Rectified Flows for Fast Multiscale Fluid Flow Modeling

Anonymous authors
Paper under double-blind review

Abstract

Statistical surrogate modeling of fluid flows is challenging due to multiscale dynamics and strong sensitivity to initial conditions. Conditional diffusion models can achieve high fidelity, but typically require hundreds of stochastic steps at inference. We introduce a rectified-flow surrogate that learns a time-dependent conditional velocity field transporting input-to-output laws along nearly straight trajectories. Sampling reduces to solving a deterministic ODE along this learned transport, so each function evaluation is substantially more effective: on multi-scale 2D benchmarks we match diffusion-class posterior statistics with as few as 8 ODE steps versus ≥ 128 steps for score-based diffusion.

On the theory side, we develop a law-level analysis tailored to conditional PDE forecasts. First, we formalize the link between our evaluation criterion—one-point Wasserstein distances on fields—and the $k=1$ correlation-marginal viewpoint in statistical solutions. Second, we provide a one-step error decomposition for the learned pushforward law into a *coverage* (high-frequency tail) term controlled by structure functions (equivalently, by spectral decay), and a *fit* term controlled directly by the training objective. Third, we show how *straightness* in rectification time governs local truncation error for ODE sampling, yielding step-count requirements and explaining why rectified transports admit large, stable steps.

Guided by this picture, we propose a curvature-aware sampler that tracks an EMA-based straightness proxy and adaptively blends and steps the velocity during inference. Across multiscale incompressible and compressible 2D flows, our method matches diffusion models in Wasserstein statistics and energy spectra, preserves fine-scale structure missed by MSE baselines, and delivers high-resolution conditional samples at a fraction of the inference cost.

1 Introduction

Partial differential equations (PDEs) [3] are the backbone of modeling physical phenomena, from atmospheric dynamics to aerodynamics and magnetohydrodynamics. Currently, solutions of PDEs are simulated with high-fidelity numerical solvers such as finite difference, finite element, and spectral methods [23]. These are often prohibitively expensive for *many–query* tasks [24] such as uncertainty quantification, optimization, and real-time prediction. In particular, these solvers encounter a significant challenge for classes of non-linear PDEs, whose solutions exhibit extreme sensitivity to initial and boundary data, leading to chaotic behavior and complex multi-scale features. In such settings, one seeks, not a deterministic trajectory, but rather the *statistical solution* of the PDE, i.e., the push-forward of the underlying probability measure on inputs (initial and boundary conditions etc) by the PDE solution operator [4]. Currently, these statistical solutions are sampled with ensemble Monte Carlo method by repeatedly calling high-fidelity PDE solvers [12]. Given the computational cost of the PDE solver, ensemble methods become infeasible as ensemble sizes grow.

Machine learning and AI based algorithms are increasingly being explored for the data-driven simulation of PDEs [20]. In particular, *neural operators* [11], which directly seek to learn the PDE solution operator from data, are already widely used in a PDE context [20]. Examples of neural operators have been proposed in [14, 16, 18, 29, 9, 15, 1] and references therein.

However, these neural operators have been found to be unsuitable for simulating PDE with chaotic, multiscale solutions such as governing equations of fluid dynamics. For instance, recently in [21], it was shown both

theoretically and empirically that neural operators, trained to minimize mean absolute (square) errors, will always collapse to the ensemble mean when used for simulating statistics of PDEs with chaotic multiscale solutions.

Given this context, generative AI models—especially score-based and diffusion models—have recently shown remarkable success at learning complex, high-dimensional distributions by progressive denoising score matching [27]. In the context of chaotic PDEs, *GenCFD* [21] introduced a conditional diffusion framework that preserves spectral content across fine scales, mitigating the "spectral collapse" of naive MSE-trained surrogates, see also [22, 8, 6, 7]. However, sampling a score-based diffusion model requires simulating a reverse-time stochastic differential equation (SDE) over 100+ discretization steps, each invoking a large ML model on a moderate-to-high resolution spatial grid, resulting in high runtimes that undermine real-time or resource-constrained applications.

To overcome this limitation of diffusion models, we adapt *rectified flows* [17]: a deterministic ODE-based analogue of diffusion that "straightens" transport paths in probability space. Rather than small random increments, rectified flows solve an ODE whose velocity field is trained to match the diffusion model's instantaneous score, but whose trajectories remain nearly linear. This structure permits large time integration, reducing the number of solver steps by an order of magnitude or more, while provably recovering the same target distribution.

What is evaluated, and why. Because statistical solutions are measures on function spaces, the most robust comparison is via *observables* or *correlation marginals* [13]. Our evaluation metric in Section 6.2—the *one-point Wasserstein distance*—is exactly the Wasserstein discrepancy between $k=1$ correlation marginals (pointwise push-forwards), averaged over space. This aligns the experimental metrics (Table 1) with the law-level theoretical statements in Sections 4.1–4.3.

Hence, our main contributions are:

- *Rectified flows for conditional PDEs.* We learn a time-dependent velocity field that transports a *conditional law* of fields from a simple reference (Gaussian noise conditioned on the input) to the target PDE pushforward law. The key theoretical point is *correctness at the level of laws*: the *barycentric rectified velocity* induces an ODE whose flow maps the initial law to the target law (**SM** C).
- *Curvature-aware integration.* The sampling cost is governed by the amount of *time-bending* (straightness/curvature in rectification time) of the learned vector field along typical trajectories. We formalize: (i) which curvature quantity controls discretization error, (ii) why a simple EMA deviation is a principled proxy for that curvature, and (iii) why the blending step is the minimizer of a quadratic (Tikhonov) correction (**SM** E).
- *Multi-scale accuracy at speed.* At fixed grid resolution there is an *irreducible* law-level error floor coming from unresolved Fourier scales. We express this as a *coverage* term controlled by structure functions / spectral tails (Crefapp:sec:coverage-structure). Rectified flows reduce curvature so that the best attainable (bandlimited) law is reached with far fewer sampling steps. This is precisely what is seen in Table 1 and in the spectral evolution plots (Figure 13).

2 Approach

Modeling multi-scale flows and their statistical solutions. We focus on time-dependent PDEs of the general form

$$\partial_t\, u(x,t) \;+\; \mathcal{L}\big(u, \nabla_x u, \nabla_x^2 u, \dots\big) \;=\; 0, \quad x \in D \subset \mathbb{R}^d,\; t \in (0,T), \tag{1}$$

subject to boundary conditions $\mathcal{B}(u) = u_b$ on $\partial D \times (0,T)$ and initial data $u(x,0) = \bar{u}(x)$. Here, the solution $u : D \mapsto \mathbb{R}^m$ is evolved in time, with respect to the differential operator $\mathcal{L}$. Concrete examples of (1) are the incompressible Navier-Stokes and compressible Euler equations.

A recurring theme in non-linear, multi-scale fluid flows is *extreme sensitivity* to initial or boundary data, which can lead to chaotic or near-chaotic behavior. Rather than fix a single u_0, one may consider an *initial measure* μ_0 to represent uncertain or variable initial states, and then study the *statistical solution* $\mu_t = \mathcal{S}^t_\# \mu_0$ pushed forward by the PDE solution operator $\mathcal{S}^t$. In strongly nonlinear regimes, μ_t spreads significantly in function space, creating complex, multi-scale distributions of possible flow fields. Accurately approximating this evolving distribution is essential for uncertainty quantification and design under rough conditions, but doing so with high-fidelity PDE solvers can be prohibitively expensive when large ensembles are required.

High cost of computing statistical solutions. In practice, characterizing μ_t directly often requires performing *many* forward simulations of the PDE solver, each initialized with a slightly perturbed u_0. As each simulation might require high-resolution discretizations (e.g. millions of degrees of freedom) and multiple time steps, the overall cost for constructing a sizable ensemble rapidly becomes prohibitive. Even with modern high-performance computing platforms, running hundreds or thousands of high-fidelity statistical simulations can be infeasible. Hence, there is a pressing need for more efficient *surrogate models* that can *sample* physically realistic final states without enumerating every realization of the forward PDE solve. Succinctly put, we aim to provide a solution to the following problem

How can we efficiently capture statistical quantities of interest for the pushforward measure $S^t_\# \mu_0$ for highly sensitive solution operators S^t?

Observables and correlation marginals (what metrics actually probe). A classical way to formalize "quantities of interest" for μ_t is via observables and their induced correlation marginals [13]. For $k = 1$, the pointwise evaluation map at $x \in D$,

$$\mathrm{ev}_x : \; u \mapsto u(x) \in \mathbb{R}^m,$$

pushes μ_t to the *one-point marginal* $(\mathrm{ev}_x)_\# \mu_t$ on $\mathbb{R}^m$ – strictly speaking, one studies the pushforwards measures $\mathrm{ev}_x \circ i_\epsilon$, here $(i_\epsilon u) := (\phi_\epsilon * u)$ where ϕ_ϵ is a compactly supported mollifier, so that point-wise evaluation is well-defined; for the sake of brevity, we will not make this distinction explicit in the sequel. Our one-point Wasserstein metric (Section 6.2) is the spatial average of $W_1\big((\mathrm{ev}_x)_\# \mu_t, (\mathrm{ev}_x)_\# \widehat{\mu}_t\big)$. By Kantorovich–Rubinstein duality, controlling this quantity controls the discrepancy of *all* pointwise 1-Lipschitz observables, averaged over space. This is the bridge between Table 1 and the law-level bounds in Section 4.1 and Section 4.3.

Rectified flows for faster sampling. To mitigate these inefficiencies, we adopt the framework of *rectified flows* [17]: a deterministic alternative to conditional diffusion designed to reduce the number of required integration steps. In essence, rather than generating samples via small random "diffusion-reversal" increments, rectified flows construct an *ordinary differential equation* (ODE) with *nearly straight* trajectories in function space. This straighter transport map can be integrated more aggressively in time, enabling sampling with far fewer solver steps. Importantly, rectified flows retain the desirable property of matching the same final distributions that standard diffusion models learn, but at a fraction of the runtime cost.

3 Related Work

3.1 Diffusion Models in Fluid Dynamics

Conditional diffusion surrogates. *GenCFD* [21] trains a conditional score-based diffusion model that starts from an initial or low-resolution flow field and iteratively refines it through hundreds of denoising steps, thereby capturing multi-scale turbulent structure. Similarly, the method of [22] conditions a diffusion model on a coarse fluid solution (generated by a learned operator) and "super-resolves" the fine-scale flow. Given these similarities, we refer to such diffusion-based solvers collectively as *conditional diffusion PDE solvers*.

Rectified flow alternative. Each sample typically needs hundreds of network calls, limiting large-scale or real-time use. Our method replaces the stochastic SDE with a deterministic ODE whose nearly straight trajectories require far fewer steps, achieving similar statistical fidelity at much lower inference cost (see Sections 5.6 and 6). Crucially, we find that rectified flows are actually more efficient from the standpoint of

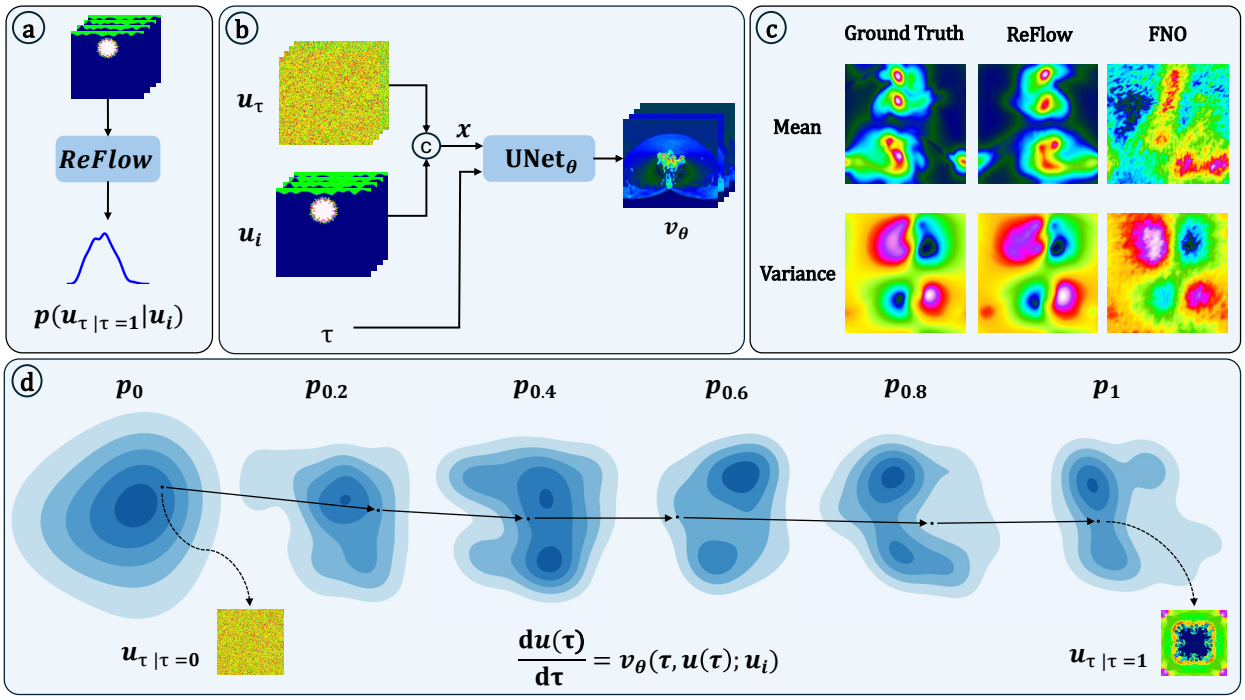


Figure 1: **Overview of ReFlow.** (**a**): ReFlow closely matches high-order statistical moments (in the Wasserstein W_1-sense) of fluid flows by learning a neural surrogate to approximate the push-forward of the distribution $p(u_{\tau|\tau=0}|u_i)$, while minimizing inference-time computational costs. (**b**) Macro-architecture overview: A UViT based network predicts the velocity field v_θ, conditioned on Gaussian noise, initial data u_i and boundary conditions, and diffusion time τ. Noise and initial / boundary conditions are concatenated as inputs. (**c**): Results for the cylindrical shear flow dataset. (**d**): The KDE plots monitor the evolution of a down-projected version of the push-forward measure $p_\tau := X_\theta(\tau)_\# p(u_{\tau=0}|u_i)$, where X_θ is the flow induced by the learned vector field v_θ. Results for the Richtmyer Meshkov dataset showing flow matching between noise ($p_{\tau=0}$) and target ($p_{\tau=1}$) distributions. ReFlow solves an ODE, learning a constant velocity field that enables straight-line inference paths (visualized with principal component analysis and kernel density estimation at different diffusion times).

training-time required to reach the same accuracy level with the conditional diffusion approaches; this is also confirmed by earlier work [2] exploring the scalability of rectified flows with transformer-based backbones.

4 Rectified Flows: An ODE-Based Alternative

4.1 Mathematical Formulation

Overview. Rectified flows provide a *deterministic* way to transform one probability distribution into another, offering an alternative to stochastic diffusion-based methods. Instead of traversing curved, noise-driven paths, rectified flows attempt to transport mass along nearly "straight" trajectories in state space, which can dramatically reduce sampling costs.

Throughout, our random variables are *state fields* $u : \mathcal{D} \to \mathbb{R}^m$ (e.g. $m = 2$ for velocity, or $m = 4$ for compressible state variables) living in a separable space $\mathcal{U} = L^2(\mathcal{D}; \mathbb{R}^m)$ (or H^s, $s > \frac{d}{2}$). All theory below extends verbatim from $\mathbb{R}^d$ because $\mathcal{U}$ is Polish; in practice we work with a Galerkin discretisation $\mathbb{R}^{d_N} \subset \mathcal{U}$, so every integral is evaluated by Monte-Carlo on its finite grid surrogate.

In all that follows, the *diffusion time* $\tau \in [0,1]$ is a purely *scale-ordering* variable: $\tau = 0$ denotes the coarsest, lowest–frequency representation of a state field, while $\tau = 1$ restores full spatial resolution; it is *not* the physical PDE time but merely an index that governs the progressive refinement of scales along the rectified flow.

Coupling γ as a statistical transport blueprint. Let $\mu_0, \mu_1 \in \mathcal{P}(\mathcal{U})$ be the initial and target field laws. Fix *any* coupling $\gamma \in \mathcal{P}(\mathcal{U} \times \mathcal{U})$ with $(\pi_0)_\#\gamma = \mu_0$ and $(\pi_1)_\#\gamma = \mu_1$ (in practice: the empirical pairing of input/output fields in the dataset). Think of $(U_0, U_1) \sim \gamma$ as drawing a *random chord* in state space, from U_0 to U_1.

For $\tau \in [0,1]$ define the linear chord point

$$U_\tau := (1-\tau)U_0 + \tau U_1, \quad \rho_\tau := \mathrm{Law}(U_\tau) = (T_\tau)_\#\gamma, \quad T_\tau(u_0, u_1) = (1-\tau)u_0 + \tau u_1.$$

Thus ρ_τ is the population of intermediate states obtained by moving along dataset chords by a fraction τ.

The barycentric velocity is the "average chord direction" at (u, τ). Along a single chord, the displacement $U_1 - U_0$ is constant in τ. However, many chords pass through the same intermediate state u at the same τ; the correct population-level velocity is the average direction among those chords. Concretely, define

$$v^\star(u,\tau) \;:=\; \mathbb{E}\big[\, U_1 - U_0 \,\big|\, U_\tau = u \,\big], \quad \rho_\tau\text{-a.e. } u. \tag{2}$$

This is the clean "one-line" definition of the barycentric field: it is a conditional expectation, so it is automatically the unique $L^2(\rho_\tau)$ minimizer of a regression problem.

Remark 4.1 (Disintegration form). If one prefers a geometric formula, (2) is equivalent to disintegrating γ along the map T_τ:

$$v^\star(u,\tau) = \int_{T_\tau^{-1}(u)} (u_1 - u_0)\, \gamma_u^\tau(du_0, du_1), \quad T_\tau^{-1}(u) = \{(u_0,u_1) : (1-\tau)u_0 + \tau u_1 = u\}.$$

Keeping this as a remark avoids leading with fiber notation.

Rectified-flow training = regression onto constant chord directions. Given samples $(u_0, u_1) \sim \gamma$ and $\tau \sim \mathrm{Unif}[0,1]$, form $u_\tau = (1-\tau)u_0 + \tau u_1$. Train v_θ by

$$\min_\theta \; \mathbb{E}\Big[\big\| v_\theta(u_\tau, \tau) - (u_1 - u_0) \big\|^2\Big]. \tag{3}$$

By the L^2 optimality of conditional expectation, the population minimizer of (3) is exactly $v^\star$ in (2). The key qualitative point is immediate: the regression target $(u_1 - u_0)$ does *not* depend on τ. So the learned field is pressured to behave as *time-stationary as the data allow* along these chords. This is the precise sense in which rectified-flow training promotes "straight" transports without separately adding an ad hoc straightness penalty.

Sampling = integrate the learned transport ODE. Once v_θ approximates $v^\star$, sampling amounts to integrating

$$\frac{du_\tau}{d\tau} = v_\theta(u_\tau, \tau), \quad u_0 \sim \mu_0, \tag{4}$$

and returning u_1 at $\tau = 1$. In the idealized chord model where $v^\star$ is constant in τ along typical trajectories, large steps are accurate; deviations from that ideal appear as *time-variation* of the velocity along the path, which is exactly what discretization error is sensitive to (see Lemma 4.4).

Curvature-aware control: estimate "how non-constant in τ" the velocity is. During sampling, write $v_t := v_\theta(u_t, \tau_t)$ and maintain an EMA trend

$$v_t^{\mathrm{ema}} = \lambda\, v_{t-1}^{\mathrm{ema}} + (1-\lambda)\, v_t, \quad \lambda \in (0,1). \tag{5}$$

Define the instantaneous straightness/curvature proxy

$$s_t := \|v_t - v_t^{\mathrm{ema}}\|_2,$$

which measures short-horizon time-variation of the velocity along the realized trajectory. Large s_t signals local bending / stiffness, precisely the regime where explicit large-step integration incurs higher local truncation error (again, Lemma 4.4).

A one-line regularized update (closed form). When s_t is large, replace v_t by a gently regularized velocity $\tilde{v}_t$ defined as the minimizer of

$$\tilde{v}_t \;:=\; \arg\min_{w} \; \|w - v_t\|_2^2 \;+\; \eta_t \, \|w - v_t^{\mathrm{ema}}\|_2^2,$$

which yields the explicit convex blend

$$\tilde{v}_t = (1-\alpha_t)v_t + \alpha_t v_t^{\mathrm{ema}}, \quad \alpha_t = \frac{\eta_t}{1+\eta_t} \in [0,1]. \tag{6}$$

Choosing η_t as a saturating function of s_t makes the correction dormant when the path is straight and active only when curvature spikes.

Adaptive step size from the same signal. Since explicit Euler LTE scales like $(\Delta\tau)^2$ times a norm of the time-variation of the vector field (Lemma 4.4), use s_t as a scale proxy and set

$$\Delta\tau_t \;\propto\; \frac{1}{\sqrt{\kappa_1 s_t + \kappa_2}}, \quad \Delta\tau_t \in [\Delta\tau_{\min}, \Delta\tau_{\max}], \tag{7}$$

with κ_2 absorbing baseline stiffness (e.g. dependence on $\|v_t\|$ and local Lipschitz effects). The state update is then

$$u_{t+1} = u_t + \Delta\tau_t \, \tilde{v}_t.$$

When the learned transport is locally straight (s_t small), the sampler automatically takes large steps and behaves like "chord shooting"; when curvature increases, it both damps the direction and contracts the step, preventing instability while retaining the large-step advantage in the straight regime.

4.2 Error Decomposition and Bounds

We now quantify why (6)–(7) help. Write the *ideal* and *learned* flows

$$\dot{u}_\tau = v_\star(u_\tau, \tau), \quad \dot{\hat{u}}_\tau = v_\theta(\hat{u}_\tau, \tau),$$

with the same u_0. Assume $v_\theta(\cdot, \tau)$ is L-Lipschitz in u and $v_\star \in L^2$.

Theorem 4.2 (Terminal error decomposition)**.** *Let U_τ solve $\dot{U}_\tau = v_\star(U_\tau, \tau)$ with $U_0 = u_0$, and let $\hat{u}_\tau$ solve $\dot{\hat{u}}_\tau = v_\theta(\hat{u}_\tau, \tau)$ with $\hat{u}_0 = u_0$. Define the time–averaged velocity $\bar{v}_\theta(u) := \int_0^1 v_\theta(u, \tau)\, d\tau$ and*

$$\varepsilon_{\mathrm{fit}}^2 := \int_0^1 \mathbb{E}\big\|\bar{v}_\theta(U_\tau) - v_\star(U_\tau, \tau)\big\|^2 \, d\tau, \quad \varepsilon_{\mathrm{curv}}^2 := \int_0^1 \mathbb{E}\big\|v_\theta(U_\tau, \tau) - \bar{v}_\theta(U_\tau)\big\|^2 \, d\tau.$$

Then, for the continuous–time flows,

$$\mathbb{E}\big\|\hat{u}_1 - u_1\big\| \;\le\; e^L \Big(\varepsilon_{\mathrm{fit}} + \varepsilon_{\mathrm{curv}}\Big). \tag{8}$$

Interpretation. The terminal error splits into a *reconstruction* term ($\varepsilon_{\mathrm{fit}}$) and a *straightness* term ($\varepsilon_{\mathrm{curv}}$). Blending (6) shrinks the within-time variation, directly tightening the bound; the adaptive stepping in (7) triggers corrections where the curvature proxy s_t spikes.

Corollary 4.3 (High-probability control of terminal error)**.** *Under the same assumptions as Theorem 4.2, suppose in addition that*

$$\int_0^1 \mathbb{E}\big\|v_\theta(U_\tau, \tau) - v_\star(U_\tau, \tau)\big\|^2 \, d\tau \;\le\; \varepsilon_{\mathrm{fit}}^2, \quad \int_0^1 \mathbb{E}\big\|v_\theta(U_\tau, \tau) - \bar{v}_\theta(U_\tau)\big\|^2 \, d\tau \;\le\; \varepsilon_{\mathrm{curv}}^2,$$

so that both the reconstruction error and straightness error are controlled in L^2. Then there exists a constant $C(L)$ such that

$$\mathbb{E}\big\|\hat{u}_1 - u_1\big\|^2 \;\le\; C(L)\,\big(\varepsilon_{\text{fit}}^2 + \varepsilon_{\text{curv}}^2\big). \tag{9}$$

Consequently, for any tolerance $\eta > 0$, Chebyshev's inequality yields

$$\mathbb{P}\big(\|\hat{u}_1 - u_1\| > \eta\big) \;\le\; \frac{C(L)\,\big(\varepsilon_{\text{fit}}^2 + \varepsilon_{\text{curv}}^2\big)}{\eta^2}. \tag{10}$$

In particular, as training drives $\varepsilon_{\text{fit}}, \varepsilon_{\text{curv}} \to 0$, the probability of a large terminal error event decays at least quadratically in the target radius η.

Proof sketch. The proof follows the same Grönwall argument as Theorem 4.2, but carried out in L^2 instead of L^1, yielding $\mathbb{E}\|\hat{u}_1 - u_1\|^2 \le C(L)(\varepsilon_{\text{fit}}^2 + \varepsilon_{\text{curv}}^2)$ for some $C(L) \sim e^{2L}$. Chebyshev's inequality then gives $\mathbb{P}(\|\hat{u}_1 - u_1\| > \eta) \le \mathbb{E}\|\hat{u}_1 - u_1\|^2/\eta^2$, which is the claimed bound. A full proof (including the sharp dependence on the straightness functional) is deferred to the theory appendix. □

Lemma 4.4 (Local truncation error for explicit Euler on $\dot{u} = v(u,\tau)$)**.** *Assume $v \in C^1$ in (u,τ) on the region traversed and that $\|\partial_\tau v(u,\tau)\| \le M_\tau$, $\|J_u v(u,\tau)\| \le L$, and $\|v(u,\tau)\| \le M$. Then one explicit-Euler step from (u,τ) with step size $\Delta\tau$ has local truncation error*

$$\begin{aligned}
\mathrm{LTE}(\tau;\Delta\tau) &:= \big\|u(\tau+\Delta\tau) - \big(u(\tau) + \Delta\tau\, v(u(\tau),\tau)\big)\big\| \\
&= \frac{(\Delta\tau)^2}{2}\,\big\|\partial_\tau v(u,\tau) + J_u v(u,\tau)\, v(u,\tau)\big\| \;+\; \mathcal{O}\big((\Delta\tau)^3\big) \\
&\le \frac{(\Delta\tau)^2}{2}\,(M_\tau + LM) \;+\; \mathcal{O}\big((\Delta\tau)^3\big).
\end{aligned} \tag{11}$$

From Lemma 4.4 to the step rule. The leading LTE term depends on $\|\partial_\tau v\|$ and $\|J_u v\, v\|$. The EMA deviation $s_t = \|v_t - v_t^{\text{ema}}\|$ is a computable proxy for the (local) time variation $\|\partial_\tau v\|$ (up to a scale factor tied to the EMA bandwidth), while $L\,\|v_t\|$ upper-bounds the spatial contribution. Enforcing LTE $\lesssim$ tol yields the square-root rule (7), with κ_1 absorbing the EMA bandwidth and $\kappa_2 \approx L\|v_t\|$. Under blending, $\|\,\tilde{v}_t - v_t^{\text{ema}}\| = (1-\alpha_t)s_t$, so the curvature term and the required step shrink *together*, stabilizing integration where needed and enlarging steps elsewhere. See **SM** E for proofs and additional theoretical context.

4.3 Spectral Capacity, Structure Functions, and Law-Level Coverage

Functional setting. We work on the periodic box $\mathcal{D} = \mathbb{T}^d$ and consider L^2 fields $\mathcal{H} := L^2(\mathcal{D};\mathbb{R}^m)$ with norm $\|\cdot\|_2$. For incompressible velocity fields (e.g. SL2D, see Section 6), one may equivalently restrict to divergence-free subspaces and adopt the statistical-solution framework of [13], which characterizes laws via correlation identities, admissible observables, and an energy inequality. In that framework the natural distance on laws is the L^2-based Wasserstein metric W_2, and small-scale regularity is expressed in terms of *structure functions* and their time-averages. The numerical ensembles used in our experiments fit this setting: they are generated by dissipative solvers on $\mathbb{T}^2$, have uniformly bounded energy, and exhibit empirical structure-function scaling (equivalently, power-law spectra) consistent with the assumptions below.

4.3.1 Bandlimited capacity and strain constraints

Any $u \in \mathcal{H}$ admits a Fourier series $u(x) = \sum_{k\in\mathbb{Z}^d} \hat{u}(k)e^{ik\cdot x}$. We write $P_{\le K}$ for the projector onto modes $\{k : |k| \le K\}$ and $P_{>K} = I - P_{\le K}$. On a grid of size N^d, the Nyquist frequency satisfies $K_{\max} \simeq N$, so every numerically represented field is effectively bandlimited to $|k| \lesssim K_{\max}$. Architecturally, downsampling and anti-aliasing may enforce an even stricter cutoff $K_c \le K_{\max}$.

The next lemma is a standard Bernstein-type inequality: bandlimited fields have a uniform Lipschitz cap.

Lemma 4.5 (Bernstein upper bound for bandlimited fields)**.** *Let $u : \mathbb{T}^d \to \mathbb{R}^m$ satisfy $u = P_{\le K}u$ for some integer $K \ge 1$. Then there exists a constant $C_d > 0$ depending only on the dimension d such that*

$$\|\nabla_x u\|_{L^\infty(\mathbb{T}^d)} \;\le\; C_d\, K^{1+\frac{d}{2}}\, \|u\|_{L^2(\mathbb{T}^d)}. \tag{12}$$

In particular, on an energy shell $\|u\|_{L^2} \le E^{1/2}$ *one has*

$$\|\nabla_x u\|_{L^\infty} \;\le\; C_d\, K^{1+\frac{d}{2}}\, E^{1/2}. \tag{13}$$

Proof sketch. Write $u(x) = \sum_{|k|\le K} \hat{u}(k) e^{ik\cdot x}$, so $\nabla_x u(x) = \sum_{|k|\le K} (ik)\hat{u}(k) e^{ik\cdot x}$ and $|\nabla_x u(x)| \le \sum_{|k|\le K} |k|\,|\hat{u}(k)|$. Cauchy–Schwarz and a lattice-counting bound $\#\{|k| \le K\} \lesssim K^d$ give $\|\nabla u\|_{L^\infty} \lesssim (\sum_{|k|\le K} |k|^2)^{1/2} \|\hat{u}\|_{\ell^2} \lesssim K^{1+d/2} \|u\|_{L^2}$. Full details are deferred to the theory appendix. □

Upper bounds alone do not say what happens if a surrogate *misses* an entire high-frequency annulus. The next lemma gives a complementary lower bound.

Lemma 4.6 (Annulus lower bound)**.** *Let* $f = P_{[K,2K]} u$ *for some* $u \in L^2(\mathbb{T}^d)$*, where* $P_{[K,2K]} := P_{\le 2K} - P_{\le K}$ *and* $K \ge 1$*. Then there exists* $c_d > 0$ *such that*

$$\|\nabla_x f\|_{L^\infty} \;\ge\; c_d\, K\, \|f\|_{L^\infty} \;\ge\; c_d\, K^{1-\frac{d}{2}}\, \|f\|_{L^2}. \tag{14}$$

Proof sketch. On the annulus $\{K \le |k| \le 2K\}$, the symbol of ∇ has magnitude $|k| \in [K, 2K]$. This yields $\|\nabla f\|_{L^\infty} \gtrsim K \|f\|_{L^\infty}$. A reverse Bernstein estimate on an annulus gives $\|f\|_{L^\infty} \gtrsim K^{-d/2} \|f\|_{L^2}$. See the theory appendix for a precise statement independent of the Littlewood–Paley cut-off choice. □

Corollary 4.7 (Capacity requirement for typical strain levels)**.** *Suppose the target flows at time* t *have typical energy* E *and typical rate-of-strain* $\|\nabla_x u\|_{L^\infty} \simeq L_\star$*. Let a surrogate produce only bandlimited outputs* $u = P_{\le K_c} u$ *with* $\|u\|_{L^2} \le E^{1/2}$*. Then necessarily*

$$C_d\, K_c^{1+\frac{d}{2}}\, E^{1/2} \;\gtrsim\; L_\star, \tag{15}$$

and hence a minimal cutoff

$$K_c \;\gtrsim\; \left(\frac{L_\star}{E^{1/2}}\right)^{\frac{2}{2+d}}. \tag{16}$$

Moreover, if the true flow has nontrivial energy in an annulus above K_c*, then any bandlimited approximation underestimates its strain by at least* $c_d K^{1-d/2} \|P_{[K,2K]} u\|_{L^2}$ *for some* $K > K_c$*.*

Remark 4.8. Corollary 4.7 formalises the intuition that spatial capacity (K_c) is a hard constraint: at fixed energy scale E, there is a minimum cutoff needed to represent typical strain levels $L_\star$. In our experiments both GenCFD and ReFlow operate at the same grid resolution, so they share the same effective K_c; rectification does not invent new small scales but makes the trajectories inside this bandlimited class straight and easy to integrate.

4.3.2 Structure functions and small-scale coverage

We now quantify how much of the *law* of the PDE solution can be captured at a given cutoff K_c. For $u : \mathbb{T}^d \to \mathbb{R}^m$ and $r > 0$ define the second-order structure function

$$S_r^2(u) := \frac{1}{|B_r|} \int_{|h|\le r} \|u(\cdot + h) - u(\cdot)\|^2_{L^2(\mathbb{T}^d)}\, dh,$$

and for a probability law μ on $\mathcal{H}$ set

$$S_r^2(\mu) := \int S_r^2(u)\, d\mu(u).$$

In homogeneous turbulence $S_r^2(\mu)$ is directly related to the energy spectrum; power-law behaviour $S_r^2(\mu) \sim r^{\zeta_2}$ at small r encodes the scaling of the cascade. In our experiments, Figure 13 visualizes this connection directly: matching the high-wavenumber tail in the spectrum is equivalent to capturing the small-r behaviour of structure functions, and therefore to controlling the coverage term below.

Guided by the statistical-solution framework of [4], we assume there exists a modulus of continuity $\omega : (0, 1] \to [0, \infty)$ with $\omega(r) \downarrow 0$ as $r \downarrow 0$ such that

$$S_r^2(\mu_t) \;\le\; \omega(r) \quad \text{for all } r \in (0, 1],\ t \in [0, T]. \tag{17}$$

In practice, ω can be taken as a piecewise power law $Cr^{2\alpha}$ calibrated from the empirical spectra of the numerical ensembles. The next lemma shows that (17) controls the high-frequency tail of the law.

Lemma 4.9 (Structure function controls Fourier tail)**.** *Let μ be a law on $L^2(\mathbb{T}^d)$ satisfying $S_r^2(\mu) \le \omega(r)$ for all $r \in (0,1]$, where $\omega(r) \downarrow 0$ as $r \downarrow 0$. Then there exist constants $c_d, C_d > 0$ such that for all integers $K \ge 1$,*

$$\int \|P_{>K} u\|_{L^2}^2 \, d\mu(u) \;\le\; C_d\, \omega\big(c_d/K\big). \tag{18}$$

Proof sketch. Expand u in Fourier series and observe $\|u(\cdot+h)-u(\cdot)\|_{L^2}^2 = \sum_k |\hat{u}(k)|^2 |e^{ik\cdot h}-1|^2$. Averaging over $h \in B_r$ yields $S_r^2(u) = \sum_k |\hat{u}(k)|^2 \Psi_r(k)$, where $\Psi_r(k) \gtrsim \min\{(|k|r)^2, 1\}$ uniformly in k, r. For $|k| \gtrsim 1/r$, one has $\Psi_r(k) \gtrsim 1$, hence $S_r^2(u) \gtrsim \|P_{>\,c/r} u\|_{L^2}^2$. Integrate over $u \sim \mu$, apply (17), and set $r \sim 1/K$. Full constants and the Littlewood–Paley formulation are deferred to the theory appendix. □

Lemma 4.10 (Bandlimited projection in W_2)**.** *For any law μ on $L^2(\mathbb{T}^d)$ and cutoff $K_c \ge 1$,*

$$W_2\big(\mu,\, (P_{\le K_c})_{\#}\mu\big) \;\le\; \Big(\int \|P_{>K_c} u\|_{L^2}^2 \, d\mu(u)\Big)^{1/2}. \tag{19}$$

Proof. Couple $u \sim \mu$ and $\tilde{u} = P_{\le K_c} u$. Then $\tilde{u} \sim (P_{\le K_c})_{\#}\mu$ and $W_2^2(\mu, (P_{\le K_c})_{\#}\mu) \le \mathbb{E}\|u-\tilde{u}\|_{L^2}^2 = \mathbb{E}\|P_{>K_c} u\|_{L^2}^2$. □

Combining Lemmas 4.9 and 4.10 gives a clean coverage bound.

Corollary 4.11 (Coverage term from the structure-function modulus)**.** *Let μ_t be the statistical solution law at time t and assume (17). Then for any cutoff $K_c \ge 1$,*

$$W_2\big(\mu_t,\, (P_{\le K_c})_{\#}\mu_t\big) \;\lesssim\; \omega\big(c_d/K_c\big)^{1/2}, \tag{20}$$

with c_d as in Lemma 4.9.

4.3.3 Capacity–coverage–fit decomposition for one-step RF

Finally, we combine the coverage term with the training error of a one-step rectified-flow surrogate.

Let $S_{\Delta t} : \mathcal{H} \to \mathcal{H}$ denote the PDE solution operator over a physical step Δt, and write

$$\mu_{t+\Delta t} := (S_{\Delta t})_{\#}\mu_t.$$

Assume the rectified-flow propagator $\mathcal{T}_{\Delta t}^{\theta}$ aims at the bandlimited target $P_{\le K_c}(S_{\Delta t} u)$ in mean-square. Concretely, sample $u \sim \mu_t$ and an internal noise seed $\xi \sim \mathcal{N}(0, I)$, and define the RF output $\widehat{u} := \mathcal{T}_{\Delta t}^{\theta}(u, \xi)$. We measure the training fit by

$$\varepsilon_{\text{train}}^2(t) := \mathbb{E}_{u\sim\mu_t}\, \mathbb{E}_{\xi\sim\mathcal{N}(0,I)} \big\| P_{\le K_c}(S_{\Delta t} u) - \widehat{u} \big\|_{L^2}^2. \tag{21}$$

Finally, we define the one-step RF law by

$$\widehat{\mu}_{t+\Delta t} := \mathrm{Law}(\widehat{u}).$$

Proposition 4.12 (One-step law-level error: capacity, coverage, and fit)**.** *Under the structure-function modulus assumption (17), the one-step W_2 error between the PDE law and the RF law satisfies*

$$W_2\big(\mu_{t+\Delta t},\, \widehat{\mu}_{t+\Delta t}\big) \;\le\; \Big(C_d\, \omega\big(c_d/K_c\big) \;+\; \varepsilon_{\text{train}}^2(t)\Big)^{1/2}, \tag{22}$$

where the first term is a coverage *error determined by the structure-function modulus and the cutoff K_c, and the second is a* fit *error determined by training.*

Proof sketch. Couple $u \sim \mu_t$ and $\widehat{u}$ using the same input; then $W_2^2(\mu_{t+\Delta t}, \widehat{\mu}_{t+\Delta t}) \le \mathbb{E}\|S_{\Delta t} u - \widehat{u}\|_{L^2}^2$. Insert $P_{\le K_c} S_{\Delta t} u$, apply the triangle inequality, and bound the tail term by Lemma 4.9 (applied at time $t+\Delta t$). □

Interpretation and experimental validation. At a fixed resolution (fixed K_c), Proposition 4.12 says that even a perfectly trained rectified flow cannot beat the coverage term $\omega(c_d/K_c)^{1/2}$ dictated by small-scale physics. This is precisely why spectrum matching (Figure 13) is a non-negotiable diagnostic on multiscale flows: if the surrogate loses the high-frequency tail, it is *necessarily* paying a coverage penalty in law. The remaining error is controlled by the training fit and the straightness properties of the learned velocity (via the continuous-time bounds in Section 4.1). Rectified flows improve efficiency precisely by reducing the curvature term in these bounds, so that a given coverage+fit accuracy can be reached with many fewer ODE steps than a diffusion-based surrogate such as GenCFD (Section 5.6).

5 Correctness and Curvature-Controlled Discretization

Organization. This section states the main correctness and discretization results that underpin the narrative above. Full proofs are deferred to the theory appendix (to be written separately); we keep only the steps needed to make the dependence on *fit*, *straightness*, and *discretization* explicit in the main text.

5.1 Exact transport under the barycentric rectified velocity

Proposition 5.1 (Exact transport under the barycentric velocity)**.** *Let $\mu_0, \mu_1 \in \mathcal{P}(\mathcal{U})$ and let $\gamma \in \mathcal{P}(\mathcal{U} \times \mathcal{U})$ be a coupling of (μ_0, μ_1). Define $T_\tau(u_0, u_1) = (1-\tau)u_0 + \tau u_1$ and $\rho_\tau := (T_\tau)_\# \gamma$. Let $v_\star(\cdot, \tau)$ be the barycentric velocity field of defined in 2. Assume (e.g. on a finite-dimensional Galerkin discretization of $\mathcal{U}$) that the continuity equation*

$$\partial_\tau \rho_\tau + \nabla \cdot (\rho_\tau \, v_\star(\cdot, \tau)) = 0 \tag{23}$$

admits a unique regular Lagrangian flow X_τ associated to $\dot{u}_\tau = v_\star(u_\tau, \tau)$. Then $(X_1)_\# \mu_0 = \mu_1$.

Proof sketch. By construction, ρ_τ is the law of $U_\tau = (1-\tau)U_0 + \tau U_1$ for $(U_0, U_1) \sim \gamma$. Differentiate $\mathbb{E}[\varphi(U_\tau)]$ and condition on U_τ to identify the weak form of (23). Uniqueness of the continuity equation solution (in the chosen Galerkin setting) implies $\rho_\tau = (X_\tau)_\# \rho_0$, hence $\rho_1 = (X_1)_\# \rho_0$, i.e. $\mu_1 = (X_1)_\# \mu_0$. □

Why this matters. Proposition 5.1 isolates a clean correctness statement: if v_θ matches the barycentric rectified velocity $v_\star$ sufficiently well (in a sense quantified by Theorem 4.2), the induced flow pushes forward μ_0 to μ_1 at the level of laws.

5.2 Curvature controls the required number of ODE steps

Corollary 5.2 (Global Euler error controlled by time-curvature)**.** *Assume the hypotheses of Lemma 4.4 for $\dot{u} = v(u, \tau)$ on $\tau \in [0, 1]$ and assume $v(\cdot, \tau)$ is L-Lipschitz in u uniformly in τ. Let u^Δ be the explicit-Euler approximation with uniform step $\Delta\tau = 1/N$. Then there exists a constant $C = C(L)$ such that*

$$\sup_{\tau \in [0,1]} \|u_\tau^\Delta - u_\tau\| \;\le\; C \, \Delta\tau \int_0^1 \Big(\|\partial_\tau v(u_\tau, \tau)\| + \|J_u v(u_\tau, \tau) \, v(u_\tau, \tau)\| \Big) \, d\tau. \tag{24}$$

In particular, when $\int_0^1 \|\partial_\tau v(u_\tau, \tau)\| \, d\tau$ is small ("straight" flow), the N required to reach a fixed tolerance is proportionally smaller.

Proof sketch. Sum the local truncation errors from Lemma 4.4 over N steps and propagate them using stability of the flow under the L-Lipschitz condition via Grönwall. □

Connection to our curvature proxy. The EMA deviation $s_t = \|v_t - v_t^{\text{ema}}\|$ used in (6)–(7) is a computable proxy for the same time-variation that drives (24). Blending reduces effective curvature; step control reduces $\Delta\tau$ when curvature spikes. This directly rationalizes why ReFlow can reach the "spectrum-correct" regime in very few steps (Figure 13), and why the same accuracy level is reached at 8 steps (Table 1) whereas diffusion baselines typically need ≥ 128.

5.3 A compact regularity interpretation of structure functions

Lemma 5.3 (Structure functions imply Besov regularity (heuristic equivalence))**.** *Let $u \in L^2(\mathbb{T}^d)$ and define the increment seminorm*

$$[u]_{B^s_{2,\infty}} := \sup_{0<|h|\le 1} \frac{\|u(\cdot+h)-u(\cdot)\|_{L^2}}{|h|^s}.$$

If the second-order structure function satisfies $S_r^2(u) \lesssim r^{2s}$ for all $r \in (0,1]$, then $[u]_{B^s_{2,\infty}} \lesssim 1$. Consequently, a law-level modulus bound $S_r^2(\mu) \lesssim r^{2s}$ implies that typical samples from μ have Besov smoothness s, consistent with the Fourier-tail decay in Lemma 4.9.

Remark 5.4. In turbulence one often observes $S_r^2(\mu) \sim r^{\zeta_2}$ over an inertial range. Heuristically this corresponds to $s \approx \zeta_2/2$, so the coverage term in Corollary 4.11 can be read as a direct proxy for inertial-range scaling.

5.4 Master inequality

Theorem 5.5 (Law-level error: coverage + fit + straightness + discretization)**.** *Fix an effective resolution cutoff K_c and assume the structure-function modulus hypothesis used in Corollary 4.11. Let $\hat{\mu}_1^{(N)}$ be the law obtained by taking N explicit-Euler steps to integrate the learned rectified ODE $\dot{u}_\tau = v_\theta(u_\tau,\tau)$ from $\tau=0$ to $\tau=1$. Assume $v_\theta(\cdot,\tau)$ is L-Lipschitz in u uniformly in τ. Then, for constants depending on L,*

$$W_2\big(\mu_1,\hat{\mu}_1^{(N)}\big) \lesssim \underbrace{\omega(c_d/K_c)^{1/2}}_{\textit{coverage}} + \underbrace{\varepsilon_{\text{fit}}+\varepsilon_{\text{curv}}}_{\textit{fit + straightness (Thm. 4.2)}} + \underbrace{\frac{1}{N}\int_0^1 \|\partial_\tau v_\theta(u_\tau,\tau)\|\,d\tau}_{\textit{discretization (Cor. 5.2)}}. \tag{25}$$

Proof sketch. The coverage term is Theorem 4.11. The continuous-time mismatch between the ideal rectified flow and the learned flow is controlled by Theorem 4.2 via ε_{fit} and $\varepsilon_{\text{curv}}$. The discretization term follows from Corollary 5.2 applied to the learned ODE. □

Reading (25) against the experiments. Equation (25) matches the empirical breakdown seen in Section 6: (i) *coverage* corresponds to reproducing the high-wavenumber energy content (Figure 13); (ii) *fit* corresponds to the overall statistical accuracy, reflected in e_μ, e_σ, and W_1 (Table 1); (iii) *straightness* and *discretization* are exactly what turn "a good model" into "a fast sampler": ReFlow reaches the same law-level quality using $N=8$ steps, whereas diffusion baselines require $N \ge 128$ (Section 5.6).

5.5 Implementation Details for PDEs

Learning velocity fields in PDE settings. When rectified flows are used to model PDE solutions, we assume access to paired initial/final states (u_i, u_f) where $u_f = \mathcal{S}(u_i)$ represents the evolved initial condition u_i, at some final physical time (e.g. $t=1$). We train a neural network to predict the velocity v_θ that displaces u_i towards u_f, with the additional mention that v_θ shall be conditioned on u_i at all diffusion times.

Overview of the model. Our network learns a velocity (or denoising) map

$$v_\theta(u_\tau, u_0, \tau) \;=\; \mathrm{UNet}_\theta\big(u_\tau, u_0, \Gamma(\tau)\big),$$

where u_τ is a low-frequency representation of u_1, u_0 is the initial datum, and $\Gamma(\cdot)$ is an MLP-based time embedding:

$$\Gamma(\tau) \;=\; \mathrm{MLP}\big(\mathrm{Sinusoid}(\tau)\big).$$

Within the U-Net (see **SM** J.1), we alternate residual blocks with multi-head attention layers, downsampling to a bottleneck and then upsampling via skip connections. A final 1×1 projection recovers either the field increment $\hat{u} \approx u_1 - u_0$ or a noise residual. By injecting the rectification time $\Gamma(\tau)$ into each block, the model performs a fully deterministic trajectory integration instead of stochastic diffusion.

Algorithm 1 Rectified Flow Training (PDE Context)

Require: Dataset $\{(u_0, u_1)\}$ with $u_1 = \mathcal{S}(u_0)$; velocity model v_θ; noise schedule $\sigma(\tau)$

1: **for** each iteration **do**
2: Sample a pair (u_0, u_1)
3: Sample $\tau \sim \mathcal{U}(0,1)$ and noise $\xi \sim \mathcal{N}(0, I)$
4: Form $u_\tau = \tau \cdot u_1 + \sigma(1-\tau)\,\xi$
5: Compute loss:

$$\mathcal{L}(\theta) \;=\; \big\|\big(u_1 - u_0\big) \;-\; v_\theta\big(u_\tau,\, u_0,\, \tau\big)\big\|^2$$

6: Update θ via gradient descent
7: **end for**

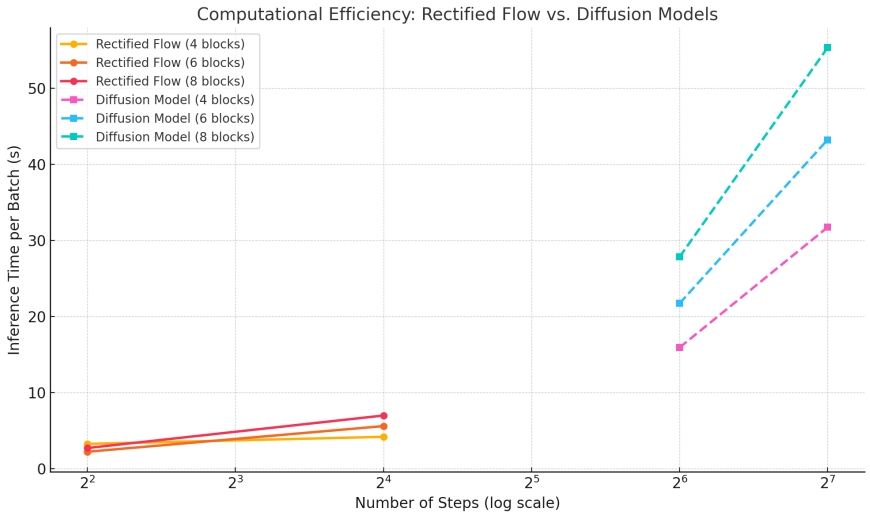


Figure 2: Inference cost versus model size. Rectified flows need 4–8 ODE steps vs. GenCFD's 64–128 SDE steps. U/Net depth varies from 4 to 8 blocks (parameters 5.5 M to 10.2 M).

Sampling procedure. Once trained, we generate new samples by numerically integrating the ODE

$$\frac{du_\tau}{d\tau} = v_\theta\big(u_\tau,\, u_0,\, \tau\big), \quad u_\tau|_{\tau=0} = \xi, \quad \xi \sim \mathcal{N}(0, I).$$

Any standard ODE solver (e.g. forward Euler, Runge-Kutta) can be used; rectified flows often tolerate relatively large step sizes due to their straighter velocity field.

Efficiency and accuracy. Because rectified flows explicitly encourage a "straighter" velocity field, accurate solutions often require fewer integration steps than their diffusion-based counterparts. This advantage can be crucial in computational fluid dynamics and other PDE applications where each solver step is expensive. See **SM** M.3 and P for a detailed comparison of step-count vs. accuracy trade-offs.

5.6 Computational Efficiency Comparison

Diffusion surrogates (e.g. GenCFD) generate samples by integrating a reverse-time SDE whose trajectory bends through latent noise; even with accelerated samplers such as DDPM or DPM-solver the procedure still requires tens of score-network evaluations. Rectified flows, in contrast, follow a single deterministic ODE whose nearly straight path tolerates large or adaptive steps, so inference needs only a handful of evaluations. Figure 2 quantifies this gap, showing how the runtime of diffusion models scales almost linearly with the number of steps, whereas rectified flows remain fast even as network size grows.

6 Experiments

6.1 Experimental Setup

We benchmark our method, termed henceforth as *ReFlow*, against four neural surrogate models. *GenCFD* and *GenCFD* $\circ$ *FNO* are conditional diffusion surrogates following the framework introduced earlier in this section, whereas FNO and UViT serve as one-shot deterministic baselines. Detailed descriptions of all baselines are provided in **SM** L.

Datasets for Training and Evaluation. We consider three canonical 2D multi-scale PDE problems (see the Supplementary Material for precise mathematical formulations and solver details):

SL2D *Shear Layer in 2D.* This setup models the evolution of an incompressible shear layer, initially concentrated near a horizontal interface, which rolls up into intricate vortical structures over time. We generate approximate solutions by numerically solving the 2D Navier–Stokes system (with periodic boundaries) at a resolution of 512^2, then downsample to 128^2 for training and testing. The data has been generated with the spectral viscosity code [28, 25].

CS2D *Cloud-Shock Interaction in 2D.* In this popular compressible-flow test, a shock wave moves through a higher-density "cloud" region, triggering complex wave interactions and small-scale turbulent mixing in the wake. We solve the compressible Euler system at an effective resolution of 256^2 using a GPU-optimized finite-volume ALSVINN code of [19, 5], then also downsample to 128^2.

RM2D *Richtmeyer–Meshkov in 2D.* In this case, a shock wave interacts with a density interface separating two fluids, depositing vorticity and triggering vigorous mixing. We numerically solve the compressible Euler system on a 256^2 grid, then downsample to 128^2 to create the training and test data. The data has been generated with a high-resolution finite-volume ALSVINN code of [5]

Training and Test Protocol. We train both our proposed rectified flow-based GenCFD-class algorithm and all baseline models on initial conditions

$$u_0 \sim \mu, \quad \text{with} \quad \mu$$

being the probability distribution specified in the Supplementary Material (**SM**. K.2). For each u_0, the PDE solver (e.g. a spectral or finite-volume code) is run forward in time to produce high-fidelity snapshots, which serve as training data.

Training-objective. Crucially, during training, the model is **only** exposed to (u_0, u_1) pairs of data as opposed to the conditional measure $p(u(t)|u_0)$.

Test-time goal: Dirac initial measures. While the training distribution μ is spread out, we are especially interested in the scenario where the *initial condition is fixed and deterministic.* Formally, we set

$$\widehat{\mu}_0 = \delta_{\widehat{u}_0},$$

which is a Dirac measure concentrated at a single function $\widehat{u}_0$. Thus, any stochasticity in the resulting flow arises solely from chaotic or multi-scale evolution over time, rather than from large variations in the initial condition itself.

To create a "ground truth" ensemble for comparison under a fixed $\widehat{u}_0$, we employ an *ensemble perturbation* procedure (detailed in **SM** K.2), where $\widehat{u}_0$ is slightly perturbed to generate multiple nearby initial states. Each perturbed state is then advanced using a high-resolution PDE solver (e.g. the Azeban code for incompressible flows, or a finite-volume code for compressible problems), yielding a set of flow realizations. This set approximates the conditional measure in question, i.e. $p\big(u(t) \mid \widehat{u}_0\big)$.

Distribution settings. We test two main configurations:

1. **Matched distribution.** We sample $\widehat{u}_0$ from the same family μ used in training. While $\delta_{\widehat{u}_0}$ is technically out-of-distribution in the sense that the training distribution was spread out over many initial states, the base function $\widehat{u}_0$ still belongs to the same general class of flows.

2. **Distribution shift.** We consider Dirac initial data $\delta_{\widehat{u}_0}$ where $\widehat{u}_0 \sim \nu$, for a *different* distribution $\nu \neq \mu$. This scenario probes the ability of our model and baselines to generalize under shifts in the initial condition distribution. Specific details on ν are provided in **SM** K.2.

We will show results in both regimes to assess how well rectified flows can capture the spread of chaotic solutions starting from fixed initial states—and how it compares to baselines.

6.2 Evaluation Metrics

We evaluate the generative performance of each method using three main metrics, with details provided in **SM** M:

- **Mean Error.** We measure the L^2 difference between the model's predicted mean flow μ and the ground-truth mean μ_{exact}. A low mean error indicates accurate large-scale behavior.
- **Standard Deviation Error.** The normalized L^2 distance between the predicted standard deviation σ and the reference σ_{exact} captures whether the model reproduces correct spatial variability.
- **One-Point Wasserstein Distance.** We compute an Wasserstein-1 distance at each spatial point (and average over the domain) to quantify local distributional agreement between ground truth and model outputs.

One-point Wasserstein as a $k=1$ marginal metric. Let μ and $\widehat{\mu}$ be laws on fields $u : D \to \mathbb{R}^m$. For each $x \in D$ define $\mu_x := (\text{ev}_x)_{\#}\mu$ and $\widehat{\mu}_x := (\text{ev}_x)_{\#}\widehat{\mu}$. Our reported metric is

$$W_1^{(1)}(\mu, \widehat{\mu}) := \int_D W_1(\mu_x, \widehat{\mu}_x)\, dx.$$

By Kantorovich–Rubinstein duality, $W_1(\mu_x, \widehat{\mu}_x) = \sup_{\text{Lip}(\varphi)\leq 1} \left| \int \varphi\, d\mu_x - \int \varphi\, d\widehat{\mu}_x \right|$, so $W_1^{(1)}$ controls the averaged discrepancy of *all* pointwise 1-Lipschitz observables. This is the precise sense in which Table 1 measures agreement of $k=1$ correlation statistics, in line with the statistical-solution viewpoint [13].

Baselines and Evaluation Settings. To stay within reasonable training times for our infrastructure, we restrict the total number of training iterations up to 120,000 for each model and recorded the best achievable results. Under this budget, only a reduced-capacity GenCFD (with two downsampling stages in its U-Net backbone, versus three in Rectified Flow) can match the accuracy reported in Table 1. We also evaluated larger GenCFD variants trained for 500,000 iterations, but observed no meaningful improvement over the results shown here. Meanwhile, at inference time Rectified Flow requires only **8** denoising steps to achieve comparable or superior accuracy, whereas diffusion baselines (including GenCFD) need **128** steps, underscoring RF's efficiency in both training and sampling. FNO and UViT also require smaller models than ReFlow for optimal performance.

Since FNO, and UViT produce *deterministic* predictions, we approximate their spread by an *ensemble perturbation* strategy (**SM** K.2), where each baseline is fed slightly perturbed inputs and outputs are aggregated to form a sample set. In contrast, both the original GenCFD (baseline) and our *ReFlow* are inherently *probabilistic*, requiring no auxiliary procedure to generate diverse samples.

6.3 Results.

Table 1 compares five surrogate models (ReFlow, together with the four baselines) on the three PDEs learning tasks using the mean error (e_μ), spread error or *standard deviation* (e_σ) and the distribution-level

Table 1: Comparison of models across datasets CS2D, SL2D, and RM2D, in terms of mean relative error (e_μ), standard deviation relative error (e_σ), and Wasserstein-1 distance (W_1). Best-performing model is colored in Blue and second-best performing model in Orange.

Model	**Metric**	**CS2D**				**SL2D**		**RM2D**			
		ρ	m_x	m_y	E	u_x	u_y	ρ	m_x	m_y	E
ReFlow	e_μ	0.0477	0.0332	0.041	0.015	0.034	0.189	0.018	0.032	0.020	0.021
	e_σ	0.1005	0.072	0.078	0.0584	0.071	0.077	0.032	0.046	0.046	0.085
	W_1	0.0091	0.0107	0.0124	0.0143	0.0214	0.0164	0.0033	0.0036	0.0028	0.0017
GenCFD	e_μ	0.0830	0.0619	0.127	0.0280	0.039	0.267	0.0049	0.0010	0.0025	0.0025
	e_σ	0.195	0.197	0.244	0.169	0.092	0.072	0.0351	0.0254	0.0435	0.0435
	W_1	0.0151	0.0184	0.0185	0.0247	0.0276	0.0213	0.0092	0.0016	0.0049	0.0049
UViT	e_μ	0.111	0.078	0.127	0.0479	0.233	0.623	0.070	0.070	0.071	0.052
	e_σ	0.415	0.343	0.412	0.296	0.28	0.178	0.425	0.570	0.570	0.617
	W_1	0.087	0.085	0.158	0.172	0.270	0.256	0.060	0.055	0.060	0.036
FNO	e_μ	0.0989	0.0748	0.0930	0.0383	0.058	0.408	0.085	0.070	0.063	0.067
	e_σ	0.2573	0.2779	0.3188	0.2777	0.137	0.110	0.366	0.443	0.436	0.450
	W_1	0.1455	0.1660	0.1879	0.1884	0.184	0.150	0.096	0.051	0.056	0.062
GenCFD ∘ FNO	e_μ	0.0638	0.0454	0.0603	0.0204	0.055	0.393	0.040	0.027	0.039	0.041
	e_σ	0.1099	0.0969	0.1245	0.0974	0.131	0.100	0.091	0.075	0.091	0.090
	W_1	0.0467	0.0534	0.0774	0.0743	0.166	0.135	0.0269	0.0158	0.0293	0.0273

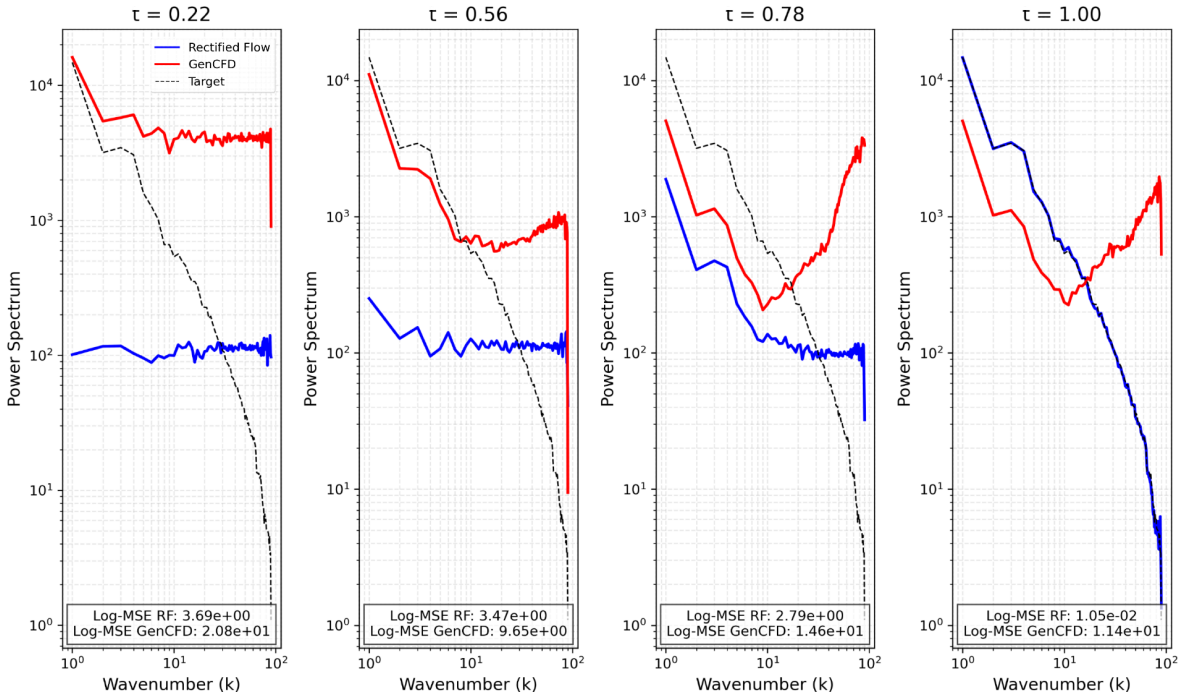


Figure 3: Energy-spectrum evolution for a random cloud–shock initial datum. ReFlow (blue) tracks high-wavenumber energy faster and more accurately than GenCFD (red).

Wasserstein-1 distance (W_1). On the cloud–shock and shear-layer datasets (CS2D, SL2D) the rectified-flow model (ReFlow) attains the lowest values on all three metrics, indicating that its samples most closely match both the mean behaviour and the higher-order statistics of the ground truth. The hybrid *GenCFD∘FNO* reduces the gap to ReFlow relative to its GenCFD component, but still records noticeably larger errors, whereas the one-shot deterministic baselines (FNO, UViT) show the largest deviations, especially in e_σ and W_1, confirming their difficulty in capturing output variability. RM2D is the only benchmark whose test

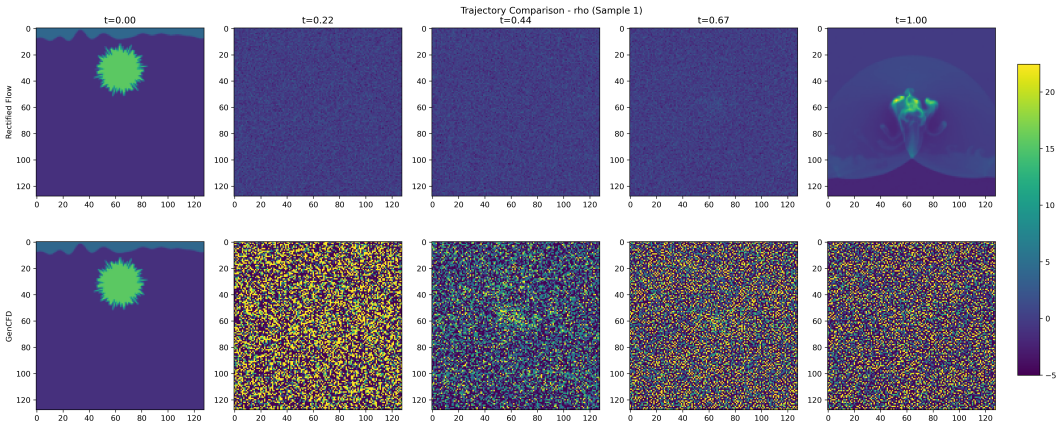


Figure 4: Density trajectories for Sample 1: ReFlow (top) versus GenCFD (bottom).

set remains strictly in-distribution (in the sense defined earlier). Under this easier regime the deterministic baselines come closest to the conditional-diffusion surrogates, but only here. Their drop-off on the out-of-distribution tasks underscores the stronger generalisation of conditional diffusion models. Under identical training budgets, ReFlow achieves lower errors on CS2D and SL2D compared to all baselines, including GenCFD with the same architecture and model size. This highlights ReFlow's superior convergence efficiency despite parity in model capacity and training iterations.

Moreover as visualized in the **SM**, ReFlow generates samples of very high quality, matching the qualitative features of the ground truth and capturing features such as (interacting) vortices as well as propagating shock waves very accurately. Another aspect where ReFlow really shines in being able to generate the correct point pdfs. Although already indicated in the very low 1-point Wasserstein distance errors from Table 1, we present 1-point pdfs in the **SM** to illustrate this observation.

Another key indicator of the quality of approximation of multiscale PDE solutions is the energy spectrum [21]. In Figure 13, we plot the (log) spectra for the energy variable of a randomly chosen Cloud Shock initial datum at four different Diffusion times, $\tau \in \{0.25, 0.50, 0.75, 1.00\}$ to observe that i) the solution is indeed highly multiscale, with spectra decaying as a power law over a range of frequencies ii) ReFloW approximates the spectra accurately within a few (5) steps and iii) on the other hand, for these few steps in solving the underlying reverse SDE, a score-based Diffusion model such as GenCFD, struggles to approximate the relevant small scales. It will require much more sampling steps to do so. These results also showcase how ReFlow is able to infer the correct statistical solutions with significantly less ODE solve steps and is hence, much faster, than a diffusion based model such as GenCFD. This gain in inference speed is also reinforced from Figure 4, where we compare ReFlow (RF) and GenCFD reconstructions of the density field ρ for the Cloud-Shock interaction over five diffusion times $\tau \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$. We see that ReFlow inpaints fine-scale features in far fewer steps, whereas GenCFD remains visibly much more noisy at these small number of steps. See **SM** M.3 for further visualizations of this effect.

7 Additional empirical analysis (integrators, robustness, and diversity)

This section consolidates additional experiments conducted in response to reviewer requests, with the goal of making the computational trade-offs and the role of the curvature-aware controller explicit. The emphasis is on (i) solver choices for the learned rectified-flow ODE, (ii) robustness of the controller to its hyperparameters, and (iii) distributional fidelity / diversity under the macro–micro evaluation protocol.

7.1 Protocol recap: macro–micro ensembles, OOD macros, and metrics

Macro–micro conditional ensembles. To evaluate *statistical fidelity* rather than only pointwise accuracy, we follow a macro–micro protocol. For each dataset, we select N_{macro} macroscopic states ("macros")

Table 2: **Fixed-step ODE integrators** at sampling time (ShearLayer2D, OOD macro–micro; 10 macros × 20 micros). We fix the trained rectified-flow model and vary only the integrator. NFE is the number of velocity evaluations; lower is faster. Cost×Err is defined in (27).

Method	Avg. NFE ↓	Rel. L^2 Error ↓ (mean ± std)	Cost×Err ↓
Model-Baseline (built-in sampler)	32.0	0.2533 ± 0.0382	8.11
`torchdiffeq` Midpoint	32.0	0.2533 ± 0.0382	8.11
Euler (16 steps)	16.0	0.2489 ± 0.0371	3.98
RK2 (16 steps; 32 NFEs)	32.0	0.2533 ± 0.0379	8.11
RK4 (16 steps; 64 NFEs)	64.0	0.2540 ± 0.0381	16.26

and for each macro generate N_{micro} microscopic perturbations ("micros") yielding a reference ensemble of size $N_{\text{macro}} \times N_{\text{micro}}$. This makes diversity concrete: for each fixed macro, the ground-truth solver produces an empirical conditional distribution over micros, and the model produces its own empirical conditional distribution. We compare these distributions directly via Wasserstein distances (below), which is the relevant question for statistical surrogates.

OOD macro selection. Unless otherwise stated, we report results in an out-of-distribution (OOD) regime by choosing $N_{\text{macro}} = 10$ macros from shifted initial-condition families / parameter ranges, and $N_{\text{micro}} = 20$ micros per macro (200 conditional samples per dataset). All models are frozen; only the sampling/integration procedure is varied.

Metrics. We report: (i) *relative L^2 error* against the PDE solution,

$$\text{Rel-}L^2(\widehat{u}, u) := \frac{\|\widehat{u} - u\|_{L^2}}{\|u\|_{L^2}}, \tag{26}$$

(ii) *NFE* (neural function evaluations), i.e. the number of velocity-field evaluations per generated sample, and (iii) a combined cost–accuracy scalar

$$\text{Cost} \times \text{Err} := (\text{Rel-}L^2) \times (\text{NFE}), \tag{27}$$

which is convenient for summarizing the NFE–error plane (lower is better). Distributional fidelity is measured via empirical W_1 between conditional ensembles; we report both aggregated and per-variable W_1 (Section 7.5).

7.2 Integrator study at sampling time: fixed-step baselines vs. curvature-aware control

A recurring concern is whether the gains are merely a consequence of choosing a particular numerical solver. To isolate this, we fix the trained rectified-flow model and vary *only* the integrator used to solve the rectified-flow ODE at sampling time, starting from noise and integrating from rectified time $\tau = 0$ to $\tau = 1$. We include a reference implementation from `torchdiffeq` and standard explicit fixed-step schemes. We report the aggregated results on *ShearLayer2D* in the OOD macro–micro regime; the second benchmark exhibits the same qualitative trends and is reported in the appendix.

Observation: Euler has a tuned "sweet spot" but remains brittle. Among fixed-step schemes, Euler with 16 steps is the strongest baseline in this regime, improving on 32/64-NFE schemes in Cost×Err. However, in multiscale mildly stiff regimes, fixed-step performance is highly sensitive to the chosen step budget: slightly fewer steps can degrade accuracy sharply, while increasing steps can *also* worsen error due to accumulated discretization bias along curved trajectories. Thus, even simple Euler is not a drop-in robust choice: it requires per-dataset/per-regime tuning to hit its efficient operating point.

Table 3: **Hyperparameter robustness** of the curvature-aware adaptive sampler (ShearLayer2D, OOD macro–micro). Across 20 controller configurations, Avg. NFE stays in a narrow band and relative errors vary by $< 1\%$, indicating the controller is not brittle and does not rely on a narrow tuning "sweet spot".

Config	Avg. NFE ↓	Rel. L^2 Error ↓ (mean ± std)	Cost×Err ↓
baseline	11.0	0.2374 ± 0.0389	2.62
ema_0.25	**10.7**	**0.2369 ± 0.0386**	**2.54**
ema_0.45	11.3	0.2361 ± 0.0384	2.67
alpha_0.05	11.0	0.2373 ± 0.0389	2.62
alpha_0.12	11.0	0.2373 ± 0.0389	2.62
alpha_0.20	11.0	0.2374 ± 0.0389	2.62
gamma_1.5	11.0	0.2374 ± 0.0389	2.62
gamma_2.0	11.0	0.2374 ± 0.0389	2.62
gamma_2.5	11.0	0.2374 ± 0.0389	2.62
gate_0.50	11.0	0.2374 ± 0.0389	2.62
gate_0.70	11.0	0.2374 ± 0.0389	2.62
calib_0.70_0.90	11.1	0.2371 ± 0.0387	2.63
calib_0.80_0.98	10.9	0.2372 ± 0.0387	2.60
adapt_1.5_0.75	12.8	0.2374 ± 0.0378	3.04
adapt_2.0_0.85	11.2	0.2360 ± 0.0376	2.63
no_ortho_filter	11.0	0.2374 ± 0.0389	2.62
damp_0.05	11.0	0.2373 ± 0.0389	2.62
damp_0.10	11.0	0.2373 ± 0.0389	2.61
damp_0.20	11.0	0.2370 ± 0.0389	2.60
damp_0.30	11.0	0.2368 ± 0.0389	2.60

7.3 Curvature-aware sampler: hyperparameter robustness and Pareto dominance

We next evaluate the curvature-based adaptive controller under a broad sweep of hyperparameters while holding the model fixed. The controller uses (i) an EMA-based straightness/curvature proxy on the learned velocity field, (ii) a mild velocity regularization/blending step triggered by curvature, and (iii) adaptive step sizing. We vary EMA time scale, thresholds, gating, damping, and calibration parameters across 20 configurations.

Observation: robust controller, automatic step allocation. Across all 20 configurations, Avg. NFE remains in a tight range (10.7–12.8) and the mean relative error stays within $[0.2360, 0.2374]$ (sub-1% relative variation), indicating that the controller does not require fragile tuning. This directly contrasts with fixed-step Euler, where efficiency is achieved only at a tuned step budget and can degrade rapidly outside it.

7.4 Best fixed-step baseline vs. best adaptive configurations

For clarity, Table 4 compares the strongest fixed-step baseline (Euler-16) against the top adaptive configurations on ShearLayer2D OOD.

Interpretation in the NFE–error plane. These comparisons summarize a consistent pattern: the adaptive curvature-aware sampler lies on (or very near) the Pareto frontier in the NFE–error plane, while fixed-step baselines are strictly dominated in the regimes tested. This is the practical meaning of the curvature theory: the controller allocates steps only where the learned transport bends, while retaining large steps when the flow is straight.

Table 4: **Best fixed-step baseline vs. top adaptive configs** (ShearLayer2D, OOD macro–micro). At roughly one third fewer velocity evaluations, the adaptive sampler improves both error and Cost×Err.

Method / Config	Avg. NFE	Rel. L^2 Err	Cost×Err	ΔNFE vs Euler	ΔErr vs Euler	Δ(Cost×Err)
Euler (16 steps)	16.0	0.2489	3.98	0%	0%	0%
ema_0.25	10.7	0.2369	2.54	−33.1%	−4.8%	−36.2%
calib_0.80_0.98	10.9	0.2372	2.60	−31.9%	−4.7%	−34.7%
damp_0.30	11.0	0.2368	2.60	−31.3%	−4.9%	−34.7%

7.5 Distributional fidelity and diversity: ensemble W_1 under macro–micro evaluation

A common concern for deterministic probability-flow ODE sampling is potential under-dispersion or reduced diversity. Our evaluation is designed to test this directly in the conditional setting where diversity is well-defined.

Empirical W_1 between conditional ensembles. For each macro, let $\{\widehat{u}^{(j)}\}_{j=1}^{N_{\text{micro}}}$ be model samples and $\{u^{(j)}\}_{j=1}^{N_{\text{micro}}}$ be reference samples. We compute a one-point (spatially averaged) empirical W_1 between the corresponding micro-ensembles, and then average across macros. We also report per-variable W_1 (density, momenta, energy) and an aggregated score.

Key empirical outcome. Across benchmarks, our rectified-flow model matches or improves upon the diffusion baseline in ensemble W_1 while using substantially fewer function evaluations. In particular, the macro–micro protocol provides a direct quantitative check against mode collapse: matching W_1 at fixed macro conditions indicates that the conditional output distribution is captured rather than collapsed. (Complete per-variable W_1 tables and additional datasets are reported in the appendix.)

7.6 Practical takeaway: speed requires both straight transports and curvature-aware integration

The integrator study separates two effects: (i) the rectified-flow training objective yields a transport that can be traversed with far fewer evaluations than score-based diffusion, and (ii) within this ODE setting, curvature-aware control is a robust way to realize the efficiency in stiff multiscale regimes without per-problem solver tuning. Empirically, this is exactly what Tables 2–4 show: the controller achieves lower error at lower NFE than the strongest tuned fixed-step baseline, and its performance is stable across wide hyperparameter ranges.

Appendix pointers. We provide: (a) the same integrator/robustness tables on the second benchmark, (b) NFE–error scatter plots (Pareto frontiers), and (c) detailed per-variable W_1 breakdowns and additional OOD regimes.

8 Conclusion and Future Work

We have demonstrated that rectified flows provide an efficient surrogate for sampling multi-scale PDE solutions: on a suite of challenging incompressible and compressible problems, our method matches the statistical fidelity of conditional diffusion models while requiring an order of magnitude fewer solver steps—yielding up to 22× faster inference in practice. Across matched and shifted initial-condition regimes, rectified flows consistently reproduce energy spectra, vorticity distributions, and one-point Wasserstein distances within sampling error, confirming their robustness and multi-scale accuracy.

Key Takeaways. On account of its straightening bias, ReFlow requires significantly fewer ODE steps at inference-time (8-10) than its conditional-diffusion-based counterparts (more than 128) to achieve the same level of performance on our statistical metrics – consult **SM** M for more details. Furthermore, ReFlow is much more training efficient, requiring up to 120,000 iterations in reaching the same level of performance, across

all experiments, as our other conditional diffusion baselines (which typically take 500-600,000 iterations), see **SM** M.2. Our results further demonstrate that straightening effects remain evident even when projecting down to a 2D PCA subspace (see **SM** M.3).

References

[1] Benedikt Alkin et al. "Universal Physics Transformers: A Framework For Efficiently Scaling Neural Operators". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=oUXiNX5KRm.

[2] Patrick Esser et al. "Scaling rectified flow transformers for high-resolution image synthesis". In: *Forty-first international conference on machine learning*. 2024.

[3] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 2010. ISBN: 9780821849743. URL: https://books.google.ch/books?id=Xnu0o_EJrCQC.

[4] Ulrik S. Fjordholm, Samuel Lanthaler, and Siddhartha Mishra. "Statistical Solutions of Hyperbolic Conservation Laws: Foundations". In: *Archive for Rational Mechanics and Analysis* 226.2 (Nov. 2017), pp. 809–849. ISSN: 1432-0673. DOI: 10.1007/s00205-017-1145-9. URL: https://doi.org/10.1007/s00205-017-1145-9.

[5] Ulrik S. Fjordholm et al. "Statistical solutions of hyperbolic systems of conservation law: Numerical approximation". In: *Mathematical Models and Methods in Applied Sciences* 30.3 (2020), pp. 539–609. DOI: 10.1142/S0218202520500141.

[6] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. "Generative Learning for Forecasting the Dynamics of Complex Systems". In: *arXiv preprint arXiv:2402.17157* (2024).

[7] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. "Generative Learning of the Solution of Parametric Partial Differential Equations Using Guided Diffusion Models and Virtual Observations". In: *arXiv preprint arXiv:2408.00157* (2024).

[8] Han Gao et al. "Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation". In: *Computer Methods in Applied Mechanics and Engineering* 427 (2024), p. 117023.

[9] Zhongkai Hao et al. "Gnot: A general neural operator transformer for operator learning". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 12556–12569.

[10] Maximilian Herde et al. "Poseidon: Efficient foundation models for pdes". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 72525–72624.

[11] Nikola Kovachki et al. "Neural operator: Learning maps between function spaces with applications to pdes". In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97.

[12] Samuel Lanthaler, Siddhartha Mishra, and Carlos Parés-Pulido. "Statistical solutions of the incompressible Euler equations". In: *Mathematical Models and Methods in Applied Sciences* 31.02 (2021), pp. 223–292.

[13] Samuel Lanthaler, Siddhartha Mishra, and Carlos Parés-Pulido. "Statistical solutions of the incompressible Euler equations". In: *Mathematical Models and Methods in Applied Sciences* 31.02 (2021), pp. 223–292.

[14] Zongyi Li et al. "Fourier neural operator for parametric partial differential equations". In: *arXiv preprint arXiv:2010.08895* (2020).

[15] Zongyi Li et al. "Geometry-Informed Neural Operator for Large-Scale 3D PDEs". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=86dXbqT5Ua.

[16] Zongyi Li et al. "Neural operator: Graph kernel network for partial differential equations". In: *arXiv preprint arXiv:2003.03485* (2020).

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. "Flow straight and fast: Learning to generate and transfer data with rectified flow". In: *arXiv preprint arXiv:2209.03003* (2022).

[18] Lu Lu et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229.

[19] Kjetil O. Lye. "Computation of statistical solutions of hyperbolic systems of conservation laws". PhD thesis. 2020.

[20] Siddhartha Mishra and Alex (Eds.) Townsend. *Numerical Analysis meets Machine Learning*. Handbook of Numerical Analysis. Springer, 2024.

[21] Roberto Molinaro et al. "Generative AI for fast and accurate Statistical Computation of Fluids". In: *arXiv preprint arXiv:2409.18359* (2024).

[22] V Oommen et al. *Integrating neural operators with diffusion models improves spectral representation in turbulence modeling. arxiv*. 2024.

[23] A. Quarteroni and A. Valli. *Numerical approximation of Partial differential equations*. Vol. 23. Springer, 1994.

[24] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*. Vol. 92. Springer, 2015.

[25] Tobias Rohner and Siddhartha Mishra. "Efficient Computation of Large-Scale Statistical Solutions to Incompressible Fluid Flows". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '24. ACM, June 2024. DOI: 10.1145/3659914.3659922. URL: http://dx.doi.org/10.1145/3659914.3659922.

[26] Chitwan Saharia et al. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in neural information processing systems* 35 (2022), pp. 36479–36494.

[27] Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* (2020).

[28] Eitan Tadmor. "Convergence of Spectral Methods for Nonlinear Conservation Laws". In: *SIAM Journal on Numerical Analysis* 26.1 (1989), pp. 30–44. DOI: 10.1137/0726003. URL: https://doi.org/10.1137/0726003.

[29] Haixu Wu et al. "Transolver: A Fast Transformer Solver for PDEs on General Geometries". In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=Ywl6pODXjB.

A Theory appendix: law-level rectified flows for multiscale CFD

Purpose and scope. This appendix provides the mathematical backbone for the claims made in the main text: (i) rectified flows can match the same conditional solution laws as diffusion-class models, (ii) their *sampling step count* is governed by a *curvature/straightness* quantity rather than by the raw dimensionality of the state, and (iii) at fixed spatial resolution, there is an *irreducible* law-level error dictated by the small-scale tail of the true PDE law, which is naturally quantified through structure functions. We keep the presentation self-contained, and we explicitly connect each theoretical ingredient to the empirical evaluations in Table 1 and to the spectral diagnostics in Figure 13 (and supplementary figures).

How to read this appendix. Section B fixes the measure-theoretic setting and links it to statistical solutions in the sense of [13] (also used in the Introduction). Section C shows that the *barycentric rectified velocity* exactly transports μ_0 to μ_1 (correctness at the level of laws). Section D gives a *terminal error decomposition* into a fit term and a straightness term and explains how the training objective controls these terms. Section E derives why curvature controls the required number of ODE steps and motivates the EMA-based curvature proxy and the blending correction used at inference. Section F links the *energy spectrum / structure functions* to an unavoidable *coverage* term at fixed cutoff K_c. Finally, Section H aggregates the pieces into a master law-level bound that matches the narrative in the abstract and in the theory paragraphs of the main paper.

B Setup: laws of PDE solutions and what the experiments measure

State space. We work on the periodic domain $\mathcal{D} = \mathbb{T}^d$ and consider (for notational simplicity) scalar or vector fields $u : \mathcal{D} \to \mathbb{R}^m$ represented on a finite grid. The natural infinite-dimensional space for the fluid benchmarks is

$$\mathcal{H} := L^2_\sigma(\mathcal{D}; \mathbb{R}^d) \quad \text{(divergence-free in the incompressible case)},$$

or $\mathcal{H} := L^2(\mathcal{D}; \mathbb{R}^m)$ for compressible state vectors (e.g. (ρ, m_x, m_y, E)). All constructions below are rigorous on the finite-dimensional Galerkin/grid surrogate $\mathbb{R}^{d_N} \subset \mathcal{H}$ used in training and evaluation. We state results in $\mathcal{H}$ language to highlight the measure-theoretic meaning.

Statistical solutions and correlation viewpoint. In strongly nonlinear regimes, it is natural to treat the PDE solution as a *random field*. Given a law $\mu_0 \in \mathcal{P}(\mathcal{H})$ for the initial condition, the law at time t is

$$\mu_t = (\mathcal{S}^t)_\# \mu_0,$$

where $\mathcal{S}^t$ is the PDE solution operator (possibly multivalued in the inviscid setting; see [13] for the correlation hierarchy framework). This is precisely the "statistical solution" language used in the Introduction.

What we evaluate empirically. Our quantitative evaluation (Section 6.2) uses: (i) errors of the mean and standard deviation fields, and (ii) a pointwise Wasserstein-1 discrepancy averaged over space. These are *marginal* discrepancies: they test whether low-order statistics and one-point laws match between the model and the PDE ensemble.

From a law-level perspective, a natural global metric is W_2 on $\mathcal{P}(\mathcal{H})$. Although our reported metric is W_1 on one-point marginals (for computational reasons), the theory below uses W_2 because (a) it interacts cleanly with L^2-based structure functions and Fourier tails, and (b) it yields transparent decompositions. We emphasize that the core mechanisms (coverage/fit/straightness/discretization) are *metric-agnostic*: they explain why a method that reduces curvature reaches good statistics with fewer steps, which is exactly what is observed in Table 1 and Figure 13.

C Rectified-flow correctness: barycentric velocity transports laws

C.1 Couplings and linear interpolation in rectification time

Let $\mu_0, \mu_1 \in \mathcal{P}(\mathcal{H})$ be two laws (in our application: conditional laws of PDE outputs given inputs; see Section 4.1). Fix a coupling $\gamma \in \mathcal{P}(\mathcal{H} \times \mathcal{H})$ of (μ_0, μ_1), i.e. $(\pi_0)_\#\gamma = \mu_0$ and $(\pi_1)_\#\gamma = \mu_1$.

For $\tau \in [0, 1]$ define the interpolation map

$$T_\tau(u_0, u_1) := (1 - \tau)u_0 + \tau u_1, \qquad \rho_\tau := (T_\tau)_\#\gamma.$$

Thus ρ_τ is the law of the random variable $U_\tau = (1 - \tau)U_0 + \tau U_1$ when $(U_0, U_1) \sim \gamma$.

C.2 Barycentric rectified velocity and the continuity equation

Definition C.1 (Barycentric rectified velocity)**.** For each $\tau \in [0, 1]$ and ρ_τ-a.e. u, define

$$v_\star(u, \tau) := \mathbb{E}\big[\, U_1 - U_0 \,\big|\, U_\tau = u \,\big].$$

Equivalently, $v_\star$ is obtained by disintegrating γ along the fibers $T_\tau^{-1}(u)$, as in equation 2 in the main text.

Why this is the "right" velocity. If we think of ρ_τ as mass moving from $\rho_0 = \mu_0$ to $\rho_1 = \mu_1$ in rectification time, then the pair $(\rho_\tau, v_\star)$ solves the continuity equation

$$\partial_\tau \rho_\tau + \nabla \cdot (\rho_\tau v_\star(\cdot, \tau)) = 0 \quad \text{(on the finite-dimensional surrogate).}$$

This is the deterministic analogue of the probability-flow (ODE) viewpoint of diffusion models: the flow map of $v_\star$ pushes ρ_0 to ρ_1.

Proposition C.2 (Exact transport under the barycentric velocity)**.** *Assume we work on a finite-dimensional surrogate of $\mathcal{H}$ so that $u \in \mathbb{R}^{d_N}$ and ρ_τ admits a density (or, more generally, is a narrowly continuous curve of measures). Let X_τ be the (classical) flow map solving*

$$\frac{d}{d\tau} X_\tau(u) = v_\star(X_\tau(u), \tau), \qquad X_0(u) = u.$$

Then $(X_\tau)_\#\mu_0 = \rho_\tau$ for all $\tau \in [0, 1]$, and in particular $(X_1)_\#\mu_0 = \mu_1$.

Proof. Let $(U_0, U_1) \sim \gamma$ and define $U_\tau = (1 - \tau)U_0 + \tau U_1$ so that $\mathrm{Law}(U_\tau) = \rho_\tau$. Fix $\varphi \in C_c^\infty(\mathbb{R}^{d_N})$. By differentiating under the expectation,

$$\frac{d}{d\tau}\mathbb{E}[\varphi(U_\tau)] = \mathbb{E}\big[\nabla\varphi(U_\tau) \cdot (U_1 - U_0)\big].$$

Condition on U_τ and use Definition C.1:

$$\mathbb{E}\big[\nabla\varphi(U_\tau) \cdot (U_1 - U_0)\big] = \mathbb{E}\big[\nabla\varphi(U_\tau) \cdot \mathbb{E}[U_1 - U_0 \mid U_\tau]\big] = \mathbb{E}\big[\nabla\varphi(U_\tau) \cdot v_\star(U_\tau, \tau)\big].$$

Thus ρ_τ satisfies the weak continuity equation

$$\frac{d}{d\tau} \int \varphi(u)\, \rho_\tau(du) = \int \nabla\varphi(u) \cdot v_\star(u, \tau)\, \rho_\tau(du).$$

On the other hand, the pushforward curve $\tilde{\rho}_\tau := (X_\tau)_\#\rho_0$ satisfies the same weak equation by standard characteristics. Uniqueness of solutions to the continuity equation on $\mathbb{R}^{d_N}$ (under the regularity ensuring existence of X_τ) yields $\rho_\tau = \tilde{\rho}_\tau$, hence $(X_1)_\#\mu_0 = \mu_1$. □

Empirical tie. Correctness in Proposition C.2 is a law-level statement: it says that if a learned velocity v_θ matches $v_\star$, then integrating the ODE recovers the target law. This is what is tested empirically through mean/spread/Wasserstein statistics (Table 1) and through the evolution of multiscale spectra along τ (Figure 13).

D Continuous-time error decomposition: fit vs. straightness

D.1 The training objective targets the barycentric velocity

In the (unconditional) rectified-flow construction, minimizing

$$\int_0^1 \mathbb{E}\big[\|(U_1 - U_0) - v(U_\tau, \tau)\|^2\big]\, d\tau$$

over measurable v yields the conditional expectation $v_\star(u, \tau) = \mathbb{E}[U_1 - U_0 \mid U_\tau = u]$. In the conditional PDE setting we use the same principle but with (U_0, U_1) sampled from the conditional pairing induced by the dataset and conditioning variables (initial/boundary data). This explains why the regression loss in Algorithm 1 is the natural analogue of score matching: it is an L^2 projection onto the optimal transport direction at each τ.

D.2 Terminal error bound with explicit decomposition

Let u_τ solve the ideal rectified ODE $\dot{u}_\tau = v_\star(u_\tau, \tau)$ and let $\hat{u}_\tau$ solve the learned ODE $\dot{\hat{u}}_\tau = v_\theta(\hat{u}_\tau, \tau)$ with the same initial data $u_0 = \hat{u}_0$.

Define the time-average velocity

$$\bar{v}_\theta(u) := \int_0^1 v_\theta(u, \tau)\, d\tau.$$

The average isolates the "straight-line" component of the learned field, while deviations from it quantify curvature in rectification time.

Theorem D.1 (Terminal error decomposition (fit + straightness))**.** *Assume $v_\theta(\cdot, \tau)$ is L-Lipschitz in u uniformly in τ and that u_τ remains in a region where the quantities below are finite. Define*

$$\varepsilon_{\text{fit}}^2 := \int_0^1 \mathbb{E}\big\|\bar{v}_\theta(u_\tau) - v_\star(u_\tau, \tau)\big\|^2\, d\tau, \qquad \varepsilon_{\text{curv}}^2 := \int_0^1 \mathbb{E}\big\|v_\theta(u_\tau, \tau) - \bar{v}_\theta(u_\tau)\big\|^2\, d\tau.$$

Then

$$\mathbb{E}\|\hat{u}_1 - u_1\| \;\le\; e^L \left(\varepsilon_{\text{fit}} + \varepsilon_{\text{curv}}\right). \tag{28}$$

Moreover, the same argument in L^2 yields

$$\mathbb{E}\|\hat{u}_1 - u_1\|^2 \;\le\; C(L)\left(\varepsilon_{\text{fit}}^2 + \varepsilon_{\text{curv}}^2\right), \qquad C(L) \sim e^{2L}. \tag{29}$$

Proof. Set $e_\tau := \hat{u}_\tau - u_\tau$. Then

$$\dot{e}_\tau = v_\theta(\hat{u}_\tau, \tau) - v_\star(u_\tau, \tau) = \big[v_\theta(\hat{u}_\tau, \tau) - v_\theta(u_\tau, \tau)\big] + \big[v_\theta(u_\tau, \tau) - v_\star(u_\tau, \tau)\big].$$

Take norms and use Lipschitz continuity:

$$\frac{d}{d\tau}\|e_\tau\| \le L\|e_\tau\| + \|v_\theta(u_\tau, \tau) - v_\star(u_\tau, \tau)\|.$$

Integrate from 0 to 1 and apply Grönwall:

$$\|e_1\| \le e^L \int_0^1 \|v_\theta(u_\tau, \tau) - v_\star(u_\tau, \tau)\|\, d\tau.$$

Now add and subtract $\bar{v}_\theta(u_\tau)$ and use triangle inequality:

$$\|v_\theta(u_\tau, \tau) - v_\star(u_\tau, \tau)\| \le \|v_\theta(u_\tau, \tau) - \bar{v}_\theta(u_\tau)\| + \|\bar{v}_\theta(u_\tau) - v_\star(u_\tau, \tau)\|.$$

Take expectations and apply Cauchy–Schwarz in τ to obtain (28). For (29), repeat with $\|e_\tau\|^2$:

$$\frac{d}{d\tau}\|e_\tau\|^2 = 2e_\tau \cdot \dot{e}_\tau \le 2L\|e_\tau\|^2 + 2\|e_\tau\|\,\|v_\theta(u_\tau, \tau) - v_\star(u_\tau, \tau)\|,$$

and apply Young's inequality plus Grönwall. The resulting constant is $C(L) \sim e^{2L}$. □

Empirical tie. Theorem D.1 explains the two central practical observations: (i) if the learned velocity is nearly *constant in* τ (small straightness error), then the continuous-time mismatch is small, and (ii) straighter trajectories are easier to integrate, which translates into fewer solver steps at inference (Section E). This is exactly the mechanism behind "8 steps vs. 128 steps" in the main text and the sharp spectral tracking in Figure 13: early steps already reconstruct high wavenumbers because the trajectory is not spending many steps "turning" in rectification time.

E Discretization: why curvature controls step count, and why EMA helps

E.1 Local truncation error and the curvature driver

We start from a clean Taylor expansion for the ODE $\dot{u} = v(u, \tau)$.

Lemma E.1 (Local truncation error for explicit Euler)**.** *Assume $v \in C^1$ in (u, τ) on the region traversed and that $\|\partial_\tau v(u, \tau)\| \le M_\tau$, $\|J_u v(u, \tau)\| \le L$, and $\|v(u, \tau)\| \le M$. Then one explicit-Euler step from (u, τ) with step size $\Delta\tau$ has local truncation error*

$$\begin{aligned} \mathrm{LTE}(\tau; \Delta\tau) &:= \big\| u(\tau + \Delta\tau) - \big(u(\tau) + \Delta\tau\, v(u(\tau), \tau)\big) \big\| \\ &= \frac{(\Delta\tau)^2}{2} \big\| \partial_\tau v(u, \tau) + J_u v(u, \tau)\, v(u, \tau) \big\| \;+\; \mathcal{O}\big((\Delta\tau)^3\big) \\ &\le \frac{(\Delta\tau)^2}{2} \,(M_\tau + LM) \;+\; \mathcal{O}\big((\Delta\tau)^3\big). \end{aligned} \tag{30}$$

Proof. Differentiate $\dot{u} = v(u, \tau)$ along the trajectory:

$$\ddot{u}(\tau) = \partial_\tau v(u(\tau), \tau) + J_u v(u(\tau), \tau)\, \dot{u}(\tau) = \partial_\tau v(u(\tau), \tau) + J_u v(u(\tau), \tau)\, v(u(\tau), \tau).$$

Taylor expand:

$$u(\tau + \Delta\tau) = u(\tau) + \Delta\tau\, v(u(\tau), \tau) + \frac{(\Delta\tau)^2}{2} \ddot{u}(\tau) + \mathcal{O}((\Delta\tau)^3).$$

Subtract the Euler update and take norms to obtain (30). The bound follows by triangle inequality and the assumed sup bounds. □

Interpretation. Lemma E.1 identifies two drivers of discretization error:

$$\|\partial_\tau v\| \quad \text{(time curvature)} \qquad \text{and} \qquad \|J_u v\, v\| \quad \text{(spatial nonlinearity)}.$$

Rectified flows are designed so that trajectories are *nearly straight in* τ, i.e. $\partial_\tau v$ is small along typical paths. This is the fundamental reason they can use large steps without losing fidelity.

E.2 Global error: straightness reduces the required number of steps

Corollary E.2 (Global Euler error controlled by time-curvature)**.** *Assume the hypotheses of Lemma E.1 and that $v(\cdot, \tau)$ is L-Lipschitz in u uniformly in τ on the region traversed. Let u^Δ be explicit Euler with uniform step $\Delta\tau = 1/N$. Then there exists $C = C(L)$ such that*

$$\sup_{\tau \in [0,1]} \|u_\tau^\Delta - u_\tau\| \;\le\; C\, \Delta\tau \int_0^1 \Big(\|\partial_\tau v(u_\tau, \tau)\| + \|J_u v(u_\tau, \tau)\, v(u_\tau, \tau)\| \Big)\, d\tau, \tag{31}$$

up to higher-order terms.

Proof. Write the one-step error recursion $e_{n+1} \le (1 + L\Delta\tau) e_n + \mathrm{LTE}_n$ and iterate it over $n = 0, \ldots, N-1$. Use $\prod_{k=0}^{N-1} (1 + L\Delta\tau) \le e^L$ and sum the LTE bounds from Lemma E.1. The integral form (31) is obtained by viewing the sum as a Riemann approximation of $\int_0^1 (\cdots) d\tau$. □

Empirical tie. Equation (31) explains the step-count gap we observe: if $\int_0^1 \|\partial_\tau v\|$ is small (straight trajectories), then the N required for a fixed tolerance is small. Diffusion-class samplers typically integrate a much more curved reverse-time dynamics (in addition to stochasticity), which translates into much larger effective curvature and hence many more steps. This is also consistent with the qualitative "trajectory inpainting" comparison in Figure 4.

E.3 EMA curvature proxy and blending as a Tikhonov correction

Discrete-time EMA as a low-pass filter. During sampling we evaluate $v_t := v_\theta(u_t, \tau_t)$ at a discrete set of times τ_t and maintain an exponential moving average (EMA)

$$v_t^{\text{ema}} = \lambda\, v_{t-1}^{\text{ema}} + (1-\lambda)\, v_t, \qquad \lambda \in (0,1).$$

If v_t varies slowly with τ_t, then v_t^{ema} tracks it closely; if v_t changes quickly, the EMA lags and the deviation

$$s_t := \|v_t - v_t^{\text{ema}}\|$$

spikes. This motivates s_t as a computable curvature/straightness proxy.

A simple bound shows s_t controls a discrete time-variation scale.

Lemma E.3 (EMA deviation controls incremental variation)**.** *Let $\{v_t\}_{t\geq 0}$ be any sequence in a normed space and define $v_t^{\text{ema}} = \lambda v_{t-1}^{\text{ema}} + (1-\lambda) v_t$ with $v_0^{\text{ema}} = v_0$. Then*

$$\|v_t - v_t^{\text{ema}}\| \leq (1-\lambda) \sum_{j=0}^{t-1} \lambda^j \, \|v_{t-j} - v_{t-j-1}\|.$$

In particular, if $\|v_k - v_{k-1}\| \leq \delta$ for all k, then $\|v_t - v_t^{\text{ema}}\| \leq \delta$ for all t.

Proof. Unroll the recursion: $v_t^{\text{ema}} = (1-\lambda)\sum_{j=0}^{t} \lambda^j v_{t-j}$. Subtract from v_t and telescope:

$$v_t - v_t^{\text{ema}} = (1-\lambda) \sum_{j=0}^{t-1} \lambda^j \, (v_t - v_{t-j-1}),$$

then use triangle inequality and rewrite $v_t - v_{t-j-1}$ as a sum of increments. □

Blending as a quadratic minimizer. We use the blended update

$$\tilde{v}_t = (1-\alpha_t) v_t + \alpha_t v_t^{\text{ema}}, \qquad \alpha_t \in [0,1].$$

This is the unique minimizer of a Tikhonov-regularized quadratic:

$$\tilde{v}_t = \arg\min_w \Big\{ \|w - v_t\|^2 + \beta_t \|w - v_t^{\text{ema}}\|^2 \Big\}, \qquad \alpha_t = \frac{\beta_t}{1+\beta_t}.$$

Thus blending is not an ad-hoc smoothing step: it is the closed-form solution of a principled stabilization objective trading fidelity to the instantaneous network prediction against adherence to a locally averaged trend.

Step-size rule from LTE control. Lemma E.1 suggests choosing $\Delta\tau_t$ so that

$$(\Delta\tau_t)^2 \Big(\|\partial_\tau v\| + \|J_u v\, v\| \Big) \approx \text{const}.$$

Since $\|\partial_\tau v\|$ is not directly available, we replace it by a multiple of the proxy s_t (Lemma E.3), and we upper-bound $\|J_u v\, v\| \lesssim L\|v_t\|$ using Lipschitz control. This yields the square-root-type rule in the main text:

$$\Delta\tau_t \propto \frac{1}{\sqrt{\kappa_1 s_t + \kappa_2}}, \qquad \kappa_2 \approx L\|v_t\|.$$

This is exactly the mechanism behind the "curvature-aware integration" contribution: when the trajectory bends (large s_t), we both (i) regularize the velocity via blending and (ii) reduce the step size to control the discretization term in (31).

F Multiscale coverage: structure functions control unavoidable law tails

F.1 Why spectra/structure functions appear in a law-level bound

Finite resolution is a hard constraint. All models operate on a fixed grid (e.g. 128^2 for training/evaluation). This enforces an effective Fourier cutoff $K_{\max} \simeq N$ (Nyquist), and architectural choices may impose an even lower effective cutoff K_c. Therefore, even a perfect generative model cannot represent energy beyond that cutoff. The best it can do is match the *projected* law $(P_{\leq K_c})_{\#}\mu$.

Structure functions quantify tail energy. In turbulence, small-scale content is captured by structure functions and energy spectra. This is exactly why our empirical analysis emphasizes spectral evolution along τ (Figure 13). The theory below formalizes the link: a structure-function modulus bound implies a Fourier-tail bound, which becomes a coverage term in W_2.

F.2 Bandlimited fields: capacity bounds and why missing an annulus matters

We record two standard Fourier facts to make the "capacity" language explicit.

Lemma F.1 (Bernstein upper bound for bandlimited fields)**.** *Let $u : \mathbb{T}^d \to \mathbb{R}^m$ satisfy $u = P_{\leq K}u$ for some integer $K \geq 1$. Then there exists $C_d > 0$ such that*

$$\|\nabla_x u\|_{L^\infty(\mathbb{T}^d)} \leq C_d\, K^{1+\frac{d}{2}}\, \|u\|_{L^2(\mathbb{T}^d)}.$$

Proof. Expand $u(x) = \sum_{|k|\leq K} \hat{u}(k)e^{ik\cdot x}$, write ∇u as a Fourier series, and apply Cauchy–Schwarz using that $\#\{|k| \leq K\} \lesssim K^d$ and Parseval. □

Lemma F.2 (Annulus lower bound)**.** *Let $f = P_{[K,2K]}u$ and $K \geq 1$. Then there exists $c_d > 0$ such that*

$$\|\nabla_x f\|_{L^\infty} \geq c_d\, K^{1-\frac{d}{2}}\, \|f\|_{L^2}.$$

Proof. On the annulus $K \leq |k| \leq 2K$, the Fourier symbol of ∇ has magnitude $\simeq K$. A reverse Bernstein estimate on an annulus yields $\|f\|_{L^\infty} \gtrsim K^{-d/2}\|f\|_{L^2}$, and $\|\nabla f\|_{L^\infty} \gtrsim K\|f\|_{L^\infty}$, giving the claim. □

Empirical tie. Lemma F.2 formalizes why "spectral collapse" is fatal for turbulent statistics: if a surrogate misses energy in a whole annulus, it underestimates strain and small-scale variability. This is exactly what appears visually when deterministic baselines blur fine scales and also what is quantified by the spread errors in Table 1.

F.3 Structure functions imply Fourier-tail control (law-level)

For $u : \mathbb{T}^d \to \mathbb{R}^m$ define the second-order structure function

$$S_r^2(u) := \frac{1}{|B_r|}\int_{|h|\leq r} \|u(\cdot + h) - u(\cdot)\|^2_{L^2(\mathbb{T}^d)}\, dh, \qquad S_r^2(\mu) := \int S_r^2(u)\, d\mu(u).$$

Following the statistical-solution framework of [13], assume there exists a modulus $\omega(r) \downarrow 0$ as $r \downarrow 0$ such that

$$S_r^2(\mu_t) \leq \omega(r) \qquad \text{for all } r \in (0, 1],\ t \in [0, T]. \tag{32}$$

In practice, ω can be calibrated from empirical spectra (Figure 13 and supplementary plots).

Lemma F.3 (Structure function controls Fourier tail)**.** *Let μ be a law on $L^2(\mathbb{T}^d)$ satisfying $S_r^2(\mu) \leq \omega(r)$ for all $r \in (0, 1]$. Then there exist $c_d, C_d > 0$ such that for all integers $K \geq 1$,*

$$\int \|P_{>K}u\|^2_{L^2}\, d\mu(u) \leq C_d\, \omega\big(c_d/K\big).$$

Proof. For fixed u, expand $\|u(\cdot+h)-u(\cdot)\|_{L^2}^2 = \sum_k |\hat{u}(k)|^2 |e^{ik\cdot h}-1|^2$. Average over $|h| \le r$ to obtain $S_r^2(u) = \sum_k |\hat{u}(k)|^2 \Psi_r(k)$ where $\Psi_r(k) = \frac{1}{|B_r|}\int_{|h|\le r} |e^{ik\cdot h}-1|^2\, dh$. A standard oscillation estimate yields $\Psi_r(k) \ge c\min\{(|k|r)^2, 1\}$. Hence $\Psi_r(k) \ge c_0$ for all $|k| \ge c'/r$, and therefore $S_r^2(u) \ge c_0 \sum_{|k|\ge c'/r} |\hat{u}(k)|^2 = c_0 \|P_{>c'/r}u\|_{L^2}^2$. Integrate in u and set $r = c_d/K$. □

Lemma F.4 (Bandlimited projection in W_2)**.** *For any law μ on $L^2(\mathbb{T}^d)$ and cutoff $K_c \ge 1$,*

$$W_2\big(\mu, (P_{\le K_c})_\#\mu\big) \le \Big(\int \|P_{>K_c}u\|_{L^2}^2\, d\mu(u)\Big)^{1/2}.$$

Proof. Couple $u \sim \mu$ with $\tilde{u} = P_{\le K_c}u$. Then $\tilde{u} \sim (P_{\le K_c})_\#\mu$ and $W_2^2 \le \mathbb{E}\|u-\tilde{u}\|_{L^2}^2 = \mathbb{E}\|P_{>K_c}u\|_{L^2}^2$. □

Corollary F.5 (Coverage term from structure-function modulus)**.** *Under (32), for any cutoff $K_c \ge 1$,*

$$W_2\big(\mu_t, (P_{\le K_c})_\#\mu_t\big) \lesssim \omega(c_d/K_c)^{1/2}.$$

Empirical tie. Corollary F.5 is the clean mathematical version of what the spectra show: at fixed resolution, there is an unavoidable tail beyond the cutoff. Both ReFlow and GenCFD operate at the same grid resolution (hence comparable K_c), so neither can beat the coverage floor. The difference is that ReFlow reaches the best attainable projected law with far fewer steps (Figure 13, Figure 4).

G One-step law error for PDE pushforwards and training fit

Let $S_{\Delta t} : \mathcal{H} \to \mathcal{H}$ denote the PDE solution operator over a physical step Δt and $\mu_{t+\Delta t} = (S_{\Delta t})_\#\mu_t$. Let $\mathcal{T}_{\Delta t}^\theta$ be the learned one-step surrogate and $\hat{\mu}_{t+\Delta t} = (\mathcal{T}_{\Delta t}^\theta)_\#\mu_t$.

Why we insert a bandlimit. Since the model outputs live at finite resolution, it is natural to compare to the *projected* PDE output $P_{\le K_c}S_{\Delta t}u$. This isolates (i) an irreducible coverage term and (ii) a trainable fit term.

Define the mean-square training error at time t:

$$\varepsilon_{\text{train}}^2(t) := \mathbb{E}_{u\sim\mu_t}\Big[\, \big\|P_{\le K_c}(S_{\Delta t}u) - \widehat{u}\big\|_{L^2}^2 \Big], \tag{33}$$

where $\widehat{u}$ is the model output given input u (i.e. after integrating the learned RF ODE).

Proposition G.1 (One-step law error = coverage + fit)**.** *Assume $\mu_{t+\Delta t}$ satisfies the structure-function modulus (32) (hence Corollary F.5). Then*

$$W_2\big(\mu_{t+\Delta t}, \hat{\mu}_{t+\Delta t}\big) \;\le\; \underbrace{W_2\big(\mu_{t+\Delta t}, (P_{\le K_c})_\#\mu_{t+\Delta t}\big)}_{\textit{coverage}} \;+\; \underbrace{\big(\varepsilon_{\text{train}}^2(t)\big)^{1/2}}_{\textit{fit}},$$

and therefore

$$W_2\big(\mu_{t+\Delta t}, \hat{\mu}_{t+\Delta t}\big) \;\lesssim\; \omega(c_d/K_c)^{1/2} + \varepsilon_{\text{train}}(t).$$

Proof. Use the triangle inequality in W_2:

$$W_2(\mu_{t+\Delta t}, \hat{\mu}_{t+\Delta t}) \le W_2\big(\mu_{t+\Delta t}, (P_{\le K_c})_\#\mu_{t+\Delta t}\big) + W_2\big((P_{\le K_c})_\#\mu_{t+\Delta t}, \hat{\mu}_{t+\Delta t}\big).$$

For the second term, couple by sampling $u \sim \mu_t$ and using the same input to produce $P_{\le K_c}S_{\Delta t}u$ and $\widehat{u}$. Then

$$W_2^2\big((P_{\le K_c})_\#\mu_{t+\Delta t}, \hat{\mu}_{t+\Delta t}\big) \le \mathbb{E}\|P_{\le K_c}S_{\Delta t}u - \widehat{u}\|_{L^2}^2 = \varepsilon_{\text{train}}^2(t).$$

The second display follows from Corollary F.5. □

Empirical tie. Proposition G.1 explains why training curves saturate at a resolution-dependent floor and why spectra are the correct diagnostic for that floor. It also makes explicit that "better training loss" means smaller law error *only up to* the coverage limit. This aligns with the observation that increasing diffusion steps improves spectral reconstruction but cannot beat finite-resolution constraints, while ReFlow reaches that regime with far fewer steps.

H Master inequality: coverage + fit + straightness + discretization

We now combine the law-level coverage/fit decomposition with the continuous-time RF mismatch and the discretization control.

Theorem H.1 (Master law-level bound for learned rectified flows)**.** *Fix an effective cutoff K_c and assume the structure-function modulus (32) for the target PDE laws. Let $\hat{\mu}_1^{(N)}$ be the law obtained by integrating the learned rectified ODE $\dot{u}_\tau = v_\theta(u_\tau, \tau)$ from $\tau = 0$ to $\tau = 1$ using N explicit-Euler steps (uniform step $1/N$). Assume $v_\theta(\cdot, \tau)$ is L-Lipschitz in u uniformly in τ.*

Then, for constants depending only on L,

$$W_2(\mu_1, \hat{\mu}_1^{(N)}) \lesssim \underbrace{\omega(c_d/K_c)^{1/2}}_{\textit{coverage}} + \underbrace{\varepsilon_{\text{fit}} + \varepsilon_{\text{curv}}}_{\textit{fit + straightness}} + \underbrace{\frac{1}{N}\int_0^1 \|\partial_\tau v_\theta(u_\tau, \tau)\|\, d\tau}_{\textit{discretization}}, \tag{34}$$

where $\varepsilon_{\text{fit}}, \varepsilon_{\text{curv}}$ are the continuous-time terms from Theorem D.1 evaluated along the ideal trajectory.

Proof. (1) *Coverage.* By Corollary F.5, $W_2(\mu_1, (P_{\le K_c})_\# \mu_1) \lesssim \omega(c_d/K_c)^{1/2}$.

(2) *Continuous-time learned flow error.* Let $\hat{\mu}_1$ denote the law produced by the *exact* (continuous-time) flow of v_θ. Then Theorem D.1 implies a bound of the form $W_2((P_{\le K_c})_\# \mu_1, \hat{\mu}_1) \lesssim \varepsilon_{\text{fit}} + \varepsilon_{\text{curv}}$ (up to the standard coupling-by-same-input argument on the finite-dimensional surrogate).

(3) *Discretization.* Let $\hat{\mu}_1^{(N)}$ be the Euler-discretized law. Corollary E.2 bounds the trajectory error by a term proportional to $\frac{1}{N}\int_0^1 \|\partial_\tau v_\theta\|\, d\tau$ (plus the spatial term, absorbed in constants under the Lipschitz assumption and bounded $\|v_\theta\|$ on the region traversed).

(4) Combine with triangle inequalities:

$$W_2(\mu_1, \hat{\mu}_1^{(N)}) \le W_2(\mu_1, (P_{\le K_c})_\# \mu_1) + W_2((P_{\le K_c})_\# \mu_1, \hat{\mu}_1) + W_2(\hat{\mu}_1, \hat{\mu}_1^{(N)}),$$

and insert the three bounds. □

What Theorem H.1 explains in the paper.

- **Why rectification accelerates sampling.** The discretization term depends on the integral of time-curvature. Straightness reduces that integral, so fewer steps suffice (8–10 in our experiments).
- **Why curvature-aware integration helps under shift.** Under distribution shift, the learned velocity may bend more in τ, increasing the last term. The EMA proxy and blending/step control reduce the effective curvature and hence stabilize sampling, which is exactly how curvature is used in our method (control, not OOD detection).
- **Why spectra matter.** The coverage term is dictated by small-scale physics (structure functions / spectrum). This is why Figure 13 is the right diagnostic: it visualizes how quickly the method reconstructs high-frequency energy up to the grid cutoff, and it clarifies what errors are unavoidable at fixed resolution.
- **Why deterministic baselines struggle.** Deterministic predictors trained by pointwise losses effectively target conditional means; in multimodal chaotic regimes this collapses variability (as emphasized by [21]). In the master inequality language, they do not produce a good fit to the law itself, which is seen in their larger spread and Wasserstein errors in Table 1.

I From law-level bounds to reported pointwise metrics

From W_2 to marginal discrepancies. While the paper reports one-point W_1 (averaged over space) for computational tractability, the mechanisms above are consistent with those marginal tests: if two laws on fields are close in W_2 and have bounded second moments, then many bounded-Lipschitz observables of the fields (including smoothed one-point marginals) are close as well. This is the conceptual bridge between the law-level theory and the reported statistics.

Why we emphasize spectra during sampling. The spectral evolution plots (Figure 13 and supplementary variants) act as a *scale-resolved* diagnostic: they visualize the progressive recovery of high frequencies along τ. This connects directly to the structure-function/coverage viewpoint (Section F) and empirically demonstrates that ReFlow reaches the best attainable high-frequency reconstruction regime in far fewer steps than diffusion-based baselines.

J Architecture of the ReFlow Model

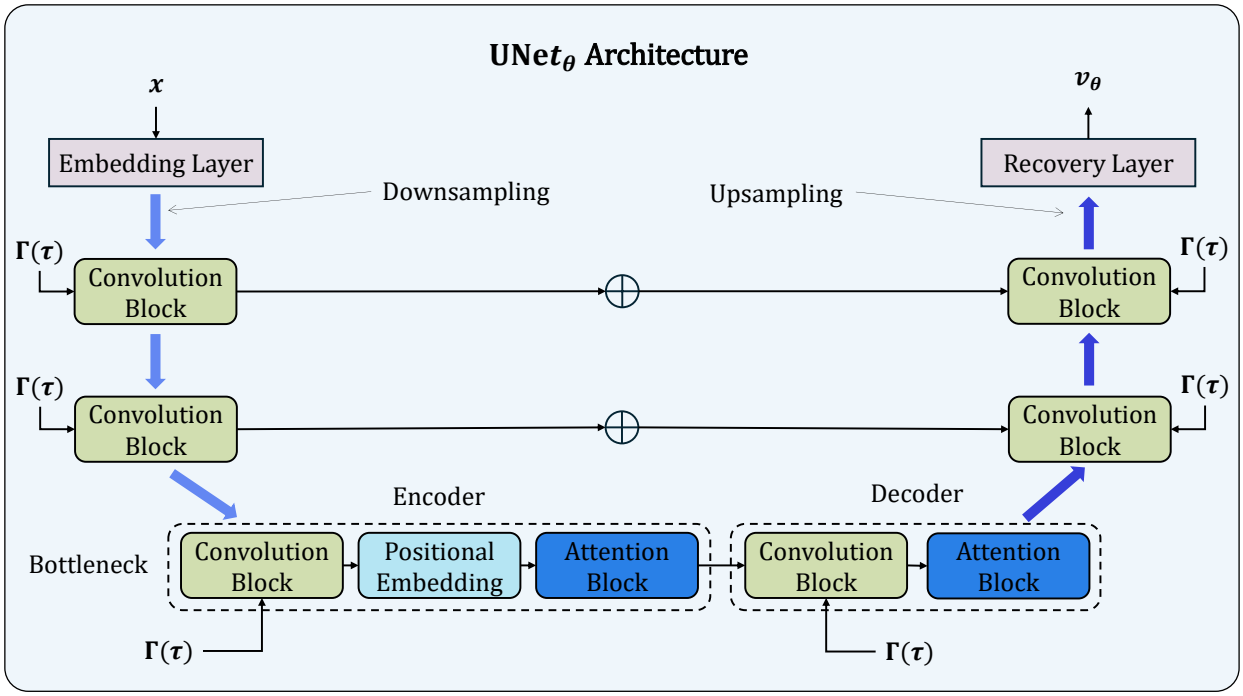


Figure 5: **UNet Backbone Architecture Used in ReFlow.** This schematic illustrates the core UNet-based architecture used within the ReFlow framework, structured across three resolution levels. For clarity, the number of blocks per level is set to one in this illustration. In the actual GenCFD and ReFlow configurations, each block is repeated four times per level. The bottleneck shows an asymmetry between the encoder and decoder sides: the block on the encoder side includes a Convolution Block, Positional Embedding, and Attention Block, while the corresponding block on the decoder side omits the Positional Embedding.

J.1 Architectural Components

Motivation. Multi-scale PDE data contain wide-ranging spatial scales, so a *multi-resolution* encoder–decoder is natural. We incorporate time embeddings to condition each layer on the interpolation fraction t.

Architecture Layout. We employ a UNet-based architecture augmented with attentional layers, typically termed as **UViT** by [26]. In addition to the MLP-based time embedding $\Gamma(\tau)$, the input signal is lifted into

a higher-dimensional embedding space and later projected back into physical space through convolutional layers. The data flows from high to low resolution across a down-sampling stack composed of convolution blocks into a bottleneck layer that encodes the fine-scale features of the inputs across its channel dimension and mixes them via a multi-head attention mechanism.
On the encoder side, exclusively at the bottleneck level, each attention block preceded by a simple linear positional embedding. This positional encoding helps preserve spatial locality before global mixing through attention. The decoder side does not apply any positional embedding.
The UNet is conditioned at every level through normalized Feature-wise Linear Modulation (FiLM) techniques embedded within the convolution blocks. The conditioning signal modulates the normalization layers via learned scale and shift parameters. This conditioning mechanism is consistently applied across all encoder and decoder stages.
The output is symmetrically reconstructed via an up-sampling pipeline. Downsampling is performed using standard convolutional layers with stride, while upsampling can be implemented either via transposed convolutions or by applying a standard convolution followed by a non-learnable pixel shuffle operation, which rearranges elements of the tensor spatially according to a fixed upsampling factor.
Importantly, the architecture described here resembles the one presented in [21], ensuring consistency with prior state-of-the-art practices in high-fidelity generative modeling for scientific computing.

J.1.1 Convolution Block

Each UNet level employs dedicated convolution blocks that process features while enabling conditioning on the diffusion time τ. These blocks consist of two convolutional layers interleaved with group normalization and Swish activations. Between the convolutions, a normalized FiLM layer adaptively scales features using the conditioning embeddings. A residual connection combines the processed features with a projected skip connection, ensuring stable gradient flow. Crucially, these convolution blocks form the fundamental building units across all encoder and decoder stages, not just the bottleneck, with consistent application of temporal conditioning through the FiLM mechanism.

J.1.2 Normalized FiLM Layer

Each convolution block uses a normalized FiLM layer and applies a convolution layer interleaved with normalization and an activation, then adds a residual skip:

$$\begin{aligned} \mathbf{z} &= \mathrm{Conv}\big(\mathrm{Norm}(\mathbf{x})\big), \qquad \mathbf{z} = \sigma(\mathbf{z} + \alpha\,\gamma(t)), \\ \mathbf{x}_{\mathrm{out}} &= \mathbf{x} + \mathrm{Conv}\big(\mathbf{z}\big), \end{aligned} \tag{35}$$

where Norm is often `RMSNorm`, GroupNorm or LayerNorm, σ is a nonlinearity (e.g. SiLU), and $\gamma(t)$ represents the time embedding (an MLP on Sinusoid(t)) that modulates scale/shift. This temporal conditioning is crucial for guiding the generative process through rectified diffusion trajectories.

J.1.3 Attention Block

The architecture strategically employs multi-head attention exclusively at the bottleneck layer to balance global interaction modeling with computational efficiency. Each attention block processes normalized features through a spatial-channel reformatting, where features are reshaped by a flattening operation. This preserves the channel structure while collapsing the spatial grid into a single flattened dimension, allowing attention to operate across spatial locations. At certain scales, a multi-head attention (MHA) mechanism captures distant flow interactions. We define

$$\mathbf{Q} = W_Q\,\mathbf{h}, \quad \mathbf{K} = W_K\,\mathbf{h}, \quad \mathbf{V} = W_V\,\mathbf{h}, \quad \mathcal{A}(\mathbf{h}) = \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\Big)\,\mathbf{V}, \tag{36}$$

where $\mathbf{h} \in \mathbb{R}^{(H\cdot W)\times d}$ denotes the flattened input sequence, with each of the $H\cdot W$ spatial positions represented as a d-dimensional feature vector. In practice, attention is split into multiple heads for better representational power, then recombined. This global mixing is key for PDE flows dominated by far-field couplings (e.g. vortex merging). The MHA operator is defined as

$$\mathrm{MHA}(\mathbf{h}) = W^O\,\mathrm{Concat}\left(\mathcal{A}_1(\mathbf{h}), \ldots, \mathcal{A}_n(\mathbf{h})\right), \tag{37}$$

where $n \in \mathbb{N}$ denotes the number of attention heads and $W^O \in \mathbb{R}^{C \times C}$ is a learned output projection matrix. The attention block applies MHA to a normalized input, then adds a residual connection to preserve the original signal and stabilize training while enabling global context to influence local features:

$$\begin{aligned} \mathbf{z} &= \mathrm{MHA}\big(\mathrm{Norm}(\mathbf{x})\big), \\ \mathbf{x}_{\mathrm{out}} &= \mathbf{x} + \mathbf{z}. \end{aligned} \tag{38}$$

J.2 Training Pipeline Details

The training pipeline follows the algorithm described in the main text faithfully.

Trainer Workflow. A high-level `Trainer` class manages:

1. Batching PDE pairs (u_0, u_1). These normalized pixel-wise with statistics extracted per each entire dataset.
2. Sampling $\tau \in [0, 1]$ and random noise ξ according to a given noise scheduler $\sigma(\cdot)$.
3. Partial interpolation + noise addition: $\tilde{u}_\tau := \tau\, u_1 + (1 - \tau)\, \xi$ (or the flow-based variant). We have experimented with non-linear interpolation methods, but have not noticed meaningful oscillations in performance.
4. Minimizing the squared deviation between model output and target displacement/noise, optionally with a consistency/EMA update.
5. Throughout training, we monitor validation performance on a separate holdout dataset across all main metrics. We explicitly note that different validation splits have been tested out to ensure there is no data leakage that would bias performance metrics.

Table 5: Representative hyperparameters for 2D fluid-flow tasks. See main text for justification of these choices.

Parameter	Value (Optimal empirical performance)	Notes
Channels per level	(128, 256, 256)	Number of channels per level.
Downsample Ratios	(2, 2, 2)	Downsampling ratio per level.
Attention Heads	8	Each attention block.
Dropout Rate	0.1	In ConvBlock for generalization.
Noise Schedule	`uniform`, `log-uniform`, `cos-map`	Helps with multi-scale noise.
ODE Steps (Sampling)	4–8	Often $> 15\times$ fewer vs. diffusion.
Batch Size	16	Tied to GPU capacity.
EMA Decay	0.9999	Teacher model for consistency.
Learning Rate	3×10^{-4}	Cosine decay in code.

In practice we noticed that the most robust denoising schedule was just the `uniform` one. For details on the `cos-map` schedule, we direct the reader to its description in [2].

J.3 Summary and Key Takeaways

The `RectifiedFlow` class, combined with a time-aware `UNet2D` architecture, builds a deterministic transport model well-suited to multi-scale PDE tasks. By interpolating between noise and data along nearly-straight paths, it reduces sampling overhead compared to conventional diffusion PDE solvers while retaining the flexibility to model complex turbulent phenomena. Further experimental and theoretical results appear in **SM** Section M. Instead of the multi-step *stochastic* integration in diffusion-based models, `RectifiedFlow` fits a *deterministic* ODE from noise to data. This leverages straighter trajectories in frequency space, enabling fewer network evaluations at inference time.

ODE Integration for Sampling. At inference:

$$\frac{d\tilde{u}_\tau}{d\tau} = v_\theta\big(\tilde{u}_\tau,\, u_0,\, \tau\big), \quad \tau \in [0,1], \quad \tilde{u}_{\tau=0} = \text{random noise}.$$

We solve this with `odeint`. Few steps (8–10) typically suffice, contrasting with the tens or hundreds required by full diffusion samplers.

K Datasets

K.1 Multi-scale Flow Datasets

Following the dataset generation procedure of Herde et al. [10], the Richtmyer–Meshkov ensemble is created by imposing a randomized sinusoidal perturbation on a two-fluid interface and driving it with a planar shock via prescribed pressure jumps. For the CloudShock and ShearFlow benchmarks, we follow the GenCFD mesh-generation and solver setup of Molinaro et al. [21], initializing CloudShock with concentric density perturbations and ShearFlow with orthogonal shear jets on adaptive Cartesian grids, and employing the same high-order finite-volume scheme and dissipation settings to ensure consistent numerical fidelity across both datasets.

Overview. We experiment on three paired-field benchmarks (u_0, u_1):

- **Richtmyer–Meshkov (RM)**: 64 000 samples
- **Cloud-Shock (CS)**: 40 000 samples

- **Shear-Layer (SL)**: 79 200 samples

All datasets track the $u_0 \to S^t(u_0)|_{t=1}$ solution mapping from inputs to the evolved outputs at $t = 1$. The first two datasets monitor the evolution of density (ρ), momentum components (m_x, m_y), and energy respectively (E). The shear-flow dataset monitors the velocity components of the flow (u_x, u_y).

Train/Validation Split. For each dataset, we reserve 80% of the samples for training and 20% for validation.

Macro–Micro Ensemble Evaluation (CS & SL). To probe out-of-distribution generalization under small perturbations, we adopt a two-stage ensemble protocol:

1. *Macro-sampling:* select $M_{\text{macro}} = 10$ distinct base initial conditions.
2. *Micro-perturbation:* for each base, generate $M_{\text{micro}} = 1{,}000$ perturbed copies within a small radius.
3. *Metrics:* compute all performance measures (mean-field error e_μ, standard-deviation error e_σ, average W_1, spectral agreement) by first averaging over each micro-ensemble, then averaging across the M_{macro} cases.

In-Distribution Evaluation (RM). For the Richtmyer–Meshkov dataset, we additionally evaluate on a held-out test set drawn from the same distribution (no macro–micro perturbations). This allows us to study how deterministic and probabilistic models scale with training set size under standard, in-distribution conditions.

K.2 Preprocessing and Data Organization

Data Processing

Before training, we compute and store global, channel-wise statistics over the entire training dataset. Concretely, for each physical variable c we calculate

$$\mu_c = \frac{1}{N_{\text{train}}\, H\, W\, [D]} \sum_{i=1}^{N_{\text{train}}} \sum_{x,y,[z]} u_c^{(i)}\big(x, y, [z], t_0\big),$$

$$\sigma_c = \sqrt{\frac{1}{N_{\text{train}}\, H\, W\, [D]} \sum_{i=1}^{N_{\text{train}}} \sum_{x,y,[z]} \Big(u_c^{(i)}\big(x, y, [z], t_0\big) - \mu_c\Big)^2}.$$

where N_{train} is the number of training samples, $H \times W\, [\times D]$ the spatial grid size, and $u_c(x, y, [z], t_0)$ the initial field for channel c. We average over all spatial locations to yield a single mean μ_c and standard deviation σ_c per channel. The same statistics are also computed over the final-time fields $u_c(t_f)$, producing μ'_c and σ'_c for the output channels.

During both training and evaluation, each input field u_c is standardized via

$$\tilde{u}_c = \frac{u_c \,-\, \mu_c}{\sigma_c + 10^{-16}}, \qquad \tilde{v}_c = \frac{v_c \,-\, \mu'_c}{\sigma'_c + 10^{-16}},$$

where u_c and v_c denote the initial and target fields, respectively. We then concatenate $\{\tilde{u}_c\}$ and $\{\tilde{v}_c\}$ along the channel axis before feeding them to the network. Because we always apply the same pre-computed statistics at test time, no information from the evaluation set ever influences these normalization parameters.

Test-time ensemble perturbation (evaluation only). We adopt the same testing setup proposed in *GenCFD* [21]. To evaluate how well our model captures the *distribution* of solutions $u(t)$ arising from a fixed, chaotic initial field $\bar{u}$, we generate a small perturbed ensemble around $\bar{u}$ and evolve each member with a high-fidelity PDE solver—*but only at evaluation time*. Specifically:

1. **Micro-ensemble generation.** Around the test initial condition $\bar{u}$, draw M_{micro} perturbed copies $\{\bar{u}_j\}_{j=1}^{M_{\text{micro}}}$ by sampling uniformly in a ball of radius ε centered at $\bar{u}$.
2. **Reference propagation.** Integrate each perturbed field $\bar{u}_j$ forward to time t using the reference solver, producing end-states $\{u_j(t)\}$.
3. **Empirical conditional law.** Form the empirical measure

$$\hat{P}_t(\,\cdot \mid \bar{u}) \;=\; \frac{1}{M_{\text{micro}}} \sum_{j=1}^{M_{\text{micro}}} \delta_{u_j(t)},$$

   which approximates the true chaotic conditional law $P_t(\,\cdot \mid \bar{u})$.

When the input distribution itself is non-degenerate, one would first draw M_{macro} base fields from that distribution and then apply the micro-ensemble procedure to each. For our Dirac initial-condition tests we set $M_{\text{macro}} = 1$.

Training vs. evaluation. During training the model only ever sees independent pairs (u_0, u_1)—one target per input—and never observes any ensemble. The above ensemble perturbation is used *only* to construct a ground-truth distribution at test time.

L Models and baselines

We tested our method against 4 other baselines, out of which 2 (namely GenCFD and its FNO-conditioned variant) are diffusion-based algorithms of the same kind. The other two baselines rely on traditional operator learning setup, with the UViT baseline satisfying also the purpose of an ablation study that quantifies the gain obtained by adding diffusion on top.

Shared Backbone. We stress that for GenCFD, GenCFD $\circ$ FNO and UViT we use the exact same UViT backbone architecture outlined in the earlier parameterization of our rectified flow as in Table 5 for the sake of consistency. The architecture is further illustrated schematically in Figure 5, which shows the UNet-based design in detail.

GenCFD

GenCFD [21] is an end-to-end conditional score-based diffusion model that directly learns the mapping from an input flow field u_0 to the target field u_1. It uses the same UViT backbone described in Table 5, with noise levels σ_τ corresponding to the standard deviation of the added Gaussian perturbation, in contrast to our ReFlow model which evolves over a rectified diffusion time $\tau \in [0, 1]$. Common choices for noise schedulers include exponential and tangent formulations. In our case, we consistently adopted an exponential noise schedule, as it yielded the best results across all datasets and metrics reported. For score-based diffusion models, either variance-exploding (VE) or variance-preserving (VP) formulations are typically used. In all GenCFD models, we opted for the VE setting throughout. During training we minimize the denoising score-matching loss over (u_0, u_1) pairs sampled uniformly in $\tau \in [0, 1]$, using Adam with initial learning rate 3×10^{-4} (cosine decay), batch size 16, and EMA decay 0.9999. As is common in score-based diffusion models, we applied denoiser preconditioning and used uniform weighting across noise levels. At inference we solve the corresponding SDE (via an Euler-Maruyama scheme with discretization 128 steps) and incorporate reconstruction guidance to condition on u_0 so that we recover the conditional posterior measure $p(u_1|u_0)$.

GenCFD ∘ FNO

In this hybrid approach, a Fourier Neural Operator (FNO) [14] is first trained under an ℓ_2 loss to predict u_1 from u_0. Its output $\hat{u}_1^{\text{FNO}}$ is concatenated channel-wise with u_0 and fed into the GenCFD score network at both train and test time. All other architectural and optimization settings (backbone, scheduler, SDE sampler) remain identical to GenCFD, allowing the FNO to provide fast low-frequency priors while the diffusion stage refines fine-scale turbulent features.

Custom implementation. This schematic closely parallels the pipeline of [22], but with two key distinctions. First, we fully replace both the diffusion backbone and its denoising stages with GenCFD. Second, and most important, we condition not only on the FNO's coarse solution but also on the original high-resolution initial condition. This ensures that our model retains fine-scale structures that would otherwise be lost if we relied solely on the FNO predictions.

FNO

Our FNO baseline follows Li *et al.* [14] and is implemented in PyTorch. First, each spatial location of the input field u_0 is lifted from C_{in} to 256 channels via a two-layer MLP with a SiLU activation, then projected down to a 128-channel hidden representation. This is followed by four `FnoResBlock`s: in each block, the hidden features undergo a Conditional LayerNorm (time embedding size 128), SiLU, and two spectral convolutions (`SpectralConv`) with $num_modes = (24, 24)$, combined through a learnable "soft-gate" residual skip. Finally, a two-layer projection MLP ($128 \to 256 \to C_{\text{out}}$, with SiLU in between) produces the output field $\hat{u}_1$. We train for 100k iterations minimizing $\|u_1 - \hat{u}_1\|_2^2$ with Adam (learning rate 10^{-3}, batch size 16), and perform inference in one forward pass—no diffusion or autoregression is used.

UViT

Our UViT surrogate [26] uses the identical UNet2D backbone of Table 5, injecting time embeddings and employing 8-head multi-head attention at the bottleneck. It is trained deterministically with MSE loss on (u_0, u_1), using Adam at 3×10^{-4}, batch size 16, for 100 k iterations (EMA 0.9999). For multi-step forecasting we roll out autoregressively, feeding each prediction back as the next input. This ablates out the diffusion component while matching all other design choices.

Baseline Model Summary and Implementation Setup

Table 6 summarizes key architectural settings for all baseline models considered in this work. These configurations were selected to reflect the strongest performance achievable within a shared training budget of up to 120,000 iterations, as used in the quantitative comparison reported in Table 1.

Padding and boundary conditions. Each dataset required padding choices tailored to its physical boundary conditions. SL and RM, which exhibit periodic boundaries, use circular padding in all convolutional layers. This significantly improved the performance of the diffusion-based models. In contrast, CS features non-periodic boundaries and is thus trained with standard zero padding. These padding choices apply to all UNet-based architectures except ReFlow, which uniformly uses zero padding across all datasets regardless of boundary conditions.

M Results

M.1 Performance Metrics

To compare our generated ensembles against the reference Monte Carlo samples, we employ three complementary measures:

Table 6: Summary of architectural depth and approximated parameter count for each model evaluated. All UNet-based models (ReFlow, GenCFD, GenCFD ∘ FNO, and UViT) share a consistent backbone design to ensure consistency across diffusion and non-diffusion baselines, while the FNO baseline uses a structurally different architecture. These configurations reflect the best-performing settings under a training budget of up to 120,000 iterations, as used in the statistical comparisons reported in Table 1.

Model	**# Levels**	**# Blocks**	**# Params**	**Noise Schedule**	**Diffusion Scheme**
ReFlow	3	4	22M	uniform	rectified continuous time variable $t \in [0, 1]$
GenCFD	2	4	5M	exponential	variance-exploding
GenCFD ∘ FNO	2	4	5M	exponential	variance-exploding
UViT	2	4	5M	—	—
FNO	—	4	11M	—	—

- **Mean-field error.** Let $\mu_{\text{ref}}(x)$ and $\mu_{\text{model}}(x)$ denote the pointwise spatial means of the reference and model ensembles, respectively. We measure their discrepancy by the L^2-norm

$$e_\mu \;=\; \big\|\, \mu_{\text{ref}} \;-\; \mu_{\text{model}} \big\|_2 \,.$$

- **Standard-deviation error.** Similarly, let $\sigma_{\text{ref}}(x)$ and $\sigma_{\text{model}}(x)$ be the pointwise standard deviations. We therefore compute

$$e_\sigma \;=\; \big\|\, \sigma_{\text{ref}} \;-\; \sigma_{\text{model}} \big\|_2 \,,$$

  Both metrics are reported after normalization by the L^2 norm of the ground truth.

- **Average 1-Wasserstein distance.** For a fixed initial condition u_0, let $\{u^{(j)}_{\text{ref}}(x)\}_{j=1}^N$ and $\{u^{(j)}_{\text{model}}(x)\}_{j=1}^N$ be the ensembles of final-time values produced by the reference solver and our model, respectively. Since these fields are discretized on M spatial points $\{x_i\}_{i=1}^M$, we obtain two empirical 1D distributions at each x_i. Denoting their inverse CDFs by $F^{-1}_{\text{ref}}(\cdot\,; x_i)$ and $F^{-1}_{\text{model}}(\cdot\,; x_i)$, the pointwise Wasserstein-1 distance is

$$W_1\big(p_{\text{ref}}(\cdot \mid x_i),\, p_{\text{model}}(\cdot \mid x_i)\big) \;=\; \int_0^1 \big|\, F^{-1}_{\text{ref}}(q; x_i) \;-\; F^{-1}_{\text{model}}(q; x_i) \big|\, \mathrm{d}q.$$

  We then report the spatial average

$$\overline{W}_1 \;=\; \frac{1}{M} \sum_{i=1}^{M} W_1\big(p_{\text{ref}}(\cdot \mid x_i),\, p_{\text{model}}(\cdot \mid x_i)\big).$$

Energy spectra. Finally, to assess how well different scales are captured, we compute the discrete energy spectrum of each velocity field. Given a sample $u(x)$ on a d-dimensional periodic grid with spacing Δ, let $\widehat{u}_k$ be its Fourier coefficient at integer wavevector $k \in \mathbb{Z}^d$. We bin modes by their ℓ_1 radius $|k|_1 = k_1 + \cdots + k_d$, defining

$$E_r \;=\; \frac{\Delta^d}{2} \sum_{\substack{k \in \mathbb{Z}^d \\ |k|_1 = r}} \big\|\widehat{u}_k\big\|^2, \quad r = 0, 1, 2, \ldots$$

and compare the ensemble-average spectra of reference and model. Our plots show E_r versus r on log–log axes to highlight scale-by-scale agreement.

M.2 Training Efficiency Observations

To assess training efficiency, we compare ReFlow with GenCFD. For fairness, we use a GenCFD variant with the same setup and parameter count (22M), matching the three-level UViT architecture of ReFlow instead of its standard lightweight configuration (5M). Both models were trained for up to 160,000 iterations with a batch size of 16.

Figure 6 summarizes performance over the training trajectory. At approximately 100,000 steps, ReFlow consistently outperforms GenCFD across all key metrics: relative L^2 error in the mean (e_μ), relative L^2 error in the standard deviation (e_σ), and the average pointwise Wasserstein-1 (W1) distance—evaluated for both u_x and u_y velocity components. Notably, the y-axis uses a logarithmic scale to emphasize differences in error magnitude across training steps.

Across all metrics, ReFlow begins to outperform GenCFD at around 100,000 training steps and continues to improve at a sharper rate. While neither model reaches full convergence within 160,000 iterations, ReFlow consistently exhibits lower errors and more favorable metric trajectories in later stages of training. These results support the observation that ReFlow is more training-efficient within a fixed compute budget and ultimately achieves stronger generalization performance in this setting.

To further probe the long-term capabilities of GenCFD, we extended its training to 600,000 iterations and checked whether it eventually surpasses ReFlow's performance at its final recorded checkpoint (150,000 steps). Table 7 summarizes the earliest training steps at which GenCFD overtakes ReFlow for each evaluation metric. While GenCFD does catch up and exceed ReFlow in several cases, it never surpasses ReFlow on the Wasserstein-1 distance for u_y within the training horizon considered.

Table 7: Number of training iterations at which GenCFD surpasses ReFlow's performance at 150,000 iterations. If not surpassed, indicated by "—".

Metric	GenCFD Iteration	GenCFD Value	ReFlow Value @ 150,000 iter.
u_x Mean Error (e_μ)	280,000	3.33×10^{-2}	3.45×10^{-2}
u_y Mean Error (e_μ)	400,000	1.70×10^{-1}	1.88×10^{-1}
u_x Std Error (e_σ)	400,000	6.48×10^{-2}	7.24×10^{-2}
u_y Std Error (e_σ)	340,000	7.91×10^{-2}	7.97×10^{-2}
u_x Wasserstein-1 (W_1)	200,000	3.89×10^{-2}	3.95×10^{-2}
u_y Wasserstein-1 (W_1)	—	—	4.11×10^{-2}

M.3 Experimental Advantages of Straightness

This subsection presents empirical evidence that the "rectified" trajectories generated by our Rectified Flow model (ReFlow) enable faster convergence and yield higher-fidelity reconstructions compared to the baseline GenCFD diffusion approach.

Evaluation settings. We use the CloudShock dataset throughout for exposition purposes. However, we stress that the same results hold across datasets. For each initial condition, both ReFlow and GenCFD are allotted $T = 10$ diffusion steps over normalized **diffusion time** $\tau \in [0, 1]$. We record the model outputs at five evenly spaced timesteps $t = \{0.0, 0.25, 0.50, 0.75, 1.0\}$ and analyze:

- *Trajectory inpainting* of the density field ρ.
- *Latent-Space Trajectory Visualization via PCA* via PCA.
- *Per-Sample Average Error Evolution*: average MSE across physical channels.
- *Evolution of the Energy Spectrum*: radial power-spectrum of energy E as inpaintig progresses.

All other components (backbone, noise schedule, batch size) are held equal for fair comparison.

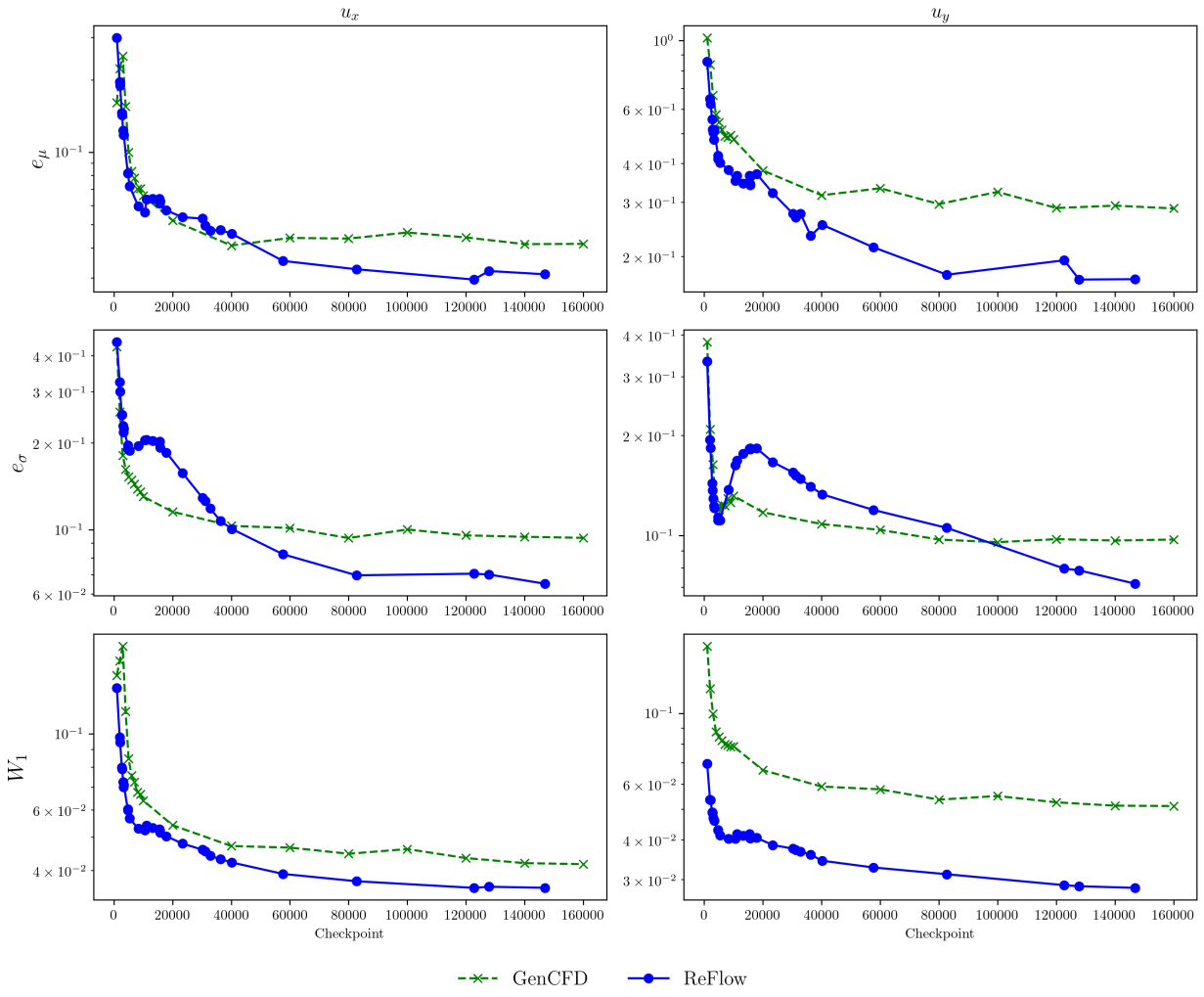


Figure 6: **Training efficiency comparison.** Performance of ReFlow and GenCFD on the Macro-Micro SL ensemble dataset evaluated at multiple training checkpoints. Metrics match those used in the main text: relative L^2 error of the mean (e_μ) and standard deviation (e_σ), and the average 1-point Wasserstein-1 distance (W1), computed for both velocity components (u_x, u_y). The y-axis uses a logarithmic scale, while the x-axis (training iteration) is linear.

Trajectory Inpainting

We compare Rectified Flow (ReFlow) and GenCFD reconstructions of the density field ρ over five normalized **diffusion** timesteps $t \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$. ReFlow inpaints fine-scale features in far fewer steps, while GenCFD remains overly diffusive. The leftmost panel corresponds to the initial condition u_0 upon which both model are conditioned, while the following panels showcase the evolution of the model's output as diffusion time progresses.

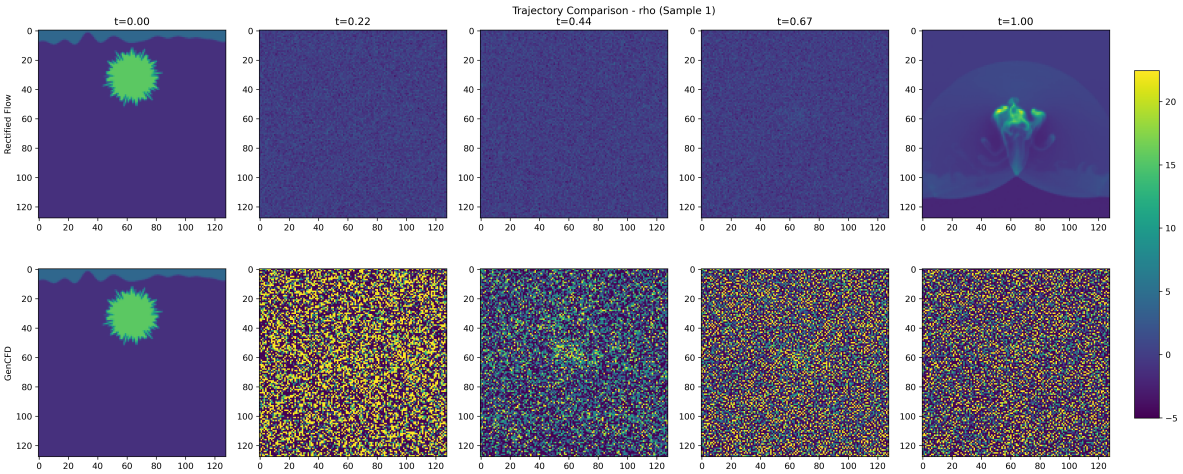


Figure 7: Density trajectories for Sample 1: ReFlow (top) vs. GenCFD (bottom). ReFlow already recovers sharp shock fronts by diffusion time $t = 1.0$.

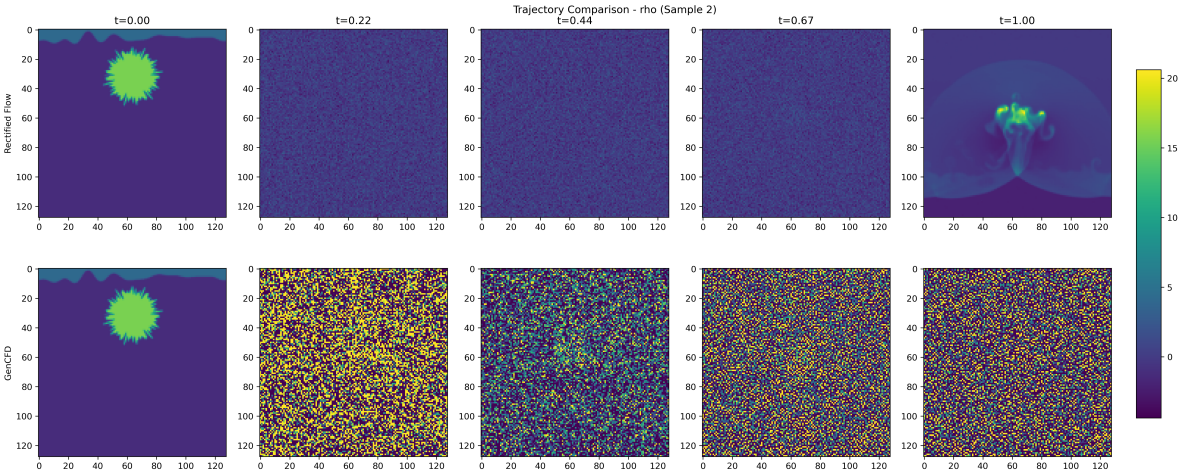


Figure 8: Density trajectories for Sample 2.

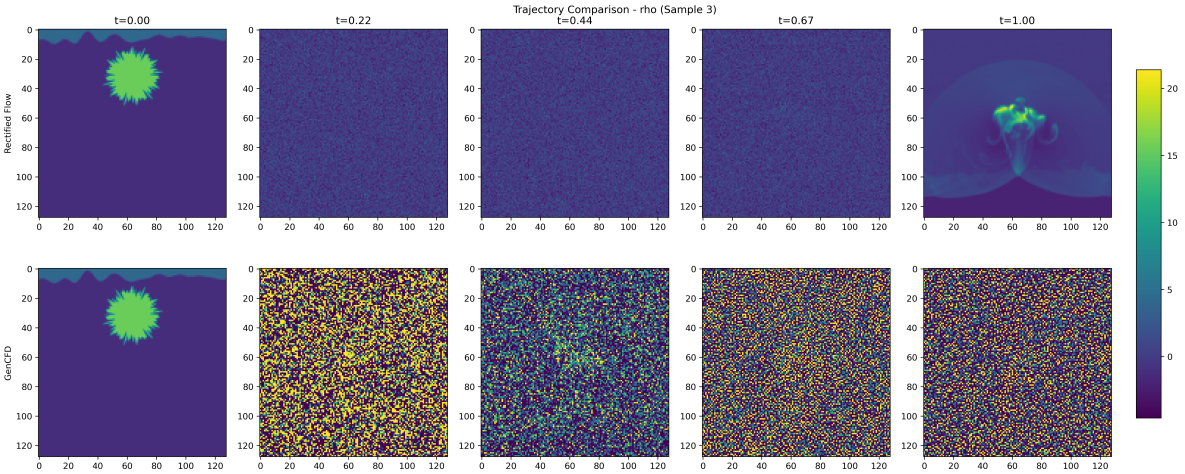
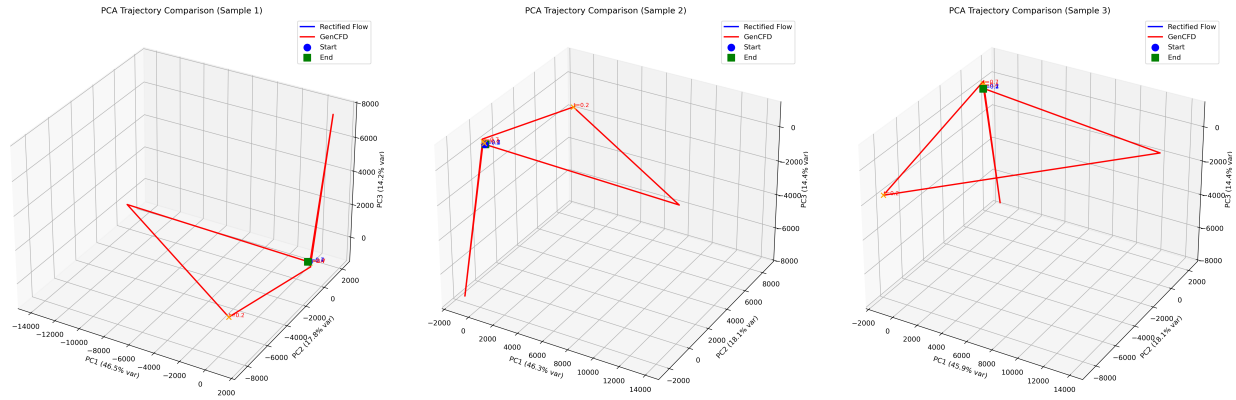


Figure 9: Density trajectories for Sample 3.

Latent-Space Trajectory Visualization via PCA

To quantify how directly each model moves through the data manifold, we flatten the $(H \times W \times C)$ field at each timestep into a single feature vector, stack all Rectified Flow and GenCFD vectors, fit a 3-component PCA, and then project each timestep into PC1/PC2/PC3. A straighter path indicates fewer diffusive detours.

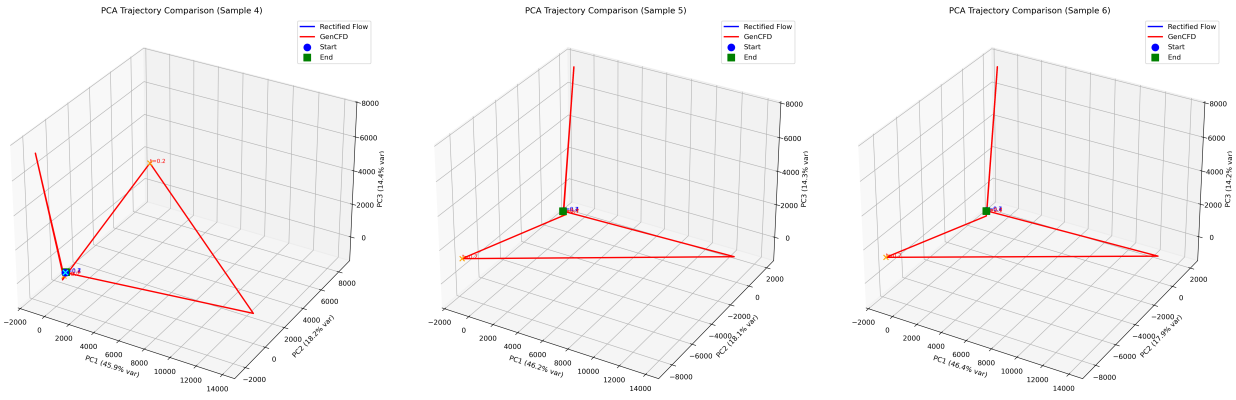
To further illustrate the nature of each model's generative dynamics, we visualize single-sample generation trajectories from both models in a 2D PCA space for the RM dataset (Figure 11). For each physical channel (ρ, m_x, m_y, p), we apply PCA to the full set of ground truth snapshots to define a common projection basis. We then overlay the generative trajectory of a single sample in this space. To contextualize the generated dynamics, we add kernel density estimates (KDEs) showing the empirical data distribution in the same projected space.
ReFlow (top) again traces a straighter, more compact path over just 10 rectified steps, moving coherently along the data manifold. GenCFD (bottom), by contrast, uses 128 stochastic denoising steps and produces a more diffuse and nonlinear trajectory, often deviating from high-density regions of the ground truth distribution. While the PCA projection inherently reduces dimensional complexity, the consistent difference in shape and alignment supports the conclusion that ReFlow's rectified flow better preserves manifold structure and sample efficiency.



(a) Sample 1

(b) Sample 2

(c) Sample 3



(d) Sample 4

(e) Sample 5

(f) Sample 6

Figure 10: **3D PCA latent-space trajectories (Samples 1–6).** We project the flattened snapshots at each of T timesteps into the top three PCA components. Rectified Flow (blue) consistently traces a straighter path than GenCFD (red), confirming its "rectified" evolution. Circles mark $t = 0$, squares mark $t = 1$, and intermediate timesteps are annotated.

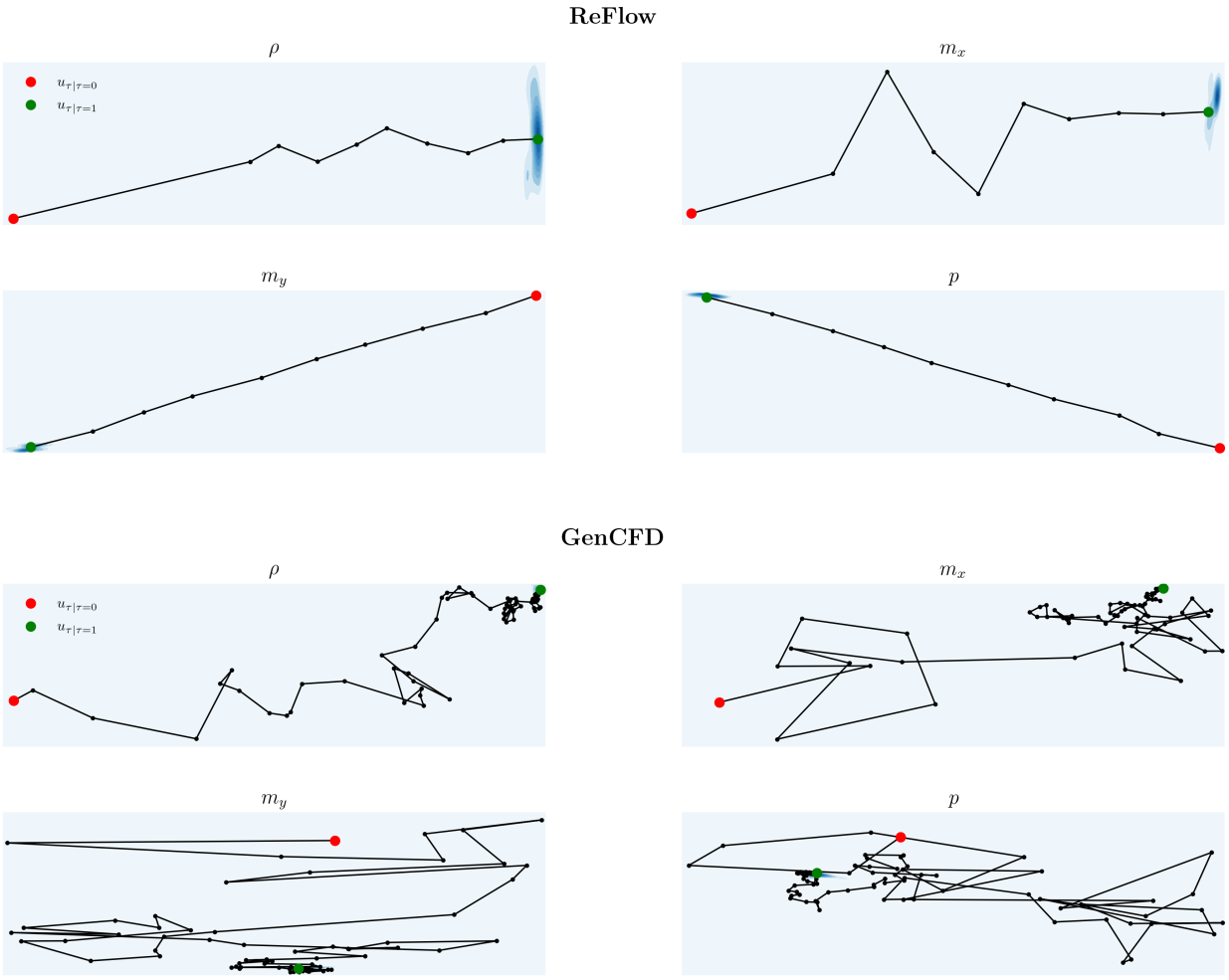


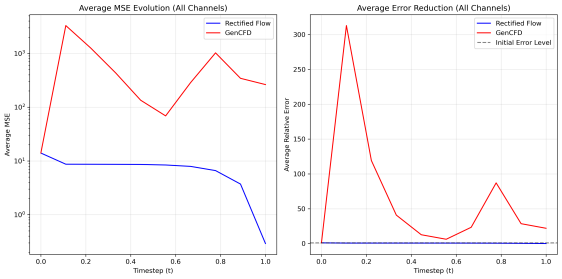
Figure 11: **2D PCA latent-space trajectories.** We project the flattened velocity snapshots from the RM dataset into a 2D PCA space, computed independently for each channel (ρ, m_x, m_y, p) across all T time steps. To visualize the underlying distribution of target fields, we apply kernel density estimation (KDE) per channel. The ReFlow model (top) evolves over 10 rectified time steps, producing visibly straighter and more structured trajectories compared to the score-based diffusion model GenCFD (bottom), which uses 128 stochastic steps. Although the projection into two principal components necessarily truncates the high-dimensional dynamics, the qualitative difference remains evident, illustrating the smoother and more directed progression learned by ReFlow.

Per-Sample Average Error Evolution

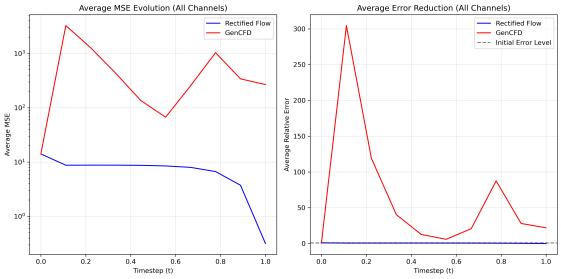
We compute the mean-squared error at each timestep t and average across all C channels:

$$\overline{\mathrm{MSE}}(t) = \frac{1}{C}\sum_{c=1}^{C} \mathrm{MSE}_c(t).$$

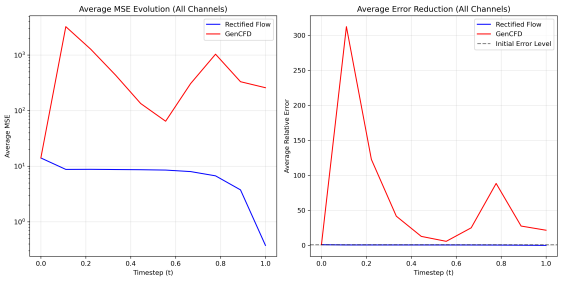
Figure 12 shows, for Samples 1–6, the log-scaled average MSE (left) and the normalized error $\overline{\mathrm{MSE}}(t)/\overline{\mathrm{MSE}}(0)$ (right). Rectified Flow (blue) consistently reduces error faster and to a lower residual than GenCFD (red).



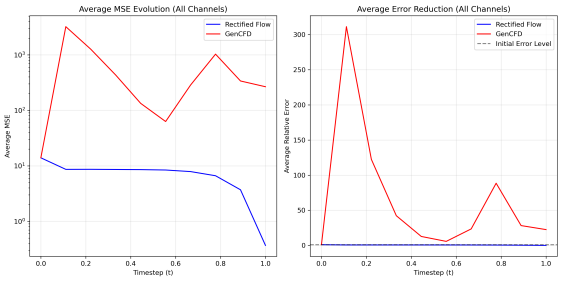
(a) Sample 1



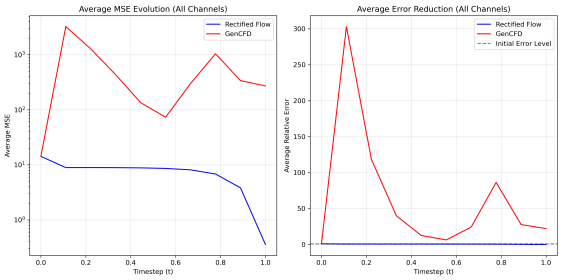
(b) Sample 2



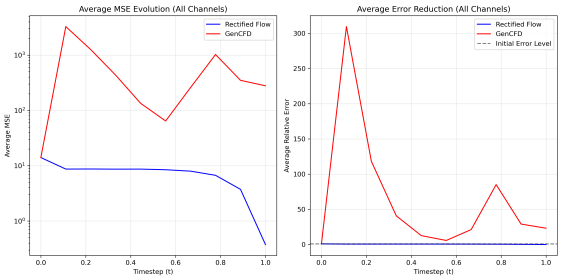
(c) Sample 3



(d) Sample 4



(e) Sample 5



(f) Sample 6

Figure 12: **Per-sample average MSE evolution.** Each subfigure plots, left half: $\log_{10} \overline{\mathrm{MSE}}(t)$; right half: $\overline{\mathrm{MSE}}(t)/\overline{\mathrm{MSE}}(0)$. Rectified Flow (blue) rapidly drives error down and keeps it orders of magnitude below GenCFD (red) across all samples.

Evolution of the Energy Spectrum

We examine the 2D radial power spectrum of the energy field E at five normalized diffusion times $\tau \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$. Each figure below shows log–log power vs. wavenumber for Rectified Flow (blue), GenCFD (red), and the ground-truth target (black dashed), with inset log-MSE error annotations. The leftmost snapshot corresponds to the spectrum of the initial data u_0 upon which the model is conditioned. Subsequent panels track the spectral evolution of the model predictions, as diffusion time progresses.

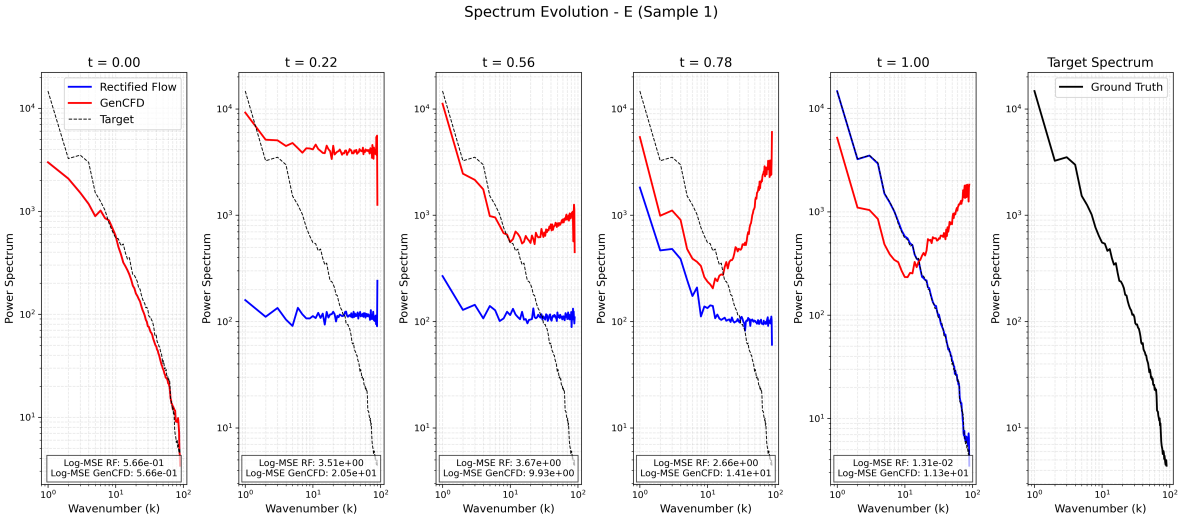


Figure 13: Energy spectrum evolution for Sample 1. ReFlow (blue) outperforms GenCFD (red) in capturing high-wavenumber energy.

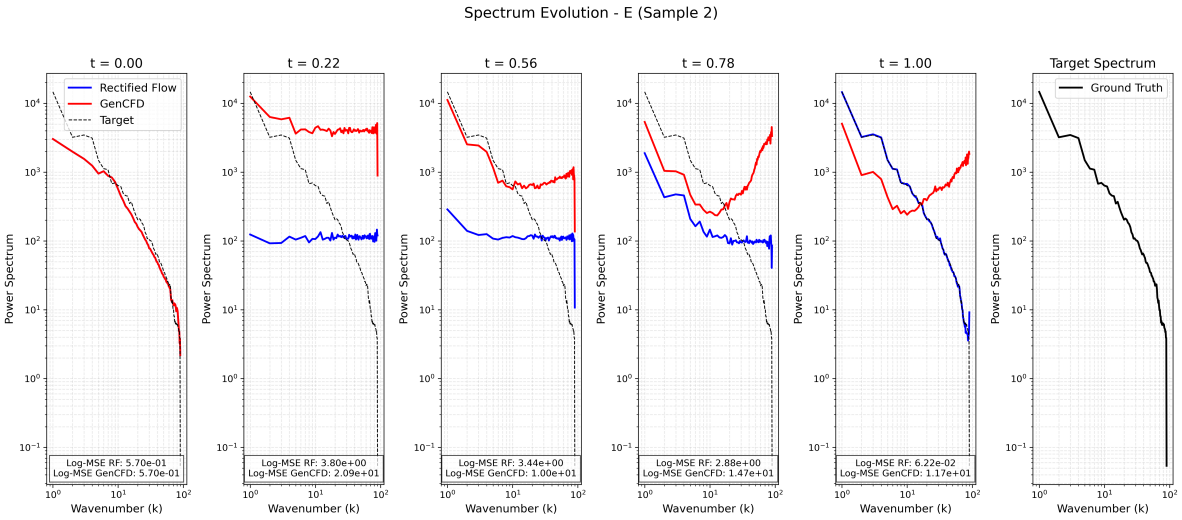


Figure 14: Energy spectrum evolution for Sample 2. ReFlow (blue) remains closer to the ground truth (dashed) than GenCFD (red).

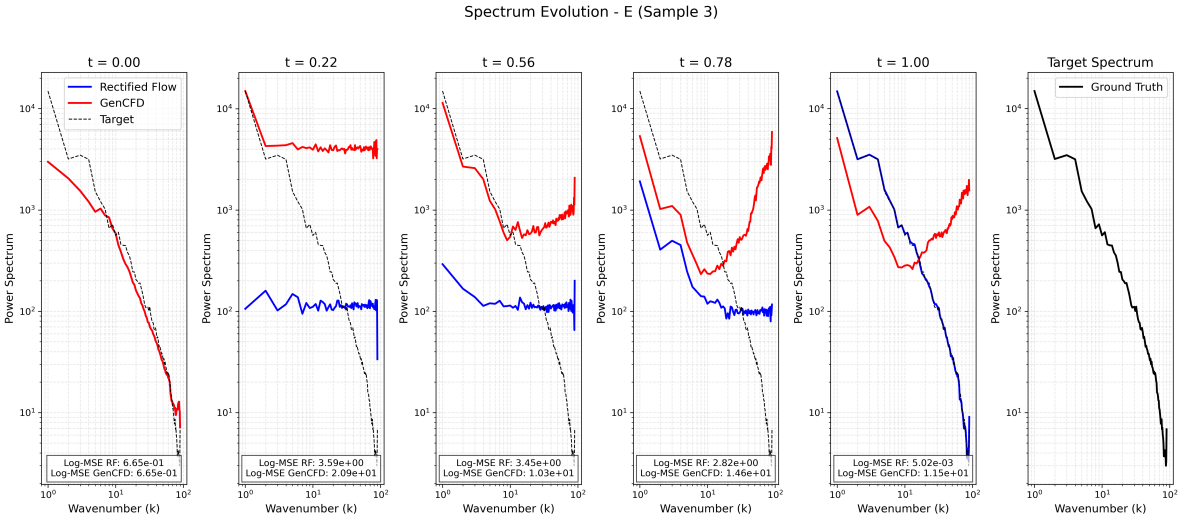


Figure 15: Energy spectrum evolution for Sample 3. The inset log-MSE annotations quantify ReFlow's advantage at each t.

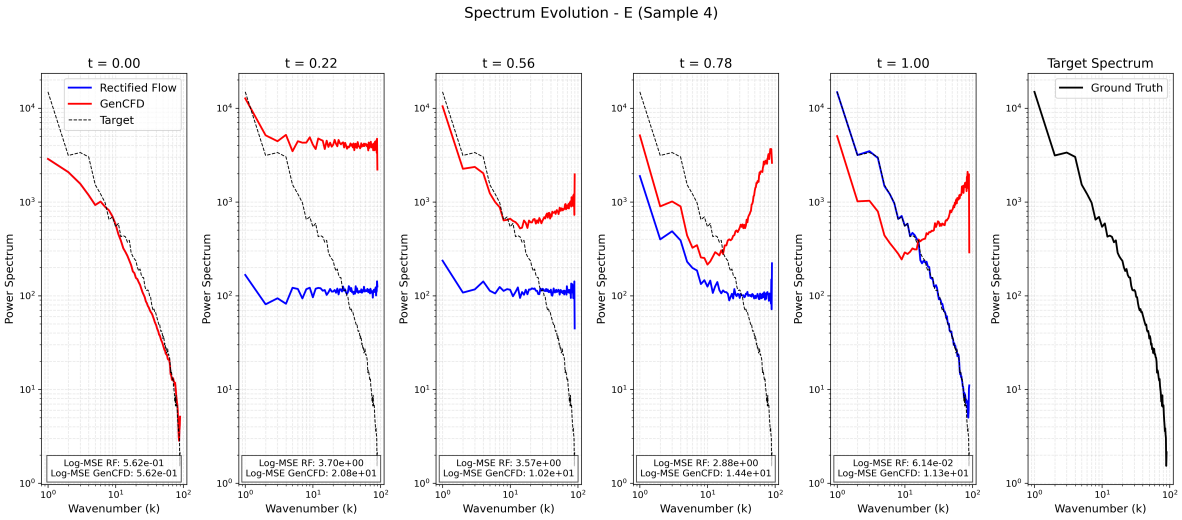


Figure 16: Energy spectrum evolution for Sample 4. ReFlow's spectrum (blue) gets closer to the target (dashed) as $\tau \to 1.0$.

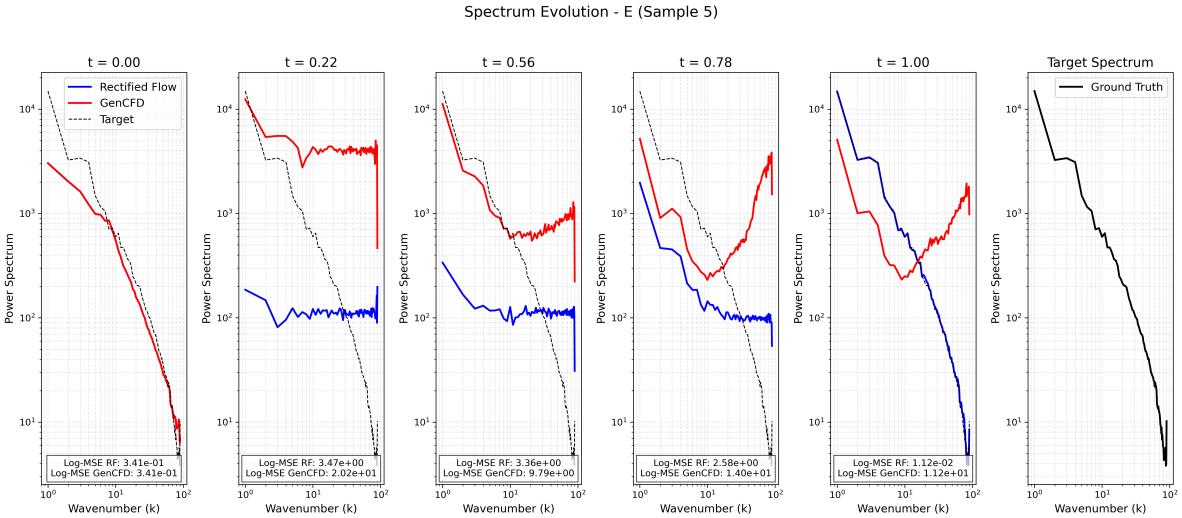


Figure 17: Energy spectrum evolution for Sample 5. ReFlow's superior high-k fidelity is evident throughout.

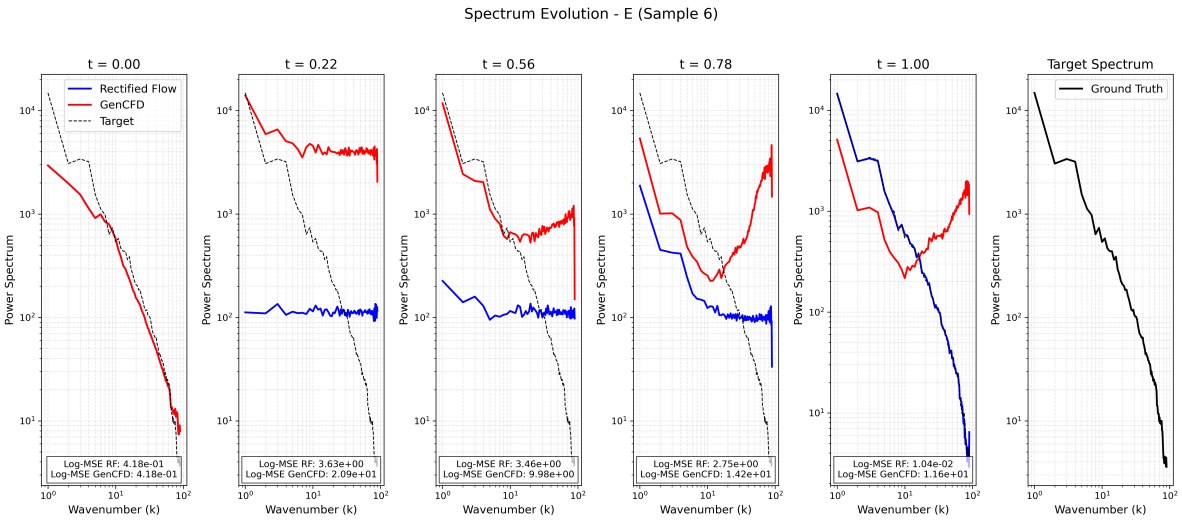


Figure 18: Energy spectrum evolution for Sample 6. ReFlow better captures the ground-truth spectrum's slope across all timesteps.

N Computational Considerations

All of our ReFlow models for 2D fluid-flow tasks use the same network configuration summarized in Table 5 (channels $(128, 256, 256)$, downsample ratios $(2, 2, 2)$, 8 attention heads, dropout 0.1, batch size 16, etc.). Consequently, every model—regardless of dataset—has the same total parameter count (approximately 8–10M parameters) and identical layer-wise dimensions. We ran each training job for 100 000 iterations on a single NVIDIA Quadro RTX 6000 (Driver 570.124.06, CUDA 12.8) in persistence mode. Resource utilization averaged as follows over the full 100 h window:

Table 8: Aggregate hardware utilization for all "usual-setup" runs.

Metric	Range	Typical
GPU power draw (W)	200–260	230
GPU utilization (%)	30–80	55
GPU memory allocated (GB)	12–17	15
GPU temperature (°C)	40–60	50
Process GPU memory usage (%)	60–80	70
Process GPU memory access time (%)	10–50	30
Process CPU threads in use	7	7
System CPU utilization (per core, %)	60–100	80
Process memory in use (GB)	up to 50	25
Disk I/O read (MB total)	up to 1.2×10^6	5×10^5
Disk utilization (%)	70–75	72
Network traffic (bytes total)	$\approx 3 \times 10^{13}$	3×10^{13}

Since all datasets are paired with models of identical size and the hardware envelope (GPU power, memory, temperature, CPU threads, I/O) is effectively the same across runs, any observed performance differences are attributable to the sampling strategy (noise schedule, ODE steps) rather than to model-size or hardware variations (the network traffic is due to Wandb usage).

All experiments were run on a single NVIDIA Quadro RTX 6000 GPU (Driver 570.124.06, CUDA 12.8), operating in persistence mode.

Each model was trained for 100 000 iterations. We explored architectures ranging from approximately 5 million to 10 million trainable parameters; across this range we observed no significant differences i n final predictive performance or convergence behavior. All other hyperparameters (learning rate schedules, batch sizes, noise schedules, etc.) were held fixed when evaluated against each other, ensuring that differences in accuracy and runtime reflect only the conditioning and sampling strategies under study.

O Code and Data Availability

Prior to acceptance, we will make available all three datasets, the complete data-processing scripts, and a fully functional training and evaluation pipeline for ReFlow, including a provided model checkpoint that can be loaded and tested. A more comprehensive, user-friendly code release will follow shortly thereafter.

P Additional Visualizations

Test-time ensemble perturbation (evaluation only). Following the protocol of *GenCFD* [21], we construct small "micro-ensembles" around each test initial condition $\bar{u}$ by sampling M_{micro} perturbed fields uniformly in an ε–ball centered at $\bar{u}$. Each perturbed member is then advanced to the evaluation time t with a high-fidelity PDE solver to produce a reference ensemble $\{u_{\text{ref}}^{(j)}(t)\}_{j=1}^{M_{\text{micro}}}$. Our model generates a corresponding ensemble $\{u_{\text{model}}^{(j)}(t)\}_{j=1}^{M_{\text{micro}}}$ from the same perturbed inputs, but without ever observing ensembles during training (which only uses independent pairs (u_0, u_1)). We compare these two ensembles

via mean-field error e_μ, standard-deviation error e_σ, and spatially averaged 1-Wasserstein distance $\overline{W}_1$ (see **SM** §M).

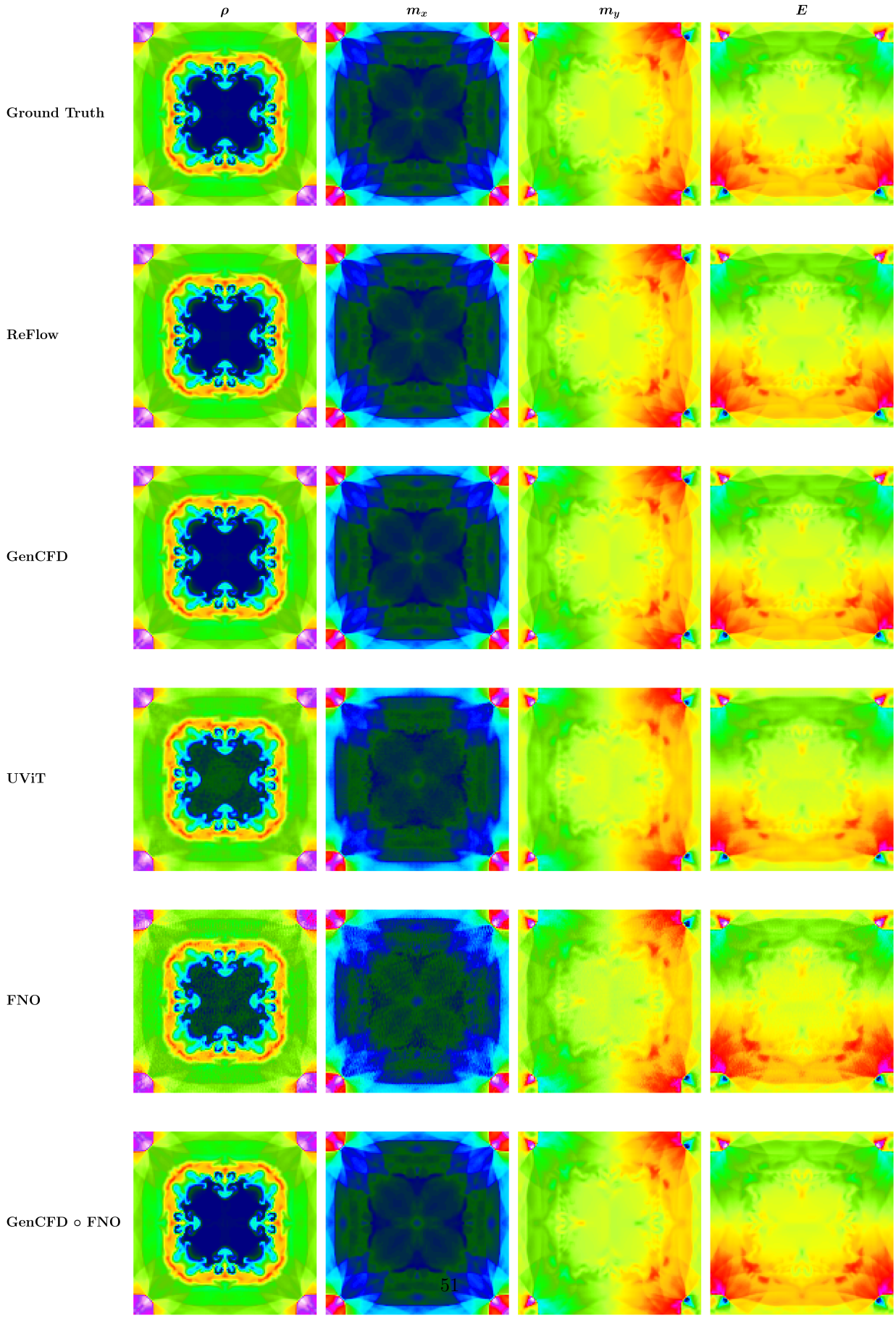
Out-of-distribution testing (cloud–shock and shear-flow). Because the micro-perturbations render each initial field slightly OOD, this evaluation probes the model's ability to generalize under small chaotic perturbations. In these tasks, our Rectified Flow ensemble closely tracks the reference uncertainty and multiscale statistics, whereas deterministic baselines collapse to a single trajectory and fail to capture the conditional variability.

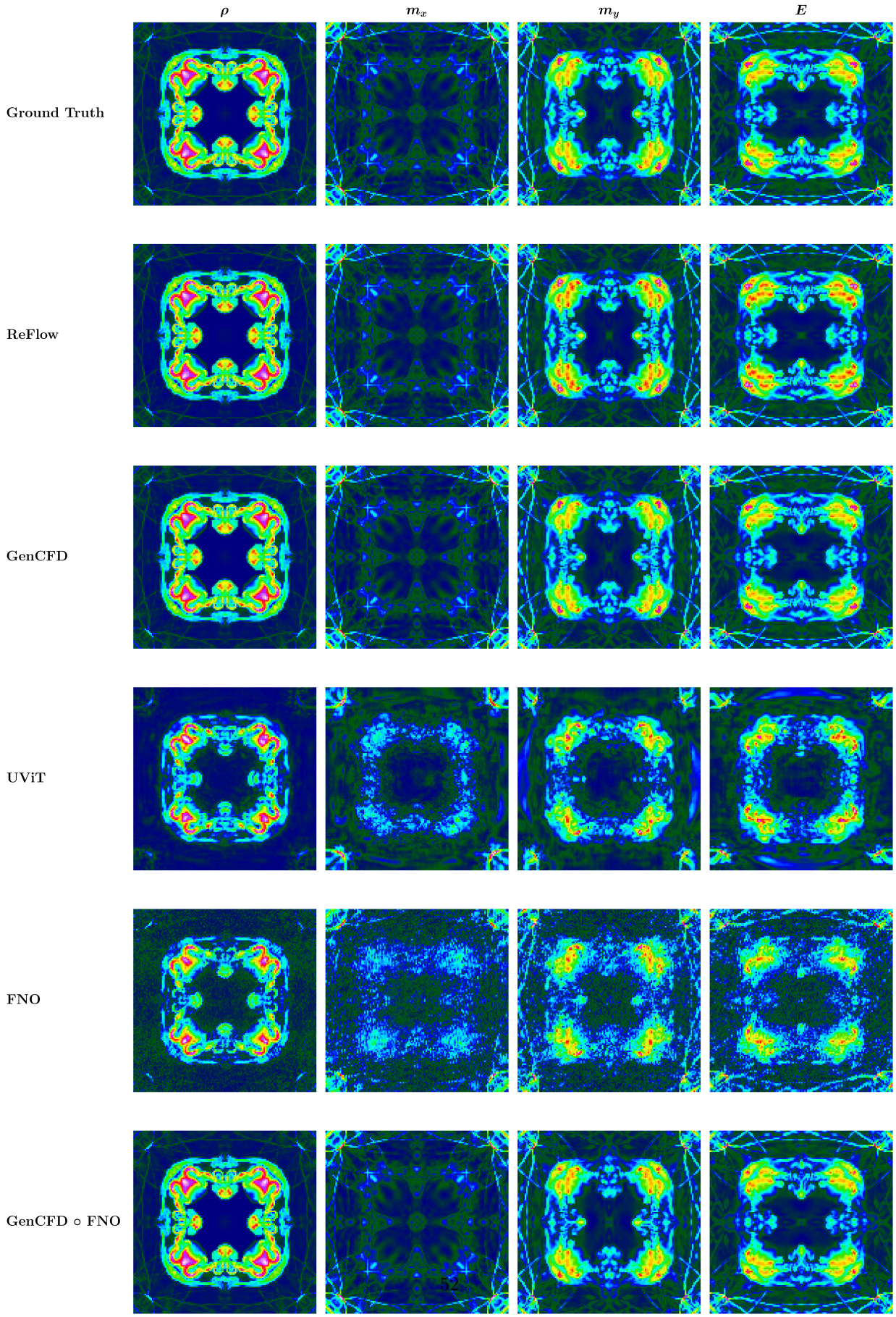
In-distribution scaling (Richtmyer–Meshkov). For the Richtmyer–Meshkov dataset, we additionally study the effect of training set size using a held-out test set from the same distribution. Here, even deterministic approaches improve with more data, but still underperform on the micro-ensemble test: they require significantly larger training sets to match ReFlow's accuracy and exhibit poor calibration under small perturbations.

In this appendix we provide a comprehensive set of qualitative comparisons between the baselines (GenCFD variants, UViT, FNO) models and our Rectified Flow (ReFlow) approach across three representative 2D fluid-flow benchmarks (Richtmyer–Meshkov, shear layer roll-up, and cloud-shock interaction). For each task we show:

- **Mean fields** Ensemble averages reveal how each deterministic model diffuses fine-scale features (UVit/FNO) versus perserving sharp interfaces (ReFlow, GenCFD variants).
- **Uncertainty maps** Per-pixel standard deviations illustrate whether uncertainty localizes in physically meaningful regions (e.g. shocks, shear interfaces) or remains overly diffuse.
- **Random samples** To qualitatively assess generative fidelity, we include representative random samples from ReFlow alongside outputs from selected baselines, specifically the deterministic FNO and the hybrid GenCFD $\circ$ FNO. The latter combines FNO's low-frequency structural prediction with GenCFD's generative modeling of fine-scale turbulence. This comparison highlights ReFlow's ability to produce realistic, high-resolution flow fields that preserve physical coherence across different scales.
- **Spectral & pointwise comparisons** (Fig. 22: Richtmyer–Meshkov global $E(k)$, local spectra, and pointwise density histograms; Fig. 25: Shear-layer global u_x spectra and pointwise u_x distributions; Fig. 29: Cloud–shock global m_y spectra and pointwise energy distributions) — combined Fourier-space and histogram panels quantify multiscale fidelity and calibration of the predictive ensembles.

Together, these visualizations underscore ReFlow's ability to maintain sharper physical structures, produce better-calibrated uncertainties, and faithfully reproduce multiscale spectral statistics.


ρ
m_x
m_y
E
Ground Truth
ReFlow
GenCFD
UViT
FNO
GenCFD ∘ FNO


ρ
m_x
m_y
E
Ground Truth
ReFlow
GenCFD
UViT
FNO
GenCFD ∘ FNO

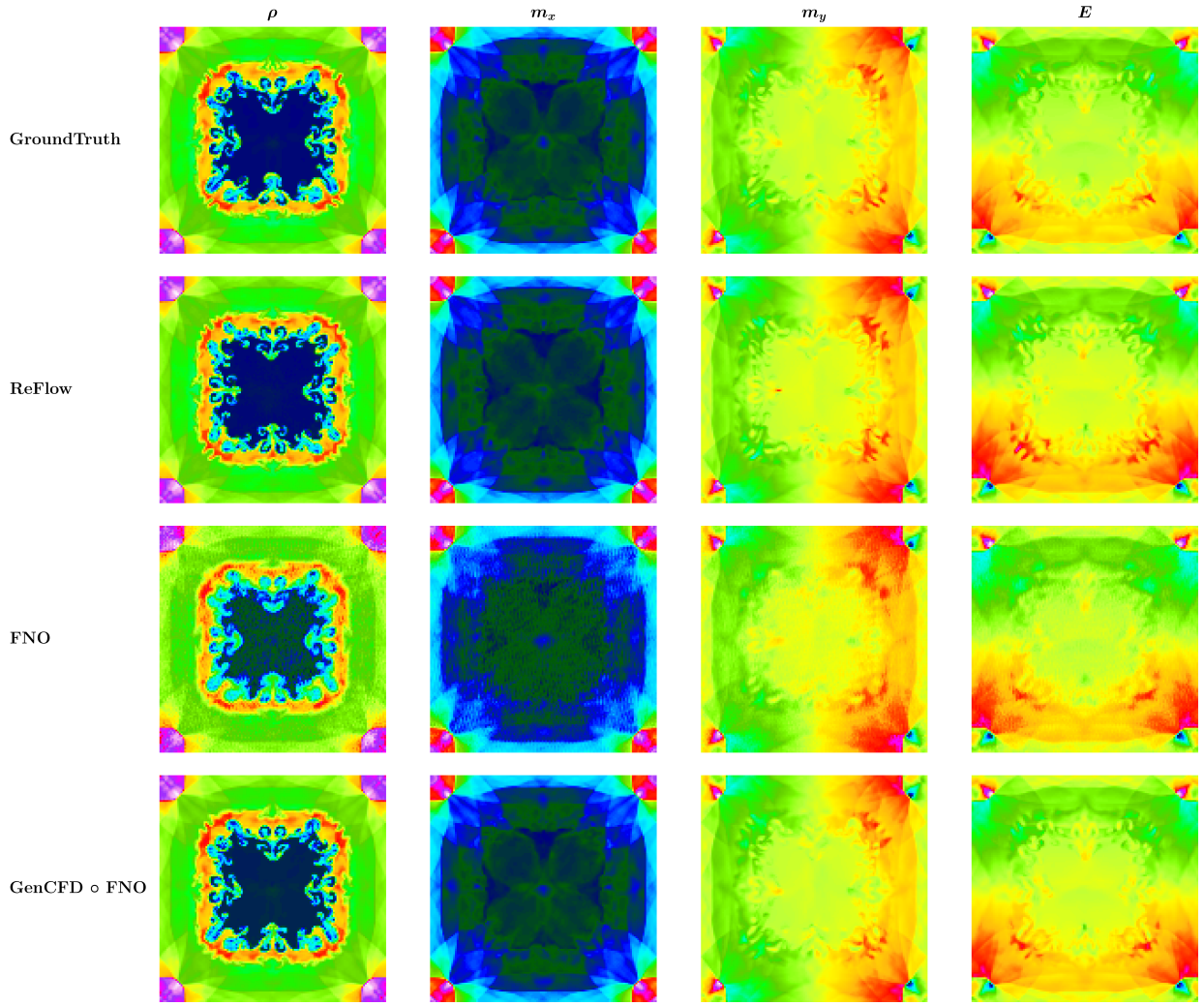
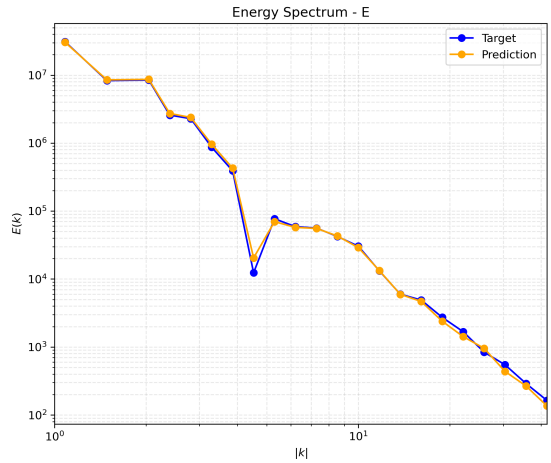
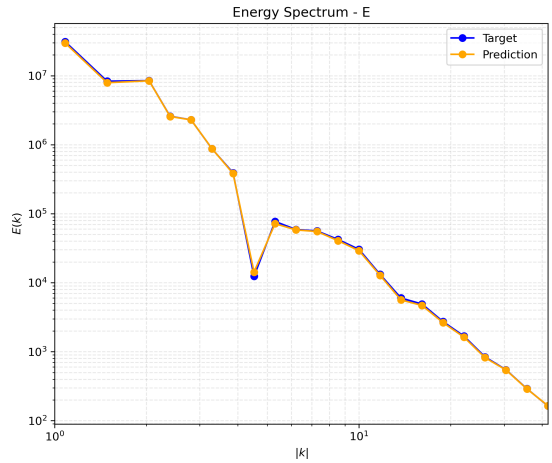


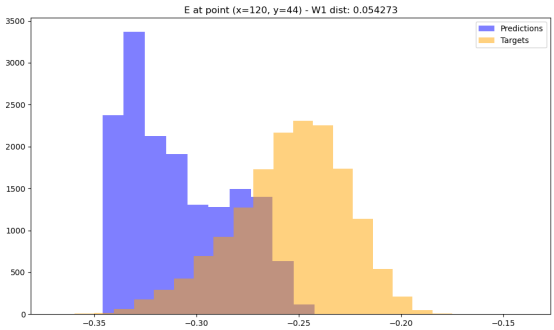
Figure 21: **Richtmyer–Meshkov Individual Sample Comparison**: A representative random test sample from the Richtmyer–Meshkov dataset, visualized across all four physical channels (ρ, m_x, m_y, p). All models produce physically plausible results, with only little variation in fidelity to fine-scale structures. ReFlow offers the closest visual match to the ground truth, preserving coherent details and sharp interfaces. Notably, the GenCFD ∘ FNO hybrid enhances spectral richness by reconstructing high-frequency features based on a low-frequency FNO prior, illustrating its capacity to recover sharper modes even when starting from a coarse prediction.



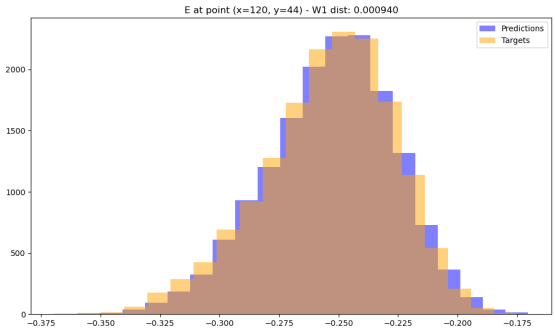
(a) UVit global $E(k)$.



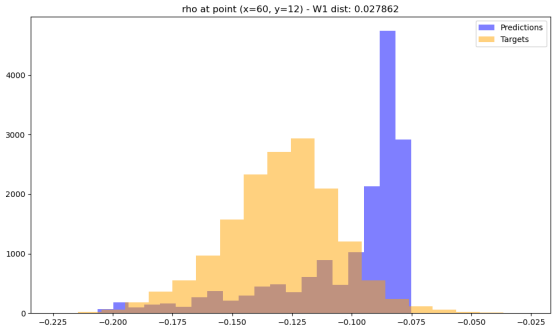
(b) RF global $E(k)$.



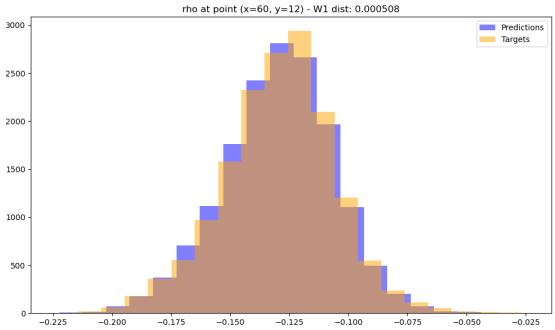
(c) UVit local at $(120, 44)$.



(d) RF local at $(120, 44)$.

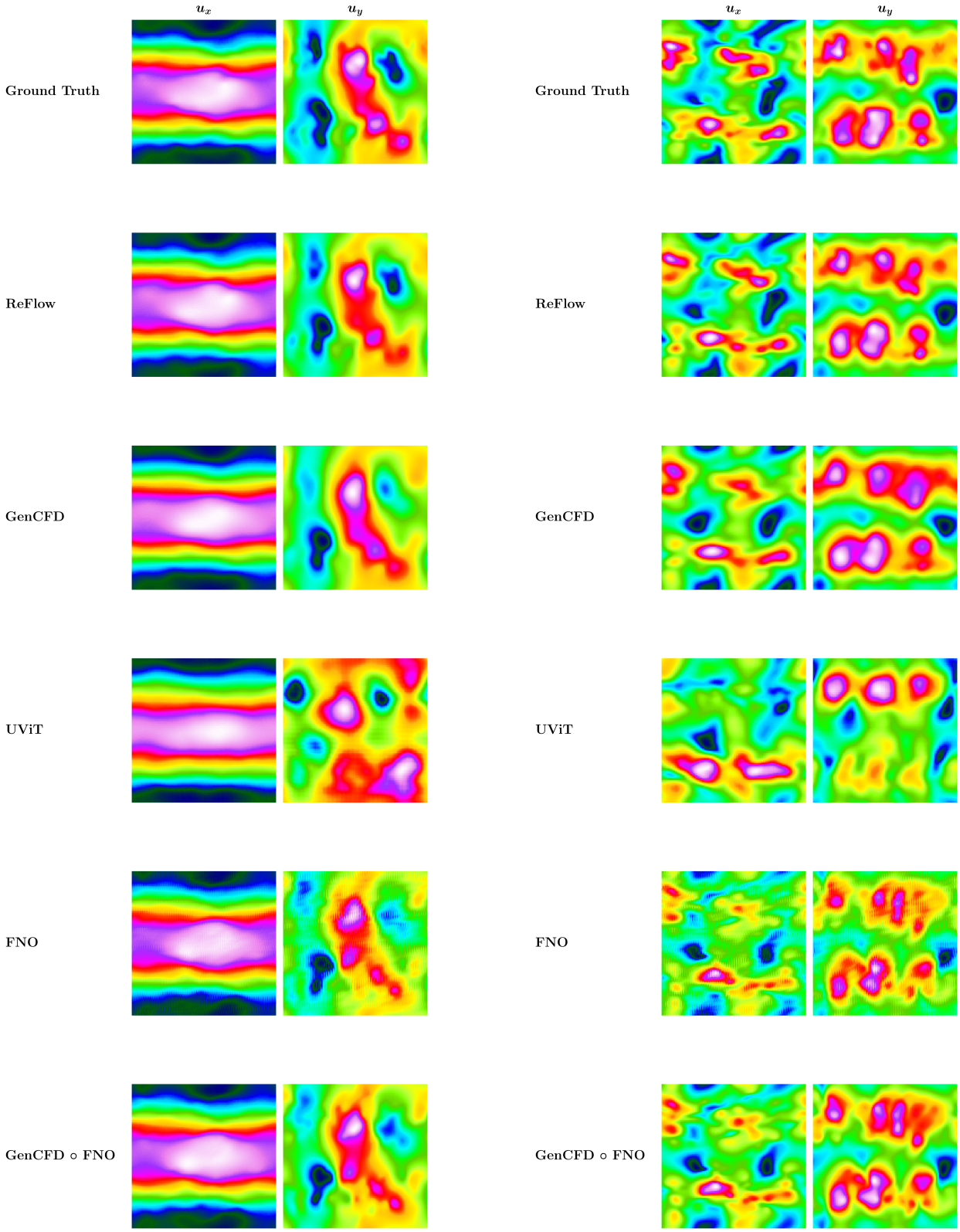


(e) UVit ρ at $(60, 12)$, $W_1 = 0.0278$.



(f) RF ρ at $(60, 12)$, $W_1 = 0.0005$.

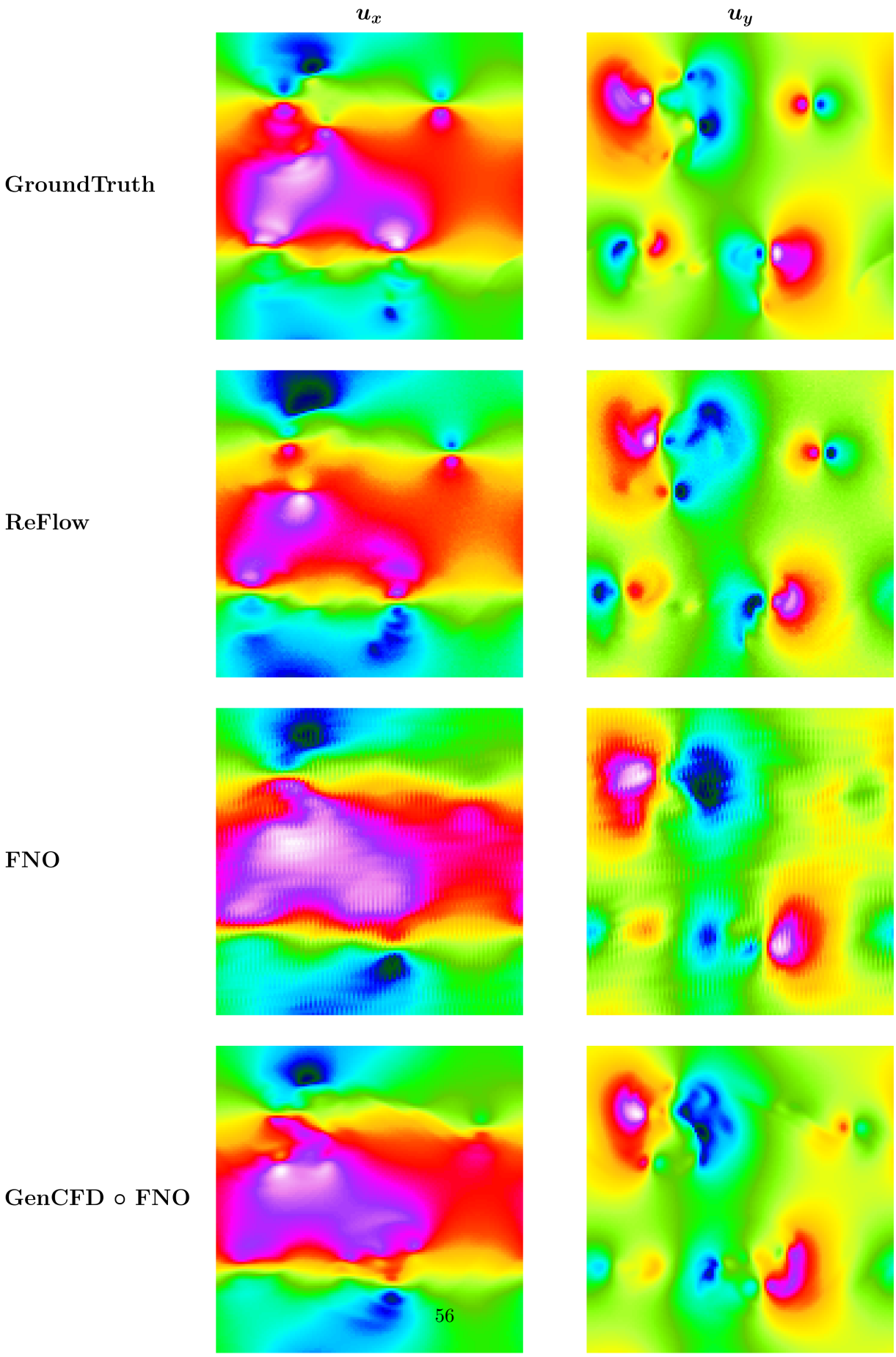
Figure 22: Richtmyer–Meshkov: spectral and pointwise distribution comparisons between UVit and Rectified Flow.

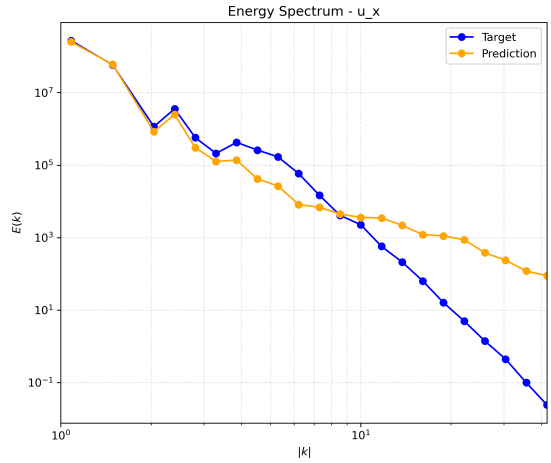


(a) **mean-field** μ

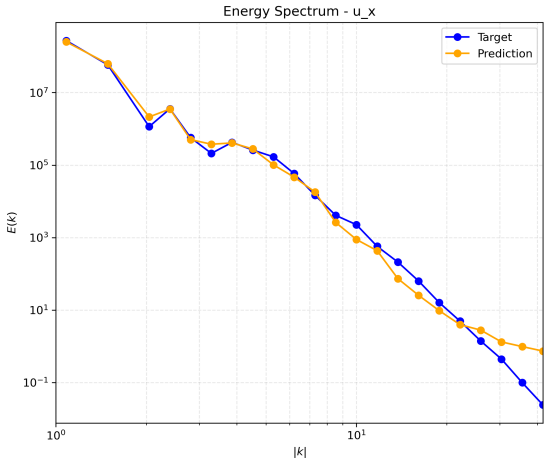
(b) **standard-deviation** σ

Figure 23: **Shear-Layer Mean and Standard Deviation Comparison**: Visualization of mean-field (μ) and standard-deviation (σ) predictions for the Shear-Layer dataset across both velocity components (u_x, u_y). For each model, the left half shows the predicted mean fields and the right half shows per-pixel standard deviations. The top row contains the reference ground truth. Despite the reduced channel complexity of this

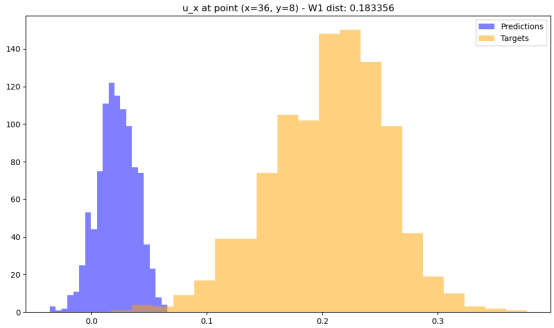

u_x
u_y
GroundTruth
ReFlow
FNO
GenCFD ∘ FNO



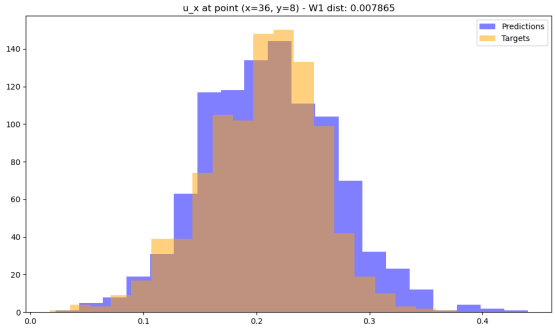
(a) FNO $E_{u_x}(r)$.



(b) RF $E_{u_x}(r)$.

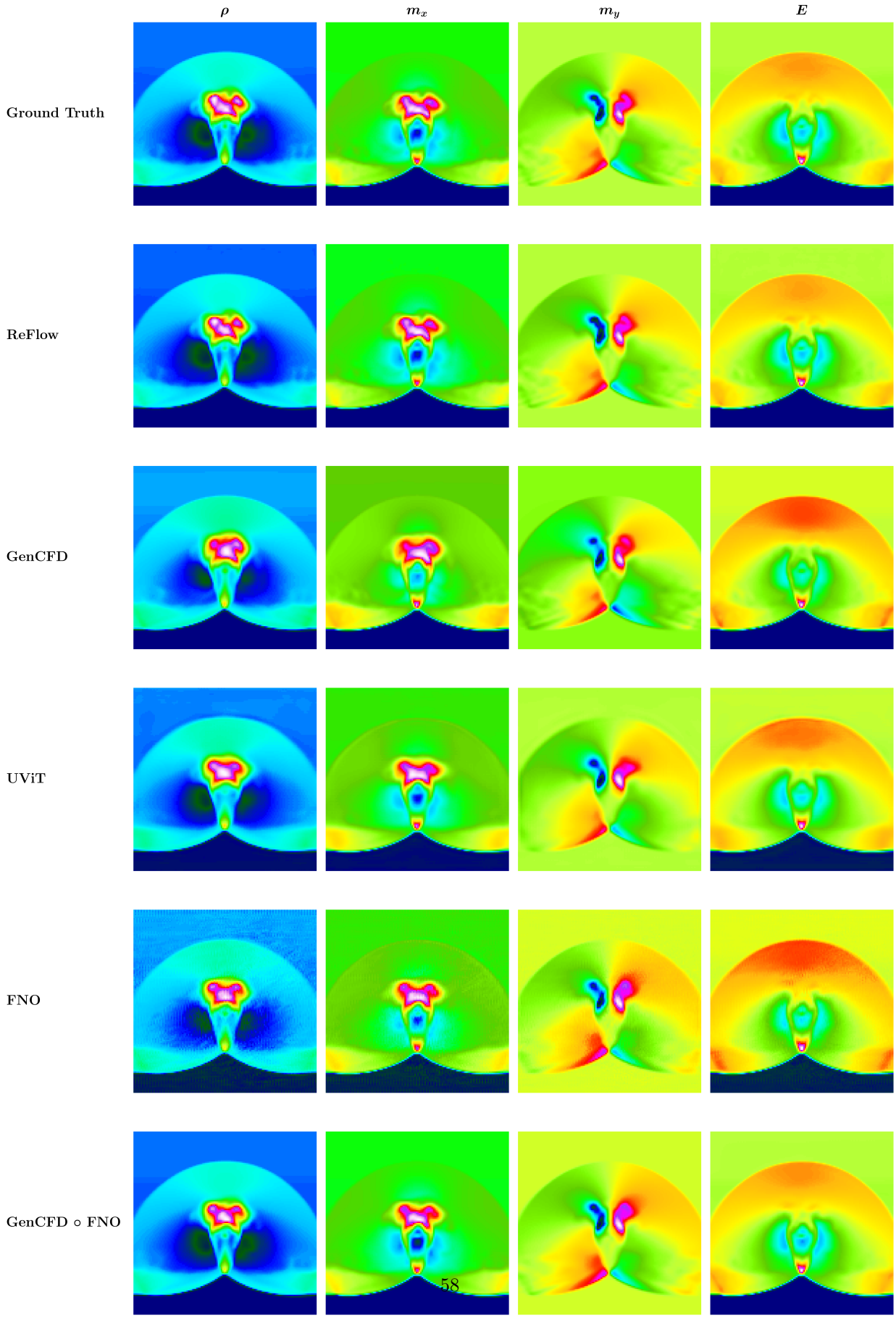


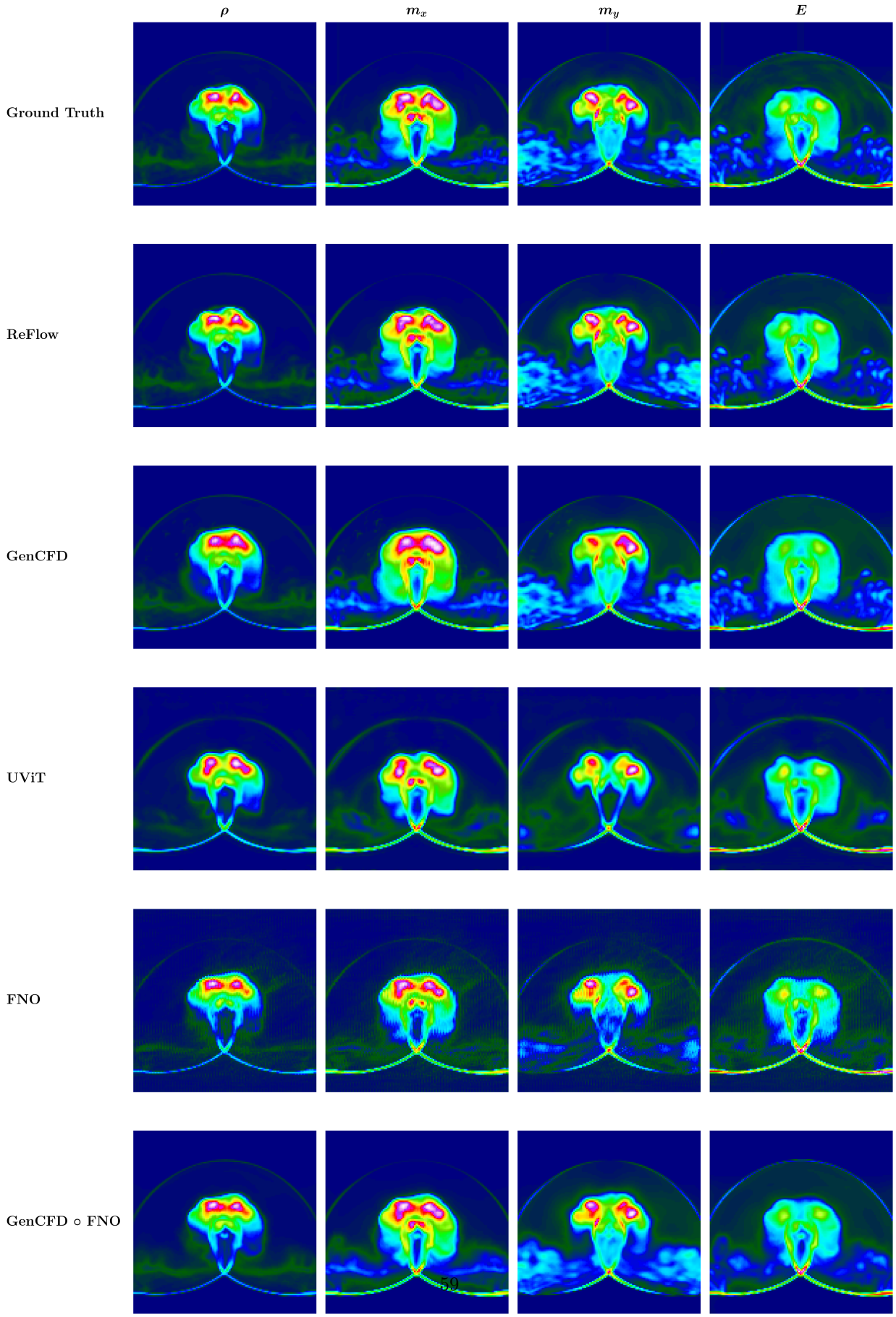
(c) FNO u_x at $(36, 8)$.



(d) RF u_x at $(36, 8)$.

Figure 25: Shear-layer: global u_x spectra and pointwise u_x distributions.


ρ
m_x
m_y
E
Ground Truth
ReFlow
GenCFD
UViT
FNO
GenCFD ∘ FNO


ρ
m_x
m_y
E
Ground Truth
ReFlow
GenCFD
UViT
FNO
GenCFD ∘ FNO

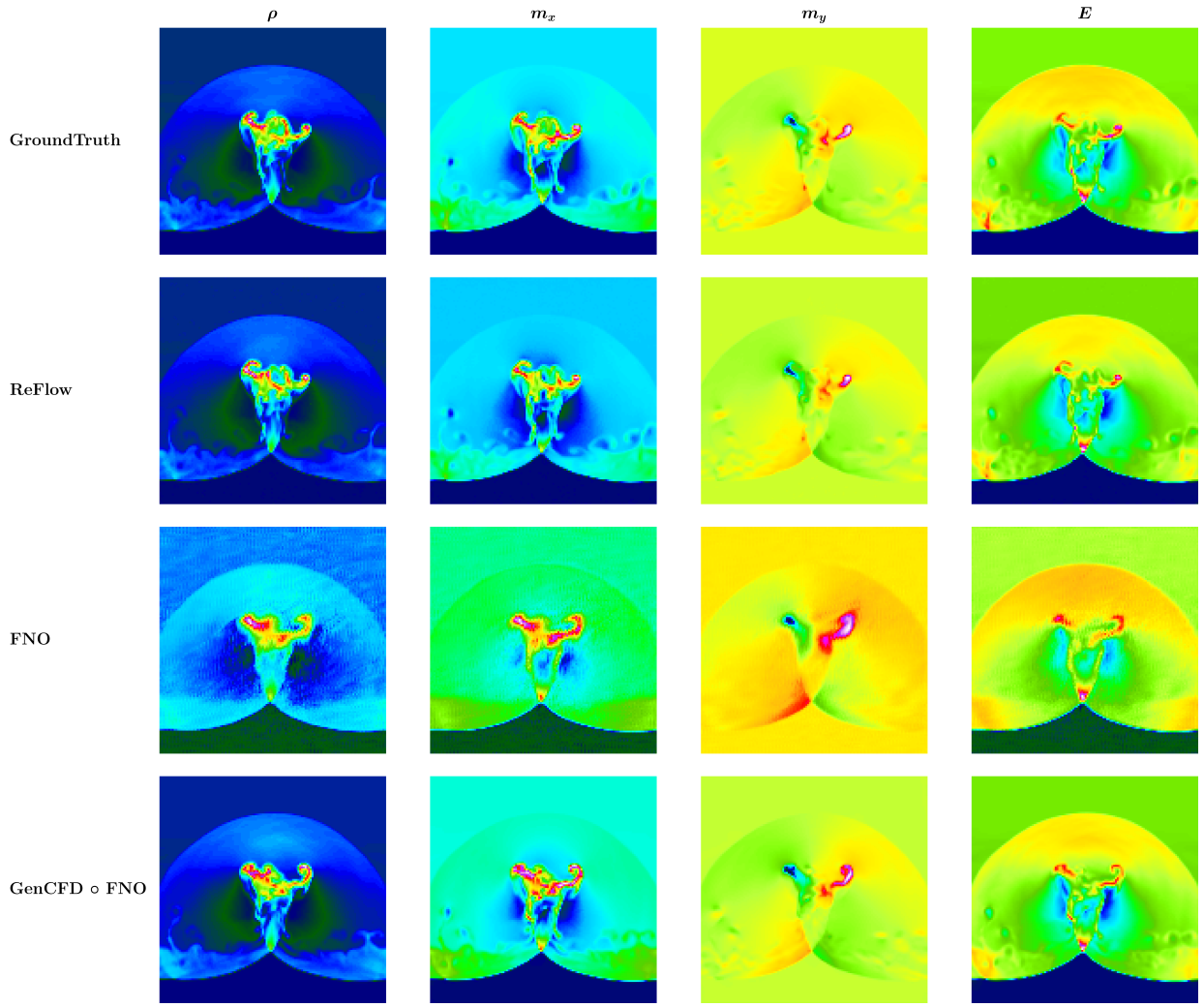
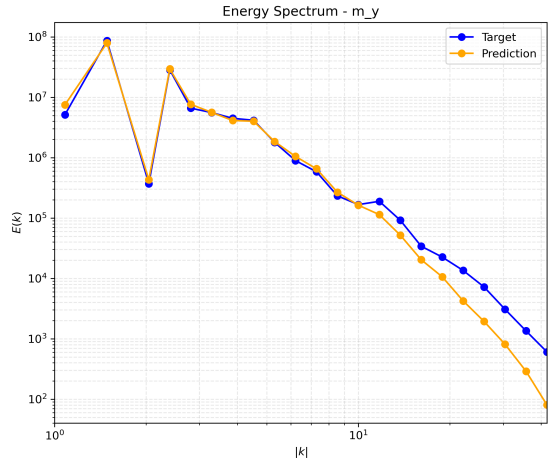
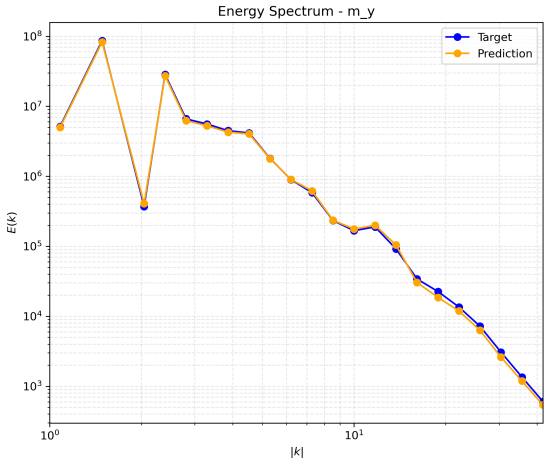


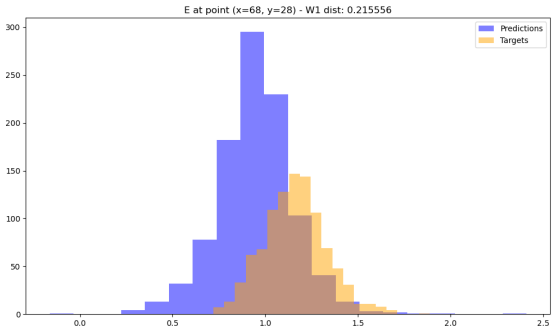
Figure 28: **Cloud–Shock Individual Sample Comparison**: A representative random test sample from the Cloud–Shock dataset, displayed across all physical channels. While all models produce plausible results, ReFlow and the hybrid GenCFD ∘ FNO most closely align with the ground truth, particularly in preserving fine-scale detail. The spectral reconstruction effect of the hybrid model is especially evident, recovering high-frequency structures from the coarse FNO prediction and enhancing the overall visual fidelity.



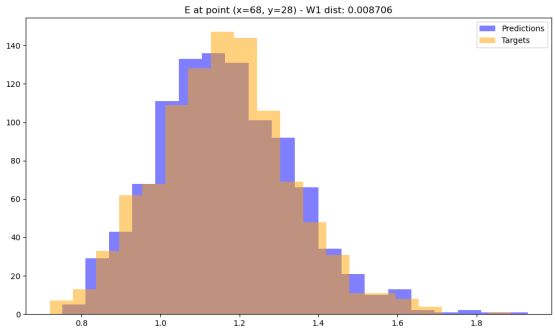
(a) UVit m_y spectrum.



(b) RF m_y spectrum.



(c) UVit E at $(68, 28)$.



(d) RF E at $(68, 28)$.

Figure 29: Cloud-shock: spectral and pointwise energy comparisons.