

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

Supplementary Material

7. Technical Details

7.1. Synthetic Data

To construct the synthetic dataset for our evaluation on novel view synthesis, we obtain a 4D smoke sequence and a static 3D scene from [CGTrader](#). The sequence is placed into the 3D scene and rendered along a prescribed camera path. For the training video, the camera pitch is fixed at $\theta_p = 10^\circ$, and the azimuth is swept linearly from -105° to -45° at a constant speed:

$$\phi(t) = -105^\circ + 60^\circ \cdot \frac{t}{T'}, \quad t \in [0, T'], \quad (6)$$

where $T' = 270$. Rendering is performed with Blender *Cycles* (max samples = 200, i.e., each pixel is estimated by averaging up to 200 light paths to reduce Monte Carlo noise). Representative frames are shown in Figure 8. In the testing video for the novel view synthesis, the camera is fixed at $\phi(t=0) = -105^\circ$ and $\theta_p = 10^\circ$, i.e., the camera pose at the first frame. During inference, we use the pose estimated by DUST3R at the first frame from the training video as the input pose.

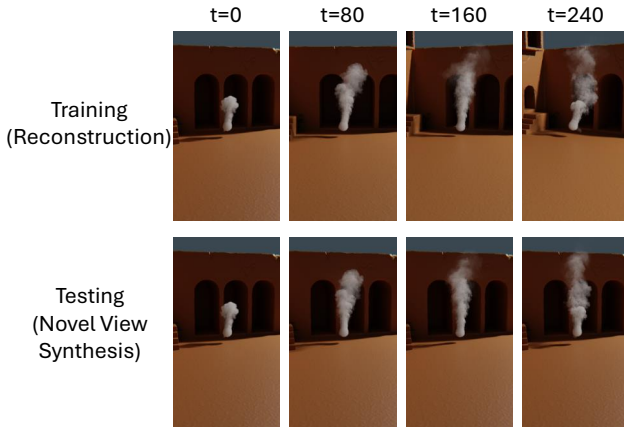


Figure 8. Frames in our synthetic smoke video. When training the reconstruction, the camera view is moving along a trajectory defined by Equation 6. During testing, we evaluate the novel view synthesis with the fixed camera pose from the first frame ($\phi(t=0)$).

7.2. Pose Estimation

We infer per-frame camera poses with a pretrained DUST3R in the one-reference mode, using the first frame as reference. Since DUST3R and Gaussian Splatting (GS) use

different conventions (GS looks along $-Z$), we convert DUST3R camera-to-world poses by *negating* the y and z axes (i.e., multiply both by -1). An example of estimated poses and point cloud is shown in Figure 9. Formally, for each DUST3R-predicted pose \mathbf{C} , we apply

$$\mathbf{C}' = \mathbf{F} \mathbf{C}, \quad \mathbf{F} = \text{diag}(1, -1, -1, 1),$$

where \mathbf{F} flips the y and z axes.

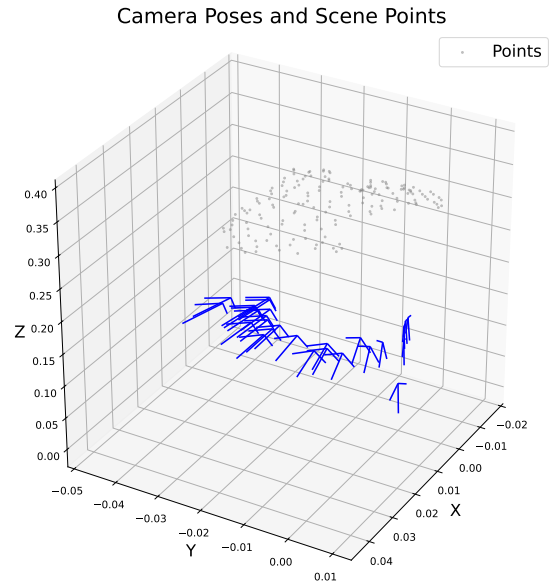


Figure 9. Camera poses and foreground smoke points (for the “valley” video in Figure 6 (middle) from Pixabay) predicted by DUST3R. Blue arrows are DUST3R camera-to-world poses, gray dots are the reconstructed foreground point clouds.

7.3. Particle Initialization

To accelerate convergence, we initialize both physical and visual particles from the foreground point cloud predicted by DUST3R. Let $\mathbf{X} \in \mathbb{R}^{T \times H \times W \times 3}$ denote the 3D points (camera-to-world coordinates) for all frames, where $\mathbf{X}_{t,h,w} \in \mathbb{R}^3$ is the 3D position of pixel (h, w) at time t . We obtain a reliable smoke foreground by intersecting the segmentation masks M_t with the DUST3R confidence mask M_t^{conf} , and collect the foreground set:

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x} = \mathbf{X}_{t,h,w}, M_t(h, w) \wedge M_t^{\text{conf}}(h, w) = 1 \right\}.$$

As DUST3R and Gaussian Splatting (GS) follow different axis conventions (GS looks along $-Z$), we convert point

Table 4. Video resolutions ($W \times H \times T$) and runtime/memory costs of our pipeline. [†]Total time is the sum over all stages; memory reports the peak GPU memory across stages.

Dataset	Synthetic ($1080 \times 1920 \times 270$)		FLAME ($1920 \times 1080 \times 270$)		Pixabay ($1920 \times 1080 \times 270$)	
Stage	Time (GPU-hours)	Mem. (GB)	Time (GPU-hours)	Mem. (GB)	Time (GPU-hours)	Mem. (GB)
Smoke Extraction	0.03	5	0.03	5	0.03	5
Background Removal	0.02	10	2.07	10	0.02	10
Multi-Views	2.35	30	2.56	30	2.55	30
Gaussian Particles	1.05	10	1.18	15	1.77	20
Total [†]	3.45	30	5.84	30	4.37	30

coordinates by *negating* the y and z axes (multiply both by -1). With $\mathbf{F}' = \text{diag}(1, -1, -1)$, we define

$$\tilde{\mathcal{P}} = \left\{ \mathbf{F}' \mathbf{x} \mid \forall \mathbf{x} \in \mathcal{P} \right\}.$$

After that, we downsample $\tilde{\mathcal{P}}$ with a voxel grid, which merges points falling into the same 3D cell into a single representative, to control the initial particle count. We retain 100–300 foreground points per video. The coordinates of the points are then used to initialize our physical and visual particles. After training, the physical and visual particle counts typically grow to approximately 5–10k.

7.4. Generated Videos

SV4D 2.0 takes yaw/pitch in radians, so we recover each frame’s camera pose by applying the generation-time angle offsets to the DUST3R pose, while using the DUST3R-predicted point cloud center as the fixed look-at target. Let the DUST3R pose of the input frame be $\mathbf{P}_0 = [\mathbf{R}_0 \mid \mathbf{t}_0] \in \mathbb{R}^{3 \times 4}$, and let $\mathbf{c} \in \mathbb{R}^3$ be the scene center estimated from the DUST3R point cloud. Given per-frame angle offsets (in radians) $\Delta\phi_t$ (yaw/azimuth) and $\Delta\theta_t$ (pitch), define

$$\mathbf{R}_y(\Delta\phi) = \begin{bmatrix} \cos \Delta\phi & 0 & \sin \Delta\phi \\ 0 & 1 & 0 \\ -\sin \Delta\phi & 0 & \cos \Delta\phi \end{bmatrix},$$

$$\mathbf{R}_x(\Delta\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Delta\theta & -\sin \Delta\theta \\ 0 & \sin \Delta\theta & \cos \Delta\theta \end{bmatrix}.$$

We apply the world-space rotation $\Delta\mathbf{R}_t = \mathbf{R}_y(\Delta\phi_t) \mathbf{R}_x(\Delta\theta_t)$ about pivot \mathbf{c} . The pose of frame t is:

$$\mathbf{R}_t = \Delta\mathbf{R}_t \mathbf{R}_0, \mathbf{t}_t = \Delta\mathbf{R}_t (\mathbf{t}_0 - \mathbf{c}) + \mathbf{c},$$

$$\mathbf{P}_t = [\mathbf{R}_t \mid \mathbf{t}_t].$$

7.5. Down-Weighting of Generated Frames

To reduce the impact of the unreliability over time (see Section 9.3) of generated frames on reconstruction, the loss

of generated frames is multiplied by an exponential decay when training visual particles:

$$w_t = w_{\min} + (1 - w_{\min}) \exp(-k(t - t_0)),$$

where t is the frame index, $t_0 = 0$, $k = 0.02$, $w_{\min} = 0.0$.

7.6. Training Methods

using position-based fluid (PBF) simulation [23, 24] with an incompressibility constraint.

7.6.1. Learnable Buoyancy Strength

In well-controlled scenes [10], the relative strength between buoyancy and gravity is fixed, whereas in the wild, buoyancy strength is underdetermined. To account for this, we make the buoyancy coefficient in the PBF simulation (Appendix B in [10]) learnable and optimize it jointly with reconstruction.

7.6.2. High-Frequency Information Loss

To sharpen fine details, we add a frequency-domain loss during visual-particle training on original input frames only (generated views are excluded due to unreliable high-frequency content). Given an RGB frame I and ground truth \hat{I} , we compute per-channel 2D FFTs $F = \mathcal{F}\{I\}$, $\hat{F} = \mathcal{F}\{\hat{I}\}$, and define amplitude $A = |F|$, $\hat{A} = |\hat{F}|$ and phase $\beta = \angle F$, $\hat{\beta} = \angle \hat{F}$. The loss is computed as the average absolute difference of amplitude and phase over all frequency bins and RGB channels:

$$\mathcal{L}_{\text{freq}} = \text{mean}(|A - \hat{A}|) + \text{mean}(|\beta - \hat{\beta}|).$$

We apply a linear warm-up weight $w_t = \min\left(1, \frac{\text{iter}}{t_{\text{warmup}}}\right)$ and scale by λ_{freq} :

$$\mathcal{L}_{\text{FFT}} = \lambda_{\text{freq}} w_t \mathcal{L}_{\text{freq}},$$

here we set $\lambda_{\text{freq}} = 0.001$.

7.6.3. Total Loss

During physical-particle training, we add a learnable buoyancy coefficient in the PBF simulation (Sec. 7.6.1). During

visual-particle training, we incorporate the high-frequency loss (Sec. 7.6.2). All remaining losses—including photometric, depth consistency, smoothness, and PBF constraints—are kept identical to FluidNexus [10].

7.7. Training and Inference Cost

We report the time and memory costs of our pipeline across all datasets. The corresponding configurations are summarized in Table 4. Even the maximum runtime is significantly lower than that of FluidNexus (which incurs a large training cost due to two generative models) and Hyfluid (which requires about 30 GPU hours to train its three-stage pipeline).

8. Alternative Components

VideoMaMa [21] for Smoke Extraction. We replace the dehazing model with VideoMaMa and additionally refine masks for thick smoke before testing. This setting changes the evaluation ground truth during testing; therefore, the numbers in Table 5 only indicate results under this alternative setup and are not directly comparable to our main results.

Table 5. PSNR comparison (higher is better) of alternative components. NVS = Novel View Synthesis, FP (IV) = Future Prediction (Input View), and FP (NV) = Future Prediction (Novel View).

	FLAME		Pixabay			Synthetic	
		City	Valley	Forest	NVS	FP (IV)	FP (NV)
VideoMaMa [21]	22.23	20.27	21.49	18.07	28.05	22.82	20.72
VGGT [34]	27.80	24.16	26.96	19.43	29.18	24.63	24.28
Ours	22.88	24.68	20.42	17.91	29.78	25.26	25.04

VGGT [34] for Initialization. We test VGGT as a replacement for DUST3R in both point initialization and pose estimation. Table 5 summarizes the experimental results.

Trajectorycrafter [42] for generation. We also test other video-to-video generation models, such as TrajectoryCrafter. However, these models do not explicitly constrain the view center under user-defined camera trajectories, so the camera target often drifts over time. As a result, generated views may gradually shift away from the intended scene center and fail to provide stable multi-view supervision. Representative failure cases are shown in Figure 10. Compared with these alternatives, SV4D 2.0 better matches our requirement of maintaining a stable view center while following custom camera motion.

Original FluidNexus [10] and HyFluid [41]. We evaluate HyFluid [41] and the original FluidNexus [10] as alternative components, but both are less suited to our in-the-wild setting. FluidNexus is mainly validated in controlled lab captures where the gravity direction and buoy-



Figure 10. Failure examples of TrajectoryCrafter under custom camera trajectories. The view center drifts, causing the main smoke region to deviate.

ancy strength are well-defined. In wild-smoke videos, the scene orientation is often unknown and plume dynamics are more diverse. A fixed buoyancy or gravity configuration can therefore produce an incorrect force balance and degrade reconstructions. This motivates our *learnable buoyancy*, which is jointly optimized with the reconstruction. FluidNexus also depends on several scene-dependent hyperparameters, such as domain bounds and validity thresholds. These are easier to specify in a bounded lab volume but hard to define in unconstrained outdoor videos. We address this with a robust *automatic initialization* that provides a reasonable starting volume and scale without manual range tuning. HyFluid uses a NeRF-style monocular reconstruction backbone, which is often less stable on in-the-wild videos. This instability reduces the benefit of our local pose perturbation strategy. HyFluid also requires a costly multi-stage training procedure, which leads to substantially longer runtimes.

9. More Results

9.1. Physical Plausibility

Divergence Metric. To quantify the physical plausibility of the reconstructed motion, we measure the velocity divergence on the predicted Gaussian particles. Given particle positions $\{\mathbf{x}_i\}_{i=1}^N$ and velocities $\{\mathbf{v}_i\}_{i=1}^N$ at each frame, we estimate the local velocity Jacobian by fitting a first-order model within a fixed physical radius r :

$$\mathbf{v}(\mathbf{x}) \approx \mathbf{v}_i + \mathbf{J}_i(\mathbf{x} - \mathbf{x}_i), \quad \mathbf{J}_i = (\mathbf{A}^\top \mathbf{W} \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{W} \mathbf{B}, \quad (7)$$

where rows of \mathbf{A} and \mathbf{B} are $(\mathbf{x}_j - \mathbf{x}_i)^\top$ and $(\mathbf{v}_j - \mathbf{v}_i)^\top$ for neighbors j satisfying $\|\mathbf{x}_j - \mathbf{x}_i\| \leq r$, and \mathbf{W} is a diagonal matrix of Gaussian weights $w_{ij} = \exp(-\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 / (2\sigma^2))$ with σ tied to r . The divergence at particle i is then computed as the trace:

$$(\nabla \cdot \mathbf{v})_i = \text{tr}(\mathbf{J}_i). \quad (8)$$

We report the mean of $|\nabla \cdot \mathbf{v}|$ in Table 6. Our method demonstrates better physical plausibility.

9.2. Smoke Extraction

Extracted Smoke Examples. Figure 11 shows representative results of our smoke extraction across several in-the-wild videos. For light smoke, we apply dehazing to remove

Table 6. Physical plausibility measured by velocity divergence on reconstructed particles. Lower is better.

	FLAME	City	Valley	Forest	Synthetic
FluidNexus	1.611	1.454	0.845	0.795	1.381
Ours	0.866	0.635	0.784	0.751	0.740

background leakage; for dense smoke, we use the masked result directly as the clean foreground.

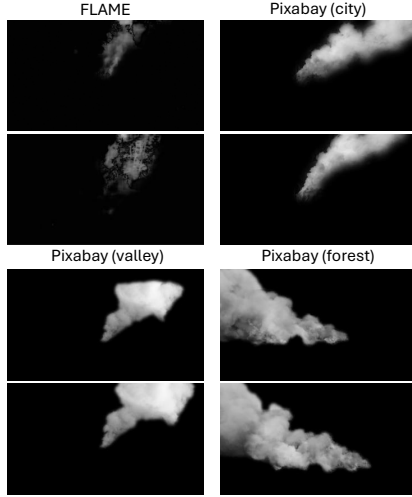


Figure 11. Visualizations of extracted smoke from wild videos.

Results without Smoke Extraction. We further train HyFluid [41] and FluidNexus [10] on inputs without our smoke extraction method. The results are shown in Table 7.

Table 7. Comparing PSNR (higher is better) of methods with/without our smoke extraction (SE) method.

	Future Prediction (Input View)			
	FLAME	City	Valley	Forest
HyFluid	10.37	12.37	11.79	9.88
HyFluid (w. our SE)	21.67	23.24	12.43	13.70
FluidNexus	6.67	8.78	9.63	4.46
FluidNexus (w. our SE)	21.78	24.18	16.44	14.63
Ours	22.88	24.68	20.42	17.91

9.3. Generated Videos from SV4D 2.0

We previously noted that the quality of generated multi-view frames from SV4D 2.0 decays over time. Figure 12 shows early and late frames of the sequence. As time progresses, the smoke structure gradually collapses.

9.4. SSIM and LPIPS

Beyond PSNR, we further report the structural similarity index measure (SSIM) in Table 8 and the perceptual metric LPIPS [45] in Table 9. Our method is still shown to perform better based on these two additional metrics.

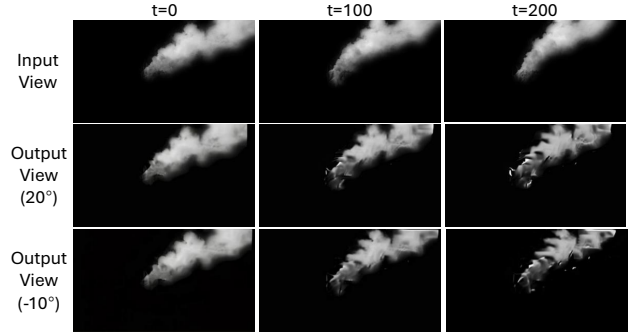


Figure 12. SV4D 2.0 multi-view generation over time. Later frames exhibit structural collapse.

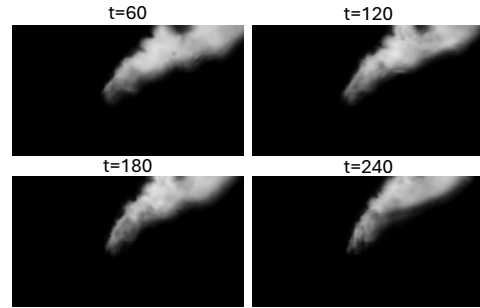


Figure 13. Novel view synthesis on wild videos. Our reconstructions preserve stable smoke structure across views.

Table 8. Comparing SSIM (higher is better) of smoke reconstruction by different methods on videos collected from FLAME [30] and Pixabay.

	Future Prediction (Input View)			
	FLAME	City	Valley	Forest
HyFluid [41]	0.87	0.55	0.79	0.46
FluidNexus [10]	0.88	0.84	0.76	0.70
Ours	0.92	0.89	0.86	0.80

Table 9. Comparing LPIPS (smaller is better) of smoke reconstruction by different methods on videos collected from FLAME [30] and Pixabay.

	Future Prediction (Input View)			
	FLAME	City	Valley	Forest
HyFluid [41]	0.13	0.33	0.18	0.36
FluidNexus [10]	0.16	0.14	0.20	0.31
Ours	0.09	0.12	0.13	0.24

9.5. Visualizations of Novel View Synthesis on Wild Videos

Figure 13 renders novel views for several wild videos. For all frames, the novel-view camera is fixed to the pose of the first frame of the training video. The reconstructed smoke remains stable over time.

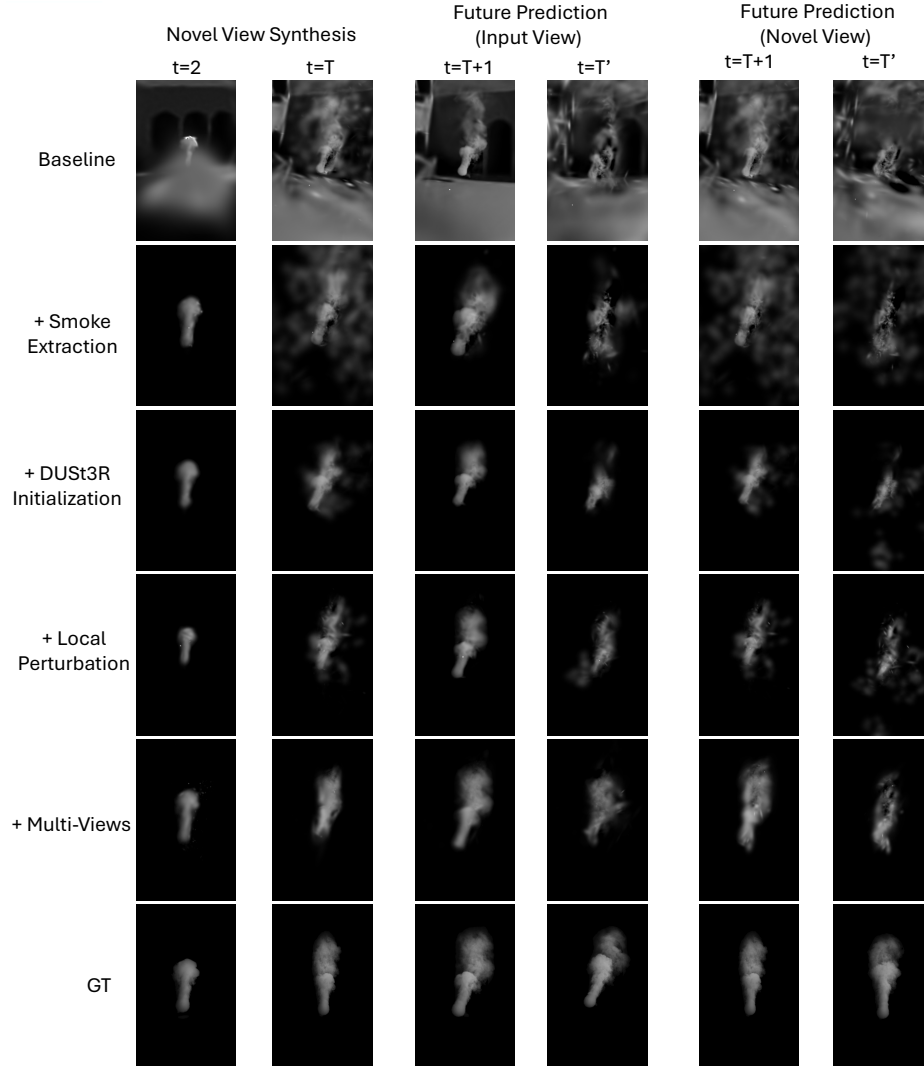


Figure 14. Ablation study of novel view synthesis and future predictions on synthetic smoke videos. *Novel view* uses the camera pose at $t = 1$, and *input view* means the camera poses along the source video. Training uses $T = 240$ frames, and the unseen future extends to $T' = 270$ frames. *GT*: Ground Truth.

9.6. Visualizations for Ablation Study on Synthetic Data

We provide additional qualitative ablations (Figure 14) complementing the quantitative table in the main paper.



Figure 15. Failure cases of pretrained dehazing model on light outdoor smoke. From left to right: input image, output of the pretrained model (fails to remove light smoke), and output of the finetuned model (significantly clearer background, enabling smoke extraction).

9.7. Failed Examples Using Pretrained Dehazing Model

We also test the extraction performance of the pretrained dehazing model [31]. As shown in Figure 15, the pretrained model fails to remove the light and spatially thin smoke, leaving most of the haze untouched. This prevents us from separating the smoke component using simple priors. In contrast, the finetuned dehazing model successfully removes most of the smoke and reveals a clearer background, which subsequently enables reliable extraction of the smoke layer using the dark channel prior.

9.8. Human Validation

We conduct a perceptual study to evaluate GIF visualizations of reconstructed smoke dynamics. We collect 14 re-

sponses via an online survey. For each question, participants are shown a target GIF and anonymized candidate results (labeled as A/B/C), and are asked to *choose the result closest to the target* or select *Not sure*. We evaluate 7 settings: four in-the-wild sequences (FLAME, Forest, City, Valley) and three synthetic settings (Novel View Synthesis (NVS), Future Prediction in Input View (FP (IV)), and Future Prediction in Novel View (FP (NV))).

We report the percentage of votes for each method in Table 10. As shown in the table, our method is preferred in all evaluated settings.

Table 10. Human preference (% , higher is better) on GIF visual-quality survey (14 respondents). Participants selected the result closest to the target.

	In-the-wild				Synthetic		
	Forest	City	FLAME	Valley	FP (IV)	NVS	FP (NV)
HyFluid	0.0	0.0	7.1	0.0	0.0	0.0	0.0
FluidNexus	0.0	7.1	0.0	0.0	7.1	14.3	0.0
Ours	100.0	92.9	86.7	100.0	85.7	71.4	71.4
Not sure	0.0	0.0	7.1	0.0	7.1	14.3	28.6