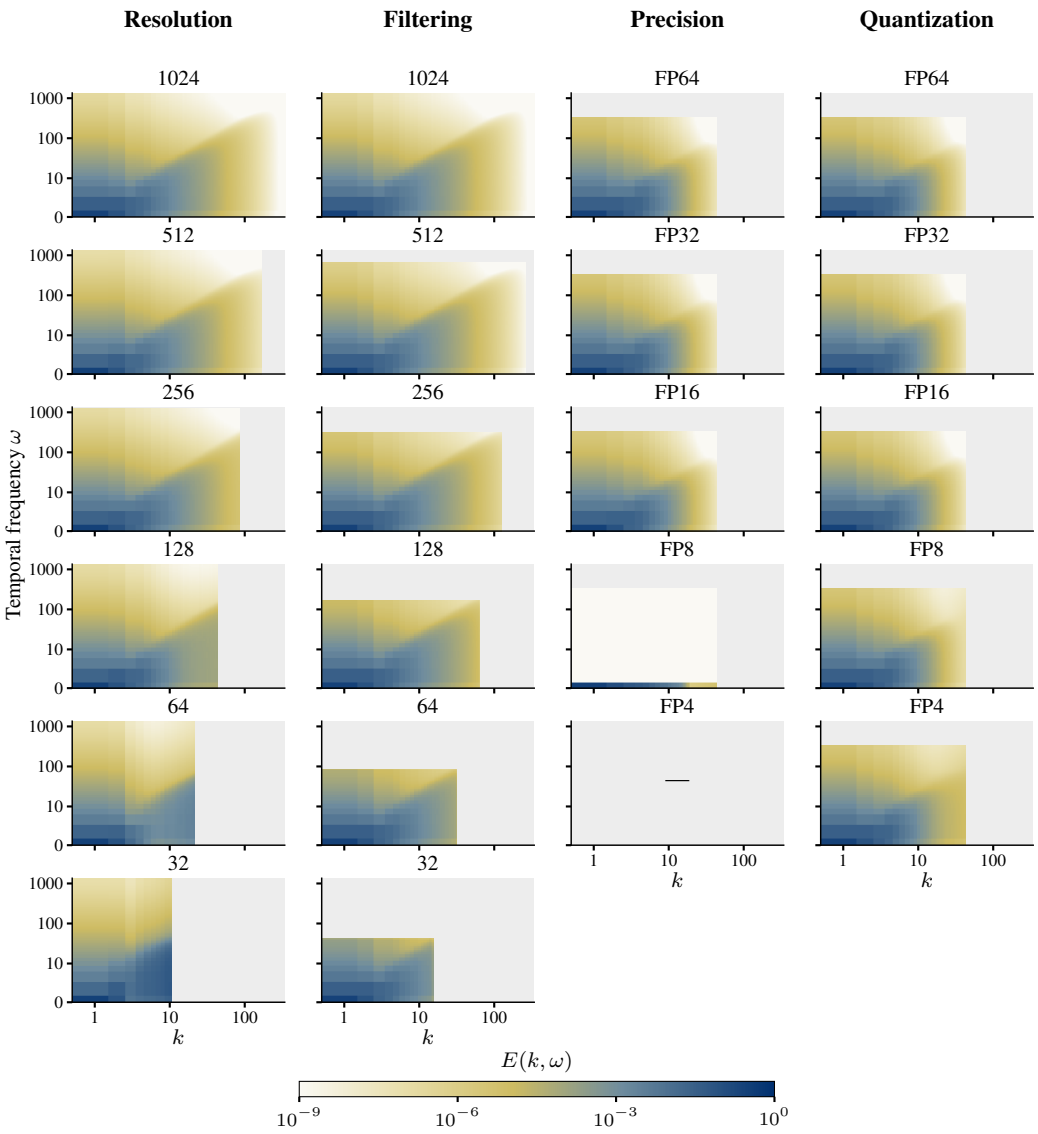


Figure 7: **Joint spatial–temporal spectra of the numerical records.** This adds the frequency distribution to the comparisons in Figure 5. *Resolution* changes the distribution even at shared scales. *Filtering* restricts its spatial and temporal extent. For *precision*, the frozen FP8 record retains zero-frequency energy, while FP4 supplies no valid temporal spectrum. Coarse *quantization* broadens the small-scale temporal content. All panels use the same absolute color scale. Gray denotes unrepresented spatial or temporal frequencies. Spectra use up to 512 sampled Fourier modes per shell, scaled to the measured full-shell energy. The temporal mean is retained. The frequency axis is linear near zero and logarithmic above it.

E EXPERIMENTS: NEURAL REPRESENTATIONS

Appendix D establishes how the computed and recorded flows differ. We now specify which part of each record is supplied to a neural representation and how that representation is fitted. We use the functions, loss, and parameter budget B defined in Appendix C.

E.1 LEARNING TARGETS

A common observation window. We apply $\mathcal{P}_{64}$ to the resolution, precision, and quantization records before fitting. This gives each comparison the same spatial cutoff, temporal cutoff, and sample times. The filtering comparison instead varies M, as listed in Table 6. All targets retain three spatial coordinates and time.

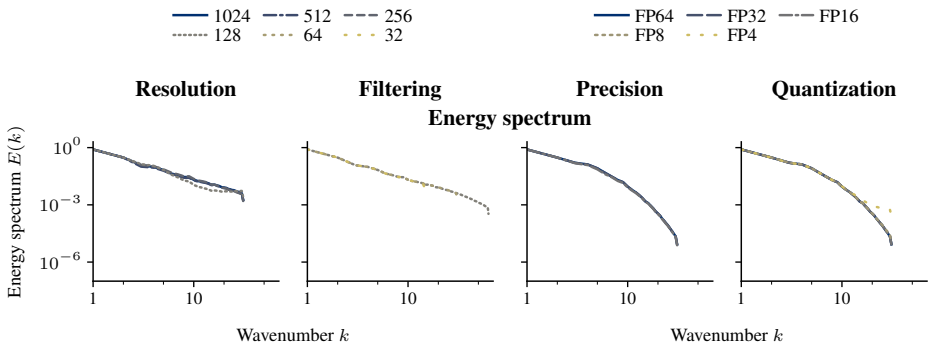
Table 6: Records supplied to neural fitting.

Comparison	Supplied record $\boldsymbol{u}^{\mathrm{rec}}$	Settings
Resolution	$\mathcal{P}_{64}\boldsymbol{u}_{N,\mathrm{FP32}}$	$N = 128, 256, 512, 1024$
Filtering	$\mathcal{P}_{M}\boldsymbol{u}_{1024,\mathrm{FP32}}$	$M = 32, 64, 128$
Precision	$\mathcal{P}_{64}\boldsymbol{u}_{128,\sigma}$	$\sigma = \mathrm{FP64}, \mathrm{FP32}, \mathrm{FP16}$
Quantization	$\mathcal{P}_{64}\mathcal{Q}_{q}\boldsymbol{u}_{128,\mathrm{FP64}}$	$q = \mathrm{FP64}, \mathrm{FP32}, \mathrm{FP16}, \mathrm{FP8}, \mathrm{FP4}$

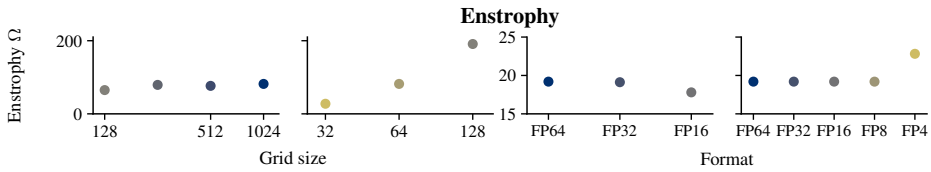
The projection uses the calibrated cutoff and window in Appendix D. At $M = 64$, $k_c = 31$, $\omega_c = 81.1980977570$, and $\Delta t_{64} = 0.03869047109$. The 64^4 samples cover the same interval of length $T_w = 2.47619014957$. For $M = 32$ and 128, $\Delta t_M = T_w/M$ and $k_c = M/2 - 1$. Each projection uses the complete source time series before evaluating this window. The resulting arrays have shape $(M, M, M, M, 3)$, ordered as $(t, x, y, z,$ velocity component$)$, and store FP32 values. This storage precision can mask differences between the finest quantization levels.

Quantization precedes projection: the learner receives a filtered version of an already rounded record. No solenoidal projection is added. The FP64 quantization target is the precision reference, and the $N = 1024$ resolution target is the $M = 64$ filtering target. These shared records are fitted once, giving 13 distinct targets. The $N = 32, 64$ records and FP8/FP4 evolution outcomes are excluded from this learning comparison. The numerical study retains them as under-resolution or arithmetic-failure cases.

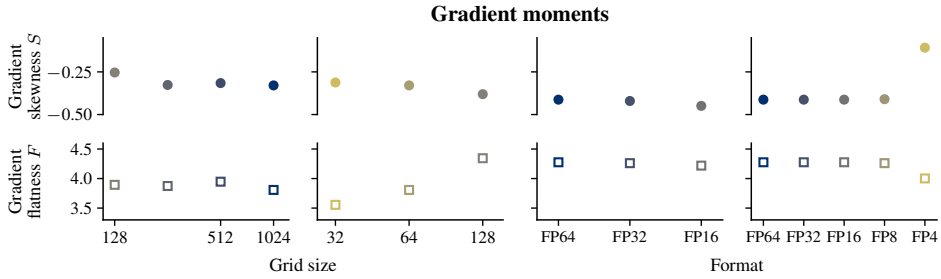
Effects of filtering. Figure 8 shows the motion supplied to fitting. The *energy spectra* and *enstrophy* retain differences between records, including the FP16 evolution deficit and the FP4 quantization excess. The *gradient moments* show a stronger change: the resolution records become more similar after their finest scales are removed, while FP4 quantization still leaves much weaker skewness. The *correlation times* retain the shortening associated with FP4 quantization and the broad scale dependence of the source records. Filtering therefore preserves the informative contrasts within the retained band. The fitting experiments concern this motion.



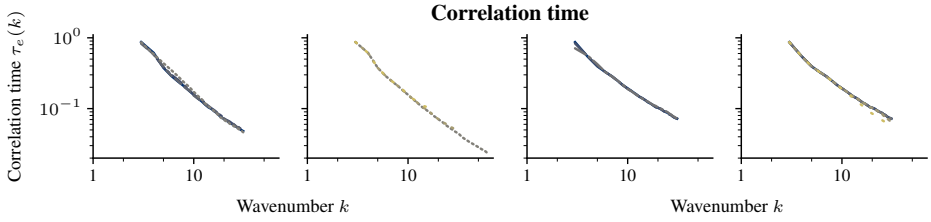
(a) The **energy spectrum** retains differences across source *resolutions*, with less small-scale energy at $N = 128$. Wider *filtering* bands recover more of the spectrum. The evolving *precision* records remain close, while FP4 *quantization* still leaves excess small-scale energy after projection.



(b) **Enstrophy** is lowest at $N = 128$ in the *resolution* comparison and grows with wider *filtering* bandwidth. The FP16 *precision* deficit remains. FP4 *quantization* still raises enstrophy, although projection removes some of the excess present in the unfiltered record.

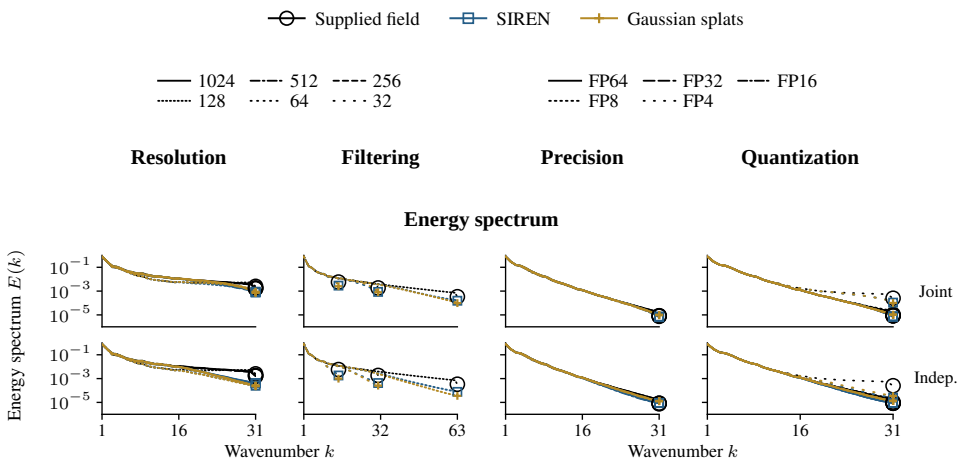


(c) The **gradient moments** become more similar across source *resolutions* after projection. Wider *filtering* bands retain more negative skewness and higher flatness. FP16 *precision* retains more negative skewness than FP64, whereas FP4 *quantization* leaves weaker skewness.

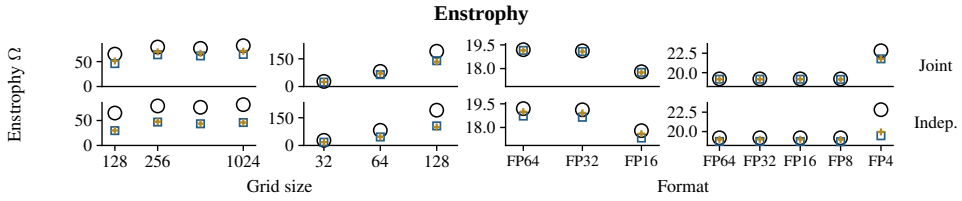


(d) **Correlation times** remain scale dependent. The $N = 128$ *resolution* record retains longer correlations at intermediate scales. Different *filtering* bandwidths largely agree on shared scales. The evolving *precision* records remain close, while FP4 *quantization* still shortens small-scale correlations.

Figure 8: Recorded motion supplied to neural fitting.

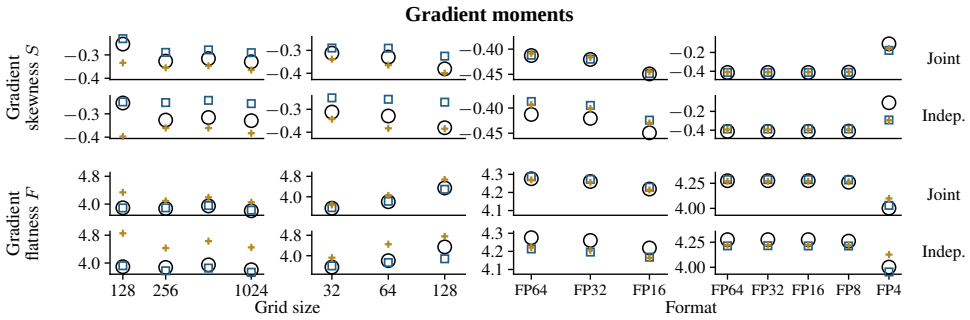


(a) The **energy spectrum** retains the energy-containing scales under both *joint* and *independent fitting*. Small-scale energy is generally underestimated, especially by independent fits. Across *resolution* and *filtering*, this removes supplied structure; for coarse *quantization*, it attenuates numerical excess.

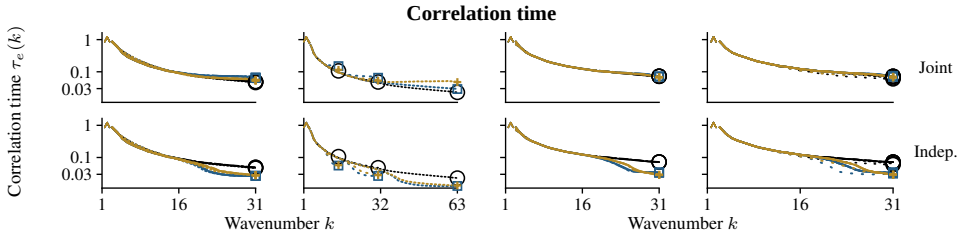


(b) **Enstrophy** is generally lower under *independent fitting* than under *joint fitting*. Deficits occur across source *resolutions* and grow with wider *filtering* bandwidths. On the FP4-quantized record, this reduction instead brings enstrophy closer to the unquantized value.

Figure 9: Joint and independent neural representations of the recorded flow.



(c) The **gradient moments** distinguish the network families: across *resolution* and *filtering*, *Gaussian splats* have more negative skewness and higher flatness than *SIRENs*. *Joint fitting* generally preserves the supplied gradient statistics more closely than *independent fitting*.



(d) **Correlation times** expose a temporal trade-off. *Independent fits* generally decorrelate too rapidly at small scales, whereas *joint fits* can remain correlated too long. This tendency spans *resolution*, *filtering*, *precision*, and *quantization*; the more accurate procedure depends on the record and network family.

Figure 9 (continued).

E.2 Absolute diagnostics of records and fits

Figure 9 gives the complete comparison, including the gradient moments omitted from the main figure. Tables 7–10 list the absolute diagnostics for each supplied record and its four fits. The supplied record is the reference for fitting, including when it contains quantization distortion. The FP64 quantization block provides the corresponding unquantized reference. We use the same definitions and averaging as Table 3, evaluated on the learning targets and reconstructed movies. The incompressibility measure d_∇ is defined in (21). No divergence-free projection is applied to the fitted fields. A dash marks a shell outside the retained band. Shared targets are repeated under the relevant comparisons, without counting them as additional fits. All values describe the runs plotted in the figure.

Table 7: Absolute diagnostics of supplied records and neural fits: resolution.

Record	Representation	$\overline{K}$	$\overline{\Omega}$	$\overline{S}$	$\overline{F}$	d_∇	$\tau_e(8)$	$\tau_e(24)$
$N = 1024$	Supplied	1.722	81.89	-0.329	3.807	5.70×10^{-6}	0.207	0.060
	SIREN, joint	1.697	64.42	-0.290	3.826	2.02×10^{-1}	0.207	0.074
	Gaussian, joint	1.708	70.75	-0.364	4.053	1.21×10^{-1}	0.207	0.063
	SIREN, indep.	1.660	45.96	-0.256	3.735	2.57×10^{-1}	0.207	0.031
	Gaussian, indep.	1.662	46.03	-0.383	4.450	1.76×10^{-1}	0.207	0.038
$N = 512$	Supplied	1.668	76.54	-0.316	3.947	4.91×10^{-6}	0.213	0.062
	SIREN, joint	1.648	61.41	-0.279	3.958	1.96×10^{-1}	0.213	0.074
	Gaussian, joint	1.656	66.67	-0.347	4.195	1.19×10^{-1}	0.213	0.064
	SIREN, indep.	1.612	43.42	-0.242	3.855	2.57×10^{-1}	0.213	0.031
	Gaussian, indep.	1.614	43.72	-0.361	4.630	1.76×10^{-1}	0.213	0.038
$N = 256$	Supplied	1.752	79.19	-0.327	3.874	5.06×10^{-6}	0.203	0.061
	SIREN, joint	1.730	63.69	-0.289	3.892	1.94×10^{-1}	0.203	0.073
	Gaussian, joint	1.739	69.48	-0.355	4.092	1.15×10^{-1}	0.203	0.064
	SIREN, indep.	1.695	46.91	-0.252	3.778	2.47×10^{-1}	0.203	0.030
	Gaussian, indep.	1.699	47.66	-0.360	4.430	1.69×10^{-1}	0.203	0.038
$N = 128$	Supplied	1.643	65.00	-0.254	3.894	5.54×10^{-6}	0.235	0.059
	SIREN, joint	1.618	45.95	-0.230	3.890	2.34×10^{-1}	0.235	0.074
	Gaussian, joint	1.626	51.26	-0.334	4.337	1.49×10^{-1}	0.235	0.063
	SIREN, indep.	1.588	29.73	-0.249	3.923	2.52×10^{-1}	0.235	0.030
	Gaussian, indep.	1.589	29.99	-0.397	4.855	1.84×10^{-1}	0.235	0.036

Table 8: Absolute diagnostics of supplied records and neural fits: filtering.

Record	Representation	$\overline{K}$	$\overline{\Omega}$	$\overline{S}$	$\overline{F}$	d_∇	$\tau_e(8)$	$\tau_e(24)$
$M = 128$	Supplied	1.780	191.02	-0.381	4.344	3.78×10^{-6}	0.207	0.058
	SIREN, joint	1.760	139.82	-0.325	4.293	2.13×10^{-1}	0.207	0.058
	Gaussian, joint	1.757	133.88	-0.399	4.675	1.70×10^{-1}	0.207	0.058
	SIREN, indep.	1.738	105.78	-0.269	3.880	2.58×10^{-1}	0.207	0.057
	Gaussian, indep.	1.735	99.52	-0.385	4.757	2.14×10^{-1}	0.207	0.057
$M = 64$	Supplied	1.722	81.89	-0.329	3.807	5.70×10^{-6}	0.207	0.060
	SIREN, joint	1.697	64.42	-0.290	3.826	2.02×10^{-1}	0.207	0.074
	Gaussian, joint	1.708	70.75	-0.364	4.053	1.21×10^{-1}	0.207	0.063
	SIREN, indep.	1.660	45.96	-0.256	3.735	2.57×10^{-1}	0.207	0.031
	Gaussian, indep.	1.662	46.03	-0.383	4.450	1.76×10^{-1}	0.207	0.038
$M = 32$	Supplied	1.609	27.60	-0.312	3.555	9.61×10^{-6}	0.209	—
	SIREN, joint	1.588	23.92	-0.290	3.574	1.63×10^{-1}	0.211	—
	Gaussian, joint	1.594	24.52	-0.338	3.682	1.08×10^{-1}	0.208	—
	SIREN, indep.	1.539	17.77	-0.250	3.602	2.48×10^{-1}	0.204	—
	Gaussian, indep.	1.542	17.62	-0.343	3.924	1.60×10^{-1}	0.203	—

Table 9: Absolute diagnostics of supplied records and neural fits: precision.

Record	Representation	$\overline{K}$	$\overline{\Omega}$	$\overline{S}$	$\overline{F}$	d_∇	$\tau_e(8)$	$\tau_e(24)$
FP64	Supplied	1.617	19.20	-0.413	4.276	1.33×10^{-7}	0.245	0.089
	SIREN, joint	1.617	19.15	-0.412	4.287	2.08×10^{-2}	0.245	0.092
	Gaussian, joint	1.617	19.15	-0.408	4.270	4.13×10^{-2}	0.245	0.091
	SIREN, indep.	1.616	18.72	-0.387	4.213	7.03×10^{-2}	0.245	0.050
	Gaussian, indep.	1.616	19.00	-0.393	4.221	7.69×10^{-2}	0.245	0.066
FP32	Supplied	1.612	19.12	-0.420	4.260	1.77×10^{-7}	0.245	0.089
	SIREN, joint	1.612	19.08	-0.420	4.272	2.09×10^{-2}	0.245	0.091
	Gaussian, joint	1.612	19.07	-0.416	4.256	4.13×10^{-2}	0.245	0.091
	SIREN, indep.	1.611	18.65	-0.395	4.196	7.07×10^{-2}	0.245	0.049
	Gaussian, indep.	1.611	18.92	-0.401	4.206	7.70×10^{-2}	0.245	0.066
FP16	Supplied	1.573	17.80	-0.449	4.219	2.15×10^{-3}	0.246	0.092
	SIREN, joint	1.573	17.76	-0.448	4.229	2.07×10^{-2}	0.246	0.094
	Gaussian, joint	1.573	17.75	-0.445	4.214	4.22×10^{-2}	0.246	0.093
	SIREN, indep.	1.572	17.33	-0.424	4.168	7.21×10^{-2}	0.246	0.048
	Gaussian, indep.	1.572	17.61	-0.430	4.164	7.90×10^{-2}	0.246	0.065

Table 10: Absolute diagnostics of supplied records and neural fits: quantization.

Record	Representation	$\overline{K}$	$\overline{\Omega}$	$\overline{S}$	$\overline{F}$	d_∇	$\tau_e(8)$	$\tau_e(24)$
FP64	Supplied	1.617	19.20	-0.413	4.276	1.33×10^{-7}	0.245	0.089
	SIREN, joint	1.617	19.15	-0.412	4.287	2.08×10^{-2}	0.245	0.092
	Gaussian, joint	1.617	19.15	-0.408	4.270	4.13×10^{-2}	0.245	0.091
	SIREN, indep.	1.616	18.72	-0.387	4.213	7.03×10^{-2}	0.245	0.050
	Gaussian, indep.	1.616	19.00	-0.393	4.221	7.69×10^{-2}	0.245	0.066
FP32	Supplied	1.617	19.20	-0.413	4.276	1.34×10^{-7}	0.245	0.089
	SIREN, joint	1.617	19.16	-0.412	4.287	2.05×10^{-2}	0.245	0.092
	Gaussian, joint	1.617	19.15	-0.408	4.270	4.12×10^{-2}	0.245	0.091
	SIREN, indep.	1.616	18.73	-0.387	4.215	7.02×10^{-2}	0.245	0.050
	Gaussian, indep.	1.616	19.00	-0.393	4.221	7.69×10^{-2}	0.245	0.066
FP16	Supplied	1.617	19.20	-0.413	4.276	1.01×10^{-4}	0.245	0.089
	SIREN, joint	1.617	19.15	-0.412	4.287	2.08×10^{-2}	0.245	0.092
	Gaussian, joint	1.617	19.15	-0.408	4.270	4.12×10^{-2}	0.245	0.091
	SIREN, indep.	1.616	18.72	-0.387	4.213	7.06×10^{-2}	0.245	0.049
	Gaussian, indep.	1.616	19.00	-0.393	4.222	7.69×10^{-2}	0.245	0.066
FP8	Supplied	1.615	19.19	-0.410	4.261	2.44×10^{-2}	0.245	0.087
	SIREN, joint	1.615	19.13	-0.411	4.284	2.56×10^{-2}	0.245	0.092
	Gaussian, joint	1.615	19.13	-0.408	4.266	4.36×10^{-2}	0.245	0.091
	SIREN, indep.	1.614	18.70	-0.387	4.212	7.12×10^{-2}	0.245	0.050
	Gaussian, indep.	1.614	18.98	-0.393	4.219	7.77×10^{-2}	0.245	0.066
FP4	Supplied	1.609	22.82	-0.108	4.002	2.59×10^{-1}	0.244	0.071
	SIREN, joint	1.607	21.78	-0.178	4.030	2.33×10^{-1}	0.244	0.082
	Gaussian, joint	1.607	21.79	-0.158	4.099	2.30×10^{-1}	0.243	0.078
	SIREN, indep.	1.601	19.46	-0.291	3.954	1.72×10^{-1}	0.244	0.032
	Gaussian, indep.	1.602	19.96	-0.295	4.127	1.72×10^{-1}	0.244	0.050

E.3 Comparison and fitting protocol

What is controlled. SIRENs share dense transformations across the domain. Gaussian fields assemble localized, adaptive contributions. These different inductive biases may preserve different aspects of the same supplied motion. We examine those differences with the spatial and temporal measures of Appendix A. At each M, both families are fitted by velocity MSE to a common stopping criterion under an approximately matched total parameter budget B. This compares the representations obtained under a specified fitting protocol. It does not require equal final MSE.

We distribute B as in Appendix C.2. A joint fit uses all parameters in one function $f_\theta(\boldsymbol{x}, t)$. An independent fit divides them nearly equally among the M functions $f_{\theta_j}(\boldsymbol{x})$. Both procedures reconstruct the same observed samples.

Coordinates and normalization. Each grid index i becomes $2i/M - 1$. Joint models receive the four normalized coordinates in the order (t, x, y, z). Independent models receive (x, y, z) and use the frame index to select parameters. We subtract one scalar mean μ and divide by one scalar standard deviation s_u over all velocity components in the movie. Quantized targets use the corresponding unquantized target's μ and s_u. The same constants are used for every fitting procedure and network family on a given target. Exporting the field reverses this transformation.

Loss and sampling. Training minimizes (16) in these standardized units. Every step samples $65{,}536$ space–time grid locations independently and uniformly, with replacement. All three velocity components contribute equally. This is one batch for the complete movie, not one batch per independent frame. For independent fitting, samples are grouped by frame without changing their weights. No derivative loss, temporal penalty, or equation residual is included.

Optimization and stopping. We use Adam (Kingma & Ba, 2015) with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, without weight decay. Each cycle contains 20,000 updates with a cosine learning-rate decay to zero. We retain the fitted parameters, but reset the Adam state and learning-rate schedule at the start of each cycle. Table 11 gives the initial rates. The joint filtering runs supplied these settings, which are reused across targets and fitting procedures without target-specific retuning.

After each cycle, we evaluate MSE over the complete supplied grid. Let $\mathcal{L}_c$ be this standardized MSE and R the fixed standardized velocity range used for that target. We compute $P_c = 10 \log_{10}(R^2/\mathcal{L}_c)$. Training stops when two consecutive values of $|P_c - P_{c-1}|$, rounded to four decimals, are below 0.1 dB. Otherwise it stops at 30 cycles, or 600,000 updates. The last cycle supplies the checkpoint. For independent fits, the criterion uses the complete movie, not a separate stopping decision for each frame. Here, *convergence* means this measured plateau, not a demonstrated global minimum. All runs included here reached the plateau criterion before the cap.

Table 11: Initial learning rates within each restart cycle.

Parameters	$M = 32$	$M = 64$	$M = 128$
SIREN weights, biases	6×10^{-4}	3×10^{-4}	3×10^{-4}
Gaussian centers	0.024	0.006	0.003
Gaussian log-diagonal and off-diagonal shape	0.040	0.010	0.005
Gaussian velocity coefficients	0.080	0.020	0.010

E.4 ARCHITECTURE SPECIFICATIONS

SIREN. All variants use five hidden sine layers and a linear three-component output. The first hidden layer applies $\sin(Mh)$ to its affine preactivation h. Later hidden layers apply $\sin(h)$. Joint networks have constant hidden width 288, 1156, or 4630 for $M = 32, 64, 128$, respectively. At $M = 32$, independent networks have width 49 for six frames and 50 for the other 26. At $M = 64$, 44 frames use width 143 and 20 use width 144. At $M = 128$, all 128 frames use width 408. Frames receiving either width are assigned by a fixed permutation with seed 20260912.

Weights entering the first layer are drawn uniformly from $[-1/d_{\text{in}}, 1/d_{\text{in}}]$. Subsequent layers, including the output, use bounds $\pm\sqrt{6/d_{\text{in}}}/M$, where d_{in} is the input width. Biases start at zero.

Additive Gaussian fields. We instantiate the field in Appendix C.3 with $g = 9, 18, 36$ cells per coordinate at $M = 32, 64, 128$. The normalized cell spacing is $s_g = 2/g$. Joint fitting places three primitives in each cell of a four-dimensional lattice. Each primitive has four center parameters, ten lower-triangular shape parameters, and three signed velocity coefficients. The off-diagonal shape entries allow space–time coupling. Independent fitting instead uses a separate three-dimensional bank per frame, with 12 parameters per primitive: three for its center, six for its shape, and three for its velocity.

A raw center parameter r gives an offset $(s_g/2)\tanh r$ from its cell center. Positive diagonal shape entries are parameterized by their logarithms. For minimum-image coordinate displacement d, the compact taper is

$$w(d) = \left[\max\left(1 - \left(\frac{\|d\|_\infty}{2s_g}\right)^2, 0\right)\right]^3.$$

Contributions are added without normalization or opacity compositing. Coordinate differences wrap with period two. This includes time in the joint Gaussian model, an imposed boundary convention rather than an assertion that the observation window is physically periodic. SIREN has no corresponding periodic boundary constraint.

Raw centers start uniformly in $[-1, 1]$, off-diagonal entries at zero, and velocity coefficients as Gaussian values with standard deviation 0.01. The initial isotropic width is $0.6s_g/3^{1/4}$ for both procedures. Independent banks distribute primitives as evenly as possible across frames. Fixed cell assignments are specified by the construction seed 20260912.

Table 12: Total learned scalar parameters per movie.

Family	Procedure	$M = 32$	$M = 64$	$M = 128$
SIREN	Joint	335,235	5,359,219	85,803,163
	Independent	335,254	5,358,988	85,804,416
Gaussian	Joint	334,611	5,353,776	85,660,416
	Independent	334,608	5,353,776	85,660,416

All learned parameters use FP32. Within each family, independent counts differ from joint counts by less than 0.006% because layer widths and primitive numbers are integers. Across the two families, counts differ by less than 0.2%. The budget increases with M, so the filtering comparison changes bandwidth and capacity together.

E.5 Evaluation and reproducibility

Common measurements. We export each fit at every supplied coordinate and recorded time, restore physical velocity units, and apply the same spectral derivatives and temporal estimator as for its target. All M frames enter the spatial statistics. Energy spectra and correlation times are displayed through the target cutoff, while scalar gradient statistics include every mode on the evaluation grid. We apply no extra filtering or divergence-free projection to the prediction. This evaluates reconstruction of the supplied movie, not forecasting or accuracy between recorded coordinates.

Numerical implementation. JAX and Optax implement the training loops. Parameters and training arrays are FP32. Full-grid error sums use binary64 on the host. The stopping evaluator retains the original JAX matrix-product setting, which can use TF32 on the GPU. Exported fields instead use the highest FP32 matrix-product precision. Consequently, exported MSE can differ slightly from the stopping value. JAX's 64-bit option is enabled for SIREN to retain the original random-sampling stream, without changing parameter dtype. Validated batching and compilation changes preserve the model, sampled loss, and stopping rule. Per-run manifests identify the backend and source hashes.

The main comparisons use initialization index zero for each setting. SIREN's base random key is $20260713 + W$, with joint width W as above. Gaussian fields use $20260801 + M$. Independent frame initializations use distinct folded keys.

Sensitivity to training randomness. The main study uses one training seed per setting, so the observed differences between joint and independent fits could in principle reflect a favorable initialization or sequence of minibatches. We test this possibility on the unquantized $N = 1024$ record at $M = 64$. For each network family, we repeat joint and independent fitting with three seeds. Only the initialization and minibatch sequence change. The data, architecture, parameter budget, optimizer, and stopping rule remain fixed, and all fits reach the prescribed plateau.

Table 13 shows that seed variation is small relative to the differences between fitting procedures. For both network families, every independent fit retains less enstrophy and has shorter correlation times than every joint fit. The differences in energy spectra and gradient moments are likewise larger than their variation across seeds. Thus, on this record, the distinction between joint and independent fitting is repeatable rather than the result of one favorable training run.

Table 13: Physical diagnostics across three training seeds on one unquantized $N = 1024$ record at $M = 64$.

Fit	Spectrum error (%)	Enstrophy	S	F	$\langle \tau/\tau_{\mathrm{rec}} \rangle_k$
Record	0	81.89	-0.329	3.807	1
SIREN, Joint	1.70 ± 0.03	64.17 ± 0.22	-0.289 ± 0.002	3.824 ± 0.003	1.116 ± 0.001
SIREN, Indep.	3.63 ± 0.01	46.12 ± 0.14	-0.255 ± 0.001	3.725 ± 0.008	0.801 ± 0.001
Gaussian, Joint	0.96 ± 0.00	70.75 ± 0.00	-0.364 ± 0.000	4.052 ± 0.001	1.029 ± 0.000
Gaussian, Indep.	3.64 ± 0.00	46.03 ± 0.00	-0.383 ± 0.000	4.448 ± 0.004	0.846 ± 0.000

Entries are means $\pm$ sample standard deviations over three seeds. The record row gives the values that a faithful fit would reproduce. Spectrum error is $100 \sum_{k=1}^{31} |E_{\mathrm{fit}}(k) - E_{\mathrm{rec}}(k)| / \sum_{k=1}^{31} E_{\mathrm{rec}}(k)$. The last column averages the ratio of fitted to recorded correlation times across $k = 3, \ldots, 31$, so values above and below one denote correlations that persist too long and decay too quickly, respectively.

E.6 SPATIAL ACCURACY, TEMPORAL BEHAVIOR, AND NETWORK FAMILY

Comparing fits at matched spatial error. On the clean movie, joint fits attain lower spatial error under the main study's fixed parameter budget. We ask whether their temporal advantage remains when snapshot accuracy is comparable. On the same unquantized $M = 64$ movie used above, we select joint and independent checkpoints whose full-grid MSEs agree within 2%. Across three seeds per family, joint fits have lower error in adjacent-frame velocity changes in every pair (Table 14). At twice this time lag, independent fits have lower error in every pair. Thus, similar snapshot accuracy can accompany different accuracy of the recorded changes, and the preferred procedure depends on the time interval.

Table 14: Trained checkpoints at matched spatial error on the unquantized $N = 1024$, $M = 64$ movie.

		Spatial MSE ($\times 10^2$)		Change error: Indep./Joint		
Family	Seed	Joint	Indep.	Δt	$2\Delta t$	$4\Delta t$
SIREN	0	4.544	4.463	1.358	0.939	0.933
SIREN	1	4.552	4.517	1.367	0.947	0.944
SIREN	2	5.467	5.470	1.391	0.984	0.942
Gaussian	0	3.457	3.427	1.229	0.917	1.003
Gaussian	1	3.456	3.426	1.229	0.916	1.003
Gaussian	2	3.460	3.427	1.229	0.916	1.002

Spatial MSE is measured in standardized velocity units over the full grid. Each pair agrees within 2%. Change error is the mean squared error in the velocity difference between two recorded times. Ratios above one favor joint fitting.

Which spatial scales carry the temporal error? A correlation time summarizes how quickly motion at a given spatial scale changes. The joint spectrum $E(k, \omega)$ shows how its energy is distributed over temporal frequencies. We compute this spectrum from the recorded Fourier coefficients over the same observation windows used for the correlation measurements. Figure 7 extends the numerical comparison, and Figure 10 compares each supplied record with its fits. Across the resolution targets, independent fits spread small-scale energy over higher temporal frequencies, whereas joint fits tend to concentrate it at lower frequencies. This pattern remains when we taper the time window to reduce endpoint leakage and when we use every spatial mode on the clean $M = 64$ movie. The maps locate the changes behind the correlation-time curves: neural fitting alters both the energy at a spatial scale and how that energy is distributed in time. The full spectrum complements a single correlation time, while describing the same second-order temporal information as the complete correlation function.

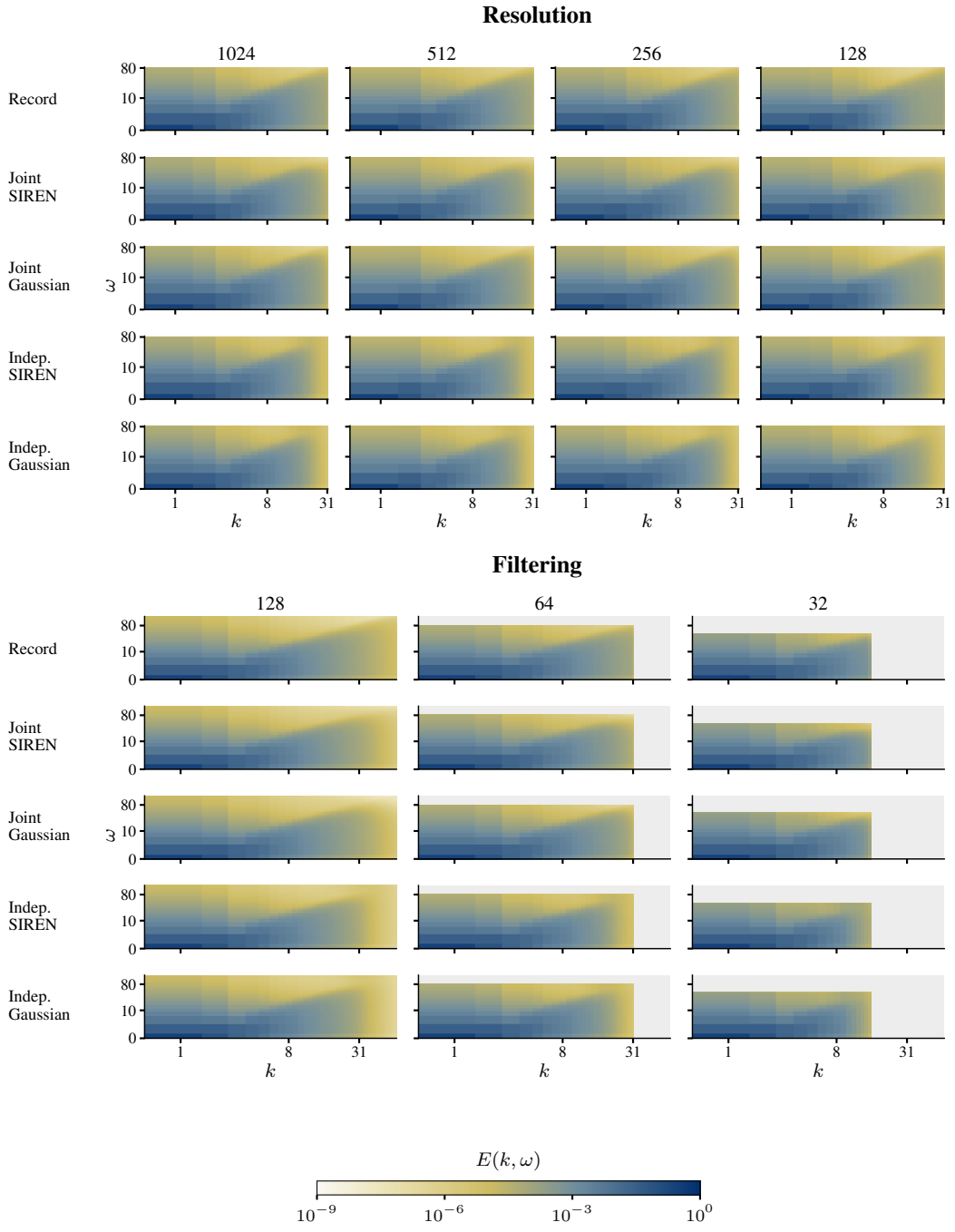


Figure 10: **Joint spectra of supplied records and their neural fits.** The resolution and filtering comparisons extend Figure 6. Every column compares a record with its joint and independent SIREN and Gaussian fits. Independent fitting broadens small-scale temporal spectra, while joint fitting tends to narrow them. The maps display absolute energy on a common color scale, without normalizing each panel. The estimator follows Figure 7. Gray denotes frequencies outside the supplied sampling range. The precision and quantization comparisons continue on the next page.

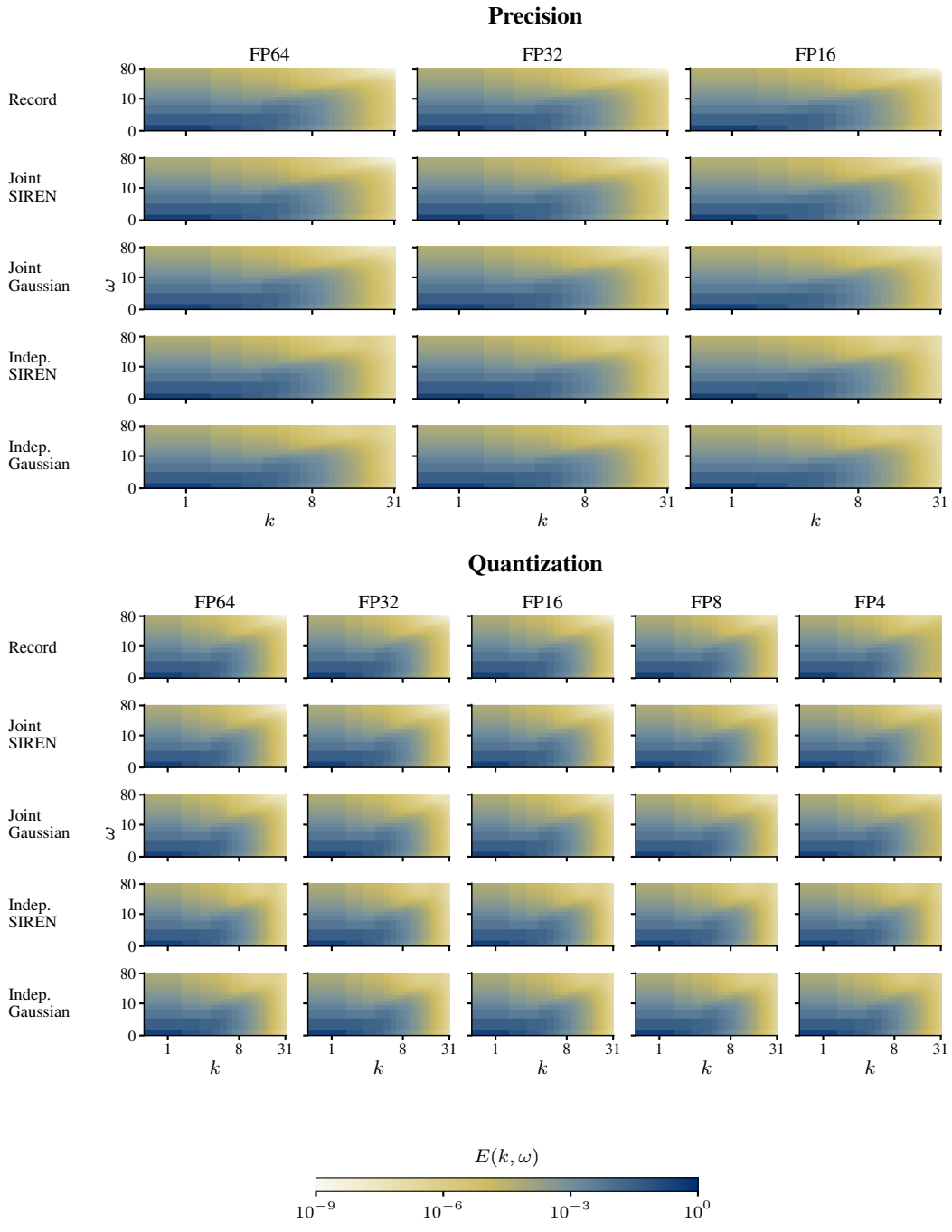


Figure 10 (continued). **Precision and quantization.** The same estimator, axes, and absolute color scale compare each record with its fits. Strong quantization and neural fitting affect both shell energy and its temporal distribution.

Why do the gradient moments differ between network families? We saw that Gaussian fits have more negative gradient skewness and higher flatness than SIRENs. We ask whether this difference can be explained simply by the spatial scales retained by each model.

To test this, we use the same clean movie with three training seeds for each architecture and fitting procedure. We restrict all fields to the common shells $k = 1, \ldots, 31$ and rescale the longitudinal-gradient power in each shell to match the record. After this matching, the difference remains: Gaussian fits still show more negative skewness and higher flatness than SIRENs under both fitting procedures (Figure 11). For independent fits, the matched flatness is 5.07 for Gaussians and 3.61 for SIRENs, compared with 3.81 in the record.

Thus, the family difference is not simply spectral: even after matching the gradient power at each spatial scale, Gaussian fits retain more negative skewness and higher flatness. The architectures therefore differ in how gradients are organized within those scales, with strong gradients contributing more prominently in the Gaussian representations.

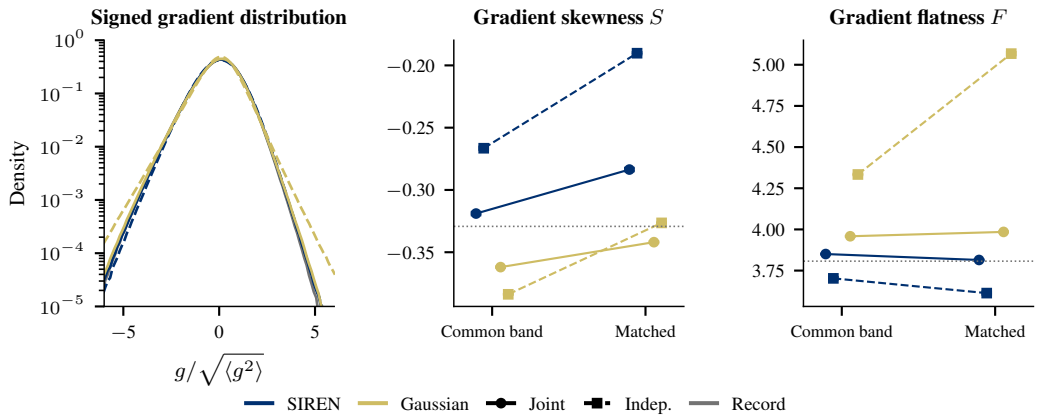


Figure 11: **Matching gradient power does not remove the family difference.** All fields are restricted to the same spatial band, and their gradient power is matched to the supplied record shell by shell. Gaussian fits still show more negative skewness and higher flatness than SIRENs under both fitting procedures. Points and error bars show means and sample standard deviations over three training seeds per architecture and fitting procedure.