

Representing Flow Fields with Divergence-Free Kernels for Reconstruction

XINGYU NI, School of Computer Science, Peking University, China

JINGRUI XING, School of Intelligence Science and Technology, Peking University, China

XINGQIAO LI, School of Intelligence Science and Technology, Peking University, China

BIN WANG*, Independent, China

BAOQUAN CHEN*, State Key Laboratory of General Artificial Intelligence, Peking University, China

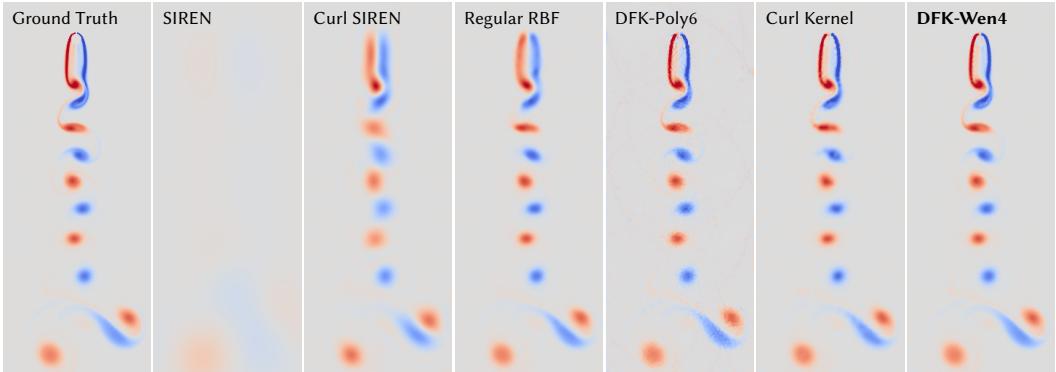


Fig. 1. Fitting experiments of the Kármán vortex street, with the resulting vorticity fields illustrated. We can optimize divergence-free kernels to compactly store the incompressible flow field data. The PSNR/SSIM values for each method are as follows: SIREN: 22.46/0.897; Curl SIREN: 28.17/0.936; Regular RBF: 30.90/0.967; DFK-Poly6: 32.80/0.824; Curl Kernel: 36.56/0.979; **DFK-Wen4: 38.78/0.990**.

Accurately reconstructing continuous flow fields from sparse or indirect measurements remains an open challenge, as existing techniques often suffer from oversmoothing artifacts, reliance on heterogeneous architectures, and the computational burden of enforcing physics-informed losses in implicit neural representations (INRs). In this paper, we introduce a novel flow field reconstruction framework based on divergence-free kernels (DFKs), which inherently enforce incompressibility while capturing fine structures without relying on hierarchical or heterogeneous representations. Through qualitative analysis and quantitative ablation studies, we identify the matrix-valued radial basis functions derived from Wendland's C^4 polynomial (DFKs-Wen4) as the optimal form of analytically divergence-free approximation for velocity fields, owing to their favorable numerical properties, including compact support, positive definiteness, and second-order differentiability.

*corresponding authors

Authors' addresses: Xingyu Ni, nixy@pku.edu.cn, School of Computer Science, Peking University, Beijing, China; Jingrui Xing, xjr01@hotmail.com, School of Intelligence Science and Technology, Peking University, Beijing, China; Xingqiao Li, lixingqiao@pku.edu.cn, School of Intelligence Science and Technology, Peking University, Beijing, China; Bin Wang, binwangbuaa@gmail.com, Independent, Beijing, China; Baoquan Chen, baoquan@pku.edu.cn, State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 2577-6193/2025/8-ART52

<https://doi.org/10.1145/3747872>

Experiments across various reconstruction tasks, spanning data compression, inpainting, super-resolution, and time-continuous flow inference, have demonstrated that DFKs-Wen4 outperform INRs and other divergence-free representations in both reconstruction accuracy and computational efficiency, while requiring the fewest trainable parameters.

CCS Concepts: • Computing methodologies → Point-based models; Physical simulation.

Additional Key Words and Phrases: incompressible flows, divergence-free kernels, implicit neural representations, fluid reconstruction

ACM Reference Format:

Xingyu Ni, Jingrui Xing, Xingqiao Li, Bin Wang, and Baoquan Chen. 2025. Representing Flow Fields with Divergence-Free Kernels for Reconstruction. *Proc. ACM Comput. Graph. Interact. Tech.* 8, 4, Article 52 (August 2025), 21 pages. <https://doi.org/10.1145/3747872>

1 INTRODUCTION

Reconstructing high-fidelity continuous flow fields from sparse, incomplete, or indirect data is crucial across various scientific and engineering domains, including meteorology, biomedicine, hydraulic engineering, automotive manufacturing, and visual effects. While traditional grid- and particle-based representations have been highly successful in forward simulations of fluid dynamics [Bridson 2015], their resolution-dependent nature and high-dimensional parameter spaces pose significant challenges for optimization, making them less effective for inverse reconstruction tasks that require optimizing a separate model for each flow field instance. This limitation has driven recent research toward implicit neural representations (INRs), which offer a resolution-agnostic framework for modeling continuous and differentiable physical fields.

However, neural network-based representations also face two major challenges. First, despite advancements such as positional encoding [Mildenhall et al. 2021] and periodic activation functions [Sitzmann et al. 2020], INRs tend to oversmooth physical fields, leading to the loss of fine-scale details. In the context of flow field inference from multi-view RGB videos, the state-of-the-art approach [Yu et al. 2023] leverages multi-resolution hash encoding [Müller et al. 2022] to enhance network expressiveness and introduce vortex particles to recover missing fluid structures. This highly heterogeneous representation compromises robustness and increases the complexity of both implementation and further improvements. Second, physics-informed neural networks (PINNs) [Raissi et al. 2019, 2020], which incorporate physical constraints as loss terms, suffer from optimization difficulties due to their heavy reliance on balancing multiple partial differential equation (PDE) loss terms. Striking an appropriate trade-off between fitting observed data and enforcing physical laws is nontrivial [Chu et al. 2022; Gao et al. 2021; Wang et al. 2020, 2024], often resulting in suboptimal or even infeasible solutions. Moreover, the need to compute high-order derivatives within the neural network significantly increases computational costs, as it leads to prohibitively large computational graphs, limiting the applicability of these methods in time-sensitive scenarios.

In this paper, we propose a novel scheme for reconstructing flow fields based on divergence-free kernels (DFKs), which obviates the need for heterogeneous or hierarchical representations and eliminates incompressibility-related penalty terms in the optimization process. Although divergence-free kernels have been studied in numerical analysis and interpolation, their application to large-scale flow field reconstruction—especially in an optimization-based setting—has not been thoroughly explored. At the core of this framework lies a set of matrix-valued radial basis functions (RBFs) [Narcowich and Ward 1994], derived by applying differential operators to Wendland's C^4 polynomial [Wendland 1995]. These kernels, which we refer to as DFKs-Wen4 for brevity, serve as the numerical basis for our flow field representation. The function space spanned by DFKs-Wen4 inherently satisfies the continuity equation for incompressible flows, rigorously enforcing

the divergence-free property of velocity fields by construction. Additionally, DFKs-Wen4 possess critical properties such as compact support, positive definiteness, and second-order differentiability, while exhibiting strong alignment with fundamental solutions for fluid flows around obstacles. Moreover, unlike INRs, our kernel-based representation offers greater modeling flexibility, as the position, radius, and weight of each kernel can be explicitly optimized to better resolve multiscale and complex fluid structures. Comparative evaluations against alternative kernel-based methods and implicit neural representations (INRs) reveal that DFKs-Wen4 achieve superior performance across diverse objective functions and varying levels of data sparsity, all while maintaining a minimal number of trainable parameters. These findings underscore the potential of DFKs-Wen4 as a compelling alternative to neural network-based approaches for fluid reconstruction tasks, offering both theoretical rigor and computational efficiency.

Our experiments focus on the following flow field reconstruction tasks: (1) fitting dense velocity data to reduce memory consumption for flow field storage; (2) fitting dense but non-divergence-free data to perform pressure projection via the Helmholtz decomposition inherent in the representation; (3) fitting dense velocity data with missing regions, leveraging the representation’s generalization ability for inpainting; (4) fitting sparse velocity data, utilizing the representation’s generalization capability for super-resolution; and (5) inferring a time-continuous velocity field from dynamic, dense passive scalar data based on the advection equation. It is important to note that our goal is not to achieve state-of-the-art performance in any specific application but rather to systematically compare different representations in terms of their expressiveness and optimization efficiency under controlled conditions. To isolate the impact of representation quality on reconstruction results, we avoid introducing confounding factors from auxiliary techniques in each case for both our method and the comparison methods. Instead, we optimize the parameters of the representations to fit the observed data with the continuity equation of fluid dynamics (i.e., the incompressibility condition) as the primary physical prior, which is either enforced as a soft constraint or directly embedded in the search space.

The technical contributions are summarized as follows:

- A systematic and practical framework for applying DFKs-Wen4 to flow field reconstruction,
- Qualitative analysis and quantitative ablation studies to identify the optimal form of the divergence-free kernels for fluid reconstruction tasks, and
- Comprehensive benchmarking against state-of-the-art INRs across diverse application scenarios.

2 RELATED WORK

Kernel-based modeling. The use of kernel functions to represent flow fields in forward fluid simulation has a long-standing history. Compared to traditional approaches such as finite element methods (FEMs) and finite difference methods (FDMs), kernel-based modeling offers simpler formulations and easier implementation, making it particularly well-suited for simulating fluids—materials characterized by highly dynamic topologies and intricate local details. Specifically, kernel functions have been employed to smooth the properties of neighboring particles (e.g., velocity, density, and pressure) [Bender and Koschier 2015; Müller et al. 2003; Yu and Turk 2013], interpolate fluid quantities (such as velocity and density) onto grids [Canabal et al. 2016; Chang et al. 2022; Foster and Fedkiw 2001], and facilitate smooth interactions between particles and grid-based solvers [Hu et al. 2018; Jiang et al. 2015; Zhu and Bridson 2005]. In these studies, kernels are typically designed to satisfy desirable mathematical properties, such as smoothness, compact support, and positive definiteness.

In recent years, kernel-based representations have also gained significant attention in the context of physical field reconstruction. Particularly, for the reconstruction of 3D radiance fields, kernel-based methods—such as those used in Point-NeRF [Xu et al. 2022] and 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023]—have demonstrated advantages over vanilla Neural Radiance Fields (NeRFs) [Mildenhall et al. 2021], offering improved optimization efficiency and faster inference speeds. Inspired by these advancements, we explore the integration of kernel-based modeling into the reconstruction of flow fields, with the aim of achieving similar superior flexibility and performance compared to neural network-based methods (i.e., INRs).

Divergence-free representations. As the continuity condition for incompressible flows, the divergence-free constraint on velocity fields plays a crucial role in their modeling. In the field of computer graphics, recent works have pioneered fluid simulations that directly utilize divergence-free representations rather than relying on pressure projection or other post-processing techniques. Some methods achieve this by maintaining a vector potential on grids and computing its curl to obtain a divergence-free flow field [Chang et al. 2022; Lyu et al. 2024], while others enforce divergence-free interpolation schemes directly on grids [Nabizadeh et al. 2024; Roy-Chowdhury et al. 2024; Schroeder et al. 2022]. In the context of data-driven methods, Kim et al. [2019] encoded fluid simulation results—expressed using aforementioned velocity potentials—into convolutional neural networks (CNNs), while Richter-Powell et al. [2024] proposed neural network parameterizations that inherently satisfy the divergence-free condition in arbitrary dimensions, using flow field modeling as a key application scenario.

The divergence-free kernels discussed in this paper originate from research on matrix-valued RBF interpolation [Narcowich and Ward 1994], with Lowitzsch [2005] presenting a formulation closest to our adopted DFKs-Wen4. Over the years, various DFKs have been employed to represent flow fields [Fuselier et al. 2016; Li et al. 2020; Skrinjar et al. 2009; Wendland 2009], electromagnetic fields [McNally 2011], and elastic potential fields [Chan-Lock et al. 2022]. The idea of using DFKs for flow field reconstruction is also evident in the works of Macêdo and Castro [2010] and Zhou et al. [2019], where support vector regression and RBF interpolation were respectively used for flow field fitting. However, the integration of DFK representations with advanced optimizers from the deep learning field (e.g., Adam [Kingma and Ba 2017], see §5) and cutting-edge visual reconstruction techniques (e.g., NeRF [Mildenhall et al. 2021], see §6.3.2) for large-scale, even passive field-based, flow field reconstruction remains largely unexplored.

Flow field reconstruction. Here we briefly review previous optimization-based methods for flow field reconstruction tasks discussed in this paper.

For data compression, Vogler and Rist [2017] demonstrated the viability of wavelet-based compression schemes in handling large-scale numerical CFD datasets. Later, Liu et al. [2019] introduced deep learning into the compression pipeline by proposing an in-situ GAN-based framework. Concurrently, Wu et al. [2020] leveraged spatiotemporal subsampling and local resimulation to construct accurate turbulent flow databases. Taking a different approach, Anantharaman et al. [2023] examined the applicability of conventional multimedia compression standards (e.g., JPEG, JP2, H.264, H.265, and AV1) to flow field data. More recently, neural representations have gained significant attention: Kim et al. [2024] presented NeuralVDB, a hierarchical neural data structure designed for compressing sparse volumetric flow data with high resolution, while Deng et al. [2023b] proposed Spatially Sparse Neural Fields (SSNFs), a hybrid INR designed for compact and continuous storage of spatiotemporal velocity fields during simulations.

For divergence-free projection, Chen et al. [2023] introduced a neural solution to transient differential equations using implicit neural spatial representation (INSR) [Xie et al. 2022], though it faces efficiency challenges due to high-order derivatives.

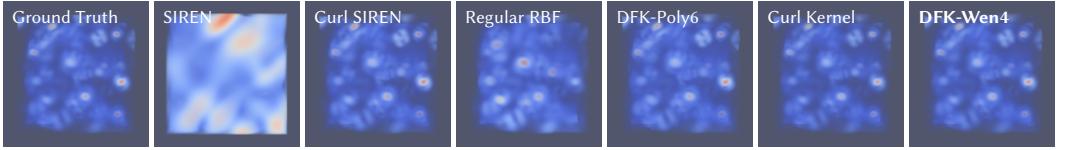


Fig. 2. Fitting experiments of the analytic vortices, with the resulting vorticity fields rendered. Physics-embedded methods clearly demonstrate superior fitting capabilities over physics-informed ones. Additionally, kernel-based approaches excel at capturing local details compared to neural network-based representations. Among all tested approaches, DFK-Wen4 achieves the lowest fitting error, as shown in Tab. 1.

For super-resolution (SR), Fukami et al. [2019] applied data-driven techniques to reconstruct low-resolution flow images for a 2D cylinder wake, and similar methods using CNNs and GANs have enhanced turbulence, plumes, and channel flows [Deng et al. 2019; Liu et al. 2020; Werhahn et al. 2019; Xie et al. 2018]. Since 2020, PINNs [Raissi et al. 2020] have been integrated to incorporate underlying physical laws, improving reliability and reducing reliance on high-resolution (HR) data. For example, Wang et al. [2020] utilized a physics-informed SR technique to reconstruct HR images in an advection-diffusion model of atmospheric pollution plumes, while Gao et al. [2021] developed a physics-constrained CNN for SR of vascular flow without labeled data.

For dynamic flow field inference, traditional approaches rely on specialized hardware [Atcheson et al. 2008; Ji et al. 2013] or particle imaging velocimetry (PIV) [Xiong et al. 2017], which tracks passive markers in the flow. Tomographic methods have also been explored [Gregson et al. 2014]. More recently, RGB-video-based techniques have reduced dependence on specialized setups. ScalarFlow [Eckert et al. 2019] introduced long-term temporal physics constraints by optimizing residuals between reconstructed and simulated density and velocity, while Franz et al. [2021] used differentiable rendering for end-to-end optimization. Deng et al. [2023a] proposed vortex particles to predict 2D fluid motion in videos. Emerging methods combining PINNs and NeRFs, such as PINF [Chu et al. 2022], HyFluid [Yu et al. 2023], and PICT [Wang et al. 2024], enforce physics constraints via soft regularization.

Furthermore, optimization-based methods for incompressible flow editing and inpainting formulate interpolation as an energy minimization problem subject to incompressibility constraints [Bhattacharya et al. 2012; Nielsen and Bridson 2011; Ozdemir et al. 2024; Sato et al. 2018]. Building on this framework, Schweri et al. [2021] utilized a physics-aware neural network to inpaint missing flow data from satellite observations.

3 PROBLEM STATEMENT

From a continuous perspective, the flow field reconstruction tasks we consider can be unified into the following constrained optimization problem:

$$\arg \min_{\mathbf{u}(\mathbf{x}, t) \in \mathcal{F}} \mathcal{L}_{\text{obs}} [\mathbf{u}(\mathbf{x}, t), f_{\text{obs}}(\mathbf{x}, t)], \quad (1)$$

$$\text{subject to } \nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, \quad (2)$$

$$\text{and } \mathbf{u} = \mathbf{u}_s, \quad \mathbf{x} \in \partial\Omega, \quad (3)$$

where Ω is the fluid domain, and $\partial\Omega$ is its boundary. The search space $\mathcal{F} = L^2(\Omega \times [0, T], \mathbb{R}^d)$ consists of d -dimensional square-integrable vector functions defined over the spatiotemporal domain.

The objective function \mathcal{L}_{obs} , which quantifies the observational loss, depends on both the flow field \mathbf{u} , to be optimized, and an observed input field f_{obs} . When the observations directly correspond



Fig. 3. Fitting experiments of the simple plume, with the resulting vorticity fields rendered. SIREN, Curl SIREN, and Regular RBF produce overly smooth results, failing to capture finer details. In contrast, Curl Kernel and DFK-Wen4 closely match the ground truth (see Tab. 1), delivering accurate and high-fidelity representations, with only 4.3% of DoFs used compared to the raw data.

to the flow field, though possibly incomplete within Ω , the objective function is given by

$$\mathcal{L}_{\text{obs}}[\mathbf{u}, \mathbf{u}_D] = \frac{1}{V_D} \int_0^T \int_{\Omega_D} \|\mathbf{u} - \mathbf{u}_D\| dV_D dt, \quad (4)$$

where Ω_D is the supervised region with volume V_D . Alternatively, if the observations originate from a passive field, such as soot concentration σ , the advection equation $\partial\sigma/\partial t + \mathbf{u} \cdot \nabla \sigma = 0$ must be incorporated into the objective function, yielding

$$\mathcal{L}_{\text{obs}}[\mathbf{u}, \sigma] = \frac{1}{V} \int_0^T \int_{\Omega} \left\| \frac{\partial \sigma}{\partial t} + \mathbf{u} \cdot \nabla \sigma \right\| dV dt. \quad (5)$$

The constraints given in Eqs. (2) and (3) arise from the continuity equation and the no-slip boundary condition for incompressible flows, respectively, where the solid velocity \mathbf{u}_s is assumed to be zero unless specified otherwise.

Physics-informed losses. Conventionally, the divergence-free and boundary conditions are *relaxed* by incorporating additional penalty terms \mathcal{L}_{div} and \mathcal{L}_{bou} , defined as

$$\mathcal{L}_{\text{div}} = \frac{1}{V} \int_0^T \int_{\Omega} \|\nabla \cdot \mathbf{u}\| dV dt, \quad (6)$$

$$\mathcal{L}_{\text{bou}} = \frac{1}{A} \int_0^T \int_{\partial\Omega} \|\mathbf{u} - \mathbf{u}_s\| dA dt, \quad (7)$$

where A denotes the boundary area of Ω . The resulting optimization problem is then formulated as

$$\arg \min_{\mathbf{u}(\mathbf{x}, t) \in \mathcal{F}} \mathcal{L} = \mathcal{L}_{\text{obs}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}} + \lambda_{\text{bou}} \mathcal{L}_{\text{bou}}, \quad (8)$$

where λ_{div} and λ_{bou} represent tunable weighting factors. The total objective function \mathcal{L} aligns with the loss formulation used in previous work [Chu et al. 2022; Wang et al. 2024; Yu et al. 2023].

Physics-embedded approaches. In contrast, we demonstrate that the divergence-free condition, $\nabla \cdot \mathbf{u} = 0$ can be more effectively enforced by embedding it directly into the search space. By employing divergence-free representations, the search space is restricted to valid solutions that inherently satisfy the continuity equation for incompressible flows, ensuring that the value of \mathcal{L}_{div} remains identically zero. This reformulates the optimization problem as

$$\arg \min_{\mathbf{u}(\mathbf{x}, t) \in \mathcal{P}[\mathcal{F}]} \mathcal{L} = \mathcal{L}_{\text{obs}} + \lambda_{\text{bou}} \mathcal{L}_{\text{bou}}, \quad (9)$$

in which $\mathcal{P}[\mathcal{F}]$ denotes the reduced search space of divergence-free fields. This embedding approach guarantees that the reconstructed flow fields are both data-driven and physically realistic, leading to more accurate results.

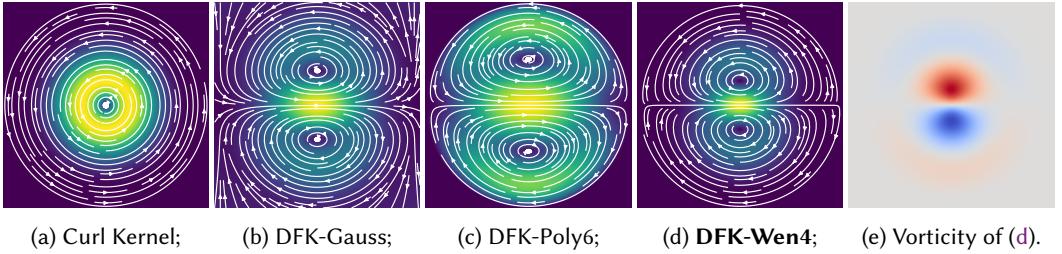


Fig. 4. Illustration of different divergence-free kernels in 2D. The left four images display streamline plots, where brighter background colors indicate higher velocity. For Curl Kernel (a), the vector field is constructed as $(\partial/\partial y, -\partial/\partial x) R_{\text{Wen4}}$, which generalizes to $\nabla \times R_{\text{Wen4}} \omega$ in 3D. For DFKs (b-d), the fields are formulated as $(-\mathbf{I} \nabla \cdot \nabla + \nabla \nabla^\top) \phi \omega$, where ϕ takes the forms $R_{\text{Gau}} = \exp(-9r^2/2)$, $R_{\text{Poly6}} = (1 - r^2)_+^3$, and R_{Wen4} , respectively. For comparison, ω is set to $(1, 0)$. Note that these plots also serve as 2D cross-sections of their 3D counterparts. The rightmost image (e) shows the corresponding vorticity field of DFK-Wen4, with cool and warm colors indicating opposite rotation directions.

Loss discretization. In practice, observations are typically provided as discrete sample points. When these data points are uniformly distributed within Ω_D (or $\partial\Omega_D$), the spatial integrals divided by V (or A) in Eqs. (4–7) can generally be approximated by averaging over the data points. Similarly, the outer temporal integral can be expressed as a summation over discrete time steps, and the time derivatives of physical quantities are computed using finite difference schemes.

4 DIVERGENCE-FREE KERNELS

We present the essential formulae for constructing the physics-consistent search space using divergence-free matrix-valued kernels, specifically DFKs-Wen4, along with a theoretical analysis of their advantages over other kernel-based representations.

4.1 Construction of DFKs-Wen4

Given the spatial dimension d , we consider the i -th kernel, located at $\mathbf{x}_i \in \mathbb{R}^d$ with a size of $h_i \in \mathbb{R}$, to be associated with a scalar-valued kernel defined as

$$\phi_i(\mathbf{x}) = \phi\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{h_i}\right), \quad (10)$$

where $\phi(r)$ is a radial basis function. From this scalar kernel $\phi_i(\mathbf{x})$, we derive a matrix-valued kernel $\psi_i : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ by applying a second-order differential operator [Narcowich and Ward 1994]:

$$\psi_i(\mathbf{x}) = (-\mathbf{I} \nabla \cdot \nabla + \nabla \nabla^\top) \phi_i(\mathbf{x}), \quad (11)$$

in which \mathbf{I} is the identity matrix and $\nabla \nabla^\top$ denotes the Hessian operator. For any choice of $\phi(r)$ and any weight vector $\omega_i \in \mathbb{R}^d$, it can be shown (see §A.2 in the supplementary document) that

$$\psi_i(\mathbf{x}) \omega_i = \nabla \times (\nabla \phi_i \times \omega_i). \quad (12)$$

Therefore, if we express a flow field $\mathbf{u}(\mathbf{x})$ as the summation

$$\mathbf{u}(\mathbf{x}) = \sum_i \psi_i(\mathbf{x}) \omega_i, \quad (13)$$

the divergence of the field, $\nabla \cdot \mathbf{u}$, is guaranteed to be zero.

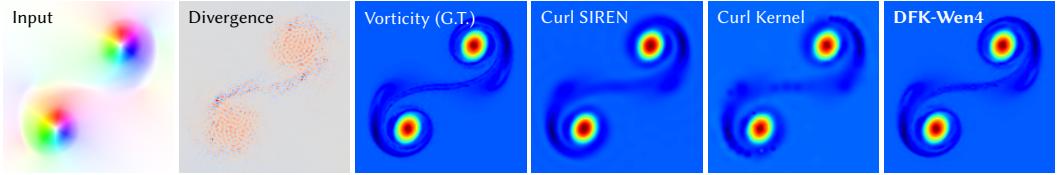


Fig. 5. Projection experiments of the Taylor vortex. The input is illustrated using HSV color encoding, with its divergence plotted. The vorticity fields of ground truth and experimental results are presented with *jet* color mapping, where our method provides the most accurate projection. The PSNR/SSIM values are as follows: Curl SIREN: 33.68/0.965; Curl Kernel: 29.80/0.970; **DFK-Wen4: 39.30/0.994**.

Considering the trade-off between the flow field smoothness and computational complexity, we adopt the C^4 -continuous piecewise-polynomial radial function proposed by Wendland [1995],

$$R_{\text{Wen4}}(r) = (1 - r)_+^6 (35r^2 + 18r + 3), \quad (14)$$

to serve as $\phi(r)$, where $(\cdot)_+$ denotes $\max(\cdot, 0)$. The concrete formulations of DFKs-Wen4 and their derivatives that are useful for both fluid mechanics and optimization algorithms are provided in §B.

4.2 Properties Analysis

Although the procedure outlined in §4.1 is not the only way to obtain divergence-free kernel-based representations, we argue that DFKs-Wen4 are particularly well-suited for flow field reconstruction due to their desirable properties.

4.2.1 Dipolarity. Unlike vortex flows derived from the curl of a radial vector potential (or the stream function in 2D), as shown in Fig. 4a, the kernel constructed via Eq. (12) exhibits distinct dipole characteristics, with two vortices rotating in opposite directions, as illustrated in Figs. 4b–4d. This dipolar structure is a fundamental phenomenon that arises when a flow interacts with a blocking body (e.g., around a cylinder or in the formation of a Kármán vortex street). Moreover, for any divergence-free vector field, the dipole, rather than a single vortex or monopole, constitutes the leading term in its multipole expansion [Agre 2011]. Accordingly, we argue that dipoles offer a more compact and accurate representation of incompressible flow fields, analogous to their role in modeling magnetic fields [Griffiths 2017]¹, another canonical example of divergence-free vector fields.

4.2.2 Compact Support. DFKs-Wen4 inherit the compact support property of Wendland functions, enabling localized optimizations that avoid global interdependence. This is ideal for representing regions with zero velocity, such as solid boundaries or stationary fluids. In contrast, Gaussian functions, often used in classification tasks, are less suited for flow field reconstruction. As shown in Fig. 4b, divergence-free kernels based on Gaussian functions (DFKs-Gauss) influence the entire space, increasing computational complexity and hindering local details capture. Truncating the Gaussian function at a certain radius disrupts continuity, making it unsuitable for this application.

4.2.3 Positive Definiteness. It is a key concept in kernel-based interpolation [Wendland 2004], ensuring that the interpolation matrix derived from distinct data points is invertible and the interpolation problem has a unique solution. For kernels of the form $\psi = (-I\nabla \cdot \nabla + \nabla\nabla^\top)\phi$, it can be proven that if ϕ is positive definite, ψ will also be positive definite (see §A.1 in the supplementary document). In flow field reconstruction, positive definiteness implies that when the number of kernels equals the number of data points, there exists a set of weights such that the reconstructed

¹Indeed, streamlines in Figs. 4b–4d closely resemble the magnetic field lines near a magnetic dipole moment.

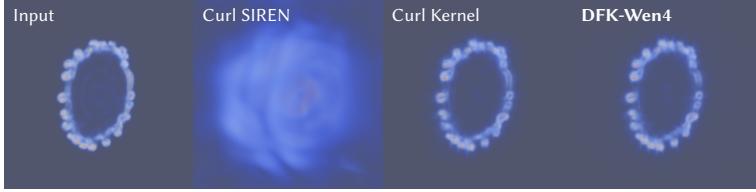


Fig. 6. Projection experiments of the vortex ring collision. The vortice fields of input and results are rendered. As demonstrated in Tab. 1, DFK-Wen4 has the lowest loss in this case, though Curl Kernel also provides comparable visualization. Note that the vorticity field output by the INR is very diffuse.

flow field interpolates all given data points exactly. Even when fewer kernels are used, this property empirically contributes to a smoother and more convex optimization landscape [Zeng et al. 2024]. In contrast, non-positive definite kernels, such as DFKs based on the Poly6 function commonly used in SPH methods [Müller et al. 2003] (DFKs-Poly6), tend to produce flow fields with steeper gradients near local extrema (see Fig. 4c).

4.2.4 Differentiability. By utilizing C^4 -continuous Wendland radial functions, DFKs-Wen4 ensure the existence and continuity of second-order derivatives, which are essential for upstream and downstream tasks. The Navier–Stokes equations that govern fluid motion involve second-order partial derivatives of the flow field for the computation of viscosity, and the gradient of vorticity (the curl of the flow field, see Fig. 4e) forms the foundation of vortex methods [Bridson 2015]. As a C^2 -continuous representation, DFKs-Wen4 guarantee differentiability for most reconstruction tasks based on physical flow field equations.

5 IMPLEMENTATION

We implemented the flow field reconstruction framework based on divergence-free kernels using the PyTorch library. To maintain a balanced distribution of kernels, we developed a custom C++ module that employs fast Poisson disk sampling [Bridson 2007] to determine the initial kernel positions. Let N denote the number of sampled kernels. To encourage that every point within the region of interest is influenced by multiple kernels, we initialize the kernel radii uniformly using the following formula:

$$h_i = \eta \left[\frac{\Gamma\left(1 + \frac{d}{2}\right) V}{N \pi^{\frac{d}{2}}} \right]^{\frac{1}{d}} \quad (15)$$

where $\Gamma(\cdot)$ is the gamma function, and η is a user-defined scaling factor. Additionally, the initial kernel weights are set to zero.

Similar to 3DGS [Kerbl et al. 2023], in most application scenarios, the position \mathbf{x}_i , radius h_i , and weight ω_i of each DFK are treated as trainable parameters, optimized using stochastic gradient descent with an Adam optimizer [Kingma and Ba 2017] and an exponential learning rate scheduler. However, for large-scale scenarios, such as multi-frame flow field inference tasks, both time and memory costs become prohibitive. To mitigate this, we fix the positions and radii of the kernels and optimize only the weights. To further boost performance, we developed a custom C++ hash grid module to precompute and cache influenced data point–kernel pairs and extended PyTorch with Taichi [Hu et al. 2019] to enable parallel computation of each pair’s contribution. In this case, we continue using the Adam optimizer but switch to full-batch gradient descent, reducing the learning rate by 10% once a plateau is detected.

Table 1. Basic specifications and final loss values at convergence of experiments. For each case, the data scale is given in the number of data points, and the result with the minimal loss is colored green. The DFKs-Wen4 won in nearly all the experiments except for *plume* and *missile*. See discussions in the main text.

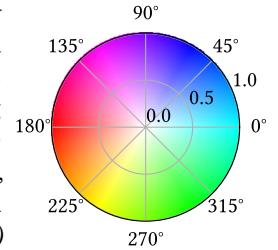
Section	Case	d	Data scale	SIREN	Curl SIREN	Regular RBF	DFK-Poly6	Curl Kernel	DFK-Wen4
6.1.1	Kármán	2	104.4 k	3.950×10^{-2}	5.289×10^{-3}	2.150×10^{-3}	2.758×10^{-3}	1.483×10^{-3}	5.421×10^{-4}
6.1.1	analytic	3	512.0 k	4.819×10^{-2}	7.580×10^{-3}	1.176×10^{-2}	1.395×10^{-2}	8.445×10^{-3}	7.561×10^{-3}
6.1.1	plume	3	2.097 M	3.783×10^{-3}	1.940×10^{-3}	2.515×10^{-3}	2.309×10^{-3}	1.517×10^{-3}	1.839×10^{-3}
6.1.2	Taylor	2	40.00 k		9.082×10^{-4}			1.357×10^{-3}	2.611×10^{-4}
6.1.2	collision	3	2.097 M		1.069×10^{-3}			4.982×10^{-4}	3.372×10^{-4}
6.2.1	flows (0°)	2	210.4 k		1.062×10^{-4}				7.124×10^{-5}
6.2.1	flows (45°)	2	210.4 k		1.654×10^{-4}				1.034×10^{-4}
6.2.1	flows (90°)	2	210.4 k		2.131×10^{-4}				1.264×10^{-4}
6.2.1	missile	2	1.012 M		2.608×10^{-4}		4.140×10^{-3}	1.134×10^{-3}	3.306×10^{-4}
6.2.1	bullet	3	5.504 M		1.174×10^{-3}			4.612×10^{-4}	3.954×10^{-4}
6.2.2	turb. A	2	4.096 k		1.327×10^{-2}		2.109×10^{-3}	2.494×10^{-3}	4.950×10^{-4}
6.2.2	turb. B	2	4.096 k		8.306×10^{-3}		1.908×10^{-3}	2.595×10^{-3}	4.936×10^{-4}
6.2.2	obstacle	3	135.2 k		5.472×10^{-3}		6.670×10^{-3}	4.789×10^{-3}	2.302×10^{-3}
6.3.1	rising	3	116.0 M	6.212×10^{-2}	6.436×10^{-2}				4.632×10^{-2}
6.3.1	teapot	3	134.0 M	8.319×10^{-2}	8.693×10^{-2}				5.447×10^{-2}
6.3.2	scalar	3	232.4 M	6.099×10^{-2}					5.032×10^{-2}

Comparison methods. For ablation studies, we implemented several divergence-free kernel-based representations, including Curl Kernels, DFKs-Poly6, and DFKs-Wen4, as well as regular (scalar-valued) RBFs that mimic the Gaussian kernel using compactly supported Wendland’s C^2 polynomial $R_{\text{Wen}2} = (1 - r)_+^4(4r + 1)$. The same initialization and training strategies outlined above are applied. For comparison with INRs, we trained neural network models that use periodic activation functions, dubbed sinusoidal representation networks (SIRENs) [Sitzmann et al. 2020], to capture fine details. In addition to standard SIRENs with physics-informed divergence penalties, we introduced Curl SIREN architectures, which model flow fields by taking the curl of network outputs [Richter-Powell et al. 2024], thereby ensuring inherent divergence-free behavior.

6 EXPERIMENTS

All experiments are conducted on a Windows 11 system equipped with an AMD Ryzen 9 7950X processor and an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM). Unless otherwise specified, all methods are trained using the same number of epochs and batch size for each scenario. We ensure that DFKs-Wen4 consistently have the fewest trainable parameters compared to the other methods by adjusting the number of initialized kernels, while the sizes of neural network models are determined based on the GPU memory limit. A summary of the test case details and the final loss values at convergence can be found in Tab. 1. For further experimental settings and statistics (e.g., the parameter count of each method), please refer to §C in the supplementary document.

Flow visualization. We use three primary approaches to visualize flow fields. (1) *Vorticity mapping*: We calculate the vorticity of the flow and map regions with higher vorticity to greater opacity and warmer colors to highlight intense rotational motion. (2) *HSV color encoding*: Drawing inspiration from optical flow techniques [Baker et al. 2011], we map the direction of the 2D velocity to hue and its (relative) magnitude to saturation, as illustrated in the inset figure. For 3D flow fields, we project the data onto a specified 2D plane in advance. (3) *Line integral convolution (LIC)* [Cabral and Leedom 1993]: This 2D technique represents the flow’s streamlines, offering a clear visualization of its structure.



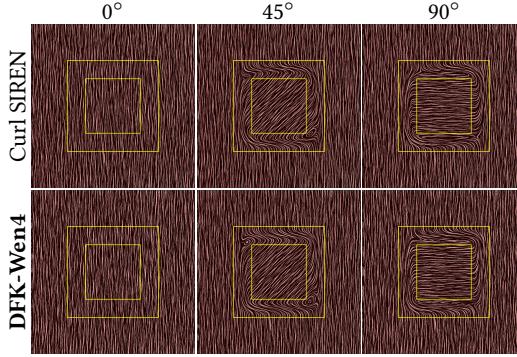


Fig. 7. Inpainting experiments of the laminar flows, visualized using the LIC method. The region between the yellow rectangles is the area where the flow is completed. DFK-Wen4 fits the existing data better and produces more symmetric results. We use large initial kernel radii to encourage smoother inpainting.

6.1 Storage of Complete Flow Fields

We begin by validating DFKs’ representation capabilities through fitting tasks with dense, complete data ($\Omega_D = \Omega$). By optimizing DFKs’ positions, radii, and weights, we are able to store high-fidelity flow fields with far fewer degrees of freedom (DoFs) than the raw data.

6.1.1 Fitting of Incompressible Flows. For dense incompressible flow field data, whether experimentally or synthetically generated, fitting accuracy on such data reflects the suitability of the chosen representation for reconstruction. We compare SIREN, Curl SIREN, Regular RBF, DFK-Poly6, Curl Kernel, and DFK-Wen4 on pure fitting tasks. The loss functions follow Eq. (4) for \mathcal{L}_{obs} , with $\lambda_{\text{div}} = 0.5$ (for non-divergence-free methods) and $\lambda_{\text{bou}} = 1$.

Kármán vortex street (2D). We simulate the steady incoming flow around a cylinder using the method of Narain et al. [2019] on a 512×204 grid, capturing the periodic shedding of counter-rotating vortex pairs, regularly arranged on both sides of the obstacle. After the formation of the vortex street, we select a single frame for fitting. In this scenario, our method employs only 5367 kernels (~ 27 k parameters, compression ratio as low as 8.6%) to achieve the least fitting error. The vorticity fields presented in Fig. 1 clearly illustrate that DFKs-Wen4 yield results with superior clarity and accuracy, effectively capturing the flow dynamics with greater precision.

Analytic vortices (3D). We generate a pointwise divergence-free vector field, going beyond the mere incompressibility requirement in the finite-volume sense typically used in simulations. Specifically, we compute the curl of the following vector potential:

$$\mathbf{A} = \begin{pmatrix} (1 - x^2)(1 - y^2)(1 - z^2) \\ (1 - x^2)(1 - y^2)(1 - z^2) + \sin \pi x \sin \pi y \sin \pi z \\ (1 - x^2)(1 - y^2)(1 - z^2) \end{pmatrix}, \quad (16)$$

and 100 vortex particles are then randomly seeded in the resulting field. The illustration in Fig. 2 indicates that our method with 21 117 kernels initialized has the minimal fitting loss.

Simple plume (3D). Using the method proposed by Fedkiw et al. [2001] and the *mantaflow* framework², we simulate a rising smoke plume on a 128^3 grid and select a single frame for fitting.

²Maintained by Nils Thuerey et al. <http://mantaflow.com/>

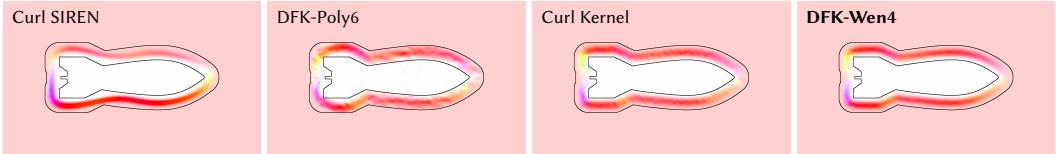


Fig. 8. Inpainting experiments of the missile, visualized using HSV color encoding, reveal clear differences in performance. Compared to DFK-Wen4, Curl SIREN fails to adhere to the solid boundary and lacks symmetry. Meanwhile, both DFK-Poly6 and Curl Kernel exhibit rough velocity fields due to kernel-based artifacts. We choose the same, large initial radii for all these kernels.

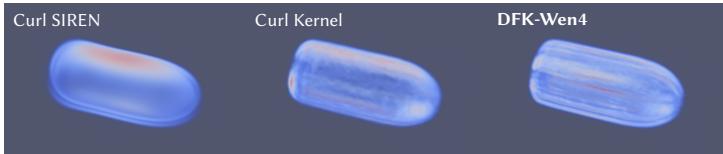


Fig. 9. Inpainting experiments of the bullet with the vorticity fields rendered. Curl SIREN fails to keep the rotational symmetry and performs poorly in fitting the shape of the solid boundary, while the result of Curl Kernel is very rough. DFK-Wen4 has the lowest fitting loss (see Tab. 1) and overcomes these shortcomings.

As shown in Fig. 3, Curl Kernel and DFK-Wen4 achieve the best results, each comprising $\sim 27\text{k}$ parameters, far less than the simulation output ($\sim 6.3\text{M}$).

6.1.2 Leray Projection. A key advantage of DFKs is their ability to provide a local, adaptive Helmholtz decomposition basis. The vector field expressed by Eq. (12) corresponds precisely to the divergence-free component of $-\nabla^2\phi_i(\mathbf{x})\boldsymbol{\omega}$. Therefore, for any given vector field, we can simply apply $-\nabla^2\phi_i$ (a regular RBF) to fit the data without divergence-free constraints, then naturally perform the Leray projection afterwards. Unlike the traditional grid-based projection [Stam 1999], this approach ensures that the resulting flow field is divergence-free at every point. Alternative continuous divergence-free projection methods that combine classical projection [Stam 1999] with recent divergence-free interpolation techniques [Roy-Chowdhury et al. 2024; Schroeder et al. 2022] are limited to uniformly discretized grids. In contrast, the DFK-based projection discussed here is inherently adaptive and free from restrictions on data point distribution.

As noted by Richter-Powell et al. [2024], other divergence-free representations can achieve similar results with proper adjustments. For instance, starting from the Curl Kernel, one can construct a curl-free Gradient Kernel, and their combination can be used for fitting, yielding only the Curl Kernel component:

$$\mathbf{v}(\mathbf{x}) = \nabla \times [R_{\text{Wen4}}(\mathbf{x}) \boldsymbol{\omega}_1] + \omega_2 \nabla R_{\text{Wen4}}(\mathbf{x}). \quad (17)$$

The concept is akin to the Curl SIREN approach. Although these methods introduce additional coefficients, we compare them with DFKs to assess their relative performance.

Taylor vortex (2D). We use velocity field from a specific frame of the Taylor vortex example produced by the optimization-based fluid solver [Xing et al. 2025], which does not guarantee a divergence-free result, as the input for the projection step. Ideally, the projection should eliminate the divergence while preserving the vorticity. As shown in Fig. 5, our method with only 5416 kernels successfully recovers a velocity field with a vorticity distribution closest to the ground truth.

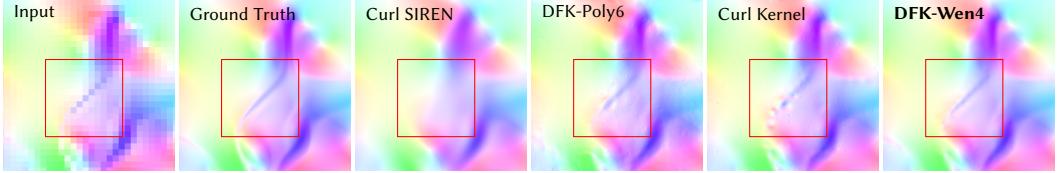


Fig. 10. Super-resolution experiments of the turbulence A, visualized using HSV color encoding, show notable performance differences. DFK-Wen4 is the only approach that recovers the extremely anisotropic, sharp structure in the red box. Note that the input field is very chaotic and asymmetric. The PSNR/SSIM values are as follows: Curl SIREN: 31.14/0.948; DFK-Poly6: 36.40/0.950; Curl Kernel: 37.08/0.965; **DFK-Wen4: 40.93/0.987**.

Vortex ring collision (3D). We project the velocity field from a specific frame of a 3D animation generated by the simulator of Xing et al. [2025]. In this example, two vortex rings gradually approach each other and eventually shred into many smaller rings upon collision. As shown in Fig. 6, our method achieves the best result with only 8544 kernels initialized.

6.2 Completion of Quasi-Static Flow Data

Due to measurement limitations, acquiring complete flow field data is often impractical. Typical challenges include missing data in specific regions or low-resolution sampling. In these cases, the optimization objective remains the same as in the fitting task, but Ω_D is only a subset of Ω now.

As demonstrated in §6.1.1, representations that enforce incompressibility via penalty terms perform significantly worse than alternative methods. Therefore, in this section, we focus our comparison exclusively on divergence-free approaches. Note that there is no additional regularization except for the divergence-free constraint. We choose η in Eq. (15) to be sufficiently large to encourage smoother completion.

6.2.1 Inpainting. Inpainting is a powerful technique for interpolating unknown regions of a flow field from known data [Ozdemir et al. 2024]. It enables the smooth integration of user-defined regions into existing flow fields and can automatically generate local perturbations caused by the introduction of new solid boundaries.

Laminar flows (2D). Inspired by Ozdemir et al. [2024], we explore the task of “stitching” two uniform flow fields with a specified angular offset. As shown in Fig. 7, the regions inside the inner yellow rectangle and outside the outer yellow rectangle exhibit different velocity directions, while the area in between is the target region for intelligent inpainting. We evaluate the performance of Curl SIREN and DFK-Wen4 for completing under three angular offsets: 0° , 45° , and 90° . The results demonstrate that DFKs-Wen4 (5416 kernels) not only better preserve the known velocity field but also generate more symmetric and accurate completions compared to Curl SIREN.

Missile (2D). In wind tunnel experiments, disturbances often arise in the background flow due to the presence of obstacles. Here, we provide a preliminary approximation of such behavior using flow field inpainting³. For this scenario, we define a target completion region around a 2D missile, marked by the area between two black contours in Fig. 8. Outside this region, the flow remains uniformly leftward, while inside, the velocity is constrained to zero due to the solid boundary. Although Curl SIREN achieves the lowest loss value, it compromises boundary adherence. In contrast, our method

³Actual wind tunnel results are influenced by various factors (e.g., scale, wind speed, and turbulence level). As a simple interpolation-based approximation, we assume that smoother results with better adherence to solids are generally preferable.

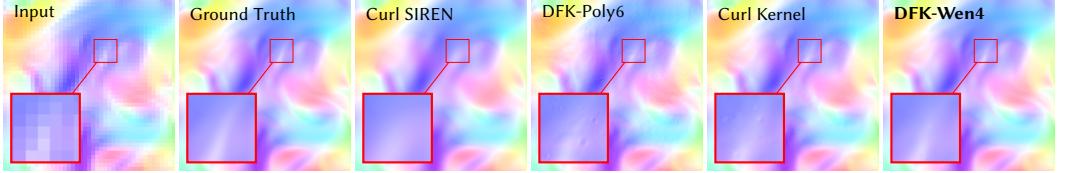


Fig. 11. Super-resolution experiments of turbulence B. The results are visualized using HSV color encoding. Although DFK-Poly6 and Curl Kernel are able to recover the flow field globally, these representations exhibit visible kernel-based artifacts. By optimizing DFKs-Wen4 with different radii and weights, we can successfully recover such chaotic and anisotropic patterns with high accuracy. The PSNR/SSIM values are as follows: Curl SIREN: 36.31/0.970; DFK-Poly6: 37.10/0.956; Curl Kernel: 38.10/0.969; **DFK-Wen4: 41.05/0.991**.

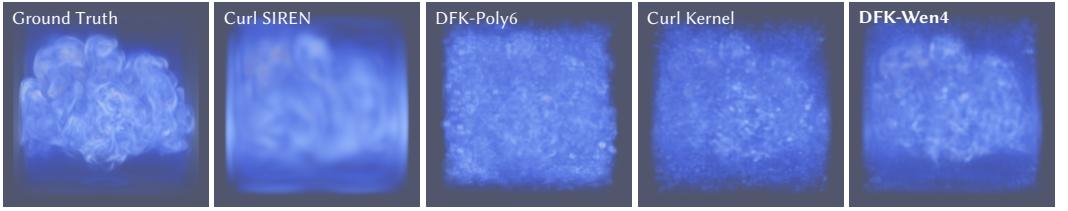


Fig. 12. Super-Resolution experiments of the flow field with a spherical obstacle. The vorticity fields are rendered. DFK-Wen4 shows the best super-resolution capability, especially in recovering the detailed vortices, while the result of Curl SIREN is overly diffuse and those of other kernels are noisy.

with 5390 kernels not only aligns better with the boundary but also produces the smoothest and most symmetric inpainting, providing a more physically realistic flow reconstruction.

Bullet (3D). We conduct experiments to complete the flow field around a 3D bullet as well. As shown in Fig. 9, Curl SIREN fails to maintain symmetry, and the result of Curl Kernel is very rough. DFK-Wen4 (25 325 kernels) generates the smoothest and most symmetric result.

6.2.2 Super-Resolution. Flow field super-resolution is an effective technique for overcoming the challenges of data collection and storage while improving the accuracy and computational efficiency of numerical simulations. By incorporating the continuity equation of incompressible flows as a physical prior, this method enables the recovery of fine-scale fluid details, even from highly compressed or low-resolution data.

Turbulence A/B (2D). We generate two turbulent flow fields caused by multiple velocity sources following Stam [1999] on a 512×512 Cartesian grid, and then downsample them to 64×64 data points. As shown in Figs. 10 and 11, the DFK-Wen4 method excels at recovering even the finest details of the original fields, preserving both the smoothness and clarity of the flow structures.

Spherical obstacle (3D). To assess super-resolution with obstacles, we test using 32^3 data sampled from a smoke simulation on a 128^3 grid, where a sphere is positioned at the center. Our method with 42 002 points initialized delivers the best accuracy as shown in Tab. 1 and Fig. 12.

6.3 Inference of Time-Continuous Flows

Directly measuring a flow field is often challenging in practice. However, passive scalar fields advected by the flow, such as soot density or smoke color, are typically easier to capture with higher accuracy. When diffusion effects are negligible, the flow field can be reconstructed from

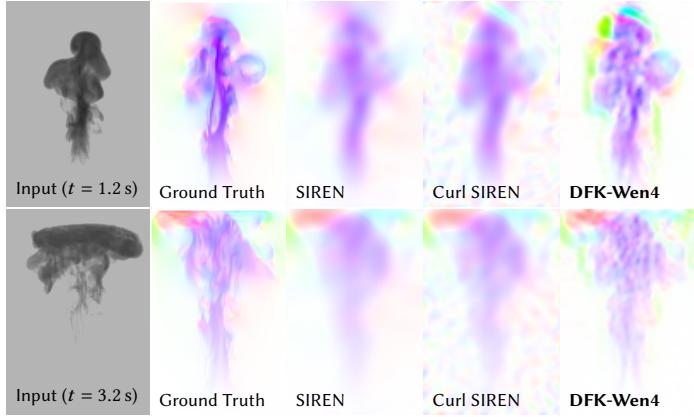


Fig. 13. Inference experiments of the rising smoke. The leftmost image depicts the rendered visualization of the input density field. The reconstructed velocity fields are shown using HSV color encoding, presented as slices viewed along the positive x -axis. The results demonstrate the detailed inference capabilities of DFK-Wen4, which uniquely captures rich flow structures.

the continuous evolution of these passive fields using the advection equation. In this context, the observational loss, \mathcal{L}_{obs} , is defined as shown in Eq. (5).

To mitigate velocity noise in regions with weak supervision and ensure the temporal continuity of the fields, we introduce regularization and discontinuity penalty terms into the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{obs}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}} + \lambda_{\text{bou}} \mathcal{L}_{\text{bou}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \quad (18)$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{V} \int_0^T \int_{\Omega} \|\mathbf{u}\| \, dV \, dt, \quad (19)$$

$$\mathcal{L}_{\text{con}} = \frac{1}{V} \int_0^T \int_{\Omega} \left\| \frac{\partial \mathbf{u}}{\partial t} \right\| \, dV \, dt, \quad (20)$$

where the time derivative of \mathbf{u} is estimated using central differences for second-order accuracy. We found that setting $\lambda_{\text{bou}} = 1$ and $\lambda_{\text{div}} = \lambda_{\text{reg}} = \lambda_{\text{con}} = 0.1$ provides a well-balanced weighting. Note that for divergence-free representations, the value of λ_{div} is ineffective because \mathcal{L}_{div} is always zero.

In dynamic scenarios, a separate SIREN model is trained for each frame in the INR-based approaches. In contrast, kernel-based approaches use fixed kernel positions and radii that are shared across frames, while each frame is assigned a unique set of weights.

6.3.1 From Passive Field Data. We compare the performance of SIREN, Curl SIREN, and DFK-Wen4 in inferring the flow field from complete and accurate passive field data.

Rising (3D). We perform a smoke simulation on a $80 \times 120 \times 80$ Cartesian grid for 150 frames and use the multi-frame density data (soot concentration) to infer the dynamic flow field. As illustrated in Fig. 13, the DFK-Wen4 method (26 375 kernels, only 40% of parameters compared to INRs) uniquely succeeds in inferring detailed vortices within the field, while network-based methods produce overly smooth results, missing critical fine-grained structures.

Teapot (3D). In a similar setup, the passive field data is acquired from the simulated motion of smoke over 100 frames on a $192 \times 128 \times 128$ grid, with a teapot placed at the center of the scene (Fig. 14). As demonstrated in the figure, while SIREN-based methods capture the global behaviors of the fluid, they fail to account for the presence of the obstacles and struggle to reconstruct the thin



Fig. 14. Inference experiments of the teapot. The two leftmost images illustrate the motion direction of the smoke and the position of the teapot. The flow fields are visualized using the HSV color encoding, with slices viewed along the negative y -axis (i.e., top-down view). Among the methods, only DFK-Wen4 accurately captures the influence of the obstacle, recovering the thin structures around it (marked with a black frame).

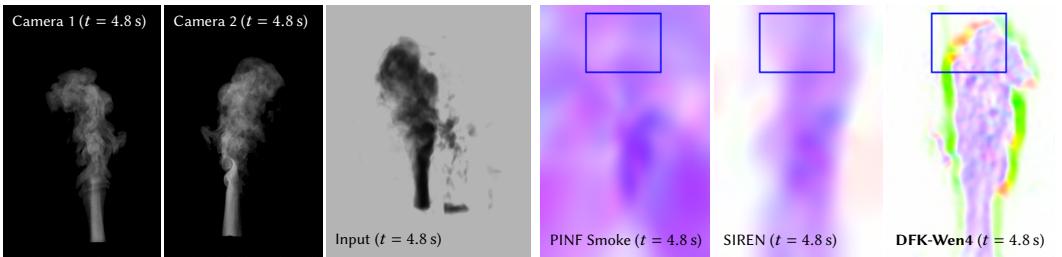


Fig. 15. Inference experiments of the scalar flow. The two leftmost images are real-world captured from two of the five cameras, while the third one presents the rendered visualization of the input density field reconstructed using NeRF. Among the comparison results, only DFK-Wen4 recovers detailed vorticities at the edge of the smoke (marked with a blue frame). Note that PINF Smoke does not include the regularization term used in our loss function.

structures within the flow field. Despite the hard divergence-free constraint, our method achieves 30% less advection loss than INRs with only 60% of trainable parameters.

6.3.2 From Multi-View Videos. By integrating with a NeRF [Mildenhall et al. 2021] frontend, our method can also infer the flow field from multi-view video sequences of dynamic smoke. Here, the implementation follows PINF-Smoke [Chu et al. 2022].

Scalar flow (3D). For inferring flow fields from real captures, we use the ScalarFlow dataset [Eckert et al. 2019], which contains videos of buoyancy-driven rising smoke plumes. Five cameras with fixed positions were evenly distributed across a 120° arc centered on the rising smoke. We use the middle 120 frames from each camera view for fluid reconstruction. After training of PINF-Smoke, its resulting density field is fed as input into our method for further processing. Fig. 15 illustrates the reconstruction results, in which our method shows the best ability to capturing local vortices, while the others only recover a general rising direction. The total loss of DFK-Wen4 is roughly 20% less with only 60% of trainable parameters compared to INRs.

7 CONCLUSION & DISCUSSIONS

In this paper, we develop a new kernel-based framework for high-fidelity and physically accurate reconstruction of incompressible flow fields. Compared to recent approaches, our framework uniquely employs matrix-valued kernel functions—specifically DFKs-Wen4—as analytically divergence-free approximations of the velocity field. DFKs-Wen4 offer several distinct advantages over alternative representations of incompressible velocity fields: their dipole nature aligns with fundamental flow solutions, enabling accurate modeling of core fluid dynamics; their compact support is ideal for capturing local vortex structures and handling boundary conditions; their positive definiteness ensures

superior convergence rates, while their differentiability allows for smooth and continuous modeling of flow fields. Across a variety of flow field reconstruction tasks, from fitting and extrapolation to inference, the framework employing DFKs-Wen4 consistently outperforms competing approaches. Consequently, we believe DFKs-Wen4 hold great promise as a foundational representation capable of replacing INRs for flow field reconstruction.

Beyond their application in inverse problems, DFKs-Wen4, with their pointwise divergence-free expressive power, strong fitting capabilities, and rapid convergence, exhibit exceptional potential for forward simulations. Experiments in §6.1.2 have demonstrated their effectiveness in pressure computation, suggesting the feasibility of a fully pointwise incompressible mesh-free fluid simulation framework. Such a framework could unify representations for both forward and inverse tasks, fostering seamless integration of fluid generation and understanding.

Zero net momentum. Assuming uniform density, a flow field represented by a single DFK inherently possesses zero net linear and angular momentum (see §A.3 for proof). Consequently, any field constructed from multiple DFKs must also exhibit this property over the entire space. However, in practical settings—where the domains of interest are bounded—it is always possible for some DFKs to have supports that extend beyond the domain boundaries, thereby inducing non-zero linear and angular momentum within the domain. This explains why DFKs can still accurately capture flows exhibiting global motion, such as the one shown in Fig. 1⁴. Furthermore, we note that any compactly supported, divergence-free kernel-based representation must have zero net linear momentum (see §A.3). Regarding angular momentum, while it is theoretically feasible to construct a novel kernel with non-zero net angular momentum by linearly combining different kernels (e.g., Curl Kernel and DFK-Wen4), such constructions typically compromise the positive definiteness. For flows with non-zero total linear or angular momentum, a more practical and effective strategy may be to augment the DFKs with an optimizable, uniform base field to enhance convergence.

Limitations. A key limitation of this work is that our validation data are mainly derived from numerical simulations rather than real-world measurements. Expanding the DFK-Wen4 representation to tackle interdisciplinary applications involving real-world data is a natural next step. Another simplification in this paper is the exclusive use of no-slip boundary conditions. While free-slip conditions can be incorporated by introducing dot products into \mathcal{L}_{bou} , modeling free-surface flows remains an open challenge that warrants further investigation. Moreover, when inferring time-continuous flows, our method may still generate spurious small-scale vortices (see Figs. 14 and 15), as it does not explicitly suppress noise—despite achieving lower losses and better boundary adherence compared to existing alternatives.

Future work. Currently, the RBF underlying DFKs-Wen4 (i.e., Wendland’s C^4 polynomial) is isotropic. Inspired by recent advancements in 3DGS [Kerbl et al. 2023], adapting DFKs to use anisotropic ellipsoids could further enhance their expressive capabilities. Moreover, the DFKs presented in this paper are restricted to flow fields in Euclidean space. Extending DFKs to surface-based kernels [Narcowich et al. 2007] and applying them to flows on manifolds—such as those found in soap films or planetary atmospheres—opens up a compelling avenue for future exploration.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was supported by the National Key R&D Program of China (2022ZD0160802). We would like to acknowledge Mengyu Chu for help on integration with PINF-Smoke.

⁴In this configuration, the fluid’s center of mass has a vertical downward velocity of 0.5 m s^{-1} .

REFERENCES

- Mark Ya Agre. 2011. Multipole expansions in magnetostatics. *Physics-Uspekhi* 54, 2 (Feb. 2011), 167.
- Vishal Anantharaman, Jason Feldkamp, Kai Fukami, and Kunihiko Taira. 2023. Image and video compression of fluid flow data. *Theoretical and Computational Fluid Dynamics* 37 (Feb. 2023), 1–22.
- Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. 2008. Time-resolved 3d capture of non-stationary gas flows. *ACM Trans. Graph.* 27, 5, Article 132 (Dec. 2008), 9 pages.
- Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. 2011. A Database and Evaluation Methodology for Optical Flow. *Int. J. Comput. Vision* 92, 1 (March 2011), 1–31.
- Jan Bender and Dan Koschier. 2015. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) (SCA '15). Association for Computing Machinery, New York, NY, USA, 147–155.
- Haimasree Bhattacharya, Michael Bang Nielsen, and Robert Bridson. 2012. Steady State Stokes Flow Interpolation for Fluid Control. In *Annual Conference of the European Association for Computer Graphics*, Vol. 33. Eurographics Association, Cagliari, Sardinia, Italy, 57–60.
- Robert Bridson. 2007. Fast Poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 Sketches* (San Diego, California) (SIGGRAPH '07). Association for Computing Machinery, New York, NY, USA, 22–es.
- Robert Bridson. 2015. *Fluid simulation for computer graphics* (2nd ed.). AK Peters/CRC Press, Boca Raton, FL, USA.
- Brian Cabral and Leith Casey Leedom. 1993. Imaging vector fields using line integral convolution. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) (SIGGRAPH '93). Association for Computing Machinery, New York, NY, USA, 263–270.
- José A. Canabal, David Miraut, Nils Thuerey, Theodore Kim, Javier Portilla, and Miguel A. Otaduy. 2016. Dispersion kernels for water wave simulation. *ACM Trans. Graph.* 35, 6, Article 202 (Dec. 2016), 10 pages.
- Antoine Chan-Lock, Jesús Pérez, and Miguel A. Otaduy. 2022. High-Order Elasticity Interpolants for Microstructure Simulation. *Computer Graphics Forum* 41, 8 (2022), 63–74.
- Jumyoung Chang, Ruben Partono, Vinicius C. Azevedo, and Christopher Batty. 2022. Curl-Flow: Boundary-Respecting Pointwise Incompressible Velocity Interpolation for Grid-Based Fluids. *ACM Trans. Graph.* 41, 6, Article 243 (Nov. 2022), 21 pages.
- Honglin Chen, Rundi Wu, Eitan Grinspun, Changxi Zheng, and Peter Yichen Chen. 2023. Implicit neural spatial representations for time-dependent PDEs. In *Proceedings of the 40th International Conference on Machine Learning* (ICML '23). JMLR.org, Honolulu, Hawaii, USA, Article 202, 16 pages.
- Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. 2022. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Trans. Graph.* 41, 4, Article 119 (July 2022), 14 pages.
- Yitong Deng, Hong-Xing Yu, Jiajun Wu, and Bo Zhu. 2023a. Learning Vortex Dynamics for Fluid Inference and Prediction. arXiv:2301.11494 [cs.LG]
- Yitong Deng, Hong-Xing Yu, Diyang Zhang, Jiajun Wu, and Bo Zhu. 2023b. Fluid Simulation on Neural Flow Maps. *ACM Trans. Graph.* 42, 6, Article 248 (Dec. 2023), 21 pages.
- Zhiwen Deng, Chuangxin He, Yingzheng Liu, and Kyung Chun Kim. 2019. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Physics of Fluids* 31, 12 (2019), 125111.
- Marie-Lena Eckert, Kiwon Um, and Nils Thuerey. 2019. ScalarFlow: a large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. *ACM Trans. Graph.* 38, 6, Article 239 (Nov. 2019), 16 pages.
- Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. 2001. Visual simulation of smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '01). Association for Computing Machinery, New York, NY, USA, 15–22.
- Nick Foster and Ronald Fedkiw. 2001. Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '01). Association for Computing Machinery, New York, NY, USA, 23–30.
- Erik Franz, Barbara Solenthaler, and Nils Thuerey. 2021. Global Transport for Fluid Reconstruction with Learned Self-Supervision. In *Proceedings - 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021 (Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition)*. IEEE Computer Society, Nashville, TN, USA, 1632–1642.
- Kai Fukami, Koji Fukagata, and Kunihiko Taira. 2019. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics* 870 (2019), 106–120.
- Edward J. Fuselier, Varun Shankar, and Grady B. Wright. 2016. A high-order radial basis function (RBF) Leray projection method for the solution of the incompressible unsteady Stokes equations. *Computers & Fluids* 128 (2016), 41–52.

- Han Gao, Luning Sun, and Jian-Xun Wang. 2021. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids* 33, 7 (07 2021), 073603.
- James Gregson, Ivo Ihrke, Nils Thuerey, and Wolfgang Heidrich. 2014. From capture to simulation: connecting forward and inverse problems in fluids. *ACM Trans. Graph.* 33, 4, Article 139 (July 2014), 11 pages.
- David J. Griffiths. 2017. *Introduction to Electrodynamics* (fourth ed.). Cambridge University Press, Cambridge, UK.
- Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.* 37, 4, Article 150 (July 2018), 14 pages.
- Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédéric Durand. 2019. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Trans. Graph.* 38, 6, Article 201 (Nov. 2019), 16 pages.
- Yu Ji, Jinwei Ye, and Jingyi Yu. 2013. Reconstructing Gas Flows Using Light-Path Approximation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. IEEE Computer Society, USA, 2507–2514.
- Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4, Article 139 (July 2023), 14 pages.
- Byungsoo Kim, Vinicius C. Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. 2019. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Computer Graphics Forum* 38, 2 (2019), 59–70.
- Doyub Kim, Minjae Lee, and Ken Museth. 2024. NeuralVDB: High-resolution Sparse Volume Representation using Hierarchical Neural Networks. *ACM Trans. Graph.* 43, 2, Article 20 (Feb. 2024), 21 pages.
- Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]
- Jingwei Li, Zhiming Gao, Zihuan Dai, and Xinlong Feng. 2020. Divergence-free radial kernel for surface Stokes equations based on the surface Helmholtz decomposition. *Computer Physics Communications* 256 (2020), 107408.
- Bo Liu, Jiupeng Tang, Haibo Huang, and Xi-Yun Lu. 2020. Deep learning methods for super-resolution reconstruction of turbulent flows. *Physics of Fluids* 32, 2 (02 2020), 025105.
- Yang Liu, Yueqing Wang, Liang Deng, Fang Wang, Fang Liu, Yutong Lu, and Sikun Li. 2019. A novel in situ compression method for CFD data based on generative adversarial network. *Journal of Visualization* 22 (Feb. 2019), 95–108. Issue 1.
- Svenja Lowitzsch. 2005. Error estimates for matrix-valued radial basis function interpolation. *Journal of Approximation Theory* 137, 2 (2005), 238–249.
- Luan Lyu, Xiaohua Ren, Wei Cao, Jian Zhu, Enhua Wu, and Zhi-Xin Yang. 2024. Wavelet Potentials: An Efficient Potential Recovery Technique for Pointwise Incompressible Fluids. *Computer Graphics Forum* 43, 2 (2024), e15023.
- Ives Macêdo and René Castro. 2010. Learning divergence-free and curl-free vector fields with matrix-valued kernels. preprint.ima.br:1823.1633
- Colin P. McNally. 2011. Divergence-free interpolation of vector fields from point values – exact $\nabla \cdot B = 0$ in numerical simulations. *Monthly Notices of the Royal Astronomical Society: Letters* 413, 1 (05 2011), L76–L80.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (Dec. 2021), 99–106.
- Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) (SCA '03). Eurographics Association, Goslar, DEU, 154–159.
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages.
- Mohammad Sina Nabizadeh, Ritoban Roy-Chowdhury, Hang Yin, Ravi Ramamoorthi, and Albert Chern. 2024. Fluid Implicit Particles on Coadjoint Orbits. *ACM Trans. Graph.* 43, 6, Article 270 (Nov. 2024), 38 pages.
- Rahul Narain, Jonas Zehnder, and Bernhard Thomaszewski. 2019. A Second-Order Advection-Reflection Solver. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 16 (July 2019), 14 pages.
- Francis J. Narcowich and Joseph D. Ward. 1994. Generalized Hermite interpolation via matrix-valued conditionally positive definite functions. *Math. Comput.* 63, 208 (Oct. 1994), 661–687.
- Francis J. Narcowich, Joseph D. Ward, and Grady B. Wright. 2007. Divergence-Free RBFs on Surfaces. *Journal of Fourier Analysis and Applications* 13, 6 (2007), 643–663.
- Michael B. Nielsen and Robert Bridson. 2011. Guide shapes for high resolution naturalistic liquid simulation. *ACM Trans. Graph.* 30, 4, Article 83 (July 2011), 8 pages.
- Tumay Ozdemir, Jiamin Shi, Nathan King, and Christopher Batty. 2024. Editing Fluid Flows with Divergence-Free Biharmonic Vector Field Interpolation. In *SIGGRAPH Asia 2024 Technical Communications (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 41, 4 pages.

- M. Raissi, P. Perdikaris, and G.E. Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378 (2019), 686–707.
- Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* 367, 6481 (2020), 1026–1030.
- Jack Richter-Powell, Yaron Lipman, and Ricky T. Q. Chen. 2024. Neural conservation laws: a divergence-free perspective. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (*NIPS* '22). Curran Associates Inc., Red Hook, NY, USA, Article 2759, 14 pages.
- Ritoban Roy-Chowdhury, Tamar Shinar, and Craig Schroeder. 2024. Higher order divergence-free and curl-free interpolation on MAC grids. *J. Comput. Phys.* 503 (2024), 112831.
- Syuhei Sato, Yoshinori Dobashi, and Tomoyuki Nishita. 2018. Editing Fluid Animation Using Flow Interpolation. *ACM Trans. Graph.* 37, 5, Article 173 (Sept. 2018), 12 pages.
- Craig Schroeder, Ritoban Roy Chowdhury, and Tamar Shinar. 2022. Local divergence-free polynomial interpolation on MAC grids. *J. Comput. Phys.* 468 (2022), 111500.
- Luca Schweri, Sébastien Fouquer, Jingwei Tang, Vinicius C. Azevedo, Tobias Günther, and Barbara Solenthaler. 2021. A Physics-Aware Neural Network Approach for Flow Data Reconstruction From Satellite Observations. *Frontiers in Climate* 3 (2021), 13 pages.
- Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) (*NIPS* '20). Curran Associates Inc., Red Hook, NY, USA, Article 626, 12 pages.
- O. Skrinjar, A. Bustoquet, J. Oshinski, K. Sundareswaran, D. Frakes, and A. Yoganathan. 2009. A divergence-free vector field model for imaging applications. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, Boston, MA, USA, 891–894.
- Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., USA, 121–128.
- Patrick Vogler and Ulrich Rist. 2017. Wavelet-Based Compression of Volumetric CFD Data Sets. In *Sustained Simulation Performance 2017*, Michael M. Resch, Wolfgang Bez, Erich Focht, Michael Gienger, and Hiroaki Kobayashi (Eds.). Springer International Publishing, Cham, 123–136.
- Chulin Wang, Eloisa Bentivegna, Wang Zhou, Levente Klein, and Bruce Elmegreen. 2020. Physics-Informed Neural Network Super Resolution for Advection-Diffusion Models. arXiv:2011.02519 [cs.CV]
- Yiming Wang, Siyu Tang, and Mengyu Chu. 2024. Physics-Informed Learning of Characteristic Trajectories for Smoke Reconstruction. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (*SIGGRAPH* '24). Association for Computing Machinery, New York, NY, USA, Article 53, 11 pages.
- Holger Wendland. 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* 4, 1 (1995), 389–396.
- Holger Wendland. 2004. *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK.
- Holger Wendland. 2009. Divergence-Free Kernel Methods for Approximating the Stokes Problem. *SIAM J. Numer. Anal.* 47, 4 (2009), 3158–3179.
- Maximilian Werhahn, You Xie, Mengyu Chu, and Nils Thuerey. 2019. A Multi-Pass GAN for Fluid Flow Super-Resolution. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 10 (July 2019), 21 pages.
- Zhao Wu, Tamer A. Zaki, and Charles Meneveau. 2020. Data compression for turbulence databases using spatiotemporal subsampling and local resimulation. *Phys. Rev. Fluids* 5 (June 2020), 064607. Issue 6.
- You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. 2018. tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Trans. Graph.* 37, 4, Article 95 (July 2018), 15 pages.
- Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022. Neural Fields in Visual Computing and Beyond. *Computer Graphics Forum* 41, 2 (2022), 641–676.
- Jingrui Xing, Bin Wang, Mengyu Chu, and Baoquan Chen. 2025. Gaussian Fluids: A Grid-Free Fluid Solver based on Gaussian Spatial Representation. In *ACM SIGGRAPH 2025 Conference Papers* (Vancouver, BC, Canada) (*SIGGRAPH* '25). Association for Computing Machinery, New York, NY, USA, Article 31, 11 pages.
- Jinhui Xiong, Ramzi Idoughi, Andres A. Aguirre-Pablo, Abdulrahman B. Aljedaani, Xiong Dun, Qiang Fu, Sigurdur T. Thoroddsen, and Wolfgang Heidrich. 2017. Rainbow particle imaging velocimetry for dense 3D fluid velocity imaging. *ACM Trans. Graph.* 36, 4, Article 36 (July 2017), 14 pages.
- Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. 2022. Point-NeRF: Point-based Neural Radiance Fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, New Orleans, LA, USA, 5428–5438.

- Hong-Xing Yu, Yang Zheng, Yuan Gao, Yitong Deng, Bo Zhu, and Jiajun Wu. 2023. Inferring hybrid neural fluid fields from videos. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) (NIPS '23). Curran Associates Inc., Red Hook, NY, USA, Article 2777, 14 pages.
- Jihun Yu and Greg Turk. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.* 32, 1, Article 5 (Feb. 2013), 12 pages.
- Chengxi Zeng, Tilo Burghardt, and Alberto M Gambaruto. 2024. RBF-PINN: Non-Fourier Positional Embedding in Physics-Informed Neural Networks. arXiv:2402.08367 [cs.LG]
- Xinhuan Zhou, Virginie Papadopoulou, Chee Hau Leow, Peter Vincent, and Meng-Xing Tang. 2019. 3-D Flow Reconstruction Using Divergence-Free Interpolation of Multiple 2-D Contrast-Enhanced Ultrasound Particle Imaging Velocimetry Measurements. *Ultrasound in Medicine & Biology* 45, 3 (2019), 795–810.
- Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972.

Representing Flow Fields with Divergence-Free Kernels for Reconstruction (Supplementary Document)

XINGYU NI, School of Computer Science, Peking University, China

JINGRUI XING, School of Intelligence Science and Technology, Peking University, China

XINGQIAO LI, School of Intelligence Science and Technology, Peking University, China

BIN WANG*, Independent, China

BAOQUAN CHEN*, State Key Laboratory of General Artificial Intelligence, Peking University, China

A MATRIX-VALUED RADIAL BASIS FUNCTIONS

As proposed by [Narcowich and Ward \[1994\]](#), matrix-valued radial basis functions (RBFs) are used to represent a vector field $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ by

$$\tilde{f}(\mathbf{x}) = \sum_{k=1}^N \psi(\mathbf{x} - \mathbf{x}_k) \boldsymbol{\alpha}_k, \quad (\text{S1})$$

where $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ represents a set of scattered points, with corresponding vector weights $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \dots, \boldsymbol{\omega}_N \in \mathbb{R}^d$, and $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ is the matrix-valued kernel.

A.1 Positive Definiteness

A continuous complex-valued matrix function $\psi : \mathbb{R}^d \rightarrow \mathbb{C}^{n \times n}$ is said to be *positive semi-definite* if, for any set of pairwise distinct points $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$ and any set of vector coefficients $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_N \in \mathbb{C}^n$, the following inequality holds:

$$0 \leq \sum_{j=1}^N \sum_{k=1}^N \overline{\boldsymbol{\alpha}_j}^\top \psi(\mathbf{x}_j - \mathbf{x}_k) \boldsymbol{\alpha}_k \in \mathbb{R}. \quad (\text{S2})$$

Furthermore, $\psi(\mathbf{x})$ is said to be *positive definite* if equality holds if and only if $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2 = \dots = \boldsymbol{\alpha}_N = \mathbf{0}$.

A common approach to constructing matrix-valued positive definite functions is to apply second-order differential operators to scalar-valued positive definite functions [\[Wendland 2009\]](#). To analyze the positive definiteness of such matrix-valued functions, we first establish Lemma A.1.

LEMMA A.1. *Let $\mathbf{x} \in \mathbb{R}^d$ have components x^1, x^2, \dots, x^d . Suppose $\psi \in C^2(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$ is a complex-valued function with Fourier transform $\widehat{\psi}(\boldsymbol{\omega})$. Then, the Fourier transform of its second-order partial derivatives satisfy*

$$\mathcal{F} \left[\frac{\partial^2 \psi}{\partial x^p \partial x^q} \right] (\boldsymbol{\omega}) = -\omega^m \omega^n \widehat{\psi}(\boldsymbol{\omega}), \quad m, n = 1, 2, \dots, d. \quad (\text{S3})$$

PROOF. The computation of the Fourier transform proceeds as follows:

$$\mathcal{F} \left[\frac{\partial^2 \psi}{\partial x^p \partial x^q} \right] (\boldsymbol{\omega}) = \int_{\mathbb{R}^d} \frac{\partial}{\partial x^p} \left[\frac{\partial \psi}{\partial x^q} (\mathbf{x}) \right] e^{-i\mathbf{x} \cdot \boldsymbol{\omega}} d\mathbf{x}$$

*corresponding authors

Authors' addresses: [Xingyu Ni](#), nixy@pku.edu.cn, School of Computer Science, Peking University, Beijing, China; [Jingrui Xing](#), xjr01@hotmail.com, School of Intelligence Science and Technology, Peking University, Beijing, China; [Xingqiao Li](#), lixingqiao@pku.edu.cn, School of Intelligence Science and Technology, Peking University, Beijing, China; [Bin Wang](#), binwangbuaa@gmail.com, Independent, Beijing, China; [Baoquan Chen](#), baoquan@pku.edu.cn, State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China.

$$\begin{aligned}
&= \int_{\mathbb{R}^d} \frac{\partial}{\partial x^p} \left[e^{-i\mathbf{x}\cdot\omega} \frac{\partial \psi}{\partial x^q}(\mathbf{x}) \right] d\mathbf{x} - \int_{\mathbb{R}^d} \frac{\partial e^{-i\mathbf{x}\cdot\omega}}{\partial x^p} \frac{\partial \psi(\mathbf{x})}{\partial x^q} d\mathbf{x} \\
&= - \int_{\mathbb{R}^d} \frac{\partial e^{-i\mathbf{x}\cdot\omega}}{\partial x^p} \frac{\partial \psi(\mathbf{x})}{\partial x^q} d\mathbf{x} \\
&= - \int_{\mathbb{R}^d} \frac{\partial}{\partial x^q} \left[\psi(\mathbf{x}) \frac{\partial e^{-i\mathbf{x}\cdot\omega}}{\partial x^p} \right] d\mathbf{x} + \int_{\mathbb{R}^d} \psi(\mathbf{x}) \frac{\partial^2 e^{-i\mathbf{x}\cdot\omega}}{\partial x^q \partial x^p} d\mathbf{x} \\
&= \int_{\mathbb{R}^d} \psi(\mathbf{x}) \frac{\partial^2 e^{-i\mathbf{x}\cdot\omega}}{\partial x^q \partial x^p} d\mathbf{x} \\
&= -\omega^p \omega^q \int_{\mathbb{R}^d} \psi(\mathbf{x}) e^{-i\mathbf{x}\cdot\omega} d\mathbf{x} \\
&= -\omega^p \omega^q \widehat{\psi}(\omega),
\end{aligned} \tag{S4}$$

where the second and fourth equalities follow from integration by parts, and the third and fifth equalities follow from the divergence theorem. \square

Let $\nabla = (\partial/\partial x^1, \partial/\partial x^2, \dots, \partial/\partial x^d)$ denote the gradient operator in column vector form. We can now establish the following theorem regarding the construction of matrix-valued positive definite functions.

THEOREM A.2. Suppose $\psi \in C^2(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$ is a scalar-valued positive definite function. Then, the matrix-valued functions

$$\psi_1(\mathbf{x}) = -\Delta\psi(\mathbf{x}) \mathbf{I}, \tag{S5}$$

$$\psi_2(\mathbf{x}) = -\nabla\nabla^\top\psi(\mathbf{x}), \tag{S6}$$

$$\psi_3(\mathbf{x}) = -\Delta\psi(\mathbf{x}) \mathbf{I} + \nabla\nabla^\top\psi(\mathbf{x}), \tag{S7}$$

are positive definite as long as they are integrable over \mathbb{R}^d .

PROOF. The case for ψ_1 follows directly from the fact that the Laplace operator preserves the positive definiteness of the function [Wendland 2004, Lemma 9.15] and the definition of matrix-valued positive definite functions. For ψ_2 and ψ_3 , let $S = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \subset \mathbb{R}^d$ be an arbitrary set of distinct points, and let $\alpha_1, \alpha_2, \dots, \alpha_N \in \mathbb{C}^d$ be arbitrary vector coefficients.

We first consider the following summation for ψ_2 using Lemma A.1:

$$\begin{aligned}
\sum_{j,k=1}^N \overline{\alpha_j}^\top \psi_2(\mathbf{x}_j - \mathbf{x}_k) \alpha_k &= - \sum_{j,k=1}^N \sum_{p,q=1}^d \overline{\alpha_j^p} \alpha_k^q \frac{\partial^2 \psi}{\partial p \partial q}(\mathbf{x}_j - \mathbf{x}_k) \\
&= -\frac{1}{(\sqrt{2\pi})^d} \sum_{j,k=1}^N \sum_{p,q=1}^d \overline{\alpha_j^p} \alpha_k^q \int_{\mathbb{R}^d} \mathcal{F} \left[\frac{\partial^2 \psi}{\partial x^p \partial x^q} \right](\omega) e^{i(\mathbf{x}_j - \mathbf{x}_k) \cdot \omega} d\omega \\
&= \frac{1}{(\sqrt{2\pi})^d} \sum_{j,k=1}^N \sum_{p,q=1}^d \overline{\alpha_j^p} \alpha_k^q \int_{\mathbb{R}^d} \omega^p \omega^q \widehat{\psi}(\omega) e^{i(\mathbf{x}_j - \mathbf{x}_k) \cdot \omega} d\omega \\
&= \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} \widehat{\psi}(\omega) \left(\sum_{j=1}^N \sum_{p=1}^d \overline{\alpha_j^p} \omega^p e^{i\mathbf{x}_j \cdot \omega} \sum_{k=1}^N \sum_{q=1}^d \alpha_k^q \omega^q e^{-i\mathbf{x}_k \cdot \omega} \right) d\omega \\
&= \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} \widehat{\psi}(\omega) \left| \sum_{j=1}^N \sum_{p=1}^d \overline{\alpha_j^p} \omega^p e^{i\mathbf{x}_j \cdot \omega} \right|^2 d\omega.
\end{aligned} \tag{S8}$$

As given, $\widehat{\psi}(\omega)$ is a non-negative function that is not identically zero, so Eq. (S8) is always greater than 0, thus ψ_2 is positive definite. Observing the proof, we can express the Fourier transform of the Hessian matrix of the second-order partial derivatives as the matrix of Fourier transforms of its components:

$$\mathcal{F} [\nabla \nabla^\top \psi](\omega) = -\omega \omega^\top \widehat{\psi}(\omega), \quad (\text{S9})$$

and thus Eq. (S8) can be written as

$$\sum_{j,k=1}^N \overline{\alpha_j}^\top \psi_2(x_j - x_k) \alpha_k = \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} \widehat{\psi}(\omega) \left| \sum_{j=1}^N \overline{\alpha_j} \cdot \omega e^{ix_j \cdot \omega} \right|^2 d\omega. \quad (\text{S10})$$

Similarly, we apply the same analysis to ψ_3 using Fourier analysis to prove:

$$\begin{aligned} \sum_{j,k=1}^N \overline{\alpha_j}^\top \psi_2(x_j - x_k) \alpha_k &= \frac{1}{(\sqrt{2\pi})^d} \sum_{j,k=1}^N \int_{\mathbb{R}^d} \overline{\alpha_j}^\top (\|\omega\|^2 I - \omega \omega^\top) \alpha_k \widehat{\psi}(\omega) e^{i(x_j - x_k) \cdot \omega} d\omega \\ &= \frac{1}{(\sqrt{2\pi})^d} \sum_{j,k=1}^N \int_{\mathbb{R}^d} \overline{\alpha_j}^\top L^\top L \alpha_k \widehat{\psi}(\omega) e^{i(x_j - x_k) \cdot \omega} d\omega \\ &= \frac{1}{(\sqrt{2\pi})^d} \int_{\mathbb{R}^d} \widehat{\psi}(\omega) \left\| \sum_{j=1}^N L \overline{\alpha_j} e^{ix_j \cdot \omega} \right\|^2 d\omega, \end{aligned} \quad (\text{S11})$$

where $L^\top L = \|\omega\|^2 I - \omega \omega^\top$ is the result of the Cholesky decomposition. Note that for any $\alpha \in \mathbb{R}^d$, we can obtain

$$\alpha^\top (\|\omega\|^2 I - \omega \omega^\top) \alpha = \|\omega\|^2 \|\alpha\|^2 - (\omega \cdot \alpha)^2 \geq 0, \quad (\text{S12})$$

so $\|\omega\|^2 I - \omega \omega^\top$ is semi-positive definite, ensuring the existence of L . \square

A.2 Helmholtz Decomposition

The matrix-valued positive definite functions constructed according to Theorem A.2 possess a series of important properties.

THEOREM A.3. *For any scalar function $\psi \in C^2(\mathbb{R}^d)$ and any vector coefficient $\alpha \in \mathbb{C}^d$, $\psi_2 \alpha = -\nabla \nabla^\top \psi(x) \alpha$ is curl-free, i.e., $\nabla \times (\psi_2 \alpha) = 0$.*

PROOF. By the linearity of the ∇ operator, we can express $\psi_2 \alpha$ as

$$-\nabla \nabla^\top \psi(x) \alpha = \nabla [-\nabla \psi(x) \cdot \alpha], \quad (\text{S13})$$

which shows that $\psi_2 \alpha$ is the gradient of a scalar field, and its curl must be zero. \square

THEOREM A.4. *For any scalar function $\psi \in C^2(\mathbb{R}^d)$ and any vector coefficient $\alpha \in \mathbb{C}^d$, $\psi_3 \alpha = -\Delta \psi(x) \alpha + \nabla \nabla^\top \psi(x) \alpha$ is divergence-free, i.e., $\nabla \cdot (\psi_3 \alpha) = 0$.*

PROOF. By the linearity of the ∇ operator, we can rewrite the expression for $\psi_3 \alpha$ as follows:

$$\begin{aligned} -\Delta \psi(x) \alpha + \nabla \nabla^\top \psi(x) \alpha &= -\nabla \cdot [\nabla \psi(x)] \alpha + \nabla [\nabla \psi(x) \cdot \alpha] \\ &= \nabla \times (\nabla \psi \times \alpha), \end{aligned} \quad (\text{S14})$$

which shows that $\psi_3 \alpha$ is the curl of a vector field, and its divergence must be zero. \square

Clearly, from Theorem A.3, Theorem A.4, and the Helmholtz decomposition theorem for vector fields, we can deduce the following corollary.

COROLLARY A.5. Let $\psi \in C^2(\mathbb{R}^d)$ be a scalar function, $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$ be an arbitrary set of distinct points, and $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_N \in \mathbb{C}^d$ be arbitrary vector coefficients. The vector field

$$\tilde{f}_1(\mathbf{x}) = - \sum_{k=1}^N \Delta\psi(\mathbf{x} - \mathbf{x}_k) \boldsymbol{\alpha}_k \quad (\text{S15})$$

can be uniquely decomposed into the curl-free field

$$\tilde{f}_2(\mathbf{x}) = - \sum_{k=1}^N \nabla^\top \nabla \psi(\mathbf{x} - \mathbf{x}_k) \boldsymbol{\alpha}_k \quad (\text{S16})$$

and the divergence-free field

$$\tilde{f}_3(\mathbf{x}) = - \sum_{k=1}^N \Delta\psi(\mathbf{x} - \mathbf{x}_k) \boldsymbol{\alpha}_k + \sum_{k=1}^N \nabla^\top \nabla \psi(\mathbf{x} - \mathbf{x}_k) \boldsymbol{\alpha}_k \quad (\text{S17})$$

as their sum.

A.3 Zero Net Momentum

For a homogeneous incompressible fluid, any flow field constructed using the formulation in Eq. (S14), with an arbitrary weight vector $\boldsymbol{\alpha}$ and radial basis function ψ , inherently has zero net linear and angular momentum.

LEMMA A.6. For any vector function $\mathbf{F} \in C^1(\mathbb{R}^d)$ that satisfies $\lim_{\|\mathbf{x}\| \rightarrow \infty} \mathbf{F}(\mathbf{x}) = \mathbf{c}$ ($\mathbf{c} \in \mathbb{R}^d$ is a constant vector), the integral of its curl over the entire space vanishes:

$$\int_{\mathbb{R}^d} (\nabla \times \mathbf{F}) dV = \mathbf{0}. \quad (\text{S18})$$

PROOF. We only prove the case $d = 3$ here, as the case $d = 2$ is similar.

Let $\mathbf{F} = (F_x \ F_y \ F_z)^\top$. We can then verify Eq. (S18) component-wise as follows:

$$\begin{aligned} \iiint_{\mathbb{R}^3} (\nabla \times \mathbf{F}) dV &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \quad \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \quad \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right)^\top dx dy dz \\ &= \left(\iint_{\mathbb{R}^2} [F_z(x, +\infty, z) - F_z(x, -\infty, z)] dx dz - \iint_{\mathbb{R}^2} [F_y(x, y, +\infty) - F_y(x, y, -\infty)] dx dy \right) \\ &\quad \left(\iint_{\mathbb{R}^2} [F_x(x, y, +\infty) - F_x(x, y, -\infty)] dx dy - \iint_{\mathbb{R}^2} [F_z(+\infty, y, z) - F_z(-\infty, y, z)] dy dz \right) \\ &\quad \left(\iint_{\mathbb{R}^2} [F_y(+\infty, y, z) - F_y(-\infty, y, z)] dy dz - \iint_{\mathbb{R}^2} [F_x(x, +\infty, z) - F_x(x, -\infty, z)] dx dz \right) \\ &= \mathbf{0}, \end{aligned} \quad (\text{S19})$$

with integration by parts applied. \square

THEOREM A.7. For any weight vector $\boldsymbol{\alpha} \in \mathbb{R}^d$ and any scalar function $\psi \in C^2(\mathbb{R}^d)$ that vanishes at infinity, i.e., $\lim_{\|\mathbf{x}\| \rightarrow \infty} \psi(\mathbf{x}) = 0$, the vector field $\mathbf{u}(\mathbf{x}) = -\Delta\psi(\mathbf{x}) \boldsymbol{\alpha} + \nabla \nabla^\top \psi(\mathbf{x}) \boldsymbol{\alpha}$ satisfies

$$\int_{\mathbb{R}^d} \mathbf{u}(\mathbf{x}) dV = \mathbf{0}. \quad (\text{S20})$$

Moreover, if ψ is a radial basis function, i.e. $\psi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$ for some scalar function ϕ , then the vector field also satisfies

$$\int_{\mathbb{R}^d} \mathbf{x} \times \mathbf{u}(\mathbf{x}) dV = \mathbf{0}. \quad (\text{S21})$$

PROOF. To simplify the exposition, we assume $d = 3$ without loss of generality.

According to Eq. (S14), Eq. (S20) follows directly from Lemma A.6 by substituting $F = \nabla\psi \times \alpha$. Let $\alpha = (\alpha_x, \alpha_y, \alpha_z)$. The integrand of Eq. (S21) can be divided into three parts:

$$\mathbf{x} \times \mathbf{u} = \alpha_x \mathbf{x} \times (\nabla \times (\nabla\psi \times \mathbf{e}_x)) + \alpha_y \mathbf{x} \times (\nabla \times (\nabla\psi \times \mathbf{e}_y)) + \alpha_z \mathbf{x} \times (\nabla \times (\nabla\psi \times \mathbf{e}_z)), \quad (\text{S22})$$

where \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z are unit vectors along x , y , and z axes, respectively. We only have to verify the integral of the first part (i.e. $\int_{\mathbb{R}^3} \mathbf{x} \times (\nabla \times (\nabla\psi \times \mathbf{e}_x)) dV = 0$) as the other parts are similar. Let $G(\mathbf{x}) = \mathbf{x} \times (\nabla \times (\nabla\psi \times \mathbf{e}_1))$ and $r = \|\mathbf{x}\|$. We can then formulate $G(\mathbf{x})$ in terms of partial derivatives of ψ as follows:

$$G(\mathbf{x}) = \begin{pmatrix} y \frac{\partial^2 \psi}{\partial x \partial z} - z \frac{\partial^2 \psi}{\partial x \partial y} & -z \left(\frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2} \right) - x \frac{\partial^2 \psi}{\partial x \partial z} & x \frac{\partial^2 \psi}{\partial y \partial z} + y \left(\frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2} \right) \end{pmatrix}^\top. \quad (\text{S23})$$

Since $\psi(x, y, z) = \phi(r)$, we can further calculate the partial derivatives as

$$\frac{\partial^2 \psi}{\partial y^2} = \left(\frac{1}{r} - \frac{y^2}{r^3} \right) \phi'(r) + \frac{y^2}{r^2} \phi''(r), \quad (\text{S24a})$$

$$\frac{\partial^2 \psi}{\partial z^2} = \left(\frac{1}{r} - \frac{z^2}{r^3} \right) \phi'(r) + \frac{z^2}{r^2} \phi''(r), \quad (\text{S24b})$$

$$\frac{\partial^2 \psi}{\partial x \partial y} = \frac{xy}{r^2} \phi''(r) - \frac{xy}{r^3} \phi'(r), \quad (\text{S24c})$$

$$\frac{\partial^2 \psi}{\partial x \partial z} = \frac{xz}{r^2} \phi''(r) - \frac{xz}{r^3} \phi'(r). \quad (\text{S24d})$$

Substituting these expressions into Eq. (S23), we obtain

$$G(x, y, z) = \begin{pmatrix} 0 & -\frac{z}{r} \phi'(r) - z \phi''(r) & \frac{y}{r} \phi'(r) + y \phi''(r) \end{pmatrix}^\top, \quad (\text{S25})$$

which immediately implies that $G(x, 0, 0) = \mathbf{0}$ and $G(x, y, z) + G(x, -y, -z) = \mathbf{0}$. Therefore, the integral can be rewritten as

$$\int_{\{(x,y,z) \in \mathbb{R}^3 : y > 0 \vee (y=0 \wedge z>0)\}} [G(x, y, z) + G(x, -y, -z)] dV = \mathbf{0}, \quad (\text{S26})$$

which proves Eq. (S21). \square

In the fluid domain of a homogeneous incompressible flow, the density field is constant throughout space and time, which we denote as ρ . The linear momentum of the velocity field $\mathbf{u}(\mathbf{x})$ is then $\int_{\mathbb{R}^d} \rho \mathbf{u} dV = \rho \int_{\mathbb{R}^d} \mathbf{u} dV = \mathbf{0}$ according to Eq. (S20). Taking an arbitrary reference point \mathbf{x}_0 , the angular momentum is $\int_{\mathbb{R}^d} (\mathbf{x} - \mathbf{x}_0) \times (\rho \mathbf{u}) dV = \rho \int_{\mathbb{R}^d} (\mathbf{x} \times \mathbf{u}) dV - \rho \mathbf{x}_0 \times \int_{\mathbb{R}^d} \mathbf{u} dV = \mathbf{0}$ according to Eqs. (S21) and (S20).

Furthermore, since the proof of Eq. (S20) is independent of the specific kernel formulation in Eq. (S14), a stronger conclusion can be drawn.

THEOREM A.8. *For any vector field $\mathbf{u}(\mathbf{x})$ that is divergence-free (i.e., $\nabla \cdot \mathbf{u}(\mathbf{x}) = 0$) and vanishes at infinity (i.e., $\lim_{\|\mathbf{x}\| \rightarrow \infty} \mathbf{u}(\mathbf{x}) = \mathbf{0}$), the integral of \mathbf{u} over the entire space is zero:*

$$\int_{\mathbb{R}^d} \mathbf{u}(\mathbf{x}) dV = \mathbf{0}. \quad (\text{S27})$$

PROOF. Take $d = 3$ as an example.

Following an approach similar to that used in vortex methods [Bridson 2015], we construct a vector field $F(\mathbf{x}) \in \mathbb{R}^3$ as

$$F(\mathbf{x}) = - \iiint_{\mathbb{R}^3} \frac{\nabla \times \mathbf{u}(\mathbf{x}')}{4\pi \|\mathbf{x} - \mathbf{x}'\|} dV'. \quad (\text{S28})$$

It is straightforward to verify that $\mathbf{u} = \nabla \times \mathbf{F}$ and that $\lim_{\|\mathbf{x}\| \rightarrow \infty} \mathbf{F}(\mathbf{x}) = \mathbf{0}$. Substituting $\mathbf{F}(\mathbf{x})$ into Lemma A.6 completes the proof. \square

In the context of a homogeneous incompressible flow, Theorem A.8 guarantees that any compactly supported, divergence-free kernel must exhibit zero net linear momentum.

B ANALYTIC EXPRESSIONS OF FUNCTIONS AND DERIVATIVES

For practical applications, we provide the analytical forms of matrix-valued radial basis functions and their derivatives, including both the general form and the specialized version for DFKs-Wen4.

B.1 Original Functions

For any radial function $\psi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$, the Laplacian $\Delta\psi$ can be expressed as

$$\Delta\psi(\mathbf{x}) = \Delta\phi(r) = \frac{d^2\phi}{dr^2} + \frac{d-1}{r} \frac{d\phi}{dr}. \quad (\text{S29})$$

Similarly, the Hessian of an Radial function can be expanded as

$$\nabla^\top \nabla \psi(\mathbf{x}) = \nabla^\top \nabla \phi(r) = \nabla^\top \left[\frac{d\phi}{dr}(r) \frac{\mathbf{x}}{r} \right] = \frac{1}{r} \frac{d\phi}{dr} \mathbf{I} + \frac{1}{r^2} \left(\frac{d^2\phi}{dr^2} - \frac{1}{r} \frac{d\phi}{dr} \right) \mathbf{x} \mathbf{x}^\top. \quad (\text{S30})$$

For the C^4 -continuous Wendland polynomial used in the DFK-Wen4 method, the result of applying the negative Laplacian is given by

$$-\Delta R_{\text{Wen4}}(r) = (1-r)^4 [d + 4dr - (5d + 30)r^2]. \quad (\text{S31})$$

The corresponding curl-free RBF takes the form

$$-\nabla^\top \nabla R_{\text{Wen4}}(r) = (1-r)^4 [(1+4r-5r^2) \mathbf{I} - 30 \mathbf{x} \mathbf{x}^\top], \quad (\text{S32})$$

and the corresponding divergence-free RBF is given by

$$-\Delta R_{\text{Wen4}}(r) \mathbf{I} + \nabla^\top \nabla R_{\text{Wen4}}(r) = (1-r)^4 \{[(d-1)(1+4r) - 5(d+5)r^2] \mathbf{I} + 30 \mathbf{x} \mathbf{x}^\top\}. \quad (\text{S33})$$

B.2 Gradients

Consider a matrix-valued RBF interpolation, where each kernel contributes in the form

$$\mathbf{u}(\mathbf{x}) = f(r) \boldsymbol{\alpha} + g(r) (\mathbf{x} \cdot \boldsymbol{\alpha}) \mathbf{x}, \quad (\text{S34})$$

which, in index notation, can be written as

$$u_i = f \alpha_i + g x_k \alpha_k x_i. \quad (\text{S35})$$

Taking the derivative with respect to x_j yields

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial f}{\partial x_j} \alpha_i + \frac{\partial g}{\partial x_j} x_k \alpha_k x_i + g \alpha_j x_i + g x_k \alpha_k \delta_{ij}, \quad (\text{S36})$$

which, in matrix form, can be expressed as

$$\begin{aligned} \nabla \mathbf{u} &= \boldsymbol{\alpha} (\nabla f)^\top + (\mathbf{x} \cdot \boldsymbol{\alpha}) \mathbf{x} (\nabla g)^\top + g \mathbf{x} \boldsymbol{\alpha}^\top + g (\mathbf{x} \cdot \boldsymbol{\alpha}) \mathbf{I} \\ &= \frac{f'(r)}{r} \boldsymbol{\alpha} \mathbf{x}^\top + \frac{g'(r)}{r} (\mathbf{x} \cdot \boldsymbol{\alpha}) \mathbf{x} \mathbf{x}^\top + g \mathbf{x} \boldsymbol{\alpha}^\top + g (\mathbf{x} \cdot \boldsymbol{\alpha}) \mathbf{I}. \end{aligned} \quad (\text{S37})$$

For the divergence-free matrix-valued RBF derived from R_{Wen4} , the coefficients in two dimensions are given by

$$f'(r) = 30r(1-r)^3(7r-3), \quad g'(r) = -120(1-r)^3, \quad g(r) = 30(1-r)^4. \quad (\text{S38})$$

In three dimensions, they take the form

$$f'(r) = 120r(1-r)^3(2r-1), \quad g'(r) = -120(1-r)^3, \quad g(r) = 30(1-r)^4. \quad (\text{S39})$$

Backpropagation. Consider a scalar loss function \mathcal{L} and define $t_{ij} = \partial u_i / \partial x_j$. Then, the gradient of the loss with respect to α_k is given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \alpha_k} &= \frac{\partial \mathcal{L}}{\partial t_{ij}} \frac{\partial t_{ij}}{\partial \alpha_k} = \frac{\partial \mathcal{L}}{\partial t_{ij}} \left[\frac{\partial f}{\partial x_j} \delta_{ik} + \frac{\partial g}{\partial x_j} x_k x_i + g x_i \delta_{jk} + g x_k \delta_{ij} \right] \\ &= \frac{\partial \mathcal{L}}{\partial t_{kj}} \frac{\partial f}{\partial x_j} + \frac{\partial \mathcal{L}}{\partial t_{ij}} \frac{\partial g}{\partial x_j} x_k x_i + \frac{\partial \mathcal{L}}{\partial t_{ik}} g x_i + \frac{\partial \mathcal{L}}{\partial t_{ii}} g x_k. \end{aligned} \quad (\text{S40})$$

In matrix form, this can be expressed as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} &= S(\nabla f) + \mathbf{x}^\top S(\nabla g)\mathbf{x} + g S^\top \mathbf{x} + g(\text{tr } S)\mathbf{x} \\ &= \frac{f'(r)}{r} S\mathbf{x} + \frac{g'(r)}{r} (\mathbf{x}^\top S\mathbf{x})\mathbf{x} + g S^\top \mathbf{x} + g(\text{tr } S)\mathbf{x}, \end{aligned} \quad (\text{S41})$$

where $S = \{s_{ij}\} = \{\partial \mathcal{L} / \partial t_{ij}\}$.

B.3 Curls

Physically, the curl of a vector field corresponds to its vorticity. Consider an RBF interpolation where the contribution of each kernel takes the form

$$\mathbf{u}(\mathbf{x}) = h(r) \boldsymbol{\alpha}, \quad (\text{S42})$$

which, in index notation, can be expressed as

$$u_i = h \alpha_i. \quad (\text{S43})$$

Taking the curl of \mathbf{u} , we obtain

$$\omega_i = (\nabla \times \mathbf{u})_i = \frac{\partial h}{\partial x_j} \alpha_k \epsilon_{ijk} = [(\nabla h) \times \boldsymbol{\alpha}]_i, \quad (\text{S44})$$

which simplifies to

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} = \frac{h'(r)}{r} \mathbf{x} \times \boldsymbol{\alpha}. \quad (\text{S45})$$

For the divergence-free matrix-valued RBF derived from R_{Wen4} (where the irrotational component automatically vanishes), the function $h'(r)$ is given by

$$h'(r) = 120r(1-r)^3(2r-1), \quad (\text{S46})$$

in 2D, which in 3D takes the form

$$h'(r) = 30r(1-r)^3(9r-5). \quad (\text{S47})$$

Backpropagation. Consider a scalar loss function \mathcal{L} . The gradient of the loss with respect to α_k is given by

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = \frac{\partial \mathcal{L}}{\partial \omega_i} \frac{\partial \omega_i}{\partial \alpha_k} = \frac{\partial \mathcal{L}}{\partial \omega_i} \frac{\partial h}{\partial x_j} \epsilon_{ijk}. \quad (\text{S48})$$

In matrix form, this can be expressed as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = \frac{h'(r)}{r} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}} \times \mathbf{x}. \quad (\text{S49})$$

B.4 Gradients of Curls

Continuing from the previous subsection, we differentiate the vorticity with respect to x_j . In index notation, this yields

$$\frac{\partial \omega_i}{\partial x_j} = \frac{\partial^2 h}{\partial j \partial m} \alpha_n \epsilon_{imn} = \left[\frac{h'(r)}{r} \delta_{jm} + \frac{1}{r^2} \left(h''(r) - \frac{h'(r)}{r} \right) x_j x_m \right] \alpha_n \epsilon_{imn}, \quad (\text{S50})$$

which can be written in matrix form as

$$\nabla \boldsymbol{\omega} = \frac{h'(r)}{r} \mathbf{A} + \frac{1}{r^2} \left(h''(r) - \frac{h'(r)}{r} \right) (\mathbf{x} \times \boldsymbol{\alpha}) \mathbf{x}^\top, \quad (\text{S51})$$

where

$$\mathbf{A} = \begin{pmatrix} 0 & \alpha_3 & -\alpha_2 \\ -\alpha_3 & 0 & \alpha_1 \\ \alpha_2 & -\alpha_1 & 0 \end{pmatrix}. \quad (\text{S52})$$

For the divergence-free matrix-valued RBF derived from R_{Wen4} , in three dimensions, the coefficient of the second term is

$$\frac{1}{r^2} \left(h''(r) - \frac{h'(r)}{r} \right) = 360 \frac{(1-r)^2(2-3r)}{r}. \quad (\text{S53})$$

Backpropagation. Consider the backpropagation process with a scalar loss function \mathcal{L} , and let $t_{ij} = \partial \omega_i / \partial x_j$ and $s_{ij} = \partial \mathcal{L} / \partial t_{ij}$. The derivative of the loss function with respect to α_k is given by

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = s_{ij} \frac{\partial t_{ij}}{\partial \alpha_k} = \left[\frac{h'(r)}{r} s_{im} + \frac{1}{r^2} \left(h''(r) - \frac{h'(r)}{r} \right) s_{ij} x_j x_m \right] \epsilon_{imk}. \quad (\text{S54})$$

In matrix form, this becomes

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = \frac{h'(r)}{r} \begin{pmatrix} s_{23} - s_{32} \\ s_{31} - s_{13} \\ s_{12} - s_{21} \end{pmatrix} + \frac{1}{r^2} \left(h''(r) - \frac{h'(r)}{r} \right) (\mathbf{Sx}) \times \mathbf{x}. \quad (\text{S55})$$

C ADDITIONAL DESCRIPTIONS OF EXPERIMENTS

C.1 Fitting

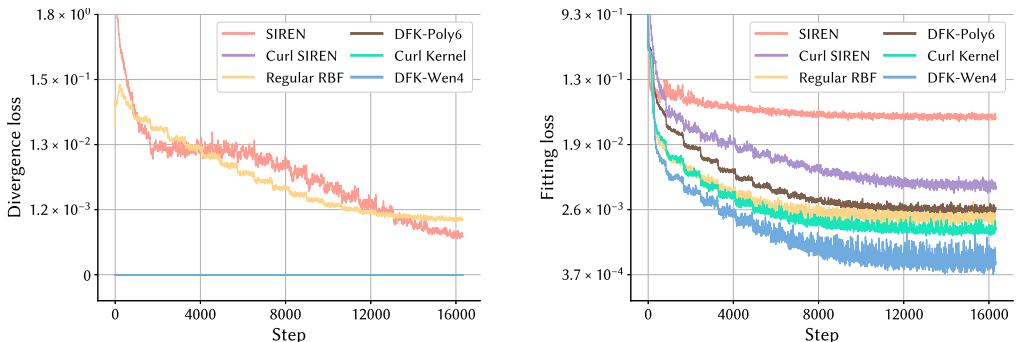


Fig. S1. Loss curves of fitting experiments for *Kármán vortex street* (2D). Divergence losses of Curl SIREN, DFK-Poly6, Curl Kernel, and DFK-Wen4 are always zero due to the intrinsic properties of representations.

Kármán vortex street (2D). SIREN consists of four hidden layers with 256 neurons each, totaling 198 658 trainable parameters. Curl SIREN consists of four hidden layers with 256 neurons each, totaling 198 401 trainable parameters. Regular RBF comprises 5367 points, resulting in 26 835 trainable parameters. DFK-Poly6 comprises 5367 points, resulting in 26 835 trainable parameters. Curl Kernel comprises 6794 points, resulting in 27 176 trainable parameters. DFK-Wen4 comprises 5367 points, resulting in 26 835 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 9$. The loss curves are illustrated in Fig. S1.

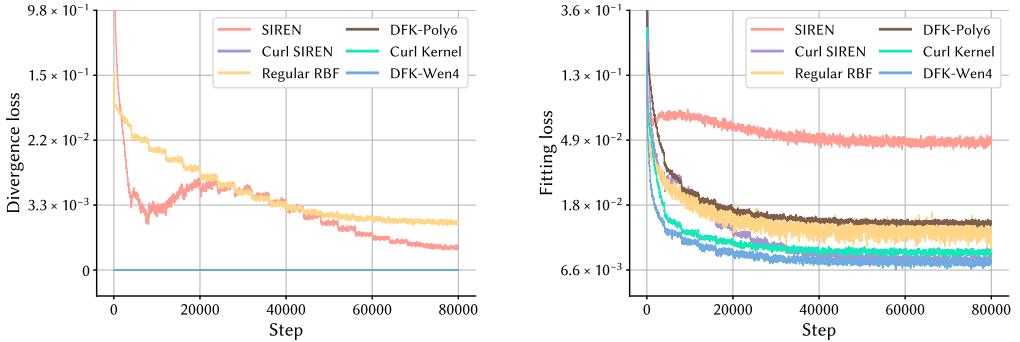


Fig. S2. Loss curves of fitting experiments for *analytic vortices (3D)*. Divergence losses of Curl SIREN, DFK-Poly6, Curl Kernel, and DFK-Wen4 are always zero due to the intrinsic properties of representations.

Analytic vortices (3D). SIREN consists of four hidden layers with 256 neurons each, totaling 199 171 trainable parameters. Curl SIREN consists of four hidden layers with 256 neurons each, totaling 199 171 trainable parameters. Regular RBF comprises 21 117 points, resulting in 147 819 trainable parameters. DFK-Poly6 comprises 21 117 points, resulting in 147 819 trainable parameters. Curl Kernel comprises 21 117 points, resulting in 147 819 trainable parameters. DFK-Wen4 comprises 21 117 points, resulting in 147 819 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} for NNs and 5×10^{-4} for kernels. When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S2.

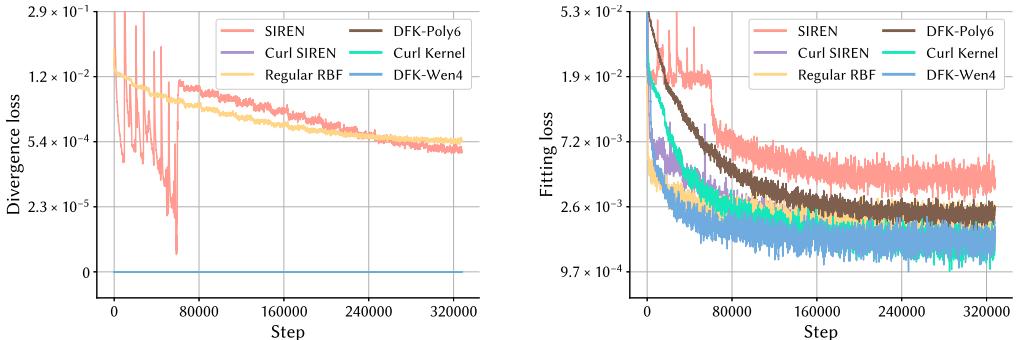


Fig. S3. Loss curves of fitting experiments for *simple plume (3D)*. Divergence losses of Curl SIREN, DFK-Poly6, Curl Kernel, and DFK-Wen4 are always zero due to the intrinsic properties of representations.

Simple plume (3D). SIREN consists of six hidden layers with 256 neurons each, totaling 330 755 trainable parameters. Curl SIREN consists of six hidden layers with 256 neurons each, totaling 330 755 trainable parameters. Regular RBF comprises 42 002 points, resulting in 294 014 trainable parameters. DFK-Poly6 comprises 42 002 points, resulting in 294 014 trainable parameters. Curl Kernel comprises 42 002 points, resulting in 294 014 trainable parameters. DFK-Wen4 comprises 42 002 points, resulting in 294 014 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} for NNs and 1×10^{-4} for kernels. When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S3.

C.2 Projection

For the projection task, Curl SIREN is combined with Gradient SIREN, which shares the same architecture, to fit the data, while the two models remain independent. Similarly, Curl Kernel and Gradient Kernel share the same point positions and radii but have independent weights.

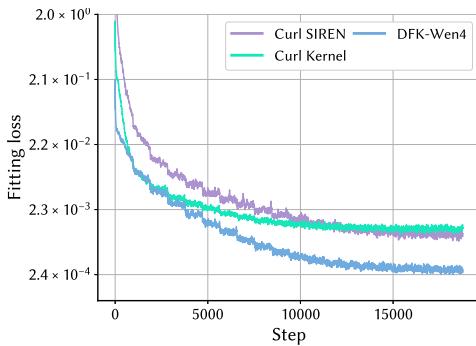


Fig. S4. Loss curves of projection experiments for *Taylor vortex (2D)*.

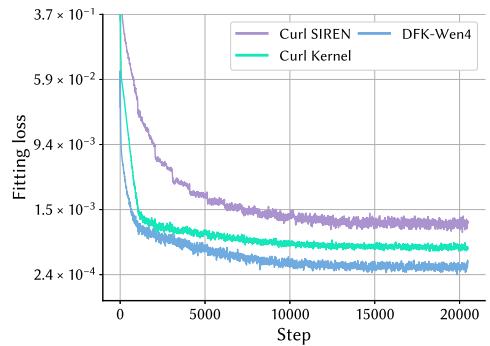


Fig. S5. Loss curves of projection experiments for *vortex ring collision (3D)*.

Taylor vortex (2D). Curl SIREN (as well as Gradient SIREN) consists of four hidden layers with 256 neurons each, totaling 396 802 trainable parameters. Curl Kernel (as well as Gradient Kernel) comprises 5416 points, resulting in 27 080 trainable parameters. DFK-Wen4 comprises 5416 points, resulting in 27 080 trainable parameters.

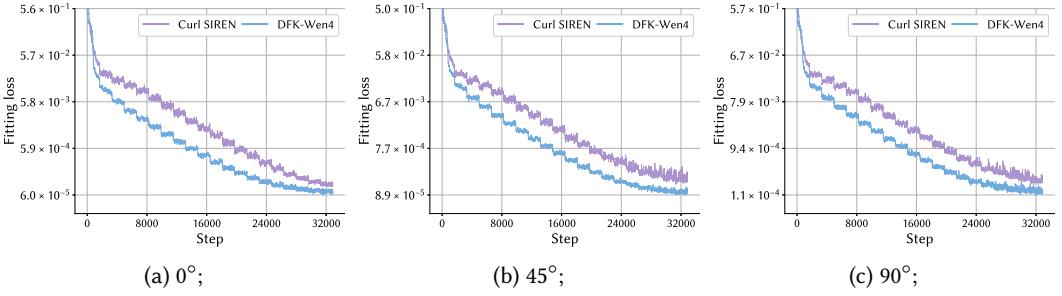
We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 27$. The loss curves are illustrated in Fig. S4.

Vortex ring collision (3D). Curl SIREN (as well as Gradient SIREN) consists of four hidden layers with 256 neurons each, totaling 397 828 trainable parameters. Curl Kernel (as well as Gradient Kernel) comprises 8544 points, resulting in 68 352 trainable parameters. DFK-Wen4 comprises 8544 points, resulting in 59 808 trainable parameters.

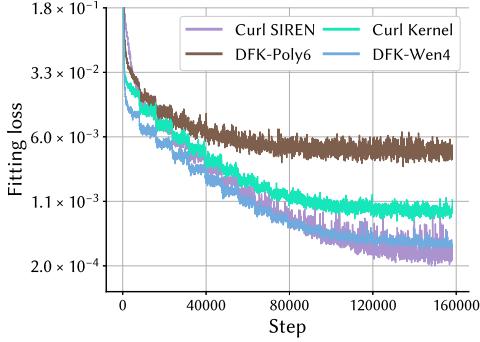
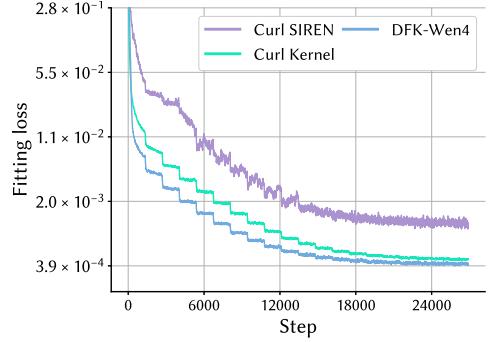
We set the batch size to 2048 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 12$. The loss curves are illustrated in Fig. S5.

C.3 Inpainting

Laminar flows (2D). Curl SIREN consists of four hidden layers with 256 neurons each, totaling 198 401 trainable parameters. DFK-Wen4 comprises 5416 points, resulting in 27 080 trainable parameters.

Fig. S6. Loss curves of inpainting experiments for *laminar flows* (2D).

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 9$. The loss curves are illustrated in Fig. S6.

Fig. S7. Loss curves of inpainting experiments for *missile* (2D).Fig. S8. Loss curves of inpainting experiments for *bullet* (3D).

Missile (2D). Curl SIREN consists of three hidden layers with 128 neurons each, totaling 33 537 trainable parameters. DFK-Poly6 comprises 5390 points, resulting in 26 950 trainable parameters. Curl Kernel comprises 6797 points, resulting in 27 188 trainable parameters. DFK-Wen4 comprises 5390 points, resulting in 26 950 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 9$. The loss curves are illustrated in Fig. S7.

Bullet (3D). Curl SIREN consists of four hidden layers with 256 neurons each, totaling 199 171 trainable parameters. Curl Kernel comprises 25 325 points, resulting in 177 275 trainable parameters. DFK-Wen4 comprises 25 325 points, resulting in 177 275 trainable parameters.

We set the batch size to 4096 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S8.

C.4 Super-Resolution

Turbulence A (2D). Curl SIREN consists of four hidden layers with 256 neurons each, totaling 198 401 trainable parameters. DFK-Poly6 comprises 5416 points, resulting in 27 080 trainable parameters. Curl Kernel comprises 6834 points, resulting in 27 336 trainable parameters. DFK-Wen4 comprises 5416 points, resulting in 27 080 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 9$. The loss curves are illustrated in Fig. S9a.

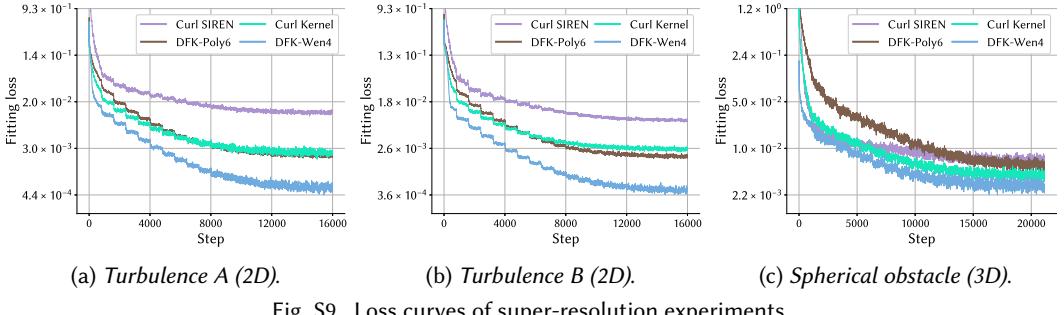


Fig. S9. Loss curves of super-resolution experiments.

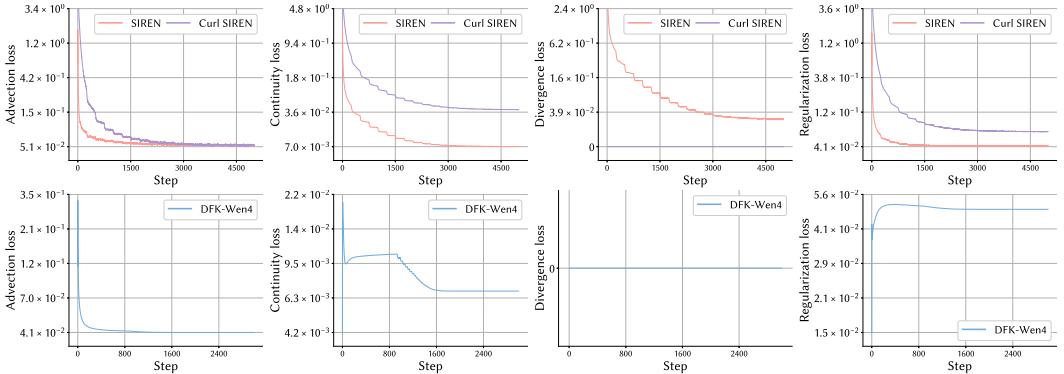
Turbulence B (2D). Curl SIREN consists of four hidden layers with 256 neurons each, totaling 198 401 trainable parameters. DFK-Poly6 comprises 5416 points, resulting in 27 080 trainable parameters. Curl Kernel comprises 6834 points, resulting in 27 336 trainable parameters. DFK-Wen4 comprises 5416 points, resulting in 27 080 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 9$. The loss curves are illustrated in Fig. S9b.

Spherical obstacle (3D). Curl SIREN consists of six hidden layers with 256 neurons each, totaling 330 755 trainable parameters. DFK-Poly6 comprises 42 002 points, resulting in 294 014 trainable parameters. Curl Kernel comprises 42 002 points, resulting in 294 014 trainable parameters. DFK-Wen4 comprises 42 002 points, resulting in 294 014 trainable parameters.

We set the batch size to 128 and trained each model for 20 epochs, with an initial learning rate of 1×10^{-3} . When initializing the kernel radii, we set $\eta = 12$. The loss curves are illustrated in Fig. S9c.

C.5 Inference

Fig. S10. Loss curves of inference experiments for *rising (3D)*.

Rising (3D). SIREN consists of four hidden layers with 256 neurons each, totaling 29 676 479 trainable parameters. Curl SIREN consists of four hidden layers with 256 neurons each, totaling 29 676 479 trainable parameters. DFK-Wen4 comprises 26 375 points, resulting in 11 868 750 trainable parameters.

We set the batch size to 3072×151 and trained each NN model for 20 epochs, with an initial learning rate of 1×10^{-3} . We trained DFKs-Wen4 using batch gradient descent for 1750 epochs,

with an initial learning rate of 1×10^{-2} . When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S10.

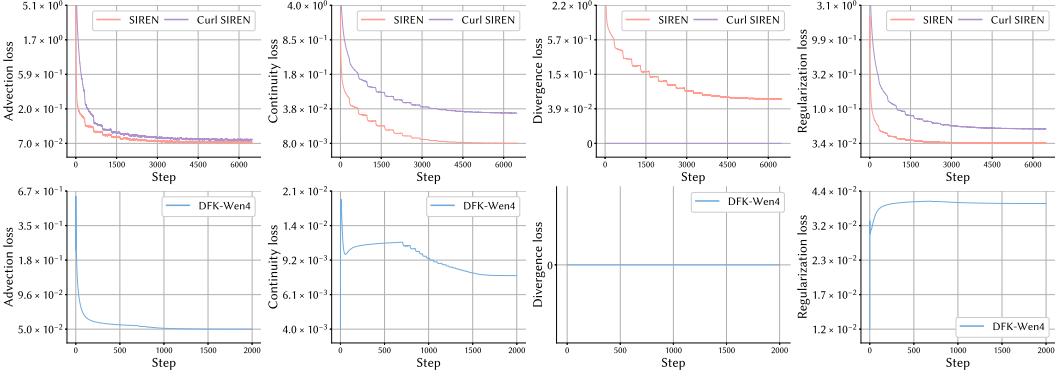


Fig. S11. Loss curves of inference experiments for *teapot* (3D).

Teapot (3D). SIREN consists of four hidden layers with 256 neurons each, totaling 19 917 100 trainable parameters. Curl SIREN consists of four hidden layers with 256 neurons each, totaling 19 917 100 trainable parameters. DFK-Wen4 comprises 42 053 points, resulting in 12 615 900 trainable parameters.

We set the batch size to 4096×101 and trained each NN model for 20 epochs, with an initial learning rate of 1×10^{-3} . We trained DFKs-Wen4 using batch gradient descent for 2000 epochs, with an initial learning rate of 1×10^{-2} . When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S11.

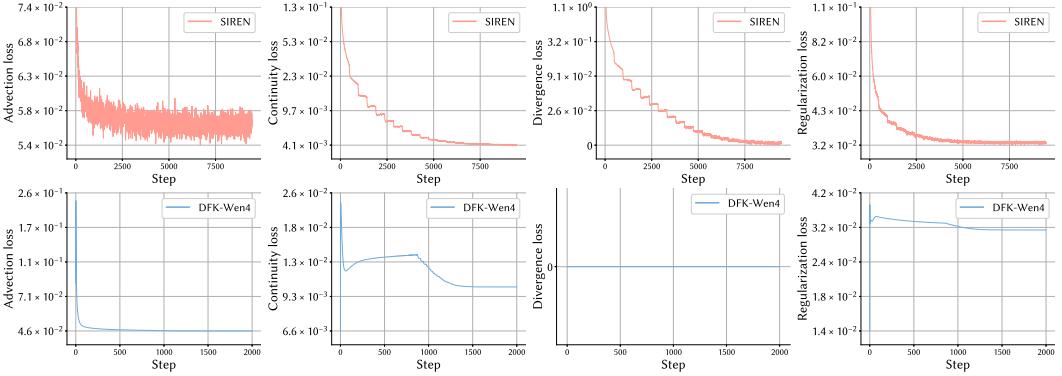


Fig. S12. Loss curves of inference experiments for *scalar flow* (3D).

Scalar flow (3D). SIREN consists of four hidden layers with 256 neurons each, totaling 23 701 349 trainable parameters. DFK-Wen4 comprises 41 959 points, resulting in 14 979 363 trainable parameters.

We set the batch size to 4096×120 and trained the NN model for 20 epochs, with an initial learning rate of 1×10^{-3} . We trained DFKs-Wen4 using batch gradient descent for 2000 epochs, with an initial learning rate of 1×10^{-2} . When initializing the kernel radii, we set $\eta = 6$. The loss curves are illustrated in Fig. S12.

REFERENCES

- Robert Bridson. 2015. *Fluid simulation for computer graphics* (2nd ed.). AK Peters/CRC Press, Boca Raton, FL, USA.
- Francis J. Narcowich and Joseph D. Ward. 1994. Generalized Hermite interpolation via matrix-valued conditionally positive definite functions. *Math. Comput.* 63, 208 (Oct. 1994), 661–687.
- Holger Wendland. 2004. *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK.
- Holger Wendland. 2009. Divergence-Free Kernel Methods for Approximating the Stokes Problem. *SIAM J. Numer. Anal.* 47, 4 (2009), 3158–3179.