VChain: Chain-of-Visual-Thought for Reasoning in Video Generation

Ziqi Huang, Ning Yu[✉][†], Gordon Chen, Haonan Qiu, Paul Debevec, Ziwei Liu[✉]
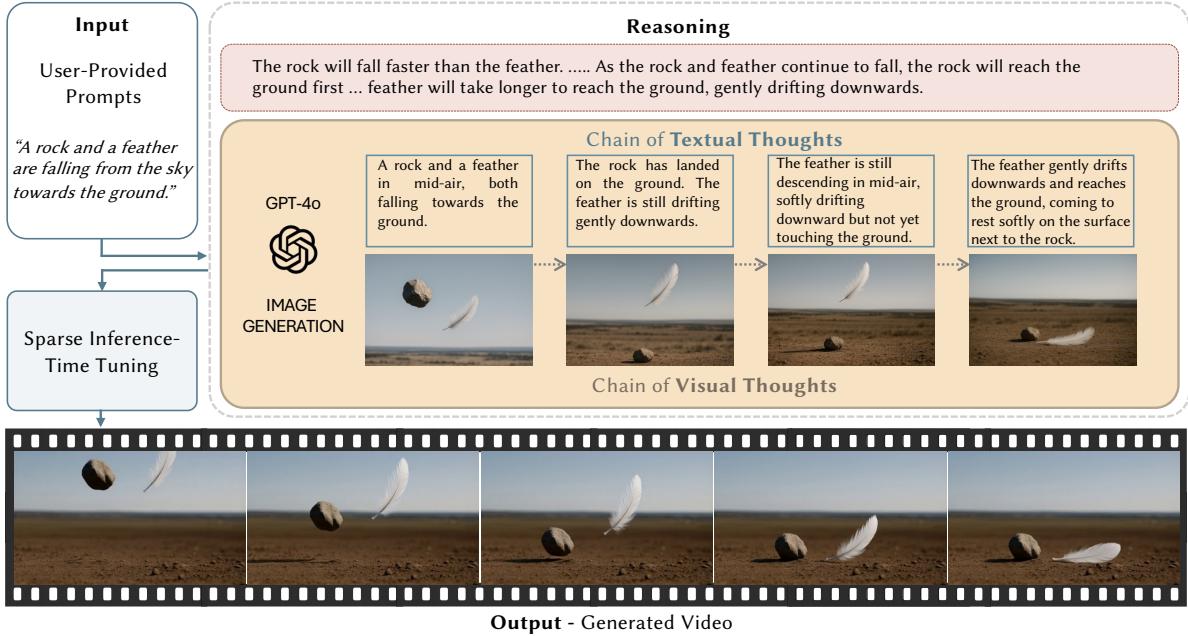
https://eyeline-labs.github.io/VChain

Figure 1: **Overview of VChain.** We introduce **VChain**, an inference-time tuning framework for reasoning in video generation. Given a user-provided prompt (*e.g.*, *"A rock and a feather are falling from the sky towards the ground."*), VChain leverages large multimodal models to generate a *Chain of Visual Thoughts*, which are a sparse set of causally important keyframes to guide the video generator via *Sparse Inference-Time Tuning*. VChain effectively improves reasoning in video generation without extensive re-training.

Abstract

Recent video generation models can produce smooth and visually appealing clips, but they often struggle to synthesize complex dynamics with a coherent chain of consequences. Accurately modeling visual outcomes and state transitions over time remains a core challenge. In contrast, large language and multimodal models (*e.g.*, GPT-4o) exhibit strong visual state reasoning and future prediction capabilities. To bridge these strengths, we introduce **VChain**, a novel inference-time chain-of-visual-thought framework that injects visual reasoning signals from multimodal models into video generation. Specifically, VChain contains a dedicated

pipeline that leverages large multimodal models to generate a sparse set of critical keyframes as snapshots, which are then used to guide the *sparse inference-time tuning* of a pre-trained video generator only at these key moments. Our approach is tuning-efficient, introduces minimal overhead and avoids dense supervision. Extensive experiments on complex, multi-step scenarios show that VChain significantly enhances the quality of generated videos.

1 Introduction

Video generation (Yang et al., 2024; Gen, 2024; kli, 2024; Kong et al., 2024; Wan et al., 2025; Team, 2025; Agarwal et al., 2025; Min, 2023) aims to synthesize coherent and realistic visual sequences, either from scratch or based on user-provided inputs such as text prompts, reference images, motion cues, or other forms of control. In recent years, this field has made remarkable progress, driven by powerful generative models such as diffusion mod-

- [✉]Corresponding Authors. [†]Project Lead.
- Ziqi Huang, Gordon Chen, Haonan Qiu, and Ziwei Liu are with *Nanyang Technological University*.
  Email: {ziqi002, chen2008, haonan002, ziwei.liu}@ntu.edu.sg
- Ning Yu and Paul Debevec are with *Eyeline Labs*.
  Email: {ning.yu, debevec}@scanlinevfx.com

els (Sohl-Dickstein et al., 2015; Song et al., 2021b; Ho et al., 2020), and supported by large-scale video datasets and increasing computational resources.

Modern video generation models have achieved impressive results in generating smooth and visually appealing video clips. However, they still struggle to reflect the intrinsic dynamics of the real world, especially when it comes to generating sequences that involve meaningful state transitions or coherent chains of consequences. As a result, current methods often fail to capture how visual states evolve over time in a logically consistent and causally grounded manner. For example, given a prompt like "a person drops a cup, it hits the ground, and the liquid splashes out," many models may render smooth deformations between frames but omit key causal steps, such as the cup deforming on impact or the splash propagating outward, resulting in scenes that are logically inconsistent or physically implausible.

In contrast, large language and multimodal models excel precisely in the areas where video generation models tend to struggle. Models such as GPT-4o (Hurst et al., 2024) have made rapid progress in general reasoning and cross-modal understanding. These models show strong capabilities in following instructions, multi-step reasoning, and aligning semantics across text and vision. Although they do not explicitly simulate visual dynamics over time, they are effective in inferring likely transitions between visual states. For instance, they can reason that if a glass tips over, it may shatter, or that if a person jumps, they will eventually land. This ability to suggest causally and logically consistent progressions offers a promising signal that current video generators lack. A natural question is raised: can we leverage this reasoning ability from large multimodal models to guide video generation models towards more coherent chains of visual consequences?

To this end, we propose **VChain**, a novel inference-time tuning framework that introduces high-level reasoning into video generation. The core idea is to represent the evolution of a scenario as a sparse sequence of *Visual Thoughts* - keyframes that capture critical intermediate states that a reasoning agent might anticipate. These visual thoughts are automatically generated using large multimodal models and serve as guidance signals for the video generator. VChain mainly consists of two main components. *1) Visual Thought Reasoning:* We design a dedicated

pipeline that leverages large multimodal models to decompose a user-provided text prompt into a concise set of causally important *Visual Thoughts*. These keyframes capture the intended chain of visual outcomes and act as a blueprint for the temporal structure of the video. *2) Sparse Inference-Time Tuning:* Then, the pre-trained video generator is quickly and efficiently fine-tuned using only the *Visual Thought* keyframes. The model is adjusted in a focused manner at these critical visual states, allowing it to capture the intended visual state transitions. Compared to tuning on video data, this approach is significantly faster and more practical for deployment.

As an inference-time tuning method, VChain offers several benefits. *(1) Self-contained:* All supervision is synthesized on the fly during inference by prompting a large multimodal model, with no need for external annotations, curated datasets, or retrieval systems. *(2) Efficient:* The tuning is only supervised by a few keyframes with limited iterations, and thus introduces minimal overhead relative to the cost of sampling the video itself. *(3) Effective:* We evaluate VChain on complex, multi-step video generation tasks that require strong causal reasoning. Across these scenarios, VChain consistently improves the dynamic fidelity of generated videos, leading to sequences that better reflect logical consequences, smooth transitions, and coherent visual narratives. Beyond a specific technique, VChain offers a new pathway: treating multimodal models as reasoning modules that complement, rather than replace, generative models in constructing causally coherent visual narratives. More generally speaking, VChain encourages the community to rethink how reasoning can be integrated into video generation - not through model retraining or dense supervision, but by transforming general-purpose multimodal intelligence into chain-of-visual-thought guidance at inference time.

In summary, our contributions are:

- We introduce **VChain**, a novel framework that uses chain-of-visual-thought from large multimodal models to bring high-level reasoning into video generation.
- We design the *Visual Thought Reasoning* pipeline, a GPT-guided pipeline that synthesizes sparse, causally grounded keyframes for guiding video generation.
- Extensive experiments demonstrate that sparse supervision on these keyframes im-

proves a model's ability to produce videos with coherent visual consequences and interpretable state transitions.

- Our method operates entirely at inference time, requires no external training data, and adds minimal computational overhead.

2 Related Work

2.1 Video Generation

Video generation has seen rapid progress (Yang et al., 2024; Gen, 2024; kli, 2024; Kong et al., 2024; Wan et al., 2025; Team, 2025; Agarwal et al., 2025; Min, 2023), driven by advances in diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2021b; Ho et al., 2020; Song et al., 2021a; Zhang and Agrawala, 2023; Blattmann et al., 2023; Esser et al., 2024; Mou et al., 2023; Ding et al., 2021, 2022; Ho et al., 2022), variational autoencoder-based compression (Kingma and Welling, 2013; Van Den Oord et al., 2017; Esser et al., 2021; Podell et al., 2023; Yu et al., 2023), and transformer-based backbones (Dosovitskiy et al., 2020; Peebles and Xie, 2022). Despite impressive progress in visual fidelity, smooth motion, and temporal alignment, most existing video generation methods remain limited to surface-level coherence (Zheng et al., 2025). They typically fail to capture deeper aspects such as causal dynamics, physical interactions, and meaningful state transitions. These models often overlook how actions lead to consequences, how objects behave under physical laws, or how scene states evolve with internal logic. To address this gap, we introduce an inference-time reasoning framework that injects high-level semantic supervision obtained from large multimodal models into the generation process. This approach enables pre-trained video generators to produce outputs that are not only visually plausible but also causally and physically grounded.

2.2 Multimodal Models for Understanding and Generation

Large language models (LLMs) like GPT-4 (OpenAI, 2023) and multimodal models such as Gemini (Team et al., 2023) and GPT-4o (Hurst et al., 2024) have shown strong capabilities in vision-language tasks, including instruction-following, visual question answering, and interactive reasoning. These models can perform reasoning about visual scenes, and more importantly, understand and generate grounded visual content. Recent works such as Transfusion (Zhou et al., 2024) incorporate these capabilities into multimodal pipelines for image generation. LMD (Lian et al., 2023a) and LVD (Lian et al., 2023b) leverage a large language model to generate coarse layouts to guide visual synthesis. However, these approaches typically treat LLMs as static prompt interpreters or high-level planners, or alternatively require dense retraining and architectural modifications. Unlike existing methods, we propose a lightweight inference-time reasoning framework for video generation that leverages off-the-shelf multimodal models and pre-trained video generators. Our method avoids dense retraining and instead injects high-level reasoning signals through sparse visual supervision, enabling more causally consistent and semantically grounded video generation with minimal overhead.

3 The VChain Framework

VChain is an inference-time reasoning framework designed to enhance the causal and physical coherence of video generation. Built on top of a pre-trained video generator, it aims to improve the model's ability to reflect reasoning, physics, causality, and commonsense understanding, producing videos that are more physically grounded and causally consistent.

As shown in Figure 2, the VChain framework has three key stages: *(1) Visual Thought Reasoning*, which uses a large multimodal model to infer key events and their consequences as a sparse sequence of visual snapshots; *(2) Sparse Inference-Time Tuning*, which injects these Visual Thoughts from stage 1 into the pre-trained video generator via lightweight LoRA adaptation; and *(3) Video Sampling*, which produces the final video by using both the stage-1 thoughts and the stage-2 tuned generator.

3.1 Preliminaries

Diffusion Models. Diffusion models are a class of generative models that reconstructs data $\mathbf{x}_0$ such as natural images or videos by iteratively denoising starting from the Gaussian prior $\mathbf{x}_T$. A widely used training loss (Ho et al., 2020) is $L_{\mathrm{DM}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\right\|^2\right]$, where $\mathbf{x}_t$ is a noisy image or video obtained by adding noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to the original visual $\mathbf{x}_0$. The network $\epsilon_\theta(\cdot)$ learns to estimate this added noise. To generate new data $\mathbf{x}_0$, the trained model $\epsilon_\theta(\cdot)$ de-

noises $\mathbf{x}_t$ iteratively from $t = T$ to $t = 0$, using the predicted noise at each step.

Video Diffusion Models. Our work builds on Wan (Wan et al., 2025), a state-of-the-art video generation foundation model trained on a mix of video and image datasets, supporting both video and image generation. Recent progress in diffusion-based video generation has been shifting from U-Net (Ronneberger et al., 2015) architectures to Diffusion Transformers (DiTs) (Peebles and Xie, 2022) with Flow Matching (Lipman et al., 2022). Wan adopts this newer paradigm, a design now common in text-to-video (T2V) systems (Kong et al., 2024). DiTs offer scalability advantages, while Flow Matching enables faster and more stable training convergence. Wan includes three main components: *1) Wan-VAE*: a spatio-temporal variational autoencoder; the *2) video diffusion transformer*, and the *3) text encoder*. Given a video $V \in \mathbb{R}^{(1+T) \times H \times W \times 3}$, Wan-VAE compresses it into VAE latent $x \in \mathbb{R}^{(1+T/4) \times H/8 \times W/8}$. The compression is spatial (by a factor of 8×8) for all frames, and temporal (by a factor of 4) for all frames except the first, which is only spatially compressed. The Wan video generation model uses the flow matching (Lipman et al., 2022; Esser et al., 2024) training objective in the Wan-VAE's latent space. Given the video (or image) latent $\mathbf{x}_1$ and noise $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noised latent is defined by the linear interpolation:

$$x_t = tx_1 + (1 - t)x_0, \quad (1)$$

where the timestep $t \in [0, 1]$ is sampled from a logit-normal distribution. The model is trained to predict the velocity:

$$v_t = \frac{dx_t}{dt} = x_1 - x_0, \quad (2)$$

using the objective:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, \mathbf{c}, t} \left\| u_\theta(\mathbf{x}_t, t, \mathbf{c}) - v_t \right\|^2, \quad (3)$$

where u_θ is the denoising model and $\mathbf{c}$ represents the embedded text prompt. The text encoder transforms the input text prompt into token embeddings, which we refer to as $\mathbf{c}$ for brevity.

Low-Rank Adaptation (LoRA). LoRA (Hu et al., 2022) is a parameter-efficient fine-tuning technique. It freezes the original model weights, and injects trainable low-rank decomposition matrices into network layers, largely reducing the number of trainable parameters. Specifically, for a pre-trained

---

Algorithm 1 Visual Thought Reasoning

```
1: given user-provided text prompt p
2:
3: % generate first frame
4: txt, consequence = chat(p)
5: img = image_generate(txt)
6: chain_vis = [img] % init chain-of-visual-thought
7: chain_txt = [txt] % init chain-of-textual-thought
8:
9: % iteratively generate subsequent frames
10: repeat
11:     txt, flag = perception(chain_vis, consequence, p)
12:     img = image_edit(chain_vis, txt)
13:     chain_vis.append(img)
14:     chain_txt.append(txt)
15: until flag==terminate
16:
17: return chain_vis, chain_txt
```
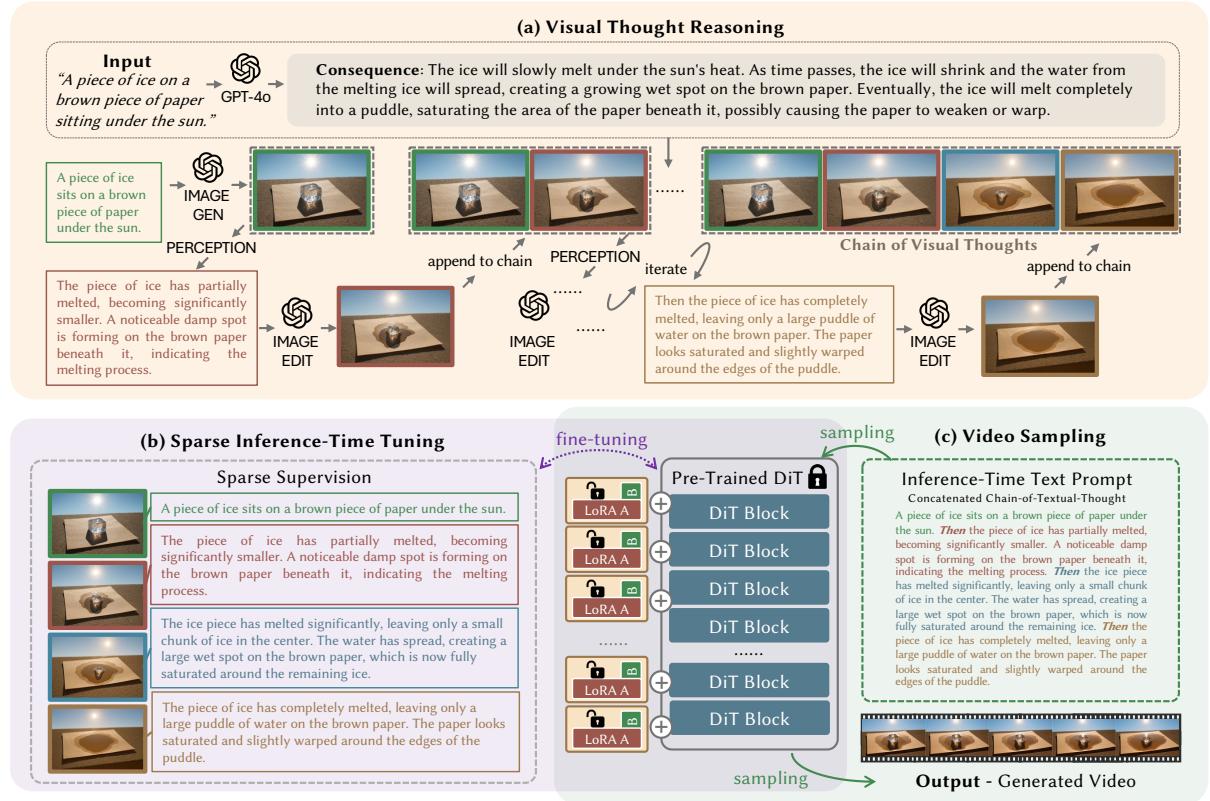
weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA reparametrizes the update as $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$. Only A and B are updated during training. Given an input x, the modified forward computation is $h = W_0 x + \Delta W x = W_0 x + BAx$. Because of the low-rank property, LoRA offers both computational and memory efficiency, making it a strong fit for fine-tuning large video diffusion models.

3.2 Visual Thought Reasoning

Given a user-provided text prompt $\mathbf{p}$ for video generation, we leverage the powerful multimodal reasoning capabilities of GPT-4o (Hurst et al., 2024) to generate a sequence of images, referred to as the *Chain of Visual Thoughts*, that capture the key moments of the intended video. The steps and definitions of *Visual Thought Reasoning* are listed in Algorithm 1.

We first prompt GPT-4o to reason about the likely outcome implied by the user input prompt $\mathbf{p}$. As illustrated in Figure 2, given a prompt, "A piece of ice on a brown piece of paper sitting under the sun", GPT-4o infers that the ice will melt due to the heat, forming a puddle that soaks the paper. This step establishes the ground-truth trajectory of the intended video, referred to as the *consequence*, which serves as the basis for constructing the key transitions of the unfolding scene.

We then instruct GPT-4o to generate a caption $\mathbf{txt}_0$ describing the first frame in the *Chain of Visual Thoughts*, which is transformed into an image $\mathbf{img}_0$ using GPT-4o's native image generation module. After that, GPT-4o predicts an editing instruction $\mathbf{txt}_i$ to produce the key moment at time step i in our chain, conditioned

4

Figure 2: **VChain Framework.** An overview of our three-stage inference-time pipeline for reasoning in video generation. **(a) Visual Thought Reasoning:** Given a user-provided text prompt, a large multimodal model (GPT-4o) infers a causal chain of events and generates a sequence of keyframes, termed the *Chain of Visual Thoughts*, via iterative reasoning and image synthesis. **(b) Sparse Inference-Time Tuning:** These visual thoughts (paired with their corresponding textual thoughts) serve as sparse supervision for fine-tuning a pre-trained video generator via LoRA. **(c) Video Sampling:** The full sequence of textual thoughts is concatenated to form a single prompt, which is used to prompt the fine-tuned model in generating the final video output.

on **p**, the *consequence*, and the *Chain of Visual Thoughts* at the current timestep, $\mathbf{chain}_{vis} = [\mathbf{img}_0, \mathbf{img}_1, \ldots, \mathbf{img}_{i-1}]$. Then $\mathbf{txt}_i$ is used to generate the subsequent image $\mathbf{img}_i$. This process continues iteratively, where GPT-4o predicts an editing instruction and generates a corresponding image, and terminates only when the *consequence* has been fully captured by $\mathbf{chain}_{vis}$.

The resulting output is a coherent sequence of keyframes, or *Chain of Visual Thoughts* [$\mathbf{img}_0$, $\mathbf{img}_1, \ldots, \mathbf{img}_{N-1}$] paired with its corresponding textual thoughts [$\mathbf{txt}_0$, $\mathbf{txt}_1$, $\ldots$, $\mathbf{txt}_{N-1}$], that captures the temporal evolution implied by the user prompt. This approach also allows users to generate causally consistent image sequences without having to explicitly anticipate or specify the underlying consequences of the described scenario. Please refer to the *Appendices* for detailed descriptions of the *Visual Thought Reasoning* process, including system prompts, intermediate outputs, and workflow details.

3.3 Sparse Inference-Time Tuning

Given the sparse and causally grounded *Chain of Visual Thoughts* generated from the previous stage, we perform lightweight inference-time tuning on a pre-trained video generator. We only use these keyframes as supervision, treating them as anchor points that encode important state changes (*e.g.*, melting, breaking, or object movement).

Formally, let $\mathbf{chain}_{vis} = [\mathbf{img}_0, \mathbf{img}_1, \ldots, \mathbf{img}_{N-1}]$ be the sequence of N *Visual Thoughts* (keyframes), and $\mathbf{chain}_{txt} = [\mathbf{txt}_0, \mathbf{txt}_1, \ldots, \mathbf{txt}_{N-1}]$ be their corresponding *Textual Thoughts*. Each $\mathbf{img}_i$ is treated as a one-frame video, paired with the caption $\mathbf{txt}_i$. These pairs $(\mathbf{img}_i, \mathbf{txt}_i)$ serve as the training data for tuning the video diffusion model using the same flow-matching objective as Equation 3:

$$\mathcal{L}_{vchain}(\theta) = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, \mathbf{c}, t} \|u_\theta(\mathbf{x}_t, t, \mathbf{c}) - v_t\|^2, \quad (4)$$

where $\mathbf{x}_1 = \mathbf{img}_i$, $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$ is sampled from a logit-normal distribution, $\mathbf{x}_t =$

Table 1: **Quantitative Evaluation.** VChain is compared with existing methods and ablation variants, achieving comparable or superior performance across all evaluation metrics.

Method	VBench Quality Score	Frame Quality	Temporal Smoothness	Video-Text Alignment	Physics	Commonsense Reasoning	Causal Reasoning
T2V	76.21%	57.24%	43.65%	40.04%	32.03%	32.42%	32.81%
T2V + Prompt Aug	77.51%	55.47%	50.59%	47.66%	38.09%	38.48%	41.99%
Without Visual Thought	78.47%	64.26%	52.93%	54.69%	44.14%	43.75%	47.51%
Without Sparse Tuning	73.35%	44.07%	29.19%	42.97%	33.24%	34.57%	34.46%
VChain (Ours)	**78.49%**	**71.67%**	**65.82%**	**67.77%**	**58.01%**	**60.16%**	**62.12%**

$t\mathbf{x}_1 + (1 - t)\mathbf{x}_0$ as in the flow-matching setup, and $\mathbf{c}$ is the text embedding of $\texttt{txt}_i$.

This sparse tuning scheme offers two key benefits: *1) Focused supervision*: By concentrating only on keyframes that encode the critical moments (*e.g.*, object breaking, melting, or appearing), we guide the model to focus on inferring causal outcomes and key visual state transitions. *2) Efficiency*: Since the tuning is *image-only*, tuning is fast and memory-efficient. This makes our method practical for inference-time adaptation. Additionally, our tuning does not require additional databases or labels. The entire supervision signal is generated internally from the *Visual* and *Textual Thoughts* (Section 3.2), making VChain easily pluggable into general pre-trained video generators.

3.4 Video Sampling

Following *Sparse Inference-Time Tuning*, we concatenate every textual thought $\texttt{txt}_i$ from the *Chain of Textual Thoughts* $\texttt{chain}_{txt}$ into a single composite prompt $\texttt{txt}_{concat}$. This final prompt is used as the input to the fine-tuned video generator to produce the output video. The resulting generation reflects both the inferred sequence of events and the adapted capabilities of the model.

4 Experiments

4.1 Experimental Setup

For *Visual Thought Reasoning*, we use the GPT family (Hurst et al., 2024) as our large multimodal model. Specifically, we use `gpt-4o` for chat and perception, and `gpt-image-1` for steps involving image generation and editing. Our main experiments are conducted using the state-of-the-art pre-trained video generator Wan2.1-T2V-1.3B (Wan et al., 2025). We design 20 diverse test scenarios for both human evaluations and quantitative comparisons. We list the implementation details, test cases, and the time cost breakdown in *Appendices*.

4.2 Comparison Methods

We compare our proposed method VChain against several baselines and ablation variants.

Baseline Comparison. We include the following baselines:

- *T2V:* The original pre-trained text-to-video generation model without any modification.
- *T2V + Prompt Aug:* The input text prompt is enhanced using GPT-based prompt augmentation.

Ablation Study. To further understand the impact of each component in VChain, we design the following ablation settings:

- *Without Visual Thought:* We use our *Visual Thought Reasoning* pipeline to produce both composite text prompts $\texttt{txt}_{concat}$ and visual thoughts $\texttt{chain}_{vis}$, but only feed $\texttt{txt}_{concat}$ to the video generator, omitting the visual thoughts for sparse tuning. This ablation evaluates the necessity of performing chain-of-thought reasoning *visually*, showing that text-only thoughts are insufficient for reasoning in video generation.
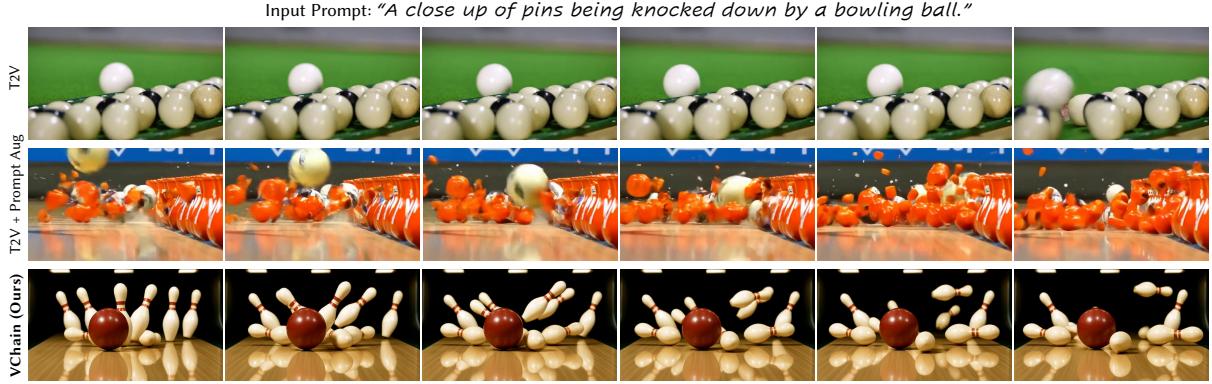
- *Without Sparse Tuning:* We use GPT-generated keyframes as-is for direct video interpolation, without fine-tuning the video generator. This variant evaluates the necessity of sparse tuning to align the dynamics with the inferred reasoning signals.

- *VChain (Ours):* Our full framework, which combines both *Visual Thought Reasoning* and *Sparse Inference-Time Tuning* to enable reasoning in video generation.

4.3 Quantitative Comparisons

We introduce the aspects used for experimental comparisons.

VBench Quality Score. To evaluate VChain's impact on fundamental video quality, independently of its reasoning or causal capabilities, we conduct quantitative evaluations using VBench (Huang et al., 2024a,b), an evaluation framework designed to assess key technical dimensions of video generation, such as frame-level fidelity, temporal con-

Input Prompt: *"A close up of pins being knocked down by a bowling ball."*

Figure 3: **Qualitative Results - Baseline Comparison.** *T2V* fails to capture the key causal interaction: the pins remain mostly static or jitter slightly, with no meaningful collision, revealing a lack of physical reasoning despite temporal coherence. *T2V + Prompt Aug* introduces relevant elements and motion, but the dynamics are erratic and implausible. Pins deform unnaturally, visual artifacts appear, and later frames become unstable, indicating poor spatial consistency. In contrast, *VChain (Ours)* produces a coherent and physically realistic sequence: the ball strikes the pins with plausible force, leading to consistent knockdown. Object geometry and material properties are well preserved across frames. These results show that VChain not only enables causal reasoning about the outcome of physical interactions, but also stabilizes spatial transitions.

sistency, and motion dynamics *etc*. As shown in Table 1, VChain achieves comparable or slightly better scores than both the original pre-trained generator and other baselines.

We also perform complementary human evaluations focused on three core aspects of video quality: **Frame Quality.** Evaluates the visual quality of individual frames, including aesthetics, imaging sharpness, and realism.
Temporal Quality. Assesses motion smoothness, temporal consistency, and overall dynamic realism across frames.
Video-Text Alignment. Evaluates how faithfully the generated video reflects the user-provided text prompt.

While VChain is primarily designed to enhance high-level reasoning in video generation (*e.g.*, commonsense, causality, and physics), the results shown in Table 1 confirm that it does not compromise basic visual quality. In fact, it often brings modest improvements. To directly assess VChain's reasoning capabilities, we conduct targeted human studies along the following dimensions:
Physics. Evaluates whether the video follows physical laws, like gravity and air friction (*e.g.*, rocks fall faster than feathers in the air). Participants rate how well the video obeys the laws of physics.
Commonsense Reasoning. Assesses whether events in the video reflect everyday real-world knowledge. For instance, blue paint mixed with yellow turning green, or oil floating on water. Users rate how well the video reflects common sense.

Causal Reasoning. Evaluates whether the video captures appropriate cause-and-effect relationships. Examples include a stone causing a splash when dropped in water, a ball failing to bounce on a pillow, or a switch turning on light. Participants are asked: "How well does the video reflect the causal consequences of the initial setup?"

Human evaluators were presented with generated videos alongside their corresponding input prompts. The outputs from our method and the baselines were shown in randomized order to avoid bias. A total of 32 evaluators rated each video on a scale from 1 to 5 for each evaluation dimension. The scores are then averaged and normalized to a percentage scale, as reported in Table 1.

VChain consistently outperforms the baseline methods, particularly in reasoning-related dimensions such as physics, commonsense, and causality. These improvements demonstrate the effectiveness of the integration of our framework in inference-time reasoning for video generation.

4.4 Qualitative Comparisons

Extensive qualitative results and comparisons are also provided in the *Appendices*.
Baseline Comparison. We present qualitative comparisons against baseline methods in Figure 3. In the *T2V* baseline, the model fails to produce any meaningful physical interaction: the pins remain mostly static or exhibit minor jittering, with no visible impact or knockdown. Although temporally stable, the output is semantically misaligned

7

Input Prompt: *"POV: You are catching a ball thrown by a friend."*

Figure 4: **Qualitative Results - Ablation Study.** We compare VChain with two ablated variants. *(1) Without Visual Thought:* Although the model recognizes that the video should be in a first-person perspective based on the textual prompt, it fails to capture the correct visual pattern for a ball-catching viewpoint. In contrast, VChain leverages the reasoned Visual Thoughts to render step-by-step intermediate visual states of the throw-and-catch process. *(2) Without Sparse Tuning:* While Visual Thoughts are included, the model performs direct frame interpolation without tuning, leading to warping artifacts due to spatial misalignments among individual frames in Visual Thoughts. VChain (Ours) produces the most coherent and physically grounded interaction, correctly depicting the ball being thrown and caught from a first-person perspective. Removing either component degrades video synthesis quality.

with the input prompt, lacking the key causal event of a bowling ball knocking down pins. The *T2V + Prompt Aug* variant introduces the ball and pins, showing some degree of collision and motion. However, the dynamics are chaotic and physically implausible. Pins deform or scatter in erratic ways, and the scene suffers from visual artifacts and temporal instability, particularly in later frames. In contrast, *VChain (Ours)* produces a coherent and physically grounded sequence. The bowling ball hits the pins with a realistic impact, and the pins fall in directions consistent with expected physical behavior. This outcome is enabled by chain-of-visual-thought reasoning, which provides the model with a structured, causal progression of events. Furthermore, object geometry and material properties are well preserved. Pins and the ball are visually distinct and accurately rendered.

Ablation Study. In Figure 4, we compare VChain with two ablated variants: *1) Without Visual Thought*, while it understands that the first-person perspective should be generated from the *Textual Thoughts*, it fails to envision the correct visual pattern of a ball-catching POV. In contrast, our method benefits from directly "seeing" the Visual Thoughts, enabling accurate spatial understanding and rendering of the interaction. *2) Without Sparse Tuning*, which includes Visual Thoughts directly performs frame interpolation, and warping artifacts emerge when attempting to bridge spatial misalignments between Visual Thought keyframes. *VChain (Ours)* produces the most coherent and physically

grounded interaction, accurately depicting the ball being thrown and caught. Removing either component leads to degraded video synthesis.

Figure 11(b) in *Appendices* highlights another example of a rubber duck and a rock falling into water. Without *Visual Thought*, the duck appears submerged in water, violating the basic physical intuition that rubber ducks are supposed to float. In contrast, our method correctly depicts the duck floating on the water's surface. This underscores the importance of having *Visual Thoughts* (versus *Textual Thoughts* only) at inference time: it's important to view the *Visual Thoughts* during inference - to actually "see" how the rubber duck floats on the water surface rather than sinks. Our demo video provides a more intuitive comparison.

5 Limitations

5.1 Limitations of Visual Thought Generation

Our framework inherits some limitations from the current state of the GPT-4o image generation model.

First, we observe that `gpt-image-1` tends to slightly oversaturate and over-smooth edited images. Since each generated image is passed back into the model as input to produce the next frame in the image sequence, this effect can accumulate iteratively, leading to a yellow color cast and over-smoothness across the image sequence. The artifact slightly undermines the photorealism of later frames and introduces slight color inconsistencies

across the sequence. Qualitative examples are provided in the *Appendices*.

Another limitation is the API cost. Each generated keyframe requires two calls to GPT-4o. Thus, the total number of API calls scales linearly with the image sequence length, and the token consumption would be quadratic. Hence, the reliance on proprietary models might limit accessibility and reproducibility for those without access to sufficient compute budget or API quotas. Despite these concerns, the overhead in practice is modest: inference-time reasoning for a video typically requires only 3-6 images, which keeps the cost relatively low.

5.2 Limitations of Sparse Inference-Time Tuning

Our method fine-tunes a pre-trained video generator using only a small number of keyframes, referred to as *Visual Thoughts*, as supervision. This sparse tuning introduces an inherent trade-off: optimizing too strongly on static keyframes may reduce motion dynamics, since the model adapts primarily to still images, while insufficient optimization may weaken the reasoning signals injected into the generator, producing results closer to the untuned baseline.

Despite the potential trade-off, this *sparse* tuning strategy offers two main advantages: *(1) Focused adaptation*: the model concentrates its capacity on semantically critical transitions (*e.g.*, melting, breaking, or object interactions) rather than reconstructing entire video sequences. *(2) Efficiency*, as it eliminates the need for dense videos, significantly reducing both data preparation and computational overhead. This makes our approach well-suited for inference-time integration into existing pipelines.

Overall, while sparse supervision cannot fully capture the dynamics in video samples, the improvements in semantic alignment and causal coherence generally outweigh the loss in dynamics. This paradigm also challenges the conventional assumption that full video sequences are required for fine-tuning, showing that a carefully selected set of keyframes can already provide sufficient guidance for adapting video generators to new prompts or scenarios.

6 Ethical Considerations

While both large multimodal models and video generators can produce vivid and compelling content, users should exercise caution when using AI-generated media. Outputs may inherit and amplify safety concerns and biases from the multimodal models and video generators they rely on. We strongly advocate for the responsible and ethical use of generative models.

Potential Risks. VChain is intended as a research contribution, but its ability to improve causal and physical coherence also increases the realism of synthetic videos. This realism could be misused for harmful purposes such as producing disinformation, deepfakes, or fabricated evidence. Moreover, because VChain depends on large multimodal models and pretrained generators, it might propagate their biases into more coherent video narratives, which may reinforce stereotypes or exclusion. We emphasize that VChain is designed for controlled research and creative exploration, not deployment in sensitive or adversarial settings.

7 Conclusion

In this work, we present **VChain**, a general inference-time framework that integrates multimodal reasoning into video generation. By representing a scenario as a sparse sequence of *Visual Thoughts* - keyframes capturing critical intermediate states inferred by large multimodal models - VChain injects causal and commonsense reasoning signals directly at inference time. This paradigm enables video generators to model meaningful state transitions without dense annotations or costly retraining. Experiments on complex, multi-step scenarios show that VChain substantially improves the coherence, causal consistency, and rationality of generated videos, while maintaining efficiency and visual quality. More broadly, VChain demonstrates how the reasoning capabilities of large multimodal models can be effectively combined with the rendering and motion priors of video generators. We view this framework as a step toward bridging reasoning and generation, and hope to inspire further research on reasoning for video generation.

Acknowledgments

References

2022. Langchain. Accessed March 31, 2025 [Online] https://www.langchain.com.

2023. Minmax team. Accessed August 31, 2024 [Online] https://hailuoai.com/.

2024. Gen-3. Accessed June 17, 2024 [Online] https://runwayml.com/research/introducing-gen-3-alpha.

2024. Kling. Accessed December 9, 2024 [Online] https://klingai.kuaishou.com/.

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, and 1 others. 2025. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, and 1 others. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and 1 others. 2021. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*.

Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. 2022. Cogview2: Faster and better text-to-image generation via hierarchical transformers. In *NeurIPS*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, and 1 others. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, and 1 others. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, and 1 others. 2024a. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2024b. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, and 1 others. 2024. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. 2023a. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. 2023b. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. 2023. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

William Peebles and Saining Xie. 2022. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent

diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising diffusion implicit models. In *ICLR*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021b. Score-based generative modeling through stochastic differential equations. In *ICLR*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

StepFun Team. 2025. *Preprint*, arXiv:2502.10248. [link].

Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. In *NeurIPS*.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, and 1 others. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, and 1 others. 2024. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and 1 others. 2023. Magvit: Masked generative video transformer. In *CVPR*.

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. 2025. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. 2024. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*.

Appendices

We provide additional implementation details in Appendix A, and qualitative results in Appendix B. A demo video is also available at this link.

A Additional Implementation Details

A.1 Implementation Details of Visual Thought Reasoning

Given a user-provided input prompt describing a video, our *Visual Thought Reasoning* pipeline synthesizes a sequence of keyframes which form the crucial moments of the video. The implementation details are as follows.

We first prompt GPT-4o's chat completions API with the system message shown in Figure 7 to instruct the model to reason about the video's likely spatial layout and anticipated causal consequences based on the user-provided input prompt. We employ LangChain (lan, 2022) to convert GPT-4o's unstructured textual outputs into structured schema-aligned responses containing:

1. *Context Frame:* A richly detailed prompt used to generate the first frame in the *Chain of Visual Thoughts*.
2. *Concise Prompt:* A concise version of the Context Frame prompt (the full version is too long, so the first image is paired with this concise prompt during sparse inference-time tuning).
3. *Consequences:* A sequence of inferred physical outcomes that define the expected trajectory of the generated video.

The Context Frame is passed to GPT's `gpt-image-1` API to produce the corresponding image.

To generate subsequent keyframes in the *Chain of Visual Thoughts*, we concatenate all previously generated images in our chain into a single composite image, as shown in Figure 2. This stitched chain of images, together with the user input prompt and the inferred consequences, is passed to GPT-4o's chat completion API using the system message in Figure 8. GPT-4o predicts the next key moment in the sequence. Specifically, the output contains: 1) an image-editing instruction and 2) a boolean flag indicating whether a terminal state has been reached. We pass the same inputs as before along with the editing instruction to the `gpt-image-1` API to generate the next keyframe. We repeat this process iteratively, where we predict the next key moment and generate the corresponding image, until the boolean flag signals that the full sequence of consequences have been realized by the chain.

All outputs, including keyframe captions and reasoning chains, are stored in a structured JSON file (see Figure 9). We then generate a CSV file where each row contains an image file path and its corresponding caption, forming the image-text pairs used to fine-tune the video generation model. The first image is paired with the concise prompt, while each subsequent image is paired with its keyframe (example in Figure 10).

Figure 5 shows an example of *Chain of Visual Thoughts* generated by our pipeline.

A.2 Time Cost

Table 2 summarizes the average runtime of each stage in VChain, providing a detailed breakdown of the overall computational cost.

A.3 Implementation Details of Sparse Inference-Time Tuning

Our main experiments are conducted using the state-of-the-art pre-trained video generator Wan2.1-T2V-1.3B (Wan et al., 2025). We use the learning rate of $1e-4$, and fine-tune with a `train_lora_rank` of 16, and `train_lora_alpha` of 16.

A.4 Details of Test Cases

We design twenty test cases to support both human and quantitative evaluations. Each case depicts a simple, physically grounded scenario that requires causal reasoning to generate coherent outcomes.

- *A rock and a feather falling from the sky towards the ground.*
- *An egg falling from the sky towards concrete ground.*
- *An ice cream cone is left out in the sun.*
- *A rubber duck and a rock fall into a water tank.*
- *A steel ball is dropped into water.*
- *Milk is poured into a cup of black coffee.*
- *A man falls off a pile of bricks.*
- *A steel ball falling through the air onto ice.*
- *A ball is dropped onto a pillow.*
- *A sandwich rotting over time.*
- *An elderly blows out a cake filled with candles.*
- *Red and yellow paint are mixed together with a brush.*
- *Concentrated sulfuric acid is poured onto a wooden table.*

Figure 5: **Example of Visual Thoughts.** We show the reasoned Visual Thoughts of the input prompt: "Concentrated sulfuric acid is poured onto a wooden table". The sequence illustrates our pipeline's inferred causal progression across keyframes.

Figure 6: **GPT Keyframe Limitations.** Qualitative examples showing the accumulated saturation and smoothness artifacts produced by gpt-image-1 during iterative keyframe generation. As each generated image is recursively used as part of the input for the next step, slight over-saturation and over-smoothing compound over time, leading to slight color shifts (*e.g.*, yellow cast) and reduced photorealism in later frames.

- *An egg is dropped onto a pillow.*
- *A mailbox rusting over time in broad daylight.*
- *A man blows into a deflated balloon.*
- *Oil is poured into a glass of milk.*
- *A chameleon eats a flying insect.*
- *Blue and yellow paint are mixed together with a brush.*
- *A cup of water is falling towards the ground on its side.*

B Additional Qualitative Results

We present additional qualitative examples illustrating the saturation limitations of Visual Thought Generation in Figure 6.

Further qualitative comparisons are shown in Figures 13, 14, 15, 16, 17, 18, and 19.

First Frame System Message

You are given a text prompt, which describes a video. You are to perform the following tasks:

1. Infer the objects/people/elements present in the scene, the perspective of the camera, the spatial relationship between the objects in the scene as well as details not explicitly mentioned in the text prompt.

2. Create a detailed, movie-like description of the scene that evokes visuals with strong detail and composition cues. This is the Context Frame. It should clearly describe the objects/people/elements present in the scene, the perspective of the camera, and the spatial relationships between the objects in it as well as the details not explicitly mentioned in the text prompt.

The context frame must depict the initial state of the scene, before any action occurs, and must not foreshadow the input prompt. Given an input prompt "A man throws a ball", the context frame should depict a man holding a ball at his side, not in mid-throw. Given an input prompt "A man squeezes a ball in his hand", the context frame should depict a man gently holding a ball in his palm, not squeezing it yet. Given an input prompt "A dolphin emerges from the water", the context frame should depict a calm ocean. The dolphin should not be visible yet. The context frame should be written as if it is depicting an image, not a video. Hence, it should not foreshadow what will happen next.

3. Create a concise version of the context frame. This should be a short, one-sentence description of the context frame.

4. Infer a sequence of consequences/changes from the text prompt, even if it is not explicitly mentioned. Use assertive languange to clearly describe the changes in appearance, shape, color, size, and position that may occur as a result.

Example:

————

Input Prompt: "A cat pushes a glass of water off a table."

Thoughts: In order for the cat to tip the glass off the table, the cat is sitting on the table next to the glass of water. In order for the glass of water to fall off the table, it should be placed precariously on the edge of the table. A side view perspective would capture the table, the cat, the glass of water in one frame.

Context Frame: A side view of a sleek tabby cat sitting upright on a wooden table in a kitchen. The glass of water is placed precariously at the very edge of the table. The cat gazes intently at the glass, its tail curled around its body. The camera is at mid-height, framing the cat, table, glass, and floor clearly in the shot.

Concise Prompt: A cat sits next to a glass of water on a table.

Consequences: The cat will touch the glass of water, causing it to tip over the edge of the table and fall towards the ground. The glass of water will touch the ground and shatter as a result. The water will spill everywhere and glass shards will be on the floor.

———— Additional Examples ————

Figure 7: **First Frame System Message.**

Next Frame System Message

You are given an input prompt, which describes a video. You are also given a sequence of keyframes (1 or more keyframes), meant to depict key moments of the video. You are also given a hint, describing what happens throughout the video. You are to predict the next keyframe in the sequence. Use precise language to clearly describe the changes that may occur in this next key. You must predict what may happen within the next 5 seconds of the video. Hence, do not predict too far into the future.

A keyframe is a still image that captures either the start, peak/intermediate stage, the end, or the consequence of an event. The next predicted keyframe MUST ONLY depict either the start, peak, end, or consequence of an event and NEVER a combination of them.

e.g. The key moments of kicking a ball into a goal are (1) The moment the foot makes contact with the ball. The ball should not have moved at this point., (2) The moment the ball is inside the goal. e.g. The key moments of ice melting are (1) When the ice is fully solid (2) The moment the ice cube is half melted (3) The moment the ice cube is completely melted with a large puddle of water. e.g. The key moments of a glass of water falling off a table are (1) The moment the glass of water is on the edge of the table, (2) The moment the glass of water is falling midair towards the ground (3) The moment the glass of water makes contact with the ground but is still in one piece. (4) The moment the glass of water shatters on the ground and the water spills everywhere.

If the next key moment involves contact between two objects, then the next keyframe must depict the moment of contact. The objects must be touching in the next predicted key rame description. Your caption for the next keyframe should not use comparative language to describe a relative change in position, distance, or size (e.g. towards, away from). Instead, it should describe the absolute position, distance, or size of the objects involved. If possible, use spatial prepositions to clarify the relationship between objects (e.g. inside of, on top of, and below). The next keyframe should describe the image as if it not in motion. Hence, avoid using phrases like 'about to', 'going to'. Describe the next keyframe as if it is a still image.

Finally, return 'True' if the next predicted keyframe is the last frame of this video, otherwise, return 'False'. If nothing significant happens after the next predicted keyframe, return 'True'.

Example:

————

Input Prompt: "A cat pushes a glass of water off a table"

Hint: The glass of water will fall off and shatter on the floor. The water will spill everywhere and glass shards will be on the floor.

Keyframe 1: [A cat is sitting on the table, and the glass of water is on the edge of the table.]

Next Predicted Keyframe: The cat's paw is touching the glass of water sitting on the table.

Last Frame: False

————— Additional Examples —————

Figure 8: **Next Frame System Message.**

```
    "sulfuric_acid": {
        "input_prompt": "Concentrated sulfuric acid is poured onto a wooden
            table.",
        "thoughts": "The scene involves a wooden table, likely in a laboratory
            or workshop setting, where concentrated sulfuric acid is about to
            be poured. The acid is typically stored in a glass or plastic
            container, and the person pouring it might be wearing protective
            gear such as gloves and goggles. The camera should capture a side
            view to show the table, the container of acid, and the person
            pouring it. The table is initially dry and intact, with visible
            wood grain.",
        "consequences": "As the sulfuric acid is poured onto the wooden table,
            it will react with the wood, causing it to char and emit smoke.
            The wood will darken and potentially start to disintegrate where
            the acid makes contact, creating a burnt, uneven surface. The
            reaction may produce heat and release fumes, necessitating proper
            ventilation and safety precautions.",
        "context_frame": "In a well-lit laboratory, a sturdy wooden table
            stands at the center of the scene, its surface smooth and polished
            , with visible wood grain patterns. A person, wearing protective
            gloves and goggles, stands beside the table, holding a glass
            container filled with concentrated sulfuric acid. The container is
             tilted slightly, poised to pour. The camera captures a side view,
             framing the table, the container, and the person, highlighting
            the contrast between the clear, viscous liquid and the warm tones
            of the wood.",
        "concise_prompt": "A person stands beside a wooden table, holding a
            container of concentrated sulfuric acid.",
        "key_frames": [
            "The area where the concentrated sulfuric acid makes contact with
                the wooden table starts to darken and emit smoke. The wood
                grain appears charred and blackened, with visible smoke rising
                 from the surface. The edges of the darkened area are
                irregular, indicating the beginning of disintegration.",
            "The concentrated sulfuric acid has been poured onto the wooden
                table. A small, blackened, and charred area is visible on the
                table where the acid has made contact. Smoke is rising from
                the reaction site, and the wood grain around the area has
                started to darken and disintegrate slightly, illustrating the
                corrosive impact of the acid.",
            "The concentrated sulfuric acid creates a deep, blackened mark on
                the wooden table where it has been poured. The wood is
                significantly charred with smoke wafting upwards, forming a
                small plume. The surrounding area of the wood appears darker,
                with slight disintegration at the center of the spill,
                indicating intense chemical reaction.",
            "The concentrated sulfuric acid has reacted with the wooden table,
                 and the area of contact has expanded. The wood appears
                darkened and severely burnt, with visible smoke and fumes
                rising prominently into the air. The wooden surface is visibly
                 damaged, with large burnt patches and disintegrated material,
                 showing an uneven and charred texture. The person remains
                focused on observing the reaction, and the container of acid
                is still slightly tilted above the table."
        ]
    }
```

Figure 9: **Reasoning Output Example.**

<div style="border:1px solid green; padding:10px;">

CSV Output Example

```
"file_name","text"
"sulfuric_acid_0.png", "A person stands beside a wooden table, holding a container of
    concentrated sulfuric acid."
"sulfuric_acid_1.png","The area where the concentrated sulfuric acid makes contact with
    the wooden table starts to darken and emit smoke. The wood grain appears charred and
    blackened, with visible smoke rising from the surface. The edges of the darkened area
    are irregular, indicating the beginning of disintegration."
"sulfuric_acid_2.png","The concentrated sulfuric acid has been poured onto the wooden
    table. A small, blackened, and charred area is visible on the table where the acid has
     made contact. Smoke is rising from the reaction site, and the wood grain around the
    area has started to darken and disintegrate slightly, illustrating the corrosive
    impact of the acid."
"sulfuric_acid_3.png","The concentrated sulfuric acid creates a deep, blackened mark on
    the wooden table where it has been poured. The wood is significantly charred with
    smoke wafting upwards, forming a small plume. The surrounding area of the wood appears
     darker, with slight disintegration at the center of the spill, indicating intense
    chemical reaction."
"sulfuric_acid_4.png", "The concentrated sulfuric acid has reacted with the wooden table,
    and the area of contact has expanded. The wood appears darkened and severely burnt,
    with visible smoke and fumes rising prominently into the air. The wooden surface is
    visibly damaged, with large burnt patches and disintegrated material, showing an
    uneven and charred texture. The person remains focused on observing the reaction, and
    the container of acid is still slightly tilted above the table."
```
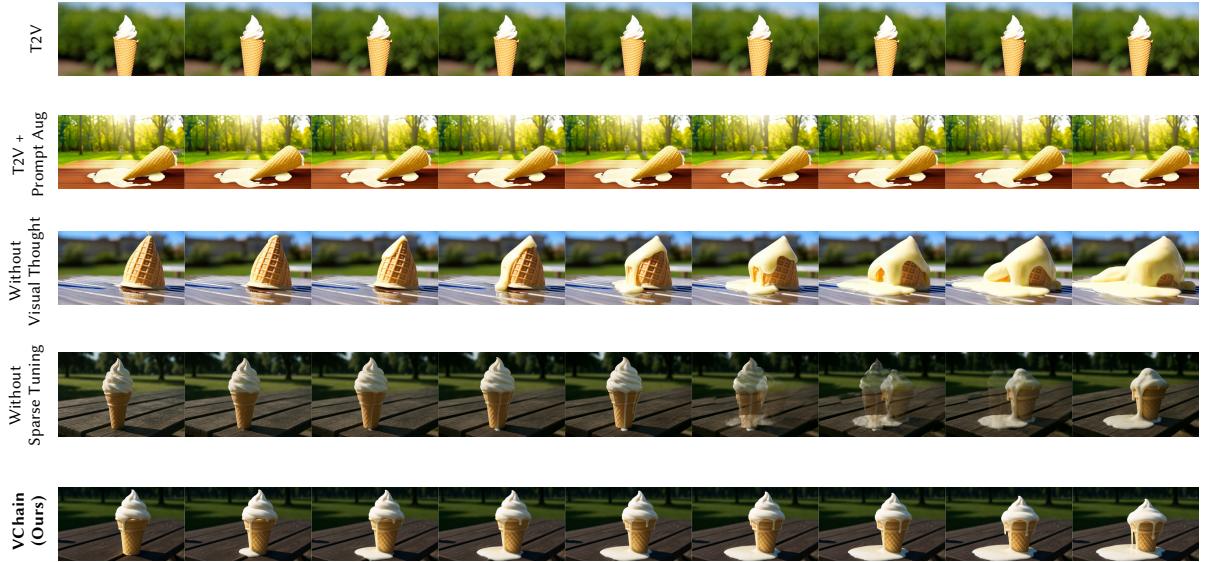
</div>

Figure 10: **CSV Output Example.**

Table 2: **Time Cost Breakdown.**

Breakdown	Time Cost	Comments
Visual Thought Reasoning	3 min 3 sec	
initial reasoning	14 sec	API: *gpt-4o* chat completions, called once for every sequence, CPU
image generation	1 min 7 sec	API: gpt-image-1 generate, called once for every sequence, CPU
image perception	16 sec	API: gpt-4o vqa, called 2.5 times (Averaged across 35 sequences), CPU
image editing	1 min 26 sec	API: gpt-image-1 edit, called 2.5 times (Averaged across 35 sequences), CPU
Sparse Inference-Time Tuning	5 min 36 sec	Wan2.1-T2V-1.3B, 480×832, 81 frames, NVIDIA A100 GPU
pre-process visual thoughts for fine-tuning	30 sec	
load model	6 sec	
fine-tuning	5 min	including checkpoint saving
Sparse Inference-Time Tuning	6 min 56 sec	Wan2.1-T2V-14B, 480×832, 81 frames, NVIDIA A100 GPU
pre-process visual thoughts for fine-tuning	30 sec	
load model	20 sec	
fine-tuning	6 min 6 sec	including checkpoint saving
Video Sampling	3 min 9 sec	Wan2.1-T2V-1.3B, 480×832, 81 frames, NVIDIA A100 GPU
model loading	14 sec	could save time by not saving then re-loading checkpoint upon tuning
sampling	2 min 46 sec	
VAE decoding & video saving	9 sec	
Video Sampling	14 min 48 sec	Wan2.1-T2V-14B, 480×832, 81 frames, NVIDIA A100 GPU
model loading	33 sec	could save time by not saving then re-loading checkpoint upon tuning
sampling	14 min 06 sec	
VAE decoding & video saving	9 sec	

(a) Input prompt: *"An ice cream cone is left out in the sun."*

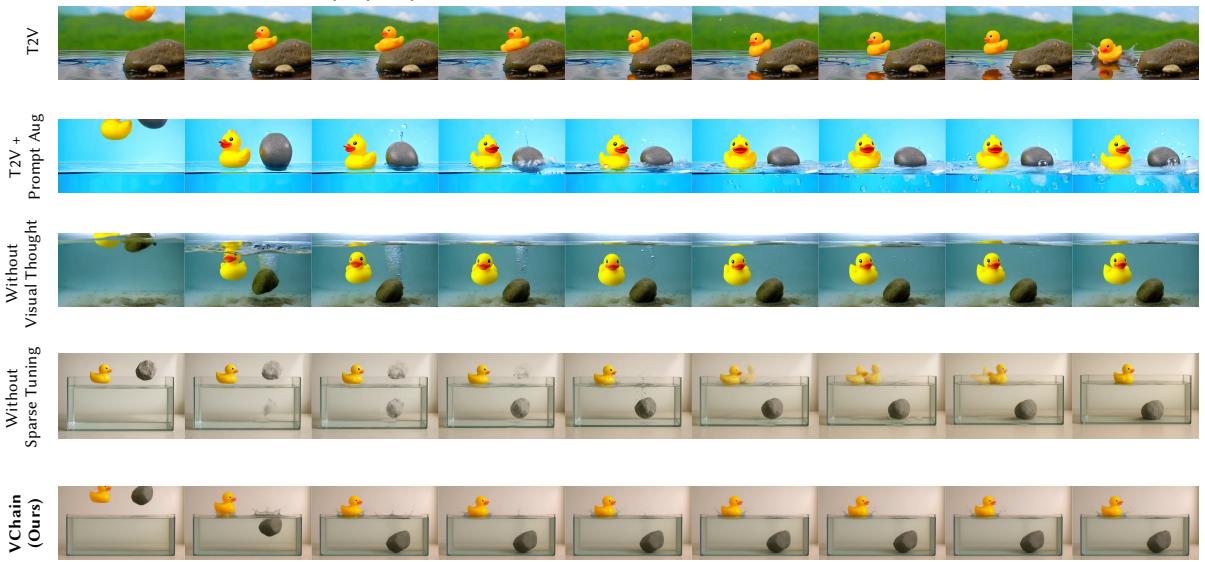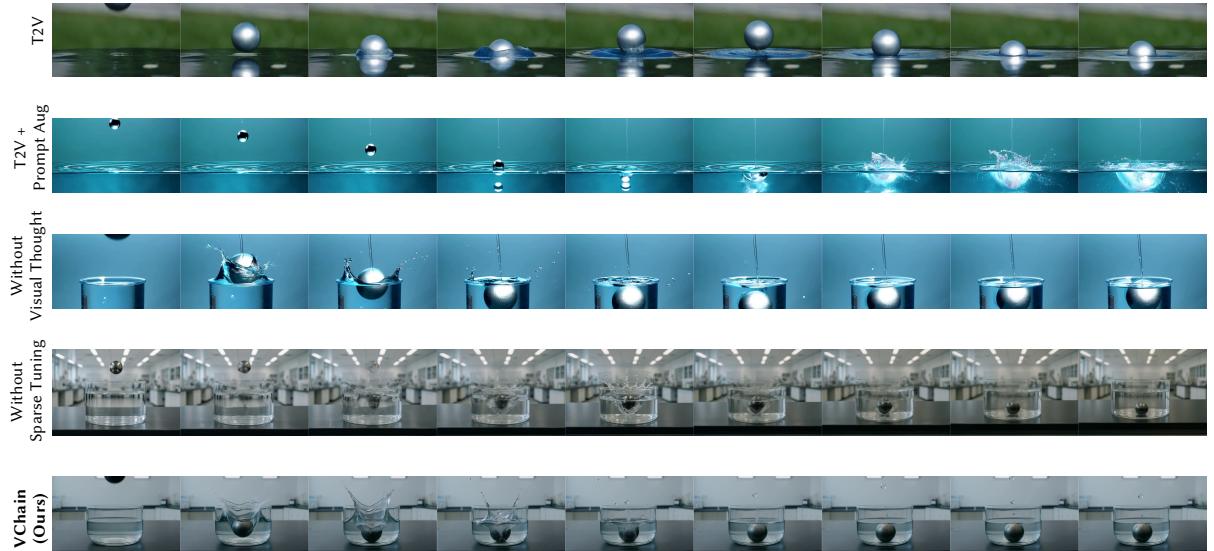(b) Input prompt: *"A rubber duck and a rock fall into a water tank."*

Figure 11: **More Qualitative Comparisons.**

(a) Input prompt: *"A steel ball is dropped into water."*

(b) Input prompt: *"Milk is poured into a cup of black coffee."*

Figure 12: **More Qualitative Comparisons.**

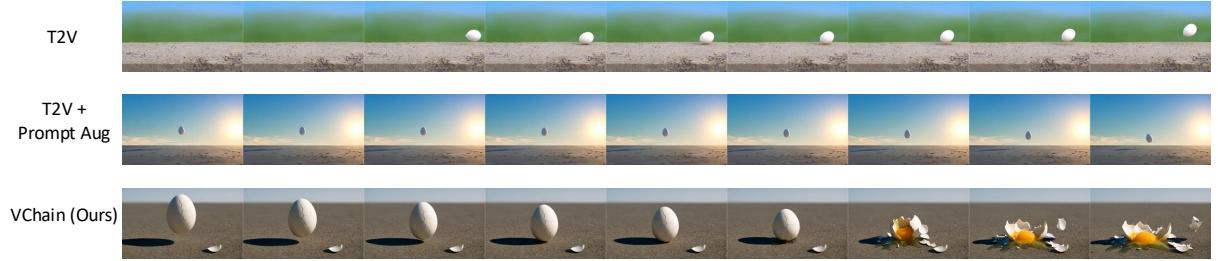Input prompt: *"An egg falling from the sky towards concrete ground"*

Figure 13: **Additional Qualitative Comparisons - Egg Fall.**

Input prompt: *"A ball is dropped onto a pillow."*

Figure 14: **Additional Qualitative Comparisons - Pillow.**

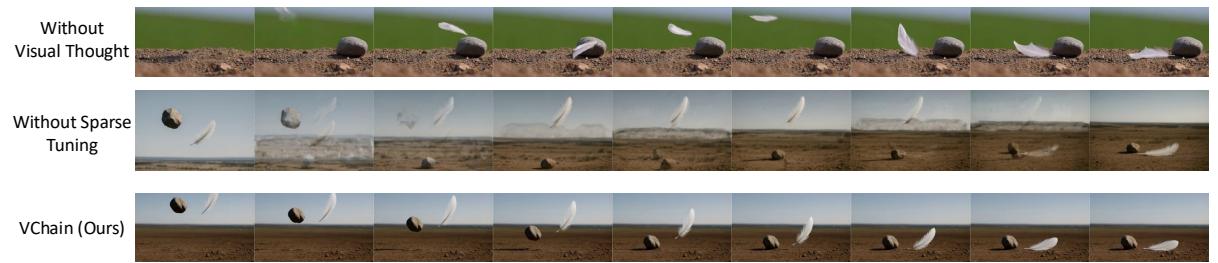Input prompt: *"A rock and a feather falling from the sky towards the ground"*

Figure 15: **Additional Qualitative Comparisons - Rocket Feather.**

Input Prompt: *"A cup of water is falling towards the ground on its side."*

Figure 16: **Additional Qualitative Comparisons - Cup.**
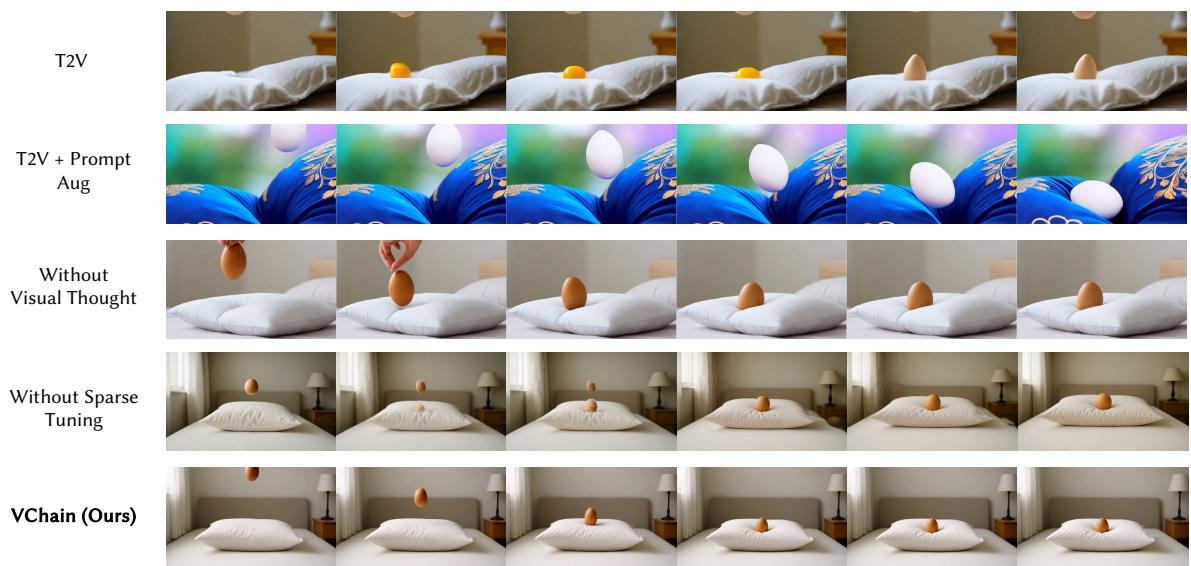
Input Prompt: *"An egg is dropped onto a pillow."*

Figure 17: **Additional Qualitative Comparisons - Egg Pillow.**

Input Prompt: *"Oil is poured into a glass of milk."*

Figure 18: **Additional Qualitative Comparisons - Oil Milk.**

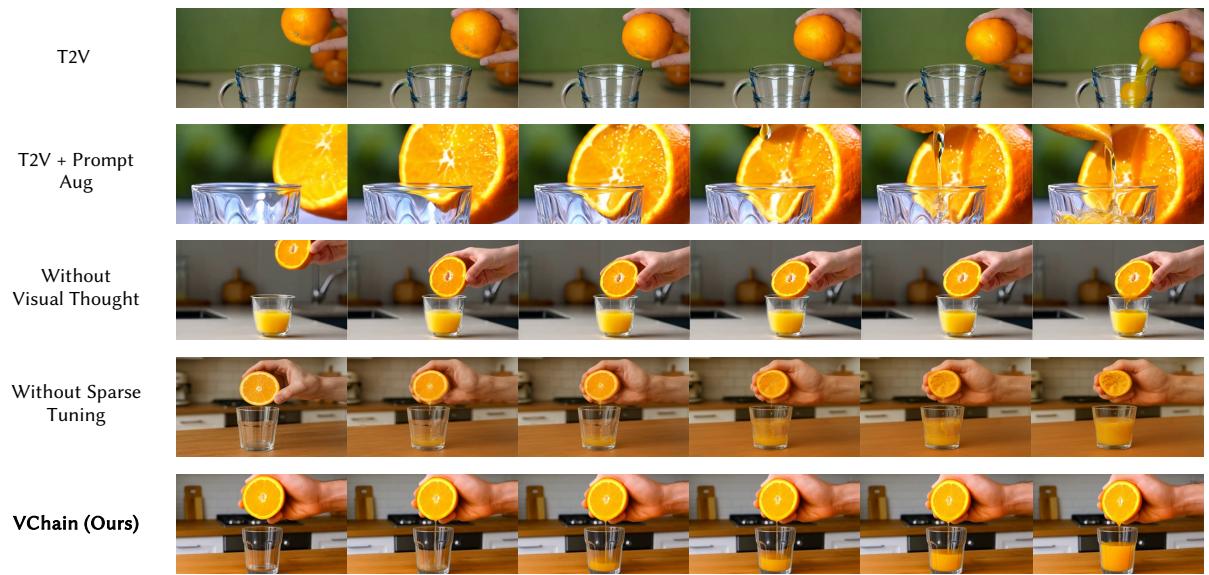Input Prompt: *"A sliced orange is squeezed right above an empty glass cup."*

Figure 19: **Additional Qualitative Comparisons - Orange.**