Laplacian Projection Based Global Physical Prior Smoke Reconstruction

Shibang Xiao, Chao Tong, Qifan Zhang, Yunchi Cen, Frederick W. B. Li and Xiaohui Liang

***Abstract*—We present a novel framework for reconstructing fluid dynamics in real-life scenarios. Our approach leverages sparse view images and incorporates physical priors across long series of frames, resulting in reconstructed fluids with enhanced physical consistency. Unlike previous methods, we utilize a differentiable fluid simulator (DFS) and a differentiable renderer (DR) to exploit global physical priors, reducing reconstruction errors without the need for manual regularization coefficients. We introduce divergence-free Laplacian eigenfunctions (div-free LE) as velocity bases, improving computational efficiency and memory usage. By employing gradient-related strategies, we achieve better convergence and superior results. Extensive experiments demonstrate the effectiveness of our method, showcasing improved reconstruction quality and computational efficiency compared to existing approaches. We validate our approach using both synthetic and real data, highlighting its practical potential.**

I. Introduction

Fluids play a crucial role in various graphics applications, such as virtual fire drills, where realistic fluid behavior significantly enhances the simulation experience. However, capturing and reproducing the complex motions of natural fluids through numerical simulation is challenging. As an alternative approach, fluid capturing aims to reconstruct fluid properties, including advection density and velocity, directly from real-world observations. This reconstruction problem has attracted significant attention in the fields of computer graphics and computer vision due to its practical implications. By reconstructing fluid flows from real-world data, various applications can be facilitated, including guiding [11], [36], control [10], stylization [22], and VR/AR applications.

Fluid flow reconstruction from multi-view videos has been extensively studied [6], [9], [12], [14], [19], [33], [42], [45]. However, the reconstruction process faces challenges due to the complexity of the setup and calibration, resulting in a limited number of available views. As a consequence, the fluid reconstruction problem becomes under-determined, making it difficult to obtain natural and realistic flow results. Previous approaches [6], [9], [12], [24], [45], [48] have attempted to improve reconstruction by incorporating physical priors, such as Navier-Stokes equations and density advection. However, these methods have primarily relied on temporally local physical priors or simplified global physical priors, which may result in the loss of global information and pose challenges in achieving the global optimum during the optimization process. In this work, we propose a novel fully differentiable framework for fluid reconstruction that maximizes the utilization of physical priors. Our framework integrates a differentiable renderer, which establishes the connection between images and 3D fluid fields, and a differentiable fluid simulator, which encompasses the entire sequence of fluid fields. By leveraging the adjoint method, we efficiently compute all necessary derivatives in a single backward pass, enabling seamless integration with gradient-based optimization algorithms.

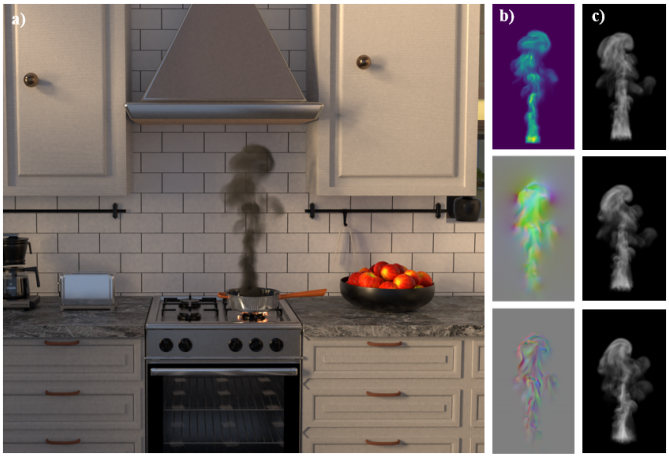


Fig. 1. a) Reconstruction of smoke beyond the charred pot in the real world captured in a video. Our algorithm reconstructs the density field and velocity field, and re-renders it in a virtual environment. b) Visualization of the reconstructed smoke, showing the density, velocity, and vorticity fields. c) Multiple views of the reconstructed smoke from different perspectives.

Differentiable physics methods have gained significant attention in inversion and control problems, enabling efficient computation of gradients on physical simulation sequences that can be optimized using gradient-based methods [15], [26]. However, incorporating differentiable physics into simulations is typically more complex than forward simulation. Some methods [3], [27] utilize automatic differentiation frameworks, which offer generality but come with high computational costs and memory overhead. In certain cases, the adjoint method is employed to design high-level automatic differentiation to improve efficiency. However, adjoint-based differentiable fluid simulation encounters two challenges: (1) accuracy of velocity derivatives relies on the precision of the linear solver in the

Manuscript created October, 2023; This work was supported by the National Natural Science Foundation of China (Project Number: 62272019). (Corresponding author: Xiaohui Liang)

Shibang Xiao, Chao Tong, Qifan Zhang and Yunchi Cen are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China (e-mail:by2106156, tongchao, zhangqf_m, cenyc@buaa.edu.cn).

Frederick W. B. Li is with the Department of Computer Science, University of Durham, Durham, UK (e-mail: frederick.li@durham.ac.uk).

Xiaohui Liang is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, and also with Zhongguancun Laboratory, Beijing, China (e-mail: liang xiaohui@buaa.edu.cn).

pressure projection step, and (2) storing fluid fields for the entire sequence to compute backward derivatives demands substantial memory resources. To address these challenges, we propose using divergence-free Laplacian eigenfunctions (div-free LE) as bases for a three-dimensional vector space, maintaining divergence-free behavior while respecting boundary conditions. We replace the low tolerance pressure projection with direct projection of the velocity field onto div-free LE, ensuring continuous divergence-free behavior with enhanced efficiency. We also leverage the dimensionality reduction capabilities of div-free LE and implement checkpoint techniques to achieve significant memory savings during the simulation process.

While the adjoint method ensures mathematically correct derivatives, its direct utilization may not always yield satisfactory results. To enhance reconstruction outcomes, we propose three gradient-related strategies within the differentiable framework. Our novel gradient-related visual hull application strategy eliminates the need for extra regularization terms. Additionally, a spectral domain multi-scale strategy optimizes large-scale structures (lower frequency components) first, leveraging their importance in overall reconstruction. We further employ a temporal gradient exponential decay strategy to prevent excessive gradient accumulation and stabilize the optimization process. Previously, we introduced a small portion of this work as an extended abstract [47], demonstrating the proposed system concept through exemplary experiments. Now, we present several novel methods, significantly enhancing the efficiency and accuracy of fluid simulation and rendering.

Introducing Laplacian eigenfunctions into differentiable fluid simulation, we have two primary objectives. First, we aim to address the trade-off between derivative precision and runtime in the divergence-free projection phase of the velocity field. Second, we seek to tackle the issue of excessive memory consumption during the simulation process. This is crucial for us, as the realization of global physical priors relies on the performance of the underlying differentiable fluid simulator. Utilizing manually pre-derived derivative functions, rather than automatic differentiation, enables faster computations compared to previous approaches. Addressing limitations in applying Neumann boundary conditions to divergence-free Laplacian eigenfunctions (div-free LE), we introduce a new boundary processing method. For fluid reconstruction with solid objects, we leverage a solid mask within the basis spectral method. Lastly, compensation correction techniques ensure single-scatter rendering nearly matches multi-scatter accuracy, resulting in highly realistic reconstructed fluid visualizations. These methods collectively represent substantial advancements in fluid simulation and rendering. Source code will be available at https://github.com/LAST-iMP/LPBGPP-Smoke-Reconstruction. Our main contributions are:

- We propose a differentiable framework that integrates a differentiable fluid simulator and a differentiable renderer, enabling efficient computation of gradients and optimization of fluid properties for improved fluid reconstruction.
- We introduce a set of divergence-free velocity bases using divergence-free Laplacian eigenfunctions, enhancing the efficiency of differentiable fluid simulation and reducing memory usage.
- We present three gradient-related strategies, namely the visual hull application strategy, spectral domain multi-scale strategy, and temporal gradient exponential decay strategy, which effectively improve the quality of reconstruction results.

II. Related Work

We now present an overview of existing research in fluid reconstruction, differentiable fluid simulation, and differentiable rendering, which are directly relevant to our proposed framework for fluid reconstruction. By exploring the advancements in these areas, we can identify the gaps in the literature and highlight the unique contributions of our approach.

A. Fluid Reconstruction

Methods have been proposed to reconstruct fluid motions and passive advected quantities based on real-world observations. While some techniques focus on reconstructing invisible fluid motions that require specialized hardware or setups for visualization, such as particle imaging velocimetry (PIV) methods [44] for water flows or background-oriented schlieren imaging (BOS) methods [2] and light path methods [20] for hot gas flows, our focus is alternatively on capturing visible and visually appealing fluid phenomena like smoke and flame, which can be directly captured by cameras. However, the limited number of available views in fluid capturing poses significant challenges, leading to under-determined optimization problems. This sparsity is a consequence of practical limitations, including restricted camera setups, limited access to fluid environments, and computational constraints, making it challenging to deploy a large number of cameras, especially in hazardous or inaccessible locations, thereby adding complexity to data processing. To address the challenges associated with sparse views, specialized techniques and priors are employed to achieve accurate fluid reconstructions and mitigate the under-determined nature of the problem. One common approach involves considering the smoothness of the reconstructed density as a prior, with methods such as limiting the number of optimization iterations for a smoother solution [19] or directly incorporating a smooth regularization term into the optimization process [14]. In our work, we employ a spectral domain multi-scale strategy as a means of achieving smoothness regularization, thereby enhancing the quality of fluid reconstructions.

Various priors have been explored in the field of fluid reconstruction. One type of prior focuses on generating novel view images by transferring appearance information from input images [33] or estimating it through interpolation [45]. While these priors can produce more realistic results with fewer input views, they may introduce conflicts and mismatches between the constraints imposed on the novel views and the input views. Another category of priors involves enforcing temporal constraints among frames. Some methods couple linearized density advection with the divergence-free condition to derive convex optimization problems for single-view fluid reconstruction [8], and further improve efficiency and inflow estimation in multi-view scenarios [9]. However,

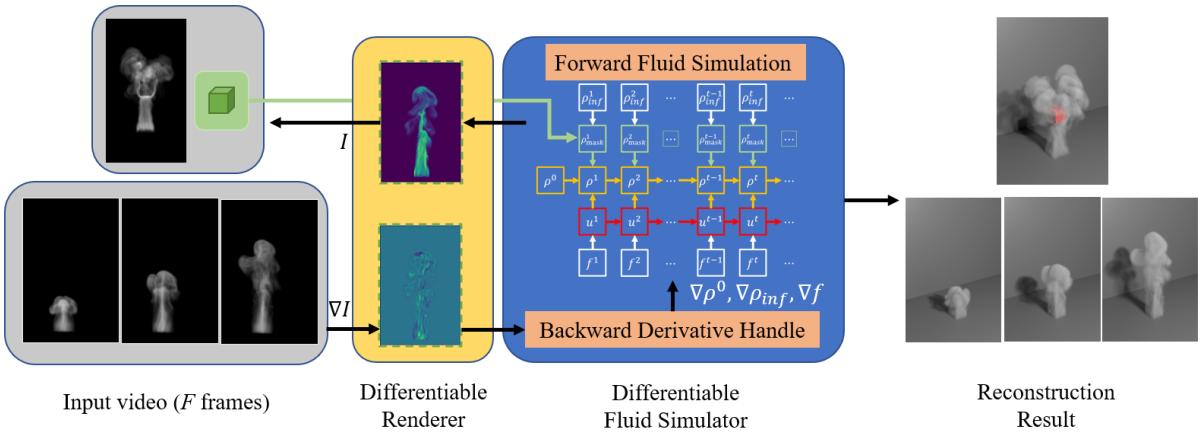


Fig. 2. Our proposed algorithm reconstructs the density $\boldsymbol{\rho}$ and velocity $\mathbf{u}$ of target fluid from F frames of an input sequence. Instead of solely passing derivatives between frames, we track the gradients of both density and velocity throughout the simulation and rendering process, leveraging them to optimize the reconstruction result.

these approaches reconstruct fluids frame by frame, relying solely on the information from previous frames, which may limit temporal coherence. To address this limitation, a global transport approach has been proposed to connect density fields of all frames and allow gradients to propagate backward [12]. Nevertheless, the independent treatment of velocity fields in this method prevents gradient propagation through them, affecting temporal consistency. In our work, we introduce a fully differentiable reconstruction framework that seamlessly integrates temporal information, overcoming the challenges of temporal coherence and gradient propagation. Additionally, some studies combine deep learning with physical priors to accelerate fluid reconstruction [35] or handle unknown lighting conditions and static obstacles [6], leveraging the power of deep learning to improve efficiency and robustness. By addressing these challenges and exploring the integration of deep learning and physical priors, significant advancements can be made in the field of fluid reconstruction.

B. Differentiable Fluid Simulation

Differentiable fluid simulation has primarily been explored in the context of key frame-based fluid control problems, where the objective is to compute control force fields at each frame to match desired fluid states specified by sparse key frames. Previous approaches, such as the work by Treuille et al. [41], relied on forward gradient computation, which is computationally expensive as it requires a full simulation to compute the derivative of each control parameter. McNamara et al. [32] improved upon this by using the adjoint method to compute gradients backward, enabling efficient computation of all necessary gradients in a single pass. However, both methods represented control forces using predefined force patterns, sacrificing accuracy for efficiency.

To address the challenges posed by high-dimensional control forces and the lack of constraints, researchers explored dimensionality reduction methods [7], [18], [23], [34], [43]. For instance, T. D. Witt et al. [43] defined global functions over the entire simulation domain to represent vorticity and velocity. Based on this, they proposed an algorithm for simulating incompressible fluid phenomena, achieving high computational efficiency. In recent studies, Tang et al. [39] proposed a multi-scale approach in the spectral domain using the Fourier transform and enforced strict divergence-free control forces through stream functions. While their approach shares similarities with ours, they employed stream functions as base functions, with the vorticity of the stream function serving as the external force field. This led to increased noise and artifacts due to the optimization process focusing on high-frequency components. In contrast, our method utilizes only the divergence-free Laplacian eigenfunctions as bases, avoiding the use of vorticity operators and providing more suitable boundary conditions. Furthermore, our approach improves the accuracy of gradients, enabling the handling of longer sequences and achieving superior convergence results.

C. Differentiable Rendering

Differentiable rendering techniques have made significant progress in addressing the non-differentiable rasterization process in mesh-based graphics renderers. Previous approaches, such as OpenDR [30] and subsequent works [5], [13], [21], approximate inverse rendering in mesh-based pipelines. Recent advancements by Liu et al. [28] and Chen et al. [5] introduce truly differentiable rasterizers for improved performance. However, existing methods lack the ability to handle global lighting effects. To address this, Li et al. [25] propose the first differentiable rendering framework for Monte Carlo ray tracing, while Zhang et al. [46] introduce a differential theory for differentiable rendering of participating media based on radiative transfer. In our approach, we incorporate precomputed radiance, efficient rendering techniques, and a gradient-related visual hull strategy to support differentiable rendering and optimize fluid reconstruction efficiently and accurately. Specifically, we approximate the calculation of multiple-scattered light by combining a single scattering model with ambient light, as shown in Figure 3.

Fig. 3. From left to right: rendering results using the single scatter model, rendering results combining the single scatter model with ambient light, and rendering results using Blender path tracing with multiple scattering. All renderings are illuminated with parallel light of the same intensity.

III. Global Physical Prior Based Reconstruction

A. Overview

Our focus is on reconstructing incompressible fluids, which offers practical advantages in accurately modeling real-world fluid behavior. Compressible fluid simulations, on the other hand, involve computationally intensive algorithms and introduce additional complexities. Therefore, in many computer graphics and vision applications, compressible fluids are often approximated as incompressible to simplify the computations.

When it comes to fluid reconstruction, it is crucial to incorporate a global fluid physics prior to enforce physical constraints and make use of image information for accurate reconstruction. However, this task poses challenges due to the complex nature of fluid dynamics and the need to reconcile the physical constraints with sparse video input. We propose an approach that integrates a differentiable fluid simulator based on divergence-free Laplacian functions **(Component 1)** and a differentiable rendering module that precomputes radiance **(Component 2)**. This integration enables efficient reconstruction of incompressible smoke from sparsely sampled videos. Figure 2 depicts our proposed framework.

Component 1: Divergence-free Laplacian-based differentiable fluid simulator utilizes Laplacian eigenfunctions to derive complete and divergence-free velocity basis functions satisfying boundary conditions. By employing projection onto these basis functions instead of the traditional pressure projection process, we improve calculation efficiency and address the trade-off between derivative accuracy and operational efficiency. Additionally, the expressive power of the basis functions' dimension reduction is leveraged to reduce the storage requirements for intermediate variables, resulting in significant memory savings. Lastly, we utilize a frequency domain multi-scale strategy to enhance fluid control effectiveness.

Component 2: Pre-computed radiance differentiable rendering. We simplify the computation of differentiable rendering by employing a single-shot scattering model. In our method, we combine this model with a novel gradient-related visual hull, which allows us to optimize the rendering differentiation process. By integrating the improved differentiable fluid simulator with the differentiable renderer, we construct a fully differentiable framework for smoke reconstruction. To prevent derivative explosion, we introduce a temporal gradient exponential decay strategy.

TABLE I
Definitions of Notations

Notation	Definition
$\mathbf{u}$	Velocity
p	Pressure
$\mathbf{f}$	Force
ρ	Density
ρ_{inf}	Density inflow
$\mathcal{A}$	Advection operator
$\mathcal{P}$	Divergence-free projection operator
$\mathbf{q}^t$	Fluid states, $[\rho^t, \mathbf{u}^t]^T$
$\mathbf{s}^t$	Optimizable quantities, $[\rho^t_{inf}, \mathbf{f}^t]^T$
$\mathbf{k}$	Wave number
$\lambda_{\mathbf{k}}$	Laplacian eigenvalue
$\Psi(\mathbf{k})$	Divergence-free laplacian eigenfunction
$\Phi_x(\mathbf{k})$	x-component of divergence-free laplacian eigenfunction
N	Size of velocity field, $[N_x, N_y, N_z, 3]$
$\hat{u}_x(\mathbf{k})$	Spectral coefficient of x-component of velocity
w	Velocity basis coefficient
$\mathcal{R}$	Rendering operator
P	Set of point lights
$\mathcal{F}$	Fluid simulation operator, $(\mathcal{P} \circ \mathcal{A})$
φ	Objective function
Θ	Set of input views
$\mathbf{q}^t_{adj}$	Adjoint state of $\mathbf{q}^t$
H	Visual hull

In Section III-B, we explain our fluid simulation approach, which utilizes divergence-free Laplacian eigenfunctions. We show how this enhances our differentiable fluid simulation. Moving to Section III-C, we introduce precomputed radiance for differentiable rendering. We elaborate the framework outlined in Section III-D, offering more insights into how we optimize fluid reconstruction.

To facilitate understanding, we provide a comprehensive list of relevant notations in Table I.

B. Laplacian Eigenfunctions Based Differentiable Fluid Simulation

In this section, we discuss the forward and backward differentiation process of a differentiable fluid simulator based on the Euler method commonly used in computer graphics. The fluid velocity field is governed by the incompressible Navier-Stokes (NS) equations and the incompressibility condition. However, solving a large linear system of equations iteratively during the backward differentiation process can result in low solution accuracy and significant derivative errors. These issues hinder convergence and introduce unrealistic fluid behavior. To address these challenges, we propose a novel approach that employs the spectral method to solve the partial differential equations in the pressure projection step. By utilizing the spectral method, we avoid the iterative solving of linear equations, ensuring high projection accuracy while maintaining computational efficiency. This eliminates the need for time-consuming equation solving and reduces computational overhead. The spectral method allows us to achieve projection accuracy similar to high-precision methods while maintaining computational efficiency comparable to low-precision approaches.

1) Differentiable Fluid Simulation: Our method is based on the incompressible Navier-Stokes equations, which provide a strong physical foundation for simulating the behavior of incompressible fluids. These equations describe the motion of fluids with constant density, such as liquids and low-velocity smoke. The incompressible Navier-Stokes (NS) equations can be expressed as follows:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\nabla p + \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \tag{1}$$

where $\mathbf{u}$ represents the fluid velocity, p denotes the pressure, and $\mathbf{f}$ encompasses various forces acting on the fluid, such as viscous forces and external body forces.

Additionally, the evolution of density, which is passively advected by the velocity field, can be described by an advection-type equation:

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = \rho_s, \tag{2}$$

where ρ represents the density and ρ_s includes source terms, such as inflow and outflow.

To simulate the fluid behavior, we solve Equation 1 and Equation 2 at each frame t using an operator splitting approach. The simulation process can be described as follows:

$$\begin{aligned} \mathbf{u}^{t+1} &= \mathcal{P}(\mathcal{A}(\mathbf{u}^t, \mathbf{u}^t)) + \mathcal{P}(\mathbf{f}^{t+1}), \\ \rho^{t+1} &= \mathcal{A}(\rho^t, \mathbf{u}^{t+1}) + \rho_{inf}^{t+1}. \end{aligned} \tag{3}$$

where $\mathcal{A}(\mathbf{u}, \mathbf{u})$ and $\mathcal{A}(\rho, \mathbf{u})$ are differentiable advection operators that advect the velocity $\mathbf{u}$ and density ρ using the velocity field $\mathbf{u}$ individually. In our work, we discretize these operators using a modified MacCormack scheme [37]. $\mathcal{P}$ is a linear projection operator that ensures the velocity field remains divergence-free. ρ_{inf} represents the inflow part of the source term ρ_s in Equation 2. It is worth noting that the form of Equation 3 is different from that presented in [4]. We extract $\mathbf{f}^{t+1}$ and ρ_{inf}^{t+1} to separate the fluid states $\mathbf{q}^t = [\rho^t, \mathbf{u}^t]^T$ and the optimizable quantities $\mathbf{s}^t = [\rho_{inf}^t, \mathbf{f}^t]^T$. Splitting the fluid state and optimizing $\mathbf{s}^t$ allows for independent control and manipulation of the inflow sources and external forces, facilitating targeted adjustments and optimizations of the fluid simulation.

The necessary gradients for optimization are computed as:

$$\begin{aligned} \frac{\partial \mathbf{q}^{t+1}}{\partial \mathbf{q}^t}^T &= \begin{bmatrix} \frac{\partial \rho^{t+1}}{\partial \rho^t} & \frac{\partial \rho^{t+1}}{\partial \mathbf{u}^t} \\ \frac{\partial \mathbf{u}^{t+1}}{\partial \rho^t} & \frac{\partial \mathbf{u}^{t+1}}{\partial \mathbf{u}^t} \end{bmatrix}^T \\ &= \begin{bmatrix} \frac{\partial \mathcal{A}(\rho^t, \mathbf{u}^{t+1})}{\partial \rho^t} & \frac{\partial \mathcal{A}(\rho^t, \mathbf{u}^{t+1})}{\partial \mathbf{u}^{t+1}} \mathcal{P}^T \frac{\mathrm{d}\mathcal{A}(\mathbf{u}^t, \mathbf{u}^t)}{\mathrm{d}\mathbf{u}^t} \\ 0 & \mathcal{P}^T \frac{\mathrm{d}\mathcal{A}(\mathbf{u}^t, \mathbf{u}^t)}{\mathrm{d}\mathbf{u}^t} \end{bmatrix}^T \end{aligned} \tag{4}$$

$$\frac{\partial \mathbf{q}^t}{\partial \mathbf{s}^t} = \begin{bmatrix} \frac{\partial \rho^t}{\partial \rho_{inf}^t} & \frac{\partial \rho^t}{\partial \mathbf{f}^t} \\ \frac{\partial \mathbf{u}^t}{\partial \rho_{inf}^t} & \frac{\partial \mathbf{u}^t}{\partial \mathbf{f}^t} \end{bmatrix} = \begin{bmatrix} I & \frac{\partial \mathcal{A}(\rho^{t-1}, \mathbf{u}^t)}{\partial \mathbf{u}^t} \\ 0 & I \end{bmatrix}. \tag{5}$$

where Equation 4 represents the derivative of the density and velocity fields between two frames, and Equation 5 represents the derivative of the inflow and force fields. These equations provide a block matrix representation of the gradients and derivatives, allowing for efficient computation and storage,

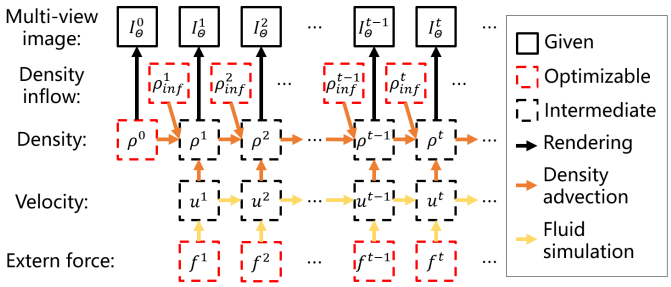


Fig. 4. Differentiable simulation process with global physical priors. We optimize the forces for each frame, as well as the density of the initial state and inflow. The velocity and intermediate density fields are indirectly modified as the simulation progresses.

and enabling the application of optimization techniques such as gradient-based methods, which leverage the derivatives to adjust and optimize the fluid simulation parameters, facilitating accurate and efficient simulations.

The optimization process depicted in Figure 4 involves using the derivatives described by equations 4 and 5. These derivatives capture the changes in density, velocity field, inflow, and force between frames. The process starts with the evolution of the fluid state from frame 0 to the last frame following the forward simulation. The derivative is then computed based on the objective function. By propagating the derivative backward from the last frame to frame 0, the impact of the control force field basis function coefficients on each frame's evolution is determined, and the external force field basis function coefficients are updated accordingly. In subsequent iterations, the updated coefficients of the control force field modify the fluid state's evolution by adjusting the control force field. This iterative process continues until the updated basis function coefficients align with the fluid state of the desired key frame. Automatic differentiation is utilized to calculate gradients related to the advection operator $\mathcal{A}$. Details about the project operator $\mathcal{P}$ and the representation of divergence-free fields can be found in Section III-B2

2) Laplacian Eigenfunctions-based Projection: Our contribution here is the utilization and application of Laplacian Eigenfunctions for the pressure projection step in Laplacian Eigenfunctions Based Differentiable Fluid Simulation. By incorporating this approach, we eliminate the need for iteratively solving linear equations, leading to improved efficiency and accuracy in the differentiable fluid simulation process.

Laplacian eigenfunctions are complete orthogonal basis functions designed specifically for a given domain with appropriate boundary conditions. They are derived from the eigenfunctions of the vector Laplacian operator Δ. One key advantage of Laplacian eigenfunctions is their ability to automatically eliminate all divergence terms by decomposing the field into linear combinations of these basis functions. Consequently, the computationally expensive pressure projection step is not required. In our 3D orthonormal Cartesian coordinate system, the vector Laplacian is equivalent to applying the scalar Laplacian to each vector component. To ensure divergence-free Laplacian eigenfunctions, we impose the additional condition that each basis field must also be divergence-free. Therefore,

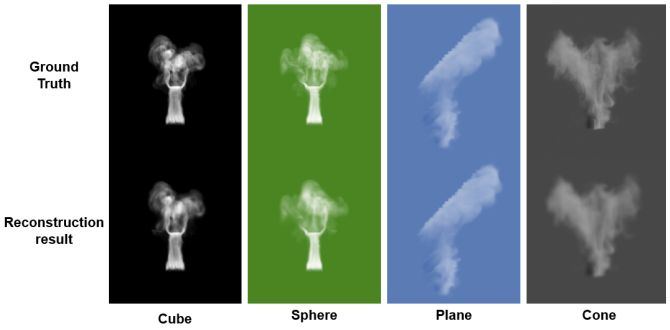


Fig. 5. The reconstruction results of smoke drifting around a cube, a sphere, a plane and a Cone.

the basis fields Ψ can be characterized by:

$$\begin{aligned} \Delta\Psi(\mathbf{k}) &= \lambda_{\mathbf{k}}\Psi(\mathbf{k}) \\ \nabla\cdot\Psi(\mathbf{k}) &= 0, \end{aligned} \tag{6}$$

where $\mathbf{k} = (k_x, k_y, k_z) \in \mathbb{Z}^3$ is the vector wave number, $\lambda_{\mathbf{k}}$ and $\Psi(\mathbf{k})$ are the eigenvalue and eigenfunction, respectively, corresponding to wave number $\mathbf{k}$. Here, $\nabla\cdot$ represents the divergence operator. It is important to note that each basis field $\Psi(\mathbf{k})$ is a 3D vector, i.e., $\Psi(\mathbf{k}) = [\Phi_x(\mathbf{k}), \Phi_y(\mathbf{k}), \Phi_z(\mathbf{k})]^T$, where $\Phi_x(\mathbf{k})$, $\Phi_y(\mathbf{k})$, and $\Phi_z(\mathbf{k})$ represent the x, y, and z velocity components, respectively.

In our work, we reconstruct fluids within a rectangular domain using no-slip Neumann boundary conditions. Although free-slip Neumann boundary conditions may align more closely with the behavior of real captured fluid, it is not possible to derive divergence-free Laplacian eigenfunctions under these boundary conditions. Our experiments show that the chosen boundary condition performs well. Under our domain and boundary condition, the basis fields have an analytic form:

$$\Psi(\mathbf{k}) = \begin{bmatrix} \Phi_x(\mathbf{k}) \\ \Phi_y(\mathbf{k}) \\ \Phi_z(\mathbf{k}) \end{bmatrix} = \begin{bmatrix} a\cos(k_x x)\sin(k_y y)\sin(k_z z) \\ b\sin(k_x x)\cos(k_y y)\sin(k_z z) \\ c\sin(k_x x)\sin(k_y y)\cos(k_z z) \end{bmatrix}, \tag{7}$$

where x, y, and z are defined within the domain $\Omega \in [0, \pi]^3$. The values of a, b, and c can be arbitrary, but they must satisfy the divergence-free condition $ak_x/N_x + bk_y/N_y + ck_z/N_z = 0$. Here, $N = [N_x, N_y, N_z, 3]$ represents the grid size of a velocity field.

This analytic form is also derived in [7]. However, they only choose one feasible solution for the divergence-free condition, resulting in an incomplete basis. We discovered that when k_x, k_y, and k_z are fixed, the divergence-free condition implies that all possible values of the vector $[a, b, c]^T$ lie in a plane perpendicular to the normal $[k_x/N_x, k_y/N_y, k_z/N_z]^T$. Therefore, any two linearly independent vectors can be used as bases to represent this plane. It is preferable for the vectors to be orthogonal and normalized to facilitate projection. In our work, we choose the following two vectors as bases under $\mathbf{k}$:

$$\begin{bmatrix} \frac{k_x k_y}{N_x N_y} \\ -(\frac{k_x^2}{N_x^2} + \frac{k_z^2}{N_z^2}) \\ \frac{k_y k_z}{N_y N_z} \end{bmatrix}, \begin{bmatrix} \frac{k_z}{N_x} \\ 0 \\ -\frac{k_x}{N_z} \end{bmatrix}, \tag{8}$$

which need to be normalized later. The chosen basis vectors in our differentiable fluid simulation capture the interactions between the wave vector components (k_x, k_y, k_z) and grid sizes (N_x, N_y, N_z) in a simplified and efficient manner. The components of these vectors represent the relationships between the various directions and grid sizes, allowing for straightforward calculations and projections. By using these specific basis vectors, we streamline the representation and computation process in our fluid simulation approach.

Moreover, the basis field with a larger vector wave number $\mathbf{k}$ corresponds to a smaller spatial scale of vorticity. We set the maximum value of the vector wave number $\mathbf{k}$ to be the same as the size of our Eulerian grids, ensuring that the smallest spatial scale of vorticity is equal to the size of a voxel. Consequently, our basis fields can effectively represent any divergence-free velocity fields up to this scale. It's important to note that one $\mathbf{k}$ corresponds to two basic fields. Assuming the number of voxels in the Eulerian grids is $N_x \times N_y \times N_z$, there exist $2 \times N_x \times N_y \times N_z$ basis fields, requiring only $2 \times N_x \times N_y \times N_z$ basis coefficients to represent a divergence-free velocity field, as depicted in Equation 9, where w_i represents the basis coefficients. In comparison to the grid-based representation with $3 \times N_x \times N_y \times N_z$ coefficients, our method allows for a one-third reduction in memory usage.

$$\mathbf{u} = \sum_{i=0}^{2\times N_x \times N_y \times N_z} w_i \Psi_i \tag{9}$$

It is worth noting that the analytic form of our basis fields consists of sine and cosine functions, enabling accelerated projection and recovery using a combination of discrete sine transform (DST), discrete cosine transform (DCT), and their inverses. For example, projecting the x component of a velocity field u_x yields:

$$\begin{aligned} \langle u_x, \Phi_x(\mathbf{k})\rangle &= \sum_x \sum_y \sum_z u_x a\cos(k_x x)\sin(k_y y)\sin(k_z z) \\ &= a\hat{u}_x(\mathbf{k}), \end{aligned} \tag{10}$$

where $\hat{u}_x(\mathbf{k})$ can be computed by performing a DCT in the x direction, followed by a DST in the y direction, and then in the z direction. To recover u_x from $\hat{u}_x(\mathbf{k})$, perform an inverse DST (IDST) in the z direction, followed by an inverse DCT (IDCT) in the y direction, and finally in the x direction. This projection step, followed by the recovery step, represents the projection operator $\mathcal{P}$ in Equation 3.

Our method shares similarities with Fourier spectral methods [29], [38] in that we both use sine and cosine functions to enforce incompressibility in the spectral domain. However, these methods do not address the Neumann boundary condition that we require. Furthermore, Fourier spectral methods enforce divergence-free conditions in the spectral domain by subtracting components with divergence, while we derive two basis vectors and project onto them. In our method, we approximate open boundary conditions using boundary conditions that allow flow from all faces. The velocity components in each direction satisfy Neumann boundary conditions on the boundary surface perpendicular to them, while the remaining components satisfy Dirichlet boundary conditions.

3) Reconstruction Coupling with Solids: To enable our method to support the interaction of solids in the reconstruction of fluids, we propose the spectral reconstruction method based on solid masks. During the reconstruction process, we introduce a regularization term to both the density field and the external force field. This regularization term enforces the density field in the region occupied by solids to be zero. Additionally, there is an outward-directed external force applied from the solid region, which pushes out any fluid density that may enter the interior of the solids. By reconstructing the density field and external force field simultaneously, we achieve accurate reconstruction of both fluids and solids. The spectral reconstruction method based on solid masks can handle arbitrary geometric shapes of solids and can seamlessly integrate with our fluid reconstruction method, allowing for precise and efficient reconstruction of fluids and solids.

Figure 5 showcases the reconstruction results of smoke interacting with a cube using our proposed method, which allows for the inclusion of solids in the fluid reconstruction process. The solid rendering is intentionally hidden to provide a clearer visualization of the fluid reconstruction. The reconstructed smoke flows convincingly around the solid cube, demonstrating the effective support of our method for interactive reconstruction involving solids. This capability is particularly valuable for applications such as fluid-structure interaction, where accurate simulation of the interaction between fluids and solids is crucial.

4) Spectral Domain Multi-scale Strategy: Efficiently reconstructing large-scale fluid motions while minimizing computational costs is essential for fluid rendering in interactive applications such as VR/AR. In conventional fluid reconstruction methods, spatial domain multi-scale strategies have been commonly used to adjust the resolution of density and velocity fields. However, in the realm of differentiable fluid simulation, the highly nonlinear nature of fluid dynamics can lead to substantial changes in fluid motions when altering spatial resolution. This can adversely affect convergence speed and overall simulation quality. To address this challenge, we introduce a spectral domain multi-scale strategy, specifically tailored for the force field, drawing inspiration from the work of [39]. While [39] employs a 3D Fourier transformation to shift the velocity field into the spectral domain, our approach adapts this strategy for the force field. We perform up-sampling and down-sampling on the control force field while maintaining the spatial resolution of the velocity and density fields unchanged. Consequently, although minor-scale movements may occur in the early optimization stages, the control force field converges starting from large scales, thereby mitigating the influence of small-scale motion derivatives.

In our approach, the force field is expressed as coefficients of basis functions derived from divergence-free Laplacian eigenfunctions. These basis functions exhibit different spatial frequencies represented by distinct wave vectors k, where each k corresponds to vortices of varying sizes on the divergence-free basis functions. Essentially, larger wave vectors k indicate basis functions with higher spatial frequency velocities. To implement a frequency domain multi-scale strategy during the optimization process, we only need to set the coefficients of basis functions to zero at larger wave vectors k. This approach proves simpler and more direct than a spatial multi-scale strategy, eliminating the requirement for managing fields of different spatial scales and avoiding the time-consuming and error-prone up-sampling and down-sampling processes.

Specifically, we define the truncated wave vector as $k_{scale} = \alpha_{scale} k_{max}$, where $k_{max} = [N_x, N_y, N_z]^T$ represents the maximum values of the wave vector. The truncation retention ratio α_{scale} is initially set to 0.01. During the optimization of basis function coefficients, only coefficients with wave vectors $k \leq k_{scale}$ are considered. Upon convergence, α_{scale} is updated to $2\alpha_{scale}$, expanding the range of optimizable wave vectors. This optimization process continues until $\alpha_{scale} \geq 1$, at which point coefficients of basis functions corresponding to all wave vectors become involved in the optimization.

Furthermore, our study identifies a challenge in the conventional application of a multi-scale strategy. Direct truncation at the top accumulates coefficients of basis functions continuously, leading to some coefficients remaining zero as optimization progresses. This results in high-frequency signals along the truncation plane, causing ringing artifacts when transforming the frequency-domain control force field to the spatial domain. To address this issue, we propose replacing direct truncation with Gaussian truncation. This entails multiplying the frequency-domain field by a three-dimensional Gaussian function with mean zero and deviation k_{scale}, maintaining the integrity of the frequency-domain information. All other procedures remain unchanged.

Our multi-scale strategy involves decomposing the force field into different frequency bands using Fourier analysis. By selectively reducing the representation of frequency components based on their significance, we achieve expedited computations while preserving essential fluid behaviors across various scales. This technique enhances the efficiency and accuracy of differentiable fluid simulation by capturing both high-frequency details and low-frequency global behaviors. Consequently, our method optimizes the overall performance and fidelity of the differentiable fluid simulation process.

C. Precomputed Radiance Differentiable Rendering

The aim of our approach is to establish a connection between image observations and fluid physical priors, allowing for the propagation of gradients in a differentiable manner. To achieve this, we employ a differentiable renderer that enables the propagation of derivatives from images to fluid quantities.

Rendering a density ρ in a given direction ω from near plane n to far plane f can be formulated as the integral:

$$\mathcal{R}(\rho, \omega) = \int_n^f L(x, \omega)\tau(n, x)dx, \tag{11}$$

where $L(x, \omega)$ represents the outgoing radiance from position x in the direction ω. The term $\tau(n, x)$ calculates the radiance attenuation along the ray from the current position x to the near plane, controlled by the extinction coefficient $\sigma_t(a)$:

$$\tau(n, x) = e^{-\int_n^x \sigma_t(a)da} \tag{12}$$

However, calculating the outgoing radiance recursively for multiple scattering processes can be time-consuming. To

expedite the rendering process, we approximate multiple scattering using single scattering with ambient light. Thus, the outgoing radiance can be expressed as:

$$L(x,\omega) = \frac{1}{4\pi}\sigma_s(x)(I_a + \sum_{p\in P} I_p \tau(x,p)), \tag{13}$$

where $\sigma_s(x)$ represents the scattering probability, $\frac{1}{4\pi}$ is the probability of scattering in the direction ω over a unit sphere, I_a is the ambient light intensity, and I_p denotes the intensity of point lights. The ambient light and all point lights in P contribute to the reduced incident radiance at position x. The coefficients $\sigma_t(x)$ and $\sigma_s(x)$ are both proportional to the density value $\rho(x)$, with proportionality constants of 1.0 and 0.8, respectively.

To handle the integrals in Equation 11, we utilize raymarching. Each sample point in Equation 11 requires a second ray marching step in Equation 13, resulting in quadratic time complexity with respect to the number of sample points. To mitigate this, we precompute the outgoing radiance at each voxel center in the given density field using Equation 13, and then perform Equation 11 by employing tri-linear interpolation at each sample point.

The process of derivative backpropagation for our differentiable rendering is derived and implemented manually. This eliminates the need to trace derivatives for each variable, resulting in reduced memory usage and accelerated computation speed. To ensure stable derivatives, we exclude derivative calculations in the precomputed outgoing radiance.

In summary, in differential fluid reconstruction, time is saved by precomputing the outgoing radiance at voxel centers, as shown in Equation 13. This eliminates the need for recursive calculations during rendering and reduces the overall time complexity. Additionally, the use of raymarching and tri-linear interpolation optimizes the estimation of radiance values at sample points, as described in Equation 11, resulting in efficient computation. By leveraging these techniques, the rendering process becomes faster and more efficient, facilitating the propagation of gradients from images to fluid quantities.

D. Reconstruction Framework

We now present our reconstruction framework, which aims to reconstruct density fields and velocity fields for all frames that align with the input sparse-view videos while respecting physical priors. To effectively leverage these physical priors, we optimize quantities that govern the fluid's behavior instead of directly optimizing the density and velocity fields.

Our optimization problem can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{s}^t} \quad & \varphi(\mathbf{q}) = \sum_{t=0}^{F-1} \varphi^t(\mathbf{q}^t) = \sum_{t=0}^{F-1} \frac{1}{|\Theta|} \sum_{\theta\in\Theta} \|\mathcal{R}(\mathbf{q}^t,\theta) - I_\theta^t\|_2^2 \\ s.t. \quad & \mathbf{q}^{t+1} = \mathcal{F}(\mathbf{q}^t) + \mathbf{s}^{t+1}, \quad t = 0,1,\ldots,F-2, \end{aligned} \tag{14}$$

In this formulation, $\mathcal{R}$ represents the rendering process, $\mathcal{F}$ represents the fluid simulation process, and F denotes the maximum number of frames. The variable t represents the quantity at frame t, and $\mathbf{q}^t = [\rho^t, \mathbf{u}^t]^T$ represents the fluid states, including the density field ρ^t and the velocity field $\mathbf{u}^t$.

Algorithm 1: Differentiable framework for fluid reconstruction

```
1  initialize all fields to zero;
2  load multi-view images of all frames;
3  for i ← 0 to max_iterations do
       // forward pass
4      for t ← 1 to F − 1 do
5          q^t ← F(q^{t−1}) + s^t;
6          store q^t overwrites s^t;
7      end
       // backward pass
8      for t ← F − 1 to 1 do
9          s^t ← q^t − F(q^{t−1});
10         calculate (∂q^{t+1}/∂q^t)^T , ∂q^t/∂s^t;
11         calculate dφ^t/dq^t;
12         if t == F − 1 then
13             q^t_adj ← dφ^t/dq^t;
14         else
15             q^t_adj ← (1 − β)(∂q^{t+1}/∂q^t)^T q^{t+1}_adj + β dφ^t/dq^t;
16         end
17         ∂φ/∂s^t ← q^t_adj^T ∂q^t/∂s^t;
18         update s^t with ∂φ/∂s^t;
19         store s^t overwrites q^t;
20     end
21     ∂φ/∂ρ^0 ← (1 − β)(∂q^1/∂ρ^0)^T q^1_adj + β dφ_0/dρ^0;
22     update ρ^0 with ∂φ/∂ρ^0;
23 end
```

Similarly, $\mathbf{s}^t = [\rho_{\text{inf}}^t, \mathbf{f}^t]^T$ represents the optimizable quantities, including the inflow density field ρ_{inf}^t and the force field $\mathbf{f}^t$. The first density field ρ^0 is also an optimizable quantity. All these fluid fields are represented by Eulerian grids. The set of input views is denoted by Θ, and I_θ^t represents the input image at view θ and frame t.

We solve this optimization problem using a gradient-based method, and the gradients are efficiently computed using the adjoint method, similar to the approach in [32]. Algorithm 1 outlines our optimization process. In each iteration, we perform a forward fluid simulation pass (lines 4-7), followed by a backward gradient calculation pass (lines 8-22). In the backward pass, the gradients within one frame are calculated in lines 10-11, and the final derivatives with respect to the optimizable quantities are computed in lines 17 and 21 using the adjoint states $\mathbf{q}_{\text{adj}}$. The adjoint states have an iterative form:

$$\begin{aligned} \mathbf{q}_{adj}^t = \frac{\partial \mathbf{q}^{t+1}}{\partial \mathbf{q}^t}^T \mathbf{q}_{adj}^{t+1} + \frac{\mathrm{d}\varphi^t}{\mathrm{d}\mathbf{q}^t}, \quad t = 0,1,\ldots,F-2 \\ \mathbf{q}_{adj}^t = \frac{\mathrm{d}\varphi^t}{\mathrm{d}\mathbf{q}^t}, \quad t = F-1, \end{aligned} \tag{15}$$

However, as shown in lines 12-16 and 21, we incorporate an exponential moving average (EMA) with decay β to the adjoint states. This technique, also used in [12], attenuates gradients from far frames exponentially. It is beneficial because near frames often provide more valuable information, and excessive accumulation of gradients can be avoided. It is worth noting that the gradients at the last frame are not multiplied by β, which serves as a bias correction to amplify gradients

at later frames. Additionally, we utilize the checkpointing technique as shown in lines 6, 9, and 19. Both fluid states $\mathbf{q}^t$ and optimizable quantities $\mathbf{s}^t$ are stored in the same location to reduce memory usage, although it requires one additional forward recomputation in each iteration.

To reduce the parameter space, a visual hull is commonly used in previous fluid reconstruction methods. The visual hull provides a conservative estimation of the regions where non-zero density values may exist, ensuring that no density value lies outside the visual hull. The visual hull is constructed using the silhouettes (SIL) of the input multi-view images and the intersection of their inverse rendering results ($\mathcal{R}^{-1}$). Mathematically, the visual hull can be defined as:

$$H^t = \bigcap_{\theta \in \Theta} \mathcal{R}^{-1}(\text{SIL}(I^t_\theta), \theta), \tag{16}$$

where Θ represents all input views.

Previous works have treated the visual hull as a hard constraint when optimizing density fields. While this approach eliminates density residuals outside the visual hull, it poses a challenge in our reconstruction framework, where density fields are considered intermediate variables that are not directly optimized. Simply cutting off density outside the visual hull would reduce the sum of density values, potentially leading to sub-optimal results. One possible remedy is to introduce a soft regularization term to the optimization problem to minimize density values outside the visual hull. However, this approach requires a carefully chosen regularization coefficient.

To overcome these limitations, we propose a novel gradient-related visual hull strategy that eliminates the need for additional regularization terms. We define a voxel v as being outside the visual hull H^t if its projection lies outside the image silhouette in at least one view. Let Θ_{out} denote the set of views where the projection of v is outside the silhouette, and Θ_{in} represent the remaining views ($\Theta_{\text{in}} = \Theta - \Theta_{\text{out}}$). It is observed that the derivatives with respect to the images from views in Θ_{out} tend to decrease the projected image values, resulting in a reduction of the density values at voxel v. However, derivatives from views in Θ_{in} do not guarantee this effect, which makes it challenging to eliminate density residuals outside the visual hull. Equation 17 presents a straightforward solution to address this issue.

$$\frac{\partial \varphi^t}{\partial \rho^t(v)} = \begin{cases} \sum_{\theta \in \Theta} \frac{\partial I^t_\theta}{\partial \rho^t(v)}^T \frac{\partial \varphi^t}{\partial I^t_\theta}, & v \in H^t \\ \sum_{\theta \in \Theta} \min(\frac{\partial I^t_\theta}{\partial \rho^t(v)}^T \frac{\partial \varphi^t}{\partial I^t_\theta}, 0). & v \notin H^t \end{cases} \tag{17}$$

In Equation 17, when a voxel v is located outside the visual hull, we enforce positive derivatives with respect to v from each view θ. By considering both Θ_{out} and Θ_{in} together, we ensure the desired effect on derivatives from Θ_{out} while eliminating conflicting derivatives from Θ_{in} that could hinder the reduction of density residuals. Treating both sets together streamlines the view partitioning process, saving computational time.

The effectiveness of our visual hull strategy is demonstrated in Figure 6, where the target dual-view image is optimized from the initial density fields in the upper-left and lower-left positions.

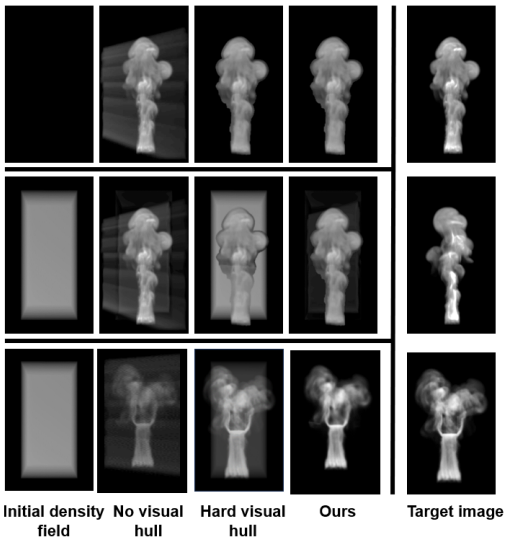


Fig. 6. Impact of different visual shell application strategies on differentiable rendering

The resulting images after 20 iterations using different methods with the visual hull strategy are combined on the far right. To highlight the contrast effect, the brightness of all images is increased while reducing the contrast proportionally. The second column exhibits artifacts parallel to the camera ray trace when the visual hull is not utilized. In the third column, limitations of the hard visual hull method are evident in handling density values outside the visual hull. On the other hand, our proposed method, shown in the fourth column, effectively avoids artifacts, handles density values outside the visual hull, and does not require the introduction of new regularization coefficients.

In summary, our reconstruction framework aims to reconstruct density fields ρ and velocity fields $\mathbf{v}$ for sparse-view videos by optimizing fluid behavior governing quantities. To achieve this, we formulate an optimization problem that minimizes the discrepancy between rendered frames and input images while respecting physical priors. The framework incorporates a novel gradient-related visual hull strategy, which effectively eliminates density residuals outside the visual hull without the need for additional regularization terms. By solving the optimization problem using a gradient-based method and leveraging the adjoint method for efficient gradient computation, we obtain the reconstructed density fields ρ and velocity fields $\mathbf{v}$. The novelty of our framework lies in optimizing fluid behavior governing quantities and effectively leveraging physical priors. Additionally, our gradient-related visual hull strategy enhances the handling of density residuals outside the visual hull, resulting in more accurate reconstructions.

IV. Experiments

To validate the effectiveness of our proposed method, we use a large number of mature synthetic data for quantitative comparison, and a set of real data for residual error comparison. Our focus was on reconstructing smoke as the target medium. Now, we describe the experimental setups, including qualitative

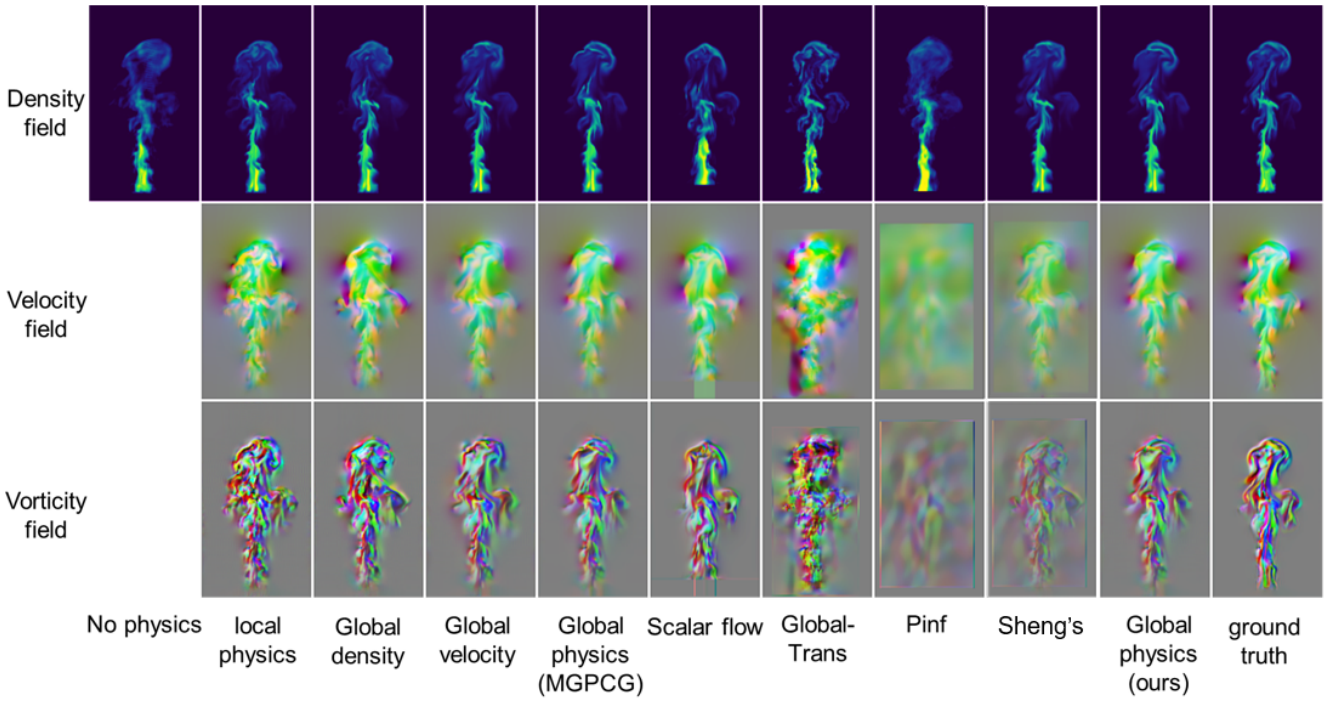


Fig. 7. Visualization of the density field (top), velocity field (middle), and vorticity field (bottom) in a slice of the reconstructed data. The color coding represents the value of reconstructed field. Density field is scalar value, while velocity and vorticity field are vector field. The ground truth is obtained from a virtual dataset constructed using the NS solver [40]. The first four columns and the tenth column represent the reconstruction results of our method without any prior, with a local physical prior, with a global density prior, with a global velocity prior, and with a global physical prior, respectively. The fifth column shows the reconstruction result with a global prior, but the Laplacian projection is replaced with traditional pressure projection. The seventh and eighth columns display the results from ScalarFlow [9], Global Transport [12], Pinf [6], and Sheng's work [35].

and quantitative evaluations, ablation experiments, and comparisons with state-of-the-art methods such as ScalarFlow (SF) [9], Global Transport (Glob-Trans) [12], Pinf [6] and Sheng's work [35]. Additionally, we provide details about the hardware and software configurations used in our experiments, including the use of the Taichi language [17] with diffTaichi [16] for automatic differentiation in advection.

Synthetic Data: We first conducted qualitative and quantitative experiments on synthetic data to evaluate the effectiveness of our method. Smoke was chosen as the target medium for reconstruction. We reconstructed the smoke from five view images, with the positions and orientations of the cameras matching those used in ScalarFlow (SF), Global Transport (Glob-Trans), Pinf and Sheng's work. These cameras were evenly spread horizontally around 120 degrees and employed perspective projection.

Real Data: To further assess the performance of our method, we performed experiments on real data and compared the results with previous works. Once again, smoke was used as the target medium for reconstruction. Similar to the synthetic data experiments, we reconstructed the smoke from 5 view images. The camera positions and orientations were identical to those used in the synthetic data experiments, ensuring consistency and fair comparisons.

Hardware and Software Configuration: All experiments were conducted on a system equipped with an Intel(R) Core(TM) i7-9700 CPU, an NVIDIA GeForce 2070 8GB GPU, and 16GB of memory. Our code was implemented using the Taichi language [17], which is a high-performance parallel programming language embedded in Python. For automatic differentiation in the advection process, we utilized diffTaichi [16]. Since most of our code relied on GPU acceleration, we used CUDA as the backend for Taichi.

A. Comparison on Synthetic Data

To evaluate our performance on synthetic data, we utilize the MantaFlow [40] open-source fluid solver for simulating the data. The simulation employs Eulerian simulation with MAC grids, utilizing a PCG pressure projection solver with a tolerance of 1e-12 and MacCormack advection [37]. Additionally, stochastic noise is introduced to the inflow region, and time-varying body forces are manually applied to generate complex fluid flow motion. The synthetic data has a spatial resolution of $100 \times 178 \times 100$ and comprises a set of rising smoke plumes with 120 frames. The density and velocity fields of each frame are saved for comparison.

For a fair comparison, we generate multi-view images using the respective rendering code of each compared method, and then reconstruct the densities and velocities using the same rendering setup. This ensures that the reconstructed data closely resembles the synthetic data. To quantitatively evaluate the reconstruction quality, we calculate the average root mean square error (RMSE) and structural similarity index (SSIM) of densities and velocities across all 120 frames. Lower RMSE values and higher SSIM scores indicate better reconstruction

TABLE II
QUANTITATIVE COMPARISON RESULTS OF BOTH DENSITY AND VELOCITY FIELDS. THE FIRST FOUR ROWS PRESENT THE QUANTITATIVE COMPARISON WITH SCALARFLOW [9], GLOBAL TRANSPORT (GLOB-TRANS) [12], PINF [6] AND SHENG'S [35]. THE LAST SIX ROWS SHOW THE RESULTS OF THE ABLATION EXPERIMENTS: NO PHYSICAL PRIOR (NP), LOCAL PHYSICAL PRIOR (LP), GLOBAL DENSITY PRIOR (GDP), GLOBAL VELOCITY PRIOR (GVP), AND GLOBAL PHYSICAL PRIOR (GP).

Method \ Metric	imgSSIM	imgLPIPS	desRMSE	desSSIM	velRMSE	velSSIM
ScalarFlow [9]	0.9706	0.02619	0.3147	0.9764	0.04972	0.9759
Glob-Trans [12]	0.9769	0.01528	0.5838	0.9562	0.1034	0.9547
Pinf [6]	0.9668	0.03747	0.4414	0.9695	0.0792	0.9591
Sheng's [35]	0.9589	0.02934	0.4762	0.9523	0.0825	0.9625
mgpcg(GP)	0.9869	0.01104	0.1679	**0.9907**	0.03870	**0.9814**
Laplace(NP)	0.9794	0.01794	0.2132	0.9845	N/A	N/A
Laplace(LP)	0.9812	0.01357	0.1941	0.9852	0.04983	0.9746
Laplace(GDP)	0.9855	0.01106	0.1709	0.9889	0.04782	0.9785
Laplace(GVP)	0.9867	**0.00999**	**0.1603**	0.9900	**0.03645**	0.9802
Laplace(GP)	**0.9869**	0.01101	0.1679	0.9907	0.03870	0.9814

TABLE III
QUANTITATIVE COMPARISON RESULTS OF RELATIVE ERROR. THE FIRST THREE ROWS SHOW THE RESULTS OF OUR REDUCED-ORDER PROJECTION METHOD WITH DIFFERENT ORDER, COMPARING WITH FULL-ORDER METHOD MGPCG.

Method \ Metric	imgRRMSE(%)	desRRMSE(%)	velRRMSE(%)
Laplace(10)	9.34	10.22	9.43
Laplace(25)	6.74	6.07	5.78
Laplace(50)	5.09	5.49	4.12
Laplace(100)	**4.87**	**5.01**	**3.60**
mgpcg	4.87	5.04	3.67

TABLE IV
TIME USAGE COMPARISON OF PROJECTION METHODS IN THE FLUID SIMULATION PROCESS. OUR LAPLACIAN-BASED PROJECTION METHOD REQUIRES LESS TIME THAN TRADITIONAL PRESSURE PROJECTION WHILE ENSURING DIVERGENCE-FREE VELOCITY.

Method \ Performance	Projection error	Projection derivation error	Residual divergence	Time cost
Ours	1305.69	2.05e-06	continuous form no divergence	27.78ms
Pressure Projection (1e-3)	1309.91	6.35e-04	9.17e-05	34.57ms
Pressure Projection (1e-6)	1296.97	2.43e-05	3.46e-06	55.52ms
Pressure Projection (1e-9)	1297.75	3.97e-07	1.21e-07	125.72ms

TABLE V
TIME USAGE COMPARISON OF RECONSTRUCTION METHODS. OUR METHOD IS SIGNIFICANTLY FASTER THAN OTHER METHODS THAT UTILIZE GLOBAL PRIOR (GLOBAL TRANSPORT [12] AND PINF [6]) WHILE INCORPORATING ADDITIONAL VELOCITY FIELD RESTRICTIONS.

Method \ Performance	Time cost
Ours	5h21m
Glob-Trans [12]	21h14m
Pinf [6]	23h23m

quality. Note that when evaluating the velocity index, we exclude velocities outside the visual hull, as our main focus is on velocities associated with densities.

1) Global Physics Prior Ablation Experiments: In our method, global physical priors play a crucial role in improving the quality of smoke reconstruction. To evaluate the effectiveness of the global physical prior, we conduct ablation experiments using the same 5-view smoke videos as input. We compare the final smoke reconstruction results of three different methods: 1) one without any physical prior, 2) one employing local physical priors, and 3) one employing global physical priors.

In the global density prior experiment, we imposed a constraint during the reverse differentiation process of the differentiable fluid simulator. Specifically, we set the derivative of the next-frame velocity field with respect to the current-frame velocity field to 0. This constraint effectively halts the propagation of velocity field derivatives between frames, creating a local physical prior within the velocity field while preserving a global one in the density field. Conversely, in the global velocity prior experiment, we applied a similar approach by setting the derivative of the next-frame density field to 0.

In these ablation experiments as shown in Figure 7 and Table II, the three methods differ in the way they handle velocity evolution while using differentiable density advection to connect all densities. The method without any physical prior considers velocities independently, and no velocity advection is performed. This means that derivatives can only affect the velocity of the current frame. Alternatively, the method using local physical priors evolves velocities using fluid simulation, but derivatives are not backward propagated through velocities. Finally, the method employing global physical priors, which is our proposed method, incorporates global physical priors into the velocity evolution process, enabling the propagation of derivatives through all frames.

Figure 7 shows the density and velocity slices at the last

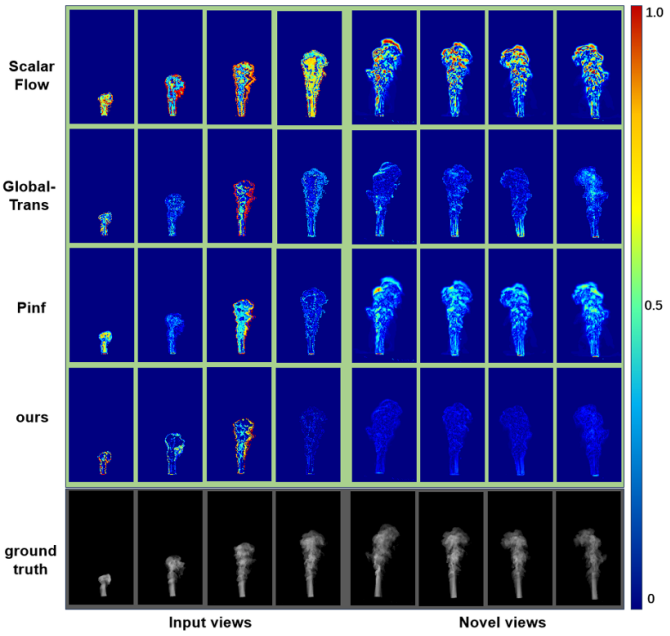


Fig. 8. Reconstruction result on real dataset. The color coding represents the difference between the ground truth and the reconstruction results of each method. In comparison to our method, the reconstruction results obtained using Glob-Trans [12] exhibit significant noise, resulting in an unrealistic density distribution. On the other hand, the reconstruction results produced by ScalarFlow [9] and Pinf [6] appear relatively vague, lacking fine details.

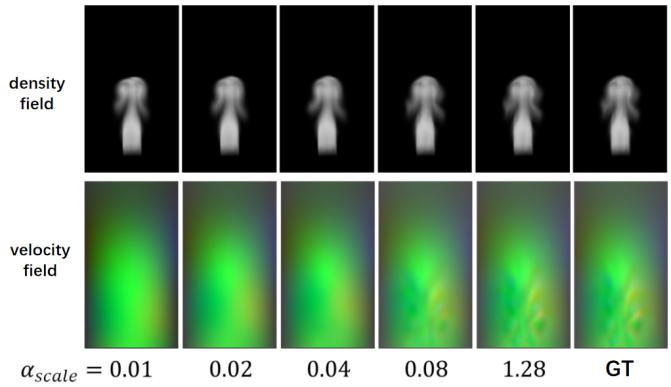


Fig. 9. Qualitative comparison results of the multi-scale strategy. As the truncation threshold increases, the details in the velocity field become more pronounced. Additionally, the use of the Gaussian truncation method ensures the absence of significant ringing artifacts.

frame of the smoke videos reconstructed using the three different methods. It is evident that the smoke reconstruction results using the global physical prior method exhibit the highest level of accuracy. The method employing local physical priors shows fluctuations in the density field at the boundaries of the smoke due to the derivative propagation being limited to a single frame. However, the method without any physical priors exhibits even more pronounced density fluctuations, and the resulting velocity field appears chaotic.

Quantitative evaluation results are presented in Table II, confirming the effectiveness of our global physical prior method in improving the accuracy of smoke reconstruction.

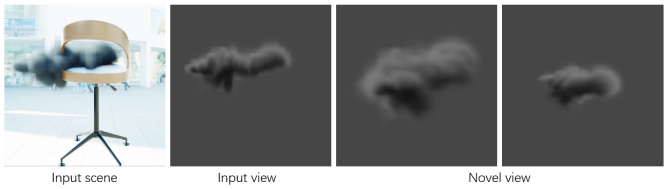


Fig. 10. Rendering and visualization of the density fields reconstructed on the "chair" scene. Our method reconstructs the density distribution of the smoke using 5 camera views.

2) Laplacian Eigenfunctions Ablation Experiments: In our method, we use the divergence-free Laplacian projection to ensure the divergence-free property of the reconstructed velocity field. To evaluate its effectiveness, we conducted ablation experiments where we replaced the divergence-free Laplacian projection with the traditional pressure projection method, specifically the Multi-Grid Preconditioned Conjugate Gradient (MGPCG) method [31].

Table IV presents a comparison of the time cost between the divergence-free Laplacian projection and the low/high tolerance MGPCG pressure projection. Interestingly, our Laplacian-based projection method exhibits shorter computation times even compared to the low tolerance MGPCG method, while still achieving a continuous form of incompressibility that closely resembles reality.

Figure 7 and Table II demonstrate the smoke reconstruction results obtained using the two projection methods. The comparison clearly shows that the use of the divergence-free Laplacian projection effectively reduces the error in smoke reconstruction and eliminates the need for the time-consuming pressure projection step.

We also evaluated the scatterless projection method of the velocity field based on the scatterless Laplacian eigenfunction proposed in our work. We compared it with the traditional pressure projection method using different iterative solution accuracies. The results are presented in Table IV. It is observed that the scatterless projection method achieves better projection accuracy compared to MGPCG with a tolerance of 1e-3 but is not as accurate as MGPCG with higher precision. This suggests that the current discrete field size may not fully highlight the projection accuracy advantage of the scatterless projection method. However, as the discrete field resolution improves, the advantage of the scatterless projection method, with its high convergence order as a spectral method, will become more apparent, resulting in faster improvement in projection accuracy.

Furthermore, we compared the calculation time of our method with the traditional pressure projection method. The calculation time of our method is less than that of MGPCG with a tolerance of 1e-3, only half that of MGPCG with a tolerance of 1e-6, and even 4.5 times less than that of MGPCG with a tolerance of 1e-9. Thus, our method's calculation time is comparable to that of the traditional method with low accuracy and exhibits significant advantages over the traditional method with high precision. Table III presents the relative RMSE error comparison between MGPCG and our method at different orders. The results reveal that as the order increases, our method

rapidly converges towards the accuracy of full-order methods. Notably, leveraging our divergence-free projection approach allows our final reconstruction accuracy to even surpass that of full-order methods.

In Figure 10, the results of our method at various scales during the optimization process are depicted. It is evident that as the scale increases, enabling a higher optimization frequency, the converged external force field exhibits more pronounced high-frequency velocities. Since the ground truth external force field is predominantly low-frequency, the final result closely resembles the ground truth due to the increased scale and optimization frequency.

The error analysis of the projected derivative is an important aspect to consider in our scatter-free projection method. It is worth noting that the calculation process of the derivative during the back-propagation phase mirrors that of the forward projection process. Hence, if we only focus on the backward differentiation process, the derivative error would exhibit similarities to the projection error shown in the first column of Table IV. However, it is crucial to take into account the entire optimization process, encompassing both the forward projection and backward differentiation stages, when evaluating the derivative. This is because the accuracy of the forward projection process directly impacts the resulting derivative.

Assuming that the target velocity field after the scatter-free projection is denoted as u_{tar}, we define the target function as $\sigma(u) = ||\mathcal{P}(u) - u_{tar}||_2^2$. Utilizing the idempotence and linearity of the projection, we deduce that $\frac{d\phi}{du} = \mathcal{P}(2(\mathcal{P}(u) - u_{tar})) = 2\mathcal{P}(u - u_{tar})$. Ideally, the derivative should be zero. However, during the actual derivative calculation, both in our proposed method and the traditional approach, the derivative is calculated using two projection operations for generality purposes. As a result, there exists a derivative error, as demonstrated in the second column of Table IV.

Remarkably, our method achieves a projected derivative accuracy that closely approximates the high precision projection accuracy, falling between the MGPCG(1e-6) and MGPCG(1e-9) methods. This signifies the effectiveness of our approach in accurately estimating the derivative, which is crucial for capturing intricate dynamics and fine details during the optimization process.

3) Comparison with State-of-the-art: We compare our method with ScalarFlow (SF) [9], Global Transport (Glob-Trans) [12], Pinf [6] and Sheng's work [35]. Table II presents the quantitative comparison results. Both ScalarFlow and Glob-Trans employ local physical priors. ScalarFlow reconstructs the fluid frame by frame, using the physical prior on velocity evolution only to initialize the estimation for the next frame. While their reconstructed velocities appear better than ours due to sharing codes with mantaflow, the reconstructed densities are inferior. This discrepancy arises because errors in previous reconstructions accumulate and hinder the convergence of the optimization process. On the other hand, Glob-Trans utilizes global density advection to connect all densities but treats velocities independently. They incorporate local physical priors by introducing additional velocity advection matching regularization. However, the reconstructed results, as shown in Table II, are unsatisfactory. The results from Pinf and Sheng's methods highlight a significant manual tuning requirement for regularization terms, as their approaches are not entirely physics-based. This characteristic may lead to suboptimal performance on new data. Upon analysis, we observe extremely high density values near the inflow region, likely caused by the low/high density value ambiguity in the rendering process mentioned in their paper, as we employ images with a black background. Additionally, their velocity divergence is high because they consider the divergence-free condition as a soft regularization term with manually set coefficients. This experiment reveals the sensitivity of manually set regularization coefficients, which may lead to suboptimal results.

Table V presents the time costs of these methods. Our method utilizes 4.188GB of GPU memory, whereas MGPCG uses 7.344GB, which is only 0.75 times of the latter. Our method outperforms Glob-Trans in terms of computational time, even when considering the time-consuming divergence-free projection in each iteration and each frame. Two key factors contribute to this improvement. First, our method requires only one-fifth of the number of iterations due to the utilization of a global physical prior. Second, the Taichi [17] language employed in our method provides higher-performance GPU computing compared to TensorFlow [1] used by Glob-Trans. In our implementation, differentiable rendering accounts for approximately 30% of the overall computation time, while differentiable fluid simulation occupies around 67% of the time (with more than 95% of this time dedicated to Laplacian projection).

4) Reconstruction on complex scene: In the final test, our algorithm is evaluated in intricate scenes featuring complex-shaped obstacles and directional lighting with an environment map. A gray color serves as the background during training. We employ five evenly distributed cameras arranged in a circle, simulating the chair scene at a resolution of $312 \times 178 \times 100$ for 120 steps. The domain incorporates a strong free stream velocity without a force field. The simulation involves a shallow sheet of smoke passing through the chair from the left. The outcome demonstrates the effectiveness of our method in reconstructing smoke amidst complex obstacles.

B. Comparison on Real Data

We obtained real captured 5-view videos of a rising smoke plume consisting of 120 frames from the ScalarFlow dataset [9]. Figure 8 showcases the input 5-view real captured images of the last frame, along with the residual rendering results of the reconstructed density by different methods. The residual is doubled to show the difference clearer. When comparing with the ground truth, Glob-Trans exhibits high-frequency noise, likely due to its reconstruction starting from random values inside the visual hull. On the other hand, ScalarFlow and Pinf produces smoother results, but we can observe disturbances near the boundary region that deviate from the ground truth. These disturbances arise from the accumulation of previous errors. In contrast, our method yields reconstructions that better align with the input images, demonstrating improved accuracy and fidelity.

V. Conclusion and Discussion

In conclusion, a novel method is introduced for recovering the density and velocity of a fluid from an input video sequence. The key novelty of our approach lies in the implementation of derivative tracking through differentiable fluid solvers and renderers, enabling the reconstruction process to consider a global physical prior rather than just focusing on derivatives between adjacent frames. Additionally, we employ the Laplacian eigenfunction for velocity projection, ensuring the divergence-free condition of the velocity field and mitigating the accumulation of reconstruction errors over the sequence. Experimental results demonstrate our effectiveness in improving the accuracy of fluid reconstruction and generating more natural velocity fields. However, there are still opportunities for further improvement, such as extending the method to handle fluids with rigid bodies and exploring advanced optimization techniques and fluid renderers to enhance the quality of the reconstruction results

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM transactions on graphics (TOG)*, 27(5):1–9, 2008.

[3] F. d. A. Belbute-Peres, K. A. Smith, K. R. Allen, J. B. Tenenbaum, and J. Z. Kolter. End-to-end differentiable physics for learning and control. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, p. 7178–7189. Curran Associates Inc., Red Hook, NY, USA, 2018.

[4] R. Bridson. *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015.

[5] W. Chen, H. Ling, J. Gao, E. J. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9605–9616, 2019.

[6] M. Chu, L. Liu, Q. Zheng, E. Franz, H.-P. Seidel, C. Theobalt, and R. Zayer. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022.

[7] Q. Cui, P. Sen, and T. Kim. Scalable laplacian eigenfluids. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[8] M.-L. Eckert, W. Heidrich, and N. Thuerey. Coupled fluid density and motion from single views. In *Computer Graphics Forum*, vol. 37, pp. 47–58. Wiley Online Library, 2018.

[9] M.-L. Eckert, K. Um, and N. Thuerey. Scalarflow: a large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019.

[10] R. Fattal and D. Lischinski. Target-driven smoke animation. *ACM Trans. Graph.*, 23(3):441–448, aug 2004. doi: 10.1145/1015706.1015743

[11] Z. Forootaninia and R. Narain. Frequency-domain smoke guiding. *ACM Transactions on Graphics (TOG)*, 39(6):1–10, 2020.

[12] E. Franz, B. Solenthaler, and N. Thuerey. Global transport for fluid reconstruction with learned self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1632–1642, 2021.

[13] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. Unsupervised training for 3d morphable model regression. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8377–8386. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00874

[14] J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich. Stochastic tomography and its applications in 3d imaging of mixing fluids. *ACM Transactions on Graphics (TOG)*, 31(4):1–10, 2012.

[15] P. Holl, V. Koltun, and N. Thuerey. Learning to control pdes with differentiable physics, 2020.

[16] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. Difftaichi: Differentiable programming for physical simulation. *ICLR*, 2020.

[17] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):201, 2019.

[18] C. Huang, J. Lei, M. Liu, and X. Peng. An improved kgf-sph with a novel discrete scheme of laplacian operator for viscous incompressible fluid flows. *International Journal for Numerical Methods in Fluids*, 81(6):377–396, 2016.

[19] I. Ihrke and M. Magnor. Image-based tomographic reconstruction of flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 365–373, 2004.

[20] Y. Ji, J. Ye, and J. Yu. Reconstructing gas flows using light-path approximation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2507–2514, 2013.

[21] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3907–3916. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00411

[22] B. Kim, V. C. Azevedo, M. Gross, and B. Solenthaler. Transport-based neural style transfer for smoke simulations. *arXiv preprint arXiv:1905.07442*, 2019.

[23] T. Kim and J. Delaney. Subspace fluid re-simulation. *ACM Transactions on Graphics (TOG)*, 32(4):1–9, 2013.

[24] A. Kontogiannis and M. P. Juniper. Physics-informed compressed sensing for pc-mri: An inverse navier-stokes problem. *IEEE Transactions on Image Processing*, 32:281–294, 2023. doi: 10.1109/TIP.2022.3228172

[25] T. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph.*, 37(6):222, 2018. doi: 10.1145/3272127.3275109

[26] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids, 2019.

[27] J. Liang, M. C. Lin, and V. Koltun. *Differentiable Cloth Simulation for Inverse Problems*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[28] S. Liu, W. Chen, T. Li, and H. Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *CoRR*, abs/1901.05567, 2019.

[29] B. Long and E. Reinhard. Real-time fluid simulation using discrete sine/cosine transforms. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pp. 99–106, 2009.

[30] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, vol. 8695 of *Lecture Notes in Computer Science*, pp. 154–169. Springer, 2014. doi: 10.1007/978-3-319-10584-0

[31] A. McAdams, E. Sifakis, and J. Teran. A parallel multigrid poisson solver for fluids simulation on large grids. In *Symposium on Computer Animation*, pp. 65–73, 2010.

[32] A. McNamara, A. Treuille, Z. Popović, and J. Stam. Fluid control using the adjoint method. *ACM Transactions On Graphics (TOG)*, 23(3):449–456, 2004.

[33] M. Okabe, Y. Dobashi, K. Anjyo, and R. Onai. Fluid volume modeling from sparse multi-view images by appearance transfer. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

[34] J. Panuelos, R. Goldade, E. Grinspun, D. Levin, and C. Batty. Polystokes: A polynomial model reduction method for viscous fluid simulation. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023.

[35] S. Qiu, C. Li, C. Wang, and H. Qin. A rapid, end-to-end, generative model for gaseous phenomena from limited views. In *Computer Graphics Forum*, vol. 40, pp. 242–257. Wiley Online Library, 2021.

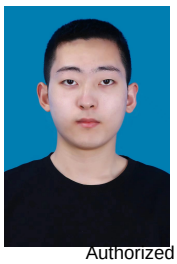
[36] S. Sato, Y. Dobashi, and T. Kim. Stream-guided smoke simulations. *ACM Transactions on Graphics (TOG)*, 40(4):1–7, 2021.

[37] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An unconditionally stable maccormack method. *Journal of Scientific Computing*, 35(2):350–371, 2008.
[38] J. Starn. A simple fluid solver based on the fft. *Journal of graphics tools*, 6(2):43–52, 2001.
[39] J. Tang, V. C. Azevedo, G. Cordonnier, and B. Solenthaler. Honey, i shrunk the domain: Frequency-aware force field reduction for efficient fluids optimization. In *Computer Graphics Forum*, vol. 40, pp. 339–353. Wiley Online Library, 2021.
[40] N. Thuerey and T. Pfaff. MantaFlow, 2018. *http://mantaflow.com*.
[41] A. Treuille, A. McNamara, Z. Popović, and J. Stam. Keyframe control of smoke simulations. In *ACM SIGGRAPH 2003 Papers*, pp. 716–723. 2003.
[42] G. Van Eyndhoven, K. J. Batenburg, D. Kazantsev, V. Van Nieuwenhove, P. D. Lee, K. J. Dobson, and J. Sijbers. An iterative ct reconstruction algorithm for fast fluid flow imaging. *IEEE Transactions on Image Processing*, 24(11):4446–4458, 2015. doi: 10.1109/TIP.2015.2466113
[43] T. D. Witt, C. Lessig, and E. Fiume. Fluid simulation using laplacian eigenfunctions. *ACM Transactions on Graphics (TOG)*, 31(4CD):10.1–10.11, 2012.
[44] J. Xiong, R. Idoughi, A. A. Aguirre-Pablo, A. B. Aljedaani, X. Dun, Q. Fu, S. T. Thoroddsen, and W. Heidrich. Rainbow particle imaging velocimetry for dense 3d fluid velocity imaging. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.
[45] G. Zang, R. Idoughi, C. Wang, A. Bennett, J. Du, S. Skeen, W. L. Roberts, P. Wonka, and W. Heidrich. Tomofluid: reconstructing dynamic fluid from sparse view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1870–1879, 2020.
[46] C. Zhang, L. Wu, C. Zheng, I. Gkioulekas, R. Ramamoorthi, and S. Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6):227:1–227:16, 2019. doi: 10.1145/3355089.3356522
[47] Q. Zhang, S. Xiao, Y. Cen, J. Han, and X. Liang. Global physical prior based fluid reconstruction for vr/ar. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 823–824. IEEE, 2023.
[48] B. Zineddin, Z. Wang, and X. Liu. Cellular neural networks, the navier–stokes equation, and microarray image reconstruction. *IEEE Transactions on Image Processing*, 20(11):3296–3301, 2011. doi: 10.1109/TIP.2011.2159231

BIOGRAPHY SECTION

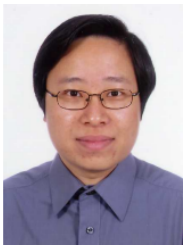
Shibang Xiao received the bachelor's degree from Beijing University of Technology, Beijing-Dublin International College. He is currently working toward the PhD degree with Beihang University advised by Prof. Xiaohui Liang. His current research field is fluid reconstruction and virtual reality.

Chao Tong received his Ph.D. degrees in 2009 in computer science from Beihang University (BUAA). He is an associate professor in the School of Computer Science and Engineering, BUAA. He was also a visiting professor in the School of Computer Science, McGill University. He has published more than 100 referred papers and filed more than 30 patents. His current research interests include machine learning, mobile computing and social networks analysis.



Qifan Zhang received a Bachelor's degree from Sichuan University and received a Master's degree from Beihang University. His research interests include fluid reconstruction and virtual reality.

Yunchi Cen received the Bachelor's degree from Guangxi Normal University and received the Master's degree from South China Normal University. Now, he is pursuing a Ph.D. degree at the School of Computer Science and Engineering, Beihang University. His research interests include differentiable rendering and fluid modeling.



Frederick W.B. Li received a B.A. and an M.Phil. degree from Hong Kong Polytechnic University, and a Ph.D. degree from the City University of Hong Kong. He is currently an Associate Professor at Durham University, researching computer graphics, deep learning, collaborative virtual environments, and educational technologies. He is also an Associate Editor of Frontiers in Education and an Editorial Board Member of Virtual Reality and Intelligent Hardware. He chaired conferences such as ISVC and ICWL.

Xiaohui Liang received his Ph.D. degree in computer science and engineering from Beihang University, China. He is currently a Professor, working in the School of Computer Science and Engineering at Beihang University. His main research interests include computer graphics and animation, visualization, and virtual reality