FluidGPT: Towards Fluid Dynamics Reasoning in Vision-Language Models

Jusang Oh Soeon Park Jason Park Kangsun Lee
Seoul National University
Seoul, South Korea
{dhwntkd412, soeon1209, jsp11235, calvin1225}@snu.ac.kr

Abstract

Vision-Language Models (VLMs) have achieved strong performance on multimodal tasks but remain limited in their understanding of physical dynamics, especially in fluid scenarios. To address this gap, we propose ***FluidGPT****, a modular framework for fluid dynamics reasoning in VLMs: we generate a comprehensive fluid simulation dataset with precise physical annotations and train a small VLM on these controlled simulations to produce structured descriptions of each scene's visual content and physical inferences—enabling tasks ranging from basic perception and cross-scene comparison to causal reasoning. These outputs guide a larger VLM to analyze the visual content of a single scene and compare physical inferences across multiple scenes for advanced reasoning. Experiments show that that a compact modular architecture yields noticeable performance improvements. FluidGPT and our simulation dataset offes a new path toward equipping VLMs with fluid physical common sense. Code and dataset will be publicly available at* `https://github.com/Ever2after/snu-mlvu-project-2025`.

1. Introduction

With the rapid evolution of Vision-Language Models (VLMs) [9, 12], which leverage Large Language Models (LLMs) as their backbone, significant progress has been made across various multimodal tasks—including image captioning, visual question answering, and more [1, 8]. However, recent studies have revealed that even state-of-the-art VLMs continue to struggle with in their **understanding of the physical world**. For instance, prior work [5] shows that the average accuracy of VLMs on physical reasoning tasks remains below half of human-level performance. This gap is particularly pronounced in **fluid dynamics scenarios**, as demonstrated by the ContPhy [18] benchmark.

As VLMs are increasingly applied to real-world settings—ranging from action generation and decision-making to robotics and embodied AI—it becomes critical for them to acquire a robust understanding of physical common sense [5, 11]. While fluids are as ubiquitous in everyday contexts as rigid bodies, their inherent continuity and deformability make them far more challenging to model and reason about. Thus, equipping VLMs with the ability to reason about fluid dynamics is an essential step toward comprehensive physical understanding [19].

Several approaches have been proposed to enhance the physical reasoning capabilities of VLMs. Prior work [17] has introduced Multimodal Chain-of-Thought prompting to improve visual reasoning; PhysBench [5] proposed the PhysAgent framework, integrating foundation vision models with external knowledge memory; and the Physics Context Builders [2] demonstrated performance gains by fine-tuning small VLMs with scene description data. Furthermore, Cosmos-Reason1 [11] explored the use of reinforcement learning (RL) following supervised fine-tuning (SFT) to further boost reasoning abilities. Yet, these studies have largely focused on rigid-body dynamics or general commonsense reasoning, with little to no attention given to fluid dynamics.

To address this gap, we introduce **FluidGPT**, a general modular framework for fluid dynamics reasoning in VLMs. FluidGPT trains a small VLM to generate structured descriptions across a hierarchy of fluid reasoning tasks—ranging from low-level perception to high-level causal reasoning—based on visually simulated scenes. The training dataset is fully generated from controlled fluid simulations, allowing precise and automatic annotation of perceptual attributes such as color, shape, and position, as well as physical properties like viscosity.

The resulting VLM is not only capable of describing the visual behavior of fluid, but also inferring its underlying physical properties and explaining the governing physical principles. Furthermore, we leverage the small VLM's structured output to guide a larger VLM in performing advanced reasoning tasks such as next scene prediction and counterfactual analysis.

Our contributions are summarized as follows:

1. We present a simulation pipeline for generating diverse fluid scenes with automatic physical and perceptual annotations.
2. We construct a large-scale dataset covering both single-scene perception and cross-scene reasoning tasks.
3. We propose **FluidGPT**, a modular framework that boosts fluid reasoning via structured scene descriptions from a lightweight sensing module.

2. Related Work

Physical Reasoning in Vision-Language Models

Recent studies have highlighted the limitations of Vision-Language Models (VLMs) in physical reasoning tasks. PhysBench [5] presents a comprehensive benchmark evaluating VLMs across object properties, object relationships, scene understanding, and physics-based dynamics. Despite achieving high performance on general multimodal tasks, VLMs still fall significantly short of human-level accuracy on physics-related scenes.

To address these limitations, several frameworks have been proposed. The Physics Context Builders (PCB) framework [2] fine-tunes a small VLM on simulated physical scenes and provides its structured descriptions as context to a larger VLM, achieving up to 13.8% performance gain. Cosmos-Reason1 [11] develops a four-stage training pipeline—including supervised fine-tuning (SFT) and reinforcement learning (RL)—to enhance physical reasoning in multimodal LLMs, with RL post-training yielding an additional 8.2% improvement in benchmark accuracy.

Multimodal Chain-of-Thought (CoT) prompting [17] improves visual reasoning by explicitly generating intermediate rationales that jointly leverage visual and textual modalities. This two-stage framework achieves higher accuracy on benchmarks such as ScienceQA [10], while also mitig ating hallucinations. PhysAgent [5] combines generalist VLMs with expert vision encoders and physics-aware memory modules, demonstrating an 18.4% improvement on physical tasks with GPT-4o.

Despite recent advances, existing works primarily focus on rigid-body dynamics or commonsense scenarios. ContPhy [18] shifts attention to fluid and deformable object reasoning using continuum-based 2D simulations, and finds that VLMs perform particularly poorly in such tasks—largely due to their limited ability to perceive and reason about highly deformable materials.

Physical Simulation for AI Training

Simulated environments have been widely used to train and evaluate AI systems for physical reasoning. For example, CLEVRER [16] and Falling Towers [2] provide synthetic video datasets designed for temporal and causal reasoning in rigid-body scenes. Physion [3] targets intuitive physical prediction, while IntPhys [13] evaluates human-like physics understanding by contrasting physically plausible and implausible sequences.

For fluid dynamics, ContPhy [18] offers 2D fluid simulations with annotations such as masks, bounding boxes, and point-level physics attributes. However, its visual diversity and scene complexity remain limited. In contrast, datasets like FLUID-LLM [19], BLAST-Net [6], and EAGLE [7] focus on engineering-oriented predictions using scalar or vector fields represented as colormaps, often relying solely on numerical solvers and volume-rendered outputs. While valuable for scientific computing, these datasets lack the visual realism and perceptual grounding required for training VLMs on fluid understanding tasks.

These efforts underscore the importance of developing simulation datasets that integrate accurate physics with visually realistic inputs to support fluid dynamics reasoning in VLMs.

3. Dataset

3.1. Scene Taxonomy

We generate a diverse set of controlled fluid simulations using Blender[1]. Scenes include both basic flow phenomena and interactions with rigid objects. Physical parameters such as viscosity, velocity, and object geometry are precisely adjustable, enabling fine-grained annotation and reproducible rendering.

Scenario	**Representative Setups**
S1 Basic Fluid Phenomena	1) Fluid flowing down a slope. 2) Jet falling into a container. 3) Ripples from surface impact.
S2 Fluid–Rigid-Body Interaction	1) Object moving in fluid. 2) Fluid flowing around a object.

Table 1. Simulated scenario types.

Table 1 outlines the core simulation blocks used to construct our dataset. The “Basic Fluid Phenomena” category captures canonical behaviors such as laminar flow, falling jets, and ripple dynamics, which are essential for assessing fundamental fluid perception. The “Fluid–Rigid-Body Interaction” category emphasizes the interplay between deformable and solid entities, enabling evaluation of more complex reasoning involving collisions, buoyancy, and wake formation. This dual-branch structure ensures comprehensive physical coverage with minimal redundancy. Figure 2 illus-

[1] https://www.blender.org

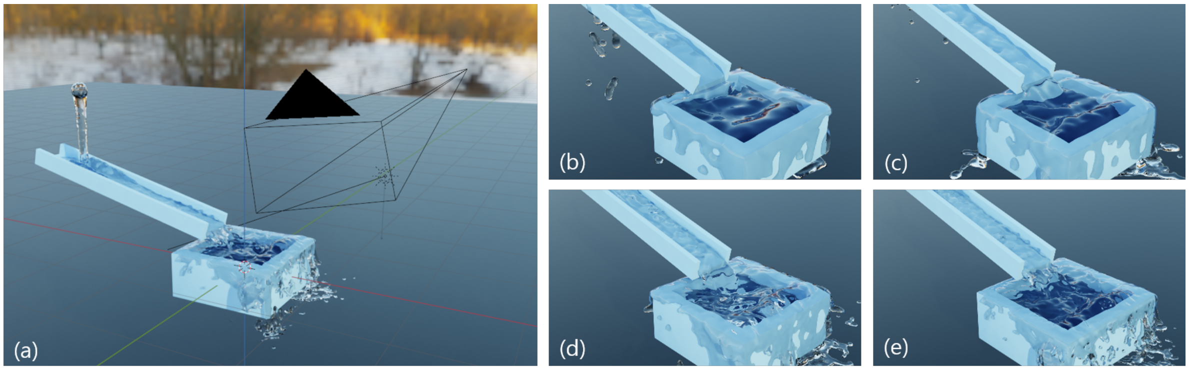
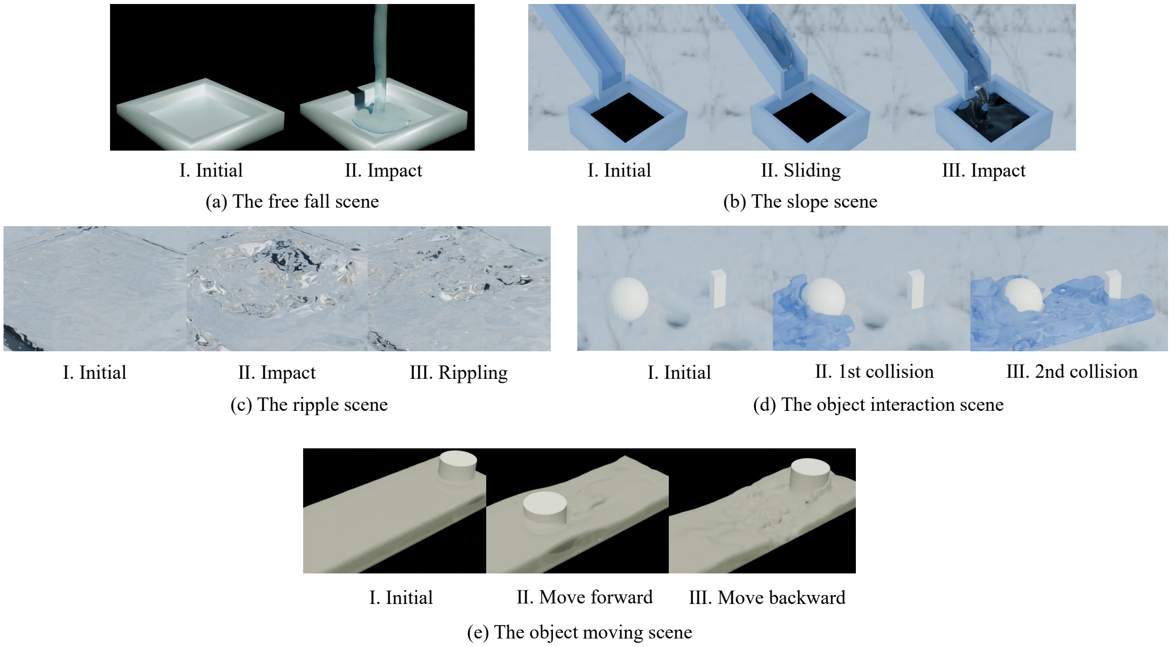


Figure 1. This figure illustrates an example scene and rendered frames created by our pipeline. The example scene shown in (a) consists of a fluid source, a slope and a sink that the fluid can interact with and a camera denoted as a black wireframe on the top-right region. (b), (c), (d), and (e) are the rendered frames with varying fluid simulation resolution((b): 96^3, (c): 128^3, (d): 192^3, and (e): 256^3). Higher fluid resolution results in more realistic fluid behavior and less unwanted leakage through the slope, at the cost of increased simulation time.



(a) The free fall scene

(b) The slope scene

(c) The ripple scene

(d) The object interaction scene

(e) The object moving scene

Figure 2. Examples of fluid simulation scenes

trates representative scenes from simulated fluid dynamics dataset.

3.2. Task Formulation

Table 2 presents our task taxonomy, organized by *scene type* (single vs. cross-scene), the *target* of reasoning (object, fluid, or their interaction), and the specific *task*.

In **single-scene settings**, tasks primarily assess visual perception capabilities, such as identifying the color, location, or rotation of an object, and tracking the motion or location of fluid. These tasks focus on recognizing and understanding properties directly observable within a single scene. Collision detection is also included to evaluate the ability to reason about interactions between objects and fluids.

In contrast, **cross-scene tasks** require comparative physical reasoning between two independent scenes. These include judging which fluid has a greater amount or lower viscosity. Unlike single-scene tasks, these cannot be answered by observing one scene alone—they require integrating perceptual understanding across both scenes and performing step-by-step reasoning over latent physical properties. Such tasks bridge visual perception and abstract inference, testing the model's ca-

Table 2. Task categorization by scene type and target.

Scene type	Target	Task
Single scene	Object	Color Location Rotation
	Fluid	Color Location Direction
	Object + Fluid	Collision
Cross-scene	Fluid	Amount Viscosity

pacity for multi-scene physical understanding.

3.3. Simulation Pipeline

3.3.1. Pipeline Overview

We propose a semi-automatic dataset generation pipeline using Blender. We first manually generate a small set of base scenes using Blender GUI, where a human sets all necessary scene properties such as object locations and fluid parameters. Base scenes refer to a set of Blender scene configurations that mainly differs from each other in regards to the global arrangement of the scene objects. This manual generation stage is employed to create diverse situations involving fluids in our dataset, thereby aiding the model in generalizing fluid properties.

Subsequently, the base anchor scenes are augmented using an automated method. We implement a script that converts a base scene into a Python file that when executed within Blender, generates the original base scene. The generated Python file explicitly includes numerical variables such as the locations of the meshes and the camera, material color and alpha, and fluid properties. Meanwhile, the mesh geometry is saved in individual files and the Python file merely loads them into the Blender scene upon execution. During the conversion to the Python file, we specify to the conversion script the set of properties that will later be augmented. Examples of these properties include fluid viscosity, material color and camera position. These properties are stored as variable names rather than as numerical constants in the resulting Python file.

In the final stage of the pipeline, we manually provide the augmentation ranges for the properties that were specified in the previous stage. Given the generated Python file and the defined augmentation ranges, an automated procedure samples a scene using the Python file, runs the fluid simulation and renders the final result into a video. This procedure simultaneously generates the ground-truth dataset annotations—such as fluid velocity and bounding box of the fluid in each frame—by utilizing the simulation cache. Specifically, we parse and process the data from .vdb and .bobj files generated during simulation, which contain the fluid particle motion data and the fluid mesh data respectively.

3.3.2. Implementation Details

We conducted a resolution ablation study to investigate the trade-off between visual fidelity, physical realism, and computational cost in our fluid simulations. We tested four domain resolutions—96^3, 128^3, 192^3, and 256^3—and evaluated both the perceptual quality of the fluid and the rendering time required per sequence. As shown in Figure 1, increasing the resolution consistently improves surface detail, splash sharpness, and the overall continuity of fluid boundaries. However, the rendering time grows nearly linearly with resolution, making the higher-resolution configurations (e.g., 192^3 and 256^3) computationally impractical for large-scale dataset generation.

While the highest resolution (256^3) achieves the most detailed fluid surfaces, it offers only marginal gains over 128^3 in terms of visual quality, while nearly doubling both simulation and rendering time. Conversely, we observed that resolutions below 90^3 introduce physical artifacts that compromise simulation reliability: specifically, the fluid tends to penetrate effector boundaries unless an artificially thick surface is applied. This results in noticeable spatial separation between the fluid and solid objects, breaking physical plausibility. Although lowering the CFL number can mitigate this issue, doing so substantially increases computational overhead.

Considering these trade-offs, we identify 96^3 as the optimal resolution for our setting—it preserves essential surface features, maintains physical accuracy, and ensures that the simulation remains computationally tractable. All experiments in our dataset are therefore conducted at 96^3 resolution by default.

To further optimize rendering for downstream vision-language model (VLM) training, we configure the Cycles renderer with 64 samples per pixel, and enable Blender's built-in denoiser to suppress high-frequency noise. Empirical testing shows that this setup maintains key visual features—such as fluid contours, transparency, and splash morphology—while keeping the average render time under 10 minutes per 5-second sequence (150 frames at 30 FPS on a GTX 1080 GPU). This balance between quality and efficiency allows scalable data generation without compromising the physical or perceptual fidelity required for multimodal learning tasks.

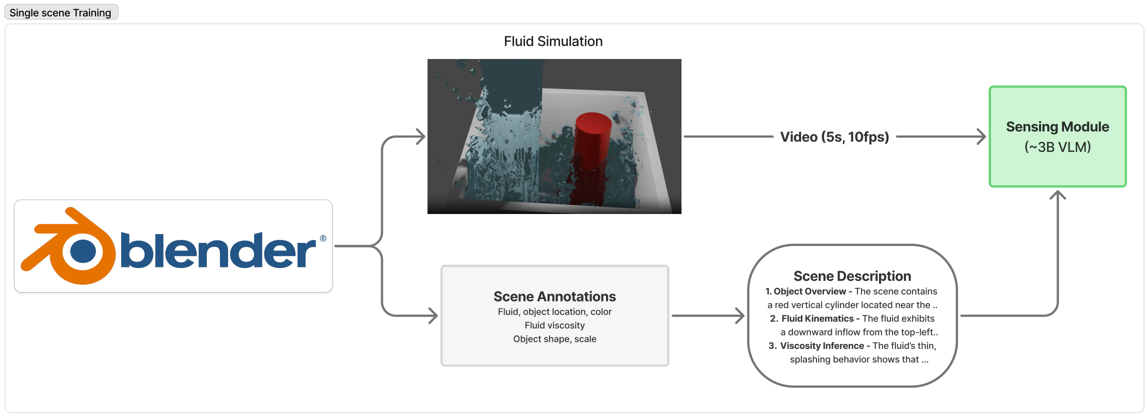

Single scene Training
Fluid Simulation
Video (5s, 10fps)
Sensing Module
(~3B VLM)
blender
Scene Annotations
Fluid, object location, color
Fluid viscosity
Object shape, scale
Scene Description
1. Object Overview - The scene contains a red vertical cylinder located near the ..
2. Fluid Kinematics - The fluid exhibits a downward inflow from the top-left..
3. Viscosity Inference - The fluid's thin, splashing behavior shows that ...

Figure 3. Training pipeline. We use Blender to generate both rendered fluid scenes and associated scene annotations, including object location, fluid velocity, and viscosity. A 3B scale Sensing Module is trained to produce structured scene descriptions covering both visual perception and physical inference.

3.4. Scene Description Generation

We implemented a pipeline that ingests raw simulation parameter logs—object morphology, color values, viscosity, and so on—and automatically generates natural language descriptions for each scene. The simulation produces annotations at each time step, and our pipeline converts these into coherent English descriptions. To capture view-dependent effects, we manually reviewed each camera angle, recording fluid movement directions from each viewpoint, and directly observed the characteristic fluid behaviors, incorporating both into the text. Each scene description consists of:

1. **Object overview**: enumerating all entities present, their categorical types, positions, and colors;
2. **Fluid kinematics**: summarizing motion patterns (flow direction, speed variations, presence of vortices);
3. **Viscosity inference**: estimating relative viscosity levels by correlating observed flow behavior with annotated parameters.

Figure 4 illustrates how raw log annotations are rendered into human-like narratives.

4. FluidGPT

4.1. Methodology

We adopt a modular, two-stage architecture for fluid dynamics reasoning, composed of a lightweight Sensing Module and a larger Reasoning Module. This separation enables the system to first extract physically grounded scene representations from visual input, and then use them for higher-level reasoning.

Scene type: slope

Scene description:
A **{fluid_description}** fluid initially rests atop a **{slope_description}** slope. Below the slope lies a **{container_description}** container.
The fluid clings to the slope and moves very slowly downward. As **the fluid sticks to the incline and barely moves**, the fluid's viscosity is estimated as **high**.

Scene type: object interaction

Scene description:
In the scene two objects appear, with **{object1}** on the left and **{object2}** to its right. A **{fluid_description}** fluid flows from **{flow_direction}** toward the objects.
The fluid collides with **{collision1}**. And then it collides with **{collision2}**.
As **the fluid flows swiftly around the objects with minimal adhesion**, the fluid's viscosity is estimated as **low**.

Figure 4. Description generation formats. (left) example of basic fluid phenomena scene. (right) example of fluid-rigid body interaction scene

4.1.1. Training the Sensing Module

As shown in Figure 3, we begin by generating synthetic fluid scenes using Blender. Each simulation produces rendered RGB frames along with structured annotations, including object layout, fluid motion, and physical pa-

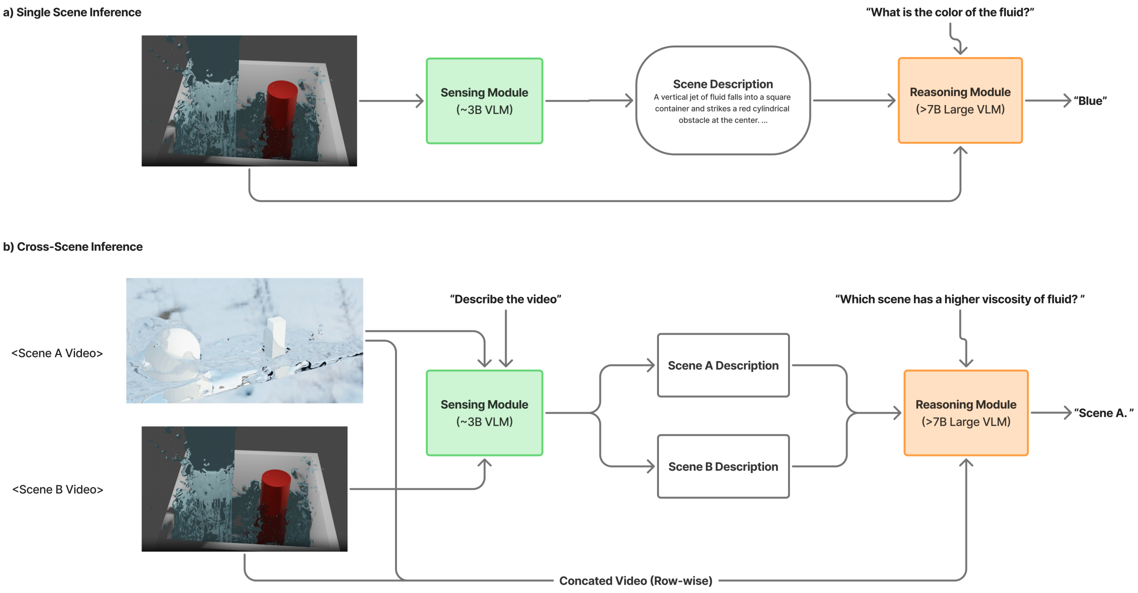


Figure 5. Inference pipeline. At test time, the Sensing Module generates a structured description from visual input of single scene. This description is passed to a larger Reasoning Module (>7B VLM), which performs higher-level tasks in both single scene and multiple scenes.

rameters such as viscosity. These annotations are used to supervise a compact vision-language model (~3B parameters, e.g., Qwen2.5-VL-3B [15]), referred to as the Sensing Module.

The Sensing Module is trained to generate structured natural language descriptions from video input, organized into three components: (1) Object Overview, detailing visible entities and their properties; (2) Fluid Kinematics, describing flow patterns and motion behaviors; and (3) Viscosity Inference, estimating latent physical attributes based on visual cues. This training objective enables the model to learn grounded, interpretable mappings from fluid scenes to descriptive representations that can assist downstream reasoning.

4.1.2. Inference via Modular Reasoning

At inference time, both the video input and the scene description are provided to the Reasoning Module (>7B VLM, e.g., GPT-4o [12] or Gemini2 [14]), which serves as the primary model responsible for final decision-making. The Sensing Module acts as an auxiliary encoder that summarizes the visual scene into a structured description comprising object configuration, fluid motion, and physical properties (e.g., viscosity), which is used to guide or complement the reasoning process.

In single-scene tasks, one video and its description are used to answer questions about the scene. In cross-scene tasks, two independent videos are each processed into separate descriptions, and both the visual inputs and paired descriptions are jointly provided to the Reasoning Module for comparative reasoning. This design improves interpretability and physical grounding while enabling the large model to perform robust, context-aware inference.

4.2. Baselines

We evaluate a range of publicly available vision-language models (VLMs) as baselines to assess their ability to handle physically grounded reasoning in fluid-dynamic visual scenes. Our selection includes both open-source and commercial models, with a mix of general-purpose VLMs and video-specialized architectures.

Among open-source models known for strong visual understanding, we include Qwen2.5-VL-7B-Instruct [15], which supports structured outputs and long-video event tracking for complex spatiotemporal queries and InternVL3-8B-Instruct [4], which adopts native multimodal pretraining and advanced visual encoding, showing strength in high-precision domains such as scientific and industrial scenes.

We also include several commercial foundation models as baselines, the GPT series (GPT-4o, GPT-4o-mini) [12] . While these models offer strong performance across a wide range of multimodal tasks, including static and video input, their behavior in fluid-physics scenarios remains largely unexplored. They are included to benchmark the generalization ability of high-capacity

Table 3. Accuracy (%) on our FluidBench. Single-scene tasks evaluate visual perception (object / fluid properties and collision), while cross-scene tasks test comparative physical reasoning (fluid amount and viscosity). Numbers in parentheses indicate the absolute gain over the corresponding base model.

Category	**Model**	**Single Scene**							**Cross-Scene (Fluid)**		**Total**
		Object			**Fluid**			Coll.	Amt.	Visc.	
		Col.	Loc.	Rot.	Col.	Loc.	Dir.				
	Random Choice	35.56	33.33	50	35.56	50	33.33	50	33.33	50	43.04
Open-source	InternVL3-8B	**86.67**	53.33	46.67	60	**60**	60	**80**	13.33	44	53.53
	Qwen2.5-VL-3B	66.67	40	40	40	53.33	40	60	13.33	48	45.29
	Qwen2.5-VL-7B	66.67	73.33	40	66.67	**60**	**93.33**	73.33	13.33	**56**	59.41
Closed-source	GPT-4o	73.33	86.67	73.33	46.67	**60**	**93.33**	80	13.33	44	59.41
	GPT-4o-mini	80	46.67	66.67	60	**60**	13.33	73.33	6.67	44	48.82
FluidGPT (Ours)	Qwen2.5-VL-7B	80	**100**	**86.67**	**100**	**60**	**93.33**	66.67	**26.67**	**56**	**70.59**
	+ 3B-SFT	(+13.33)	(+26.66)	(+46.67)	(+33.33)	(0)	(0)	(-6.66)	(+13.33)	(0)	(+11.18)
	GPT-4o-mini	80	**100**	86.67	**86.67**	53.33	53.33	73.33	**26.67**	52	64.71
	+ 3B-SFT	(0)	(+63.33)	(+20)	(+26.67)	(-6.66)	(+40)	(0)	(+20)	(+8)	(+15.89)

models in complex physical reasoning tasks.

We fine-tune a lightweight, open-source VLM, Qwen2.5-VL-3B [15], on a domain-specific dataset composed of simulated fluid scenes. We will then evaluate how providing the physical context generated by this fine-tuned small VLM affects the performance of larger baseline models, enabling a direct comparison between their zero-shot reasoning and context-augmented reasoning in fluid environments.

5. Experiment

5.1. Experimental Setups

For training the Sensing Module, we used the Qwen2.5-VL-3B model with a batch size of 16, learning rate of 2e-6, and 2 training epochs. The training was conducted on a single A100 GPU and completed within approximately one hour.

5.2. Results

As shown in Table 3, open-source baselines achieve 53.5% (InternVL3-8B), 45.3% (Qwen2.5-VL-3B), and 59.4% (Qwen2.5-VL-7B), while closed-source models record 59.4% (GPT-4o) and 48.8% (GPT-4o-mini); these results, obtained without any scene descriptions, show that the largest open-source model matches the top closed-source performance but that all models still face limitations in fluid reasoning. By augmenting each with our FluidGPT sensing module (Qwen2.5-VL-3B-SFT), Qwen2.5-VL-7B's accuracy rises to 70.6% (+11.2) and GPT-4o-mini's to 64.7% (+15.9). The largest gains are observed in object localization questions—where accuracy reaches 100%—highlighting the value of enriched contextual information. We also see notable improvements in cross-scene fluid amount (up to +13.3) and viscosity reasoning (up to +8). These results demonstrate that integrating a lightweight, specialized context module substantially enhances VLMs' performance across both single- and cross-scene reasoning tasks.

6. Conclusion

In this work, we proposed a novel dataset generation pipeline that leverages an existing simulator tool to generate diverse scenes involving fluids—ranging from free-fall droplets and inclined-plane flows to surface rippling and object–fluid interactions—while automatically annotating key perceptual and dynamic attributes. Leveraging this pipeline, we compile a comprehensive 7.2 K–sample train/test dataset that is rigorously categorized by task type (object color, location, rotation; fluid color, location, direction; cross-scene amount inference, and viscosity reasoning), representing the first realistic fluid-scene dataset of its kind and filling a crucial gap for VLM evaluation. Building atop these resources, we propose FluidGPT, a modular context-builder framework in which a lightweight sensing module (Qwen2.5-VL-3B-SFT) enriches base VLMs with structured scene descriptions. By adopting this modular design, FluidGPT allows compact VLMs to be trained with minimal overhead yet deliver substantial performance boosts—easily achieving state-of-the-art results on both single- and cross-scene reasoning tasks.

7. Future Work

Despite the scalability and structured diversity of our dataset, several limitations remain. First, due to hardware constraints, we were forced to reduce simulation resolution and rendering sample rates in certain cases, which led to a subset of videos with unnatural surface behavior or physically implausible motion. This degradation in visual and physical fidelity may hinder learning, particularly for models sensitive to fine-grained fluid details such as splash continuity and particle coherence. Second, our study does not include a validation of sim-to-real (sim2real) transfer—that is, we did not assess whether models trained on synthetic fluid simulations generalize to real-world fluid dynamics. As a result, the applicability of our dataset to real-world tasks such as robotic fluid manipulation or physical scene understanding is uncertain. Third, our evaluation is limited to perception-level tasks and does not cover higher-order reasoning abilities, such as next scene prediction or counterfactual inference. This limits our understanding of whether current models can move beyond perception to capture the causal and temporal structure of fluid-based events, which is critical for physical reasoning under uncertainty.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[2] Vahid Balazadeh, Mohammadmehdi Ataei, Hyunmin Cheong, Amir Hosein Khasahmadi, and Rahul G. Krishnan. Physics context builders: A modular framework for physical reasoning in vision-language models, 2025. 1, 2

[3] Daniel Bear, Elias Wang, Damian Mrowca, Felix Binder, Hsiao-Yu Tung, Pramod RT, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, Fei-Fei Li, Nancy Kanwisher, Josh Tenenbaum, Dan Yamins, and Judith Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 2

[4] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 6

[5] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*, 2025. 1, 2

[6] Wai Tong Chung, Bassem Akoush, Pushan Sharma, Alex Tamkin, Ki Sung Jung, Jacqueline H. Chen, Jack Guo, Davy Brouzet, Mohsen Talei, Bruno Savard, Alexei Y. Poludnenko, and Matthias Ihme. Turbulence in focus: Benchmarking scaling behavior of 3D volumetric super-resolution with BLASTNet 2.0 data. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023. 2

[7] Steeven Janny, Aurélien Benetteau, Nicolas Thome, Madiha Nadri, Julie Digne, and Christian Wolf. Eagle: Large-scale learning of turbulent fluid dynamics with mesh transformers. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1

[9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1

[10] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[11] NVIDIA, Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, Jacob Huffman, Jingyi Jin, Brendan Johnson, Rizwan Khan, George Kurian, Elena Lantz, Nayeon Lee, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Yen-Chen Lin, Ming-Yu Liu, Andrew Mathau, Yun Ni, Lindsey Pavao, Wei Ping, David W. Romero, Misha Smelyanskiy, Shuran Song, Lyne Tchapmi, Andrew Z. Wang, Boxin Wang, Haoxiang Wang, Fangyin Wei, Jiashu Xu, Yao Xu, Xiaodong Yang, Zhuolin Yang, Xiaohui Zeng, and Zhe Zhang. Cosmos-reason1: From physical common sense to embodied reasoning, 2025. 1, 2

[12] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, et al. Gpt-4 technical report, 2024. 1, 6

[13] Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning, 2020. 2

[14] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 6

[15] Qwen Team. Qwen2.5-vl, 2025. 6, 7

[16] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *ICLR*, 2020. 2

[17] Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024. 1, 2

[18] Zhicheng Zheng, Xin Yan, Zhenfang Chen, Jingzhou Wang, Qin Zhi Eddie Lim, Joshua B. Tenenbaum, and Chuang Gan. Contphy: continuum physical concept learning and reasoning from videos. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024. 1, 2

[19] Max Zhu, Adrián Bazaga, and Pietro Liò. Fluid-llm: Learning computational fluid dynamics with

spatiotemporal-aware large language models, 2024. 1, 2