DOI: 10.1111/cgf.70300

COMPUTER GRAPHICS *forum*
Volume 0 (2025), number 0, e70300

Fluid Composer: Fluid Detail Composition and Rendering Using Video Diffusion Models

Duowen Chen,[1] Zhiqiang Lao,[2] Yu Guo[2] and Heather Yu[2]

[1]School of Computer Science, Georgia Institute of Technology, Atlanta, USA
dchen322@gatech.edu
[2]IC Lab, Futurewei Technology, Basking Ridge, NJ, USA
{zlao, yguo1, hyu}@futurewei.com

Abstract

We introduce a hybrid pipeline that combines classical fluid simulation with modern generative video models to produce high-quality, controllable fluid effects without implementationally difficult solvers or costly ray-tracing. First, a lightweight physics-based simulator enforces core properties like incompressibility and lets artists specify layout, boundary conditions, and source positions. Second, we render a simple 'control video' via real-time rasterisation (diffuse shading, masks, depth) to capture scene structure and material regions. Third, a text-guided diffusion transformer (e.g., VACE) treats this control video as a canvas, refining it by adding foam, bubbles, splashes, and realistic colour blending for multiple materials. Our method leverages pre-trained video generators' implicit physical priors, while masking and noise-warping ensure precise, per-material control and seamless mixtures in latent space. Compared to purely simulation-based or generative model based text-only approaches, we avoid implementing specialised multiphase algorithms and expensive rendering passes, yet retain full artistic control over fluid behaviour and appearance. We demonstrate that this training-free strategy delivers photorealistic fluid videos, supports diverse effects (multiphase flows, transparent media and wet foams), and simplifies the artist's workflow by unifying simulation, shading, and generative rendering in a single, extensible framework.

Keywords: controllable video generation with generative models, physically grounded fluid rendering, physics-guided fluid prior modeling

CCS Concepts: • Computing methodologies → Physical simulation; Computer graphics; Machine learning

1. Introduction

Various fluid phenomena have been studied in physics simulation since the introduction of Stable Fluids [Sta99], and realistic fluid effects are now ubiquitous in film and game production. Achieving visually compelling results depends both on sophisticated simulation methods for physical accuracy and on rapidly evolving rendering techniques to convert simulated data into photorealistic imagery. To capture specific behaviours—such as foam, fluid mixtures and splashes—specialised simulation algorithms must be incorporated [WFS22, RLY*14, PAKF13]. In many cases, sparse or adaptive grids [AGL*17, XCW*20], narrowband techniques [SWT*18, FAW*16] and GPU implementations are necessary to reach the desired quality, but these optimisations are difficult to implement and reproduce. Beyond simulation, rendering effects like whitewater and underwater bubbles further increase the pipeline's complexity and computational cost. Inspired by recent advances in generative models, we propose a nontraditional approach to alleviate these burdens.

Recent generative video models have achieved unprecedented quality [WWA*25, Gen24, YTZ*25, Ope24, Kua06]. Fluid synthesis, in particular, remains a challenging benchmark for physical accuracy [ZXM*25]. Current approaches either rely on large language models to inject physics priors—assuming that textual guidance can enforce correct behaviour [ZXM*25] or use 2D optical-flow techniques in latent noise space [MSAA*24], which fail to capture fluid's inherently 3D nature. Moreover, existing generative frameworks lack the layout-driven controllability required by graphics-based fluid pipelines: specifying boundary conditions, source positions, and scene layout is difficult to express and control via human language alone. At the same time, massive training datasets embed

Figure 1: *We introduce Fluid Composer, a novel fluid detail composer and renderer that transforms greyscale shaded videos from a basic fluid simulator into rich fluid effects like wet foam, multi-material mixtures, and transparent water splashes, while automatically applying solid textures, materials, and lighting inferred from textual descriptions.*

both realistic appearance and a degree of physical plausibility in models like DiTs [PX23], which we can leverage.

Therefore, we propose using a video generator as a 'fluid detail composer' and 'appearance renderer.' The pipeline proceeds as follows: **(1) Physics-based simulation**. A base fluid simulator (e.g., Stable Fluids or PIC/FLIP) enforces core fluid properties—such as incompressibility—and gives artists precise control over scene layout (Section 3.1); **(2) Real-time shading and control estimation**. We render a lightweight 'control video' of the simulation, consisting of simple diffuse shading, using a real-time shader (Section 3.2); **(3) Generative detail and material editing**. A language-guided video generator refines the control video by adding fluid details (foam, bubbles, mixtures), assigning appropriate materials and colours (including multiple fluid types or solid objects), and producing a final photorealistic sequence (Section 3.3).

By combining a classical simulator with a generative model, we preserve the traditional graphics workflow, scene layout to simulation to rendering, while offloading complex phenomena (multiphase flow, mixture models, etc.) and rendering (skins, foams, etc.) to the generative stage. This approach provides artists with full control over the simulation's physics and scene layout, yet avoids implementing specialised solvers for every fluid effect and saves timing issues raised in raytracing.

In summary, by taking advantage of existing video-generation models with controls (e.g., VACE [JHM*25]), we propose a training-free approach that enforces physics priors as shaded video from a basic fluid simulation, while giving artists full control over scene generation. Our method enables easy fluid detail creation and material editing. By using masking and noise warping, it supports multiple material composition in a single pass.

We summarise our contributions as follows:

1. A unified framework for simple yet high-quality fluid-simulation generation and visualisation using video diffusion.
2. A practical way to enforce physics priors for fluids in a video-generation pipeline.
3. A fast fluid-rendering pipeline with modified controllable video-generation model.
4. A realistic material and fluid-detail composition method via a diffusion framework.

2. Related Work

2.1. Video Generation and Control

Video Generation. Diffusion-based transformers have rapidly advanced text-to-video synthesis. Early closed-source systems like OpenAI's Sora [Ope24] kicked off the era of high-quality, professional video generation, followed by public previews of Kling [Kua06], Luma 1.0 [Lum06], Gen-3 [Run06], Vidu [BXY*24], Pika 1.5 [Pik10], Movie Gen [PZB*] and Veo 2 [Dee12] by year's end, each pushing frame-level fidelity, physics realism, and support for multimodal prompts. Concurrently, the open-source community has built on latent diffusion and transformer blocks to scale video models [Gen24, KTZ*24, HCB*24, ZPY*24, LGC*24, JSL*24, YTZ*25]. U-Net [RFB15] extensions such as VDM [HSG*22] introduce full 3D convolutions, while 1D temporal with 2D spatial attention [ZWY*22, WYC*23, GYR*24] hybrids reducing computations. Diffusion Transformers (DiTs) [PX23] replace all convolutional structure with pure transformer layers, yielding superior image fidelity [CYG*23] and easily transferring to video original DiTs [PX23, HCB*24] and MM-DiT's multimodal concatenation of text embeddings [Gen24, KTZ*24]. Spatio-temporal VAEs [HCB*24, WWA*25] then compress video into compact latents, enabling billion-scale model pre-training on trillions of tokens and real-time inference on consumer GPUs. Subsequent models

such as LTX-Video [HCB*24] scaled these approaches to real-time, high-resolution generation, while open-source variants Mochi [Gen24], CogVideoX [YTZ*25], Hunyuan [KTZ*24] and Wan 2.1 [WWA*25] have further closed the gap with proprietary systems. Despite these gains in visual fidelity and realism, most of these models rely solely on text or image conditioning and offer limited mechanisms for precise, physics-aware control of emerging content.

Controlling and Edition. Video diffusion models have been extended with a variety of conditional controls to guide spatial structure, appearance, and motion. One group of approach includes pixel-aligned signals—depth, pose, scribble, optical flow for video-to-video editing tasks [ZRA23, ZWJ*23, HX23, WYZ*23, HCL*23, JMP*24]. Along this track, task unified visual control over generation model was proposed for images [QZY*23, HJP*24, XWZ*25] and, recently extended to video generations [JHM*25]. Beyond visual signal control, inloop physics simulator for video generation are explored on both rigid bodies [LRGW24], softbodies [CJL*25], clothes [XZJJ25] and more general cases [MSAA*24, GYL*25]. Such method usually starts from an image, and apply simulation methods like PBD [MMC16] guide the video generation from the input image but mostly only limited to simple physics behaviours. More recent work [LYL*25, GHF*25] extends these to more diversified physics phenomenons.

2.2. Fluid Simulation and Rendering

Basic Fluid Simulation. Since the introduction of Hamholz decomposition applied on the Naive Stokes equation computation in Stable Fluid [Sta99] and its extention to air–liquid interface [EFFM02] through the introduction of particle level set (PLS), numerous methods in this field are being explored. We shall mainly focusing on reviewing free-surface-related methods as those being most related. Existing fluid simulation methods usually categorised to pure Eulerian methods [Sta99, LCPF12, EMF02, IGLF06], pure Lagrangian methods [BT07, MCG03, SP09, DGWH*15, LBC*24] and Hybrid methods [ZB05, HHK08, RWT11, AT11, JSS*15, FGW*21, FAW*16]. Most of the time, surface tension driven phenomenons like milk crown tends to favour pure Eulerian methods due to the accurate level set value computation [ZZKF15] while splashes tends to favour hybrid methods due to the less damping during advection. As for the divergence-free condition for fluid, current methods either use weakly compressible enforcement for real time performance [MCG03, BT07], or applies multigrid Poisson solver for pressure projection [MST10]. Although the basic fluid simulator is rather simple and clean for implementation, special effects in fluid usually require special treatment and are usually hard to achieve and render.

Special Fluid Effects. Special fluid effects, depending on fluid types, vary. In the scope of this study, we focus on free-surface fluids. For free surface fluid, the most interesting effects include foams like beer foams and wet foams, bubbles and mixture models. For foams, Losasso *et al.*[LTKF08] coupled SPH with PLS for simulating foams over ocean waves. Ihmsen *et al.* and Bender *et al.*[IAAT12, BKKW18] introduced procedual wet foam generations using SPH methods. The current state-of-art for wet-foam simulation comes from [SWBD20, WFS22] combining FLIP with a SPH simultion of foams. For bubbles, Thürey *et al.*[TSS*07, KSK10] used an ad-hoc methods to simulate bubbles under water. Kim *et al.*[KLL*07] used two-phase flow for more accurate bubble behaviours. Patkar *et al.*[PAKF13] considered the state equation for compressiblity of bubbles. Besides these, Li *et al.*[LMLD22, LD23] tried to solve two-phase flow using LBM methods and reaches astonishing result. For mixtures, Ren *et al.*[RLY*14] used multiphase SPH for fluid mixtures and [LSSF06] applies on grid-based methods. Those methods, though physically based or intended, all require special treatment and implementation and are only partially unified in some non-opensource commercial software like Houdini or WetaFX [LSD*22].

3. Overall Pipeline

We leverage diffusion models to produce complex fluid phenomena while remaining faithful to the classical graphics pipeline, preserving artists' creative freedom and control, and we rely on a lightweight fluid solver solely to enforce the core physical behaviour. Our method is divided into the following components (see Figure 2): (1) Fluid Simulator as Control and Physics Prior; (2) Graphics Shader, Masking, and Depth/Scribble Estimation; (3) Controlled Diffusion Video Generator as Detail Creator, Material Synthesiser, and Renderer. Below, in each individual subsection, we describe each component in detail.

3.1. Fluid Simulator

We implemented a fluid simulator based on the APIC/AFLIP [JSS*15, FGW*21] method in Taichi [HLA*19], enabling fast GPU computation across all scenarios presented. To enforce a divergence-free velocity field and improve volume preservation, we implemented a GPU-accelerated MGPCG solver for the Poisson equation. For free-surface pressure projection, we employ the standard ghost-cell [GFCK02] treatment and, for simplicity, omit both surface tension and viscosity. We support Dirichlet and Neumann boundary conditions to accommodate a variety of use cases. Simulation outputs are represented as meshes obtained via marching cubes on the grid-stored level set, using the particle union technique proposed in [FF01, BB12], which dramatically reduces storage and computational overhead compared to handling millions of particles. Because our fluid solver serves solely as the physics controller for basic fluid motion—volume preservation, simple surface waves, and splashes—any other solver may be integrated into the pipeline effortlessly.

3.2. Graphics Shader

Real-time rasterisation offers an efficient way to preview view direction and camera motion, even though it does not match the photorealism of ray tracing. Rasterisation also provides native support for transparency and masking throughout the graphics pipeline. We exploit transparency from fluid and solid elements, solid–fluid masks, and fluid–material masks to exert fine-grained control over material interactions. By default, we use Blender's viewport renderer with its standard material settings: the resulting shaded videos convey only shape information and serve as the basis for subsequent

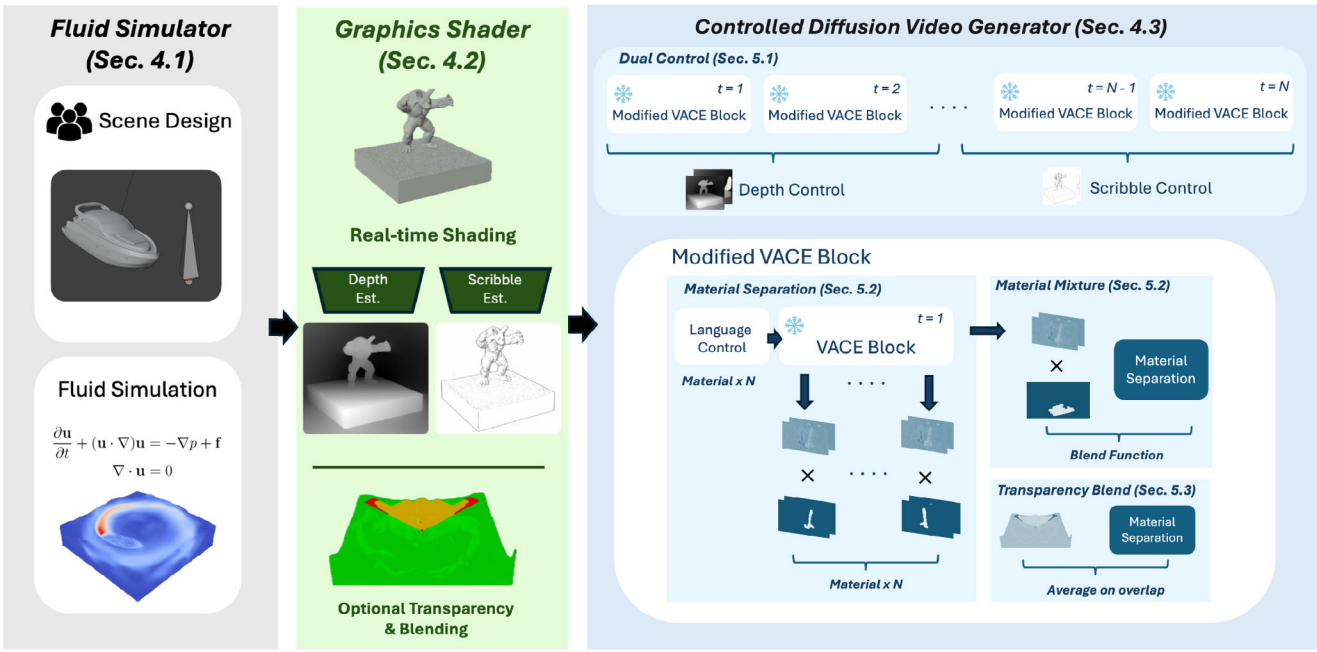

Figure 2: *Overview of the proposed pipeline. (Left) Artists specify scene layout–initial, boundary, and source conditions—which are simulated by a naive free-surface fluid solver to produce physics priors. (Centre) The simulated volume is converted to a 3D mesh and shaded into a control video; depth and scribble maps are extracted and our transparency-blending mask is applied. (Right) These control signals, together with textual descriptions of fluid and solid materials, are fed into the modified VACE diffusion block, which supports multi-material mixing and generates the final photorealistic fluid video without any changes to the underlying simulator.*

depth and scribble estimation. For depth prediction, we employ the method of [RLH*20], and for generating scribble-style art, we follow [CDI22]. The produced depth maps and scribble videos then act as control inputs to the VACE video-generation model. Here, the two control videos can also be generated using graphics softwares such as Blender or Houdini, but to reduce the dependencies on the prior knowledge of using such softwares, we choose to stick with the pipeline from VACE.

3.3. Controlled Diffusion Video Generator

Our controlled video generator builds upon the WAN video generator [WWA*25] and the controllability framework of VACE [JHM*25]. VACE accepts depth maps or scribble pictures as control signals, together with a text prompt, to generate a video. However, we find this approach inadequate for fluid-centric tasks. First, depth estimation typically omits the fine-scale motion and surface geometry of fluids. As shown in Figure 4, using only depth control fails to reconstruct the Armadillo's mesh details, while the generator hallucinates fluid details in regions of ambiguous depth—often misaligned with the artist-created shaded video. Scribble-based control, by contrast, introduces excessive noise in most cases. Second, without explicit material segmentation, relying solely on language prompts produces unrealistic outputs even in simple scenarios (Figure 3). In more complex cases—such as fluid mixtures—masks are required to specify material blending. Finally, effects like transparency cannot be conveyed through text alone. To overcome these limitations, we introduce extensions to the VACE pipeline—described in the next section—that enable precise control over fluid

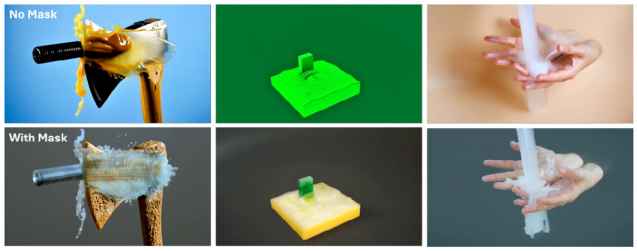

Figure 3: *We show without the material separation mask, video generation models cannot accurately capture the textual description of materials.*

appearance, material separations and mixtures, and transparency, thereby supporting a wide range of desired fluid effects.

4. Control Signals

As we mentioned in previous section, three major concerns need to be addressed specifically for fluid targeted generation within the VACE model. We shall elaborate each of them.

4.1. Dual Control for Generation

To faithfully replace traditional fluid rendering, a central challenge is accurately capturing the myriad of small water droplets and fine mesh details produced by fluid simulations. As shown in Figure 4,

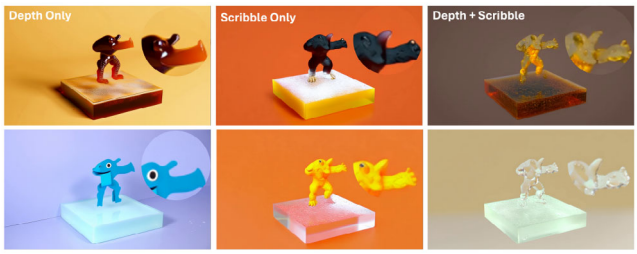

Figure 4: *We show without our dual control for both depth and scribble, either the shape of armadillo is not faithfully preserved or the material is generated incorrectly.*

depth estimation alone cannot recover these fine-scale features. Conversely, scribble-based control often introduces spurious edge noise in fluid regions, resulting in a grainy appearance. Motivated by the observation that diffusion models tend to generate coarse structure and colour in early denoising steps and finer details in later steps [SHWC25], we propose to use depth control in early stages for colouring and general shape control and taking advantage of the detail control of scribble signals in later stages. In such way, grainy feelings of scribble-only control are mostly removed due to pre-determined colouring from early stage depth control generation. Meanwhile, the later stage scribble control adds the details neglected from depth-only signals. In actual implementation, we use the first 20% of timesteps with depth control and later 80% with scribble controls. We refer readers to Section 6.5 for ablation study on the choice of the parameters.

4.2. Material Separations and Mixtures

Video diffusion models still adapts attention mechanisms. Text prompts are first encoded by a CLIP encoder [RKH*21] and then injected into the DiT backbones via cross-attention. However, when a single prompt references multiple materials, the resulting conditioning can become ambiguous (see Figure 3). To address this, we supply distinct text embeddings for each material. Concretely, for material i we extract a binary mask $\mathcal{M}_i \in \{0,1\}^{T_F \times H_F \times W_F}$ in the shader domain and downsample it to the VACE model's latent dimensions $T_L \times C_L \times H_L \times W_L$, where the subscript $*_F$ denotes video-domain dimensions (frames T_F, height H_F, width W_F) and $*_L$ denotes latent-space dimensions (frames T_L, channels C_L, height H_L, width W_L). Let $\mathcal{I}_t$ be the noisy latent image at diffusion step t. We then apply the following modifications for material separation and mixture handling.

Material Separation. At diffusion step t, we compute the material-wise separated update by

$$\mathcal{I}_{t+1}^{\text{separation}} = \sum_i \mathcal{M}_i \otimes \mathcal{D}_t(\mathcal{G}_u(\mathcal{I}_t) + \beta \mathcal{G}_{c_i}(\mathcal{I}_t)) \tag{1}$$

where $\mathcal{I}_t$ is the noisy latent image at step t, $\mathcal{G}_u$ and $\mathcal{G}_{c_i}$ are the unconditional and context-conditional generators, β is the classifier-free guidance (CFG) scale, $\mathcal{D}_t$ denotes the denoising operator, and $\mathcal{M}_i$ is the downsampled binary mask for material i. Here, CFG [HS22] follows from most of the recent DiT models [KTZ*24, WWA*25, HCB*24] where guidance from prompts are enforced. And the strength of guidance is controlled by β. The element-wise product $\otimes$ ensures that only region i is denoised using its own textual embedding c_i. For pixels where multiple masks overlap, we choose one material's mask and set the others to zero during this separation pass.

Material Mixture. We denote the textual embedding for the mixed material as c_{mix}. Let the mixture injection begin at diffusion step t_s within a total of t_{total} steps. We define the latent update for the mixture region by

$$\mathcal{I}_{t+1}^{\text{mix}} = \mathcal{M}_{\text{mix}} \otimes \Big(u_t \mathcal{D}_t(\mathcal{G}_u(\mathcal{I}_t) + \beta \mathcal{G}_{c_{\text{mix}}}(\mathcal{I}_t)) + (1-u_t)\mathcal{I}_{i+1}^{\text{separation}} \Big) + (1 - \mathcal{M}_{\text{mix}}) \otimes \mathcal{I}_{i+1}^{\text{separation}} \tag{2}$$

where the interpolation weight

$$u_t = \left(\frac{t - t_s}{t_{\text{total}} - t_s - 1} \right)^{\gamma} \tag{3}$$

controls the blend speed via the exponent γ and t_s. Although the masks are combined in a discrete fashion, this simple schedule produces coherent mixtures. All regions start from pure Gaussian noise, which conceals seam artifacts, and each denoising step $\mathcal{D}_{t+1}$ attends to the full latent image, naturally enforcing consistency across material boundaries. The composite mask $\mathcal{M}_{\text{mix}}$ is obtained by aggregating each material's binary mask over all frames: for every spatial location, $\mathcal{M}_{\text{mix}}$ is set to 1 only if all individual material masks have been active at least once at that location during the video, preventing unrealistic colour shifts in areas that contains only single materials (see Section 6.5).

4.3. Transparency of Meshes

Transparent materials require special handling in fluid-oriented generation. While uncoloured transparency—such as in water tanks or clear wine glasses—can be rendered directly via standard rasterisation, coloured media like tinted glass or beverages (e.g., beer, wine) demand explicit guidance. Rather than expecting the diffusion model to infer accurate colour blending from text alone, we instruct the model to use the blended colouring of the transparency material and other materials. The key distinction between transparent and opaque-material scenarios is that transparency permits multiple materials to coexist at a given pixel, causing the sum of their mask values to exceed one. To handle such overlaps, we extend our mixture strategy: wherever n material masks overlap, we replace the individually separated outputs with their simple average in latent space, that is, $\frac{1}{n}\mathcal{I}_{t+1}^{\text{separation}}$, thereby ensuring a cohesive blend of all contributing materials.

4.4. Material Assignment

Now, we have everything in place, the material for each mask is directly assigned through language guidance. The material property will directly be included in the prompt guidance and each mask need to be attached with one prompt. As an example, for Figure 5, the prompt for the two masks will be 'marble textured board' and 'dark caramel coloured coke cola' and the overall consistency for

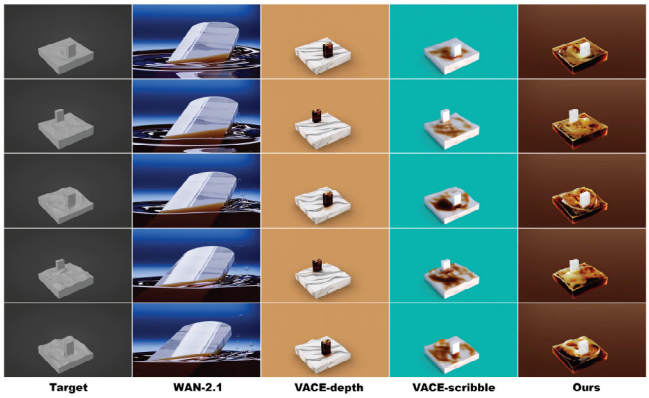

Figure 5: *Text prompt: 'A marble textured board sliding in a tank of dark caramel coloured coke cola. The video should have a single coloured background.' We see other methods either fail to preserve the shape of the objects in the target video or cannot accurately capture the material of fluid and board.*

the video is given by the entire prompt 'marble textured board sliding in a tank of dark caramel coloured coke cola.' The video should have a single coloured background.

In summary, we introduce three targeted enhancements, that is, dual control, material blending and transparency blending, to the VACE framework to support realistic fluid effects; our quantitative and qualitative results are presented in the subsequent sections together with ablation studies on the choices we made described above.

5. Implementation Details

Fluid Simulation. All of our simultions are performed under the resolution of 128 with CFL number set to 0.5. For cases where more splashes are desired, for example, the Cruiser Scene in Figure 1, we adopt AFLIP as proposed in [FGW*21]. Otherwise, APIC [JSS*15] is used for simulation. We conduct all simulations on a Nvidia RTX 5090 GPU and it takes at most 20–30 min of simulation time for generating the simulation data for the videos we present.

Video Generation. All video generations are performed using a single Nvidia H100 GPU where the base model set to be WAN-14B DiT model with 20 denoising steps to balance the speed of generation and output video quality, as we did not observe obvious worsen quality of generated video compared with using 50 denoising steps with our method. For each video shown in this paper, the diffusion process takes 10 min to complete. In comparison experiments, WAN videos are generated with prompt enhancement being turned on, while VACE experiments are turned off by default. All comparison experiments use the same input prompt and materials across the four methods being compared.

6. Experiments

We evaluate our method both qualitatively and quantitatively under various scenarios. We mainly focus on seven daily fluid materials and various kinds of solid materials. We conducted 20 base fluid simulations with diverse initial and boundary conditions. With the 20 base fluid simulations, 176 different videos are generated with 88 different kinds of fluid and solid combinations. Some examples of fluid material include wine, water, beer, cola, chocolate and milk. Solid materials are more diverse with allowing DiTs to assign

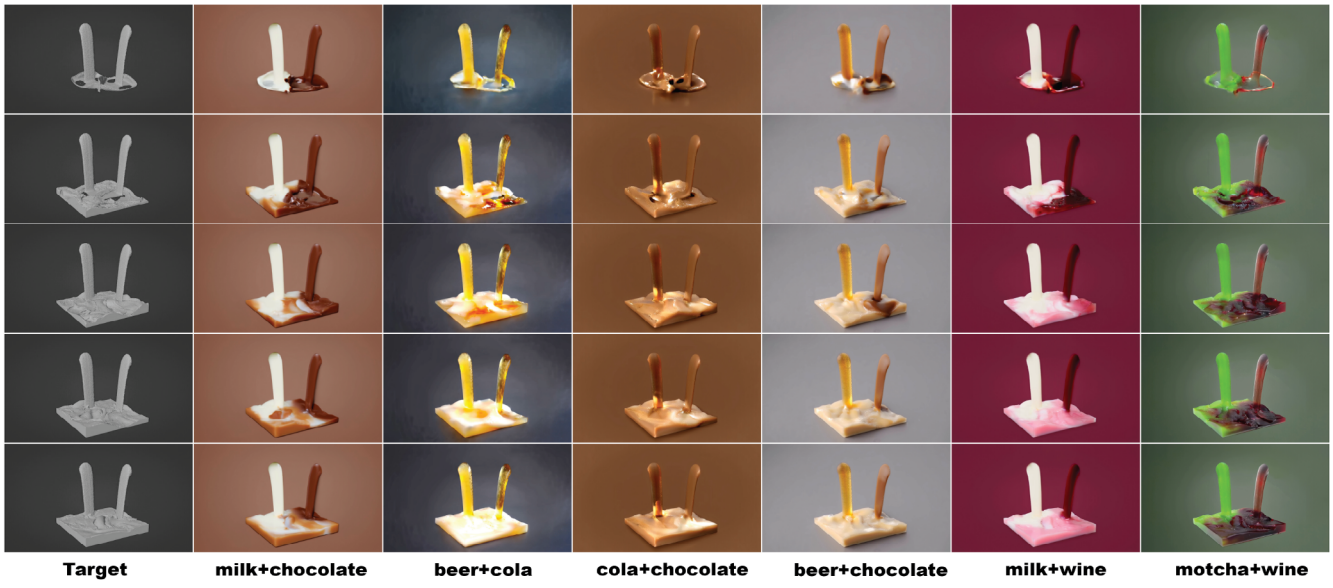

Figure 6: *In this figure, we show our method is capable of handling a varity of material blending and composing details of fluid like foams inferred from fluid material. All of the videos are generated using the prompt: 'MATERIAL1 and MATERIAL2 poured down into an empty squared tank and mixed together. The final colouring should be the mix of the two materials.' where MATERIAL1 and MATERIAL2 are the two fluid materials being mixed here.*

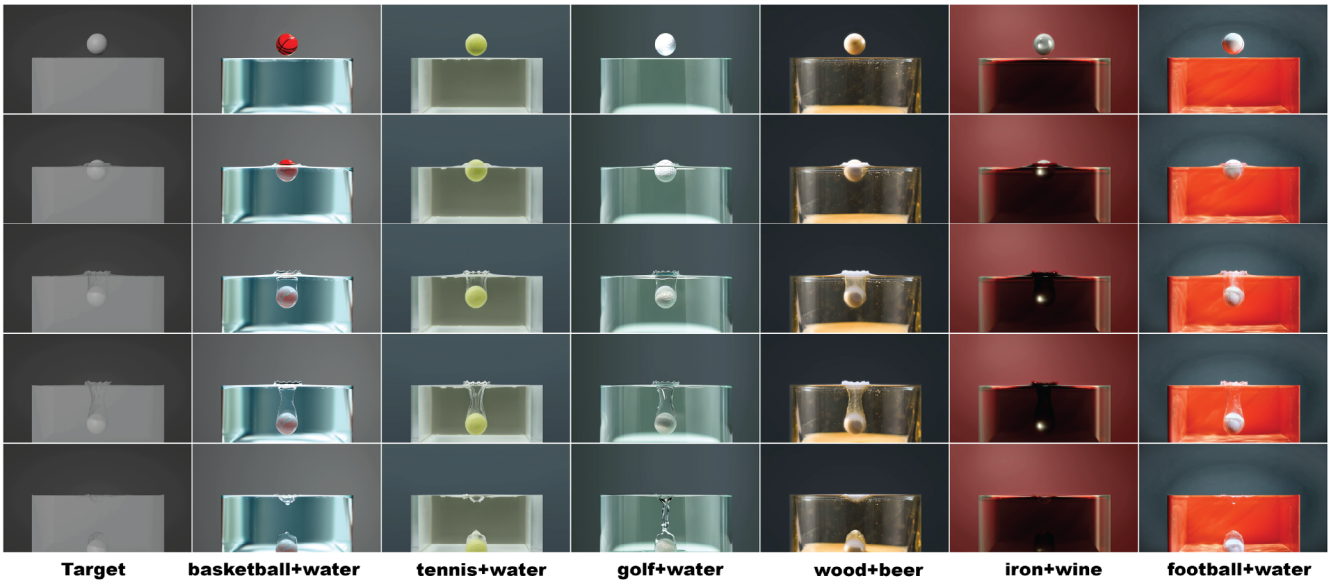

Figure 7: *We show by providing the shaded video on the left using a simple APIC fluid simulation without any two-phase flow techniques, our model and compose under fluid bubbles based on text prompts and applies different solid materials with a single word change.*

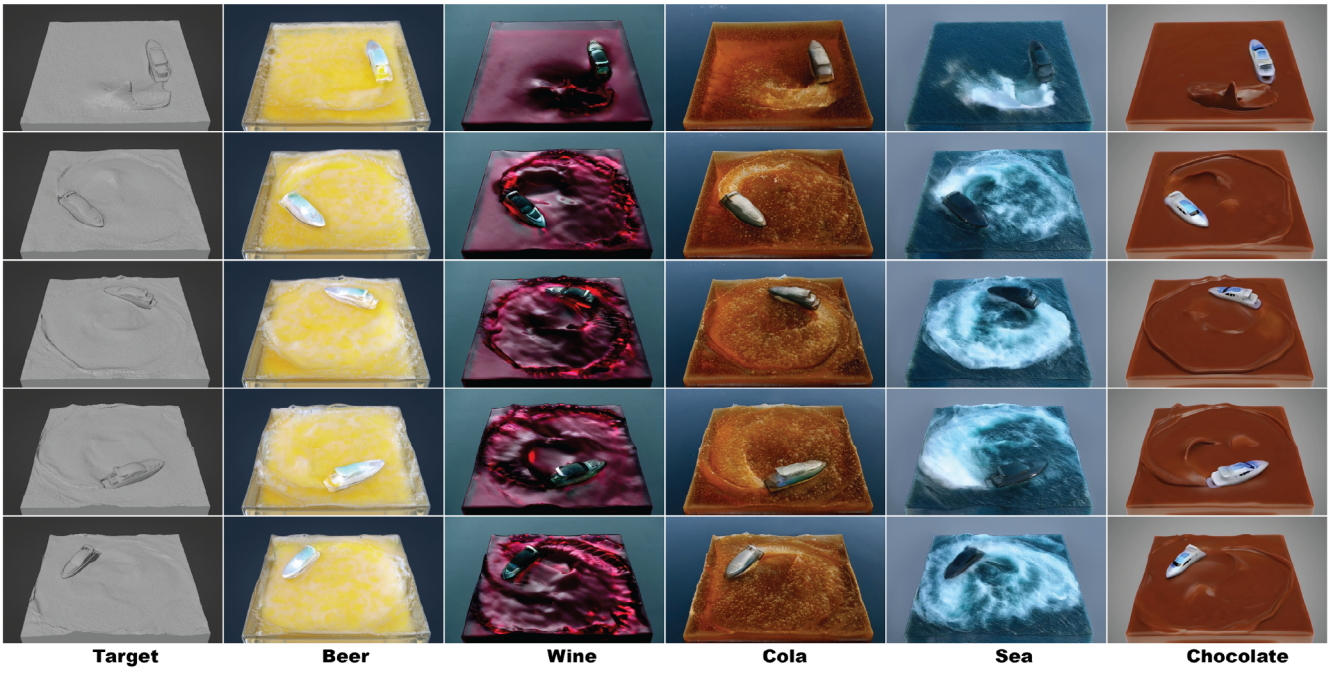

Figure 8: *We show the scene of a ship moving in a circle on different fluid materials. Wet-foams and bubbles automatically appear from the video generation pipeline. All of the videos are generated using the prompt: 'An iron-built ship moving on the surface of a tank of MATERIAL. The ship is moving in a circle trajectory. Focus on the trajectory and the surface details due to its motion. The video should have a single coloured background.' where MATERIAL being different fluid materials.*

material given object description. For example, B-2 Bomber, Rocky Mountain, Fountain made of jade and etc. In Figures 1, 6–8 and 9, we show some examples of our generation. In Figures 5,10–12, we show some examples of our qualitive comparisons. We refer readers to our Supporting Information for more results for our generation.

Among such videos, we categorise them into (1) pure fluid videos where only a single fluid type appeared; (2) solid and fluid interaction videos where static and dynamic solids are involved in the simulation; (3) fluid mixture videos where different fluid types appear in the same video and create a mixture through the generation pipeline.

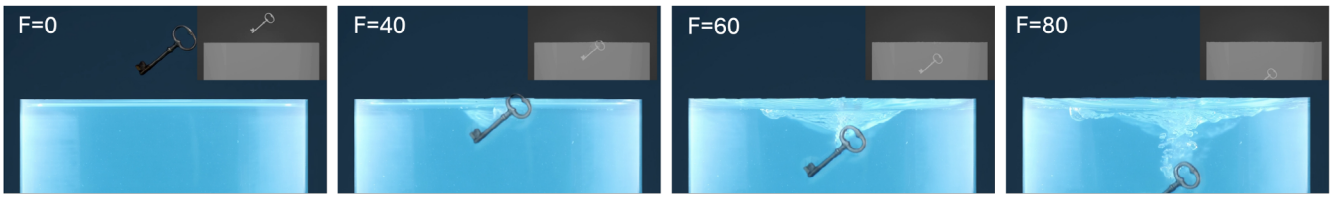

Figure 9: *We show our method is capable of composing underwater bubbles similar to the result of simulated two-phase flow as shown in [LD23]. The grey coloured picture is the shaded result where none of the underwater effect can be observed initially.*

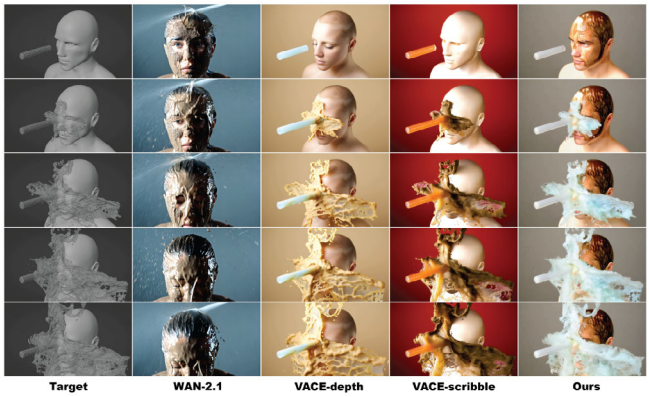

Figure 10: *Text prompt: 'Water jet on a human face covered by dirty mud creating splashes and turbulant fluid motion. The video should have a single colored background.' We see only our method faithfully recovers the target video while aligned with the textual description where mud is coving face.*

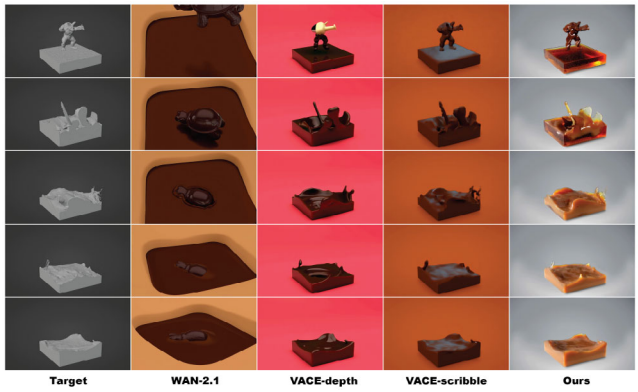

Figure 11: *Text Prompt: 'An Armadillo made of dark brown coloured hot chocolate drop into a squared tank of dark caramel coloured coke cola. The two materials blend together.' Our method is capable of accurately capture the initial material separation between cola and chocolate and their mixture in later frames while other methods fails to achieve such result.*

6.1. Human Evaluations

We anonymously collected 70 responses from participants on randomly 24 videos from our 176 generated ones. Each of our videos is paired with VACE and WAN 2.1 videos, with the shaded video labelled as 'Target' and other video labelled as Method A to D. We show our study result in Figure 13. We refer readers to Supporting Information for examples of our setup.

6.2. Machine Metrics

To assess control alignment, we first apply classical vision metrics—comparing each method directly against the provided target video. Following [CJL*25], we then combine quantitative VBench scores [HHY*24] with GPT-4o-based ratings. VBench measures motion smoothness, aesthetic quality, imaging quality, overall consistency, and temporal flicker, while GPT-4o evaluates physical realism, visual quality, semantic consistency, and alignment with the target video. The evaluation prompt we use for GPT evaluation are provided in the Supporting Information.

6.3. Result Observations

We discuss the result based the result of our evaluation shown in Tables 1–3 and Figure 13.

Controllability. As reported in Table 1, our method produces videos that are structurally the most faithful to the provided control signals across all categories. GPT-based evaluations (Table 3) further confirm that it achieves the highest alignment with both the textual prompts and the target video. By contrast, approaches relying solely on text control exhibit the poorest correspondence. These findings underscore the necessity of integrating simulation-based guidance into diffusion pipelines to satisfy the precision requirements of graphics workflows, since text alone is too sparse and ambiguous for fine-grained control.

Visual Quality. In addition to controllability, our goal is a fast rendering pipeline for fluid simulations. Our results show that the proposed framework strikes an effective balance between fidelity to the control input and overall aesthetic appeal. And to the best of our knowledge, we believe current ray tracing with fluid simulation techniques cannot give a comparable quality result in all scenarios that we showed in this literature given the same computational time and resource limit.

Physical Realism. Our method achieves the highest motion quality and temporal consistency under the VBench metrics, while matching VACE in both overall consistency and aesthetic quality. As expected, WAN-2.1 leads on physical realism: our focus is on

Table 1: *We perform classical vision validations comparing different methods with the target video (shaded simulation video).*

	Total score			Pure fluids			Solid–fluid interactions			Fluid–fluid mixtures		
Models	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
WAN-2.1	8.867	0.752	0.455	8.631	0.709	0.468	8.961	0.767	0.442	8.544	0.703	0.512
VACE-depth	8.052	0.655	0.511	7.401	0.577	0.560	8.257	0.680	0.489	7.466	0.586	0.589
VACE-scribble	9.053	0.583	0.531	9.441	0.566	0.665	9.152	0.584	0.500	8.182	0.591	0.594
Ours	**13.429**	**0.411**	**0.691**	**13.679**	**0.320**	**0.741**	**13.464**	**0.432**	**0.674**	**13.033**	**0.370**	**0.742**

Note: We see our method gives the best score across all scenarios.

Table 2: *We selected five categories from VBench [HHY*24] scores that's most related to the subject that we are studying, namely motion smoothness (MS), Aesthetic Quality (AQ), Imaging Quality (IQ), Overall Consistency (OC) and Temporal Flickering (TF).*

	Total score					Pure fluids					Solid–fluid interactions					Fluid–fluid mixtures				
Models	MS↑	AQ↑	IQ↑	OC↑	TF↑	MS↑	AQ↑	IQ↑	OC↑	TF↑	MS↑	AQ↑	IQ↑	OC↑	TF↑	MS↑	AQ↑	IQ↑	OC↑	TF↑
WAN-2.1	0.988	**0.554**	0.612	**0.231**	0.980	0.987	**0.556**	0.574	**0.198**	0.981	0.987	**0.565**	0.623	**0.242**	0.979	0.992	**0.488**	0.584	**0.200**	0.987
VACE-depth	0.990	0.515	0.630	0.221	0.984	0.994	0.491	0.551	0.169	0.992	0.989	0.531	0.650	0.237	0.981	**0.994**	0.451	0.584	0.169	0.992
VACE-scribble	**0.992**	0.461	0.590	0.202	**0.989**	**0.995**	0.418	0.564	0.165	0.993	**0.992**	0.477	0.606	0.215	0.987	**0.994**	0.409	0.529	0.163	**0.993**
Ours	**0.992**	0.480	**0.648**	0.221	**0.989**	0.994	0.447	**0.620**	0.168	**0.994**	**0.992**	0.500	**0.660**	0.237	**0.988**	**0.994**	0.395	**0.607**	0.180	**0.993**

Note: We show our method reaches the best temporal, motion and quality scores in all cases and generally performs better comparing to VACE.

Table 3: *Following [CJL*25], we use GPT-4o to evaluate the following metrics based on 10 evenly sampled frames.*

	Total score				Pure fluids				Solid–Fluid Interactions				Fluid–Fluid Mixtures			
Models	Phys↑	Vis↑	T-Align↑	V-Align↑	Phys↑	Vis↑	T-Align↑	V-Align↑	Phys↑	Vis↑	T-Align↑	V-Align↑	Phys↑	Vis↑	T-Align↑	V-Align↑
WAN-2.1	**0.793**	**0.798**	0.671	0.551	0.775	**0.822**	0.750	0.630	**0.794**	**0.793**	0.655	0.529	**0.805**	**0.806**	0.694	0.603
VACE-depth	0.706	0.725	0.666	0.710	0.793	0.782	0.801	0.796	0.682	0.708	0.638	0.682	0.765	0.772	0.712	**0.789**
VACE-scribble	0.665	0.704	0.640	0.682	0.762	0.776	0.802	0.780	0.635	0.682	0.603	0.652	0.747	0.765	0.708	0.765
Ours	0.740	0.773	**0.743**	**0.737**	**0.807**	0.804	**0.858**	**0.823**	0.724	0.763	**0.719**	**0.718**	0.769	0.800	**0.773**	0.767

Note: The criteria are physical realism (Phys), Visual Quality (Vis), Semantic Consistency (T-Align) and Target Video Consistency (V-Align).

giving artists fine-grained control over fluid effects, not on maximising scene accuracy. For example, in the human-face scenario (Figure 10), WAN-2.1 naturally produces head tracking and eye-blinking in response to water jets, whereas our output leaves the face static. Crucially, however, our framework fully supports moving solid boundaries—so artists can easily introduce exactly those motions to achieve any desired effect. By contrast, WAN-2.1's head movements, while seemingly more 'physical,' cannot be modified or controlled by the user.

6.4. Comparison with Traditional Rendering

In addition to video generation models, we also compare our method against results from a traditional rendering pipeline (Houdini). Our comparison includes three representative materials–water, milk chocolate, and milk (see Figure 14). Other materials such as beer, cola, or white-water effects depend heavily on the underlying simulation method and cannot be achieved with single-phase FLIP simulation, which was one of the key motivations for our work, and are therefore not included in the comparison.

6.5. Ablation and Parameter Study

In this section, we study the necessity of depth+scribble control, the transparency blending for tinted colouring, the mixture rate γ and the necessity of mixture mask in the fluid material blending

Dual Control. In Figure 15, we show the influence of different portions of depth control over scribble control applied. We believe the choice of 20% gives the reasonable result between quality and details. In that figure, the first row studies if we apply depth control first and how the portion of scribble will influence the result with the intuition that depth control gives more smooth signals and will be more suitable for early-stage denoising, which focuses on colouring generation instead of motion and details [SHWC25]. The second row shows other different schemes applying the controls. We see if applying scribble control in the early stage, even for just 10%,

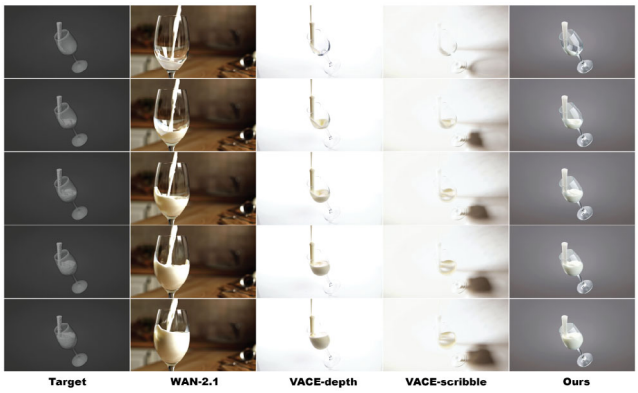

Figure 12: *Text Prompt: 'White milk poured down into a tilted wine glass.' We show that Wan 2.1 cannot achieve precise control only from textual input while VACE suffers from letting fluid material influence solid and background generation, causing over-exposure. The transparency is already applied to shaded video and is used as a control for VACE.*

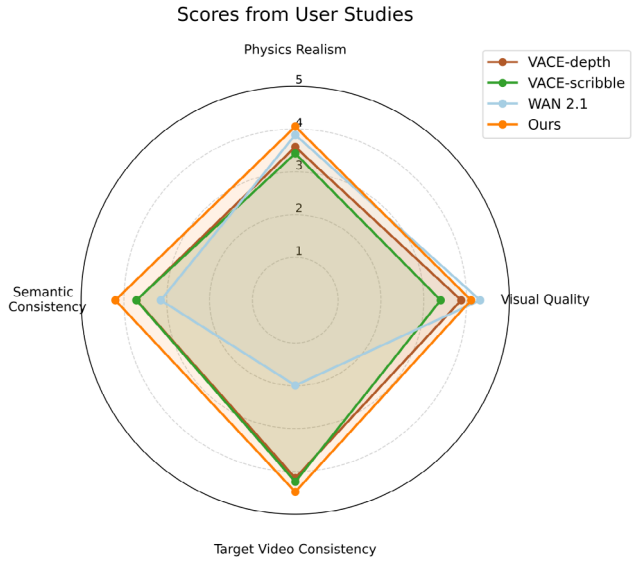

Figure 13: *In this figure, we show the result of our user studies. The result indicates that our method gives the best alignment and controllability with minimal sacrifice of visual qualities.*

will cause a noisy result, which is aligned with the observation in [SHWC25] and the intuition. Other schemes, like cyclic or random does not improves the simple binary split scheme shown in the first row and, therefore, is not used.

Transparency Blending. In Figure 16, we show that without using the transparency blending but only use the transparent shaded video, VACE pipeline either only gives solid or fluid material. With blending, more realistic effects of plane wings covered by fluid can be observed.

Mixture Rate. We study the control over speed of mixture between two fluids in Figure 17 with $\gamma \in \{1.0, 4.0, 8.0\}$ and $t_s = 5$ or $\gamma = 1.0$ and $t_s \in \{5, 12, 17\}$.

Mixture Mask. We show in Figure 18 that without a mixture mask, due to our fluid mixture mechanism of latent space blending between mixture material and separate materials, the result naturally changes the colouring of the area for fluids where mixtures should not be influencing.

7. Discussion

In this section, we first discuss the relationship between our method and some concurrent works in this area. Following this, we discuss the usage of material masks and mixture masks. Finally, we will discuss the relationship between our method and image-to-video models.

7.1. Relation to Concurrent Works

Here, we first discuss the relation between our method, WonderPlay [LYL*25], FluidNexus [GYZW25], and PhysGen3D [CJL*25]. Then, we show qualitive comparison between our method and diffusion as Shader (DaS) [GYL*25], followed by discussion on the results.

FluidNexus [GYZW25] focuses on reconstructing a 3D smoke plume from a single image and animating it using position-based dynamics (PBD). While effective for turning static images into multi-view sequences, their method does not support controllable effects such as colour changes, smoke material switching. While for WonderPlay [LYL*25], on the other hand, is conceptually closer to PhysGen3D [CJL*25], where the primary task is to manipulate an object given a picture and external force guidance. Their pipeline similarly relies on 3D reconstruction combined with in-the-loop PBD/MPM simulations at coarse resolution. Both methods are not designed for adding special effects for shaded fluid videos and did not fully utilise the capacity of video diffusion models.

Comparing with DaS [GYL*25], its problem formulation differs fundamentally from ours. DaS targets image manipulation under motion guidance, using an image-diffusion model (FLUX) with temporal consistency enforced through a tracking video. Their accuracy is tightly coupled to the quality of this tracking video, estimated via optical flow or mesh-based keypoint tracking, both of which fail under severe occlusions or topology changes where such conditions are common in fluids. In contrast, our method builds on a video-diffusion model, where physical plausibility is grounded in simulation results and dual-control signals from shaded videos. Temporal consistency arises directly from the video-diffusion backbone, and controls are enforced more strictly. We compare our method in a qualitative way with result from DaS as shown in Figure 19

7.2. Material Mask and Mixture Mask

In Section 6.5, we demonstrated the crucial role of masks in both material separation and fluid mixture. However, we acknowledge that masking can also introduce artifacts, particularly when solids

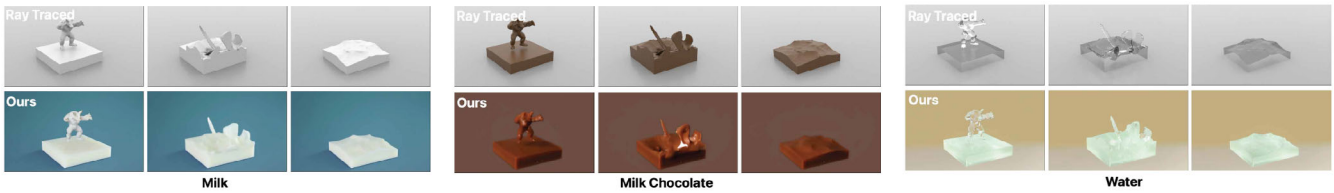

Figure 14: *Comparison with rendered fluid material. We show our method gives comparable quality for materials can be easily ray traced.*

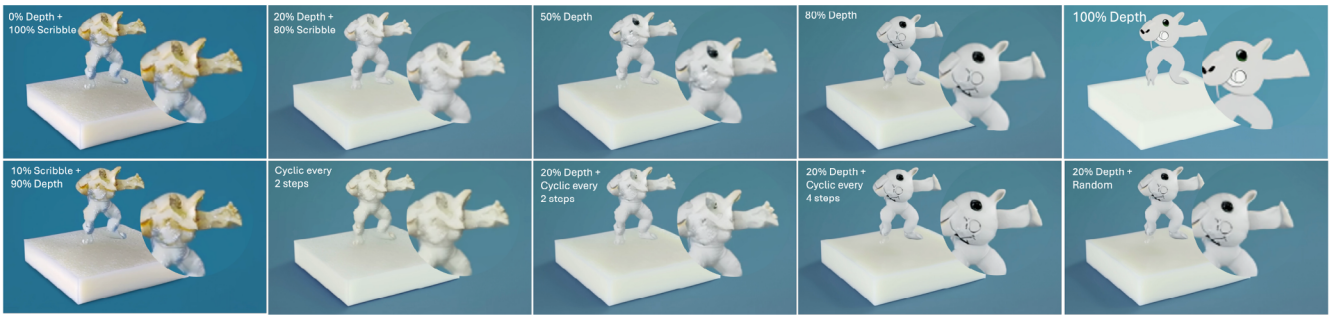

Figure 15: *We study the portion and method of depth versus scribble in the dual control process. The naming convention is that the control applied first will be in front of the '+' sign and the control applied after is behind it. In the first row, we study the portion of depth being injected in the pipeline with following the intuition that smooth signal should help colour generation in early steps. In the second row, we study other alternatives without following the intuition. 'Cyclic' means alternate control signal between depth and scribble and always start with the depth signal first. 'Random' means randomly select control signal uniformly. As justified by the first row of this figure, our choice of 20% gives the most satisfying result as what's reasoned from the observation in [SHWC25]. In the second row, we see cyclic schedules only work if depth is applied first and do not actually make much difference in two-steps cases and will cause problems if more depth is injected.*

Figure 16: *In the first row, without our transparency blending, the B-2 is either completely obscured by the fluid or not covered at all, both of which are unrealistic. In the second row, with transparency blending enabled, partial fluid retention on the solid surface is faithfully captured.*

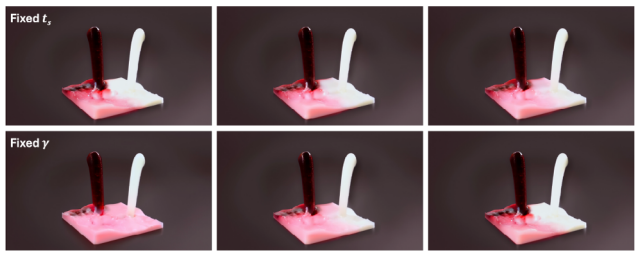

Figure 17: *We show the speed of mixture is controlled by γ and t_s. In the first row, t_s is fixed with γ increasing from left to right, meaning the increasing of mixing speed. In the second row, γ is fixed and t_s increasing from left to right, meaning the increasing of mixture starting time.*

and fluids are tightly interwoven or when fluids form thin structures such as condensation on glass, liquid-coated brushes, or hair. In these cases, fine features may not be preserved with the same fidelity as in the underlying simulation. Meanwhile, though our method is capable of generating results shown in this paper, our mask generation method is purely based on a basic fluid simulation without considering any physical mixture models and only aims for providing visual effects of mixture. In cases where the mask being timely jittering or have significant occlusion, that is, the other material being very tiny or is totallly invisible from the view camera, our method will not be able to give good result. We believe it remains a good direction for future work to study for explicitly modelling mixtures of multiple fluids.

7.3. Image-to-Video Models

Finally, we conclude this section by briefly discussing the relationship between our method and image-to-video models. Our approach naturally fits within the image-to-video setting, with the additional requirement that the initial image is generated under explicit control signals. Since modern video diffusion models are temporal extensions of image generators, frameworks such as VACE inherently

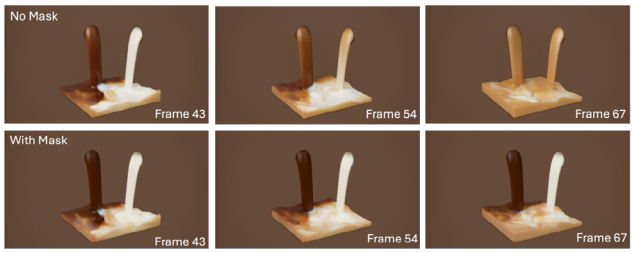

Figure 18: *We justify the necessity of the blending mask. We see without the blending mask, two fluid unaturally changes to the mixed colour in the later stage of video generation steps, while the masked version does not.*

support image-to-video generation, and our method can be directly applied in this context. However, without control videos, purely image-driven image-to-video models have limited controllability over future frames. This motivates the use of control videos even in the image-to-video regime, a design choice that is also reflected in the DoS pipeline [GYL*25].

8. Limitation and Future Work

We identify the following limitations of our method and possible future works from mainly four perspectives.

First, language guidance on material can sometimes cause inaccuracy and give non-desired result. In such cases, a style referenced picture guiding the material edition will be desirable. However, going such direction requires fine-tuning on existing models, which can be hard to achieve since both simulating/rendering realistic beer/coke still remains a time-consuming and difficult task especially considering the size of data required for fine-tuning a diffusion model. Despite this, one possible solution will be to manipulate the attention map within the DiT blocks as previously explored under stable diffusion models for video, as shown in [SLZ*24].

Second, our method applies control in a enforced way that though giving full control to artists for graphics pipline, did not take full advantage of general knowledge that's learnt from trillions of data embedded in the large model of DiTs. A weak guidance that remaining highly controllable would be more desirable.

Third, the limitation of our method comes from the ability of the backbone we are using, that is, the WAN-2.1 model both in terms of the length and the quality of the video. Future development in such model will directly improve the result of this work.

Finally, for future works, our current method has only been tested on free-surface fluid simulation where fluid usually can be represented by a surface mesh representing clear fluid surface. For other fluid types, including volumetric fluid like smoke and fire, more fine grind fluid like sands and snow or elastofluid like jelly or creams, we have not tested our method on such domain. One can definitely substitute the naive Phong shader with an OpenGL implementation of a simple volumetric data shader or other data types and still get real-time performance in such a situation. Hence, possible future direction includes extending current work to a full pipeline supporting all fluid phenomenons.

9. Conclusion

In conclusion, we have introduced a training-free framework that seamlessly combines classical fluid simulation, real-time shading, and controllable video diffusion to produce rich, photorealistic fluid effects. By leveraging a base incompressible simulator for precise layout and physics priors, and converting its output into a lightweight control video. We preserve full artist control without the implementation overhead of specialised fluid solvers. A language-guided diffusion model then enriches this control signal, adding foam, bubbles, mixtures, multi-material blending, and transparency via mask-based separation and blending in latent space.

Our experiments demonstrate that this hybrid pipeline achieves high visual fidelity across challenging scenarios (fluid mixtures, splashes, complex material interactions) while avoiding costly ray-tracing or difficult solver development. By building on off-the-shelf video-generation backbones with no additional training, this represents, to our knowledge, the first accessible, controllable, and physically grounded approach to fluid synthesis for graphics applications.

Acknowledgements

This work is conducted during the summer internship at Futurewei Technology Inc.

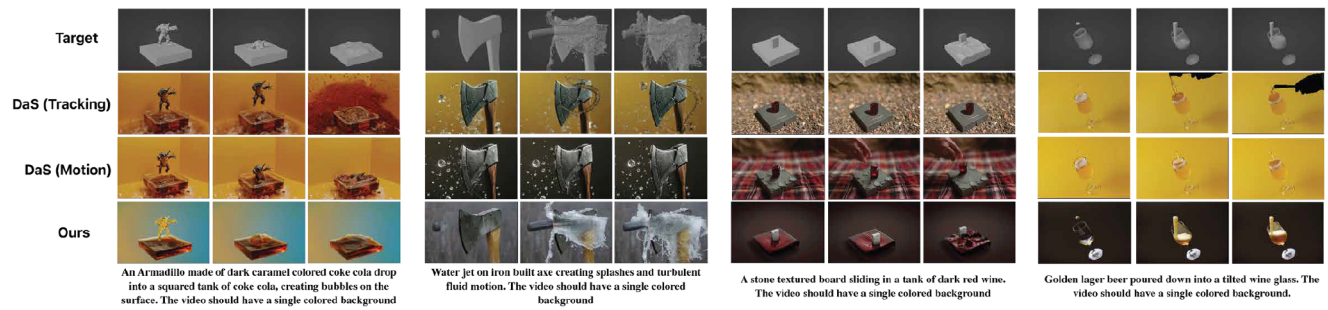

Figure 19: *We show a qualitative comparison between our method and Diffusion as Shader (DaS). Here, we compare against tracking video generated from Blender (Das (Tracking)) or from using motion transfer (Das (Motion)).*

Ethical Statement

This research did not involve human participants, animal subjects, or the use of any personal or sensitive data. Therefore, ethical approval was not required.

References

[AGL*17] Aanjaneya M., Gao M., Liu H., Batty C., Sifakis E.: Power diagrams and sparse paged grids for high resolution adaptive liquids. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–12.

[AT11] Ando R., Tsuruno R.: A particle-based method for preserving fluid sheets. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), pp. 7–16.

[BB12] Boyd L., Bridson R.: Multiflip for energetic two-phase fluid simulation. *ACM Transactions on Graphics (TOG) 31*, 2 (2012), 1–12.

[BKKW18] Bender J., Koschier D., Kugelstadt T., Weiler M.: Turbulent micropolar sph fluids with foam. *IEEE Transactions on Visualization and Computer Graphics 25*, 6 (2018), 2284–2295.

[BT07] Becker M., Teschner M.: Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 209–217.

[BXY*24] Bao F., Xiang C., Yue G., He G., Zhu H., Zheng K., Zhao M., Liu S., Wang Y., Zhu J.: Vidu: A highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233* (2024).

[CDI22] Chan C., Durand F., Isola P.: Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7915–7925.

[CJL*25] Chen B., Jiang H., Liu S., Gupta S., Li Y., Zhao H., Wang S.: Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 6178–6189.

[CYG*23] Chen J., Yu J., Ge C., Yao L., Xie E., Wu Y., Wang Z., Kwok J., Luo P., Lu H., et al.: Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426* (2023).

[Dee12] DeepMind G.: Veo 2. https://deepmind.google/technologies/veo/veo-2/ (2024.12).

[DGWH*15] De Goes F., Wallez C., Huang J., Pavlov D., Desbrun M.: Power particles: an incompressible fluid solver based on power diagrams. *ACM Transactions on Graphics 34*, 4 (2015), 50–1.

[EFFM02] Enright D., Fedkiw R., Ferziger J., Mitchell I.: A hybrid particle level set method for improved interface capturing. *Journal of Computational physics 183*, 1 (2002), 83–116.

[EMF02] Enright D., Marschner S., Fedkiw R.: Animation and rendering of complex water surfaces. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), pp. 736–744.

[FAW*16] Ferstl F., Ando R., Wojtan C., Westermann R., Thuerey N.: Narrow band flip for liquid simulations. In *Computer Graphics Forum* (2016), vol. *35*, Wiley Online Library, pp. 225–232.

[FF01] Foster N., Fedkiw R.: Practical animation of liquids. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (2001), pp. 23–30.

[FGW*21] Fei Y., Guo Q., Wu R., Huang L., Gao M.: Revisiting integration in the material point method: a scheme for easier separation and less dissipation. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–16.

[Gen24] GenmoTeam: Mochi 1. https://github.com/genmoai/models, 2024.

[GFCK02] Gibou F., Fedkiw R. P., Cheng L.-T., Kang M.: A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *Journal of Computational Physics 176*, 1 (2002), 205–227.

[GHF*25] Gillman N., Herrmann C., Freeman M., Aggarwal D., Luo E., Sun D., Sun C.: Force prompting: Video generation models can learn and generalize physics-based control signals. *arXiv preprint arXiv:2505.19386* (2025).

[GYL*25] Gu Z., Yan R., Lu J., Li P., Dou Z., Si C., Dong Z., Liu Q., Lin C., Liu Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. *arXiv preprint arXiv:2501.03847* (2025).

[GYR*24] Guo Y., Yang C., Rao A., Liang Z., Wang Y., Qiao Y., Agrawala M., Lin D., Dai B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations* (2024).

[GYZW25] Gao Y., Yu H.-X., Zhu B., Wu J.: Fluidnexus: 3d fluid reconstruction and prediction from a single video. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 26091–26101.

[HCB*24] HaCohen Y., Chiprut N., Brazowski B., Shalem D., Moshe D., Richardson E., Levin E., Shiran G., Zabari N., Gordon O., Panet P., Weissbuch S., Kulikov V., Bitterman Y., Melumian Z., Bibi O.: Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103* (2024).

[HCL*23] Huang L., Chen D., Liu Y., Shen Y., Zhao D., Zhou J.: Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778* (2023).

[HHK08] Hong W., House D. H., Keyser J.: Adaptive particles for incompressible fluid simulation. *Visual Computer 24* (2008), 535–543.

[HHY*24] Huang Z., He Y., Yu J., Zhang F., Si C., Jiang Y., Zhang Y., Wu T., Jin Q., Chanpaisit N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 21807–21818.

[HJP*24] Han Z., Jiang Z., Pan Y., Zhang J., Mao C., Xie C., Liu Y., Zhou J.: Ace: All-round creator and editor following instructions via diffusion transformer. *arXiv preprint arXiv:2410.00086* (2024).

[HLA*19] Hu Y., Li T.-M., Anderson L., Ragan-Kelley J., Durand F.: Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–16.

[HS22] Ho J., Salimans T.: Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[HSG*22] Ho J., Salimans T., Gritsenko A., Chan W., Norouzi M., Fleet D. J.: Video diffusion models. *arXiv preprint arXiv:2204.03458* (2022).

[HX23] Hu Z., Xu D.: Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet. *arXiv preprint arXiv:2307.14073* (2023).

[IAAT12] Ihmsen M., Akinci N., Akinci G., Teschner M.: Unified spray, foam and air bubbles for particle-based fluids. *Visual Computer 28* (2012), 669–677.

[IGLF06] Irving G., Guendelman E., Losasso F., Fedkiw R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. In *ACM SIGGRAPH 2006 Papers*. 2006, pp. 805–811.

[JHM*25] Jiang Z., Han Z., Mao C., Zhang J., Pan Y., Liu Y.: Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598* (2025).

[JMP*24] Jiang Z., Mao C., Pan Y., Han Z., Zhang J.: Scedit: Efficient and controllable image diffusion generation via skip connection editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition* (2024), pp. 8995–9004.

[JSL*24] Jin Y., Sun Z., Li N., Xu K., Xu K., Jiang H., Zhuang N., Huang Q., Song Y., Mu Y., Lin Z.: Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954* (2024).

[JSS*15] Jiang C., Schroeder C., Selle A., Teran J., Stomakhin A.: The affine particle-in-cell method. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–10.

[KLL*07] Kim B., Liu Y., Llamas I., Jiao X., Rossignac J.: Simulation of bubbles in foam with the volume control method. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 98–es.

[KSK10] Kim D., Song O.-y., Ko H.-S.: A practical simulation of dispersed bubble flow. In *ACM SIGGRAPH 2010 Papers*. 2010, pp. 1–5.

[KTZ*24] Kong W., Tian Q., Zhang Z., Min R., Dai Z., Zhou J., Xiong J., Li X., Wu B., Zhang J., et al.: Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603* (2024).

[Kua06] Kuaishou: Kling ai. https://klingai.kuaishou.com/ (2024.06).

[LBC*24] Li Z., Börcsök B., Chen D., Sun Y., Zhu B., Turk G.: Lagrangian covector fluid with free surface. In *ACM SIGGRAPH 2024 Conference Papers* (2024), pp. 1–10.

[LCPF12] Lentine M., Cong M., Patkar S., Fedkiw R.: Simulating free surface flow with very large time steps. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation* (2012), pp. 107–116.

[LD23] Li W., Desbrun M.: Fluid-solid coupling in kinetic two-phase flow simulation. *ACM Transactions on Graphics 42*, 4 (2023), 1–14.

[LGC*24] Lin B., Ge Y., Cheng X., Li Z., Zhu B., Wang S., He X., Ye Y., Yuan S., Chen L., et al.: Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131* (2024).

[LMLD22] Li W., Ma Y., Liu X., Desbrun M.: Efficient kinetic simulation of two-phase flows. *ACM Transactions on Graphics 41*, *4* (2022), 114.

[LRGW24] Liu S., Ren Z., Gupta S., Wang S.: Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision* (2024), Springer, pp. 360–378.

[LSD*22] Lesser S., Stomakhin A., Daviet G., Wretborn J., Edholm J., Lee N.-H., Schweickart E., Zhai X., Flynn S., Moffat A.: Loki: A unified multiphysics simulation framework for production. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–20.

[LSSF06] Losasso F., Shinar T., Selle A., Fedkiw R.: Multiple interacting liquids. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 812–819.

[LTKF08] Losasso F., Talton J., Kwatra N., Fedkiw R.: Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 797–804.

[Lum06] LumaLabs: Dream machine. https://lumalabs.ai/dream-machine (2024.06).

[LYL*25] Li Z., Yu H.-X., Liu W., Yang Y., Herrmann C., Wetzstein G., Wu J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. *arXiv preprint arXiv:2505.18151* (2025).

[MCG03] Müller M., Charypar D., Gross M.: Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 154–159.

[MMC16] Macklin M., Müller M., Chentanez N.: Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (2016), pp. 49–54.

[MSAA*24] Montanaro A., Savant Aira L., Aiello E., Valsesia D., Magli E.: Motioncraft: Physics-based zero-shot video generation. *Advances in Neural Information Processing Systems 37* (2024), 123155–123181.

[MST10] McAdams A., Sifakis E., Teran J.: A parallel multigrid poisson solver for fluids simulation on large grids. In *Symposium on Computer Animation* (2010), vol. *65*, p. 74.

[Ope24] OpenAI: Video generation models as world simulators, 2024. URL: https://openai.com/index/video-generation-models-as-world-simulators/.

[PAKF13] Patkar S., Aanjaneya M., Karpman D., Fedkiw R.: A hybrid lagrangian-eulerian formulation for bubble generation and dynamics. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2013), pp. 105–114.

[Pik10] PikaLabs: Pika 1.5. https://pika.art/ (2024.10).

[PX23] Peebles W., Xie S.: Scalable diffusion models with transformers. In *ICCV* (2023), pp. 4195–4205.

[PZB*] Polyak A., Zohar A., Brown A., Tjandra A., Sinha A., Lee A., Vyas A., Shi B., Ma C.-Y., Chuang C.-Y., Yan D., et al.: Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*.

[QZY*23] Qin C., Zhang S., Yu N., Feng Y., Yang X., Zhou Y., Wang H., Niebles J. C., Xiong C., Savarese S., et al.: Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147* (2023).

[RFB15] Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (2015), Springer, pp. 234–241.

[RKH*21] Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., Clark J., et al.: Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (2021), PmLR, pp. 8748–8763.

[RLH*20] Ranftl R., Lasinger K., Hafner D., Schindler K., Koltun V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 3 (2020), 1623–1637.

[RLY*14] Ren B., Li C., Yan X., Lin M. C., Bonet J., Hu S.-M.: Multiple-fluid sph simulation using a mixture model. *ACM Transactions on Graphics (TOG) 33*, 5 (2014), 1–11.

[Run06] Runway: Gen-3. https://runwayml.com/ (2024.06).

[RWT11] Raveendran K., Wojtan C., Turk G.: Hybrid smoothed particle hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), pp. 33–42.

[SHWC25] Shaulov A., Hazan I., Wolf L., Chefer H.: Flowmo: Variance-based flow guidance for coherent motion in video generation. *arXiv preprint arXiv:2506.01144* (2025).

[SLZ*24] Song Q., Lin M., Zhan W., Yan S., Cao L., Ji R.: Univst: A unified framework for training-free localized video style transfer. *arXiv preprint arXiv:2410.20084* (2024).

[SP09] Solenthaler B., Pajarola R.: Predictive-corrective incompressible sph. In *ACM SIGGRApH 2009 Papers*. 2009, pp. 1–6.

[Sta99] Stam J.: Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (1999), pp. 121–128.

[SWBD20] Stomakhin A., Wretborn J., Blom K., Daviet G.: Underwater bubbles and coupling. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks* (2020), pp. 1–2.

[SWT*18] Sato T., Wojtan C., Thuerey N., Igarashi T., Ando R.: Extended narrow band flip for liquid simulations. In *Computer Graphics Forum* (2018), vol. *37*, Wiley Online Library, pp. 169–177.

[TSS*07] Thürey N., Sadlo F., Schirm S., Müller-Fischer M., Gross M.: Real-time simulations of bubbles and foam within a shallow water framework. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 191–198.

[WFS22] Wretborn J., Flynn S., Stomakhin A.: Guided bubbles and wet foam for realistic whitewater simulation. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–16.

[WWA*25] Wan T., Wang A., Ai B., Wen B., Mao C., Xie C.-W., Chen D., Yu F., Zhao H., Yang J., et al.: Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025).

[WYC*23] Wang J., Yuan H., Chen D., Zhang Y., Wang X., Zhang S.: Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023).

[WYZ*23] Wang X., Yuan H., Zhang S., Chen D., Wang J., Zhang Y., Shen Y., Zhao D., Zhou J.: Videocomposer:

Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems 36* (2023), 7594–7611.

[XCW*20] Xiao Y., Chan S., Wang S., Zhu B., Yang X.: An adaptive staggered-tilted grid for incompressible flow simulation. *ACM Transactions on Graphics 39*, 6 (2020), 171–1.

[XWZ*25] Xiao S., Wang Y., Zhou J., Yuan H., Xing X., Yan R., Li C., Wang S., Huang T., Liu Z.: Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 13294–13304.

[XZJJ25] Xie T., Zhao Y., Jiang Y., Jiang C.: Physanimator: Physics-guided generative cartoon animation. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 10793–10804.

[YTZ*25] Yang Z., Teng J., Zheng W., Ding M., Huang S., Xu J., Yang Y., Hong W., Zhang X., Feng G., Yin D., Gu X., Zhang Y., Wang W., Cheng Y., Liu T., Xu B., Dong Y., Tang J.: CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *ICLR* (2025).

[ZB05] Zhu Y., Bridson R.: Animating sand as a fluid. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 965–972.

[ZPY*24] Zheng Z., Peng X., Yang T., Shen C., Li S., Liu H., Zhou Y., Li T., You Y.: Open-sora: Democratizing efficient video production for all, March 2024.

[ZRA23] Zhang L., Rao A., Agrawala M.: Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision* (2023), pp. 3836–3847.

[ZWJ*23] Zhang Y., Wei Y., Jiang D., Zhang X., Zuo W., Tian Q.: Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077* (2023).

[ZWY*22] Zhou D., Wang W., Yan H., Lv W., Zhu Y., Feng J.: Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018* (2022).

[ZXM*25] Zhang K., Xiao C., Mei Y., Xu J., Patel V. M.: Think before you diffuse: LLMS-guided physics-aware video generation. *arXiv preprint arXiv:2505.21653* (2025).

[ZZKF15] Zheng W., Zhu B., Kim B., Fedkiw R.: A new incompressibility discretization for a hybrid particle mac grid representation with surface tension. *Journal of Computational Physics 280* (2015), 96–142.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

Supporting Information

Supporting Information