Global Transport for Fluid Reconstruction with Learned Self-Supervision

Erik Franz
Technical University of Munich
erik.franz@tum.de

Barbara Solenthaler
ETH Zurich
solenthaler@inf.ethz.ch

Nils Thuerey
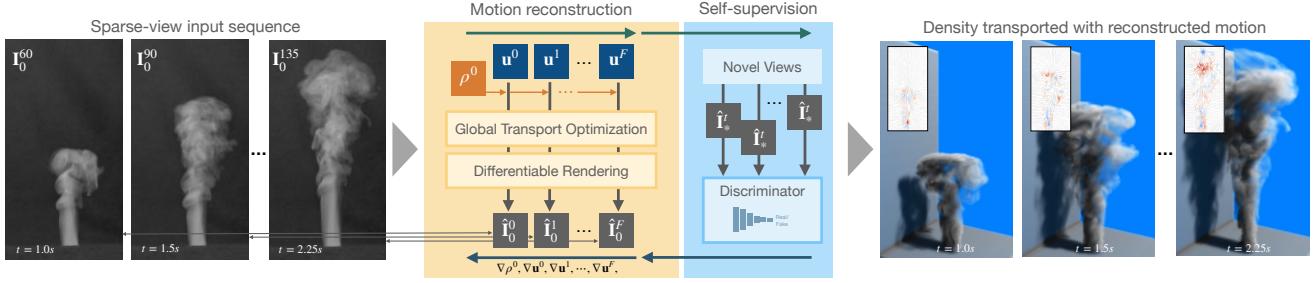Technical University of Munich
nils.thuerey@tum.de

Figure 1: We propose a novel algorithm that reconstructs the motion $\mathbf{u}^{0..F}$ of a single initial state ρ^0 over the full course of F frames of an input sequence, *i.e.*, its *global transport*. Based on a learned self-supervision, our algorithm yields a realistic motion for highly under-constrained scenarios such as a single input view.

Abstract

We propose a novel method to reconstruct volumetric flows from sparse views via a global transport formulation. Instead of obtaining the space-time function of the observations, we reconstruct its motion based on a single initial state. In addition we introduce a learned self-supervision that constrains observations from unseen angles. These visual constraints are coupled via the transport constraints and a differentiable rendering step to arrive at a robust end-to-end reconstruction algorithm. This makes the reconstruction of highly realistic flow motions possible, even from only a single input view. We show with a variety of synthetic and real flows that the proposed global reconstruction of the transport process yields an improved reconstruction of the fluid motion.

1. Introduction

The ambient space for many human activities and manufacturing processes is filled with fluids whose motion we can only observe indirectly [53, 61, 72, 79]. Once passive markers, *e.g.*, in the form of ink or dye, are injected into the fluid, it is possible to draw conclusions about the transport induced by the motion of the fluid. This form of motion reconstruction is highly important for a large variety of applications, from medical settings [59], over engineering [67]

to visual effects [15], but at the same time poses huge challenges.

Fluids on human scales are typically turbulent, and exhibit a highly complex mixture of translating, shearing, and rotating motions [60]. In addition, the markers by construction need to be sparse or transparent in order not to fully occupy and occlude the observed volume. Several works have alleviated these challenges with specialized hardware [18, 77] or by incorporating the established physical model for fluids, the *Navier-Stokes* equations, into the reconstruction [11, 21]. However, despite improvements in terms of quality of the reconstruction, the high non-linearity of the reconstruction problem coupled with ambiguous observations can cause the optimizations to find undesirable minimizers that deviate from ground truth motions.

In order to obtain a solution, existing approaches compute volumetric observations over time [21, 9, 79]. Hence, despite including a physical model, the observed quantities represent unknowns per time step, and are allowed to deviate from the constraints of the model. We make a central observation: when enforcing the physical model over the course of the complete trajectory of the observed fluid, the corresponding reconstructions yield a motion that better adheres to the ground truth. Thus, instead of a sequence, our reconstruction results in a *single initial state* of the density. This state is evolved via a *global transport* over time purely by the physical model and the temporal reconstruction of the motion field. In addition to an improved reconstruction of the motion, this strict differentiable physics prior [7, 28]

Source code: https://github.com/tum-pbs/Global-Flow-Transport

allows us to work with very sparse observations, with only a single viewpoint in the extreme.

In order to better constrain the degrees of freedom in this single-viewpoint scenario, we propose a method inspired by generative adversarial networks (GANs) [19, 76]. Due to the complete lack of observations from other viewpoints, we make use of a small dataset of example motions, and train a convolutional neural network alongside the motion reconstruction that serves as a *discriminator*. This discriminator is evaluated for randomly sampled viewpoints in order to provide image space constraints that are coupled over time via our global transport formulation.

While existing works also typically focus on linear image formation models [30, 22], we combine the visual and transport constraints with a fully differentiable volumetric rendering pipeline. We account for complex lighting effects, such as absorption and self-shadowing, which are key elements to capture the visual appearance of many real-world marker observations.

To summarize, the main contributions of our work are:

- A global multi-scale transport optimization via a differentiable physical model that yields purely transport-based reconstructions.

- A learned visual prior to obtain motions from very sparse and single input views.

- The inclusion of differentiable rendering with explicit lighting and volumetric self-shadowing.

To the best of our knowledge, these contributions make it possible for the first time to construct a fluid motion from sparse views in an end-to-end optimization, even from a single viewpoint. An overview is given in Figure 1.

2. Related Work

Flow Reconstruction Reconstructing fluid flows from observations has a long history in science and engineering. A wide variety of methodologies have been proposed, ranging from Schlieren imaging [6, 2, 1], over particle imaging velocimetry (PIV) methods [20, 12], to laser scanners [24, 16] and structured light [23] and light path [33] approaches. Especially PIV has been widely used, and seen a variety of extensions and improvements, such as synthetic apertures [3], specialized lighting setups [77], and algorithms for custom hardware setups [14]. In the following, we focus on visible light capturing approaches, as they avoid the reliance on specialized, and often expensive hardware.

This capturing modality is especially interesting in conjunction with sparse reconstructions, *e.g.*, medical tomography settings often favor view sparsity [68, 4, 66]. Here, the highly under-constrained setting of a large number of degrees of freedom of a dense grid to be reconstructed from a very small number of input views is highly challenging for reconstruction algorithms. One avenue to obtain a reduced solution space is to introduce physical priors. For fluids, the *Navier-Stokes equations* [60] represent a well-established physical model for incompressible fluids.   *E.g.*, the adjoint method was used for gradient-based optimization of volumetric flows, such as fluid control [48]. On the other hand, Gregson *et al.* [21] introduced a multi-view convex optimization algorithm that makes use of a discrete advection step and a divergence-free projection. This approach was extended to incorporate view-interpolation to obtain additional constraints [79]. Convex optimization was also used for the reconstruction from sparse views [9], or specialized single view reconstruction [8]. However, this requires a hand-crafted regularizer to suppress depth-aligned motions. In contrast, we rely on gradients provided by deep-learning methods for our reconstructions. And while several of these methods introduce transport terms for inter-frame constraints, none introduces an "end-to-end" constraint that yields a single, physical transport process from an initial state to the last observed frame. Such gradient propagation through PDEs is also being explored in differentiable physical simulation [28].

3D Reconstruction and Appearance Solving the tomography for static scattering volumetric media has been studied via inverse scattering [17] or for large-scale cloud-scapes [43]. Although we target volumetric reconstruction, our goals also partially align with settings where a clearly defined surface is visible, *e.g.*, reconstructions of 3D geometry [54, 42], including its deformation [78]. Mesh-based methods were proposed to create a deformation of a labeled template mesh [35], or of spherical prototypes [37]. While marker density volumes do not exhibit a clear surface, they share similarities with voxel-based reconstructions [58, 52]. In addition, learned representations [69, 45, 70, 51] have the flexibility to encode information about transparent materials and their reflectance properties, and temporally changing deformations [50, 55].

In conjunction with sparse-view reconstructions, regularization becomes increasingly important to obtain a meaningful solution. One line of work has employed different forms of appearance transfer [10, 32], *e.g.*, to match the histograms of fluid reconstructions [57], while others have proposed view interpolations via optical flow [79]. In the context of style transfer for natural images, learned approaches via GANs [19, 62] were shown to be especially powerful [81, 36]. GANs were likewise employed in fluid synthesis settings [76], and we propose a learned discriminator that works alongside a single flow reconstruction process.

A similar idea was used for single-view tomographic reconstruction of static objects: The approach by Henzler *et al.* [26] was extended to include a learned discriminator for

a specific class of objects [25] to constrain unseen views. We extend this approach to sequences, and show that the concept of learned self-supervision provides a highly useful building block for physical reconstructions.

Differentiable Rendering   Computing derivatives of pixels in an image with respect to input variables is essential for many inverse problems. Several fast but approximate differentiable rendering methods have been presented that model simple light transport effects [46, 38]. Physics-based neural renderers account for secondary effects, such as shadows and indirect light. Li *et al.* [44] presented the first general-purpose differentiable ray tracer, and Nimier *et al.* [56] a versatile MCMC renderer that was, among other examples, applied to smoke densities. Neural renderers [71] were used for 3D reconstruction problems, *e.g.*, to render an explicit scene representation [25] or to synthesize novel views [51].

In the context of fluid simulations, differentiable renderers were used with and without differentiable solvers to optimize robotic controllers [63], to initialize a water wave simulator [29], and to transfer a style from an input image onto a 3D density field [39, 40]. Additive, *i.e.*, linear, lighting models were often used for flow reconstruction [9, 79]. They are suitable for thin marker densities that have little absorption under uniform lighting conditions. In contrast, our method employs a differentiable rendering model that handles non-linear attenuation and self-shadowing.

3. Method

The central goal of our algorithm is to reconstruct a space-time sequence of volumetric fluid motions $\mathbf{u}$ such that a passively transported field of marker density ρ matches a set of target images. These targets are given as $c \in \boldsymbol{C}$ calibrated input views $\boldsymbol{I}_c^t$ for a time sequence of $\boldsymbol{F}$ frames with $\{t | t \in \mathbb{N}_0, t < \boldsymbol{F}\}$. We represent ρ and $\mathbf{u}$ as dense Eulerian grids, which are directly optimized through iterative gradient updates calculated from the proposed loss functions and propagated through the full sequence to achieve a solution that adheres to a global transport formulation.

3.1. Physical Prior

We derive our method on the basis of a strong physical prior given by the Navier-Stokes equations that describe the motion of an incompressible fluid:

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} \text{ , s.t. } \nabla \cdot \mathbf{u} = 0. \quad (1)$$

In addition to the fluid velocity $\mathbf{u}$, p denotes the pressure and ν the kinematic viscosity. We implement these equations via a differentiable transport operator $\mathcal{A}(s, \mathbf{u})$ that can advect a generic field s (can be ρ or $\mathbf{u}$) using a velocity field $\mathbf{u}$. For the discretized operator $\mathcal{A}$ we use a second-order transport scheme to preserve small-scale content on the transported fields [65]. Details are in Appendix A.3.

Global Density Transport Optimization   Given an initial density ρ^0, the state ρ^t at time t is given by

$$\rho^t := \mathcal{A}^t(\rho^0) = \mathcal{A}(\mathcal{A}(\mathcal{A}(\rho^0, \mathbf{u}^0), \mathbf{u}^1) \ldots, \mathbf{u}^{t-1}) \quad (2)$$

and we only optimize ρ^0 and the velocity sequence $\{\mathbf{u}^0, \ldots, \mathbf{u}^{t-1}\}$, instead of optimizing all ρ^t individually, as done by previous approaches. With this formulation we enforce a correct transport over the entire fluid flow trajectory, resulting in a reconstructed motion that better adheres to Equation (1). For optimization, all ρ^t are constrained by a composite loss $\mathcal{L}_\rho$, that we detail in Section 4. Thus, the gradient for ρ^0 is the sum of derivatives of $\mathcal{L}_\rho$, of all frames $t \in \boldsymbol{F}$, w.r.t. to the density of the first frame:

$$\nabla \rho_{\mathcal{A}}^0 = \sum_{t=0}^{\boldsymbol{F}} \frac{\partial \mathcal{L}_\rho(\rho^t)}{\partial \rho^0} = \sum_{t=0}^{\boldsymbol{F}} \frac{\partial \mathcal{A}^t(\rho^0)}{\partial \rho^0} \frac{\partial \mathcal{L}_\rho(\mathcal{A}^t(\rho^0))}{\partial \mathcal{A}^t(\rho^0)}. \quad (3)$$

Back-propagation of the gradients $\nabla \rho^{t+1}$ through every transport step $\mathcal{A}(\rho^t, \mathbf{u}^t)$ takes the iterative form

$$\nabla \rho_{\mathcal{A}}^t = \nabla \rho^t + \frac{\partial \mathcal{A}(\rho^t, \mathbf{u}^t)}{\partial \rho^t} \nabla \rho_{\mathcal{A}}^{t+1} \quad (4)$$

where $\nabla \rho^t = \partial \mathcal{L}_\rho(\rho^t) / \partial \rho^t$ are the individual per-frame gradients and $\nabla \rho_{\mathcal{A}}^{\boldsymbol{F}-1} = \nabla \rho^{\boldsymbol{F}-1}$. In practice, we found that using an exponential moving average (EMA) with decay β to accumulate the gradients, *i.e.*,

$$\nabla \rho_{\mathcal{A}}^t = (1 - \beta) \nabla \rho^t + \beta \frac{\partial \mathcal{A}(\rho^t, \mathbf{u}^t)}{\partial \rho^t} \nabla \rho_{\mathcal{A}}^{t+1}, \quad (5)$$

leads to better defined structures and smoother motion. This construction, which we will refer to as *global transport* formulation, requires an initial velocity sequence to create $\mathcal{A}^t(\rho^0)$, which we acquire from a forward-only pre-optimization pass, as described in Section 4. The velocity also receives gradients $\nabla \mathbf{u}_{\mathcal{A}}^t = \sum_{i=t}^{\boldsymbol{F}} \partial \mathcal{L}_\rho(\rho^i) / \partial \mathbf{u}^t = \partial \mathcal{A}(\rho^t, \mathbf{u}^t) / \partial \mathbf{u}^t \nabla \rho_{\mathcal{A}}^{t+1}$ from the back-propagation through the density advection. Those are, however, not back-propagated further through velocity self-advection.

Transport Losses   A per-frame transport loss that takes the general form of

$$\mathcal{L}_{\mathcal{A}(s)} = |\mathcal{A}(s^{t-1}, \mathbf{u}^{t-1}) - s^t|^2 + |\mathcal{A}(s^t, \mathbf{u}^t) - s^{t+1}|^2 \quad (6)$$

connects s^t to the previous and subsequent frames at times $t - 1$ and $t + 1$. The self-advection loss $\mathcal{L}_{\mathcal{A}(\mathbf{u})}$ induces temporal coherence in the velocity fields while the density transport loss $\mathcal{L}_{\mathcal{A}(\rho)}$ is used to build the initial velocity sequence. Despite the inclusion of and back-propagation through the global transport via Equation (5), both still contribute important gradients that help to reconstruct the inflow region and the motion within the visual hull, which we

show with an ablation in Appendix B.2. The divergence-free constraint of Equation (1) is enforced by additionally minimizing

$$\mathcal{L}_{\text{div}} = |\nabla \cdot \mathbf{u}|^2 . \tag{7}$$

3.2. Differentiable Rendering

To connect the physical priors to the target observations we leverage recent progress in GD-based differentiable rendering [37, 25]. We employ a non-linear image formation (IF) model that includes attenuation and shadowing to handle varying lighting conditions and denser material while still being able to solve the underlying tomographic reconstruction problem. Our rendering operator $\mathcal{R}()$ creates the image seen from view c of the density ρ given the outgoing light $\boldsymbol{L}$ at each point in the volume by solving the integral

$$\mathcal{R}(\rho, \boldsymbol{L}, c) = \int_{\text{n}}^{\text{f}} \boldsymbol{L}(x) e^{-\int_{\text{n}}^{x} \rho(a) da} dx \tag{8}$$

for each pixel-ray $\text{n} \rightarrow \text{f}$ of the image, using the Beer-Lambert law for absorption [13].

The outgoing light $\boldsymbol{L}$ is computed from point lights $p \in P$ at positions x_p with inverse-square falloff and single-scattering as well as ambient lighting to approximate the effect of multi-scattering. The intensities are i_p and i_a, respectively. The total outgoing light at point x is given as

$$\boldsymbol{L}_\rho(x) = i_a \rho(x) + \sum_{p \in P} i_p \rho(x) \frac{1}{1 + ||x_p - x||_2} e^{-\int_{x_p}^{x} \rho(a) da} . \tag{9}$$

In practice we employ ray-marching to approximate the shadowing term and render the image and its transparency. Implementation details and an evaluation of the rendering model can be found in Appendix A.4.

With this formulation we can constrain the renderings of the reconstruction to the target images $\boldsymbol{I}_c^t$ with

$$\mathcal{L}_{\text{tar}} = \frac{1}{c} \sum_{c \in \boldsymbol{C}} |\boldsymbol{I}_c^t - \mathcal{R}(\rho^t, \boldsymbol{L}_{\rho^t}, c)|^2 . \tag{10}$$

3.3. Learned Self-Supervision

Our physical priors and rendering constraints enable the reconstruction of a physically correct flow even from sparse views. However, ρ, and therefore the transport, are increasingly under-constrained with fewer views, and the huge space of indistinguishable solutions can lead to a mismatch between real and computed motions, severely reducing accuracy when using only a single view. To address these ambiguities, we train a fully convolutional neural network alongside the motion reconstruction that serves as a discriminator between images of real and reconstructed smoke. This discriminator learns to transfer features of real flows to the reconstruction via their visual projections over the

course of the optimization. While the discriminator can not recover the true volumetric structure pertaining to a given sample, it constrains the results to be plausible w.r.t. the given target samples. We chose the RaLSGAN (relativistic average least squares) loss variant [34] as our discriminator loss for increased training stability:

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}(\rho, l) := & \mathbb{E}_{r \sim R} \left[\left(\mathcal{D}(\boldsymbol{I}_r) - \mathbb{E}_{f \sim \Omega} \mathcal{D}(\hat{\boldsymbol{I}}_f) - l \right)^2 \right] \\ & + \mathbb{E}_{f \sim \Omega} \left[\left(\mathcal{D}(\hat{\boldsymbol{I}}_f) - \mathbb{E}_{r \sim R} \mathcal{D}(\boldsymbol{I}_r) + l \right)^2 \right] . \end{aligned} \tag{11}$$

$\boldsymbol{I}_r$ are randomly sampled from a set of reference images R, and $\hat{\boldsymbol{I}}_f := \mathcal{R}(\rho, \boldsymbol{L}_\rho, f)$ are rendered random views of the reconstruction. The label l is 1 when training the discriminator, and -1 when using it as a loss to optimize the density via $\partial \mathcal{L}_{\mathcal{D}}(\rho, -1)/\partial \rho$. This auxiliary loss term provides gradients for the optimization process and in this way constrains the large space of possible solutions for the fluid motion. Further details are given in Appendix A.2.

Extending the discriminator in the temporal domain via a triplet of inputs $(\boldsymbol{I}^{t-i}, \boldsymbol{I}^t, \boldsymbol{I}^{t+i})$ was proposed for fluid in other contexts [76]. We show in Section 5.2 that such an extension does not yield benefits as our transport priors already successfully constrain the physicality of the motion.

3.4. Additional Constraints

During reconstruction we restrict the density to the visual hull $\boldsymbol{H}$ constructed from the target views $\boldsymbol{C}$. After background subtraction the targets are turned into a binary mask, and projected into the volume ($\mathcal{R}^{-1}$) to create the hull via intersection:

$$\boldsymbol{H}^t := \bigcap_{c \in \boldsymbol{C}} \mathcal{R}^{-1}(H(\boldsymbol{I}_c^t - \epsilon), c) , \tag{12}$$

with H being the Heaviside step function. Using the visual hull as hard constraint avoids residual density and tomography artifacts in outer regions of the reconstruction volume. For single-view reconstruction the hull is constructed using auxiliary image hulls created by rotation of the original view around a symmetry axis of the physical setup (Appendix A.5). While this hull is itself not physically correct, it constraints the solution to plausible shapes.

As we focus on fluid convection scenarios such as rising smoke plumes, we model dedicated inflow regions for ρ that are degrees of freedom, optimized together with the flow. We allow for free motions via Neumann boundary conditions for $\mathbf{u}$.

To reduce the non-linearity of the transport constraints, we follow a commonly employed multi-scale approach [49, 21] that first optimizes larger structures without being affected by local gradients of small-scale features. This is especially important for the global formulation, as
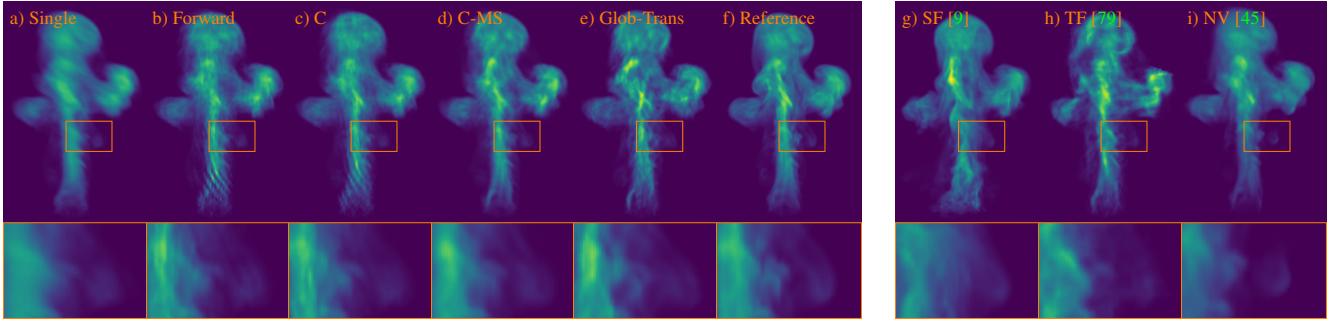
4

Figure 2: Multi-view evaluation with synthetic data: (a-e) Ablation with reference shown in (f). The different versions of the ablation (details in Section 5.1) continually improve density reconstruction and motion. (g-i) Comparison with previous work ScalarFlow [9], TomoFluid [79], and NeuralVolumes [45]. Our method in (e) yields an improved density reconstruction, in addition to a coherent and physical transport (see also Table 1a).

it enables the optimization of large-scale displacements when constructing the velocity field, and makes it easier to handle larger features in the density content, without details obstructing the velocity transport gradients, as $\partial \mathcal{A}(s, \mathbf{u}) / \partial \mathbf{u}$ depends on the spatial gradients of s (see also Appendix A.3).

4. Reconstruction Framework

To reconstruct a specific flow, the density ρ and velocity $\mathbf{u}$ are first initialized with random values inside the visual hull H. We alternately update ρ and $\mathbf{u}$ via Adam iterations [41]. The targets $\boldsymbol{I}_c^t$ for the rendering loss are given as image sequences with their corresponding camera calibrations $\boldsymbol{C}$. When using targets with a background, we composite the background and the reconstruction using the accumulated attenuation of the volume in order to match the appearance of the targets. While we experimented with optimizing the lighting (Appendix A.4.4), we use light positions and intensities that were determined empirically in our reconstructions. Below we give an overview of our framework, while algorithmic details and parameter settings are given in Appendix A.1.

Pre-Optimization Pass We run a single *forward* pass to obtain a density sequence $\tilde{\rho}$ as initial guess, and to compute an initial velocity sequence as starting point for the global transport optimization from Section 3.1. The density is computed via tomography using $\mathcal{L}_{\text{tar}}$ and the visual hull H without transport constraints. The velocities $\mathbf{u}^t$ are optimized using $\tilde{\rho}^t$ and $\tilde{\rho}^{t+1}$ via the density transport $\mathcal{L}_{\mathcal{A}(\rho)}$ and $\mathcal{L}_{\text{div}}$. To improve temporal coherence in this step the frames are initialized from the transported previous frame before optimization: $\tilde{\rho}^t := \mathcal{A}(\tilde{\rho}^{t-1}, \mathcal{A}(\mathbf{u}^{t-2})), \mathbf{u}^t := \mathcal{A}(\mathbf{u}^{t-1}, \mathbf{u}^{t-1})$, for which the initial velocity $\mathbf{u}^0$ is constructed using the multi-scale scheme of Section 3.1. The initial density sequence $\tilde{\rho}$ is discarded when the global transport constraints are introduced.

Coupled Reconstruction The main part of the reconstruction couples the observations over time by incorporating the remaining priors, namely $\mathcal{L}_{\mathcal{A}(\rho)}$ for the density and $\mathcal{L}_{\mathcal{A}(\mathbf{u})}$ for velocity. Combined, for the density loss in Equation (3) this yields $\mathcal{L}_\rho = \mathcal{L}_{\text{tar}} + \mathcal{L}_{\mathcal{A}(\rho)} + \mathcal{L}_{\mathcal{D}}(\rho, -1)$. Subsequent iterations of the optimizer run over the whole sequence, updating degrees of freedom for all time steps. When optimizing via the global transport formulation (Eq. (5)) the degrees of freedom for densities are reduced to ρ^0, and a density sequence is created by advecting ρ^0 forward before updating the velocities via backpropagation through the sequence while accumulating the gradients for ρ^0. The multi-scale approach is realized by synchronously increasing the resolution for all densities and velocities of the complete sequence with an exponential growth factor of $\eta = 1.2$ in fixed intervals. Target and rendering resolution for $\mathcal{L}_{\text{tar}}$ and $\mathcal{D}$ are adjusted in accordance with the multi-scale resolutions.

5. Results and Evaluation

We use a synthetic data set with known ground truth motion to evaluate the performance for multiple target views, before evaluating real data for the single-view case. For all cases below, please see the supplemental video for details. The differences of the velocity reconstructions are especially apparent when seen in motion. The video and source code can be found on the project website.

5.1. Multi-view Reconstructions

We simulate a fluid flow with a publicly available Navier-Stokes solver [47] with a resolution of $128 \times 196 \times 128$. Observations are generated for 120 frames over time with 5 viewpoints spread over a 120 degree arc using real calibrations.

Ablation We first illustrate our algorithm with an ablation study. A qualitative comparison of these versions is shown
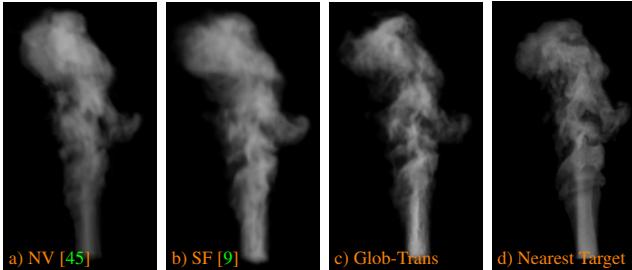
Figure 3: Other methods for multi-view reconstruction (a,b) compared to our approach (c), versus the nearest target from a $30°$ rotated viewpoint (d). Even for the new viewpoint, our reconstruction matches the structures of the target, while the other methods produce overly smooth reconstructions. The SF [9] result is representative for TF [79] here.

in Figure 2. The simplest *single* version uses the single pass described in Section 4. It yields a very good reconstruction of the input views, and low image space and volumetric errors for $\rho^{\boldsymbol{H}}$ and $\boldsymbol{I}_{\text{rand}}$ in Table 1, respectively, but strong tomographic artifacts. Due to the known deficiences of direct metrics like RMSE for complex signals, we additionally compute SSIM [74] and LPIPS [80] metrics for $\boldsymbol{I}_{\text{rand}}$. Although the single-pass version has good SSIM and LPIPS scores, it results in incoherent and unphysical motions over time. This is visible in terms of the metric $\mathcal{A}(\rho)^{\boldsymbol{H}}$ in Table 1, which measures how well the motion from one frame explains the next state, *i.e.*, $\mathcal{A}(\rho)^{\boldsymbol{H}} = |\mathcal{A}(\rho^{t-1}) - \rho^t|_2$.

The *forward* reconstruction (Section 4) introduces a temporal coupling, and yields slight improvements in terms of temporal metrics and reduces the artifacts. Next, enabling *coupling* (version C) over time by optimizing the sequence with transport losses greatly improves the transport accuracy to approx. 25% of the error of the forward version. Version C-MS in Table 1 activates *multi-scale* reconstruction during the coupled optimization and yields an improved reconstruction of larger structures, at the expense of an increased transport error. With version Glob-Trans we arrive at our *global transport* optimization, as described in Section 3.1. It results in sharper reconstructions, matching the perceived sharpness of the reference. This version achieves very good reconstructions in terms of density errors, and ensures a correct transport in terms of $\mathcal{A}(\rho)^{\boldsymbol{H}}$ up to numerical precision, see Table 1.

State of the Art We compare our global transport approach to three state of the art methods. Specifically, we compare with *ScalarFlow* (SF) [9] and *TomoFluid* (TF) [79] as representatives of physics-based reconstruction methods, and *NeuralVolumes* (NV) [45] as a purely visual, learning-based method. We use the synthetic data set from before, while targets are rendered in accordance with each method for fairness. Quantitative results are summarized in Table 1,
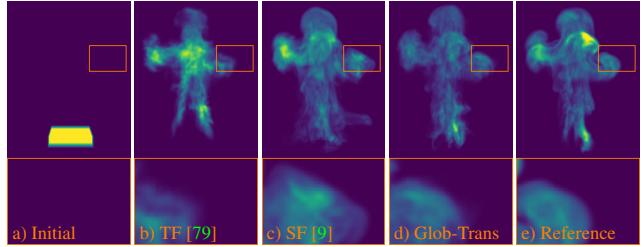

Figure 4: Evaluation of transport accuracy for an initial state (a) advected with the velocity reconstructions of TF (b) and SF (c). Our result (d) is closest to the reference in (e).

while qualitative comparisons are depicted in Figure 2 (g-i).

The SF reconstruction [9] employs physical priors in a single forward pass, similar to a constrained physics-based simulation. This leads to a temporally coherent velocity, but underperforms in terms of matching the target views. Note that SF has a natural advantage for the velocity reconstruction metrics in Table 1 for $\mathbf{u}^{\boldsymbol{H}}$ as it is based on the same solver used to produce the synthetic data (see Appendix B). The TF method [79] improves upon this behavior by employing an optimization with physical priors, similar to the *coupled* variant of our ablation. However, due to its reliance on inter-frame motions and spatio-temporal smoothness, the computed motion is temporally incoherent. It additionally relies on a view-interpolation scheme to constrain intermediate view directions, which can be severely degraded far from the target views, *e.g.*, as shown in Figure 2 (h), and hinder convergence. The purely visual neural scene representation of NV [45] does not employ physical priors. Despite encoding a representation over time by relying on inputs from the observed sequence for the inference of new views, it does not contain information about the motion between consecutive frames. To compute volumetric samples that are comparable, we extract a density by discarding black occlusions. While the measurements for ρ are in a similar range as our approach, the image space errors for $\boldsymbol{I}_{\text{rand}}$ are larger, despite not providing velocities.

Compared to all three previous methods, our global-transport approach outperforms the others in terms of image space reconstruction and most volumetric metrics in Table 1. This is also highlighted in Figure 3, where we compare a target frame qualitatively to a nearby target view. Most importantly, our method reconstructs a temporally consistent motion over the full duration of the observed sequence – a key property for retrieving a realistic fluid motion. We highlight this behavior with an additional test in Figure 4, where an initial state is advected over the course of 120 steps with the optimized velocity sequence. Our approach yields the closest match with the reference configuration.

6

Table 1(a):

Metrics	$\mathcal{A}(\rho)^H$	ρ^H		$\mathbf{u}^H$		I_{rand}	
Method	RMSE↓	RMSE↓	SSIM↑	RMSE↓	SSIM↑	PSNR↑	LPIPS↓
Single	0.1352	1.300	0.599	0.453	0.148	41.51	.0203
Forward	0.1222	2.429	0.464	0.463	0.165	39.81	.0281
C	0.0291	2.268	0.502	0.452	0.180	40.26	.0244
C-MS	0.0505	1.309	0.585	0.445	0.167	41.59	.0186
Glob-Trans	**1.4e-7**	1.844	**0.533**	0.452	0.171	**39.86**	**.0257**
SF [9]	1.573*	2.814	0.328	**0.440**	**0.220**	35.80	.0529
TF [79]	0.3879	**1.707**	0.473	0.544	0.082	30.19	.0492
NV [45]	N/A	2.002	0.337	N/A	N/A	27.77	.0674

(a) Multi-view: mean volume and image statistics and errors for synthetic data.

Table 1(b):

Metrics	Target views		I_{rand}
Method	PSNR↑	LPIPS↓	FID↓
Forward	29.42	.0538	164.6
C-MS	28.31	.0541	166.7
Glob-Trans	27.29	.0484	151.2
Full	27.40	.0489	**140.3**
Full+$\mathcal{D}_t$	**27.49**	**.0478**	147.3
pGAN [25]	22.91	.0891	185.4

(b) Single-view: mean image statistics evaluated for five new random views.

Table 1: Error metrics for multi-view (left) and single-view cases (right). $\mathcal{A}(\rho)^H$ denotes the motion difference $\mathcal{A}(\rho^{t-1}) - \rho^t$, ρ^H and $\mathbf{u}^H$ denote volumetric errors; all of these are measured inside the visual hull. I_{rand} measures differences to rendered targets using 32 random views. Bold numbers denote best scores for full algorithms, *i.e.*, without ablation variants. The * for $\mathcal{A}(\rho)^H$ of SF indicates resampling the velocities to the same grid resolution as other methods for comparison.
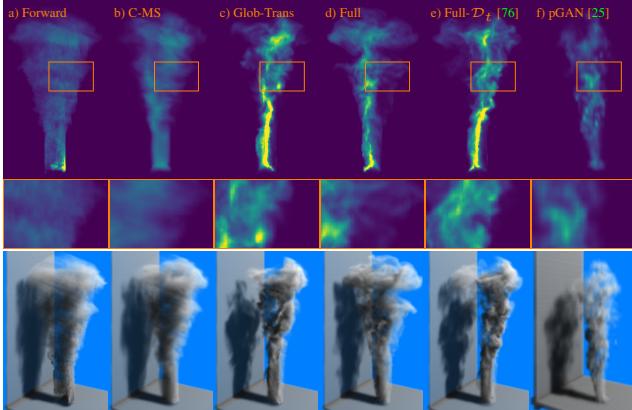

Figure 5: Single-view reconstructions of a captured fluid (target at $0°$). Top & middle: density from a $90°$ side view. Bottom: rendered visualization at $60°$. Our version with discriminator (d) yields coherent and sharp flow structures, preventing any smearing along the unconstrained direction (a, b) or streak artefacts (c). Temporal supervision [76] has no added benefit (e), while the purely learning-based method [25] fails to generate coherent features (f).
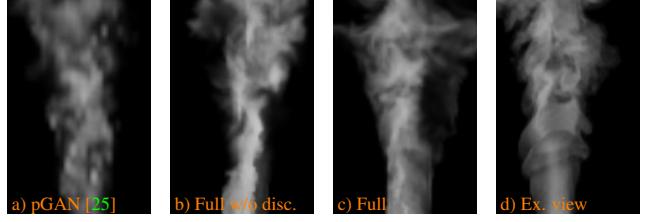

Figure 6: Unseen view of a single-view reconstruction: (a) previous work, our algorithm without (b) and with discriminator (c). The discriminator in (c) yields an improved reconstruction of small to medium scale structures in the flow that match the style of an example flow (d).

5.2. Single-View Reconstruction

We demonstrate the learned self-supervision, *i.e.*, the efficacy of the discriminator loss, on a reconstruction of 120 frames from a real capture [9]. To evaluate the results from unconstrained directions we also include FID measurements [27] in Table 1 (b).

Ablation Results of the single-view ablation for an unseen view at a $90°$ angle are shown in Figure 5. Here, the *forward* variant (a) highlights the under-constrained nature of the single-view reconstruction problem with strong smearing along depth. The *coupling* version (b) refines the reconstructed volumes, but still contains noticeable striping artifacts as well as a lack of details. Our *global transport*

formulation (c) resolves these artifacts and yields a plausible motion, which, however, does not adhere to the motions observed in the input views. The *full* version (d) introduces the methodology explained in Section 3.3, and arrives at our full algorithm including global transport and the discriminator network. It not only sharpens the appearance of the plume, but also makes the flow more coherent by transferring information about the real-world behavior to unseen views. This is depicted in Figure 6 (c), where flow details closely match the style of an example view (d).

State of the Art Only few works exist that target volumetric single-view reconstructions over time. As two representatives, we evaluate the *temporal discriminator* D_t proposed in previous work for fluid super-resolution [76], and the *Platonic GAN* approach for volumetric single view reconstructions [25].

The temporal self-supervision (Full-D_t) [76] was proposed for settings without physical priors. We have extended our full algorithm with this method, and note that it does not yield substantial improvements for the optimized motion, see Figure 5 (e). This highlights the capabilities of the global transport, which already strongly constrains the

7

solution. Thus, a further constraint in the form of a temporal self-supervision has little positive influence, and slightly deteriorates the FID results in Table 1 (b).

The *Platonic GAN* (pGAN) was proposed for learning 2D supervision per class of reconstructed objects. We have used this approach to pre-train a network for per-frame supervision of the rising smoke case of our single-view evaluation yielding improvements in terms of depth artifacts. However, due to the lack of constraints over time, the solutions are incoherent, and do not align with the input sequence, leading to an FID of more than 180.

In contrast, the solution of our full algorithm yields a realistically moving sequence that very closely matches the input view while yielding plausible motions even for highly unconstrained viewing directions. This is reflected in the FID measurements in Table 1 (b), where our full algorithm outperforms the other versions with a score of 140.3. We found the discriminator to be especially important for noisy, real-world data, where hard, model-based constraints can otherwise lead to undesired minima in the optimization process. Both comparisons highlight their importance of learned self-supervision for single-view reconstructions, and their capabilities in combination with strong priors for the physical motion.

5.3. Real Data Single-View Reconstructions

We show two additional real flow scenarios reconstructed from single viewpoints with our full algorithm in Figure 7. For each case, our method very closely recovers the single input view, as shown in the top row of each case. In addition, the renderings from a new viewpoint in the bottom row highlight the realistic motion as seen from unconstrained viewpoints. Here, our optimization successfully computes over 690 million degrees of freedom, *i.e.*, grid cells, in a coupled fashion for the wispy smoke case. Resolution details are given in Appendix B, while the supplemental video shows these reconstructions in motion. Interestingly, we found it beneficial to re-train the discriminator alongside each new reconstruction. Reusing a pretrained network yields inferior results, which indicates that the discriminator changes over the course of the reconstruction to provide feedback for different aspects of the solution.

6. Discussion and Conclusions

We have presented a first method to recover an end-to-end motion of a fluid sequence from a single input view. This is achieved with a physical prior in the form of the global transport formulation, a differentiable volumetric renderer and a learned self-supervision for unseen angles.

Our single-view method comes with the limitation that the learned self-supervision assumes that the flow looks similar to the input view from unseen angles. The discriminator can only make the results look plausible by trans-
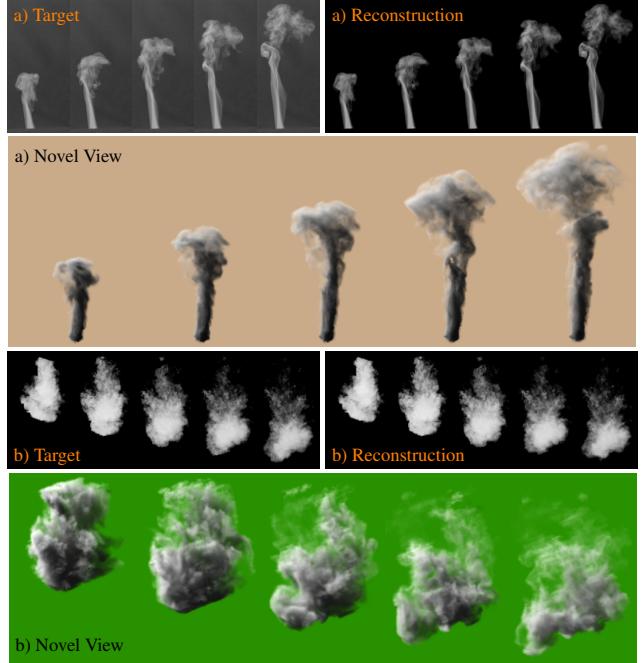

Figure 7: Additional single-view reconstructions. Top left: input views, top right: reconstruction from input view, bottom: different viewpoint with modified lighting. Our algorithm successfully reconstructs realistic motions for wispy smoke (a), as well as thicker volumes (b).

ferring the visual style of the given references. Physical correctness is induced by the direct, physical transport constraints. However, fluid flow naturally exhibits self similarity, and the discriminator performs significantly better than re-using the same input for constraints from multiple angles. Dissipation effects are not modeled in our approach, and would thus lead to errors in $\mathcal{L}_{\text{div}}$ or $\mathcal{L}_{\mathcal{A}(\rho)}$. We further do not support objects/obstacles immersed in the flow. Additionally, we rely on positional information about the inflow into the reconstruction domain. Jointly optimizing for inflow position and flow poses an interesting challenge for future work. For our full method we have measured an average reconstruction time of ~ 30 minutes per frame using a single GPU, more details are in Appendix A.9.

In conclusion, our method makes an important step forward in terms of physics- and learning-based reconstruction of natural phenomena. We are especially excited about future connections to generative models and other physical phenomena, such as turbulent flows [73], liquids [64], or even elastic objects[75].

7. Acknowledgements

8

Supplemental Document

This is the supplemental material for the CVPR 2021 paper *Global Transport for Fluid Reconstruction with Learned Self-Supervision*. The accompanying video as well as the source code can be found on the project website.

A. Details on Method and Implementation

A.1. Algorithm

Here we detail the different algorithms used in the ablation. Algorithm 1 shows how the density of a single frame is reconstructed. This variant corresponds to the *single* version using the single pass of Section 4. The *forward* reconstruction is summarized in Algorithm 2; it uses the density reconstruction of Algorithm 1. In Algorithm 3 we depict the *coupling* (C) over time, and Algorithm 4 shows our full method with global transport optimization *global transport* (Glob-Trans).

Algorithm 1: Update Steps

1 **Function** UpdateDensity($\rho, \mathcal{L}_\rho$)**:**
2    $\boldsymbol{L} \leftarrow$ BuildLighting(ρ, *lights*)**;**
3    $\hat{\boldsymbol{I}}_c \leftarrow \mathcal{R}(\rho, \boldsymbol{L}, c)$;
4    $\nabla\rho \leftarrow \partial\mathcal{L}_\rho / \partial\rho$;
5   Update ρ with $\nabla\rho$ using Adam;
6    $\rho \leftarrow \max(\rho, 0)$ **;** //clip negative density
7 **Function** UpdateVelocity($\rho, \mathcal{L}_\mathbf{u}$)**:**
8    $\nabla\mathbf{u} \leftarrow \partial\mathcal{L}_\mathbf{u}/\partial\mathbf{u}$;
9   Update $\mathbf{u}$ with $\nabla\mathbf{u}$ using Adam;
10 **Function** UpdateDiscriminator(ρ)**:**
11    $\boldsymbol{I}_r \sim R$;
12    $f \sim \Omega$; //sample random views
13    $\hat{\boldsymbol{I}}_f \leftarrow \mathcal{R}(\rho, \boldsymbol{L}, f)$;
14    $\hat{\boldsymbol{I}}_h \sim$ history;
15   history $\leftarrow$ history $\circ \hat{\boldsymbol{I}}_f$;
16    $\hat{\boldsymbol{I}}_f \leftarrow \hat{\boldsymbol{I}}_f \circ \hat{\boldsymbol{I}}_h$;
17   Update $\Theta_\mathcal{D}$ with $\partial\mathcal{L}_\mathcal{D}(\rho, 1)/\partial\Theta_\mathcal{D}$;
    //Equation (11)

A.2. Discriminator

Our discriminator $\mathcal{D}$ has a fully convolutional architecture with 14 convolution layers (filters, strides): (8,1) (16,2) (16,1) (24,2) (24,1) (32,2) (32,1) (32,1) (64,2) (64,1) (64,1) (16,1) (4,1) (1,1). All layers use a filter size of 4×4 and are followed by LReLU activations, with leak of 0.2. We use SAME padding, but instead of padding with zeros we use the mirrored input to avoid issues described in [31]. For data augmentation we apply random crop, scale, rotation, intensity and gamma scale. We keep a history of rendered

---

Algorithm 2: Forward Pass

1 **Function** OptimizeDensity(ρ)**:**
2   **for** n **do**
3   UpdateDensity($\rho, \mathcal{L}_\text{tar}$)**;**
4 **Function**
   OptimizeVelocity($\mathbf{u}, \rho_{from}, \rho_{to}, n_{MS}$)**:**
5   **for** $i = 0$ **to** n **do**
6   **if** $i \in n_{MS}$ **then**
7        $\mathbf{u} \leftarrow$ ResizeGrid($\mathbf{u}$)**;**
8   UpdateVelocity($\mathbf{u}, \mathcal{L}_{\mathcal{A}(\rho)} + \mathcal{L}_\text{div}$)**;**
9 **Function** ReconstructForward($\{\rho\}, \{\mathbf{u}\}$)**:**
10    $\rho^0 \leftarrow \boldsymbol{H}^0 \cdot c$;
11    $\mathbf{u}^0 \leftarrow \boldsymbol{H}^1 \cdot$ rand;
12   OptimizeDensity(ρ^0)**;** //Algorithm 1
13    $\rho^1 \leftarrow \rho^0$;
14   OptimizeDensity(ρ^1)**;**
15   OptimizeVelocity($\mathbf{u}^0, \rho^0, \rho^1, n_{MS}$)**;**
16   **for** $t = 1$ **to** $F - 1$ **do**
17      $\mathbf{u}^t \leftarrow \mathcal{A}(\mathbf{u}^{t-1}, \mathbf{u}^{t-1})$;
18      $\rho^{t+1} \leftarrow \mathcal{A}(\rho^t, \mathbf{u}^t)$;
19   OptimizeDensity(ρ^{t+1})**;**
20   OptimizeVelocity($\mathbf{u}^t, \rho^t, \rho^{t+1}, \emptyset$)**;**
21    $\mathbf{u}^{F-1} \leftarrow \mathcal{A}(\mathbf{u}^{F-2}, \mathbf{u}^{F-2})$;

Algorithm 3: Coupled Optimization

1 ReconstructForward($\{\rho\}, \{\mathbf{u}\}$)**;**
2 **for** $i = 0$ **to** n **do**
3   **for** $t = 0$ **to** F **do**
4   UpdateDensity($\rho^t, \mathcal{L}_\text{tar} + \mathcal{L}_{\mathcal{A}(\rho)} + \mathcal{L}_\mathcal{D}$)**;**
5   UpdateVelocity($\mathbf{u}, \mathcal{L}_{\mathcal{A}(\rho)} + \mathcal{L}_{\mathcal{A}(\mathbf{u})} +$
       $\mathcal{L}_\text{div}$)**;**
6   **if** *useDiscriminator* **then**
7   UpdateDiscriminator($\{\rho\}$)**;**

---

fake samples $\hat{\boldsymbol{I}}_f$ to reuse as additional fake samples later for training the discriminator which improves the quality of the reconstruction and reduces rendering operations. The batch size for each discriminator or density update is 8 real samples and 4 rendered fake samples, with 4 additional fake samples from the history when training the discriminator. The 'average' formulation of the relativistic GAN [34] allows to use different resolutions and batch sizes for real and fake samples, while the least-squares variant gives empirically better results in our case. We also use an L_2 regularization for the weights of the discriminator $\Theta_\mathcal{D}$.

Algorithm 4: Full Method

```
1  ReconstructForward({ρ̃}, {u});
2  ρ⁰ ← ρ̃⁰;
3  for i = 0 to n do
4      if i ∈ n_MS then
5          ρ⁰ ← ResizeGrid(ρ⁰);
6          for t = 0 to F do
7              uᵗ ← ResizeGrid(uᵗ);
8      // build sequence from first frame ρ⁰
9      for t = 1 to F do
10         ρᵗ ← A(ρᵗ⁻¹, uᵗ⁻¹);
11     ∇ρ_A^F ← 0; // init gradient EMA
12     for t = F − 1 to −1 do
13         ∇ρᵗ ← ∂L_ρ(ρᵗ)/∂ρᵗ;
14         ∇ρ_A^t ← ∂A(ρᵗ, uᵗ)/∂ρᵗ · ∇ρ_A^{t+1};
15         if t == 0 then
16             Update ρᵗ with ∇ρᵗ + λ_{ρ_A}∇ρ_A^t;
17         ∇u_A^t ← ∂A(ρᵗ, uᵗ)/∂uᵗ · ∇ρ_A^{t+1};
18         Update uᵗ with ∇uᵗ + λ_{u_A}∇u_A^t;
19         ∇ρ_A^t ← β * ∇ρ_A^t + (1 − β) * ∇ρᵗ;
20     if useDiscriminator then
21         UpdateDiscriminator({ρ});
```

A.3. Differentiable Warping

For the discretized operator $\mathcal{A}$ we use a second-order transport scheme in order to preserve small-scale content on the warped fields [65]. This MacCormack advection is based on Semi-Langrangian advection $\mathcal{A}_{SL}$, where the advected quantity s^{t+1} at position $\vec{x}$ is the tri-linear interpolation of s^t at $\vec{x} - \vec{u} * \Delta t$. With this, the MacCormack-advection is defined as

$$
\begin{aligned}
\hat{s}^{t+1} &:= \mathcal{A}_{SL}(s^t, \mathbf{u}^t), \\
\hat{s}^t &:= \mathcal{A}_{SL}(\hat{s}^{t+1}, -\mathbf{u}^t), \\
s^{t+1} &:= \hat{s}^{t+1} + 0.5(s^t - \hat{s}^t).
\end{aligned}
\tag{13}
$$

However, we use a variant that reverts to $\mathcal{A}_{SL}$ when the corrected term s^{t+1} would exceed the values used for interpolation in the first $\mathcal{A}_{SL}$:

$$
s^{t+1} := \begin{cases} s^{t+1} & \text{if } \hat{s}^{t+1}_{min} \leq s^{t+1} \leq \hat{s}^{t+1}_{max} \\ \hat{s}^{t+1} & \text{else} \end{cases}.
\tag{14}
$$

This reversion is necessary as the un-clamped MacCormack correction causes issues like negative density and escalating inflow from the open boundaries. The result is smoother than the one obtained without clamping, but still more detailed than $\mathcal{A}_{SL}$.

The advected generic quantity s can be both ρ and $\mathbf{u}$, in case of $\mathbf{u}$ the components are handled individually and treated as scalars.

Differentiation We implemented a differentiable SL advection as tensorflow CUDA operation. Since the MacCormack advection is composed of two SL steps and a condition, its gradients are simply handled by automatic differentiation. Since we optimize ρ and $\mathbf{u}$, both $\partial\mathcal{A}(s, \mathbf{u})/\partial s$ and $\partial\mathcal{A}(s, \mathbf{u})/\partial\mathbf{u}$ are required. For $\partial\mathcal{A}(s, \mathbf{u})/\partial s$, the gradients are just scattered to the grid positions used for the interpolation in the forward step, after being multiplied with the interpolation weights. $\partial\mathcal{A}(s, \mathbf{u})/\partial\mathbf{u}$ is $\mathcal{A}(\nabla s, \mathbf{u})$, *i.e.*, during back-propagation the gradients are multiplied component wise with the advected spatial gradients of the scalar grid s.

A.3.1 Boundary Handling

We use open boundaries for both density and velocity. Technically that means that velocities at the boundaries are not constrained (or handled differently from the inner velocities in any way). In the advection, look-ups that fall outside of the volume are clamped to the boundaries, thus using the values there. Their gradients are also accumulated at the boundaries accordingly.

A.3.2 Inflow

Inflow is only handled explicitly for the density, the velocity has to solve all inflow via the open boundaries. As we target rising smoke plumes we place an inflow region at the lower end of the visual hull H, with some overlap, although it could be composed of arbitrary regions of the volume without changing the algorithm (we use a simple mask to indicate inflow cells). These cells are additional degrees of freedom and are optimized alongside the density. They are individual for every frame, and still used when using our global transport formulation. The advection then becomes $\mathcal{A}(\rho^t + inflow, \mathbf{u}^t) = \rho^{t+1}$ and is used as such in all losses and the global transport. This is also how the inflow receives gradients. The only constraint on the inflow is that $\rho^t + inflow \geq 0$, meaning that the inflow itself can be (partially) negative, and thus serve as outflow. Density can also enter and leave the volume via the boundaries, just like the velocity.

A.4. Differentiable Rendering

For rendering we use a discrete ray-marching scheme that essentially replaces the integrals in Equation (8) with discrete sums, stepping though the volume along the view-

depth (z) with a fixed step size:

$$\sum_{z=0}^{Z} \boldsymbol{L}(x_z) e^{-\sum_{a=0}^{x_z} \rho(a)}, e^{-\sum_{z=0}^{Z} \rho(x_z)}. \quad (15)$$

To implement the single-scattered self-shadowing we use a shadow volume, instead of casting shadow rays from every sample point. Thus, the shadow computation is like rendering the transparency from the point of view of the light, but storing every z-slice to create the shadow volume.

We implement our custom ray-marching kernel and its analytic gradient as tensorflow CUDA operation. The backwards pass inside this operation is what standard auto-differentiation would yield: go backwards along the ray and scatter gradients to the locations used for interpolation in the forward pass.

A.4.1 Sampling Gradients

The naive back-propagation of gradients through the grid-sampling, *i.e.*, scattering the gradients of a sample point to all used interpolants, weighted with the original interpolation weights, yields regular artifacts in the reconstruction, likely due to the regular grid sampling. Using an intermediate grid allows us to invert the sampling for back-propagation with another sampling operation as in the forward pass, which can then employ mip-mapping, and thus reduces aliasing in both directions. Conceptually, this approach does not differentiate the sampling, but rather transforms the volume of gradients into another space, resulting in spatially smooth gradients. It is, however, very memory intensive as all samples have to be stored. An equivalent gradient can also be obtained with scattering, by normalizing the gradients with their accumulated interpolation weights. This also avoids aliasing, as long as the rendering resolution is large enough, and is the method we use for our final results.

A.4.2 Comparison to Linear Image Formation

The comparisons in Figures 8 and 9 highlight the importance of the non-linear IF for visible light capturing. While a linear model reduces densities to account for effects like shadowing, the differentiable rendering in our optimization yields correct gradients to recover the original density, (b) *vs*. (c) in Figure 8.

A.4.3 Comparison to Path-tracer

We compare our renderer (Section 3.2) to a path-traced reference in Figure 10. Using 0 light bounces in the path-tracer (*i.e.* only single-scattering shadow rays) produces results very similar to our renderer, while the results with more bounces (multi-scattering) change gracefully.
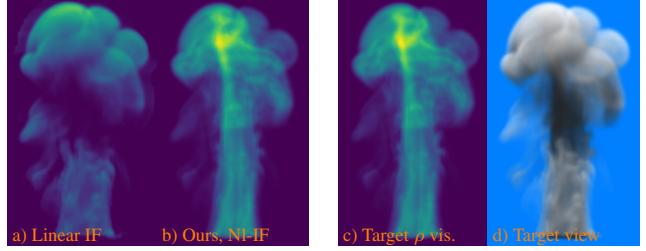

Figure 8: Non-linear IF optimization: a linear model "bakes" lighting into the density reconstruction (a), while our pipeline can recover the correct density distribution (b) via a differentiable renderer. (c,d) show target density as visualization and with lighting, respectively.
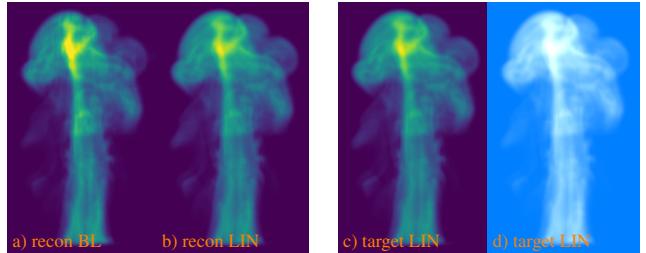

Figure 9: A target rendered with a linear (LIN) IF can be recovered by the attenuated model (BL), as long as sufficient lighting is provided.

Figure 10: Our renderer vs. Blender 2.91 [5] Cycles path-tracer (128 SPP, denoised), 0 and 4 light bounces.

A.4.4 Backgrounds and Lighting Optimization

When using a black background or monochrome target image there is an ambiguity between high light intensity in conjunction with low density and lower light intensity with more density as these two cases cannot be distinguished in a target image. With a black background the transparency does not matter, and with a monochrome target (with any background) contributions from reflected light and background are indistinguishable. Using a background with a color distinguishable from the smoke, the transparency, and therefore density, along a ray can be determined success-

Figure 11: Side view (90°) of a density reconstructed with black (a) and light blue RGB(0,127,255) (b) background. A colored background helps to recover the correct density in shadowed regions.

fully.

The reconstruction consequently yields better results when using a colored background, as seen in Figure 11. However, as our real target dataset [9] uses only grey-scales, we likewise used black backgrounds for our synthetic tests.

We also experimented with optimizing the light intensities (i_a and i_p), which worked better with colored backgrounds. Because this adds additional degrees of freedom, making the reconstruction harder, we ultimately determined our light intensities empirically.

A.5. Visual Hull

For our multi-view reconstruction experiments with 5 target views, the visual hull $\boldsymbol{H}$ is constructed only from the 5 available views, as described in Section 3.4. After the background subtraction the target images are turned into a binary mask, using a threshold of $\epsilon = 0.04$ to cut of residual noise. These masks are slightly blurred with $\sigma = 1$ (in pixels) before being projected into the volume (Eq. (12)). The hulls are then blurred again with $\sigma = 0.5$ (in cells).

To construct a visual hull from a single view, we create 4 additional, evenly spread views from the available view by rotating it around the central y-axis (up) of the volume. The masks created from these auxiliary views are additionally mirrored at the projected rotation axis to reduce cutoffs in the original view when constructing the volume hull by intersection. This works well for relatively symmetric rising plumes, but is not guaranteed to work for arbitrary shapes. Using a visual hull to guide the densities noticeably improves the quality of the reconstruction, but constructing a hull for sparse or single-view reconstructions that adheres to physically correct motion remains a topic for future work.

A.6. Multi-Scale Approach

We employ a multi-scale approach during reconstruction, increasing the grid resolution while the spatial size of the domain stays the same. The up-sampling is done using tri-linear interpolation. In our full method we use a scaling factor per step of 1.2. Thus, for a reference resolution of

128 (see also Appendix B), the velocity of the first frame is scaled 4 times ($14 \rightarrow 18 \rightarrow 21 \rightarrow 25 \rightarrow 30$) in the pre-optimization, and density and velocity synchronously in the coupled phase an additional 8 times ($30 \rightarrow 36 \rightarrow 43 \rightarrow 51 \rightarrow 62 \rightarrow 74 \rightarrow 88 \rightarrow 106 \rightarrow 128$). As the first frame velocity optimization operates on a lower resolution than the densities ρ^t and ρ^{t+1} used therein, the densities are temporarily down-sampled (with filtering) to match the resolution of the velocity.

A.7. Gradient Descent with Global Transport

Naive automatic differentiation of a full 120 frame sequence via back-propagation would require a lot of memory as all intermediate results of the forward pass have to be recorded for the backwards pass. To reduce the memory requirements we split the sequence to only do automatic back-propagation for one frame at a time. The back-propagation between frames is handled more explicitly (see also line 19 in Algorithm 4). The result is the same, at least when using Equation (4) instead of the EMA version (Eq. (5)). We keep all resources in (CPU) RAM, only the current frame and intermediate steps needed for automatic differentiation are moved to the GPU.

A.8. Hyperparameters

The hyperparameters, which can be set individually for the first frame in the pre-pass, the remaining frames of the pre-pass, and the main optimization, are reported in Table 2. Some benefit from a linear ($\xrightarrow{lin}$) or exponential ($\xrightarrow{exp}$) fade-in or growth. We do not use any smoothness regularization for density or velocity.

A.9. Hardware and Runtime Statistics

We used the following hard- and software configuration for our experiments: Hardware: CPU: Intel(R) Core(TM) i7-6850K, GPU: Nvidia GeForce GTX 1080 Ti 11GB, RAM: 128GB; Software (versions): Python 3.6.9, Tensorflow 1.12 (GPU), CUDA 9.2. We implemented custom tensorflow operation in CUDA for rendering (ray-marching and shading) and advection as well as their gradient operations.

For our full method, using the resolution detailed in Appendix B, we have measured an average reconstruction time of 43 - 65 hours (~ 30 minutes per frame, variation based on system load) for 120 frames of single view reconstruction from real targets with our current implementation and hardware. The pre-optimization (Sec. 4) takes only 2% of the total time. The majority of the time is spent on the density optimization (60%, 38% of which is needed for the discriminator loss and rendering the needed fake images), while Calculating and back-propagating though the losses for $\mathbf{u}$ makes up 30% of the time. The manual back-propagation

Loss \Pass	Fwd first	Fwd	Main
Density			
n	600	600	4200
n_{MS}	-	-	400×8
η_ρ	3	$3 \xrightarrow{lin} 1$	2.4
$\mathcal{L}_{\text{tar}}$	1.74e-5		
$\mathcal{L}_{\mathcal{A}(\rho)}$	0	0	2.7e-10 $\xrightarrow{lin}$ 5.4e-10
$\mathcal{L}_{\mathcal{D}}(\rho, -1)$	0	0	1.5e-5
Velocity			
n	6000	600	4200
n_{MS}	1000×4	-	400×8
$\eta_{\mathbf{u}}$	0.04	0.02	$\xrightarrow{exp}$ 0.016
$\mathcal{L}_{\mathcal{A}(\rho)}$	4.1e-10		
$\mathcal{L}_{\mathcal{A}(\mathbf{u})}$	0	0	4e-11 $\xrightarrow{lin}$ 8e-11
$\mathcal{L}_{\text{div}}$	8.6e-10 $\xrightarrow{exp}$	2.6e-9	$\xrightarrow{exp}$ 1.7e-8
Discriminator[†]			
$\eta_{\mathbf{u}}$	-	-	2e-4
$|\Theta_{\mathcal{D}}|^2$	-	-	2e-3

Table 2: Hyperparameters for the first (lines 12 and 15 in Algorithm 2) and remaining frames of the forward reconstruction and our coupled global transport optimization (Algorithm 4). learning rate η, iterations n with multi-scale intervals n_{MS} with a resize factor of 1.2.

though the global transport via Equation (5) takes 4% of the total runtime.

B. Additional Evaluation

The Reference has a grid size of $128 \times 196 \times 128$. As base grid size for our reconstruction we use $128 \times 227 \times 128$, which is cropped to the union of the AABBs of the visual hulls of all frames plus a padding of 4 cells on all sides. For our reconstructions the grid sizes are therefore reduced to $125 \times 187 \times 93$, $94 \times 186 \times 95$, $97 \times 164 \times 77$ and $86 \times 161 \times 76$ for synthetic multi-view, synthetic single-view, real multi-view and real single-view for the base plume, respectively. The spatial resolution remains the same. As lighting model we use a single white point light with single-scattering, hand-placed above the central camera, and white ambient light to approximate multi-scattering. The light intensities are $i_p = 0.85$ and $i_a = 0.64$, respectively.

B.1. Additional Multi-View Ablations

An extended evaluation of the motion of the synthetic case is shown in Figure 12. The *forward* reconstruction removes the streak-like artifacts, but does so at the expense of spatial reconstruction accuracy regarding the metrics. The step to *coupled* improves the motion it results in only minor improvements in ρ. The multi-scale approach (C-MS) brings the metrics for ρ on-par with the *single* version. The bottom row show a continuously improving transport of a test density, compared to the reference (Figure 12 f), seen in the resulting distribution of the density. While ScalarFlow

might appear closer to the reference it shows a mismatch in shape at the stem and backside (right side, shown in the detail) of the resulting plume.

An ablation of multi-view reconstructions of the SF data is shown in Figure 14. Here, C-MS shows an unphysical accumulations of density at the top which are resolved by our *global transport* (Glob-Trans). The discriminator (Full) has only little effect in the multi-view reconstruction, sometimes causing slight halo-artifacts, and is thus not used here. Previous methods give a more diffuse result.

B.2. Warp Loss in Global Transport

In Figure 13 we show an additional ablation that highlights the effects of keeping the per-frame warp loss $\mathcal{L}_{\mathcal{A}(\rho)}$ for both density ($\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \rho$) and velocity ($\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \mathbf{u}$), even when global transport (Sec. 3.1) is used. Our global transport already provides gradients from the advection of the density for both ρ (Eq. (5)) and $\mathbf{u}$, meaning that the explicit density warping loss seems redundant for both. In fact, since the sequence is defined via transport (Eq. (2)), the warp loss, and therefore its gradients, should be zero. However, since the visual hull is applied after every advection step (in the initial transport, not the warp loss), the error stems from the parts removed by the hull. Using the hull as a loss, instead or in addition to a hard constraint, does not change anything about the results in Figure 13.

As visible in the figure, $\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \rho$ is still mostly redundant, only in the real data case it reduces the divergence of the reconstructed density grid (not visible in the rendering). However, it also has no negative influence on the reconstruction. Removing $\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \mathbf{u}$ from the optimization has adverse effect in the real case, resulting in visible divergence in the lower part of the plume, turning the inflow into an outflow. In the Synthetic case the reconstruction still works well, showing only less inflow. We conjecture that the differences between the synthetic and real cases are caused by a mismatch of the physical model to the real case. Using $\mathcal{L}_{\mathcal{A}(\rho)}$ puts more emphasis (gradients) on *local* correctness of the transport itself. $\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \rho$ adapts the density to fit the existing transport. However, because the density also has to fit the observations, which have more weight, there is no change, and thus removing these gradients has no big impact. $\partial \mathcal{L}_{\mathcal{A}(\rho)} / \partial \mathbf{u}$, on the other hand, adapts the velocity, *i.e.*, the transport, to fit the existing density, including the hull constraint. These results seem to suggest that explicit adaption of the local transport to the densities is still necessary when using the global transport.

B.3. State of the Art

We re-implemented TomoFluid (TF) in our own Framework as no original source-code is publicly available as of the time of writing. We use the loss functions and view-interpolations as described in the original work, but the hy-
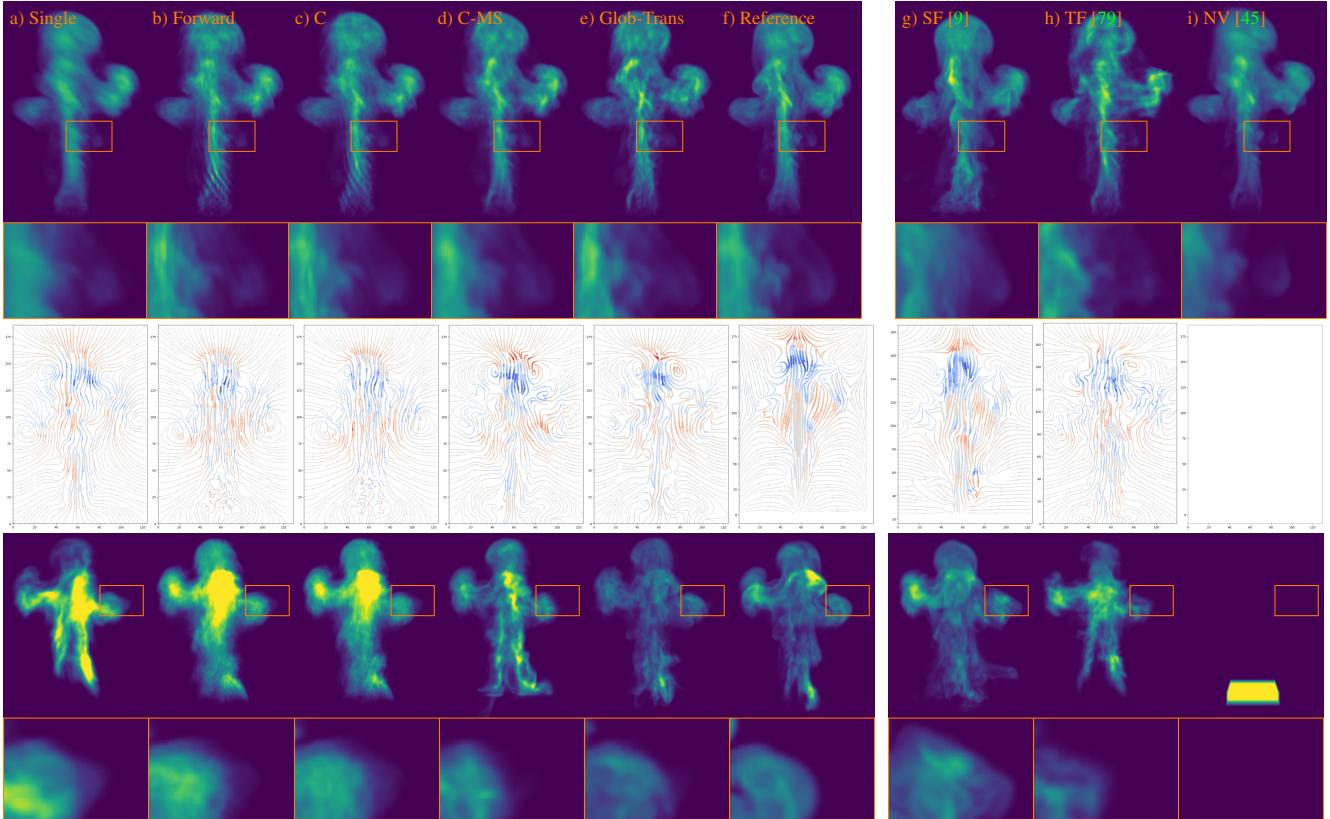
Figure 12: Multi-view evaluation with synthetic data: (a-e) Ablation with reference shown in (f). The different versions of the ablation continually improve density reconstruction (top) and motion accuracy, illustrated by advecting the initial state shown in (i) with the reconstructed velocity sequence, in the bottom row. (g-i) Comparison with previous work: ScalarFlow [9], TomoFluid [79], and NeuralVolumes [45]. Our method in (e) yields an improved density reconstruction, in addition to a coherent, and physical transport. Note that (i) does not produce velocities.
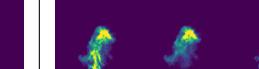

(a) Synthetic

(b) Real
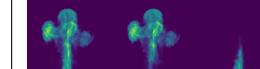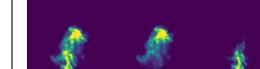
Figure 13: Evaluation of the influence of the density warp loss $\mathcal{L}_{\mathcal{A}(\rho)}$ when *global transport* is also used. The tables show an ablation of the $\mathcal{L}_{\mathcal{A}(\rho)}$ applied to density and velocity, for both synthetic and real data. Every image shows the following triplet: left, the final result of the optimization, obtained by advecting ρ^0 with the velocity sequence and adding the inflow in every step; the center shows the same, but without any inflow; on the right, only the inflow is advected, *i.e.*, ρ^0 is set to all zero. The top left images are the desired results. While the gradients provided by this loss are are almost redundant in the synthetic case (a), the gradients w.r.t. $\mathbf{u}$ are necessary when using real data(b), even though the same dependencies are modeled by the global transport. Gradients w.r.t. ρ are sill redundant, but do not hurt the reconstruction either.
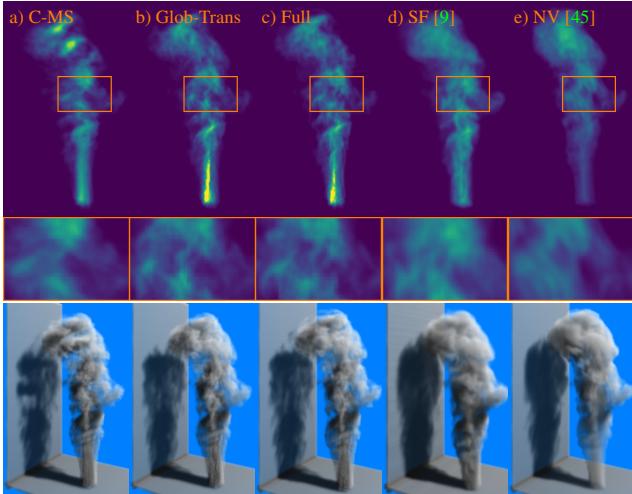
Figure 14: Multi-views (5 targets) reconstruction with data from [9]. Top: perspective density from a $90°$ view. Bottom: rendering with lighting and inc. density from $60°$. Global transport prevents artifacts, *e.g.*, visible in (a), while previous work (d,e) is smoother.

(a) yields a strong smearing out of information along the unconstrained direction of the viewing rays, despite the existence of transport initialization. Although the qualitative examples from a $90°$ angle in Figure 5 top illustrate this behavior, it is even more clearly visible in the supplemental videos. The *coupling* version (b) refines the reconstructed volumes, but still contains noticeable striping artifacts as well as a lack of details. Our *global transport* formulation (c) resolves these artifacts and yields a plausible motion, which, however, does not adhere to the motions observed in the input views. In particular, it yields relatively strong single streaks of transported density which are more diffusive in the real world flow. The discriminator matches the appearance of the reconstruction to the real world smoke, thus guiding the motion reconstruction to adhere to features of real flows (d).

perparameters had to be adapted to work in our setting. As TF is implemented in our framework its grid is also cropped.

We use the original ScalarFlow code, which is based on mantaflow, the solver we used to create our synthetic test case. Due to this shared code-base SF has a certain advantage in the synthetic case. It shows in the visual similarity of the reconstructed motions compared to the reference, as seen in the stream-plots of Figure 14 (f,g). Nevertheless, our method matches the targets better than the SF version. The high transport error of SF in Table 1a might be caused, in part, by the missing inflow as the ScalarFlow algorithm removes the synthetic inflow after reconstruction. SF runs on the full $128 \times 196 \times 128$ grid in the synthetic test case while the real reconstructions from the provided dataset use a grid size of $100 \times 178 \times 100$.

NeuralVolumes typically works with many more viewpoints than we provide. This results in artifacts in our sparse-view setting, *i.e.*, colored or black background is reconstructed within the volume. The reconstruction is however still very good. To compute metrics and rendering we sample a RGBα volume at the resolution of the reference and extract a density via $\rho_{NV} := 0.03[\alpha]_0^1(R + G + B)/3$, which removes black opaque artifacts, thus giving NV an advantage in these comparisons.

B.4. Details for the Single-View Ablation

For the single-view ablation, Figure 5, we focus on the key versions of our previous, multi-view ablation and evaluate them on real targets as the effect of the discriminator is more prominent here. The tomographic reconstruction

References

[1] Bradley Atcheson, Wolfgang Heidrich, and Ivo Ihrke. An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids*, 46(3):467–476, Mar 2009. 2

[2] Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM Trans. Graph.*, 27(5), Dec. 2008. 2

[3] Jesse Belden, Tadd T Truscott, Michael C Axiak, and Alexandra H Techet. Three-dimensional synthetic aperture particle image velocimetry. *Measurement Science and Technology*, 21(12):125403, 2010. 2

[4] Yang Chen, Zhou Yang, Yining Hu, Guanyu Yang, Yongcheng Zhu, Yinsheng Li, Wufan Chen, Christine Toumoulin, et al. Thoracic low-dose ct image processing using an artifact suppressed large-scale nonlocal means. *Physics in Medicine & Biology*, 57(9):2667, 2012. 2

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2020. 11

[6] S.B. Dalziel, G.O. Hughes, and B.R. Sutherland. Whole-Field Density Measurements by 'Synthetic Schlieren'. *Experiments in Fluids*, 28:322–335, 2000. 2

[7] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. In *Advances in neural information processing systems*, 2018. 1

[8] M.-L. Eckert, W. Heidrich, and N. Thuerey. Coupled Fluid Density and Motion from Single Views. *Computer Graphics Forum*, 37(8):47–58, 2018. 2

[9] Marie-Lena Eckert, Kiwon Um, and Nils 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. 1, 2, 3, 5, 6, 7, 12, 14, 15

[10] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 2

[11] Gerrit E Elsinga, Fulvio Scarano, Bernhard Wieneke, and Bas W van Oudheusden. Tomographic particle image velocimetry. *Experiments in fluids*, 41(6):933–947, 2006. 1

[12] Gerrit E Elsinga, Fulvio Scarano, Bernhard Wieneke, and Bas W van Oudheusden. Tomographic particle image velocimetry. *Experiments in fluids*, 41(6):933–947, 2006. 2

[13] Klaus Engel, Markus Hadwiger, Joe M Kniss, Aaron E Lefohn, Christof Rezk Salama, and Daniel Weiskopf. Real-time volume graphics. In *ACM Siggraph 2004 Course Notes*, pages 29–es. 2004. 4

[14] Timothy W Fahringer, Kyle P Lynch, and Brian S Thurow. Volumetric particle image velocimetry with a single plenoptic camera. *Measurement Science and Technology*, 26(11):115201, 2015. 2

[15] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In Eugene Fiume, editor, *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22. ACM, ACM Press / ACM SIGGRAPH, 2001. 1

[16] Christian Fuchs, Tongbo Chen, Michael Goesele, Holger Theisel, and Hans-Peter Seidel. Density estimation for dynamic volumes. *Computers & Graphics*, 31(2):205–211, 2007. 2

[17] Ioannis Gkioulekas, Anat Levin, and Todd Zickler. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 685–701, Cham, 2016. Springer International Publishing. 2

[18] E. Goldhahn and J. Seume. The Background Oriented Schlieren Technique: Sensitivity, Accuracy, Resolution and Application to a Three-Dimensional Density Field. *Experiments in Fluids*, 43(2–3):241–249, 2007. 1

[19] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *stat*, 1050:10, 2014. 2

[20] I. Grant. Particle Image Velocimetry: a Review. *Proceedings of the Institution of Mechanical Engineers*, 211(1):55–76, 1997. 2

[21] James Gregson, Ivo Ihrke, Nils Thuerey, and Wolfgang Heidrich. From capture to simulation: connecting forward and inverse problems in fluids. *ACM Trans. Graph.*, 33(4):139, 2014. 1, 2, 4

[22] James Gregson, Michael Krimerman, Matthias B Hullin, and Wolfgang Heidrich. Stochastic tomography and its applications in 3d imaging of mixing fluids. *ACM Trans. Graph.*, 31(4):52–1, 2012. 2

[23] Jinwei Gu, S.K. Nayar, E. Grinspun, P.N. Belhumeur, and R. Ramamoorthi. Compressive Structured Light for Recovering Inhomogeneous Participating Media. *IEEE Trans. Pattern Analysis and Mach. Int.*, 35(3):555–567, 2013. 2

[24] Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. *ACM Trans. Graph.*, 24(3):812–815, 2005. 2

[25] Philipp Henzler, Niloy J. Mitra, and Tobias Ritschel. Escaping Plato's Cave: 3D Shape From Adversarial Rendering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9984–9993, 2019. 3, 4, 7

[26] Phlipp Henzler, Volker Rasche, Timo Ropinski, and Tobias Ritschel. Single-image Tomography: 3D Volumes from 2D Cranial X-Rays. *Computer Graphics Forum*, 37(2):377–388, 2018. 2

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 7

[28] Philipp Holl, Nils Thuerey, and Vladlen Koltun. Learning to Control PDEs with Differentiable Physics. In *International Conference on Learning Representations*, 2020. 1, 2

[29] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. Differen-

tiable programming for physical simulation. In *International Conference on Learning Representations*, 2020. 3

[30] Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 365–373. Eurographics Association, 2004. 2

[31] Amirul Islam, Sen Jia, and D. B. Neil Bruce. How much position information do convolutional neural networks encode. *International Conference on Learning Representations*, 2020. 9

[32] Omar Javed, Khurram Shafique, and Mubarak Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 26–33. IEEE, 2005. 2

[33] Yu Ji, Jinwei Ye, and Jingyi Yu. Reconstructing gas flows using light-path approximation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2

[34] Alexia Jolicoeur-Martineau. The relativistic discriminator: A key element missing from standard GAN. In *International Conference on Learning Representations*, 2018. 4, 9

[35] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning Category-Specific Mesh Reconstruction from Image Collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[37] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D Mesh Renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 2, 4

[38] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[39] Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. Transport-based neural style transfer for smoke simulations. *ACM Transactions on Graphics*, 38(6):1–11, nov 2019. 3

[40] Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. Lagrangian neural style transfer for fluids. *ACM Transactions on Graphics*, 39(4), July 2020. 3

[41] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015. 5

[42] Anestis Koutsoudis, Blaž Vidmar, George Ioannakis, Fotis Arnaoutoglou, George Pavlidis, and Christodoulos Chamzas. Multi-image 3d reconstruction data evaluation. *Journal of Cultural Heritage*, 15(1):73–79, 2014. 2

[43] Aviad Levis, Yoav Y. Schechner, Amit Aides, and Anthony B. Davis. Airborne three-dimensional cloud tomography. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[44] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 3

[45] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Transactions on Graphics*, 38(4):1–14, 2019. 2, 5, 6, 7, 14, 15

[46] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *Computer Vision – ECCV 2014*, volume 8695 of *Lecture Notes in Computer Science*, pages 154–169, Sept. 2014. 3

[47] MantaFlow, 2018. http://mantaflow.com. 5

[48] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint method. *ACM Trans. Graph.*, 23(3):449–456, Aug. 2004. 2

[49] Enric Meinhardt-Llopis, Javier Sánchez Pérez, and Daniel Kondermann. Horn-schunck optical flow with a multi-scale strategy. *Image Processing on line*, 2013:151–172, 2013. 4

[50] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[51] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3

[52] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 2

[53] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic refraction stereo. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1518–1531, 2011. 1

[54] Przemyslaw Musialski, Peter Wonka, Daniel G Aliaga, Michael Wimmer, Luc Van Gool, and Werner Purgathofer. A survey of urban reconstruction. In *Computer graphics forum*, volume 32(6), pages 146–177. Wiley Online Library, 2013. 2

[55] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019. 2

[56] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), Dec. 2019. 3

[57] Makoto Okabe, Yoshinori Dobashi, Ken Anjyo, and Rikio Onai. Fluid volume modeling from sparse multi-view images by appearance transfer. *ACM Transactions on Graphics*, 34(4):1–10, 2015. 2

[58] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pages 2027–2034, 2013. 2

[59] Charles S Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972. 1

[60] Stephen B Pope. Turbulent flows, 2001. 1, 2

[61] Yiming Qian, Minglun Gong, and Yee-Hong Yang. Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1269–1278, 2017. 1

[62] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Proc. ICLR*, 2016. 2

[63] Connor Schenck and Dieter Fox. Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, 2018. 3

[64] Connor Schenck and Dieter Fox. Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*, pages 317–335, 2018. 8

[65] Andrew Selle, Ronald Fedkiw, ByungMoon Kim, Yingjie Liu, and Jarek Rossignac. An Unconditionally Stable MacCormack Method. *Journal of Scientific Computing*, 35(2):350–371, 2008. 3, 10

[66] Tiancheng Shen, Xia Li, Zhisheng Zhong, Jianlong Wu, and Zhouchen Lin. R-net: Recurrent and recursive network for sparse-view ct artifacts removal. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 319–327. Springer, 2019. 2

[67] M Shur, PR Spalart, M Strelets, and A Travin. Detached-eddy simulation of an airfoil at high angle of attack. In *Engineering turbulence modelling and experiments 4*, pages 669–678. Elsevier, 1999. 1

[68] Emil Y Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008. 2

[69] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[70] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 2

[71] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhöfer. State of the art on neural rendering, 2020. 3

[72] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic fluid surface reconstruction using deep neural network. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21–30, 2020. 1

[73] Kiwon Um, Robert Brand, Philipp Holl, and Nils Fei, Raymond Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in Neural Information Processing Systems*, 2020. 8

[74] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[75] Sebastian Weiss, Robert Maier, Daniel Cremers, Rudiger Westermann, and Nils Thuerey. Correspondence-free material reconstruction using sparse surface constraints. pages 4686–4695, 2020. 8

[76] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Transactions on Graphics*, 37(4):95:1–95:15, 2018. 2, 4, 7

[77] Jinhui Xiong, Ramzi Idoughi, Andres A Aguirre-Pablo, Abdulrahman B Aljedaani, Xiong Dun, Qiang Fu, Sigurdur T Thoroddsen, and Wolfgang Heidrich. Rainbow particle imaging velocimetry for dense 3d fluid velocity imaging. *ACM Trans. Graph.*, 36(4):36, 2017. 1, 2

[78] Guangming Zang, Ramzi Idoughi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. Space-time tomography for continuously deforming objects. *ACM Trans. Graph.*, 37(4), July 2018. 2

[79] Guangming Zang, Ramzi Idoughi, Congli Wang, Anthony Bennett, Jianguo Du, Scott Skeen, William L. Roberts, Peter Wonka, and Wolfgang Heidrich. TomoFluid: Reconstructing Dynamic Fluid from Sparse View Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2020. 1, 2, 3, 5, 6, 7, 14

[80] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[81] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2