WildSmoke: Ready-to-Use Dynamic 3D Smoke Assets from a Single Video in the Wild

Anonymous CVPR submission

Paper ID 20

Abstract

*We propose a pipeline to extract and reconstruct **dynamic 3D smoke assets** from a **single in-the-wild video**, and further integrate interactive simulation for smoke design and editing. Recent developments in 3D vision have significantly improved reconstructing and rendering fluid dynamics, supporting realistic and temporally consistent view synthesis. However, current fluid reconstructions rely heavily on carefully controlled clean lab environments, whereas real-world videos captured in the wild are largely underexplored. We pinpoint three key challenges of reconstructing smoke in real-world videos and design targeted techniques, including smoke extraction with background removal, initialization of smoke particles and camera poses, and inferring multi-view videos. Our method not only outperforms previous reconstruction and generation methods with high-quality smoke reconstructions (+2.22 average PSNR on wild videos), but also enables diverse and realistic editing of fluid dynamics by simulating our smoke assets. We promise to release our models, data, and 4D smoke assets. We also include more video results in the supplement.*

1. Introduction

Fluid phenomena, from eddies around high-speed trains to smoke rings from jet engines, are ubiquitous. A key challenge is reconstructing unobserved quantities such as velocity and density over the full spatiotemporal domain (3+1D) from visual inputs (2D photos or video). This task, formally inferring **3D fluid fields** from imagery, i.e., reconstructing fluid dynamics from visual observations, underpins applications from high-fidelity smoke rendering in visual effects [18, 37] to diagnostics in industrial flows [3, 28, 33]. Recent 3D vision advances have accelerated progress: multi-view benchmarks [9] provide accurately calibrated flow videos, and new methods infer fluid fields by jointly optimizing differentiable physics (via physics-based constraints) and neural scene representations (via rendering losses) from video [6–8, 10, 11, 41].

Despite notable advances, most existing approaches are designed and heavily evaluated on **deliberately controlled multi-view** recordings for reconstructing fluid phenomena, which **differ substantially from real-world smoke in the wild**. Datasets such as ScalarFlow [9], TomoFluid [44], and FluidNexus [10] rely on fixed, precisely calibrated cameras, controlled smoke generation in laboratory settings, and pre-captured simplistic backgrounds. Such conditions are impractical for in-the-wild smoke, which is often recorded with a single moving camera (e.g., handheld or drone footage), with cluttered backgrounds and unpredictable camera motion.

At the same time, demand for ready-to-use 3D assets has surged across graphics and simulation applications [27, 38, 46]. **Dynamic 3D smoke assets are particularly valuable** for downstream visual effects editing and real-time simulations [2, 5, 47], owing to their intricate motion and evolving volumetric properties. Yet, despite their importance, the generation of dynamic 3D smoke assets remains largely underexplored. We are motivated to ask:

> *How to bridge the gap between existing lab-controlled smoke reconstruction methods and in-the-wild smoke videos?*

In this work, we aim to reconstruct ready-to-use dynamic 3D smoke assets from a single in-the-wild video. We identify three key challenges of reconstructing smoke fields from the wild (Fig. 1, Section 2.2): noisy backgrounds, unknown camera poses, and coupled camera viewpoints and timesteps. Correspondingly, we introduce three key techniques: 1) In-context smoke segmentation, plus dehazing for light smoke background removal; 2) Initializations of smoke particles and pose estimation; 3) Decoupling spatial and temporal camera trajectories by multi-view generation and local pose perturbation. More importantly, we demonstrate practical applications of our reconstructed smoke assets through fluid simulation.

Our main contributions are summarized as follows:

1. We design a comprehensive pipeline that makes smoke reconstruction practical for unconstrained in-the-wild videos, extracting and reconstructing dynamic 3D smoke assets with consistent improvements over prior work (+2.22 dB PSNR on average on in-the-wild videos) (Section 4.3).
2. We comprehensively study alternative options for each component in our framework, and quantify the impact of each alternative on the final reconstruction performance (Section 3 and Supplement 8).
3. Our smoke assets are ready-to-use: we support realistic editing of smoke through interactive fluid simulations (Section 3.5 and 4.4).

2. Background

2.1. Smoke Reconstruction from Videos

Given a single-view video capturing upward-rising smoke, our goal is to reconstruct a dynamic 3D representation of the smoke field over time ($t = 1, \cdots, T$) that is coherent, view-consistent, and editable. Following FluidNexus [10], we represent fluid with two types of particles.

1. 3D physical particles for positions and velocities: $\mathbf{p}_t^{\text{phy}}, \mathbf{u}_t \in \mathbb{R}^{N_t^{\text{phy}} \times 3}$, where N_t^{phy} is the number of physical particles at t. The density field $\rho_t : \mathbb{R}^3 \mapsto \mathbb{R}$ and velocity field $\mathbf{V}_t : \mathbb{R}^3 \mapsto \mathbb{R}^3$ can be further mapped from particles to grids via kernel-weighted interpolation [10, 14–16, 25].
2. N_t^{vis} visual particles (grayscale) at t with attributes: $\{\mathbf{p}_t^{\text{vis}} \in \mathbb{R}^{N_t^{\text{vis}} \times 3}, \mathbf{c}_t \in \mathbb{R}^{N_t^{\text{vis}}}, \mathbf{s}_t \in \mathbb{R}^{N_t^{\text{vis}} \times 3}, \mathbf{o}_t \in \mathbb{R}^{N_t^{\text{vis}}}, \mathbf{r}_t \in \mathbb{R}^{N_t^{\text{vis}} \times 4}\}$, representing position, color, scale, opacity, and rotation, respectively.

2.2. Smoke Reconstruction in the Wild: Importance and Challenges

While smoke can be plausibly simulated with strong artistic control, simulation alone is often underconstrained to match a specific real shot and its scene interactions. Reconstructing smoke from in-the-wild videos recovers scene-conditioned density/velocity that can initialize and steer downstream simulation-based editing.

Our primary technical contribution is, *for the first time*, to identify and summarize three key challenges in reconstructing 4D smoke assets from a single in-the-wild video (Fig. 1), and to propose a dedicated pipeline that explicitly addresses these challenges and outperforms existing smoke reconstruction methods [10, 41] (Section 3).

Noisy Backgrounds and Boundaries. High-quality videos of real smoke like ScalarFlow [9], TomoFluid [44], and FluidNexus [10] were collected with carefully controlled backgrounds and further post-processed to remove environmental noise. However, videos in the wild are inevitably contaminated by complex backgrounds visible through semi-transparent smoke.

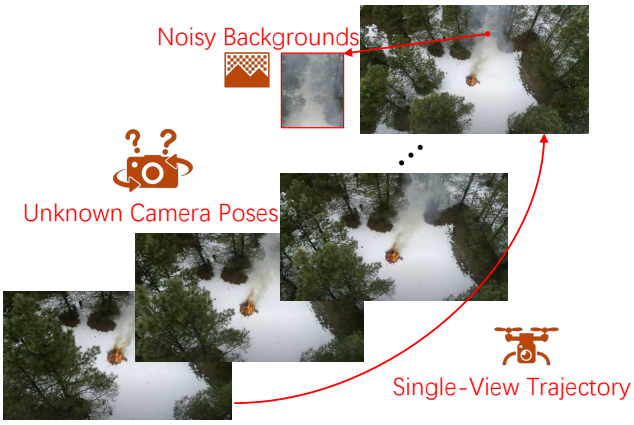

Figure 1. Challenges of smoke reconstruction from a single in-the-wild video: 1) noisy backgrounds and boundaries; 2) unknown camera poses; 3) single video with coupled camera viewpoints and timesteps.

Unknown Camera Poses. Although videos of smoke in the wild are captured with moving cameras, their camera poses are typically neither measured nor released with the footage.

Single-Camera Trajectory. As a video of smoke in the wild typically provides only one camera trajectory over time, spatial viewpoints and temporal frames become inherently entangled: there is an (approximately) one-to-one correspondence between camera viewpoints and timesteps.

3. Methods

To address the above challenges, we propose a unified pipeline to reconstruct clean, background-free smoke from unconstrained real-world videos, as shown in Fig. 2.

3.1. Smoke Extraction

In real-world scenarios, smoke often exhibits complex, irregular boundaries and is intertwined with significant noise from backgrounds. To address this issue, we propose a pipeline that extracts clean smoke regions from a single in-the-wild video.

(a) Smoke Mask Extraction. We first segment the smoke and obtain a binary mask sequence $\mathbf{M} = \{M_t\}$ over all timesteps t in the video. Owing to the *widely observed limitation* of off-the-shelf semantic segmentation models, which generally struggle to generalize to in-the-wild smoke, we adopt a one-shot learning strategy, using SAM [19] for annotation and SegGPT [36] for propagation[1]. For each

[1] Although it is possible to fine-tune a segmentation model for general smoke extraction, the required data collection and annotation effort is beyond the scope of this work.

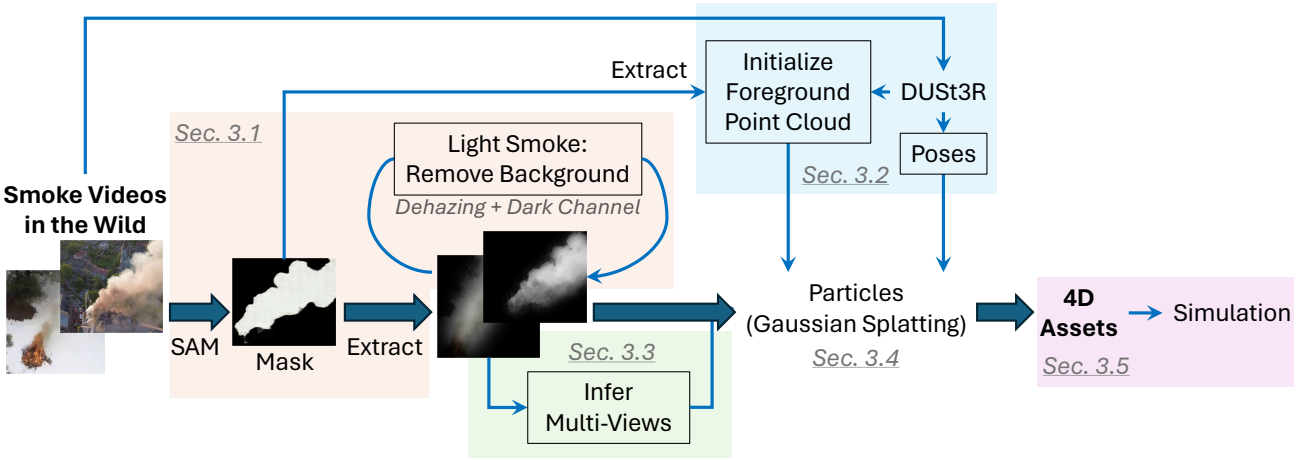

Figure 2. To reconstruct smoke from a single video in the wild, our pipeline includes five steps: 1) Smoke extraction, with background removed for light smoke due to translucence (Section 3.1); 2) Pose estimation and coarse initialization for 3D point cloud (Section 3.2); 3) Inferring multi-views for smoke (Section 3.3); 4) Training smoke particles (Section 3.4); 5) composition and simulation of 4D smoke assets (Section 3.5). "SAM": segment anything [19]. "DUSt3R" [35].

video, we select an early reference frame I_{ref}, interactively annotate its smoke with SAM[2] to obtain M_{ref}, and then feed the pair $(I_{\text{ref}}, M_{\text{ref}})$ to SegGPT [36] to perform one-shot inference and produce masks M_t for the remaining frames I_t. This combines SAM's high-quality per-frame annotation with SegGPT's frame-wise generalization.

The boundary of segmented smoke could still be noisy. We apply a Gaussian filter f_g to smooth the boundary of inferred masks, yielding a more natural smoke contour. The smoke areas are extracted via $\tilde{S}_t = f_g(M_t) \cdot I_t$. As shown in Fig. 3(a), this in-context segmentation can reliably segment smoke boundaries. See Supplement9.2 in the supplement for more smoke segmentation examples.

At this stage, it is necessary to further distinguish dense from light smoke. We categorize them based on the visibility of the background. For dense smoke, the background is completely occluded due to high optical density (low transmittance). In this case, background contamination is negligible, and the smoke can be directly treated as the masked region. However, for the light smoke, the background remains partially visible (nonzero transmittance).

(b) Extract Clean Smoke. Similar to fog or haze, background objects remain visible through light smoke due to light transmission, introducing artifacts during reconstruction. For light smoke, we fine-tune a pretrained DehazeFormer [31] to isolate the smoke and remove the background. We construct a fine-tuning dataset of synthetic smoky inputs $\tilde{I}$ (pixels normalized to $[0, 1]$) by blending coarsely extracted smoke $\tilde{S}$ with video frames of clean background without smoke I^{clean}, which serve as supervision targets:

$$\tilde{I} = I^{\text{clean}} \cdot \tilde{T} + A \cdot \tilde{S}, \tag{1}$$

where $\tilde{T} = 1 - \tilde{S}$ is the coarse transmission map, representing the amount of background light that reaches the camera through the smoke. We estimate the atmospheric light A using the dark channel prior [12]. As shown in Fig. 3 (c), the fine-tuned DehazeFormer can remove the foreground smoke and recover the background $\tilde{I}^{\text{clean}}$.

For dense smoke where background contamination is negligible, the extracted smoke $\tilde{S}$ from masks can be used directly without this dehazing step.

For light smoke, we extract the clean foreground smoke based on Equation (1):

$$S = 1 - \frac{I - A}{\tilde{I}^{\text{clean}} - A}, \tag{2}$$

where I is the smoky frame in the original video, and $\tilde{I}^{\text{clean}}$ is the background recovered by DehazeFormer. The resulting foreground-smoke images are then used for smoke-field reconstruction.

Alternative Approach: Mask-to-Matte Refinement. We also provide an alternative for dehazing. We refine the coarse binary masks into pixel-accurate alpha mattes using VideoMaMa [21], a mask-guided video matting model that converts masks to alpha.

We apply VideoMaMa to the coarse smoke mask sequence $\{M_t\}$ and obtain an alpha matte sequence $\{\alpha_t\}$, where $\alpha_t \in [0, 1]$ encodes smoke opacity. We then use $S = \alpha_t * I_t$ as our grayscale smoke observation for all subsequent steps (i.e., background-free smoke input to reconstruction).

3.2. Pose Estimation and Coarse Geometry

To obtain camera intrinsics and extrinsics for frames, we initialize camera poses and focal lengths with estimates

[2]We use an online SAM-based annotator https://roboflow.com/.

CVPR #20

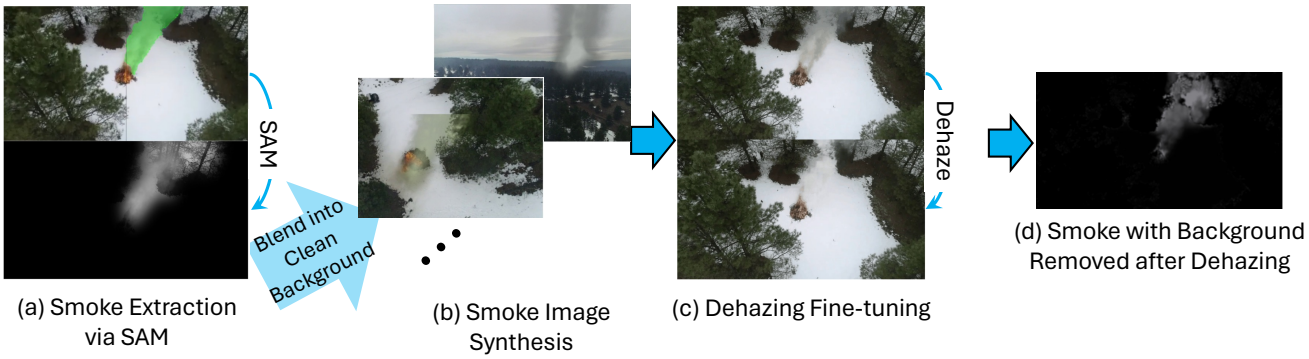

Figure 3. Smoke extraction, with background removal for light smoke.

from the pretrained DUSt3R [35]. We further initialize both physical and visual particles from a sparse 3D point cloud $\{\mathbf{p}_i\}_{i=1}^{N}$ estimated by DUSt3R, which provides a rough geometry of the scene, including both smoke and background regions. We filter the DUSt3R-generated 3D point cloud using the extracted smoke masks, retaining only the smoke-relevant points to initialize our particles. Although DUSt3R produces coarse results, this initialization provides meaningful spatial priors for optimizing our particles; without it, training may fail to converge. See more details in Supplement7.2 in the supplement.

Alternative Approach: VGGT [34]. This initialization can be replaced by any pose estimation work pretrained on outdoor datasets. We also test the results on VGGT [34] (See Supplement 8 for experiments and more details).

3.3. Inferring Multi-View Videos

In wild smoke videos, cameras typically follow a single trajectory. For example, a drone may fly from the ground upwards along with the smoke plume to capture the video. Although the camera may span a wide spatiotemporal extent, camera viewpoints and timesteps are highly coupled, i.e., each camera viewpoint is associated with a unique timestep, and vice versa. This coupling can cause overfitting and degrade novel-view synthesis for unseen viewpoints or timesteps.

To decouple the spatiotemporal camera trajectory, in our work we employ generative multi-view synthesis. To obtain multi-view supervision from single-view input, we use pretrained SV4D 2.0 [40] to generate smoke videos from novel views. We choose azimuth angles of $[-10°, 10°, 20°, 30°]$ relative to the pose of the current frame as our novel viewpoints. Since SV4D 2.0 does not support long temporally consistent video generation, we split our video sequences into short clips, and overlap one frame between neighboring clips, ensuring valid conditioning for every novel-view segment. Per-frame camera-to-world poses are obtained by applying the angle set to the DUSt3R-initialized poses.

Moreover, to address the unreliable frames produced by SV4D at later timesteps, we apply an exponentially decaying weight over the frame index. This strategy progressively down-weights the influence of later frames generated by SV4D 2.0 during Gaussian-particle training. See Supplement 7.4 and 7.5 for more details.

Alternative Approach: The multi-view generator can be replaced by other novel-view video synthesis models, e.g., TrajectoryCrafter [42]. However, for outdoor videos with cluttered backgrounds, their quality degrades severely under large viewpoint changes and long sequences. In our experiments, SV4D 2.0 remains the most reliable drop-in choice after background removal. See examples in Supplement 8.

3.4. Training Gaussian Particles

Following the training strategy of FluidNexus [10], we separately train visual and physical particle representations. Particles are optimized by minimizing both visual supervision (photometric errors between input frames and rendered views using 3D Gaussian Splatting [17]) and also physical regularization based on fluid simulation. To overcome the *"numerical diffusion"* during fluid simulation that leads to artificial smoothing [1], we introduce a frequency-aware regularization during training. Moreover, to determine the *unknown buoyancy in the wild*, we make its strength learnable. For more training details, we refer readers to Supplement 7.6.

To further disentangle spatial and temporal viewpoints in single-camera trajectories from in-the-wild videos, we not only train with our generated multi-view trajectories (Section 3.3), but also progressively enrich the view trajectory by **perturbing local camera poses**. Specifically, after the particles at time t converge, we introduce perturbed viewpoints by shifting the camera pose forward by Δt along the trajectory (modulo the sequence length T). Concretely, together with the original pose $(\mathbf{R}_t, t)$, we also include $(\mathbf{R}_{(t+\Delta t) \bmod T}, t)$ as an input and its corresponding rendering result as the target. Here, $\mathbf{R}$ is the rotation matrix of the camera's extrinsics; $\Delta t \ll T$, i.e., we perturb within a very short period relative to the whole temporal domain. Essentially, we effectively decouple viewpoint and timestep by associating pose $\mathbf{R}_{(t+\Delta t) \bmod T}$ with timestep t via "local pose perturbation." Since Δt is small, perturbed

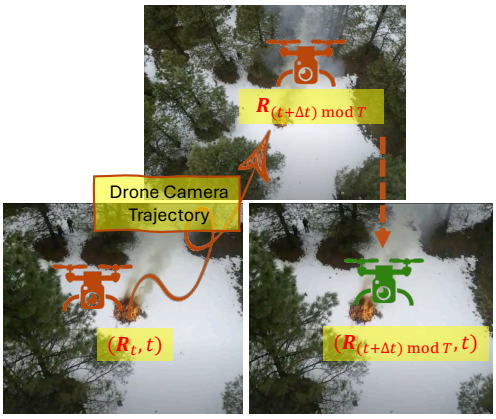

Figure 4. Local pose perturbation. Given the original camera pose $(\mathbf{R}_t, t)$ on a single-video trajectory ("red drone"), we additionally introduce a perturbed viewpoint by shifting the camera pose by Δt along the trajectory with wrap-around $(t+\Delta t) \bmod T$ ("green drone"). For this perturbed viewpoint, the rendered image serves as supervision. Both original and perturbed samples are used during following training.

novel viewpoints still reside in neighborhoods of the original video trajectory. In practice, we progressively increase Δt from 2 to 4 during training.

Alternative Approach. We also consider HyFluid [41] and FluidNexus [10], but both are less suitable for in-the-wild videos in our setting. FluidNexus assumes a fixed buoyancy term and lacks our spatiotemporal augmentation (local pose perturbation), which can lead to incorrect force balance and poorer reconstruction, which motivates our learnable buoyancy and augmentation design. Detailed comparisons are reported in Supplement 8.

3.5. Ready-to-Use 4D Smoke Assets

A key use case of our reconstructed smoke assets is visual-effects (VFX) editing of novel smoke scenes. In our work, we consider simulating smoke interactions using PhiFlow [13]. The simulation is initialized with the reconstructed density field ρ from visual particles, and velocity field $\mathbf{V}$ from physical particles:

Density Field from Visual Particles. For each visual particle i with center $\mathbf{p}^{\text{vis}}_{t,i}$, scale $\mathbf{s}_{t,i}$, rotation $\mathbf{r}_{t,i}$, and opacity $\mathbf{o}_{t,i}$, we define a Gaussian kernel

$$\phi_{t,i}(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{p}^{\text{vis}}_{t,i})^\top \mathbf{\Sigma}^{-1}_{t,i}(\mathbf{x}-\mathbf{p}^{\text{vis}}_{t,i})\right), \quad (3)$$

where $\mathbf{\Sigma}_{t,i} = R(\mathbf{r}_{t,i})\,\mathrm{diag}(\mathbf{s}^2_{t,i})\,R(\mathbf{r}_{t,i})^\top$, $\mathbf{r}_{t,i} \in \mathbb{R}^4$ is the unit quaternion representing the particle rotation, and $R(\mathbf{r}_{t,i}) \in \mathbb{R}^{3\times3}$ is the corresponding rotation matrix obtained from $\mathbf{r}_{t,i}$. The density field is the weighted sum of all particle kernels:

$$\rho_t(\mathbf{x}) = \sum_{i=1}^{N^{\text{vis}}_t} \mathbf{o}_{t,i}\,\phi_{t,i}(\mathbf{x}). \quad (4)$$

Velocity Field from Physical Particles. Given a physical particle with position $\mathbf{p}^{\text{phy}}_{t,i}$ and velocity $\mathbf{u}_{t,i}$, we map them to the grid. The Gaussian kernel $\phi'_{t,i}$ for the velocity field has the same form as $\phi_{t,i}$ but is centered at $\mathbf{p}^{\text{phy}}_{t,i}$:

$$\mathbf{V}_t(\mathbf{x}) = \frac{\sum_{i=1}^{N^{\text{phy}}_t} \phi'_{t,i}(\mathbf{x})\,\mathbf{u}_{t,i}}{\sum_{i=1}^{N^{\text{phy}}_t} \phi'_{t,i}(\mathbf{x}) + \varepsilon}, \quad (5)$$

here we use $\varepsilon = 10^{-8}$. To ensure spatial consistency between the density and velocity fields, both visual and physical splatting are restricted to the same bounding region defined by the visual particles.

After the initialization of density and velocity fields, we continue the fluid dynamics in PhiFlow [13] with the standard MacCormack semi-Lagrangian method [29] for advection and the projection-based pressure-Poisson solver for incompressible flow. We consider either external wind forces or an inserted obstacle to interact with the smoke.

4. Experiments

4.1. Settings

Datasets. We evaluate our pipeline on both synthetic and real-world collections of smoke, assessing the reconstruction quality using view rendering. Note that, in the **absence** of any existing large-scale benchmark dedicated to real-world smoke videos, we undertake a *substantial data collection effort* and construct our dataset from scratch.

- We **synthesize** two high-resolution and photorealistic smoke videos by combining a 4D smoke VDB sequence with a real 3D scene. These paired videos, rendered with distinct camera trajectories, allow direct comparison between our reconstructed results and the ground-truth. Specifically, we download a synthetic 4D smoke VDB sequence and a 3D scene from CGTrader[3], combine them and render two videos in Blender[4] with two camera trajectories. One trajectory is used for training and another for evaluation. See Supplement7.1 for more rendered samples.
- We also evaluate on two **real-world** testbeds: 1) FLAME dataset [30] is a fire-imaging dataset collected by drones during a prescribed burning of piled detritus in an Arizona pine forest. The dataset includes video recordings and thermal heatmaps captured by infrared cameras. 2) We further collect three videos of smoke from Pixabay [5], covering diverse scenarios.

To unify our training and evaluation settings, we standardize all our videos to 270 frames; training timesteps are

[3] https://www.cgtrader.com/3d-models
[4] https://www.blender.org
[5] Pixabay content is released under the Pixabay Content License; see https://pixabay.com/service/terms/.

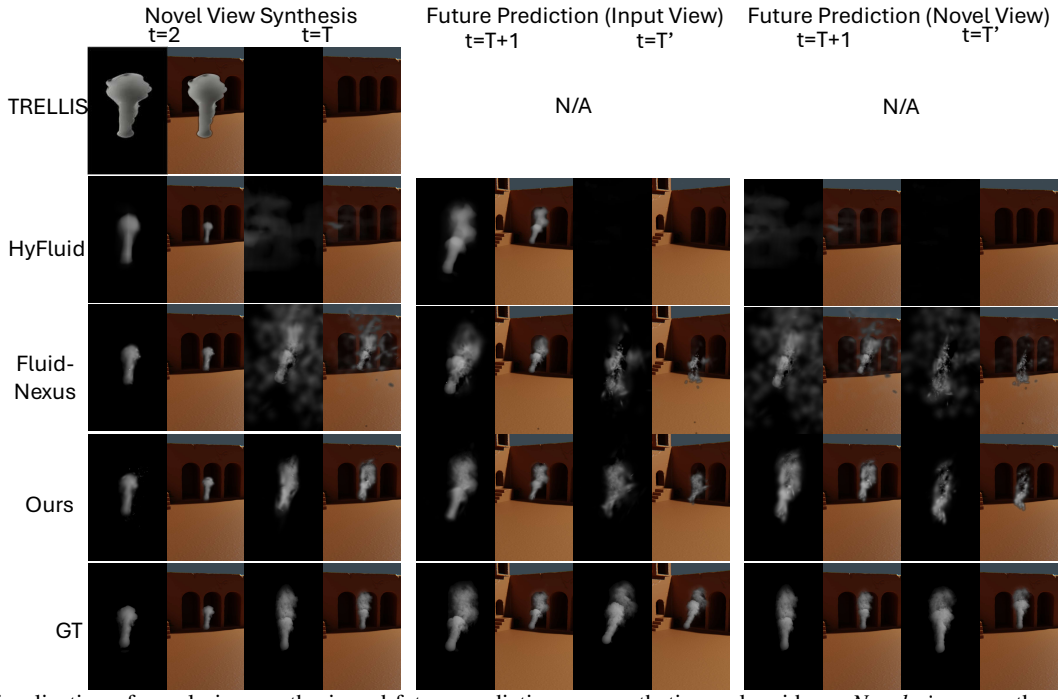

Figure 5. Visualization of novel view synthesis and future predictions on synthetic smoke videos. *Novel view* uses the camera pose at $t = 1$, and *input view* means the camera poses along the source video. Training uses $T = 240$ frames, and the unseen future extends to $T' = 270$ frames. *GT*: Ground Truth.

$t = 1, \ldots, T$ with $T = 240$, and unseen future timesteps are $t = T+1, \ldots, T'$ with $T' = 270$.

We measure view changes along the video trajectory via the camera's relative rotation angle $\Delta\theta_{t_1,t_2} = \arccos\left(\frac{\mathrm{tr}(\Delta\mathbf{R}_{t_1,t_2})-1}{2}\right)$, where $\Delta\mathbf{R}_{t_1,t_2} = \mathbf{R}_{t_2}^{\top}\mathbf{R}_{t_1}$ ($\mathbf{R}$ is the rotation matrix of the camera's extrinsics). In our synthetic data setting, the camera undergoes a rotation of $53°$ from $t = 1$ to $t = T$, which is larger than the $7°$ rotation during the future steps from $t = T+1$ to $t = T'$. Smoke in our videos is always centered in the scene.

Tasks. As explained by $\Delta\theta_{1,T}$ and $\Delta\theta_{T+1,T'}$ above, larger pose changes occur within the training timesteps than in the future steps. Synthesizing frames from the fixed viewpoint at the beginning of video ($t = 1$) is more challenging than following the camera pose trajectory.

Therefore, in our evaluation, the **novel view** is defined with camera pose at $t = 1$, and the **input view** follows the ground-truth camera trajectory.

Following [41], we consider the following two tasks:

- **Novel view synthesis:** We render smoke views from the fixed novel viewpoint over the training timesteps. Specifically, we fix the camera pose at $t = 1$ and synthesize novel views through $t = 2, \cdots, T$. We study novel view synthesis only on our synthetic dataset, due to the lack of ground-truth multi-view videos on real-world videos (FLAME and Pixabay).
- **Future prediction:** We extrapolate the fluid dynamics into the future steps $t = T+1, \cdots, T'$. No model is ever trained with ground-truth future frames from videos. On synthetic videos, we study both quantitative and visual results, and predict futures for both input view (i.e. follow the ground-truth camera pose trajectory during $t = T+1, \cdots, T'$) and novel view (fixed camera pose at $t = 1$). Due to the lack of ground-truth multi-view videos, on real-world videos (FLAME and Pixabay), we can only study future predictions based on the input view.

During inference, the learned velocity field is used to advect (evolve) the visual particles for future prediction. We refer the reader to [41] for more details about these tasks.

Evaluation Metrics. We report the peak signal-to-noise ratio (PSNR) averaged over frames, and defer the *structural similarity index measure* (SSIM) and the *perceptual metric LPIPS* [45] in Supplement 9.4. These metrics are *widely adopted* in prior smoke reconstruction [10, 41] and deblurring works [20, 26].

While perfectly faithful recovery of turbulent smoke from a single in-the-wild video is arguably ill-posed, and novel-view PSNR from a single input trajectory cannot fully resolve the resulting ambiguities, we design our method and evaluation protocol to provide, to the best of our knowledge, the most faithful reconstructions currently attainable under these challenging conditions.

Baselines. We compare with both reconstruction and generation methods, including HyFluid [41], FluidNexus [10], and Trellis [38]. We follow FluidNexus to train with the default grayscale input setting.

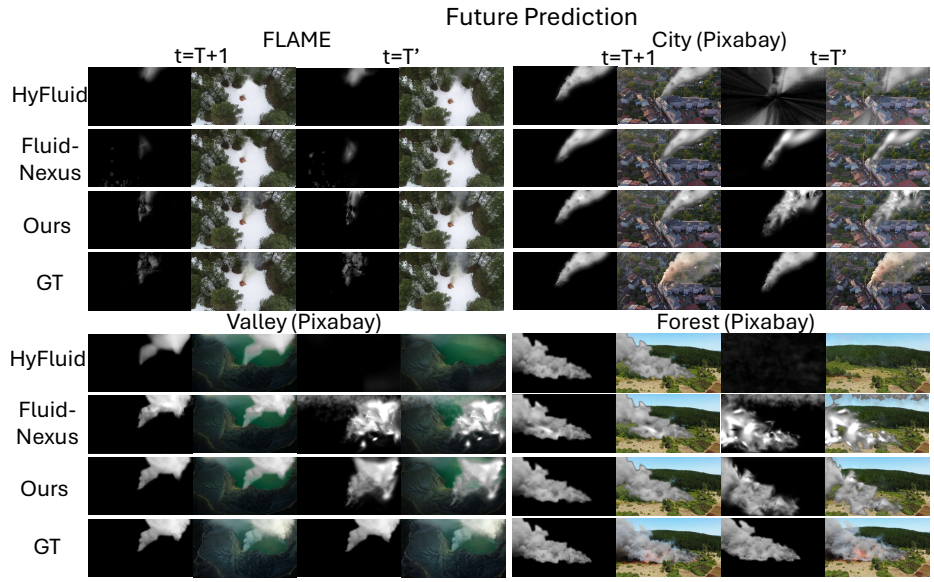

Figure 6. Visualization of future predictions (input view) on the FLAME dataset [30] and videos collected from Pixabay. Training uses $T = 240$ frames, and the unseen future extends to $T' = 270$ frames. *GT*: Ground Truth.

Table 1. Comparing PSNR (higher is better) of smoke reconstruction by different methods on the synthetic dataset. *Novel view* uses the camera pose at $t = 1$, and *input view* means the camera poses along the source video.

Methods	Novel View Synthesis	Future Prediction (Input View)	Future Prediction (Novel View)
Trellis [38]	19.98	-	-
HyFluid [41]	24.26	22.54	22.18
FluidNexus [10]	29.26	23.61	21.54
Ours	**29.78**	**25.26**	**25.04**

Table 2. Cumulative ablation study (PSNR) on the synthetic smoke video. *Novel view* uses the camera pose at $t = 1$, and *input view* means the camera poses along the source video.

	Novel View Synthesis	Future Prediction (Input View)	Future Prediction (Novel View)
Baseline	22.55	16.69	18.17
+ Smoke Extraction	29.26	23.61	21.54
+ DUSt3R Init.	29.48	26.45	22.92
+ Local Perturbation	29.77	**26.85**	23.59
+ Multi-Views (Ours)	**29.78**	25.26	**25.04**

4.2. Synthetic Data

Results. We first show quantitative results in Table 1 and visualizations in Fig. 5. Trellis [38] fits a static 3D asset per frame (image-to-3D), estimating neither time-varying particles nor velocities; hence future prediction is ill-defined. Moreover, per-frame inconsistency in the novel-view synthesis setting causes the smoke to progressively fade, yielding a black frame at time T. Compared with HyFluid [41] and FluidNexus [10], our pipeline achieves higher PSNR and more stable visualizations.

Ablation Study. We further provide ablation studies in Table 2; see visualization results in Supplement9.6. Smoke extraction (segmentation), initializing poses/particles, and adding inferred multi-view supervision progressively improve PSNR and visual quality. Note that, due to significant spatiotemporal deviations, the task of *future prediction at novel view* is substantially more challenging than either *novel view synthesis* or *future prediction at input view*. As a result, *future prediction at novel view* demands additional geometric cues, even coarse ones generated by SV4D, and therefore benefits most (+1.45 PSNR from 23.59 to 25.04) from incorporating "+ Multi-Views."

Table 3. Comparing PSNR (higher is better) of smoke reconstruction by different methods on the FLAME dataset [30] and videos collected from Pixabay.

Dataset	FLAME	Pixabay		
		City	Valley	Forest
HyFluid [41]	21.67	23.24	12.43	13.70
FluidNexus [10]	21.78	24.18	16.44	14.63
Ours	**22.88**	**24.68**	**20.42**	**17.91**

4.3. Smoke Videos in the Wild

We further evaluate on in-the-wild smoke videos. Due to the lack of ground-truth novel views in real videos, here we evaluate only the task of *future prediction at input view*, meaning the camera poses along the source video.

FLAME [30] and Pixabay. The FLAME dataset [30] includes drone-captured videos in the wild forest. The smoke is light, and thus background removal is necessary. We also evaluate three thick-smoke videos from Pixabay. We show the results in Table 3 and Fig. 6.

Overall, across both light and dense smoke over diverse real-world videos, our method outperforms prior work, achieving an average PSNR improvement of +2.22 dB. Qualitatively, the reconstructions remain consistent and blend back into the original backgrounds. We also test

the physical plausibility via velocity divergence in Supplement 9.1, and human validation in Supplement 9.8.

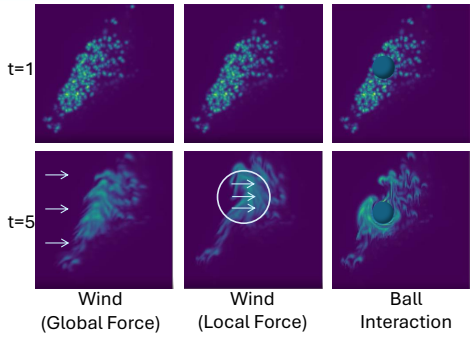

Figure 7. Visualizations of simulations with our smoke assets in different interaction scenarios. t means simulation time, different from the video frames we used in previous figures. The simulation starts with particles reconstructed at the last frame of the "city" video from Pixabay (Fig. 6 top).

4.4. Interactive Simulations

Finally, we demonstrate interactive simulations enabled by our fluid reconstruction. We consider two scenarios: (i) external wind (both global and local forces), and (ii) a rigid spherical obstacle. For the global wind, we apply a global constant force $f_w = (0.005, 0, 0)$ uniformly across the domain. The local wind uses the same magnitude and direction but is confined to a sphere of radius 30 at the scene center. For the obstacle case, we place a rigid ball (radius 10) centered at $(50, 70)$ and simulate the resulting interactions. All simulations use the reconstruction from the "city" video from Pixabay (Fig. 6 top); results are shown in Fig. 7 (XY-plane from Z-axis). We can see that our smoke assets support realistic editing via diverse simulation scenarios.

4.5. Training and Inference Cost

Table 4 in the Supplement 7.7 reports the per-stage GPU hours. Across datasets and resolutions, our pipeline reconstructs smoke from in-the-wild videos in 4.47 GPU-hours on average among all 5 videos. Notably, with the DUSt3R-based pose initialization, smoke can be localized without seeding a large number of particles, reducing the Gaussian-particle training stage by 1.5 GPU-hours.

5. Related Works

5.1. Fluid Field Extraction from Videos

A series of recent works tackled fluid reconstruction from videos, but most relied on controlled, multi-view recordings that differed substantially from in-the-wild footage. PINF [6] coupled Navier–Stokes PDEs with a continuous spatiotemporal NeRF to recover flow; NeuroFluid [11] introduced a particle-driven neural renderer that embedded fluid properties and a particle transition model; HyFluid [41] estimated hybrid neural fields for density and velocity using physics-based losses; FluidNexus [10] reconstructed smoke from a single video by leveraging generative priors, yet its multi-view generator was fine-tuned on laboratory smoke with fixed, calibrated cameras and clean backgrounds. To our knowledge, reconstructing fluid fields from a single in-the-wild video and turning them into ready-to-use dynamic 3D smoke assets remains largely underexplored.

5.2. Physics-based Fluid Simulation and Editing

Classical fluid simulation in computer graphics has been broadly categorized into Eulerian [32] and Lagrangian formulations [4]. Production solvers edited flows by manipulating external forces, boundary conditions, and solid–fluid interactions (obstacles), which underpinned VFX and interactive applications. Recent differentiable-physics systems such as DiffTaichi [15] and PhiFlow [13] made long-horizon, gradient-based editing practical. These tools provided useful platforms for controlled physical interactions. Building on these advances, our work reconstructs smoke directly from in-the-wild videos and integrates simulation for realistic editing, extending such physics-based editing to unconstrained real-world scenarios.

5.3. 4D Generation

Recent advances in generative modeling have markedly improved image-to-3D and image-to-novel-view-synthesis, and video diffusion models with stabilized virtual cameras produced short clips with a degree of multi-view consistency [22, 43, 48]. SV4D [39] and SV4D 2.0 [40] extended view-consistent generation to dynamic scenes, reinforcing temporal continuity. In our pipeline, we build on this generative consistency to obtain auxiliary multi-view trajectories, but go further by reconstructing physically plausible dynamic smoke fields.

6. Conclusion

In this paper, we addressed the challenge of recovering time-varying 3D smoke fields from a single in-the-wild video and producing ready-to-use dynamic assets. The proposed pipeline tackles three core challenges (noisy backgrounds, unknown camera poses, and coupled spatiotemporal views) through one-shot segmentation and dehazing, pose estimation, and decoupled spatiotemporal trajectory through multi-view generation and local pose perturbation. We further showed that the reconstructed assets support physically consistent editing through fluid simulation, aligning visual fidelity with physical plausibility and supporting downstream simulation workflows.

References

[1] Numerical diffusion. https://mooseframework.inl.gov/modules/porous_flow/numerical_diffusion.html. PorousFlow module documentation, Idaho National Laboratory, accessed 2025. 4

[2] Kai Bai, Wei Li, Mathieu Desbrun, and Xiaopei Liu. Dynamic upsampling of smoke through dictionary-based learning. *ACM TOG*, 40(1):1–19, 2020. 1

[3] Samuel J Baker, Michael A Hobley, Isabel Scherl, Xiaohang Fang, Felix CP Leach, and Martin H Davy. Enginebench: flow reconstruction in the transparent combustion chamber iii optical engine. *arXiv preprint arXiv:2406.03325*, 2024. 1

[4] Andrew Bennett. *Lagrangian fluid dynamics*. Cambridge University Press, 2006. 8

[5] Mengyu Chu, Nils Thuerey, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Learning meaningful controls for fluids. *ACM TOG*, 40(4):1–13, 2021. 1

[6] 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 (ToG)*, 41(4):1–14, 2022. 1, 8

[7] Yitong Deng, Hong-Xing Yu, Jiajun Wu, and Bo Zhu. Learning vortex dynamics for fluid inference and prediction. *arXiv preprint arXiv:2301.11494*, 2023.

[8] Yitong Deng, Hong-Xing Yu, Diyang Zhang, Jiajun Wu, and Bo Zhu. Fluid simulation on neural flow maps. *ACM TOG*, 42(6):1–21, 2023. 1

[9] Marie-Lena Eckert, Kiwon Um, and Nils Thuerey. Scalarflow: a large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. *ACM TOG*, 38(6):1–16, 2019. 1, 2

[10] Yue Gao, Hong-Xing Yu, Bo Zhu, and Jiajun Wu. Fluidnexus: 3d fluid reconstruction and prediction from a single video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26091–26101, 2025. 1, 2, 4, 5, 6, 7, 8, 3

[11] Shanyan Guan, Huayu Deng, Yunbo Wang, and Xiaokang Yang. Neurofluid: Fluid dynamics grounding with particle-driven neural radiance fields. In *International Conference on Machine Learning*, pages 7919–7929. PMLR, 2022. 1, 8

[12] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE TPAMI*, 33(12): 2341–2353, 2010. 3

[13] Philipp Holl and Nils Thuerey. Φ_{flow} (PhiFlow): Differentiable simulations for pytorch, tensorflow and jax. In *International Conference on Machine Learning*. PMLR, 2024. 5, 8

[14] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM TOG*, 38(6):201, 2019. 2

[15] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *ICLR*, 2020. 8

[16] Yuanming Hu, Jiafeng Liu, Xuanda Yang, Mingkuan Xu, Ye Kuang, Weiwei Xu, Qiang Dai, William T. Freeman, and Frédo Durand. Quantaichi: A compiler for quantized simulations. *ACM TOG*, 40(4), 2021. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. 4

[18] Byungsoo Kim, Vinicius C Azevedo, Markus Gross, and Barbara Solenthaler. Lagrangian neural style transfer for fluids. *ACM TOG*, 39(4):52–1, 2020. 1

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 2, 3

[20] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 6

[21] Sangbeom Lim, Seoung Wug Oh, Jiahui Huang, Heeji Yoon, Seungryong Kim, and Joon-Young Lee. Videomama: Mask-guided video matting via generative prior. *arXiv preprint arXiv:2601.14255*, 2026. 3

[22] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9298–9309, 2023. 8

[23] Miles Macklin and Matthias Müller. Position based fluids. *ACM TOG*, 32(4):1–12, 2013. 2

[24] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM TOG*, 33(4):1–12, 2014. 2

[25] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159, 2003. 2

[26] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 6

[27] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1

[28] Pankaj Saini, Christoph M Arndt, and Adam M Steinberg. Development and evaluation of gappy-pod as a data reconstruction technique for noisy piv measurements in gas turbine combustors. *Experiments in Fluids*, 57(7):122, 2016. 1

[29] Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. An unconditionally stable maccormack method. *Journal of Scientific Computing*, 35(2): 350–371, 2008. 5

[30] Alireza Shamsoshoara, Fatemeh Afghah, Abolfazl Razi, Liming Zheng, Peter Z Fulé, and Erik Blasch. Aerial imagery pile burn detection using deep learning: The flame dataset. *Computer Networks*, 193:108001, 2021. 5, 7, 4

[31] Yuda Song, Zhuqing He, Hui Qian, and Xin Du. Vision transformers for single image dehazing. *IEEE TIP*, 32:1927–1941, 2023. 3, 5

[32] M'hamed Souli and David J Benson. *Arbitrary Lagrangian Eulerian and fluid-structure interaction: numerical simulation*. John Wiley & Sons, 2013. 8

[33] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58 (1):25–36, 2020. 1

[34] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 4, 3

[35] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024. 3, 4

[36] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, pages 6830–6839, 2023. 2, 3

[37] Xiaokun Wang, Yanrui Xu, Sinuo Liu, Bo Ren, Jiri Kosinka, Alexandru C Telea, Jiamin Wang, Chongming Song, Jian Chang, Chenfeng Li, et al. Physics-based fluid simulation in computer graphics: Survey, research trends, and challenges. *Computational Visual Media*, pages 1–56, 2024. 1

[38] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 1, 6, 7

[39] Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024. 8

[40] Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. *arXiv preprint arXiv:2503.16396*, 2025. 4, 8

[41] Hong-Xing Yu, Yang Zheng, Yuan Gao, Yitong Deng, Bo Zhu, and Jiajun Wu. Inferring hybrid neural fluid fields from videos. *NeurIPS*, 36, 2024. 1, 2, 5, 6, 7, 8, 3, 4

[42] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, pages 100–111, 2025. 4, 3

[43] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 8

[44] 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 *CVPR*, pages 1870–1879, 2020. 1, 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 4

[46] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 1

[47] Zhiwei Zhao, Alan Zhao, Minchen Li, and Yixin Hu. Vid2fluid: 3d dynamic fluid assets from single-view videos with generative gaussian splatting. *arXiv preprint arXiv:2503.00868*, 2025. 1

[48] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 8