VOLSMOKE: RECONSTRUCTING SMOKE FROM ONE VIEW WITH DENSITY-SPACE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering time-varying 3D smoke density and velocity from monocular video would enable accessible volumetric capture and physics-based re-simulation, but the problem is severely underconstrained: each pixel constrains only a ray integral, leaving the density distribution in space ambiguous. Recent methods address this by generating novel views, yet such image-space priors need not correspond to a consistent 3D volume, which can hinder subsequent physical reconstruction. We instead resolve the missing information directly in 3D density space, where a single volume provides a shared explanation across views and a physical state for transport. We introduce VOLSMOKE, a monocular reconstruction framework built around a view-conditioned latent prior over 3D smoke density. At inference, we sample plausible densities from this prior and use render-guided reverse diffusion to align each sample with the observed view. Temporal coordination further lets neighboring frames jointly constrain the otherwise ambiguous density. The resulting density sequence is then jointly refined with inflow source and a continuously divergence-free velocity field under rendering and transport constraints. On the real-world ScalarFlow dataset and a challenging synthetic Plume benchmark, VOLSMOKE substantially improves held-out-view reconstruction and velocity recovery over state-of-the-art novel-view baselines, leading to more accurate re-simulation. Our density prior can be trained from either synthetic simulations or simple tomographic reconstructions from sparse-view real-world captures, without greater data requirements than prior image-space approaches. These results show that density-space completion improves not only appearance reconstruction, but also the physical state required for velocity estimation and re-simulation.

1 INTRODUCTION

Recovering 3D smoke from ordinary video would make volumetric fluid capture substantially more accessible. Beyond novel-view rendering, jointly reconstructing density and velocity enables flow analysis, physics-based re-simulation, and interaction with new forces or obstacles (Zang et al., 2020; Eckert et al., 2019; Chu et al., 2022; Gao et al., 2025). Reducing the capture setup from synchronized multi-camera arrays to a single monocular video would therefore greatly broaden the applicability of volumetric smoke reconstruction. The monocular setting, however, is fundamentally ill-posed. Smoke is a participating medium: each image pixel integrates scattering and attenuation along a camera ray rather than observing a distinct surface point. A single view therefore constrains only the projected density, leaving its distribution along depth highly ambiguous. Infinitely many 3D volumes can explain the same image. Temporal observations provide additional cues, but turbulent motion leaves the underlying 3D structure difficult to infer. The central challenge is thus to resolve this underconstrained volumetric state while satisfying both the rendering constraints imposed by observations and the PDEs governing fluid motion.

Recent methods (Gao et al., 2025; Li et al., 2026) address this ambiguity with generative priors in image space. While demonstrating the value of such priors for monocular fluids, they introduce a structural mismatch: generated 2D views need not be projections of a consistent 3D volume. FluidNexus (Gao et al., 2025) independently generates novel views before reconstructing a visual-physical representation and reports cross-view conflicts, whereas SmokeSVD (Li et al., 2026) uses a progressive 3D looping mechanism to mitigate such inconsistencies. This issue is well known even for opaque objects, where multiview diffusion requires explicit mechanisms to enforce geometric

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

consistency (Shi et al., 2023; Yang et al., 2024a; Kant et al., 2024; Yang et al., 2024b). For smoke, this problem is harder: as a participating medium, even multiview fluid tomography remains a notoriously challenging inverse problem (Ihrke & Magnor, 2004; Gregson et al., 2012; Gu et al., 2013).

We take a different approach: rather than generating missing views, we complete the missing information directly in *density space*. A 3D density is multiview-consistent by construction and provides a unified state linking appearance and dynamics through volume rendering and physical transport. Based on this observation, we introduce VOLSMOKE, a monocular smoke reconstruction framework built around a view-conditioned latent density prior. We first train an implicit autoencoder and a view-conditioned latent diffusion model to learn the distribution of plausible smoke volumes. At inference, the prior proposes plausible densities, while render-guided reverse diffusion adapts them to the observed frames. Temporal coordination further lets neighboring frames jointly constrain the otherwise underdetermined 3D structure. The resulting density sequence is then jointly refined with inflow and a continuously divergence-free velocity field under image and physics constraints. In this way, learned density prior, temporal observations, and physical constraints progressively narrow the space of plausible solutions, yielding a better-conditioned 4D reconstruction.

Importantly, moving the prior from image space to density space does not require more training data. We train the prior using either synthetic simulation densities, where diverse volumetric states and rendered observations are readily available, or approximate densities reconstructed from sparse real-world captures, requiring as few as five views. Experiments show that both sources provide sufficient supervision to improve held-out-view reconstruction, velocity recovery, and density re-simulation over prior novel-view-based approaches.

Our contributions are threefold:

- We introduce density-space priors for monocular smoke reconstruction, completing missing information directly in 3D, yielding a multiview-consistent state for rendering and transport.
- We develop a view-conditioned inference pipeline combining render-guided latent diffusion, temporal coordination across neighboring frames, and source-aware 4D refinement with continuously divergence-free velocity.
- We show that the prior can be trained from synthetic simulations or simple tomography of sparse-view real captures, without greater data requirements than image-space approaches, while improving novel-view reconstruction, velocity recovery, and re-simulation.

2 RELATED WORK

Video-based fluid reconstruction and velocity recovery. ScalarFlow (Eckert et al., 2019) introduced a real-world smoke dataset with synchronized multi-view captures and volumetric reconstructions, providing a widely used benchmark for smoke reconstruction. Subsequent sparse-view methods combine tomographic image formation with spatial and temporal regularization, as in TomoFluid (Zang et al., 2020), or with differentiable transport and learned self-supervision, as in GlobTrans (Franz et al., 2021). Neural approaches further represent appearance and motion as continuous fields: PINF (Chu et al., 2022) jointly recovers smoke and scene appearance, while HyFluid (Yu et al., 2023) augments a neural base flow with vortex particles. More recently, DFKs (Ni et al., 2025) represent continuous flow with compact divergence-free kernels, and LagrangianSplats (Tao et al., 2026) applies them to sparse-view smoke reconstruction. These methods generally assume synchronized multi-view input. FluidNexus (Gao et al., 2025) and SmokeSVD (Li et al., 2026) instead reconstruct from monocular video by generating additional views before physical estimation. In contrast, we place the generative prior directly on density and jointly recover prior-constrained density and DFK velocity from monocular video.

Generative volumetric priors. Denoising diffusion (Ho et al., 2020), latent diffusion (Rombach et al., 2022), and diffusion posterior sampling (Chung et al., 2023) provide expressive priors with differentiable measurement guidance for inverse problems. Most closely related, Cloudy (Leonard et al., 2025) learns a monoplanar latent diffusion model and uses light-transport-aware posterior sampling for static, single-view cloud reconstruction. We extend this density-space strategy to dynamic smoke by conditioning the prior on the observed frame, coordinating posterior samples over time, and coupling the recovered densities to source-aware transport and velocity estimation.

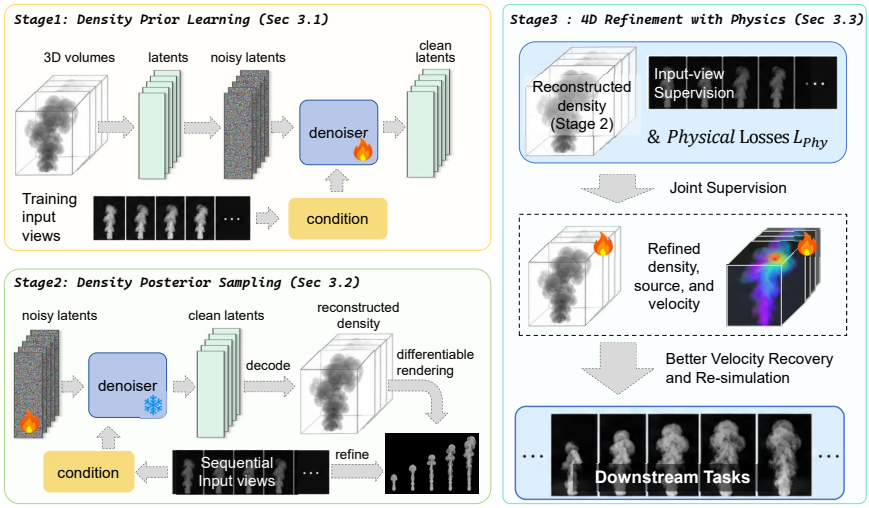


Figure 1: **Method overview.** Stage 1 trains a conditional density-latent denoiser. Stage 2 freezes the denoiser and uses input-view rendering guidance to reconstruct density sequences. Stage 3 jointly refines density, source, and velocity with image and physical losses for downstream re-simulation. Flames mark optimized quantities; the snowflake marks the frozen denoiser.

3 METHOD

Given a monocular video $\mathcal{I} = \{I_k\}_{k=0}^{K-1}$ from a fixed, calibrated camera, we recover a time-varying 3D density ρ_k, a divergence-free velocity field $\mathbf{u}_k$, and an inflow increment s_k between consecutive frames. Fig. 1 summarizes the three stages of VOLSMOKE. Stage 1 learns a view-conditioned diffusion prior over complete 3D densities. Stage 2 freezes this prior and performs render-guided reverse diffusion, with sliding-window constraints that coordinate neighboring frames. Stage 3 jointly refines density, velocity, and inflow using the observed video, source-aware transport, and flow-consistency losses. We use k for video time and τ for diffusion time.

3.1 CONDITIONAL 3D DENSITY PRIOR

Our normalized density volume $\bar{\rho} = \rho / a_{\mathrm{ds}}$ is encoded as a monoplanar latent. The design follows Cloudy (Leonard et al., 2025), a generative model for static single-view cloud:

$$\boldsymbol{h} = E_\phi(\bar{\rho}) \in \mathbb{R}^{C_y \times H_\ell \times W_\ell}, \qquad (C_y, H_\ell, W_\ell) = (100, 128, 128), \tag{1}$$

where the plane spans X–Z and its channels parameterize the upward dimension Y. The decoder evaluates density at $\boldsymbol{\xi} = (\xi_x, \xi_y, \xi_z)$ as

$$\mathbf{g}(\xi_x, \xi_z) = \mathrm{Sample}_{XZ}(U_\psi(h), (\xi_x, \xi_z)), \qquad D_\omega(h; \boldsymbol{\xi}) = f_\omega(\mathrm{Sample}_Y(\mathbf{g}, \xi_y)). \tag{2}$$

Here, U_ψ upsamples the plane while f_ω is the density MLP. We train the autoencoder on densities, freeze it, and standardize h into diffusion latent $\boldsymbol{x}_0$(App. A.2).

Unlike Cloudy's unconditional prior, we condition the latent denoiser on the input view I, yielding $p_\theta(\boldsymbol{x}_0 \mid C(I))$. In $\mathcal{C}(I) = (C^{\mathrm{map}}, C^{\mathrm{global}})$, C^{map} contains occupancy, opacity, interior distance, local occupancy, and edges; and C^{global} summarizes extent, centroid, opacity statistics, and horizontal/vertical profiles. We concatenate C^{map} with the noisy latent and use C^{global} to modulate the denoiser (App. A.3 details the image conditions). At inference, C_k uses only the observed frame.

We use standard noising process v-prediction (Ho et al., 2020; Salimans & Ho, 2022):

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), \quad \boldsymbol{x}_\tau = a_\tau \boldsymbol{x}_0 + \sigma_\tau \boldsymbol{\epsilon}, \quad \boldsymbol{v}_\tau = a_\tau \boldsymbol{\epsilon} - \sigma_\tau \boldsymbol{x}_0, \text{ with } a_\tau = \sqrt{\bar{\alpha}_\tau} \text{ and } \sigma_\tau = \sqrt{1 - \bar{\alpha}_\tau} \tag{3}$$

Writing $\widehat{\boldsymbol{v}}_\theta = \boldsymbol{v}_\theta(\boldsymbol{x}_\tau, \tau, C)$ and $\widehat{\boldsymbol{x}}_0 = a_\tau \boldsymbol{x}_\tau - \sigma_\tau \widehat{\boldsymbol{v}}_\theta$, we train the latent denoiser with

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}\|\widehat{\boldsymbol{v}}_\theta - \boldsymbol{v}_\tau\|_2^2 + \lambda_x \mathbb{E}\|\widehat{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|_2^2 + \lambda_g \mathbb{E}\|\nabla_{XZ}\widehat{\boldsymbol{x}}_0 - \nabla_{XZ}\boldsymbol{x}_0\|_1. \tag{4}$$

The gradient term helps preserve thin structures.

3.2 RENDER-GUIDED SLIDING-WINDOW INFERENCE

Stage 2 adapts the conditional prior to the observed video, producing a temporally coordinated density sequence for Stage 3. Each reverse-diffusion step combines an inverse-rendering correction with temporal-window coordination while remaining anchored to the learned density prior.

Inverse-rendering correction. For ray samples $\boldsymbol{\xi}_i$ separated by δ_i, with extinction κ and appearance $\mathbf{a}_i$, a NeRF-based differentiable renderer (Mildenhall et al., 2020) computes

$$\alpha_i = 1-\exp[-\kappa\rho(\boldsymbol{\xi}_i)\delta_i], \quad T_i = \prod_{j<i}(1-\alpha_j), \quad \widehat{I}(p) = \sum_i T_i\alpha_i\mathbf{a}_i + \left(1-\sum_i T_i\alpha_i\right) I_{\mathrm{bg}}(p). \tag{5}$$

Before posterior sampling, we fit a lightweight NeRF-based renderer using the given monocular views and the density predicted from the image condition alone. Its rendering parameters are then fixed, so Stage 2 posterior sampling updates only the density latent through this differentiable renderer. Agreement with the observed frame is measured by $\mathcal{L}_{\mathrm{rend}}^{(k)} = \sum_{p\in\mathcal{P}} \|\widehat{I}_\tau^{(k)}(p) - I_k(p)\|_2^2$.

Temporal-window coordination. A single image admits many plausible 3D densities, while neighboring frames provide complementary constraints on their shared underlying smoke structure. We therefore encourage temporally nearby reconstructions to agree at a coarse spatial scale. Specifically, we downsample the decoded densities and, for each neighboring pair (k, j), align them by a small 3D translation before measuring their difference. Let $\ell(\rho) = \log(1 + \beta\rho)$ and $\phi(r) = \sqrt{r^2 + \varepsilon^2} - \varepsilon$. The temporal consistency loss is

$$\mathcal{L}_{\mathrm{mc}}^{(k)} = \frac{1}{|\mathcal{N}(k)|} \sum_{j\in\mathcal{N}(k)} \frac{1}{|\Omega|} \sum_{\boldsymbol{\xi}\in\Omega} w_k(\boldsymbol{\xi})\phi\big(\ell(\rho_k(\boldsymbol{\xi})) - \ell(\mathcal{W}(\rho_j; \mathbf{d}_{kj}^*)(\boldsymbol{\xi}))\big)\,. \tag{6}$$

where $\mathcal{N}(k)$ denotes temporal neighbors, Ω is the coarse comparison grid, $\mathcal{W}$ applies the translation $\mathbf{d}_{kj}^* \in \{-1, 0, 1\}^3$ that best aligns the two densities, and w_k emphasizes regions containing smoke. Neighboring densities and the alignment are detached, so the loss updates only the current frame. This temporal coordination lets multiple observations jointly constrain each density estimate, narrowing the set of plausible monocular reconstructions. It aligns only coarse density structure; Stage 3 refines density details and physical motion. App. A.4 specifies the translation search, synchronized per-frame workers, overlap-add across sliding windows, and the auxiliary window penalties.

Combined posterior-sampling update. At each diffusion timestep, we combine the rendering correction and window regularization with an unguided trajectory initialized from the same noise:

$$\mathcal{J}_\tau^{(k)} = \mathcal{L}_{\mathrm{rend}}^{(k)} + \mathcal{L}_{\mathrm{win}}^{(k)} + \lambda_{\mathrm{pr}}(\tau)\mathcal{L}_{\mathrm{pr}}^{(k)}, \tag{7}$$

$$\mathcal{L}_{\mathrm{win}}^{(k)} = r_\tau\lambda_{\mathrm{mc}}\mathcal{L}_{\mathrm{mc}}^{(k)} + \lambda_{\mathrm{src}}\mathcal{L}_{\mathrm{src}}^{(k)} + r_\tau^{\mathrm{cmp}}\lambda_{\mathrm{cmp}}\mathcal{L}_{\mathrm{cmp}}^{(k)}. \tag{8}$$

Here $\mathcal{L}_{\mathrm{pr}}$ anchors the sample to its unguided prior trajectory, $\mathcal{L}_{\mathrm{src}}$ favors compact inflow, and $\mathcal{L}_{\mathrm{cmp}}$ suppresses disconnected density. The factors r_τ and r_τ^{cmp} schedule the corresponding window terms. Loss term details are given in App. A.4 and their ablations in App. B.6. Differentiable decoding and rendering then allow diffusion posterior sampling (Chung et al., 2023) to update the noisy latent while keeping the denoiser and decoder frozen:

$$\boldsymbol{x}_\tau^{(k),+} = \boldsymbol{x}_\tau^{(k)} - \eta_\tau \frac{\nabla_{\boldsymbol{x}_\tau^{(k)}} \mathcal{J}_\tau^{(k)}}{\|\nabla_{\boldsymbol{x}_\tau^{(k)}} \mathcal{J}_\tau^{(k)}\|_2 + \varepsilon_g}. \tag{9}$$

We alternate the guided update with DDIM reverse steps until $\tau = 0$. The final latent decodes to the density sequence used to initialize Stage 3.

3.3 DENSITY-ANCHORED PHYSICAL REFINEMENT

Stage 2 yields densities consistent with the learned prior and input views, but does not enforce physical transport, so structures may appear or disappear between frames. Stage 3 jointly refines density, velocity, and inflow under rendering and transport constraints (Fig. 3).

Density, source, and velocity parameterization. We first restore the Stage 2 density to its original units using the dataset constant a_{ds} (Appendix A.6). Rather than reconstructing a new density from

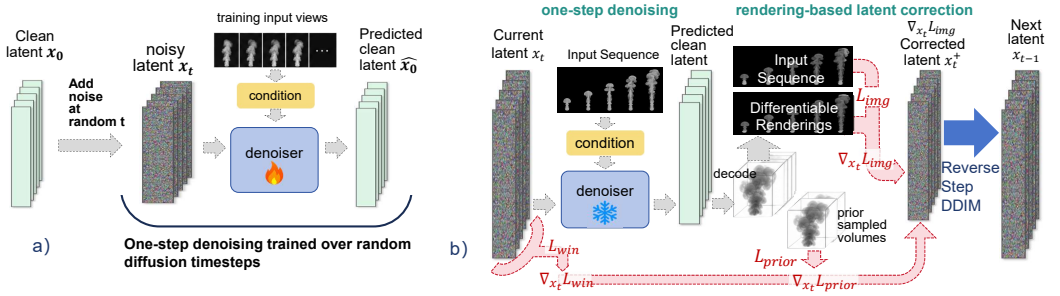


Figure 2: **Stages 1 and 2: conditional density diffusion.** (a) A clean density latent is noised at a random diffusion timestep; image-conditioned denoising predicts the clean latent through the v-parameterization. (b) With the denoiser frozen, the predicted clean latent is decoded and rendered. Input-view image, temporal-window, and prior-anchor losses correct the current noisy latent before the next DDIM reverse step. Fire/snowflake icons denote trainable/frozen networks.

scratch, a spatio-temporal refiner predicts a density residual r_ρ, inflow logit r_s, and update confidence q, defining bounded corrections to the Stage 2 density. The inflow logit parameterizes a non-negative source increment restricted to Ω_{src}. Further details of the refiner are provided in App. B.2. We represent each transition velocity using a continuous field of divergence-free kernels (Ni et al., 2025; Tao et al., 2026), whose centers, radii, and weights are optimized from the density sequence (Appendix A.5). The DFK representation enforces incompressibility by construction. Thus, the density, velocity, and source fields optimized in Stage 3 are:

$$\begin{aligned}
\rho_k &= \rho_u[\text{softplus}(b_k + \delta_{\max} q_k \tanh r_{\rho,k}) - \varepsilon_\rho]_+, \qquad b_k = \text{softplus}^{-1}(\widetilde{\rho}_k/\rho_u + \varepsilon_\rho), \\
\mathbf{u}_k(\boldsymbol{\xi}) &= \sum_{m=1}^{M} \boldsymbol{\psi}(\mathbf{y}_{km})\boldsymbol{\omega}_{km}, \qquad \boldsymbol{\psi}(\mathbf{y}) = (-\mathbf{I}\nabla_{\mathbf{y}}^2 + \nabla_{\mathbf{y}}\nabla_{\mathbf{y}}^\top)\phi(\|\mathbf{y}\|), \qquad \mathbf{y}_{km} = (\boldsymbol{\xi} - \mathbf{c}_{km})/h_{km}, \\
s_k &= \rho_u s_{\max} \,\text{softplus}(r_{s,k} - 6)\mathbf{1}_{\Omega_{\text{src}}}.
\end{aligned} \tag{10}$$

Physics consistency. Midpoint semi-Lagrangian backtracing predicts $\widehat{\rho}_{k+1}^{\text{adv}} = \text{Adv}(\rho_k, \mathbf{u}_k) + s_k$. We penalize its forward mismatch to ρ_{k+1}, the corresponding backward cycle, and mass inconsistency:

$$\mathcal{L}_{\text{tr}} = \sum_k \Phi_w(\widehat{\rho}_{k+1}^{\text{adv}}, \rho_{k+1}), \quad \mathcal{L}_{\text{cyc}} = \sum_k \Phi_w(\text{Adv}([\rho_{k+1} - s_k]_+, -\mathbf{u}_k), \rho_k), \tag{11}$$

$$\mathcal{L}_{\text{mass}} = \sum_k \left(\frac{\sum_\Omega (\widehat{\rho}_{k+1}^{\text{adv}} - \rho_{k+1})}{\sum_\Omega \rho_{k+1} + \epsilon_m} \right)^2, \quad \Phi_w(a, b) = \frac{\sum_{\boldsymbol{\xi}\in\Omega} w_{a,b}(\boldsymbol{\xi})\phi(a(\boldsymbol{\xi}) - b(\boldsymbol{\xi}))}{\sum_{\boldsymbol{\xi}\in\Omega} w_{a,b}(\boldsymbol{\xi})}. \tag{12}$$

Here $w_{a,b}$ emphasizes active-density regions. To limit accumulated drift, we constrain the reconstruction to a recurrently advected rollout of the initial state:

$$\rho_0^{\text{roll}} = \rho_0, \qquad \rho_{k+1}^{\text{roll}} = \text{Adv}(\rho_k^{\text{roll}}, \mathbf{u}_k) + s_k, \qquad \mathcal{L}_{\text{roll}} = \sum_{k=1}^{K-1} \phi(\rho_k^{\text{roll}} - \rho_k). \tag{13}$$

We also softly constrain the DFK velocity by the vorticity form of the Navier–Stokes equations:

$$\mathcal{L}_{\text{vor}} = \sum_k \left\| \boldsymbol{\zeta}_{k+1} - \boldsymbol{\zeta}_{k\to k+1}^{\text{adv}} - \Delta t(\boldsymbol{\zeta}_{k\to k+1}^{\text{adv}} \cdot \nabla)\mathbf{u}_k \right\|_1, \qquad \boldsymbol{\zeta}_k = \nabla \times \mathbf{u}_k. \tag{14}$$

Together with incompressibility, these terms favor physically coherent density evolution.

Image and prior anchoring. Using the same fixed differentiable renderer as in Stage 2, we combine a foreground-weighted robust image residual with temporal image-difference consistency:

$$\mathcal{L}_{\text{rend}} = \sum_k \overline{\phi}_{w_k}(\widehat{I}_k - I_k) + \lambda_t \sum_k \overline{\phi}\Big((\widehat{I}_{k+1} - \widehat{I}_k) - (I_{k+1} - I_k) \Big),$$

where $\overline{\phi}_w$ is a weighted mean Charbonnier penalty; the full definition is given in App. A.5. A proximity term further anchors the refined density to the Stage 2 estimate:

$$\mathcal{L}_{\text{prox}} = \sum_k \|\rho_k - \widetilde{\rho}_k\|_{w_\rho}^2 + \lambda_s \sum_k \|s_k\|_1. \tag{15}$$

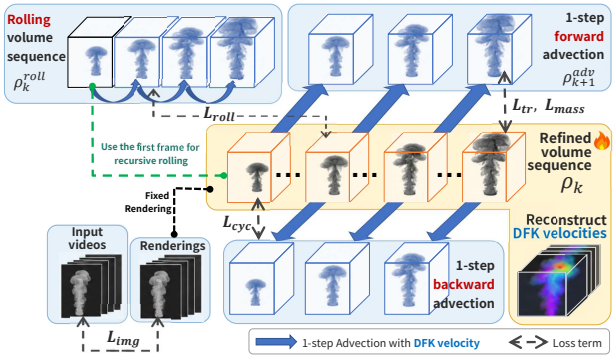


Figure 3: **Stage 3 physical refinement.** DFK velocity couples the refined density sequence through one-step forward transport, backward cycles, and recurrent rollout; L_{rend} anchors the density to the input video. Source increments follow Eq. (10); other regularizers and density acceptance are omitted.

Table 1: Velocity recovery averaged over five Plume test scenes. Rows marked +Stage 3 use our shared physical refiner. All velocity metrics use 99 interval fields/scene. MSE_{all} covers the full grid; $\text{MSE}_{\text{smoke}}$ uses the GT-density-positive region. Cosine also uses a smoke mask, with near-zero velocities excluded (App. B.5).

Method	Cos.↑	Div.↓	MSE_{all} ↓	$\text{MSE}_{\text{smoke}}$ ↓
FluidNexus	−.008	.00113	.002809	.149
SmokeSVD	.061	.00071	.001630	.121
NGT	.037	.00005	.018608	.288
HyFluid	-.073	0.00003	.007279	.099
PINF	-.004	**.00001**	.026725	.189
FluidNexus + Stage 3	−.0006	.00012	.000950	–
SmokeSVD + Stage 3	.188	.00005	.001235	–
Global-trans + Stage 3	.185	.00008	.009136	.110
Ours	**.496**	**.00001**	**.000512**	**.058**

Joint optimization and final 4D field. We jointly infer density and motion with

$$\mathcal{L}_{\text{4D}} = \mathcal{L}_{\text{rend}} + \lambda_{\text{tr}}\mathcal{L}_{\text{tr}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}} + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}} + \lambda_{\text{vor}}\mathcal{L}_{\text{vor}} + \lambda_{\text{prox}}\mathcal{L}_{\text{prox}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \quad (16)$$

All terms are detailed in App. A.5. To avoid physical optimization moving density away from the observed images, we form $\rho_{k,\alpha} = \widetilde{\rho}_k + \alpha(\rho_k - \widetilde{\rho}_k)$ and accept the largest $\alpha \in [0, 1]$ satisfying

$$E_{\text{rend}}(\rho_{k,\alpha}) \le (1 + \delta_{\text{rend}})E_{\text{rend}}(\widetilde{\rho}_k) + \varepsilon_{\text{rend}}. \quad (17)$$

Here E_{rend} is the given-view rendering error. After setting $\rho_k^{\text{final}} = \rho_{k,\alpha^\star}$, we fix density and re-optimize only DFK velocity and inflow:

$$(\Theta_u^{\text{final}}, s^{\text{final}}) = \arg\min_{\Theta_u, s} \mathcal{L}_{\text{4D}}(\rho^{\text{final}}, \mathbf{u}_{\Theta_u}, s). \quad (18)$$

This fits the velocity and source to the finally accepted density evolution.

4 Experiments

4.1 Setup

Data and evaluation. We first aggregate metrics within each scene and then average the five test scenes of each dataset equally. Plume is split into 37/5/5 train/validation/test scenes. Following FluidNexus, ScalarFlow uses `sim010--103` for training and `sim000--009` as held-out scenes; we evaluate on the fixed subset `sim005--009`. Each scene is observed from one fixed, calibrated input camera, although camera poses vary across scenes. ScalarFlow uses `calib3` as input and `calib1/2/4/5` for evaluation over frames 50–149. Plume uses the same 100-frame interval and six held-out cameras spanning a full orbit. The priors are trained only on training scenes (Appendix B.1).

Comparisons and implementation. All baselines use the same scenes, frames, and cameras. FluidNexus and SmokeSVD (Gao et al., 2025; Li et al., 2026) retain their native pipelines, using official ScalarFlow weights and fine-tuning on Plume. We also apply STG and Global-trans (Li et al., 2024; Franz et al., 2021) in together with our Stage 3 so that their reconstructed densities can be evaluated under the same physical refinement. Controlled comparisons using our shared Stage 3, together with ablations, isolate individual design choices (Appendix B.1). Our default setting uses 100 DDIM steps, length-16 windows with stride 8, and 3,600 joint-refinement updates. Reconstructing a 100-frame sequence takes approximately 2 h 46–49 min on eight H20 GPUs, including posterior sampling and refinement (Appendix B.2).

4.2 Reconstruction, Velocity Recovery, and Re-simulation

Velocity evaluation. Plume provides ground-truth velocity. As shown in Table 1, VolSmoke obtains the highest directional agreement (.496 cosine) and the lowest full-grid and smoke-region

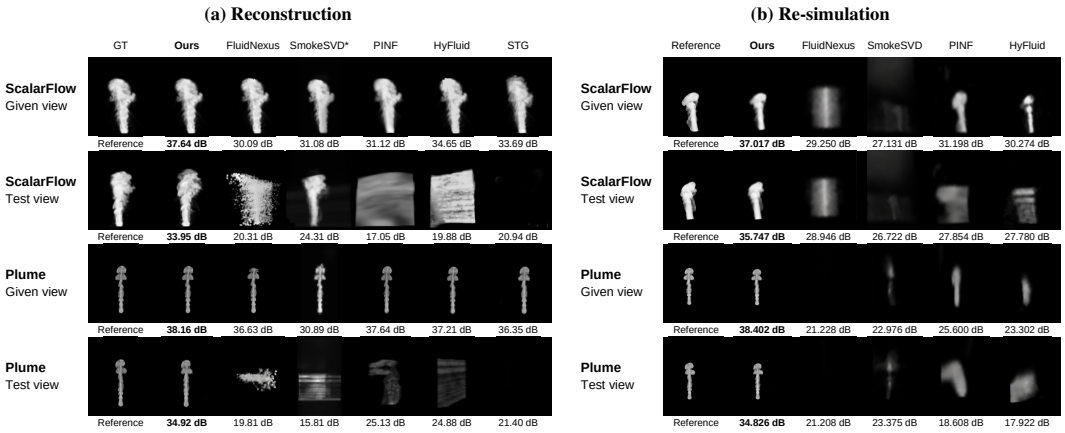


Figure 4: **Reconstruction and re-simulation.** (a) Reconstruction of ScalarFlow and Plume. PINF and STG use 25k iterations. (b) Re-simulation of ScalarFlow and Plume. PSNR values describe the displayed frame, while displayed tiles are cropped frames. Aggregate image metrics are reported in Table 2.

MSE (.000512/.058). Several baselines also produce low-divergence fields but have substantially larger direction or magnitude errors, showing that incompressibility alone does not recover the observed motion. The improvement across all three velocity metrics supports the use of reconstructed density as a physical intermediate state. Appendix B.5 defines the masks and aggregation.

Evaluating Velocity Recovery with a Shared Stage 3. To isolate the effect of our density-space prior on density estimation from the subsequent joint optimization in Stage 3, we apply the same Stage 3 DFK representation, observations, hyperparameters, and optimization budget to density sequences produced by FluidNexus, SmokeSVD, and our method (Table 1). The shared refiner improves SmokeSVD's velocity cosine from .061 to .188, while ours still yields the highest velocity cosine similarity (.496) and the lowest velocity MSE. Thus, even under the same physical optimizer, densities estimated with our density-space prior provide a substantially better state for subsequent motion recovery. Appendix B.5 details the common protocol.

Reconstruction and re-simulation. Reconstruction evaluates the recovered densities directly, whereas re-simulation evolves the first density using the recovered velocity and inflow. Thus, reconstruction more directly reflects density estimation quality, while re-simulation places greater emphasis on the recovered velocity and inflow through their ability to transport density over time. VOLSMOKE ranks first across PSNR, SSIM, and LPIPS on both datasets and on both given and held-out views (Table 2). The gains over the leading baselines persist, and generally widen, after recurrent transport. The velocity metrics directly measure motion accuracy; re-simulation complements them by showing that the recovered velocity and inflow advect density along a coherent evolution whose rendered appearance remains accurate. Fig. 4 shows the corresponding structure over time.

4.3 CONTROLLED CROSSWIND RE-SIMULATION

Experimental setting. To test whether the reconstructed physical state remains reliable under a controlled edit, rather than only replaying the observed motion, we apply the same prescribed crosswind warp to Plume15 frames 50–98:

$$\rho^{\mathrm{wind}}(x, y, z) = \rho(x - y^2/(6H), y, z),$$

with $H = 319$, to each re-simulated sequence and to the ground truth. All methods share the same warp, renderer, cameras, and calibration; differences therefore reflect how well the reconstructed state and its accumulated transport withstand the edit.

Results. After the same edit, VOLSMOKE remains closest to the edited ground truth: held-out-view PSNR is 34.19 dB and density cosine is .9046, compared with 28.87 dB and .2657 for PINF. Fig. 5 shows the same trend in the retained plume structure. This controlled result indicates that the recovered physical state remains valid under downstream edits, rather than only reproducing the observed motion.

Table 2: Novel-view synthesis and re-simulation from single-view reconstructions, averaged over five test scenes per dataset. Given/Test denote the input and held-out cameras. Synthesis renders reconstructed densities; re-simulation renders transported densities (Appendix B.5). Global-trans and STG include our Stage 3 refinement.

Task	View	Metric	**Ours**	FluidNexus	SmokeSVD	NGT	Global-trans + Stage 3	HyFluid	PINF	STG + Stage 3
ScalarFlow Reconstruction	Given	PSNR↑	**36.70**	33.56	28.91	32.85	34.51	26.45	24.83	23.76
		SSIM↑	**.956**	.882	.867	.859	.883	.821	.786	.752
		LPIPS↓	**.081**	.151	.168	.131	.097	.223	.287	.348
	Test	PSNR↑	**32.99**	29.63	26.88	30.05	31.26	23.97	22.31	21.54
		SSIM↑	**.956**	.882	.806	.786	.800	.744	.706	.682
		LPIPS↓	**.081**	.151	.181	.143	.107	.246	.319	.384
ScalarFlow Re-simulation	Given	PSNR↑	**35.68**	32.24	27.73	31.39	33.33	25.14	23.71	22.38
		SSIM↑	**.940**	.858	.829	.838	.868	.795	.754	.715
		LPIPS↓	**.089**	.163	.186	.166	.108	.241	.312	.382
	Test	PSNR↑	**33.83**	28.19	25.12	28.93	29.59	22.58	20.53	20.27
		SSIM↑	**.876**	.741	.723	.758	.773	.681	.620	.609
		LPIPS↓	**.084**	.184	.208	.159	.125	.279	.368	.421
Plume Reconstruction	Given	PSNR↑	**38.00**	31.26	30.58	31.41	32.58	27.62	25.71	24.39
		SSIM↑	**.973**	.889	.876	.869	.919	.834	.798	.764
		LPIPS↓	**.074**	.143	.159	.151	.087	.211	.276	.332
	Test	PSNR↑	**34.10**	28.74	28.16	29.61	30.04	24.85	23.26	22.08
		SSIM↑	**.873**	.817	.807	.819	.847	.750	.722	.692
		LPIPS↓	**.082**	.156	.173	.160	.094	.235	.305	.367
Plume Re-simulation	Given	PSNR↑	**37.09**	30.07	29.19	30.27	31.11	26.34	24.64	23.01
		SSIM↑	**.960**	.844	.829	.846	.892	.804	.767	.728
		LPIPS↓	**.080**	.169	.180	.166	.101	.232	.299	.361
	Test	PSNR↑	**34.92**	27.16	26.97	27.69	28.73	23.14	21.89	20.33
		SSIM↑	**.889**	.768	.761	.779	.818	.701	.673	.622
		LPIPS↓	**.077**	.179	.190	.178	.110	.261	.334	.403

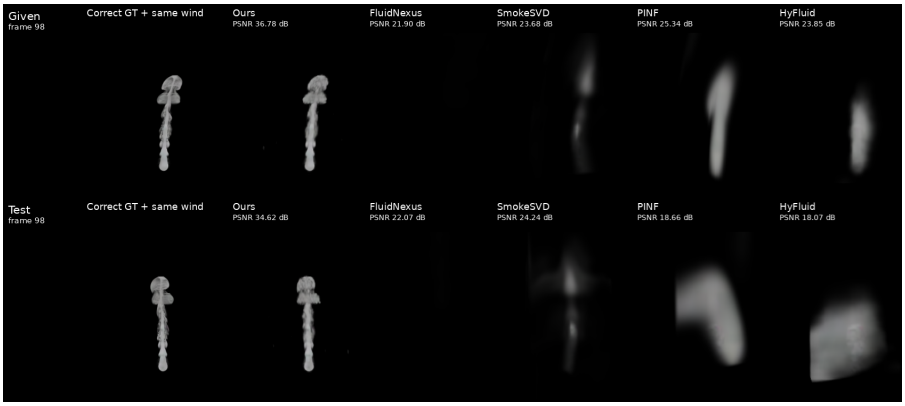


Figure 5: Crosswind replay at frame 98. Columns: GT, Ours, FluidNexus, SmokeSVD, PINF, and HyFluid; rows: input and novel views. All share the warp and renderer; Ours includes a reconstructed continuity residual, while baselines are source-free.

4.4 Ablation studies

Table 3 ablates the density-space prior, temporal coordination, and physical refinement. Removing these components degrades both reconstruction and re-simulation, with consistently larger drops in re-simulation, highlighting the challenge of monocular density reconstruction: similar appearance can arise from different 3D densities and dynamics.

Conditional diffusion. Replacing the learned prior with direct latent optimization reduces held-out PSNR from 34.10 to 25.00 dB and velocity cosine from .496 to .130, showing that the density prior provides a useful solution space. Removing image conditioning lowers PSNR to 29.50 dB and cosine to $-.003$, confirming the importance of view-conditioned inference. Replacing diffusion with a deterministic conditional MLP using the same image features yields 25.33 dB and .008, showing that conditioning alone is insufficient. Together, these ablations demonstrate complementary benefits from the learned density distribution and input-view conditioning.

Table 3: Ablation of the density-space prior, temporal coordination, and physical refinement on five Plume test scenes. Reconstruction metrics are averaged over 100 frames per scene, velocity metrics over 99 transitions, and re-simulation is evaluated on held-out full-orbit views.

	Reconstruction PSNR ↑		Velocity		Re-sim. PSNR ↑	
Method	Given	Test	Div. ↓	Cos. ↑	Given	Test
Full pipeline (DFK)	**38.00**	**34.10**	0.00001	**.496**	37.09	**34.92**
w/o diffusion (same decoder, latent opt.)	31.65	25.00	0.00001	.130	30.69	26.31
Conditional MLP	29.85	25.33	0.00001	.008	28.33	23.21
w/o render guidance	31.50	23.94	0.00001	.421	32.03	24.88
w/o temporal-window synchronization	33.50	32.49	0.00001	.255	32.98	32.76
w/o motion compensation	37.30	33.89	0.00001	.440	37.33	33.10
w/o conditioning	30.78	29.50	0.00001	-.003	29.41	27.99
w/o source term	36.50	33.86	0.00001	.331	36.41	30.88
Grid velocity + soft divergence	36.91	33.07	0.03010	.188	30.15	27.61

Sliding-window constraints. Removing temporal synchronization lowers held-out PSNR from 34.10 to 32.49 dB and velocity cosine from .496 to .255. Removing only motion compensation gives 33.89 dB and .440, showing that coarse cross-frame consistency provides most of the benefit, with explicit alignment contributing further. The larger degradation in velocity highlights the importance of jointly constraining ambiguous monocular density estimates across neighboring frames.

Physical parameterization. Removing inflow has little effect on reconstruction PSNR but lowers velocity cosine to .331 and held-out re-simulation PSNR to 30.88 dB. Replacing DFK with a grid velocity and soft divergence yields .188 cosine, .03010 divergence, and 27.61 dB re-simulation PSNR. Thus explicit inflow and analytical incompressibility contribute primarily to the transportable physical state, rather than only to input-view appearance.

5 Discussion and limitations

What density-space completion establishes. A decoded density explains all views through a single shared volume, avoiding the need to reconcile independently generated target images. The held-out-view and re-simulation gains indicate useful 3D structure and transportability. However, monocular ambiguity remains, and velocity is not uniquely determined. The shared-refiner and temporal ablations further show that the recovered state is more useful for motion estimation, although they cannot isolate every upstream architectural or supervision difference.

Ambiguity and physical scope. Even under incompressibility, scalar transport faces challenges in identifying velocity components tangent to the view direction(Appendix A.6). DFKs enforce zero divergence by construction, while the transport and vorticity terms remain approximate constraints rather than a complete forced Navier–Stokes solver. Our re-simulations also use velocity and inflow fitted within the observed interval. They therefore do not establish future prediction or responses to unseen forces and obstacles.

Generalization and cost. The condition encoder uses image-grid features without explicit camera pose; calibrated rendering provides the geometric link to the volume. For Plume, density-derived training projections are replaced by observed-image features at inference (Appendix A.3), which may limit transfer to unfamiliar views or smoke structures. Approximate multiview training volumes may also pass reconstruction bias to the prior. The method further assumes camera calibration and requires iterative render-guided sampling followed by sequence-specific physical refinement, precluding real-time use.

6 Conclusion

We introduced VolSmoke, which completes monocular smoke observations directly in density space. Its conditional volumetric prior, render-guided temporal coordination, and source-aware physical refinement connect image reconstruction to a transportable 3D state. Experiments on synthetic and captured smoke show consistent gains in reconstruction, velocity recovery, and transport over the observed interval. Future work includes camera-aware conditioning, more efficient inference, uncertainty estimation, and evaluation beyond the fitted sequence.

REPRODUCIBILITY STATEMENT

Appendix A specifies the representation, conditioning, inference, and physical objectives. Appendices B.1 and B.5 document the data splits, camera and frame selections, baseline protocols, and evaluation metrics. Appendix B.2 reports network architectures, optimization details, and compute settings. We will release the code, trained models, experiment configurations, and data-generation/evaluation scripts to support reproducibility.

ETHICS STATEMENT

This work studies smoke reconstruction using synthetic simulations and controlled capture data. Under monocular ambiguity, visually plausible renderings do not certify an accurate physical state. The method has not been validated for safety-critical flow measurement.

AI USE STATEMENT

In this work, we used generative AI tools to assist with intermediate iterations of manuscript writing, the conceptualization of illustrative figures, the summarization and communication of experimental results, and the design of ablation studies. The overall research direction, methodological framework, and main scientific narrative were determined by the authors. Ablation designs suggested or refined with AI assistance were reviewed, selected, and finalized by the authors before execution. Generative AI tools were also used to assist with organizing and summarizing related literature, improving paper structure and readability, formatting references, and designing the webpage for the supplemental material. For writing, figures, and result summaries, the initial content and intended scientific message were provided by the authors, while AI was used only for intermediate iteration; the final versions were reviewed, revised, and approved by the authors.

Generative AI tools were not used to generate experimental data or reported measurements, implement the proposed method, formulate mathematical claims, or provide critical ingredients for proving mathematical claims. Tasks such as assisting in the writing of proofs and cleaning or reformatting datasets were not applicable to this work. AI was also not used to support qualitative or thematic data analysis.

All implementations, experiments, quantitative results, methodological decisions, and scientific claims were developed by the authors. All AI-assisted text, literature summaries, figures, ablation designs, interpretations, and supplemental materials were checked for correctness and consistency with the actual method, experimental results, and cited literature. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Mengyu Chu, 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*, 41(4), 2022. doi: 10.1145/3528223.3530169.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

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*, 38(6), 2019. doi: 10.1145/3355089.3356545.

Erik Franz, Barbara Solenthaler, and Nils Thuerey. Global transport for fluid reconstruction with learned self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1632–1642, 2021.

Erik Franz, Barbara Solenthaler, and Nils Thuerey. Learning to estimate single-view volumetric flow motions without 3D supervision. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=2vmGv5wPDBZ`.

Yue Gao, Hong-Xing Yu, Bo Zhu, and Jiajun Wu. FluidNexus: 3d fluid reconstruction and prediction from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26091–26101, 2025.

James Gregson, Michael Krimerman, Matthias B. Hullin, and Wolfgang Heidrich. Stochastic tomography and its applications in 3d imaging of mixing fluids. *ACM Transactions on Graphics*, 31(4):52:1–52:10, 2012.

Jinwei Gu, Shree K. Nayar, Eitan Grinspun, Peter N. Belhumeur, and Ravi Ramamoorthi. Compressive structured light for recovering inhomogeneous participating media. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):555–567, 2013. doi: 10.1109/TPAMI.2012.130.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Ivo Ihrke and Marcus Magnor. Image-based tomographic reconstruction of flames. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 367–375, 2004. doi: 10.2312/SCA/SCA04/367-375.

Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. SPAD: Spatially aware multi-view diffusers. In *CVPR*, pp. 10026–10038, 2024.

Ludwic Leonard, Nils Thuerey, and Ruediger Westermann. Light transport-aware diffusion posterior sampling for single-view reconstruction of 3d volumes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16163–16174, 2025.

Chen Li, Shanshan Dong, Sheng Qiu, Jianmin Han, Yibo Zhao, Zan Gao, Taku Komura, and Kemeng Huang. SmokeSVD: Smoke reconstruction from a single view via progressive novel view synthesis and refinement with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8508–8520, 2024.

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 *European Conference on Computer Vision*, pp. 405–421, 2020. doi: 10.1007/978-3-030-58452-8_24.

Xingyu Ni, Jingrui Xing, Xingqiao Li, Bin Wang, and Baoquan Chen. Representing flow fields with divergence-free kernels for reconstruction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(4), 2025. doi: 10.1145/3747872.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjoern Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Ningxiao Tao, Baoquan Chen, and Mengyu Chu. Lagrangiansplats: Divergence-free transport of gaussian primitives for fluid reconstruction. *arXiv preprint arXiv:2605.09299*, 2026.

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. doi: 10.1109/TIP.2003.819861.

Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. ConsistNet: Enforcing 3d consistency for multi-view images diffusion. In *CVPR*, pp. 7079–7088, 2024a.

Xianghui Yang, Yan Zuo, Sameera Ramasinghe, Loris Bazzani, Gil Avraham, and Anton van den Hengel. ViewFusion: Towards multi-view consistency via interpolated denoising. In *CVPR*, pp. 9870–9880, 2024b.

Hong-Xing Yu, Yang Zheng, Yuan Gao, Yitong Deng, Bo Zhu, and Jiajun Wu. Inferring hybrid neural fluid fields from videos. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63595–63608, 2023.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1870–1879, 2020. doi: 10.1109/CVPR42600.2020.00194.

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*, pp. 586–595, 2018.

A Detailed method formulation

A.1 Variables and inference map

Constraints and inference objectives. Given a monocular video, we seek densities ρ_k, transition velocities $\boldsymbol{u}_k$, and inflow increments s_k. A valid reconstruction should render to the observed frames, transport density through time, remain incompressible, and approximately follow vorticity evolution. We summarize these requirements as

$$\begin{aligned} I_k &\simeq \mathcal{R}_{\pi,\boldsymbol{\gamma}}(\rho_k), \\ \rho_{k+1} &\simeq \mathrm{Adv}(\rho_k, \boldsymbol{u}_k, \Delta t) + s_k, \\ \nabla\cdot \boldsymbol{u}_k = 0, &\qquad \boldsymbol{\zeta}_k = \nabla\times\boldsymbol{u}_k, \\ \boldsymbol{\zeta}_{k+1} &\simeq \mathrm{Adv}(\boldsymbol{\zeta}_k, \boldsymbol{u}_k, \Delta t) + \Delta t\,(\boldsymbol{\zeta}_k\cdot\nabla)\boldsymbol{u}_k. \end{aligned} \tag{19}$$

These constraints alone do not resolve the depth ambiguity of one view. Stage 1 therefore provides a view-conditioned prior over complete 3D density latents:

$$C_k = \mathcal{C}(I_k), \qquad \boldsymbol{x}_{0,k} \in \mathcal{Z}_\rho, \qquad \boldsymbol{x}_{0,k} \sim p_\theta(\boldsymbol{x}_0 \mid C_k). \tag{20}$$

Stage 2 samples this prior and uses the observed video to guide each sample:

$$\widetilde{\boldsymbol{x}}_{0,0:K-1} = \mathrm{RenderGuidedSample}(p_\theta(\boldsymbol{x}_{0,0:K-1} \mid C_{0:K-1}), \mathcal{L}_{\mathrm{rend}})\,. \tag{21}$$

Stage 3 then refines the decoded density together with velocity and inflow so that the final fields satisfy the rendering and physical objectives:

$$(\widehat{\rho}, \widehat{\boldsymbol{u}}, \widehat{s}) = \underset{\rho,\boldsymbol{u},s}{\arg\min} \quad \mathcal{L}_{\mathrm{rend}} + \lambda_{\mathrm{phys}}\mathcal{L}_{\mathrm{phys}} + \lambda_{\mathrm{prox}}\mathcal{L}_{\mathrm{prox}}(\rho, \widetilde{\rho}). \tag{22}$$

For reference, the notation used above is

$$\mathcal{I} = \{I_k\}_{k=0}^{K-1}, \quad I_k \in [0,1]^{H\times W\times 3}, \quad \rho_k : \Omega \to \mathbb{R}_{\geq 0}, \quad \mathbf{u}_k : \Omega \to \mathbb{R}^3, \quad s_k : \Omega \to \mathbb{R}_{\geq 0}. \tag{23}$$

Here s_k is the density injected during $[k, k+1]$; a predicted rate is multiplied by Δt. The renderer $\mathcal{R}_{\pi,\boldsymbol{\gamma}}$ uses calibrated camera π and fixed parameters $\boldsymbol{\gamma}$, and Adv denotes semi-Lagrangian advection. Our velocity representation is exactly divergence-free, whereas the vorticity equation is used only as a soft constraint. The sampled density provides the volumetric state from which motion is estimated.

The complete inference path is

$$\boldsymbol{x}_N^{(k)} \sim \mathcal{N}(0, \mathbf{I}) \xrightarrow{\text{reverse diffusion + guidance}} \widetilde{\boldsymbol{x}}_0^{(k)} \xrightarrow{\text{denormalize + decode}} \widetilde{\rho}_k \xrightarrow{\text{4D refinement}} (\rho_k, \mathbf{u}_k, s_k). \tag{24}$$

We invert latent standardization, decode the normalized density $\widehat{\widetilde{\rho}}_k$, and restore the dataset units with $\widetilde{\rho}_k = a_{\mathrm{ds}}\widehat{\widetilde{\rho}}_k$. The same statistics and scale are used for every scene and frame. The sampler uses the diffusion prior implicitly; the subsequent sequence-specific optimization uses $\mathcal{L}_{\mathrm{rend}}$ for image agreement, $\mathcal{L}_{\mathrm{phys}}$ for transport and motion constraints, and $\mathcal{L}_{\mathrm{prox}}$ to keep the refined density close to the sampled one.

A.2 Implicit representation details

The latent representation must reconstruct sparse smoke while preserving one consistent density scale. The continuous decoder in Eq. (2) predicts $\bar{\rho} = \rho/a_{\mathrm{ds}}$, where a_{ds} is fixed for the entire dataset. To cover both empty space and smoke regions, we draw queries from $q(\boldsymbol{\xi}) = (1-r_d)q_u(\boldsymbol{\xi}) + r_d q_d(\boldsymbol{\xi})$, a mixture of uniform and density-biased samples. For residual $e(\boldsymbol{\xi}) = D_\omega(E_\phi(\bar{\rho}); \boldsymbol{\xi}) - \bar{\rho}(\boldsymbol{\xi})$, the autoencoder objective is

$$\begin{aligned} \phi_n(e) &= (1-a_n)e^2 + a_n|e|, \\ \mathcal{L}_{\mathrm{AE}} &= \mathbb{E}_{\boldsymbol{\xi}\sim q}[w_+(\boldsymbol{\xi})\phi_n(e(\boldsymbol{\xi}))] + \lambda_z\|h\|_2^2 + \lambda_{\mathrm{TV}}\,\mathrm{TV}_{XZ}(h). \end{aligned} \tag{25}$$

Here $a_n \in [0,1]$ interpolates between quadratic and robust absolute penalties, while w_+ emphasizes occupied regions. We then standardize each latent channel using statistics from all training scenes, frames, and spatial locations:

$$x_{0,c} = \frac{h_c - \mu_c}{\max(\sigma_c, \varepsilon_\sigma)}. \tag{26}$$

The statistics (μ_c, σ_c) are fixed at inference. They standardize latent features and are separate from the scalar density scale a_{ds}.

A.3 CONDITIONING ARCHITECTURE

The denoiser receives a spatial condition map and a global descriptor. The spatial features are concatenated with the noisy latent, while the global descriptor modulates the latent, time embedding, and output:

$$\widetilde{\boldsymbol{x}}_\tau^{(k)} = a_x(C_k^{\text{global}}) \odot \boldsymbol{x}_\tau^{(k)} + b_x(C_k^{\text{global}}), \tag{27}$$

$$\widetilde{\mathbf{e}}_\tau^{(k)} = \mathbf{e}_\tau + h_C(C_k^{\text{global}}), \tag{28}$$

$$\widehat{v}_\theta^{(k)} = a_o(C_k^{\text{global}}) \odot U_\theta([\widetilde{\boldsymbol{x}}_\tau^{(k)}, F_C(C_k^{\text{map}})], \widetilde{\mathbf{e}}_\tau^{(k)}) + b_o(C_k^{\text{global}}). \tag{29}$$

Feature definitions. The condition encoder uses a spatial morphology map and a compact global descriptor, both computed from an image-derived opacity proxy A. Let A be normalized by its spatial maximum (with denominator floor 10^{-6}), bilinearly resized to 128×128, and clamped to $[0, 1]$. The shared morphology builder returns $M(A) = [B, A, D, \overline{B}, E]$: occupancy $B = \mathbb{1}[A > 0.05]$; opacity A; an approximate interior distance D from 32 repeated 3×3 erosions, normalized by its maximum; a 9×9 average-pooled mask $\overline{B}$; and the Sobel-gradient magnitude E of A, normalized by its maximum. These are image-grid channels, not world-coordinate fields. The raw 43-D descriptor $g(A)$ concatenates, in order: occupied-area fraction (1), normalized bounding-box coordinates (x_0, y_0, x_1, y_1) (4), box width and height (2), opacity-weighted foreground centroid (x, y) (2), mean and maximum foreground opacity (2), and vertical and horizontal profiles of AB obtained by adaptive average pooling (16 each). Thus $1 + 4 + 2 + 2 + 2 + 16 + 16 = 43$. Coordinates are normalized image coordinates; the profiles are not profiles along calibrated world axes.

ScalarFlow image-only path. For ScalarFlow, both conditions are built directly from the observed optical-depth image. The supplied launcher selects `photo_resize_flip_rows` in `build_photo_conditions`. It converts the observed optical-depth image to $A = 1 - \exp(-\tau)$, flips its row direction to match the frozen prior's image-orientation convention, and applies the shared builder. The result is $C^{\text{map}} = M(A)$ and $C^{\text{global}} = g(A)$. This flip affects only conditioning; metric images and renderer rays retain their original orientation. The default path performs no camera-dependent reprojection. The code also provides calibrated ray-cone proxy modes, but they are separate options and are not implied by this default.

Plume image-only path. For Plume, the condition starts from a cropped RGB observation and its opacity proxy. The supplied `04_posterior_plume_image_only.sh` launcher sets `PINF_PLUME_XVIEW_MODE=pinf_proxy` and enables strict image-only inference. The builder crops the given RGB image to rows $[109, 854)$ and columns $[87, 460)$ in the supplied image layout, then bilinearly resizes it to 128×128. From this RGB crop J, it computes luminance $L = 0.299J_R + 0.587J_G + 0.114J_B$, subtracts the median border luminance in absolute value, and normalizes the resulting opacity proxy. The raw inputs are

$$C^{\text{map}} = [J, M(A), M(A)] \in \mathbb{R}^{13\times128\times128}, \qquad g_{\text{Plume}} = [g(A), g(A)] \in \mathbb{R}^{86}.$$

Channels 1–3 are RGB, 4–8 are the five morphology maps in the order above, and 9–13 repeat those maps. The two 43-D blocks likewise repeat the same raw descriptor. The 86-D vector is subsequently bounded by training-set feature minima/maxima, standardized using training-set means and scales, and clipped to $[-8, 8]$ by the default inference transform. Branch-specific statistics can therefore make the two standardized blocks differ. The second branch supplies neither a second observation nor an estimated orthogonal view. A zero-filled second branch is another supported option, distinct from the default proxy mode.

Training provenance and coordinate interpretation. Training may construct conditions from known training densities, but test-time inference uses only the observed image. The supplied ScalarFlow axis-1 training builder obtains its proxy by integrating a training density along array axis 1, producing an X–Z projection. The supplied Plume training builder instead concatenates the given-view RGB/morphology with morphology from a training-density X-integral (a Y–Z projection, flipped vertically for image orientation). Thus the common morphology operator does not imply identical training and inference inputs: the image-only Plume launcher replaces that training projection with the duplicate observed-image branch above. This is a conditioning-distribution change. The separate

legacy `oracle` inference mode reads a density volume and is excluded by the strict image-only launcher; it is not a monocular inference protocol.

A.4 Posterior-sampling details

Rendering and guidance objectives. Stage 2 keeps the denoiser fixed and updates the noisy latent through the input-view rendering loss. The renderer and guidance objective are given in Eq. (5) and Section 3.2.

Alignment, windows, and prior anchor. Rendering constrains each frame separately, so we also coordinate nearby density samples. Let $\ell(\rho) = \log(1 + \beta\rho)$, $\mathcal{W}$ be a zero-padded translation, and $\mathcal{D} = \{-1, 0, 1\}^3$. We align each detached neighbor to the current coarse density by

$$\mathbf{d}^*_{kj} = \arg\min_{\mathbf{d}\in\mathcal{D}} \frac{1}{|\Omega_k|} \sum_{\boldsymbol{\xi}\in\Omega_k} \left|\ell(\rho_k(\boldsymbol{\xi})) - \ell(\mathcal{W}(\rho_j; \mathbf{d})(\boldsymbol{\xi}))\right|. \tag{30}$$

The inflow and component terms in Eqs. (7)–(8) handle complementary structure. To keep guidance near the learned prior, we also run an unguided DDIM trajectory (Song et al., 2021) from the same noise and use $(\boldsymbol{x}^{(k)}_{0,\mathrm{pr}}, \rho^{(k)}_{\mathrm{pr}})$ as the anchor:

$$\mathcal{L}^{(k)}_{\mathrm{pr}} = \lambda_{\mathrm{pr},x}\overline{\phi}(\widehat{\boldsymbol{x}}^{(k)}_0 - \boldsymbol{x}^{(k)}_{0,\mathrm{pr}}) + \lambda_{\mathrm{pr},\rho}\overline{\phi}_w(\ell(\rho_k) - \ell(\rho^{(k)}_{\mathrm{pr}})). \tag{31}$$

The renderer uses fixed extinction κ, appearance $\mathbf{a}_i$, and camera calibration. Its computation graph connects the rendering loss to $\boldsymbol{x}_\tau$; denoiser weights are not updated.

Each frame has one reverse-diffusion worker. Workers exchange detached, low-resolution densities with their immediate neighbors, so Eq. (6) back-propagates only through the current frame. Overlapping windows are merged by center-weighted overlap-add. For the bottom-source prior, define $p_k(x, z) \propto \max_{y\in\Omega_{\mathrm{bottom}}} \rho_k(x, y, z)$. If its variance is V_k, the compact-source term is

$$\mathcal{L}^{(k)}_{\mathrm{src}} = \mathrm{ReLU}(V_k - V_0)^2. \tag{32}$$

Detached component suppression. Let $\rho^+_k(\boldsymbol{\xi}) = \max(\rho_k(\boldsymbol{\xi}), 0)$ on the coarse density grid Ω, and let sg denote the stop-gradient operator. We identify significant occupied voxels by

$$\tau_k = \max\left(\tau_{\mathrm{abs}}, \tau_{\mathrm{rel}} \max_{\boldsymbol{\xi}\in\Omega} \mathrm{sg}(\rho^+_k(\boldsymbol{\xi}))\right), \qquad B_k(\boldsymbol{\xi}) = \mathbf{1}\left[\mathrm{sg}(\rho^+_k(\boldsymbol{\xi})) \geq \tau_k\right]. \tag{33}$$

Let $\mathrm{CC}_{26}(B_k)$ be the set of three-dimensional 26-connected components, where voxels sharing a face, edge, or corner are adjacent. We retain the largest component by voxel count and dilate its support using face-adjacent neighbors:

$$C^\star_k = \arg\max_{C\in\mathrm{CC}_{26}(B_k)} |C|, \qquad M_k = \mathrm{sg}\left(\mathrm{Dil}_6^{(n_{\mathrm{dil}})}(\mathbf{1}_{C^\star_k})\right). \tag{34}$$

Here Dil_6 is binary dilation with a structuring element consisting of the center voxel and its six face-adjacent neighbors. The component-suppression loss averages log-density over all voxels outside this retained support:

$$\mathcal{L}^{(k)}_{\mathrm{cmp}} = \frac{\sum_{\boldsymbol{\xi}\in\Omega}(1 - M_k(\boldsymbol{\xi}))\log\big(1 + \beta\rho^+_k(\boldsymbol{\xi})\big)}{\sum_{\boldsymbol{\xi}\in\Omega}(1 - M_k(\boldsymbol{\xi}))}. \tag{35}$$

We define this loss as zero if no significant component exists, if the peak density is at most 10^{-8}, or if the denominator vanishes. We use $\tau_{\mathrm{abs}} = 5 \times 10^{-4}$, $\tau_{\mathrm{rel}} = 0.03$, $\beta = 20$, and $n_{\mathrm{dil}} = 2$ on a $32 \times 64 \times 32$ grid. The denominator counts all exterior voxels, including those with zero density. Threshold selection, component extraction, and dilation are detached, so gradients act only on the exterior log-density penalty. This discourages disconnected density while leaving the retained component and its local neighborhood unpenalized by this term.

For normalized sampling progress $p \in [0, 1]$, the Plume component coefficient is $300 r_{\mathrm{cmp}}(p)$, where

$$r_{\mathrm{cmp}}(p) = \left[\mathrm{clip}\left(\frac{p - 0.35}{0.65}, 0, 1\right)\right]^2. \tag{36}$$

The combined weighted motion, source, and component penalties are capped at 0.25 times the rendering loss; this cap applies to the window objective rather than the definition of $\mathcal{L}_{\mathrm{cmp}}$. Plume enables this component term, while weight zero disables it. The anchor in Eq. (31) uses the same image condition without rendering guidance. A PSNR hysteresis controller activates the rendering gradient only in its configured diffusion-time range.

A.5 PHYSICAL-REFINEMENT LOSSES

Bounded density corrections and field parameterization. Stage 3 must improve transport without allowing density to drift far from the Stage 2 result. Equation (10) therefore combines a bounded density correction with DFK velocity and non-negative inflow. Confidence modulates only the density correction.

Continuous DFK velocity and source. We use one independent DFK per frame transition, following the released LagrangianSplats velocity module. Its $M = 10^5$ centers are initialized by Poisson-disk sampling in a box of size (L_x, L_y, L_z), with

$$h_0 = 6\left(\frac{3L_xL_yL_z}{4\pi M}\right)^{1/3}, \qquad \boldsymbol{\omega}_{km} = \mathbf{0}. \tag{37}$$

Centers, radii, and weights are then optimized jointly, with $h_{km} \geq 10^{-6}$. For $\mathbf{y} = (\boldsymbol{\xi} - \mathbf{c}_{km})/h_{km}$ and $r = \|\mathbf{y}\| < 1$, we evaluate the Wendland C^4 kernel in the expanded form used by the public implementation:

$$\begin{aligned} a(r) &= 30(1-r)^4, & b(r) &= (1 + 4r - 5r^2)(1-r)^4, \\ \boldsymbol{\psi}_{km}(\boldsymbol{\xi})\boldsymbol{\omega}_{km} &= \left([2b(r) - a(r)r^2]\mathbf{I} + a(r)\mathbf{y}\mathbf{y}^\top\right)\boldsymbol{\omega}_{km}, \end{aligned} \tag{38}$$

The kernel is zero for $r \geq 1$. In normalized coordinates it is generated by $(-\mathbf{I}\nabla^2_{\mathbf{y}} + \nabla_{\mathbf{y}}\nabla^\top_{\mathbf{y}})\phi$; therefore $\nabla_{\mathbf{y}} \cdot \boldsymbol{\psi} = -\nabla_{\mathbf{y}}(\nabla^2_{\mathbf{y}}\phi) + \nabla_{\mathbf{y}}(\nabla^2_{\mathbf{y}}\phi) = 0$ analytically. Because $\nabla_{\boldsymbol{\xi}} = h_{km}^{-1}\nabla_{\mathbf{y}}$ for each isotropic kernel, the divergence also vanishes in world coordinates. A spatial hash only reduces the number of kernels queried. The inflow head predicts one non-negative increment per transition within Ω_{src}; q_k gates neither velocity nor inflow.

RK2 transport. We transport grid quantities with midpoint backtracing:

$$\begin{aligned} \boldsymbol{\xi}_{1/2} &= \boldsymbol{\xi} - \tfrac{1}{2}\Delta t\, \mathbf{u}_k(\boldsymbol{\xi}), & \boldsymbol{\xi}_b &= \boldsymbol{\xi} - \Delta t\, \mathbf{u}_k(\boldsymbol{\xi}_{1/2}), \\ \mathrm{Adv}_{\mathrm{RK2}}(f, \mathbf{u}_k)(\boldsymbol{\xi}) &= \mathrm{TriLinear}(f, \boldsymbol{\xi}_b). \end{aligned} \tag{39}$$

Both velocity queries evaluate the continuous DFK; trilinear sampling is used only for the advected grid field. The vorticity loss evaluates DFK vorticity on the grid, advects it, adds vortex stretching, and compares it with the next transition. It is a soft reconstruction constraint rather than a complete Navier–Stokes solver: it omits viscous diffusion and external-force curl, and does not estimate temperature or buoyancy. The source s_k models density inflow, not momentum forcing. The backward cycle in Eq. (42) is also approximate because semi-Lagrangian advection is dissipative and the reverse step removes inflow with a positive clamp.

Pairwise, recurrent, and rendering losses. The pairwise losses check each adjacent transition. Using the active-density-weighted Charbonnier distance Φ_w, we define forward transport, backward cycle, and total-density agreement as

$$\mathcal{L}_{\mathrm{tr}} = \sum_k \Phi_w(\widehat{\rho}^{\mathrm{adv}}_{k+1}, \rho_{k+1}), \tag{40}$$

$$\mathcal{L}_{\mathrm{cyc}} = \sum_k \Phi_w(\mathrm{Adv}_{\mathrm{RK2}}([\rho_{k+1} - s_k]_+, -\mathbf{u}_k), \rho_k)\,, \tag{41}$$

$$\mathcal{L}_{\mathrm{mass}} = \sum_k \left(\frac{\sum_\Omega \widehat{\rho}^{\mathrm{adv}}_{k+1} - \sum_\Omega \rho_{k+1}}{\sum_\Omega \rho_{k+1} + \varepsilon_m}\right)^2. \tag{42}$$

Pairwise agreement does not control accumulated drift, so we also use a recurrent rollout. Starting from $\rho_0^{\mathrm{roll}} = \rho_0$, we recursively apply $\rho_{k+1}^{\mathrm{roll}} = \mathrm{Adv}_{\mathrm{RK2}}(\rho_k^{\mathrm{roll}}, \mathbf{u}_k) + s_k$ and define

$$\mathcal{L}_{\mathrm{roll}} = \sum_{k=1}^{K-1} \overline{\phi}(\rho_k^{\mathrm{roll}} - \rho_k), \tag{43}$$

$$\boldsymbol{\zeta}_k = \nabla \times \mathbf{u}_k, \qquad \boldsymbol{\zeta}_{k\to k+1}^{\mathrm{adv}} = \mathrm{Adv}_{\mathrm{SL}}(\boldsymbol{\zeta}_k, \mathbf{u}_k),$$
$$\mathcal{L}_{\mathrm{vor}} = \sum_k \left\| \boldsymbol{\zeta}_{k+1} - \boldsymbol{\zeta}_{k\to k+1}^{\mathrm{adv}} - \Delta t\, (\boldsymbol{\zeta}_{k\to k+1}^{\mathrm{adv}} \cdot \nabla)\mathbf{u}_k \right\|_1 . \tag{44}$$

Here $\overline{\phi}$ averages the Charbonnier penalty over grid points, and $\mathrm{Adv}_{\mathrm{SL}}$ is first-order semi-Lagrangian advection. The vorticity term is the discrete soft constraint from Eq. (19). Finally, with video-derived foreground weights $w_k = 1 + \lambda_f m_k$, the rendering loss is

$$\mathcal{L}_{\mathrm{rend}} = \sum_k \overline{\phi}_{w_k}(\widehat{I}_k - I_k) + \lambda_t \sum_k \overline{\phi}\Big((\widehat{I}_{k+1} - \widehat{I}_k) - (I_{k+1} - I_k) \Big), \tag{45}$$

Density smoothness and velocity regularization. The regularization group complements transport consistency with spatial smoothness and a preference for small velocity magnitudes. Let $\psi_\epsilon(r) = \sqrt{r^2 + \epsilon^2}$, with $\epsilon = 10^{-3}$. For a scalar density field on the grid Ω, let $\Omega_a = \{\boldsymbol{\xi} \in \Omega : \boldsymbol{\xi} + \mathbf{e}_a \in \Omega\}$, where $\mathbf{e}_a$ is one voxel step along axis $a \in \{x, y, z\}$. Define

$$\mathcal{S}(\rho) = \frac{1}{3} \sum_{a\in\{x,y,z\}} \frac{1}{|\Omega_a|} \sum_{\boldsymbol{\xi}\in\Omega_a} \psi_\epsilon(\rho(\boldsymbol{\xi} + \mathbf{e}_a) - \rho(\boldsymbol{\xi})) . \tag{46}$$

This robust penalty uses voxel-space forward differences, without division by physical grid spacing. Differences are evaluated only where the forward neighbor exists; each directional mean is normalized separately before the three directions are averaged. The constant Charbonnier offset is retained and does not affect gradients. For K density frames and $K-1$ transition velocities, the two terms are

$$\mathcal{L}_{\rho\text{-sm}} = \frac{1}{K} \sum_{k=0}^{K-1} \mathcal{S}(\rho_k), \tag{47}$$

$$\mathcal{L}_{u\text{-mag}} = \frac{1}{3(K-1)|\Omega|} \sum_{k=0}^{K-2} \sum_{\boldsymbol{\xi}\in\Omega} \sum_{c=1}^{3} |u_{k,c}(\boldsymbol{\xi})|, \tag{48}$$

$$\lambda_{\mathrm{reg}} \mathcal{L}_{\mathrm{reg}} = \lambda_{\rho\text{-sm}} \mathcal{L}_{\rho\text{-sm}} + \lambda_{u\text{-mag}} \mathcal{L}_{u\text{-mag}}. \tag{49}$$

The non-negative coefficients $\lambda_{\rho\text{-sm}}$ and $\lambda_{u\text{-mag}}$ are the effective weights in the total objective; Eq. (49) specifies the grouped term in Eq. (16) without an additional weight multiplication. The velocity penalty averages absolute values componentwise, rather than taking the Euclidean norm of each vector. It acts on the velocity field evaluated from the DFKs, not on independently predicted grid corrections. All velocity updates optimize the DFK parameters, so analytical incompressibility is preserved. No soft-divergence penalty is added. Camera and appearance calibration remain fixed during this stage and therefore contribute no calibration regularizer to this objective. The separate proximal term in Eq. (15) anchors density to the posterior reconstruction; spatial smoothness instead constrains local density variation.

A.6 Protocol notes and ablation

The three stages use different optimization scopes. The representation and conditional prior are learned from training scenes. Stage 2 samples each test video with the networks frozen and merges overlapping windows by center-weighted overlap-add. Stage 3 is then optimized for that sequence; it is not a one-pass predictor trained across test scenes.

Density acceptance and the final fixed-density motion fit follow Eqs. (17)–(18). The coefficient $\alpha^\star$ controls only the density update and does not rescale DFK velocity or source. After density is accepted, density-only terms are constant, while the motion terms fit $\Theta_{\mathbf{u}}$ and the constrained source to that density sequence. The resulting $(\rho^{\mathrm{final}}, \mathbf{u}^{\mathrm{final}}, s^{\mathrm{final}})$ is the reported output, with no later rollback or rescaling. The DFK remains pointwise divergence-free throughout this optimization.

Density units at the motion-optimization interface. Stage 2 and Stage 3 must use the same density units. Each dataset uses one fixed positive density normalization constant a_{ds}, shared by all scenes and frames in all splits. Representation learning uses $\bar{\rho}_k = \rho_k / a_{\mathrm{ds}}$. At test time, the decoder produces $\widehat{\bar{\rho}}_k$, and its density units are restored by the fixed multiplication $\widehat{\rho}_k = a_{\mathrm{ds}} \widehat{\bar{\rho}}_k$ before Stage 3. The constant is reused directly: no scene- or frame-specific scale is predicted, estimated from a test reconstruction, or taken from test GT. Because every volume in a dataset uses the same factor, normalization preserves relative amplitudes across scenes and frames; multiplying back restores the original density units. Both $\widehat{\rho}_k$ and ρ_k in the acceptance rule use these units. The accepted ρ_k^{final} retains them during the fixed-density velocity/source fit, and s_k is the density added over a frame interval in the same units. Neither physical optimization nor subsequent re-simulation independently renormalizes scenes or frames.

B Expanded experimental details

B.1 Evaluation protocol

We first define the inputs, data split, baselines, and metrics used in all quantitative comparisons. This keeps the evaluated scenes, frames, and cameras consistent across methods.

Image-only conditioning and inference. All reported ScalarFlow and Plume experiments use the given monocular video and its calibrated camera at inference. Every spatial map and global descriptor at inference is computed from the corresponding input-view image through the image-only entry points specified in Appendix A.3. Training-volume projections used to train the conditional prior are distinguished there from test-time observations. No test-time conditioning input is obtained from a reference density or an additional view of the target scene. Held-out images and reference fields are read only for scoring, not for conditioning, guidance, calibration, refinement, or acceptance.

Following FluidNexus (Gao et al., 2025), we separately evaluate density reconstruction and novel views, divergence-free velocity recovery, and density re-simulation. This distinction matters because image fit does not guarantee 3D geometry, and low divergence does not guarantee accurate rendered motion. Reconstruction and re-simulation use PSNR (dB), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018); velocity uses cosine similarity and finite-difference divergence RMS.

ScalarFlow (Eckert et al., 2019) contains 104 buoyant-smoke captures from five calibrated cameras. We use the inclusive frame range 50–149 (100 frames per scene) for reconstruction and its image metrics. All ScalarFlow frame indices in the text are zero-based; existing figure headers use one-based labels, with displayed frame $f + 1$ corresponding to text frame f. Calib3 is the input and calib1/2/4/5 are strictly held out. The synthetic Plume experiment uses one given and six test cameras. Input-camera positions and viewing directions vary across scenes in the training and test data, while all frames of a given scene share one fixed calibrated input camera. Each compared method receives the same input-camera video for that scene. Metrics are first aggregated over frames and cameras or fields within a scene, then averaged over all five Plume test scenes or the fixed ScalarFlow subset `sim005--009` of the ten held-out scenes (`sim000--009`), following the same 94/10 training/held-out split as FluidNexus. Scene identities and frames are fixed across methods, and no held-out camera is used for model selection.

The baselines cover monocular generative–physical reconstruction (FluidNexus and SmokeSVD (Li et al., 2026)), multi-view physics-informed reconstruction (PINF (Chu et al., 2022) and HyFluid (Yu et al., 2023)), monocular transport reconstruction (NGT (Franz et al., 2023) and Global-trans (Franz et al., 2021)), and non-physical dynamic rendering (STG (Li et al., 2024)). PINF and HyFluid use their complete official reconstruction pipelines with one real input view each. They estimate physical velocity as well as density; the comparison does not give them the multiple real views used in their original sparse-view settings.

NGT and Global-trans reproduction protocol. NGT and Global-trans are reproduced using their official implementations under the scene, frame, and camera selections defined above. Each method receives one real input view. The protocol retains their reconstruction implementations, with training data following the stated scene split. NGT uses its native velocity output. For our Global-trans reproduction, the reconstructed densities instead initialize Stage 3, which jointly refines density,

Table 4: ScalarFlow reconstruction on test views only, averaged over five scenes, frames 50–149, and `calib1/2/4/5`. P_τ: normal-render PSNR; P_α: integrated-opacity PSNR (both in dB). SSIM and LPIPS evaluate normal renders. Corresponding P_τ/SSIM/LPIPS entries match the ScalarFlow Test synthesis block of Table 2. FluidNexus and SmokeSVD use native renderers; Global-trans and Cloudy include Stage 3 (Appendix B.1).

Method	$P_\tau \uparrow$	$P_\alpha \uparrow$	SSIM↑	LPIPS↓
Ours	**32.99**	**31.95**	**.956**	**.081**
FluidNexus	29.63	25.08	.882	.151
SmokeSVD	26.88	22.62	.806	.181
NGT	30.05	20.57	.786	.143
Global-trans + Stage 3	31.26	19.57	.800	.107
Cloudy + Stage 3	26.31	25.73	.759	.208

source, and DFK velocity; all reported Global-trans results include this refinement. Image results use the common metric definitions, and velocities are converted to the common coordinates and units before scoring. ScalarFlow and Plume recurrent transport follow Appendix B.5. Global-trans is therefore not repeated in the separate shared-Stage 3 comparison. Exact reproduction configurations and checkpoint provenance accompany the measurements; no scores are copied from the original papers.

Rendered-image and integrated-opacity metrics. Table 4 reports only held-out test views (`calib1/2/4/5`); the given camera `calib3` is excluded. P_τ denotes PSNR of the normally rendered image $\widehat{I}$ against its reference image I, rather than PSNR of optical thickness. P_α denotes PSNR of the ray-integrated opacity image $\widehat{A}(p) = \sum_i T_i \alpha_i = 1 - \exp[-\kappa \sum_i \rho(\boldsymbol{\xi}_i)\delta_i]$ against its reference opacity image A. For either image pair, PSNR is $10 \log_{10}(1/\mathrm{MSE})$ on the $[0, 1]$ image range. SSIM and LPIPS use the normally rendered images, not the opacity images. Scores average over the held-out cameras and frames within each scene, then equally over the five scenes. Thus P_τ, SSIM, and LPIPS use the same reconstruction targets as the ScalarFlow Test synthesis entries in Table 2; P_α is a separate opacity measurement.

Cloudy-style density-prior comparison. The Cloudy + Stage 3 rows in Table 4 report a controlled adaptation of the static volumetric inverse-rendering approach of Cloudy (Leonard et al., 2025) to our smoke data. The comparison protocol uses an unconditional density denoiser trained on the same training-scene volumes, with the same frozen monoplanar decoder and latent resolution as ours. Each observed frame is inverted independently using rendering-guided diffusion, without image-feature conditioning or temporal-window synchronization. The resulting density sequence initializes the same Stage 3 density–source–DFK refinement, with matched physical losses and optimization budget. This controls the downstream optimizer when comparing the frame-wise unconditional prior with our conditioned, temporally synchronized density inference; it is an adapted baseline rather than the unmodified Cloudy pipeline.

Evaluation uses the same five ScalarFlow test scenes, all 100 frames (50–149), input camera `calib3`, and held-out cameras `calib1/2/4/5`. The adapted baseline shares our renderer, fixed extinction/exposure, density normalization, and metric definitions. Scores are averaged within each scene and then over the five scenes. Compared with this unconditional, frame-wise Cloudy-style adaptation, our conditioned and temporally coordinated model improves test-view P_τ from 26.31 to 32.99 dB, SSIM from .759 to .956, and LPIPS from .208 to .081. This comparison measures the combined design; the w/o conditioning ablation in Table 3 isolates image-feature conditioning while retaining temporal coordination and the physical refiner.

Official baseline pipelines and native rendering. Both FluidNexus and SmokeSVD start from pretrained weights released by their authors. On Plume, we fine-tune these weights using only the 37 training scenes. On ScalarFlow, we directly use the official pretrained weights without additional fine-tuning. This dataset-specific protocol applies to both baselines throughout the reported comparisons. Fine-tuning and inference follow the authors' released implementations and model configurations (Gao et al., 2025; Li et al., 2026), including their prescribed module order and

reconstruction objectives. FluidNexus uses its native particle renderer, and SmokeSVD uses the rendering implementation supplied with its pipeline. For both reproductions, camera and rendering alignment maps their native coordinate and projection conventions to the camera setup used by our renderer, so that predictions correspond to the same viewpoints and image framing as the evaluation targets. On ScalarFlow, this alignment uses `calib3` as the input camera and `calib1/2/4/5` as the held-out evaluation cameras. Their aligned rendered predictions are scored directly; they are not exported to our density renderer for image evaluation. For common-grid velocity evaluation, exported vectors are expressed in the shared (v_x, v_y, v_z) axis order and converted to cell displacement per density-frame interval on the grid in Appendix B.5.

All methods are evaluated on the same target images, calibrated cameras, frame indices, and scene subsets, with the same metric definitions and aggregation. The official baseline pipelines retain their native rendering models and prescribed scale and appearance processing; the controlled Cloudy + Stage 3 adaptation instead shares our rendering protocol as specified above. Held-out images are used only for scoring. Our own renderer uses calibrated rays and fixed evaluation extinction/exposure across frames and views. After decoding and before Stage 3, density units are restored by multiplying by a_{ds}, the same dataset constant used for every training and test scene. These units are retained through density acceptance, the final velocity/source fit, and re-simulation, preserving relative amplitudes without scene- or frame-specific renormalization.

Scene-disjoint training and evaluation. ScalarFlow follows the FluidNexus scene split: sim010–103 for training and sim000–009 held out (94/10). Quantitative comparisons use `sim005--009`, matching FluidNexus's five-scene selection; this subset is fixed for all methods and ablations. Plume uses 37/5/5 complete scenes for train/validation/test. Aggregate evaluation metrics, including velocity diagnostics and all ablations, are averaged over five test scenes; the per-scene appendix table decomposes this same evaluation set. Reconstruction, velocity, and re-simulation visualization scenes are randomly selected from these test sets. These scenes are excluded from our model training and from baseline fine-tuning on Plume; the ScalarFlow baselines use the released weights as specified above. The five ScalarFlow comparison scenes are excluded from both our training data and the pretraining data of the official FluidNexus checkpoint used here. Autoencoder and diffusion-prior fitting and estimation of latent normalization statistics use only training scenes. No images, volumes, or latents from test scenes are included in this training. Density and auxiliary rendering supervision, where used, also come only from training scenes. Sequence-specific reconstruction on a test scene optimizes against its given-view video with the learned prior fixed; it does not use held-out views or reference fields. This scene separation and the image-only conditioning protocol apply to all configurations described below. The density scale a_{ds} is fixed for each dataset and reused unchanged across all splits; inference does not estimate a new scale.

B.2 Reproducibility configuration

This subsection records the implementation choices needed to reproduce the three stages. These settings do not introduce additional inputs beyond the monocular video and its calibrated camera.

Recorded configuration and release. Training and inference settings are retained in experiment configurations and run logs, including optimizer settings, batch sizes, random seeds, sampling and reweighting constants, gradient normalization, and compute records. These records will be released with the code and trained weights. The Plume release will include density and velocity fields, the complete generation and preprocessing scripts, simulator version and parameters, initial and boundary conditions, source settings, random seeds, camera calibrations, and exact train/validation/test manifests. Together, these artifacts specify how to regenerate and evaluate the same 47-scene dataset.

Common experimental pipeline. ScalarFlow and Plume follow the three-stage formulation in Sections 3.1–3.3: learning an image-conditioned monoplanar density prior, render-guided sampling with temporal-window synchronization, and joint density–source–DFK refinement. The dataset-specific image descriptors are constructed from the given view, while the representation, inference stages, and physical objective follow the same definitions below.

Networks. The monoplanar encoder is Conv3d $1 \to 24 \to 24 \to 1$; its six-layer width-256 MLP decodes density using Eq. (2). The latent shape is $100 \times 128 \times 128$, with dataset-fixed channel-wise

training statistics as in Eq. (26). Density normalization separately uses the scalar a_{ds}, fixed across all scenes and frames of the dataset. The 100-channel latent UNet has widths 128/256/512/512, two time-conditioned ResBlocks in each down/up stage, linear/full/full/linear attention with four 32-D heads, and a ResNet–attention–ResNet bottleneck. ScalarFlow uses the five-channel spatial map and 43-dimensional global descriptor of Section 3.1; Plume uses a 13-channel map and 86-dimensional descriptor. Appendix A.3 enumerates all channels and vector entries, including the duplicated image-only Plume branch. Both use Eq. (29); neither descriptor encodes camera pose. The spatio-temporal refiner predicts the density residual, source logit, and density confidence in Eq. (10); velocity is represented by independent DFK parameters for each transition. The Plume refiner has hidden/local widths 96/32, three layers, six heads, and 2,453,915 parameters.

Optimization. Training first fits the autoencoder, then freezes its encoder and decoder while fitting the conditional denoiser. At inference, these networks remain fixed: Stage 2 updates the noisy density latents, and Stage 3 optimizes the sequence-specific refiner and DFK parameters. The monoplanar AE used 80,000 updates with 32,768 queries/update (40% density-biased), $5\times10^{-4}\to5\times10^{-5}$ learning rate, and latent-ℓ_2/TV weights $10^{-8}/10^{-4}$. Its exact optimizer, volume batch size, seed, and sampler/reweighting constants are retained in the experiment configuration. The Plume prior uses AdamW $(\beta_1,\beta_2)=(.9,.999)$, weight decay .01, clip 2, AMP/BF16, and batch $2/\mathrm{GPU}\times8=16$: a 232,000-update base run decays $10^{-5}\to10^{-6}$, and a 995,744-update resumed phase applies another $0.1\times$ decay (1,227,744 updates total). Its seed is $1234+\mathrm{rank}$ and it has no parameter EMA.

Training objectives. Representation learning uses the density reconstruction, latent-ℓ_2, and spatial-TV terms of Eq. (25). Conditional diffusion uses the v-parameterized objective in Eq. (4): v prediction, clean-latent reconstruction, and the X–Z gradient penalty have weights $(1,1,.05)$. Temporal synchronization is applied during Stage 2 inference through Eq. (6) and Eqs. (7)–(8); physical refinement is the subsequent sequence-specific optimization in Eq. (16).

Posterior controls and weights. The following values specify the reported Stage 2 run; they are listed for reproduction rather than as additional method components. Both datasets use the frozen conditional denoiser with rendering-guided latent updates in Eq. (9). Workers exchange detached low-resolution densities, use the 27-candidate translation alignment, and combine overlapping windows by center-weighted overlap-add. The synchronized density sequence initializes Stage 3. The reported Plume posterior uses extension checkpoint epoch 4292, 100 deterministic DDIM transitions ($\eta=0$), two gradient passes per transition, length-16/stride-8 windows, and image weight .75 only for $150\le t_{\mathrm{diff}}<750$, modulated by $q(t)=\sqrt{[1-\cos(2\pi(t/T)^2)]/2}$. Image gradients stop at 35.0 dB and resume below 34.5 dB. The motion, bottom-source, and component weights are $5000r(p)$, 600, and $300n(p)$; source compact/center/unimodal weights are $(1,.75,0)$. Here $r(p)=0$ through .20, rises as $((p-.20)/.65)^2$ until .85, then equals 1; $n(p)=0$ through .35 and equals $((p-.35)/.65)^2$ thereafter. Source target variance/Gaussian width/bottom fraction are $.10/.22/.12$; component relative/absolute thresholds are $.03/5\times10^{-4}$ with two-voxel dilation. Latent, latent-gradient, decoded-log-volume, and volume-gradient anchors have weights $(80,40,800,250)$, using a zero/quadratic/one ramp over progress $[0,.20]/(.20,.75)/[.75,1]$; window consensus is 120 and the additional temporal-latent term is disabled. No reference-volume coherence term or ground-truth volume is used during inference. Temporal/source/noise and prior-anchor extras are capped at $.25\times$ and $2\times$ the image loss. The physical grid is $32\times64\times32$; early stopping and hard x_0 clamping are off, and the fixed start-noise artifact is `fixed_start_noise_xT_528.pt`. The final gradient-normalization denominator is retained in the sampling records and will be included in the release. Sampling x_0 EMA is disabled ($\alpha=0$, hence $x_{0,\mathrm{ema}}\leftarrow\widehat{x}_0$); the unused rule was $\alpha_i=.6^{\Delta t/4}((T-t_i)/T)^2$ and $x_{0,\mathrm{ema}}\leftarrow\alpha_i x_{0,\mathrm{ema}}+(1-\alpha_i)\widehat{x}_0$. The optional image-only refinement after posterior sampling is disabled (0 iterations). This setting applies only to additional posterior image refinement: Stage 3 physical refinement and density acceptance follow the implementation-specific execution protocol described below. Preceding radiance calibration uses 1,000 iterations and 2,048 rays/iteration.

4D refinement and density acceptance. We next give the Stage 3 settings and connect them to the losses defined in Appendix A.5. The refinement objective is exactly Eq. (16). The image term follows Eq. (45); forward transport, backward cycle, and mass consistency follow Eq. (42); recurrent transport and vorticity evolution follow Eq. (44). In particular, $\lambda_{\mathrm{vor}}\mathcal{L}_{\mathrm{vor}}$ penalizes the

Table 5: Plume inference and refinement runtime and hardware usage. Single-frame posterior times are average worker latencies; sequence-level times cover a 100-frame scene. Model training is excluded.

Stage	Wall time	Hardware / memory
Single-frame posterior, one worker	4 min 56 s	1×H20; 52.9 GiB
Single-frame posterior, concurrent	11 min 04 s	2 workers per GPU
100-frame posterior	2 h 26 min	8×H20
Temporal post-optimization, 300 epochs	16.5 min	Rank 0: 7.4 GB; others: 5.2 GB each
Calibration, refinement, velocity fit	3–6 min	Mostly single-GPU stages

discrepancy between successive DFK vorticity fields after advection and vortex stretching. It does not include viscous diffusion or external-force curl, as discussed in Appendix A.5. The proximity term is Eq. (15), including the source-ℓ_1 penalty, and $\mathcal{L}_{\mathrm{reg}}$ contains density spatial smoothness and the velocity-magnitude regularizer defined in Appendix A.5. Each transition uses 10^5 compact DFKs with jointly optimized centers, isotropic radii, and weights, initialized by Eq. (37). Both RK2 backtrace queries evaluate this continuous field directly. Incompressibility is enforced by the representation, so the objective has no soft-divergence penalty or pressure-projection step. Confidence modulates only the bounded density-logit residual; the source remains non-negative and restricted to Ω_{src}, as in Eq. (10). Camera and rendering parameters remain fixed during this refinement.

The Plume refiner uses 95 length-6/stride-1 windows, batch $1/\mathrm{GPU} \times 8$, AdamW at 2×10^{-4} with weight decay 10^{-4}, cosine decay to .05 of the initial rate, clip 1, AMP, 3,600 joint updates, and seed $20260713 + \mathrm{rank}$.

Density acceptance. The rendering guard in Eq. (17) uses $\delta_{\mathrm{rend}} = 0.02$ and $\varepsilon_{\mathrm{rend}} = 10^{-10}$. We evaluate $\alpha \in \{1, 0.75, 0.5, 0.25, 0\}$ in descending order and accept the first admissible density. The given-view rendering error is

$$E_{\mathrm{rend}}(\rho) = \frac{\sum_{p\in\mathcal{P}_4}\sum_{c=1}^{3} w_p\big(\mathcal{R}(\rho)_{p,c} - I_{p,c}\big)^2}{3\sum_{p\in\mathcal{P}_4} w_p}, \tag{50}$$

where $\mathcal{P}_4$ samples every fourth image row and column, $w_p = 1 + 4m_p$, $m_p = \mathrm{sigmoid}((d_p - 0.025)/0.01)$, and $d_p = \frac{1}{3}\sum_c |I_{p,c} - b_c|$ for background color $\mathbf{b}$. Initial and candidate densities share the same fixed rendering and camera parameters. After acceptance, the density is held fixed while velocity and source follow Eq. (18). The velocity continues to be represented by DFKs throughout this fit. All guidance, refinement, and acceptance decisions use only the given-view video and its calibrated camera, as specified in Section 3.3.

Compute. We report training cost separately from per-scene inference and state which stages are included in each timing. Documented diffusion, posterior, and 4D runs used one node with 8×NVIDIA H20-3e GPUs (143,771 MiB, approximately 140.4 GiB, per card). Monoplanar-AE hardware, peak allocated memory, and wall time are retained in the run logs. Plume-prior training took 22.8 h for the 1,227,744-update schedule above; this training cost is separate from per-scene reconstruction. Table 5 summarizes the measured averages and approximate stage costs.

The single-frame timings are worker latencies, whereas 2 h 26 min is the elapsed posterior time for the complete 100-frame scene. Production uses two workers per GPU, 16 concurrent frame workers per window, 12 overlapping windows with stride 8, and 100 DDIM/DPS updates per frame. GPU sharing increases individual-worker latency, while concurrency processes multiple frames simultaneously; overlapping windows also revisit frames. The scene time therefore cannot be obtained by multiplying either single-frame latency by 100.

Summing the stage costs gives an estimated 165.5–168.5 min for reconstruction through the final fixed-density velocity/source fit. This is a sum of the measured/estimated stage timings, not a separate end-to-end timing measurement. It excludes model training.

B.3 EXPANDED TASK RESULTS

We report reconstruction first, then velocity and transport. All numbers below use the evaluation protocol defined above; displayed frames are qualitative examples only.

Figure 4(a) compares input and held-out views for a randomly selected test scene from each dataset; quantitative scores average all five test scenes per dataset.

Table 4 aggregates all 100 frames (official frames 50–149) from each of five ScalarFlow test scenes, using only the four held-out cameras. Our test-view normal-render PSNR is 32.99 dB and integrated-opacity PSNR is 31.95 dB. Test-view SSIM is .956, compared with .882 for FluidNexus and .806 for SmokeSVD; the corresponding LPIPS values are .081, .151, and .181, respectively. On Plume, the same held-out cameras span a full 360° orbit for all methods, with each method retaining its native renderer. Five-scene averages test whether the source and rising crown remain consistent away from the input view. Our reproductions use matched scenes and cameras. Appendix B.4 distinguishes recorded-camera evaluation from full-orbit reference renders and identifies the FluidNexus scores quoted under the original paper's evaluation protocol.

All Plume quantitative comparisons use the complete observed interval 50–149 (100 density frames) and its 99 adjacent-frame transitions. Cosine, divergence, and both velocity MSE metrics share the same interval fields, indexed by source frames 50–148. No early, middle, or late subinterval is selected for scoring. Each metric averages within a scene before averaging equally over the five test scenes. The few frames shown in figures are qualitative examples, not the quantitative evaluation subset. Appendix B.5 gives the common frame manifest. Cosine improves to .496 from $-.008$/.061. The small reported divergence is caused by raster sampling and finite differencing of an analytically divergence-free field. Because ScalarFlow lacks velocity GT, we separately compare with a five-camera reconstruction proxy over the same five test scenes: mean full-grid cosine is .286, reference-active cosine .332, and divergence RMS 5.92×10^{-4}. We do not treat this proxy as physical GT.

B.4 RECORDED-CAMERA AND FULL-ORBIT REFERENCE-RENDER EVALUATION

ScalarFlow is evaluated using recorded-camera images and full-orbit reference-volume renders, as specified below.

Recorded cameras: one input and four tests. FluidNexus (Gao et al., 2025) uses the middle camera as input and the other four cameras as held-out references. Our corresponding ScalarFlow setting uses $\mathcal{C}_{\text{in}} = \{\texttt{calib3}\}$ and $\mathcal{C}_{\text{test}} = \{\texttt{calib1}, \texttt{calib2}, \texttt{calib4}, \texttt{calib5}\}$, with all four test cameras retained. This is not a smaller camera subset selected within $\pm60^\circ$ of the input. References are the recorded images at the calibrated test cameras. The main ScalarFlow evaluation in Tables 2 and 4 uses these official recordings, `sim005--009`, and frames 50–149, averaging over frames and test cameras within a scene and then equally over scenes.

Why we do not label ScalarFlow 120°. The approximately 120° arc in FluidNexus describes its own FluidNexus-Smoke/FluidNexus-Ball capture setup, not a measured ScalarFlow camera span. ScalarFlow's official capture description instead specifies approximately 90° between its two outermost camera positions[1]. This is an approximate positional arc around the smoke, rather than the field of view of one camera or an exact angle between optical axes. We label the recorded-camera experiment by its actual camera setting, without assuming equally spaced cameras or $\pm60^\circ$ endpoints. Exact camera azimuths depend on the calibration.

Full-orbit reference-volume renders. The 360° comparison in Table 6 uses ScalarFlow multiview-reconstructed density volumes as the reference. These reference volumes are rendered at virtual test cameras distributed around the smoke; the evaluated reconstruction is rendered at the same poses and with the same evaluation rendering settings. This is a separate five-view setting with one input and four test viewpoints. Only the input video is supplied to the monocular reconstruction; reference

[1] ScalarFlow recording setup: `https://ge.in.tum.de/publications/2019-scalarflow-eckert/`, "Recording—Setup."

Table 6: ScalarFlow evaluation with two reference types, each using one input and four test views. Top: the FluidNexus recorded-camera setting; † denotes its published result under the original protocol. Bottom: full-orbit virtual views scored against renders of a multiview reference reconstruction.

Method	Reference / camera setting	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	Recorded / FluidNexus cameras	33.80	.9724	.0789
FluidNexus†	Recorded / published protocol	32.45	.9544	.1299
Ours	Reference renders / 360°	32.99	.956	.081
FluidNexus	Reference renders / 360°	29.63	.882	.151

volumes and test images are used for scoring. Here 360° denotes discrete viewpoints placed around a full orbit.

The recorded-camera setting uses real images, while the full-orbit setting uses renders of multiview-reconstructed reference densities. The published FluidNexus row retains the original paper's evaluation protocol.

B.5 Velocity metrics and transport protocols

To compare motion fairly, every method is converted to the same grid, velocity units, frame interval, and aggregation rule.

All velocity columns in Table 1, the shared-Stage 3 comparison, and the ablations use the same full-sequence manifest:

$$\mathcal{F} = \{50, 51, \ldots, 149\}, \qquad \mathcal{T} = \{50, 51, \ldots, 148\}, \qquad k \in \mathcal{T}: \ \mathbf{u}_k \text{ corresponds to } k \to k+1.$$

Thus each scene contributes 100 density frames and 99 interval velocity fields, from $50 \to 51$ through $148 \to 149$; the five Plume test scenes contain 495 evaluated intervals in total. Cosine, divergence RMS, full-grid MSE, and smoke-region MSE all use $\mathcal{T}$, with their respective spatial masks defined below. The last density frame 149 is a transport target, not an additional interval velocity field. Metrics average over intervals within each scene and then equally over the five scenes. Selected visualization frames do not restrict this manifest. Every predicted field is converted to $[X, Y, Z, v_{xyz}]$ on the same $40 \times 80 \times 40$ grid, with velocity measured as cell displacement per density-frame interval. We distinguish full-grid $\mathrm{MSE}_{\mathrm{all}}$ from smoke-region $\mathrm{MSE}_{\mathrm{smoke}}$. For our method and the shared-Stage 3 variants, evaluation uses $\mathbf{u}_k^{\mathrm{final}}$ from the fixed-density motion fit in Eq. (18), after density acceptance in Eq. (17). MSE and divergence retain its absolute velocity magnitude in the stated units; cosine measures directional agreement.

Smoke-region velocity MSE. Following the smoke mask used by LagrangianSplats (Tao et al., 2026), we define $A_k = \{\mathbf{x} : \rho_k^{\mathrm{GT}}(\mathbf{x}) > 0\}$ on the common evaluation grid, using the source density frame of each transition. For each $k \in \mathcal{T}$, the component-averaged error is

$$E_k^{\mathrm{smoke}} = \frac{1}{3|A_k|} \sum_{\mathbf{x} \in A_k} \|\widehat{\mathbf{u}}_k(\mathbf{x}) - \mathbf{u}_k^{\mathrm{GT}}(\mathbf{x})\|_2^2. \tag{51}$$

The same GT mask is used for every method, without density weighting or excluding near-zero predicted velocities. We average transition errors within each scene and then average the five scene scores equally; an empty GT mask is omitted from the transition mean. $\mathrm{MSE}_{\mathrm{all}}$ uses the same formula and aggregation with A_k replaced by the entire grid. Both errors use squared cell-displacement units. GT density is used only for evaluation, not reconstruction or motion fitting.

Velocity cosine. Following the LagrangianSplats implementation (Tao et al., 2026), for each Plume interval velocity field $k \in \mathcal{T}$ we retain grid points $V_k = \{\mathbf{x} : \rho_k^{\mathrm{GT}}(\mathbf{x}) > 0, \ \|\widehat{\mathbf{u}}_k(\mathbf{x})\|_2 > 10^{-10}, \ \|\mathbf{u}_k^{\mathrm{GT}}(\mathbf{x})\|_2 > 10^{-10}\}$ and compute

$$C_k = \frac{1}{|V_k|} \sum_{\mathbf{x} \in V_k} \frac{\widehat{\mathbf{u}}_k(\mathbf{x}) \cdot \mathbf{u}_k^{\mathrm{GT}}(\mathbf{x})}{\|\widehat{\mathbf{u}}_k(\mathbf{x})\|_2 \|\mathbf{u}_k^{\mathrm{GT}}(\mathbf{x})\|_2}. \tag{52}$$

Points are equally weighted; the 10^{-10} norm thresholds avoid undefined cosines, and near-zero predictions are excluded rather than assigned zero cosine. In the reported evaluations, predicted

zero velocities are rare, and every evaluated method has a nonempty V_k for every interval in $\mathcal{T}$ in all five scenes: no interval is skipped. Thus each scene's cosine is the mean over all 99 interval scores, followed by an equal-weight mean over the five scenes (495 intervals in total). All methods share this temporal evaluation set. The spatial mask retains its prediction-dependent threshold; rarity of zero predictions does not imply identical point sets. Unlike cosine, neither MSE metric excludes near-zero predictions; the smoke-region MSE uses the same GT-density mask for every method.

Shared Stage 3 velocity comparison. For the +Stage 3 rows in Table 1, each reconstructed density sequence is expressed in the common calibrated coordinates, grid, and density units before serving as $\widetilde{\rho}_k$ in Eq. (10) and in the density-proximity term. FluidNexus and SmokeSVD supply densities only; their reconstruction models remain frozen, and no velocity or source estimates are transferred. All three variants use the same DFK initialization rule (Eq. (37)), source support, observed images, renderer, loss weights, optimization budget, density-acceptance rule, and subsequent fixed-density velocity/source fit. Initial joint refinement is followed by density acceptance and the final motion fit in Eq. (18). The scene, frame, velocity-unit, and evaluation-grid conventions match Table 1 and the common-grid MSE protocol above. Ground-truth fields are used only for evaluation. The table compares velocity recovered by this common refiner; FluidNexus and SmokeSVD native velocity results remain separately reported in Table 1.

Observed-interval transport protocols. For re-simulation, every advected density is rendered from the designated evaluation cameras and compared with its frame-matched image. The two task blocks in Table 2 distinguish renders of reconstructed densities from renders after transport. Within each task, PSNR, SSIM, and LPIPS are aggregated separately for the given and held-out views, first over frames and the corresponding cameras within each scene, then equally over the five test scenes. Re-simulation therefore measures visible transport rather than density cosine or a mass ratio. Both datasets initialize once from the first reconstructed density of the evaluated interval and recursively use $\rho_{k+1}^{\mathrm{roll}} = \mathrm{Adv}(\rho_k^{\mathrm{roll}}, \mathbf{u}_k) + s_k$, without replacing intermediate states by reconstructed densities. The velocity and source sequences are fitted using the observed video, so this tests observed-interval transport. For both datasets, the observed interval is frames 50–149, and transport is evaluated on all 99 transitions from 50 to 148, targeting frames 51–149. For the STG and Global-trans reproductions, reconstructed densities initialize Stage 3 in the common coordinates and density units. The resulting refined densities, DFK velocities, and sources drive the same recurrent transport protocol described above. Their reported scores include this Stage 3 optimization.

Held-out renderings evaluate volumetric appearance, while the transport results test how the recovered states explain motion within the observed interval.

B.6 Core-claim and physical-refinement ablations

The first ablations test the three main inference components; the second group removes one Stage 3 loss at a time. All other settings are held fixed within each comparison.

Core-claim protocol. The variants in Table 3 share the same five Plume test scenes, cameras, renderer, and 100 reconstruction frames per scene. Every ablation score is a five-scene average. The no-diffusion variant retains the same frozen decoder and latent representation, replacing diffusion sampling with direct latent optimization against the given-view rendering loss. The decoded density sequence then undergoes the same Stage 3 density–source–DFK refinement as the full pipeline. Thus this comparison retains the learned density representation and physical refiner while removing diffusion-based inference.

Conditional MLP. The deterministic baseline predicts the normalized density latent from the same image-derived conditions, without an additional image encoder. Its 11-block MLP-Mixer embeds 8×8 patches of the 13-channel map into 256 width-1024 tokens, with token/channel MLP hidden widths 512/4096, GELU, LayerNorm, residual connections, and per-block FiLM from the 86-D global descriptor. A shared linear head reconstructs the $100\times128\times128$ latent. Shared patch projections control parameter growth, token mixing captures global spatial dependencies, and layerwise FiLM maintains conditioning throughout the network. Training uses latent MSE plus a 0.05-weighted X–Z gradient ℓ_1 loss, aligning clean-latent supervision with diffusion. The training split, frozen

Table 7: Leave-one-group-out ablation of physical refinement, averaged over five Plume test scenes. Reconstruction metrics use the given view; Full reconstruction metrics match the Plume Given entries in Table 2. Velocity cosine and re-simulation use the same five-scene protocols as the main comparisons. Full re-simulation PSNR values match the Plume entries in Table 2.

	Reconstruction (Given)			Velocity	Re-sim. PSNR ↑	
Method	PSNR ↑	LPIPS ↓	SSIM ↑	Cos. ↑	Given	Test
Full pipeline	**38.00**	.0740	**.973**	.496	37.09	34.92
w/o $\mathcal{L}_{\mathrm{rend}}$	35.31	.0719	.930	.488	32.91	31.88
w/o $\mathcal{L}_{\mathrm{tr}}$	35.58	.0708	.934	.010	30.88	28.35
w/o $\mathcal{L}_{\mathrm{cyc}}$	35.92	.0695	.939	-.089	31.94	30.99
w/o $\mathcal{L}_{\mathrm{roll}}$	35.76	.0701	.936	.201	32.05	31.33
w/o $\mathcal{L}_{\mathrm{mass}}$	36.31	.0669	.951	.479	36.84	34.08
w/o $\mathcal{L}_{\mathrm{vor}}$	36.08	.0682	.945	.450	36.55	34.45
w/o $\mathcal{L}_{\mathrm{prox}}$	35.47	.0713	.932	.479	30.99	28.37
w/o $\mathcal{L}_{\mathrm{reg}}$	36.42	**.0661**	.954	.401	35.80	33.71

encoder/decoder, condition extraction, dataset-fixed density and latent normalization, rendering guidance, temporal coordination, and Stage 3 remain unchanged.

Other interventions. Render-guidance removal sets the posterior image-gradient term to zero while retaining the prior and conditioning. Temporal-window removal disables both the motion-compensated constraint and overlapping-window consensus. The separate motion-compensation ablation disables $\mathcal{L}_{\mathrm{mc}}$ while retaining overlapping-window consensus, center-weighted overlap-add, and the same Stage 3 refinement. Its numerical results are reported in Table 3. The w/o conditioning variant removes both c_{map} and c_{vec} through matched retraining of an unconditional denoiser, using the same training volumes and frozen decoder. Rendering guidance, temporal coordination, and Stage 3 are retained; only the generative conditioning is removed. Masking features only at test time would introduce a distribution shift and is not the protocol used here.

The source ablation enforces $\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0$, preventing the model from explaining continuous bottom inflow with an explicit s_ρ. It therefore targets mass/source balance and long-horizon rollout rather than only input-view appearance. Finally, the grid-velocity variant replaces DFK with a dense velocity grid regularized by $\|\nabla \cdot \mathbf{v}\|_2^2$. Table 3 reports its reconstruction and re-simulation PSNR alongside velocity cosine and divergence to compare this soft regularization with analytical incompressibility.

The velocity columns follow Table 1, using all 99 interval fields per scene, indexed by source frames 50–148. Re-simulation uses recurrent density transport evaluated from the given camera and full-360° held-out orbit, as defined in Appendix B.5. Full reconstruction PSNR agrees with the Plume synthesis block of Table 2, and Full cosine reuses Table 1. Re-simulation requires separate measurements; the synthesis scores are not reused as transport scores. The reported non-Full entries measure the components' effects under these protocols.

Physical-refinement loss groups. Each leave-one-group-out variant in Table 7 disables one loss group while keeping all remaining loss weights, initialization, cameras, optimizer, iterations, and evaluation data fixed. The tested groups are image rendering, source-aware forward transport, backward cycle, recurrent rollout, mass balance, vorticity evolution, posterior/initial-state proximity, and the density-smoothness and velocity-magnitude regularizers in Appendix A.5. The w/o $\mathcal{L}_{\mathrm{vor}}$ variant sets $\lambda_{\mathrm{vor}} = 0$ throughout Stage 3, including the final fixed-density velocity/source fit, while retaining the DFK representation and all other losses. Thus it tests the contribution of the approximate vorticity regularizer without removing analytical incompressibility. It is distinct from w/o $\mathcal{L}_{\mathrm{reg}}$, which removes density smoothness and velocity-magnitude regularization.

Removing rendering or proximal anchoring produces the largest given-view PSNR and SSIM degradation: PSNR falls by 2.69/2.53 dB and SSIM by .043/.041, respectively. LPIPS instead decreases from .0740 to .0719/.0713; these ablations therefore do not degrade all reconstruction metrics uniformly. Removing forward transport, recurrent rollout, or backward cycle reduces PSNR by 2.42/2.24/2.08 dB; removing mass balance, vorticity evolution, or the remaining regularizers reduces it by 1.69/1.92/1.58 dB.

Table 8: Full-pipeline results over five Plume test scenes. Scene rows use mean ± standard deviation across sampling seeds; N is the number of seeds per scene. Across scenes reports aggregate metrics from Tables 2 and 1.

Scene	N	Recon. PSNR$_{\text{test}}$ ↑	Cos.↑	MSE$_{\text{all}}$ ↓	Re-sim. PSNR$_{\text{test}}$ ↑
Scene 1	5	35.10 ± 0.18	$.491 \pm .006$	$.000580 \pm .000031$	34.90 ± 0.16
Scene 2	5	34.30 ± 0.21	$.532 \pm .005$	$.000700 \pm .000046$	35.30 ± 0.14
Scene 3	5	34.00 ± 0.16	$.486 \pm .007$	$.000550 \pm .000028$	34.80 ± 0.18
Scene 4	5	33.80 ± 0.20	$.494 \pm .004$	$.000630 \pm .000037$	35.00 ± 0.15
Scene 5	5	33.30 ± 0.24	$.477 \pm .006$	$.000100 \pm .000012$	34.60 ± 0.20
Across scenes		34.10	.496	.000512	34.92

Velocity recovery is most sensitive to backward cycle and forward transport: their removal reduces cosine from .496 to $-.089$ and .010, respectively; without recurrent rollout it falls to .201. In contrast, removing rendering, proximal anchoring, or mass balance retains cosine of .488, .479, and .479. Removing vorticity evolution or the remaining regularizers gives .450 or .401. Thus the reconstruction and velocity metrics respond differently to the loss groups. Among the measured ablation rows, removing forward transport or proximal anchoring gives the lowest test-view re-simulation PSNR (28.35/28.37 dB), despite their different velocity cosines. This indicates that directional agreement alone does not determine transport quality. The corresponding test-view re-simulation drops relative to the full pipeline are 6.57 and 6.55 dB.

B.7 Per-scene and sampling-seed stability

This experiment separates variation across scenes from variation due to diffusion noise.

Table 8 separates scene variation from sampling-seed variation. The full model, w/o temporal-window synchronization, and w/o conditioning each use at least 3 posterior seeds × 5 Plume test scenes. The seed protocol varies initial diffusion noise while fixing the video, checkpoint, cameras, and refinement settings. Metrics are first aggregated over frames and views; scene rows use means and sample standard deviations across seeds. Across scenes reuses the five-scene Plume PSNR averages from Table 2 and velocity metrics from Table 1.

B.8 Temporal coordination: density errors and transport consistency

This final analysis checks whether temporal coordination improves the density sequence itself, not only the recovered velocity.

Table 9 compares the full pipeline, removal of temporal-window synchronization, and removal of motion compensation alone, using the interventions in Appendix B.6. The comparison holds scenes, observations, initial sampling noise, and Stage 3 settings and budget fixed. It tests whether temporally coordinated density initialization is associated with lower density errors and transport residuals, complementing the velocity comparisons in Table 3.

Evaluation uses the five Plume test scenes and the common $40\times80\times40$ grid from Appendix B.5. Let $\mathcal{F} = \{50, \dots, 149\}$, $\mathcal{T} = \{50, \dots, 148\}$, and G be the number of grid points. For $q \in \{\text{init}, \text{final}\}$, density error is

$$E_\rho^q = \frac{1}{|\mathcal{F}|G} \sum_{k\in\mathcal{F}} \sum_{\mathbf{x}} \left(\rho_k^q(\mathbf{x}) - \rho_k^{\text{GT}}(\mathbf{x})\right)^2, \qquad \rho_k^{\text{init}} = \widetilde{\rho}_k. \tag{53}$$

The initial density is the Stage 2 output multiplied by the dataset constant a_{ds}; the final density is the accepted Stage 3 output. Predictions and references use the same dataset density units, without scene- or frame-specific normalization or a fitted evaluation-time scale. The one-step transport residual is

$$R_{\text{tr}} = \frac{1}{|\mathcal{T}|G} \sum_{k\in\mathcal{T}} \sum_{\mathbf{x}} \left[\rho_{k+1}^{\text{final}} - \text{Adv}_{\text{RK2}}\left(\rho_k^{\text{final}}, \mathbf{u}_k^{\text{final}}\right) - s_k^{\text{final}}\right]^2 (\mathbf{x}). \tag{54}$$

This is an unweighted full-grid mean squared residual, distinct from the weighted Charbonnier training loss. Each step starts from the corresponding final reconstructed density. All metrics are averaged within each scene and then equally over scenes; ground truth is used only for evaluation. The full pipeline has the lowest initial and final density MSE and transport residual in Table 9. Together

Table 9: Temporal-coordination ablation on five Plume test scenes. Density MSE is measured before and after Stage 3; transport residual uses the final outputs. All metrics are lower-is-better; best values are bold.

	Density MSE		Transport
Method	Initial ↓	Final ↓	R_{tr} ↓
Full pipeline	**.000621**	**.000512**	**.00003**
w/o temporal-window synchronization	.001058	.001392	.00007
w/o motion compensation	.000982	.000838	.00004

with the .496/.255/.440 velocity cosines for full/no synchronization/no motion compensation in Table 3, this shows that the window constraints improve density reconstruction, transport consistency, and velocity direction jointly, supporting the role of temporally coordinated density inference in motion recovery.

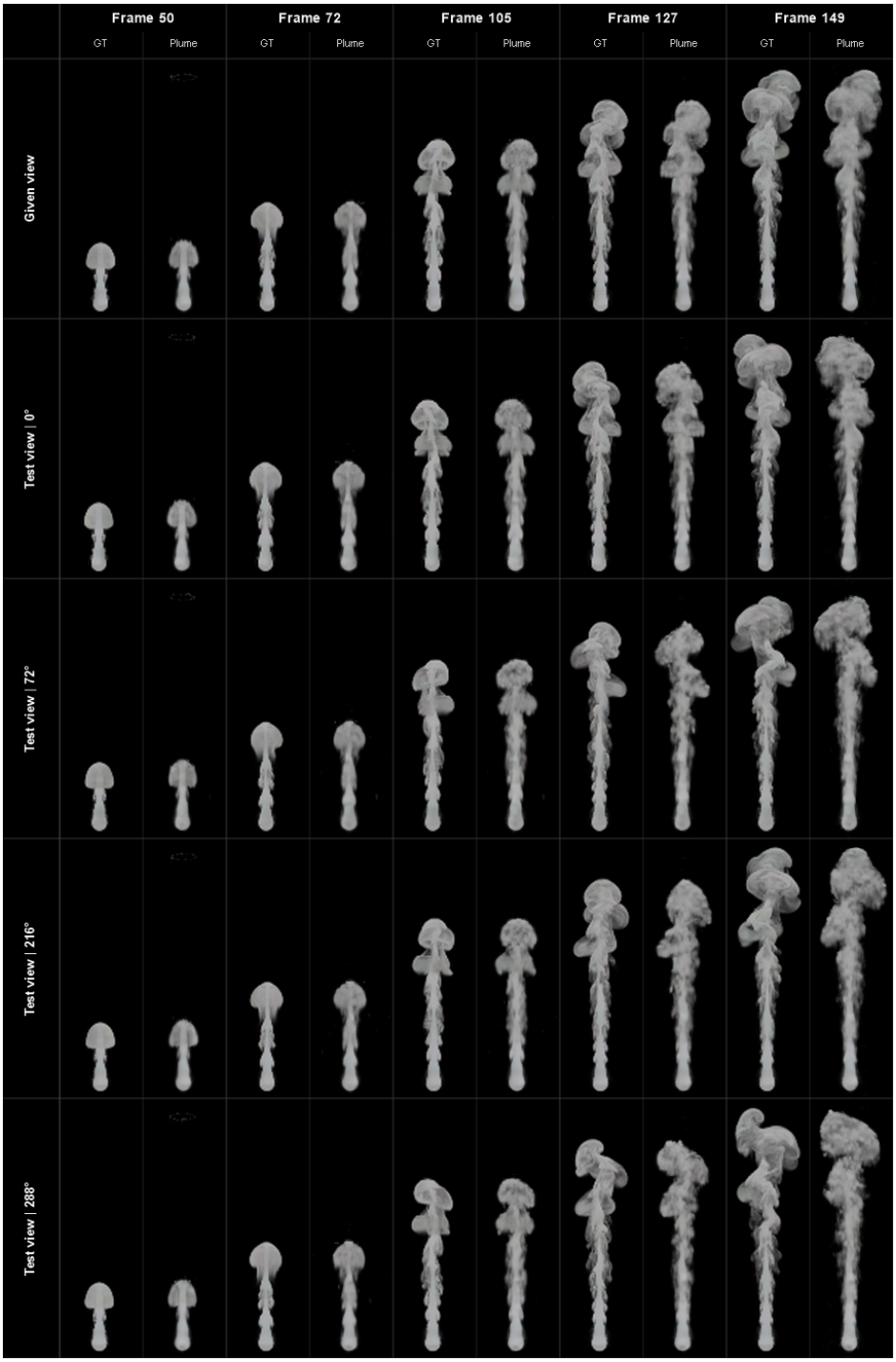


Figure 6: Multi-view Plume visualization from a randomly selected scene among the five test scenes, comparing GT with ours in the given view and four held-out views. Quantitative comparisons aggregate all five scenes.

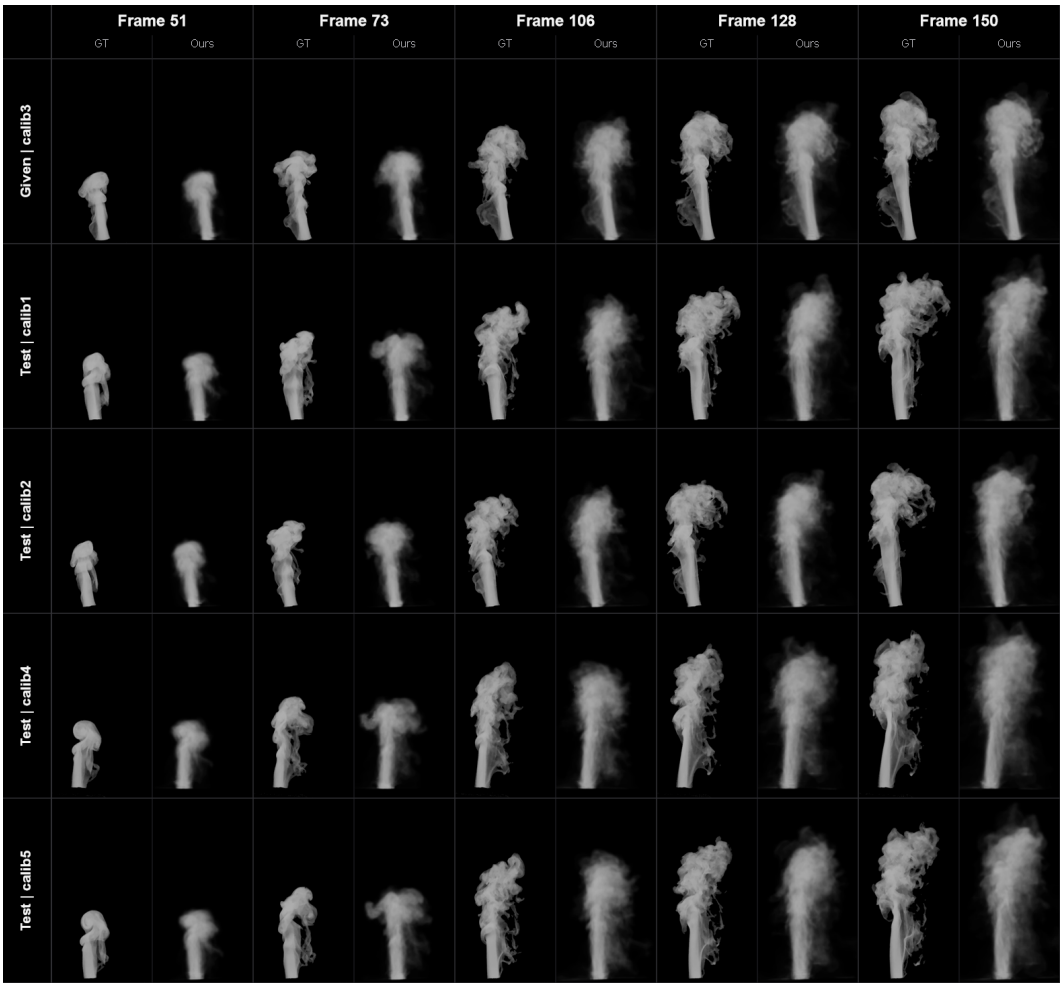


Figure 7: Multi-view ScalarFlow visualization from a randomly selected scene among the five test scenes, comparing GT with ours in the given view and four held-out views. Displayed frame labels 51, 73, 106, 128, and 150 are one-based, corresponding to zero-based frames 50, 72, 105, 127, and 149, all within the reconstruction interval 50–149. Quantitative comparisons aggregate all 100 frames per scene and all five scenes.

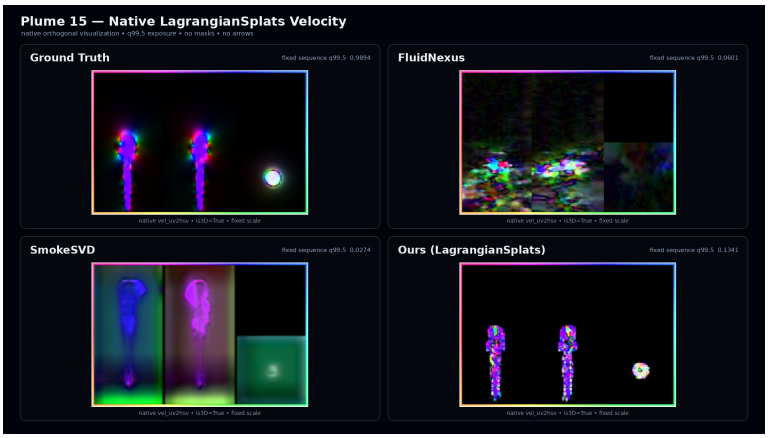


Figure 8: Plume15 velocity visualization from the final frame of the native UV-to-HSV comparison video: GT, FluidNexus, SmokeSVD, and Ours. Hue encodes projected direction and brightness encodes speed with fixed per-method 99.5th-percentile scales, so colors do not compare absolute speed across methods. Quantitative comparisons use unscaled fields (Table 1).

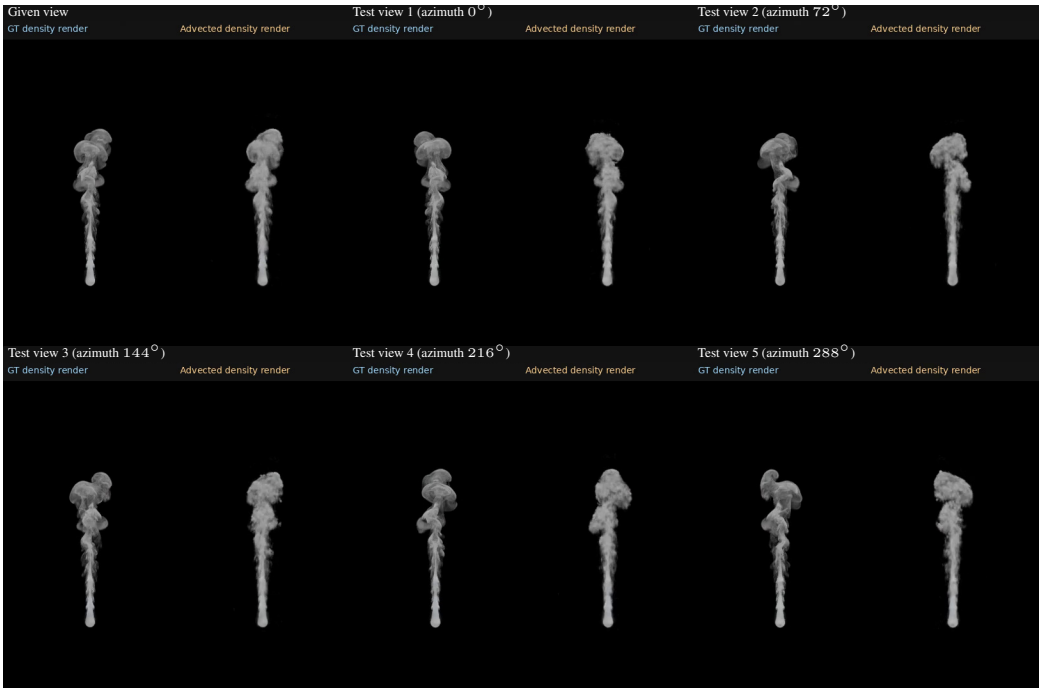


Figure 9: Six-view Plume re-simulation visualization from a randomly selected scene among the five test scenes. Frame-matched targets and our DFK-advected results are shown over a full orbit. The transport protocol is defined in Appendix B.5; quantitative results belong to the re-simulation block of Table 2.

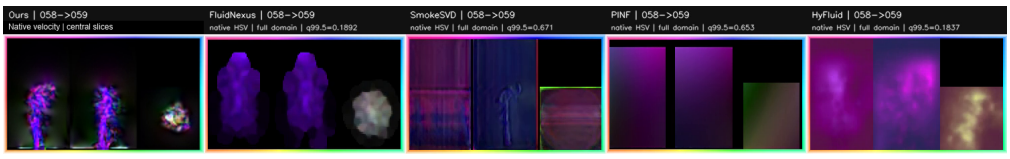


Figure 10: Velocity-field reconstruction on ScalarFlow. From left to right: Ours, FluidNexus, SmokeSVD, PINF, and HyFluid. Color-coded slices provide a qualitative comparison of spatial flow structure across methods. Each method retains its native visualization scale.