Efficient Physics Informed Dynamic Neural Fluid Fields Reconstruction From Sparse Videos

Yangcheng Xiang† and Yoshinori Dobashi†

†Graduate School of Information Science and Technology, Hokkaido University Kita 14 Nishi 9, Kita-ku, Sapporo, 060-0814 Japan

E-mail: †yangcheng.xiang.k0@elms.hokudai.ac.jp, †doba@ist.hokudai.ac.jp

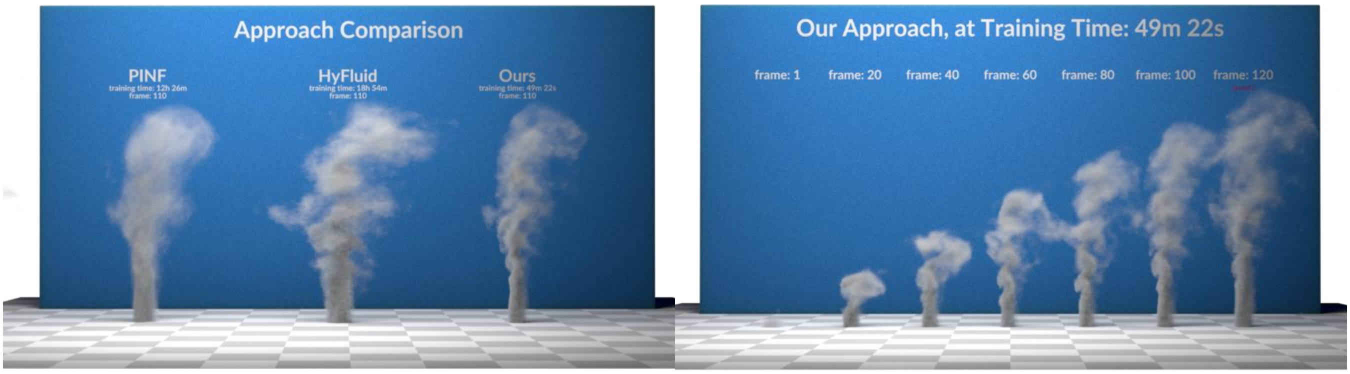

Figure 1: The left image compares the final reconstruction results of our method with the most related works, PINF and HyFluid. Our method achieves high-quality reconstruction in significantly less time. The right image presents the re-simulation of the reconstructed smoke dynamics with our method, where the result at frame 120 is obtained through prediction based on the previously reconstructed velocity field.

Abstract Efficiently inferring the latent physical properties of fluids from sparse 2D videos remains a significant challenge, particularly in scenarios with complex lighting conditions and occlusions. This work aims to reconstruct realistic neural density and velocity fields of fluids from sparse video inputs and achieve rapid training by leveraging known physical priors of fluid dynamics and preprocessed spatial information. Specifically, we introduce a preprocessing step to the traditional physics-informed neural reconstruction pipeline to reduce the scale of the inference problem associated with the density transport and divergence-free losses. To address the turbulent nature of fluid velocity and maintain long-term consistency between the reconstructed density and velocity fields, we design an inter-frame difference loss and an image-supervised vorticity confinement term to ensure visual consistency with the turbulent details present in the supervision video. Our approach achieves physically realistic fluid reconstruction results while significantly improving training efficiency, opening up new possibilities for faster fluid re-simulation, editing, future prediction, and neural dynamic scene synthesis.

Keyword Physics-Informed Neural Fields, Fluid Reconstruction, NeRF

1. INTRODUCTION

Fluid dynamics is a fundamental physical phenomenon widely observed in the real world, manifesting in diverse forms such as smoke, flames, clouds, and dye dispersal. In contrast to motions with low degrees of freedom and fixed shapes, such as rigid body dynamics, fluid flows exhibit extremely high degrees of freedom and intricate vortex structures, which makes accurate analysis and reconstruction from sparse 2D video observations particularly challenging.

Recent research in Physics-Informed Neural Fields (PINFs) [1] has shown great promise in fluid reconstruction tasks. These methods can recover fluid density and velocity fields from sparse-view videos, enabling data-driven fluid reconstructions. However, they often suffer from excessively long training and inference time, along with complex and computationally demanding workflows.

In this work, we fully leverage physical priors of fluid dynamics and preprocessing techniques to infer the latent physical information from sparse-view fluid videos in an efficient way. We first narrow the optimization space by rapidly estimating the density field distribution and leveraging spatial priors to significantly reduce the number of optimization parameters required for velocity field inference. Then, we employ inter-frame differences to ensure the effectiveness and stability of the velocity field over long time scales. During training procedure, we adopt an overlapping schedule scheme to accelerate the optimization process. Finally, we use video frames as supervision to enrich vortex details in the velocity field. O

Our approach significantly reduces the training time compared to traditional Physics-Informed reconstruction methods.

In summary, our contributions can be outlined as follows:

1. We introduce a fast-preprocessed coarse density field as a spatial prior and an overlapping scheduling scheme to accelerate the velocity estimation process.
2. We leverage inter-frame differences to enforce temporal consistency between the velocity and density fields, thereby enhancing the stability of extended re-simulations and improving the accuracy of future predictions.
3. We apply image-based vorticity confinement as a post-processing step to recover vortex details that were lost during the reconstruction process.

2. RELATED WORK

2.1. Fluid Reconstruction

Fluid reconstruction has been extensively studied in the fields of computer graphics and computer vision.

Sparse view reconstruction based on optimization theory represents a significant research direction. Gregson et al. (2012)[2] and Gregson et al. (2014)[3] proposed methods respectively for reconstructing fluid density fields and velocity fields using 3D tomography and linear imaging formation. However, these approaches require highly precise capture setups and multiple camera viewpoints, while also suffering from slow reconstruction speeds. Okabe et al. (2015)[4] utilized appearance transfer to enhance the realism of the reconstruction. Eckert et al. (2019)[5] employed joint reconstruction to simultaneously reconstruct fluid density and velocity fields, significantly enhancing the physical realism of the reconstruction. Franz et al. (2021)[6][15] introduced a differentiable simulation framework based on volumetric rendering, incorporating differentiable rendering and differentiable simulation into the optimization framework.

However, these traditional optimization-based methods typically assume prior knowledge of environmental lighting conditions or other physical scene information, making them challenging to directly apply to reconstruction tasks in real-world captured scenarios.

2.2. Neural Dynamics Fields Representation

NeRF (Neural Radiance Fields)[7] is a neural network-based method for synthesizing realistic 3D scenes and novel views by representing a scene's volumetric density and color as continuous functions using a neural network. Pumarola et al. (2021)[8] extended the application of NeRF to dynamic scenes, enabling the reconstruction of dynamic videos.

The original NeRF method is typically slow to train, prompting the development of numerous techniques to accelerate its training and inference. Instant NGP[9] significantly reduces training time by integrating hash encoding with an occupancy grid. Sun et al. (2022)[10] achieved faster training and inference speeds while maintaining reconstruction accuracy by directly training NeRF parameters on layered density grids.

The implicit representation of neural radiance fields, compared to traditional optimization-based methods, effectively encodes environmental lighting information, expanding the potential for realistic reconstruction of real-world data.

At the same time, the implicit representation of neural radiance fields offers unique advantages in compressing the size of physical fields. Kim et al. (2022)[11] proposed Neural VDB, which stores dynamic physical scene information in implicit neural radiance fields, significantly reducing model storage requirements while improving utilization efficiency.

2.3. Physics Informed Deep Learning

Recently, the integration of deep learning and physical priors for reconstructing fluid dynamics has emerged as a growing trend, injecting new vitality into the understanding and reconstruction of fluid motion.

Chu et al. (2022)[1] proposed Physics-Informed Neural Fields (PINF), which utilize physics-informed losses (Raissi et al., 2019)[12] and learned priors from synthetic data to reconstruct fluid flows from sparse-view videos. Building on this foundation, Yu et al. (2022)[13] proposed HyFluid, which can jointly reconstruct the neural radiance field of fluids from sparse-view videos, achieving promising reconstruction results. To address the issue of physical constraints breaking down in long-term fluid motion reconstruction, Wang et al. (2024)[14] introduced PICT, which preserves momentum conservation over extended periods by tracking fluid motion trajectories.

However, these methods invariably require extensive training and inference time, making them challenging to apply in industrial projects.

3. BACKGROUND CONTEXT

3.1. Incompressible Fluid Dynamics Priors

In general, natural fluid dynamics (smoke, fire, clouds, etc.) follow the Navier-Stokes equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla \cdot \nabla \boldsymbol{u} + \boldsymbol{f}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2}$$

Here, $\boldsymbol{u}$ is the velocity field, p is the pressure, ρ is the fluid density, ν is the kinematic viscosity, and $\boldsymbol{f}$ represents external forces, such as gravity or buoyancy forces. The first equation is the momentum equation, describing the conservation of linear momentum in a fluid, the second equation is the incompressibility condition, implying the conservation of mass for incompressible flows.

The Navier-Stokes equations reveal that general fluid behavior should adhere to key properties such as approximate incompressibility, mass conservation, and momentum conservation. Therefore, if we can fully exploit these physical priors, it will be highly beneficial for better understanding the underlying fluid dynamics in sparse video data and reconstructing realistic fluid motion.

3.2. Neural Radiance Field for Dynamics Scene

Given a set of images from multiple viewpoints along with the corresponding camera pose information, NeRF (Neural Radiance Fields) can leverage these data to reconstruct a neural radiance field containing the color and density information of the scene, enabling realistic novel view synthesis of complex scenes:

$$F_{\Theta}: (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma). \tag{3}$$

Here, the function F_{Θ} represents a neural network parameterized by Θ which maps the 3D spatial coordinates $\boldsymbol{x} = (x, y, z)$ of a point and the view direction $d = (\theta, \phi)$ to its corresponding radiance (color) $\boldsymbol{c} = (r, g, b)$ and volumetric density σ.

Using volumetric rendering techniques, NeRF aggregates the colors and densities along a given ray to compute the final pixel color:

$$\boldsymbol{C}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma\big(\boldsymbol{r}(t)\big)c(\boldsymbol{r}(t), \boldsymbol{d})\ dt, \tag{4}$$

where $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ represents a point along the ray, with $\boldsymbol{o}$ being the ray's origin and $\boldsymbol{d}$ its direction, $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))\ ds\right)$ denotes the accumulated transmittance, accounting for the probability of light traveling without obstruction up to depth t, $\sigma\big(r(t)\big)$ and $\boldsymbol{c}(r(t), d)$ are the volumetric density and emitted radiance (color) at the point $r(t)$, respectively.

D-NeRF extends the input and output framework to dynamic scenes, enabling the reconstruction of neural radiance fields for dynamic physical scenes:

$$f_{\Theta}: (\boldsymbol{x}(\tau), \tau, \boldsymbol{d}) \rightarrow (\sigma, c). \tag{5}$$

Here, τ represents the physical time of the input videos.

3.3. Physics-Informed Velocity Estimation

The density field is visible and can be directly inferred from multi-view videos using differentiable volumetric rendering. However, the velocity field is implicit and difficult to track visually. Since the motion of the density field is determined by this implicit velocity field, we can indirectly derive it using the density transport equation:

$$\nabla \cdot (\sigma \boldsymbol{u}) + \frac{\partial \sigma}{\partial t} = 0, \tag{6}$$

where σ denote the density field and $\boldsymbol{u}$ denote the velocity field. Note that this is a PDE, and the partial derivatives of each term can be computed separately. With the help of a differentiable nature of NeRF architecture and differentiable simulation solver, we can easily obtain the values of these partial derivatives during the learning process. In other words, we can introduce a Density Transport Loss to constrain the relationship between the velocity field and the density field:

$$\mathcal{L}_{\text{density\_transport}} = E_{\sigma,\boldsymbol{u},t}\left[\left|\nabla \cdot (\sigma \boldsymbol{u}) + \frac{\partial \sigma}{\partial t}\right|^2\right]. \tag{7}$$

4. APPROACH

Algorithm 1 Efficient Physics-Informed Dynamic Neural Fluid Fields Reconstruction From Sparse Videos

Input: multi-view smoke video with camera calibration
Output: Reconstructed neural density σ_{fine} and velocity fields $\mathbf{u}_{fine}$

Step 1: Preprocess $\sigma_{\mathbf{coarse}}$ and $\mathcal{O}(x, y, z)$
for each frame $t = 1$ to T **do**
  optimize $\sigma_{\text{coarse},t}$ by $\mathcal{L}_{render}$
  record $\mathcal{O}_t(x, y, z)$
end for
Compute the intersection of all $\mathcal{O}_t(x, y, z)$ for $t = 1$ to T: $\mathcal{O}_{\text{final}}(x, y, z) = \bigcap_{t=1}^{T} \mathcal{O}_t(x, y, z)$

Step 2: Estimate Velocity Fields $\mathbf{u}_{base}$
for each frame $t = 1$ to T **do**
  perform advection on $\sigma_{\text{coarse},t}$ over Δt
  optimize $\mathbf{u}_{base}$ by $\mathcal{L}_{velocity} = w_1\mathcal{L}_{transport} + w_2\mathcal{L}_{NSE} + w_3\mathcal{L}_{render,\mathbf{u}}$
  Update velocity field using Overlapping Schedule Scheme
end for

Step 3: Image-based Vorticity Confinement
for each frame $t = 1$ to T **do**
  Apply Helmholtz Hodge decomposition $\mathbf{u}_{\text{base, t}} = \nabla\phi_{\text{base, t}} + \nabla \times \mathbf{A}_{\text{base, t}} + \mathbf{h}_{\text{base, t}}$
  $\mathbf{A}_{\text{final},t} = (1 + w_t)\,\mathbf{A}_{\text{base},t}$
  perform advection on $\sigma_{\text{coarse},t}$ over Δt
  optimize w_t by $\mathcal{L}_{render}$
  $\mathbf{u}_{\text{final}} = \nabla\phi_{\text{base}} + \nabla \times \mathbf{A}_{\text{final}} + \mathbf{h}_{\text{base}}$
end for

Step 4: Re-Simulation
$\sigma_{\text{reconstructed},1} = \sigma_{\text{coarse},1}$
for each frame $t = 2$ to T **do**
  $\sigma_{\text{reconstructed},t} = \mathcal{A}(\sigma_{\text{reconstructed},t-1}, \mathbf{u}_{\text{final},t-1})$
end for=0

Algorithm 1: A pseudocode of our approach

4.1. Image-based Coarse Density Estimation

Density Rendering Loss. Recent methods[1][13] have demonstrated that NeRF-based approaches can successfully capture the approximate contours of translucent objects, separate scene components (e.g., colliders and background) from dynamic fluids, and achieve accurate rendering results from novel viewpoints. However, due to the inherent limitations of NeRF and the translucent and shapeless nature of smoke, the rendered results are often significantly blurred. Even increasing the resolution of the input video cannot resolve this issue.

Therefore, unlike previous methods, our approach only utilizes this coarse density output obtained by the common NeRF pipeline as a preprocessing result to accelerate the subsequent optimization process:

$$\mathcal{L}_{\text{render}} = E_{\boldsymbol{o},\boldsymbol{d}}\left[\left|\boldsymbol{C}_{\text{render}}(\boldsymbol{o},\boldsymbol{d}) - \boldsymbol{C}_{\text{image}}(\boldsymbol{o},\boldsymbol{d})\right|^2\right], \quad (7)$$

where $\boldsymbol{C}_{\text{render}}(\boldsymbol{o},\boldsymbol{d})$ is our volume rendered values by Eqn. 4 and $\boldsymbol{C}_{\text{image}}(\boldsymbol{o},\boldsymbol{d})$ is sampled from video frames. Notably, since the density field of fluids is typically physically continuous, it is important to avoid the random sampling strategy commonly used in standard NeRF methods. Instead, training should be conducted sequentially frame by frame to ensure good temporal continuity in the training results.

Training Acceleration Structures. The original NeRF requires constructing a deep MLP, which results in extremely long training and inference times. Instant-NGP, an accelerated extension of NeRF, reduces the training time for neural radiance fields from several hours to just a few minutes. Inspired by Instant-NGP, we extend its core components, including the Multi-Resolution Hash Encoder and Occupancy Grid acceleration structure, into forms suitable for dynamic scenes.

First, the space and time dimensions are subdivided into multiresolution to obtain a 4D input tensor $\boldsymbol{v} = [\text{x}, \text{y}, \text{z}, \text{t}]^T$. In our case, we construct 16 levels ranging from $[16, 16, 16, 16]$ to $[256, 256, 256, 128]$ as multi resolution grids.

Then, we apply the following hash function to encode the input tensor and pass the encoded tensor $h(\boldsymbol{v})$ into a shallow MLP:

$$h(\boldsymbol{v}) = \left(\bigoplus_{i=1}^{d} v_i \pi_i\right) mod\ T, \quad \boldsymbol{v} = (x, y, z, t), \quad (8)$$

where $\bigoplus$ denotes the bit-wise XOR operation, T is the max hash map size, and π_i are unique, large prime numbers, we use $[1, 2654435761, 805459861, 3674653429]$ in our case.

As shown in Figure 2, the use of a Multi-Resolution Hash Encoder significantly reduces the hidden layers of the MLP in the original NeRF, thereby greatly accelerating the model's training and convergence speed. For further details about Multi-Resolution Hash Encoder and Occupancy Grid, please refer to original Instant-NGP[9].

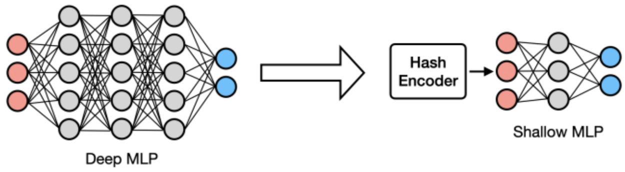

Figure 2: Reduced MLP size by applying Multi-Resolution Hash Encoder

During training, corresponding Occupancy Grids are constructed to represent the density distribution in space, enhancing the ratio of effective training. Upon completion of preprocessing training, these dynamic occupancy grids are combined into one for subsequent velocity estimation training as shown in Figure 3.

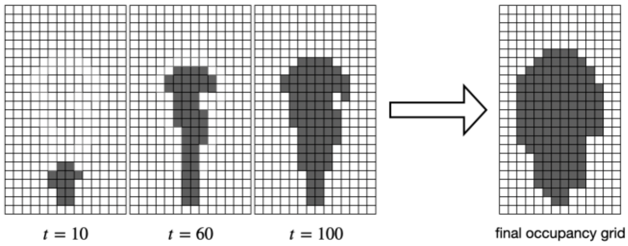

Figure 3: Combination of dynamic occupancy grids

The coarse density field σ_{coarse} and occupancy grid $\mathcal{O}(x, y, z)$ obtained through the aforementioned method are used as preprocessing results, serving as additional supervision to reduce the degrees of freedom of the subsequent velocity estimation model.

4.2. Physics Informed Velocity Estimation

Adaptive Density Transport Loss. PINN model proposed by Raissi et al.[13] demonstrates that deep learning models can be trained as data-driven solutions of physical problems via optimizing the governing PDEs. For fluid dynamics, they trained a model, $\mathcal{F}_{\text{fluid}}: (x, y, z, t) \to (d, \boldsymbol{u}, p)$, to replace the Navier-Stokes equations with unknown parameters for predicting future simulation results. Recent studies (PINF[1] and HyFluid[13], etc.) have focused on optimizing the density transport equation to enhance the correlation between the velocity field and the density field at adjacent time steps ($\hat{\sigma}_{t+1} = \mathcal{A}(\sigma_t, \boldsymbol{u}_t)$, where $\mathcal{A}$ means advection operators).

Like these approaches, we also employ the density transport equation (Eqn. 9) with importance sampling (Eqn. 10) to efficiently obtain a basic a baseline velocity field $\boldsymbol{u}_{base}$ corresponding to the input coarse density field σ_{coarse}.

$$\mathcal{L}_{\text{transport}} = E_{\sigma,\boldsymbol{u},t}\left[\left|\nabla \cdot (\sigma\boldsymbol{u}) + \frac{\partial\sigma}{\partial t}\right|^2\right]. \quad (9)$$

In practice, we first perform spatial discrete sampling on velocity network $\mathcal{F}_{\text{vel},\theta}(\boldsymbol{x})$ at the current training time t_{train}. In our case, we utilize a discrete spatial resolution of [256,256,256] for sampling. Then, through automatic differentiation of the neural network, we can obtain the partial derivatives at each sampled point as described in Eqn. 9. Finally, these values are accumulated to compute the final $\mathcal{L}_{\text{transport}}$. Unlike HyFluid, since in practical scene, a lot of spatial positions usually lack density distributions, the velocity values there actually negligible for the final density transport loss $\mathcal{L}_{\text{transport}}$. Therefore, in our method, to reduce this computational overhead, we filter out the partial derivative values at these positions using spatial information from the occupancy grid $\mathcal{O}(x,y,z)$:

$$\mathcal{L}_{\text{transport}}(\boldsymbol{x}_{\text{sampled}}) = \begin{cases} E_{\sigma,\boldsymbol{u},t}\left|\nabla\cdot(\sigma\boldsymbol{u}) + \frac{\partial\sigma}{\partial t}\right|^2, \text{if } \mathcal{O}(\boldsymbol{x}_{\text{sampled}}) = 0, \\ 0,\ otherwise. \end{cases} \quad (10)$$

Occupancy Grid Bounded NSE Loss. To ensure that the motion of the velocity field complies with the fluid dynamics principles, we introduce $\mathcal{L}_{\text{NSE}}$ (loss of Navier-Stokes Equations) introduced in section 3.1:

$$\mathcal{L}_{\text{NSE}} = E_{u,t}\left[\left|\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \frac{\partial\boldsymbol{u}}{\partial t}\right|^2\right] + \lambda_{div}E_u\left[\left|\boldsymbol{u} - \boldsymbol{u}_p\right|^2\right] \quad (11)$$

where λ_{div} is a hyper-parameter that controls the incompressibility of the fluid, $\boldsymbol{u}_p$ is the projected velocity field.

However, because $\mathcal{L}_{\text{transport}}$ restricts the solution domain to the occupancy grid, optimizing the velocity field across the entire sampling domain using the NSE loss, as done in PINF and HyFluid, may result in uncontrolled velocity fields in regions without density distribution. This lack of control can lead to difficulties in convergence. To address this issue, in our method, regions where $\mathcal{O}(x,y,z) = 0$ are treated as Dirichlet Boundaries during the computation of pressure projection and velocity field advection. Therefore, the reconstructed $\boldsymbol{u}_{base}$ will ultimately exist only within the space defined by $\mathcal{O}(x,y,z)$.

Inter-frame Rendering Difference Loss. In the preceding steps, we can generate a velocity field that aligns with fluid motion based on the preprocessed coarse density field σ_{coarse}. However, due to the accumulation of model errors and numerical errors, the generated velocity field cannot guarantee correctness and stability over long time scales. Therefore, in this step, we utilize the input videos and inter-frames difference images as supervision to ensure that the velocity field correctly advects the density field over long time scales:

$$\mathcal{L}_{\text{render},\,\boldsymbol{u}} = \mathcal{L}_{\text{advection}} + \lambda_{diff}\mathcal{L}_{\text{IRD}}. \quad (12)$$

$$\text{where}\begin{cases} \mathcal{L}_{\text{advection}} = E_{\boldsymbol{o},\boldsymbol{d}}\left[\left|C_{render,\ t+\Delta t}(\boldsymbol{o},\boldsymbol{d}) - C_{\text{image},t+\Delta t}(\boldsymbol{o},\boldsymbol{d})\right|^2\right] \\ \mathcal{L}_{IRD} = E_{\boldsymbol{o},\boldsymbol{d}}\left|\left(C_{\text{render},t+\Delta t}(\boldsymbol{o},\boldsymbol{d}) - C_{\text{render},t}(\boldsymbol{o},\boldsymbol{d})\right) - \left(C_{\text{image},t+\Delta t}(\boldsymbol{o},\boldsymbol{d}) - C_{\text{image},t}(\boldsymbol{o},\boldsymbol{d})\right)\right|^2 \end{cases}$$

Here, $C_{render,t+\Delta t}$ means volume rendering result of the advected density $\sigma(t+\Delta t) = \mathcal{A}_m \circ \dots \circ \mathcal{A}_2 \circ \mathcal{A}_1(\sigma, u(t))$, $C_{image,t+\Delta t}$ means the ground truth image sampled by the input video. The same meanings apply to other similar parameters.

Next, we provide a detailed explanation of the specific meaning of these two components of the loss. The objective of $\mathcal{L}_{\text{advection}}$ is to ensure that advecting the current density σ_t multiple times to reach the final density $\sigma_{t+\Delta t}$ aligns with the results of the supervised images. $\mathcal{L}_{IRD}$ focuses on supervising the density changes between two frames. This differential supervision facilitates the convergence of the correct velocity field.

In practice, a too large Δt can lead to difficulties in convergence, while a too small Δt can result in instability during long-term simulations. In our experiments, we set Δt to three times the frame rate. Inspired by Treuille et al.[16], we adopted an Overlapping Schedule Scheme to accelerate training process.

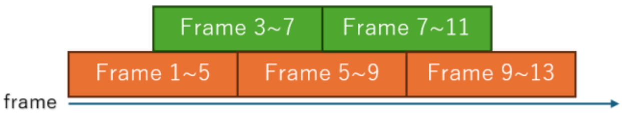

Figure 4: Overlapping Schedule Scheme.

As shown in Figure 4, when we choose Δt as five times the frame interval, the execution order of the Overlapping Schedule Scheme is as follows: Starting with the density field of the first frame, the advection operation is performed four times to reach the fifth frame. The difference between the predicted results and the ground truth is evaluated to compute the loss, followed by density optimization. Then, the optimized density field of the third frame is used as the starting point, performing the advection operation four times to reach the seventh frame. The loss is again calculated based on the difference from the ground truth, followed by optimization. This process is repeated iteratively.

4.3. Image-based Vorticity Confinement

In the previous sections, we utilized the coarse density field σ_{coarse} to infer u_{base}. However, the simulation results at this stage are often overly smooth and lack the vortex details commonly observed in fluid dynamics. Recovering accurate vortex structures from blurry and sparse perspective videos is nearly an impossible task. Therefore, inspired by Sato et al.[17], we propose an image-supervised vorticity confinement method to enhance $\boldsymbol{u}_{\text{base}}$ with physically realistic and visually plausible vortex details.

Using the Helmholtz Hodge decomposition, any smooth vector field u can be separated into a irrotational (curl-free) field $\nabla\phi$, a solenoidal (divergence-free) field $\nabla\times\boldsymbol{A}$ and a harmonic field $\boldsymbol{h}$:

$$\boldsymbol{u} = \nabla\phi + \nabla\times\boldsymbol{A} + \boldsymbol{h}. \quad (13)$$

The velocity field $\boldsymbol{u}_{\text{base}}$ obtained through the algorithm in Section 4.2 should be approximately divergence-free but insufficiently curl-rich. Therefore, we first perform a Helmholtz-Hodge decomposition on this velocity field:

$$\boldsymbol{u}_{\text{base}} = \nabla\phi_{\text{base}} + \nabla\times\boldsymbol{A}_{\text{base}} + \boldsymbol{h}_{\text{base}}. \quad (14)$$

Next, our target is to optimize the solenoidal field $\boldsymbol{A}$ such that the rendering results of the density field reconstructed through the velocity field align as closely as possible with the supervised video. To simplify the complexity of this task, we introduce a learnable weight field w_t to optimize the solenoidal field:

$$\boldsymbol{A}_{\text{final},t} = (1 + w_t)\boldsymbol{A}_{\text{base},t}, \quad (15)$$

where the value of w_t is significantly smaller compared to the norm of $A_{\text{base},t}$. It is worth noting that simply scaling A_{base} does not change the original velocity field's motion tendencies or its divergence-free property. The final velocity field will be represented as:

$$\boldsymbol{u}_{\text{final}} = \nabla\phi_{\text{base}} + \nabla\times\boldsymbol{A}_{\text{final}} + \boldsymbol{h}_{\text{base}}. \quad (16)$$

Finally, we use the input video as supervision to train w_t:

$$\mathcal{L}_{\text{vort}} = E_{\boldsymbol{o},\boldsymbol{d}}\left[\left|C_{\mathcal{A}(\sigma,u_{\text{final}})}(\boldsymbol{o},\boldsymbol{d}) - C_{\text{image}}(\boldsymbol{o},\boldsymbol{d})\right|^2\right] \quad (17)$$

5. EVALUATION

5.1. Datasets

Figure 5: Real-world smoke training dataset from five viewpoints (background preprocessed to black, sourced from ScalarFlow)

For evaluation, we utilize real-world recordings from the ScalarFlow dataset (Eckert et al., 2019) [5] as seen in Figure 5, which captures buoyancy-driven rising smoke plumes from the real world. During the capture, five fixed cameras are evenly distributed along a 120° arc centered on the smoke plume. Each video comprises 120 frames at a resolution of 1080 × 1920, with post-processing applied to remove the background.

In our experiments, we select the first five scenes from the dataset. we use frames 0–110 from the first four videos as the training set and frames 0–110 from the fifth video as the novel-view test set. Additionally, frames 110–120 are designated as the future prediction test set in re-simulation, serving to evaluate the reliability of the reconstructed velocity field.

We use this training set to comprehensively evaluate our method against the most related works, PINF and HyFluid, in terms of reconstruction quality, training time, and re-simulation performance.

5.2. Density and Velocity Reconstruction

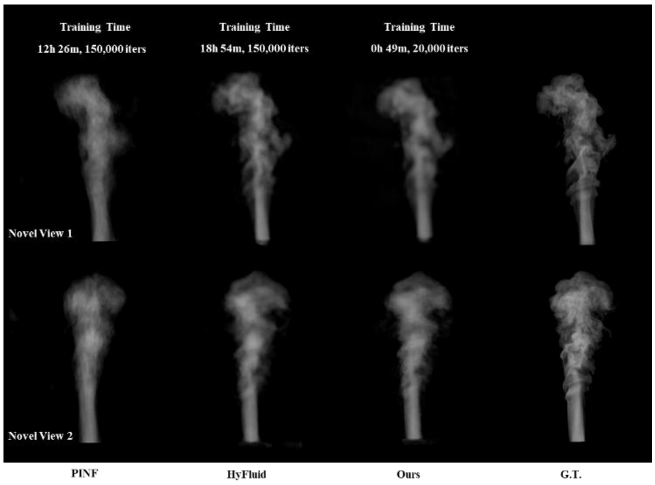

Figure 6: Comparisons of the trained density render results and training time of PINF/HyFluid/Ours and G.T. images from two novel views.

In Figure 6, we compare our method with PINF and HyFluid in terms of reconstruction quality and reconstruction speed. First, our method, along with PINF and HyFluid, is able to recover the overall shape of the smoke. However, PINF exhibits a significant lack of detail in both edge definition and vortex structures. HyFluid achieved a reconstruction with rich details comparable to the Ground Truth, while our method achieved similar results compared to HyFluid while requiring only 1/23 of the training time.

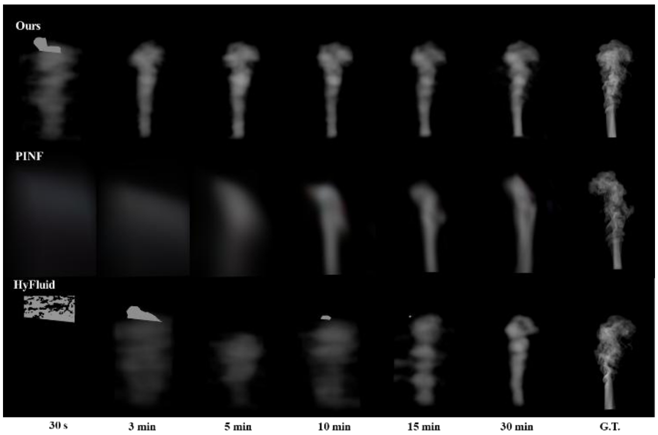

Figure 7: Comparison of convergence speed for Ours, PINF, and HyFluid under the same training time.

As shown in Figure 7, our model rapidly converges to a reasonable smoke contour within the same training time. In contrast, both PINF and HyFluid struggle to achieve fast convergence. These intermediate reconstruction results demonstrate that the preprocessing stage in our method significantly accelerates the convergence of the physics-informed neural model by substantially reducing the initial problem's degrees of freedom.

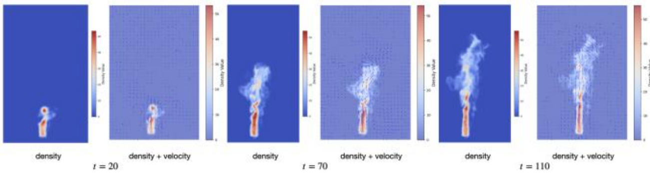

Figure 8: Reconstructed smoke velocity fields with density background in the validation view (2D slice at the midplane along the z-axis)

As shown in Figure 8, in the "velocity fields with density background" visualization, the reconstructed velocity fields are primarily concentrated in regions with significant density motion, and the vortex structures in the velocity field are clearly captured. From the density maps, our reconstruction effectively estimates the location of the density source. This demonstrates that neural radiance fields achieve satisfactory results in both density estimation and depth estimation for fluid fields.

5.3. Re-simulation and Future Prediction

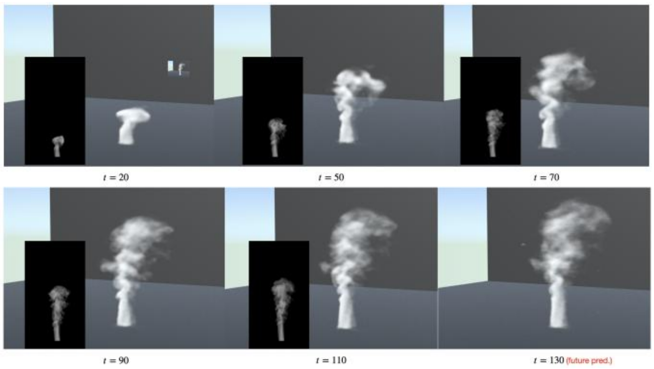

Figure 9: Re-simulate the smoke plume using the reconstructed velocity and density fields with the bottom left showing the ground truth frame for the current view, and the last image predict the future smoke density at t=130.

We performed a re-simulation of the reconstructed density and velocity fields in a new scene, as well as predictions of future dynamics. As shown in Figure 9, the reconstructed velocity field is sufficiently smooth, and the density field largely aligns with the ground truth in the bottom-left corner. The smoke morphology and motion visually conform to real-world laws.

Although our training data is limited to 120 frames, the predicted results for the 130th frame in the new scene remain robust. Additionally, we can apply external forces (e.g., wind or scene collisions) in the new scenarios to create smoke effects that better fit the environment.

As shown in Figure 1, we also provide an additional test scenario. The left image compares the final reconstruction results of our method with the most related works, PINF and HyFluid. Our method achieves high-quality reconstruction in significantly less time. The right image presents the re-simulation of the reconstructed smoke dynamics with our method, where the result at frame 120 is obtained through prediction based on the previously reconstructed velocity field. Compared to PINF and HyFluid, our method achieves superior results in terms of training time and future prediction.

Therefore, our method shows significant potential for understanding, perceiving, and reconstructing fluid dynamics.

5.4. Ablations

Our experiments are conducted in an environment equipped with an Intel i9-10980XE CPU, 256 GB of memory, and two NVIDIA RTX 6000 GPUs. The experimental results obtained are presented below.

First, we compared commonly used fluid dynamics data storage formats and quantitatively evaluated the storage compression ratio achieved by implicit neural radiance fields for representing dynamic fluid scenes. As shown in Table 1, the storage size of dynamic fluid scenes using implicit neural radiance fields is less than 10% of that of traditional compression formats. Even compared to VDB, the most widely used fluid storage format in the industry, implicit neural radiance fields exhibit remarkable compression efficiency.

Methods	Size (1) Min	Size (1) Max	Size (All)	Opt. Ratio (1) Avg.	Opt. Ratio (All)
Dense Grids (origin)	78,6433 KB	78,6433 KB	92,160.01 MB	1.00	1.00
Compressed Grids	2,104 KB	26,372 KB	2,044.89 MB	0.181	0.022
VDB	1,317 KB	42,292 KB	1,955.84 MB	0.276	0.021
Neural Fields (ours)	1,638 KB	1,638 KB	196,637 KB	0.021	0.002

Table 1: Comparison of Storage Methods for Density Fields under resolution 512x768x512

Next, we compared our method with PINF[1] and HyFluid[13], which are the two most closely related works to this study. See Table 2 and Figure 7, under the premise of achieving similar loss, PSNR values, and reconstruction quality, our method is nearly 15 times faster than PINF, and approximately 10 times faster than HyFluid. Moreover, given the same training time, our method better preserves temporal consistency and achieves a closer fit to the ground truth.

Method	Input Video Res	PSNR ↑	$\mathcal{L}_{\text{RGB}}$ (AVE.) ↓	TRN TIME	Speedup
PINF	540x960	24.51	0.00191	12.4 hr	1x
HyFluid	540x960	38.12	0.00018	18.9 hr	0.65x
Ours (High Res)	1080x1920	36.88	0.00019	2 hr 30 min	5x
Ours (Low Res)	540x960	35.11	0.00022	49 min	15x

Table 2: Performance comparison of PINF, HyFluid, and ours, with speedup relative to PINF.

6. SUMMARY AND LIMITATIONS

In this paper, we propose an efficient approach for reconstructing fluid dynamics from sparse multi-view videos by leveraging physical priors, demonstrating significant potential in novel-view re-simulation and fluid future prediction.

However, the proposed approach has several limitations. First, constrained by the characteristics of Neural Radiance Fields, the current method faces challenges in generalizing to complex real-world scenes, making it difficult to extract and interpret physical dynamics from real captured data. Second, the method assumes fluid motion without scene collisions, and in scenarios with complex collision interactions, the lack of sufficient physical priors can significantly degrade reconstruction quality. Lastly, achieving high-fidelity reconstruction requires higher resolution input videos and a larger number of viewpoints, which exponentially increases the training time. Therefore, exploring approaches to guide high-fidelity synthesis from low-sampling reconstruction results is a promising direction for future research.

REFERENCE

[1] Chu, Mengyu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt and Rhaleb Zayer. "Physics informed neural fields for smoke reconstruction with sparse data." ACM Transactions on Graphics (TOG) 41 (2022): 1 - 14.

[2] Gregson, James, Michael Krimerman, Matthias B. Hullin and Wolfgang Heidrich. "Stochastic tomography and its applications in 3D imaging of mixing fluids." ACM Transactions on Graphics (TOG) 31 (2012): 1 - 10.

[3] Gregson, James, Ivo Ihrke, Nils Thürey and Wolfgang Heidrich. "From capture to simulation." ACM Transactions on Graphics (TOG) 33 (2014): 1 - 11.

[4] Okabe, Makoto, Yoshinori Dobashi, Ken-ichi Anjyo and Rikio Onai. "Fluid volume modeling from sparse multi-view images by appearance transfer." ACM Transactions on Graphics (TOG) 34 (2015): 1 - 10.

[5] Eckert, Marie-Lena, Kiwon Um and Nils Thuerey. "ScalarFlow." ACM Transactions on Graphics (TOG) 38 (2019): 1 - 16.

[6] Franz, Erik, Barbara Solenthaler and Nils Thuerey. "Global Transport for Fluid Reconstruction with Learned Self-Supervision." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 1632-1642.

[7] Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi and Ren Ng. "NeRF." Communications of the ACM 65 (2020): 99 - 106.

[8] Pumarola, Albert, Enric Corona, Gerard Pons-Moll and Francesc Moreno-Noguer. "D-NeRF: Neural Radiance Fields for Dynamic Scenes." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020): 10313-10322.

[9] Müller, Thomas, Alex Evans, Christoph Schied and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding." ACM Transactions on Graphics (TOG) 41 (2022): 1 - 15.

[10] Sun, Cheng, Min Sun and Hwann-Tzong Chen. "Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 5449-5459.

[11] Kim, Doyub, Minjae Lee and Ken Museth. "NeuralVDB: High-resolution Sparse Volume Representation using Hierarchical Neural Networks." ACM Transactions on Graphics 43 (2022): 1 - 21.

[12] Raissi, Maziar, Paris Perdikaris and George Em Karniadakis. "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.

[13] Yu, Hong-Xing, Yang Zheng, Yuan Gao, Yitong Deng, Bo Zhu and Jiajun Wu. "Inferring Hybrid Neural Fluid Fields from Videos." ArXiv abs/2312.06561 (2023): n. pag.

[14] Wang, Yiming, Siyu Tang and Mengyu Chu. "Physics-Informed Learning of Characteristic Trajectories for Smoke Reconstruction." International Conference on Computer Graphics and Interactive Techniques (2024).

[15] Franz, Erik, Barbara Solenthaler and Nils Thürey. "Learning to Estimate Single-View Volumetric Flow Motions without 3D Supervision." ArXiv abs/2302.14470 (2023): n. pag.

[16] Treuille, Adrien, Antoine McNamara, Zoran Popovic and Jos Stam. "Keyframe control of smoke simulations." ACM SIGGRAPH 2003 Papers (2003): n. pag.

[17] Syuhei Sato, Yoshinori Dobashi and Theodore Kim. "Stream-guided smoke simulations." ACM Transactions on Graphics (TOG) 40 (2021): 1 - 7.