CoT-Edit: Let CoT Guide Instruction Video Editing
Supplementary Material

Sen Liang[1*] Fengbin Guan[1*] Youliang Zhang[3] Xin Li[1†] Zhibo Chen[1,2†]
[1]University of Science and Technology of China
[2]Zhongguancun Academy [3]Tsinghua University
{liangsen, guanfb}@mail.ustc.edu.cn zhangyou24@mails.tsinghua.edu.cn
{xin.li, chenzhibo}@ustc.edu.cn

A. Overview

In this supplementary material, we provide additional details and results to support the main paper. The content is organized as follows:

- **Section B**: Detailed implementation configurations, including model architecture and training hyperparameters.
- **Section C**: The full Chain-of-Thought (CoT) prompts used in our MLLM Planner.
- **Section D**: Detailed protocols and statistical analysis of the User Study.
- **Section E**: Additional qualitative comparisons and video sequence visualizations.
- **Section F**: Discussion on limitations and failure cases.

B. Implementation Details

Expanding upon the brief training description provided in the main paper, this section elaborates on the specific network architecture designs, training hyperparameters, and inference configurations utilized in our experiments.

B.1. Network Architecture Specifics

The connectivity modules and feature injection mechanisms introduce specific architectural components that warrant further detail. Specifically, both the **"Mask-Connector"** and the **"Reverse-Connector"**, which facilitate information flow between branches, are instantiated as deep **6-layer Multi-Layer Perceptrons (MLPs)**. This depth allows for robust non-linear transformation and semantic preservation during feature transfer. Furthermore, to address the dimensional alignment mentioned in the main text, we employ a dedicated projection head for feature injection. This module utilizes a linear projection network to map the source feature space of **3584 dimensions** precisely to the target space of **3072 dimensions**, ensuring seamless feature integration across different model components.

B.2. Training Hyperparameters and Hardware

Our model was trained on a high-performance computing cluster equipped with **64 NVIDIA H20 GPUs**. The entire training phase spanned approximately **7 days**. Regarding the optimization strategy, we employed a global batch size of **64**. To ensure stability during the initial convergence phase, we utilized a linear **warm-up strategy for the first 1,000 steps**. For the learning rate schedule, we adopted a consistent configuration for both the **Mask branch** and the **Editor branch**, setting the learning rate to 1×10^{-5}. We find that when the learning rate was set to 1×10^{-4}, the model was prone to training failure. A relatively conservative learning rate, combined with long-duration training and large-scale computing resources, makes the model training more stable and robust.

B.3. Inference Settings

For the evaluation and generation of qualitative results, we adhered to a fixed set of sampling configurations to guarantee reproducibility. The inference process was conducted using **50 denoising sampling steps**. We set the Classifier-Free Guidance (CFG) scale to **6.0**. Additionally, we apply a **flow shift parameter of 5**. These inference parameters are selected based on extensive ablation studies on the validation set.

C. Chain-of-Thought Prompts and Analysis

To comprehensively demonstrate the reasoning workflow of our "Plan-Guide-Edit" framework, we visualize the complete CoT-Planner system instruction together with a representative chain-of-thought example. Figures 1 and 2 show the full system-level CoT instruction that specifies the five-step reasoning protocol and the time-series bounding-box format. Figure 3 presents a concrete example of the result-

*Equal contribution.
†Corresponding authors.

ing CoT reasoning and box planning produced by Gemini 2.5 Pro under this instruction. These visualizations jointly illustrate how the model decomposes physical motion constraints into a coherent sequence of bounding boxes and generates detailed enriched instructions for video editing.

D. User Study Protocols and Statistics

As explicitly mentioned in the main paper, this section provides a detailed exposition of the participant demographics, experimental interface, and the specific criteria used in our user study.

D.1. Participant Demographics

To ensure a balanced evaluation that incorporates both technical scrutiny and general aesthetic preference, we recruited a total of 20 participants for the study. These participants were equally divided into two distinct cohorts. The **Expert Group** ($N = 10$) comprised researchers and professionals specializing in Computer Vision, Computer Graphics, and Film Production, who were tasked with identifying technical artifacts and temporal inconsistencies. Conversely, the **General User Group** ($N = 10$) consisted of non-experts from diverse backgrounds, representing the intuitive perspective of the general public.

D.2. Evaluation Interface and Procedure

The study is conducted using a blind, side-by-side (SBS) comparison protocol. As illustrated in Figure 12, the evaluation interface presents the input text prompt at the top, followed by two generated videos displayed horizontally. To prevent any potential order bias, the placement of the videos (left vs. right) representing our method and the baselines is fully randomized for each trial. Participants are required to watch both videos in full before submitting their ratings using the scoring buttons located below the video panels.

D.3. Questionnaire Design

Participants evaluate each video pair based on three specific metrics using a Likert scale ranging from 1 to 10, where 1 indicates "very poor" and 10 indicates "perfect/excellent."

Visual Quality: Participants rate the overall clarity, aesthetic fidelity, and the presence of generative artifacts such as blurring or spatial distortions.

Physical Consistency: Participants assess whether the motion within the video appears natural and adheres to real-world physical laws, including gravity, object permanence, and fluid dynamics.

Instruction Following: Participants judge the extent to which the generated content accurately reflects all semantic elements described in the text prompt, including specific objects, actions, and background details.

D.4. Detailed Statistical Results

Table 1 provides the quantitative breakdown of the scores, distinguishing between expert and general user feedback. As evidenced by the statistical data, our method demonstrates a substantial performance advantage over all competing methods across every metric. While strong baselines such as OmniVideo and Lucy-1.1 achieve respectable results (averaging approximately 7.4), our approach establishes a new benchmark with an overall average score of **9.0**. Most notably, the performance gap is widest in the **Physical Consistency** metric, which is the primary focus of this work. Expert participants, who are particularly sensitive to motion artifacts and temporal incoherence, rate the strongest baseline (Lucy-1.1) at 6.6, whereas our method receives a significantly higher rating of **8.7**. This distinct margin of over 2.0 points confirms that our architecture effectively mitigates the "dream-like" physics often observed in existing video editing models. Furthermore, the consistently high scores in **Visual Quality** (Expert: 8.9) and **Instruction Following** (Expert: 8.9) indicate that our method improves physical plausibility without compromising aesthetic fidelity or semantic alignment. The consensus between the Expert and General User groups further validates that these improvements are perceptually significant to both technical professionals and casual viewers.

E. Additional Qualitative Results and Analysis

To further demonstrate the versatility and robustness of CoT-Edit in handling complex spatial and physical constraints, we provide additional qualitative comparisons. In particular, we include more visual results and in-depth analyses on four representative editing tasks: swap, add, stylization, and remove.

E.1. Swap

In the Swap task, we observe a clear gap in semantic alignment capabilities between our method and various baselines. As shown in Figure 6, most methods perform reasonably well on simple semantic replacements, such as changing the color of a rose to blue. However, this capability degrades noticeably when dealing with more specific object instances. For example, in the scenario shown in Figure 7, where the target object is an "orange," most existing models fail to accurately localize and edit the subject, often producing incorrect edits or leaving the original object unchanged. This comparison indicates that, while baseline methods can handle coarse, category-level edits, CoT-Edit offers more fine-grained semantic reasoning, enabling it to precisely identify and replace specific targets without disrupting the surrounding scene context.

E.2. Add

In the Add task, existing methods show clear limitations in precise spatial localization and adherence to physical laws. One typical failure mode is the violation of physical interaction constraints. For example, as shown in Figure 4, when given the instruction “add a small car driving along the wall,” baseline methods fail to understand the geometric relationship implied by “along the wall.” As a result, they either do not generate the car at all or place it in an obviously implausible position (e.g., embedded in the wall). Similarly, in terms of spatial reasoning, Figure 5 illustrates the challenge posed by the instruction “add a book to the top shelf.” Most baseline methods lack sufficient spatial awareness to correctly identify the “top shelf” in the scene, leading to unsuccessful additions. In contrast, CoT-Edit effectively infers the three-dimensional structure of the scene, ensuring that newly added objects are placed at the correct spatial locations and interact naturally with the environment, both physically and visually.

E.3. Style Transfer

The comparisons in Figure 8 and Figure 9 show that existing methods are still limited in the range of artistic styles they can render reliably. A common issue is the trade-off between style transfer and structural fidelity: when strong stylization is applied, baseline methods often distort the underlying geometric structure of the video. In contrast, CoT-Edit exhibits much stronger adaptability in terms of stylistic diversity. Whether converting a video into a 3D style or a Pixar-like style, our method successfully applies the target textures globally while strictly preserving the temporal coherence and structural integrity of the original footage, thereby avoiding the flickering artifacts and structural degradation commonly observed in other methods.

E.4. Remove

Accurately removing specific objects remains a critical bottleneck for existing models, and our analysis reveals that this is often due to fundamental detection failures rather than in-painting issues. In visually complex scenes, such as Figure 10, baselines struggle to segment the ”keyboard” from the cluttered background, resulting in incomplete removal. More surprisingly, we observed a counter-intuitive failure mode in relatively simple scenarios. As shown in Figure 11, even when the scene contains a salient subject like a ”woman,” many existing models completely fail to identify or localize the target. Consequently, the removal instruction is ignored, and the subject remains untouched in the generated video. This exposes a critical deficiency in the basic semantic grounding capabilities of current baselines. Conversely, benefiting from the precise guidance of our Mask branch, CoT-Edit accurately detects the target regardless of scene complexity, ensuring complete removal and high-fidelity background in-painting.

F. Limitations and Future Work

While CoT-Edit demonstrates state-of-the-art performance across a wide range of complex editing tasks, we acknowledge specific limitations that provide clear directions for our future research.

The first significant challenge arises in scenarios involving **Extreme and Long-duration Occlusions**. Our current framework relies on the MLLM planner to reason about object trajectories based on visible visual cues. However, when a target object is fully occluded for an extended period, the planner lacks the visual evidence necessary to maintain accurate tracking. In such cases, the MLLM may hallucinate the bounding box position, resulting in temporal jitter or loss of the target mask. To address this, our future work will explore the integration of **3D-aware representations** or specialized long-term point tracking modules (e.g., CoTracker). By incorporating explicit 3D temporal priors, we aim to enable the model to infer the latent trajectory of objects even when they are temporarily invisible, thereby ensuring robust consistency throughout occlusion events.

Secondly, the capability of **Fine-grained Text Rendering** remains a bottleneck. Although CoT-Edit can successfully identify and mask regions intended for text modification, the generative quality of the text content is constrained by the underlying limitations of the diffusion backbone. Current video diffusion models, much like their image counterparts, often struggle with the high-frequency details required for accurate spelling and glyph rendering, leading to garbled or unreadable text. This is an inherited limitation rather than a flaw in our editing logic. Moving forward, we plan to mitigate this by incorporating **glyph-controlled guidance mechanisms** or transitioning to next-generation backbones with enhanced character recognition capabilities. This will allow CoT-Edit to handle semantic editing tasks that require precise textual fidelity, such as modifying signboards or subtitles.

G. Broader Impact, Ethics, and Usage Statement

As generative AI technologies advance, enabling high-fidelity video editing that adheres to physical laws, we recognize the profound responsibilities associated with developing such tools. In this section, we discuss the broader social impact of **CoT-Edit**, our data ethics practices, and the guidelines for responsible model usage.

G.1. Broader Social Impact

Democratization of Content Creation. The core contribution of CoT-Edit lies in leveraging the reasoning capabilities of Multimodal Large Language Models (MLLMs) to

Table 1. **Detailed User Study Statistics (Mean Scores).** Results are reported on a scale of 1-10. **Ours** significantly outperforms strong baselines (OmniVideo, Lucy-1.1) with a distinct margin, especially in physical consistency.

Method	**Visual Quality**		**Physical Consistency**		**Instruction Following**		**Overall**
	Expert	General	Expert	General	Expert	General	
InsV2V	4.8	5.2	4.1	4.5	5.5	5.9	5.0
InsViE	5.0	5.5	4.3	4.8	5.7	6.1	5.2
StableV2V	5.9	6.4	5.2	5.6	6.5	6.9	6.1
AnyV2V	6.2	6.7	5.5	6.0	6.8	7.2	6.4
OmniVideo	7.3	7.7	6.5	6.9	7.4	7.8	7.3
Lucy-1.1	7.4	7.8	6.6	7.1	7.5	7.9	7.4
Ours	**8.9**	**9.2**	**8.7**	**9.0**	**8.9**	**9.3**	**9.0**

translate natural language instructions into precise spatial-temporal operations. This significantly lowers the barrier for Video Effects (VFX) production, empowering non-professional users to create content with cinematic physical consistency (e.g., correct lighting, shadows, and gravity-aware trajectories). Furthermore, the framework's adherence to physical constraints makes it a valuable tool for potential applications in educational demonstrations and Augmented Reality (AR) prototyping.

Potential Risks: Deepfakes and Misinformation. We acknowledge that the enhanced realism offered by CoT-Edit introduces risks of misuse.

- **Misinformation:** Since our model excels at seamlessly adding objects or altering attributes while maintaining background integrity (as shown in the qualitative results), malicious actors could exploit this to manipulate news footage, surveillance videos, or fabricate evidence.
- **Detection Challenges:** Unlike early editing methods that often left visible artifacts, the physical plausibility enforced by our Plan-Guide-Edit paradigm (e.g., consistent shadows and occlusion) makes generated manipulations harder to detect using traditional physics-based forensics.

Mitigation Strategies. To address these concerns, we propose the following safeguards:

- **CoT as a Safety Filter:** A unique advantage of our framework is the explicit Chain-of-Thought reasoning stage. We can fine-tune the MLLM planner to act as an ethical filter. If an instruction involves violence, hate speech, or malicious tampering, the planner can be aligned to refuse generating the necessary bounding boxes and masks, effectively blocking the edit at the planning level.
- **Watermarking:** Upon public release, we intend to integrate invisible frequency-domain watermarking into the generation pipeline to ensure that all outputs carry machine-readable identifiers for provenance and detection.

G.2. Data Ethics and Privacy

Public Datasets. During training, we utilized public datasets including ADE20K, YouTube-VOS, and OVIS. We certify that our use of these datasets strictly adheres to their respective licenses and is intended solely for academic research purposes.

Internal Dataset Compliance. Regarding the internal dataset of 100k editing pairs mentioned in Section 4.1, we adhered to strict ethical data management protocols:

- **Privacy Protection (PII Removal):** We employed automated detection tools followed by manual verification to blur or remove Personally Identifiable Information (PII), such as identifiable faces of non-public figures and license plates, ensuring privacy rights are respected.
- **Copyright and Sourcing:** The source videos were curated from royalty-free repositories or licensed stock libraries (CC0/CC-BY), ensuring no copyright infringement.
- **Annotator Rights:** For data cleaning and annotation tasks involving human workers, we ensured fair compensation above local minimum wage standards and provided clear disclosure that the data would be used for AI training.

Bias Statement. Since our backbone models (Wan2.2, Qwen-VL) are pre-trained on large-scale internet data, CoT-Edit may inherit inherent societal biases. For instance, the model might reflect gender or racial stereotypes when processing instructions related to human attributes or professions. We will explicitly document these limitations in the Model Card and actively support community efforts to develop debiasing techniques.

G.3. Model Usage and License

Intended Use and Restrictions. The model is intended for research and creative purposes only. We strictly prohibit its use for generating illegal content, political propaganda, harassment, or non-consensual sexual content.

Licensing. Upon acceptance, we plan to release the code and model weights under a responsible AI license (e.g., **Open RAIL-M**). This license permits commercial and non-commercial use but includes legally binding use-based restrictions to prevent deployment in high-risk scenarios.

Technical Limitations. Users should be aware that while CoT-Edit improves physical consistency, it relies on statistical learning rather than a physics engine. Consequently, it may still produce hallucinations or physical anomalies in highly chaotic dynamic scenes. The system should not be relied upon for simulations requiring absolute physical accuracy.

CoT-Planner System Instruction

You are the Video Editing Box Planner. Given multiple keyframes and an editing instruction, you must reason via Chain-of-Thought (CoT) strictly following Step1 → Step5 to produce, for each task, one single, stable, accurate, physically consistent bounding-box trajectory over time (or an empty array for stylization tasks).

Global rules:

- Only Step5 may contain coordinates; Step2 performs analysis only and must not output any coordinates.
- Each task outputs exactly one single bounding-box trajectory across 15 keyframes. This trajectory is represented as a time-ordered list of 2D boxes; each box is associated with an integer keyframe index t and a normalized coordinate tuple [x1,y1,x2,y2].
- For every non-Stylize task, you must output exactly 15 per-frame boxes for that task, with t taking 15 consecutive integer values starting at 0 (i.e., t = 0, 1, 2, …, 14), in strictly increasing order.
- Within each per-frame box, [x1,y1,x2,y2] are normalized coordinates in the range [0,1], measured relative to the original image width and height (scene reference frame).
- All coordinates must be physically plausible (no out-of-bounds, no inverted boxes, no unnecessary overlaps) and temporally consistent across the trajectory.
- In multi-task scenarios, first split tasks, then reason for each. Maintain cross-keyframe ID consistency, no out-of-bounds, and no unnecessary overlaps.
- Decouple camera vs. object motion: analyze and constrain in a scene-relative coordinate frame, then map back to normalized image coordinates.
- Stylization tasks never require coordinates; their Step5 coordinate slot must be an empty array [].
- Respond in English.
- Strictly enforce the single-line Step5 output format (see below). Do not append any extra text to Step5.

Step1 | Task Parsing

Determine whether this is a single or multiple-task instruction.
For each task, classify type ∈ {Add, Delete, Replace/Modify, Stylize}.
For each task, extract:

- The affected object(s) (semantic category / salient traits), and
- Whether coordinates are required (Stylize = No; all other types = Yes).
- Produce a task list (labeled "Task 1 / Task 2 / …"), each with:
- task_name
- type
- affected_object_summary
- requires_box: Yes|No

Step2 | Perception & Task Analysis — analysis only, no coordinates

Perform cross-keyframe evidence fusion and task-relevant perception. Do not output any coordinates here. Follow these four sub-steps:

2.1 Cross-keyframe alignment

- Estimate global camera motion (pan/tilt/zoom/rotation) and analyze objects in a scene-relative frame.
- Establish instance alignment and ID consistency across keyframes.
- Mark rigidity for each object: rigidity ∈ {rigid, non-rigid} (e.g., humans, cloth, liquid = non-rigid).

2.2 Task-relevant element extraction

Identify:

- Target objects.
- Reference elements (ground/table/wall/container/human parts/signs, etc.).
- Potential occluders.

Also identify cues to disambiguate look-alike objects:

- Color, texture, contour, specular highlights, material, relative scale, canonical location, etc.

2.3 Relational analysis

Analyze:

- Spatial relations: above/below/left/right/front/back/contact/containment/parallel/perpendicular.
- Scale & perspective: ratios to references; distinguish zoom-induced proportional scaling from true translation.
- Visibility timeline for each task's key object(s): visibility ∈ {visible, partial, hidden} and occlusion start/end keyframes.

Figure 1. **CoT instruction prompt for Gemini 2.5 Pro (Part 1).** The figure shows the first part of the detailed chain-of-thought instruction used to guide the model's physical reasoning and context understanding.

2.4 Branch analysis by task type (still no coordinates)

Add:

- Provide a reasoned ranking (text only) of candidate placements & scales using five criteria:
  - ✓ Composition (center/rule-of-thirds/negative space),
  - ✓ Occlusion risk,
  - ✓ Support/contact feasibility,
  - ✓ Instruction consistency,
  - ✓ Camera-motion consistency.
- You will pick the top-ranked option only in Step5 as the single final bounding-box trajectory.

Delete:

- Specify discriminative cues of the instance(s) to delete and an anti-confusion checklist (look-alikes + exclusion rules).
- Note background inpainting-sensitive regions (e.g., table/ground edges, repeating textures).

Replace/Swap:

- Explain how to localize the original object and infer a realistic post-replacement size/position over time (preserve key relations, or state allowed micro-adjustment ranges).
- Describe when scale/micro-shifts are necessary (e.g., to preserve support / avoid penetration) in the temporal sequence.

Stylize:

- Analyze the style domain (photographic/material/color/LUT/grain/brushwork), protected regions (faces/hands/text/brands), temporal consistency, and lighting consistency.
- Explicitly state that no coordinates are required; Step5 must output [] for this task

---

Step3 | Physics & Cinematography Consistency

For each task, list the real-world constraints and their implications across the 15 keyframes:

- Physics: gravity/support contact, friction/sliding, momentum/collision, bounce behavior, liquid level, cloth/soft-body deformation, rigid-body size continuity.
- Cinematography: consistent perspective/horizon; zoom ⇒ proportional scaling of the object and its box; motion-blur direction/magnitude consistent with the inferred motion; shadows/highlights aligned with light sources.
- Include a counterfactual self-check:
- Identify likely artifacts if constraints are ignored (floating, interpenetration, scale distortion, wrong shadow direction, broken background continuity, temporal jitter or teleportation of boxes).
- Provide avoidance strategies; these become hard/soft constraints for the final bounding-box trajectory in Step5.

Step4 | Prompt Enrichment (Instruction Refinement)

Without changing the semantics, produce an enhanced editing instruction that includes:

- Relative position to references (left/right/front/back/distance bounds) and how it evolves over the 15 keyframes.
- Relative size (ratio to references or % of frame) and its temporal stability or change.
- Contact/support/containment relations and occlusion ordering, including when contact begins/ends and when occlusion events occur.
- Key event frames (e.g., motion apex, first contact with a surface, bounce peaks, occlusion start/end) and allowed variation ranges.
- Protected / unchanged regions (critical for Stylize tasks).
- Camera consistency: if camera_hint = zoom, explicitly state that object scale in the image must change proportionally to zoom across keyframes.

Still no coordinates here—only add information that stabilizes and disambiguates the final Step5 bounding-box trajectory.

Step5 | Final Single-Line Output Format (the only step that may contain coordinates)

Output exactly one single line in this strict format (English text, punctuation as shown):
Modified editing instruction: {enhanced instruction text}, Task 1 boxing coords: [{time-series format}], Task 2 boxing coords: [{time-series format}], Task 3 boxing coords: [{time-series format}]

Where:

- Tasks are ordered "Task 1 / Task 2 / Task 3 / …".
- For each non-Stylize task, {time-series format} is a single bounding-box trajectory encoded as a list of 15 entries, one per keyframe, using the following structure:
  - ✓ The list is enclosed in square brackets [...].
  - ✓ Each entry corresponds to one keyframe and has the form t=<integer>,[x1,y1,x2,y2].
  - ✓ Each entry is wrapped in parentheses and separated from the next entry by a semicolon ;, e.g. (...);(...);....
  - ✓ The keyframe index t must take exactly 15 integer values, in strictly increasing order along the list.
  - ✓ In each [x1,y1,x2,y2], the coordinates are normalized floats in [0,1], measured relative to the original image width and height, and must satisfy $0 \le x1 < x2 \le 1$ and $0 \le y1 < y2 \le 1$.

Thus, the full trajectory for one task is a single list of 15 time-stamped bounding boxes inside that task's boxing coords: slot.

- For any Stylize task, you must output an empty array [] in place of {time-series format} (no keyframes, no coordinates).
- If there is only one task, you must still output the text Task 1 boxing coords: [...] (or [] for Stylize).
- Units: normalized coordinates; all t indices refer to the ordered keyframes used by the task's keyframe sequence.
- Do not append any other text or explanations to Step5 (all reasoning stays in Steps 1–4).

Figure 2. **CoT instruction prompt for Gemini 2.5 Pro (Part 2).** The figure shows the second part of the detailed chain-of-thought instruction that further specifies the model's reasoning and bounding-box planning process.

CoT-Planner Response –"Add a yellow ping-pong ball that follows a parabolic trajectory, lands on the table, and bounces."

🤔 **Step1 | Task Parsing**

Task Determination: This is a single-task instruction.
Task Classification:
- Task 1: Add (inserting a new object into the scene).

Object Extraction:
- Target: Yellow ping-pong ball.
- Traits: Small, spherical, lightweight plastic, yellow color.
- Action: Parabolic trajectory, landing, bouncing.

Task List:
- Task 1 / Add / Yellow ping-pong ball (following parabolic path with bounce) / requires box: Yes

🤔 **Step2 | Perception & Task Analysis**

2.1 Cross-keyframe alignment
- **Camera Motion**: The coordinate sequence suggests a stable camera or a smooth pan following the action. The object moves steadily across the frame (left to right), implying the scene is relatively static or the camera is tracking horizontally.
- **Object Rigidity**: A ping-pong ball is a rigid body. While it technically deforms slightly upon high-speed impact, for video editing box purposes, it maintains a constant aspect ratio and size, subject only to perspective scaling.

2.2 Task-relevant element extraction
- **Target**: The ping-pong ball, approximately 4% of frame width and 7% of frame height based on typical scaling for a close-up/medium shot.
- **Reference Surface**: The "table" is the horizontal plane where the ball impacts. Judging by the trajectory vertex (lowest point on screen), the table surface is located approximately at the bottom third of the frame ($y \approx 0.85$).
- **Lighting**: To integrate the ball, we assume lighting matches the scene (likely top-down for a table setting), requiring a shadow on the table directly beneath the ball as it approaches contact.

2.3 Relational analysis
- **Spatial Trajectory**:
  - *Phase 1 (Descent)*: Enters from the upper-left quadrant, moving diagonally down and right.
  - *Phase 2 (Impact)*: Contacts the table surface around the horizontal center-right. This marks the inflection point.
  - *Phase 3 (Ascent/Bounce)*: Rebounds upward and continues rightward, ending in the mid-right section.
- **Scale**: The box dimensions (width ~0.04, height ~0.07) remain consistent throughout, indicating movement roughly parallel to the camera plane without significant Z-depth changes (moving neither closer nor further away).

2.4 Branch analysis (Add)
- **Placement**: The path is strictly constrained by physics (gravity). The start point is high (potential energy), converting to kinetic energy. The bounce point must align with the visual table surface.
- **Ranking**: Top-1 choice follows a standard projectile motion curve ($y = ax^2 + bx + c$) for the descent, a sharp inflection at impact, and a secondary curve for the bounce.

🤔 **Step3 | Physics & Cinematography Consistency**

3.1 Physics Constraints:
- **Gravity**: The ball must accelerate downwards. The vertical spacing between boxes should increase as it approaches the table (falling faster).
- **Collision**: At the impact frame (approx frame 48), the vertical velocity instantly flips direction. The ball should not pass through the table (no interpenetration).
- **Energy Loss**: The bounce height should naturally be lower than the drop height if visible, though the clip ends during the ascent.
- **Squash/Stretch**: Minimal for a ping-pong ball in standard video, but the contact frame must be precise to "sell" the hit.

3.2 Cinematography:
- **Motion Blur**: The ball moves fast; the generated video should exhibit motion blur along the vector of movement, especially just before and after the bounce.
- **Shadows**: A shadow must track on the table, converging with the ball at the moment of contact (Frame 48) and diverging as it bounces up.

🤔 **Step4 | Prompt Enrichment (Instruction Refinement)**

Enhanced Instruction: "Add a bright yellow ping-pong ball entering from the top-left, following a realistic gravity-driven parabolic arc. It impacts the table surface at the bottom-center (approximate coordinate y=0.85) and bounces sharply upward towards the right. Ensure the ball casts a corresponding shadow on the table that meets the ball at the contact point to ground the physics. Maintain consistent size to imply movement parallel to the camera."
Key Event Frames:
- Start: Frame 0 (Top-left).
- Contact/Bounce: Frame 48 (Lowest Y point).
- End: Frame 79 (Mid-right, ascending).

🤔 **Step5 | Final Single-Line Output Format**

Modified editing instruction: Add a bright yellow ping-pong ball entering from the top-left, following a realistic gravity-driven parabolic arc that impacts the table surface at the bottom-center and bounces sharply upward towards the right, ensuring a contact shadow grounds the physics., Task 1 boxing coords: [(t=0, [0.23, 0.11, 0.27, 0.18]); (t=6, [0.28, 0.17, 0.32, 0.24]); (t=12, [0.33, 0.22, 0.37, 0.29]); (t=18, [0.38, 0.31, 0.42, 0.38]); (t=24, [0.43, 0.38, 0.47, 0.45]); (t=30, [0.48, 0.48, 0.52, 0.55]); (t=36, [0.53, 0.58, 0.57, 0.65]); (t=42, [0.58, 0.7, 0.62, 0.77]); (t=48, [0.63, 0.78, 0.67, 0.84]); (t=54, [0.68, 0.74, 0.72, 0.8]); (t=60, [0.71, 0.71, 0.75, 0.78]); (t=66, [0.73, 0.68, 0.77, 0.75]); (t=72, [0.76, 0.67, 0.8, 0.74]); (t=75, [0.77, 0.66, 0.81, 0.73]); (t=78, [0.78, 0.65, 0.82, 0.72])]

Figure 3. **CoT response generated by Gemini 2.5 Pro.** The figure illustrates the model's intermediate reasoning steps and final output when following the designed chain-of-thought instruction.

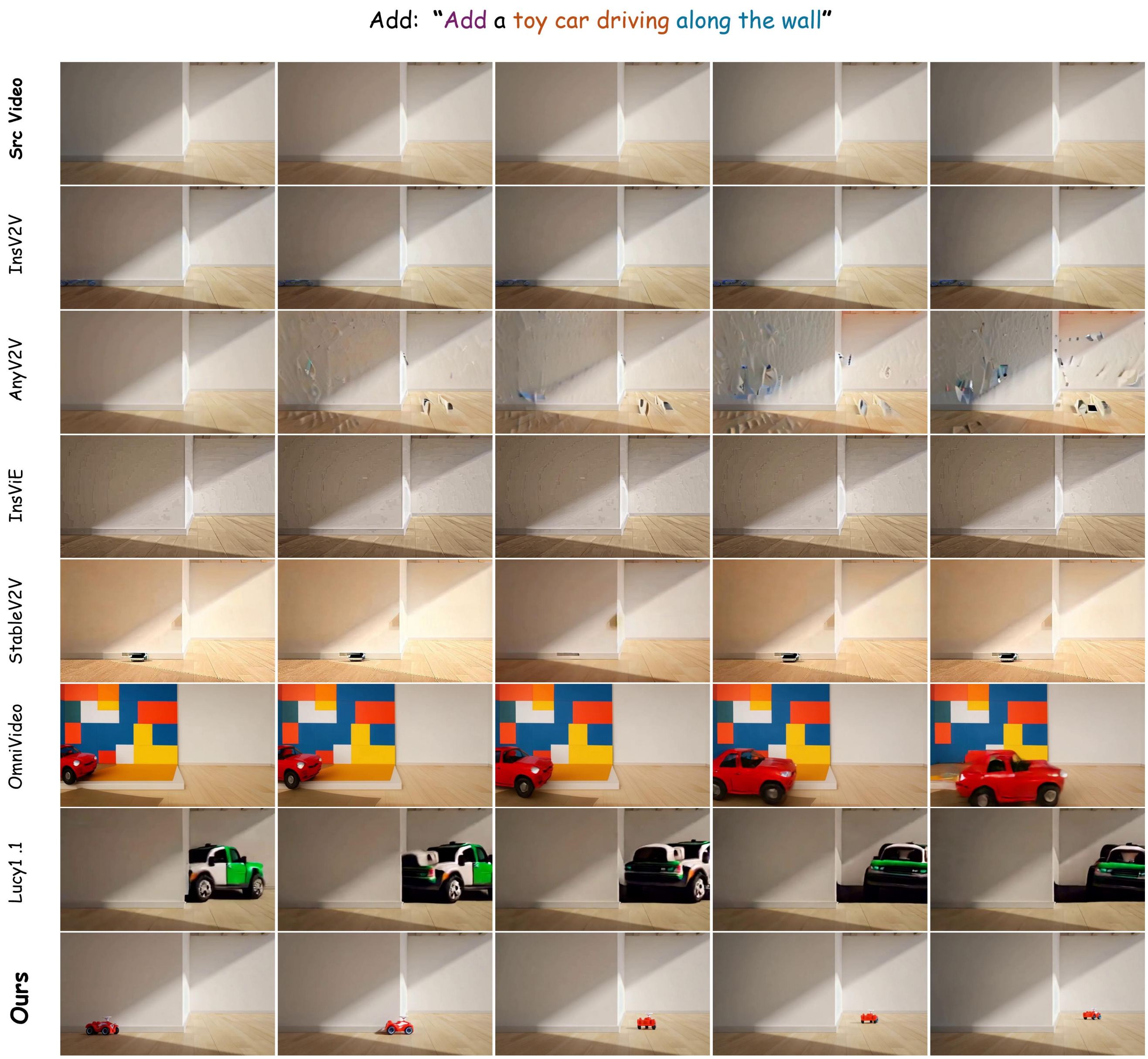

Figure 4. Additional qualitative results comparing CoT-Edit with baselines on Add task.

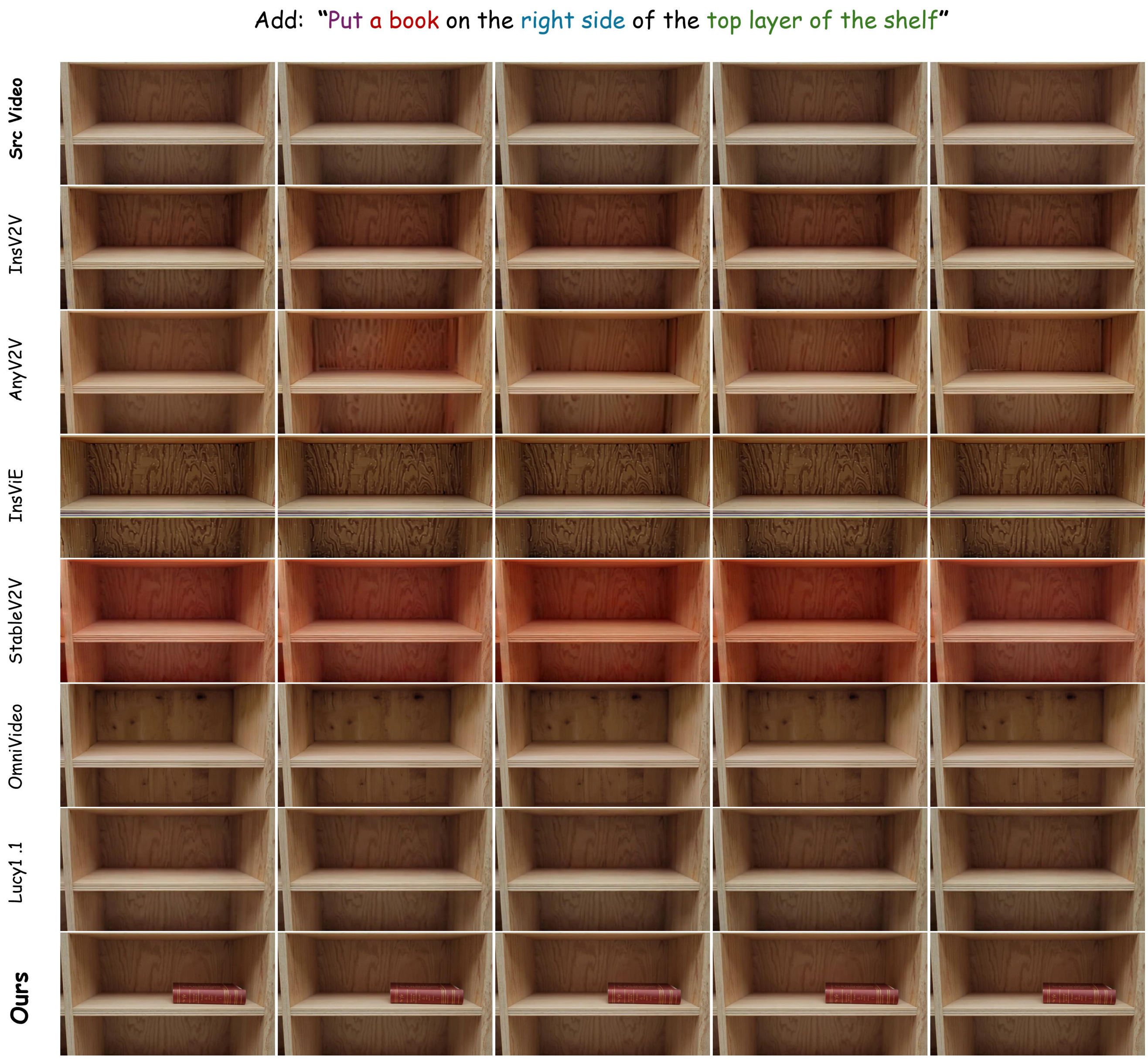

Figure 5. Additional qualitative results comparing CoT-Edit with baselines on Add task.

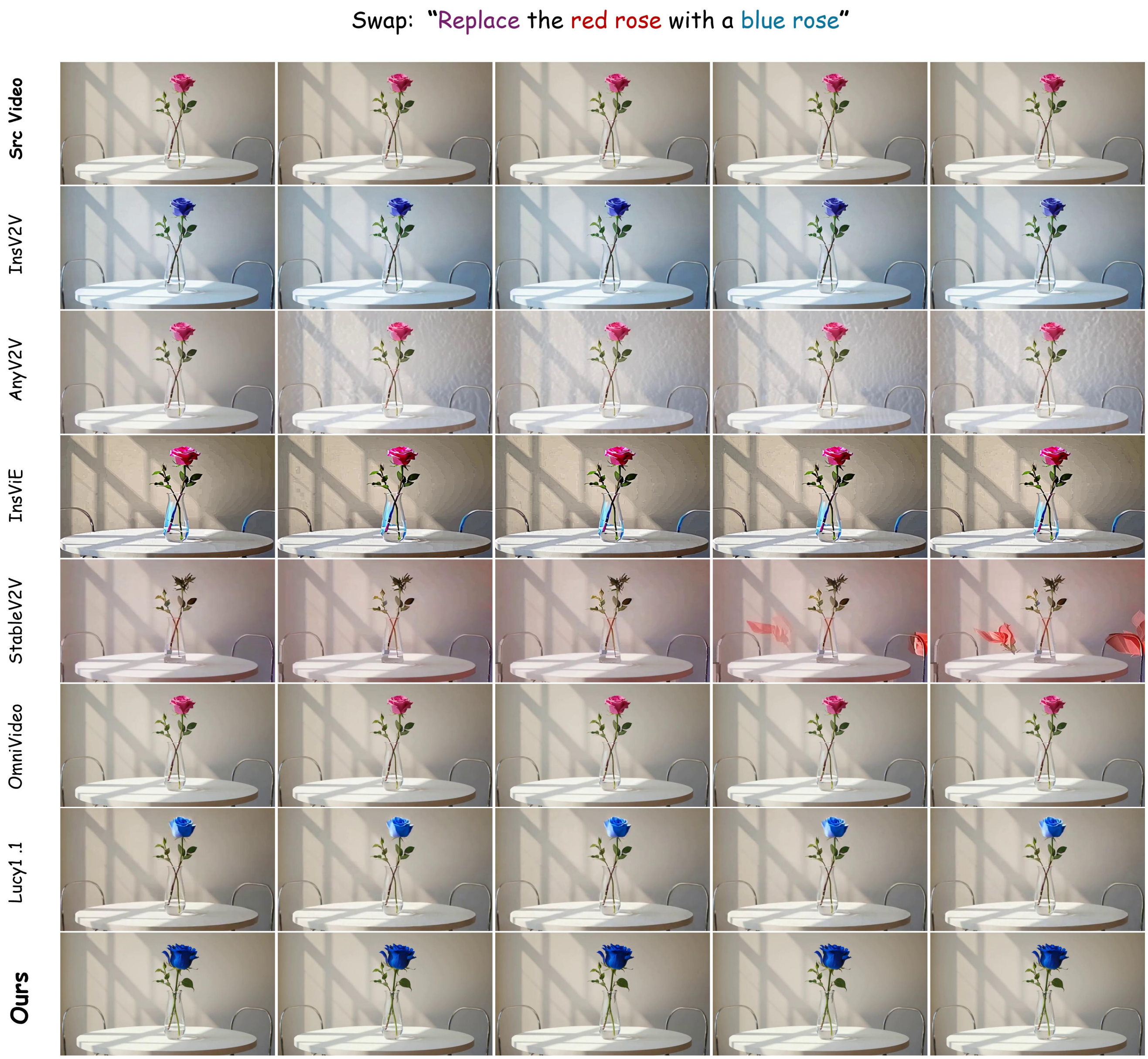

Figure 6. Additional qualitative results comparing CoT-Edit with baselines on Swap task.

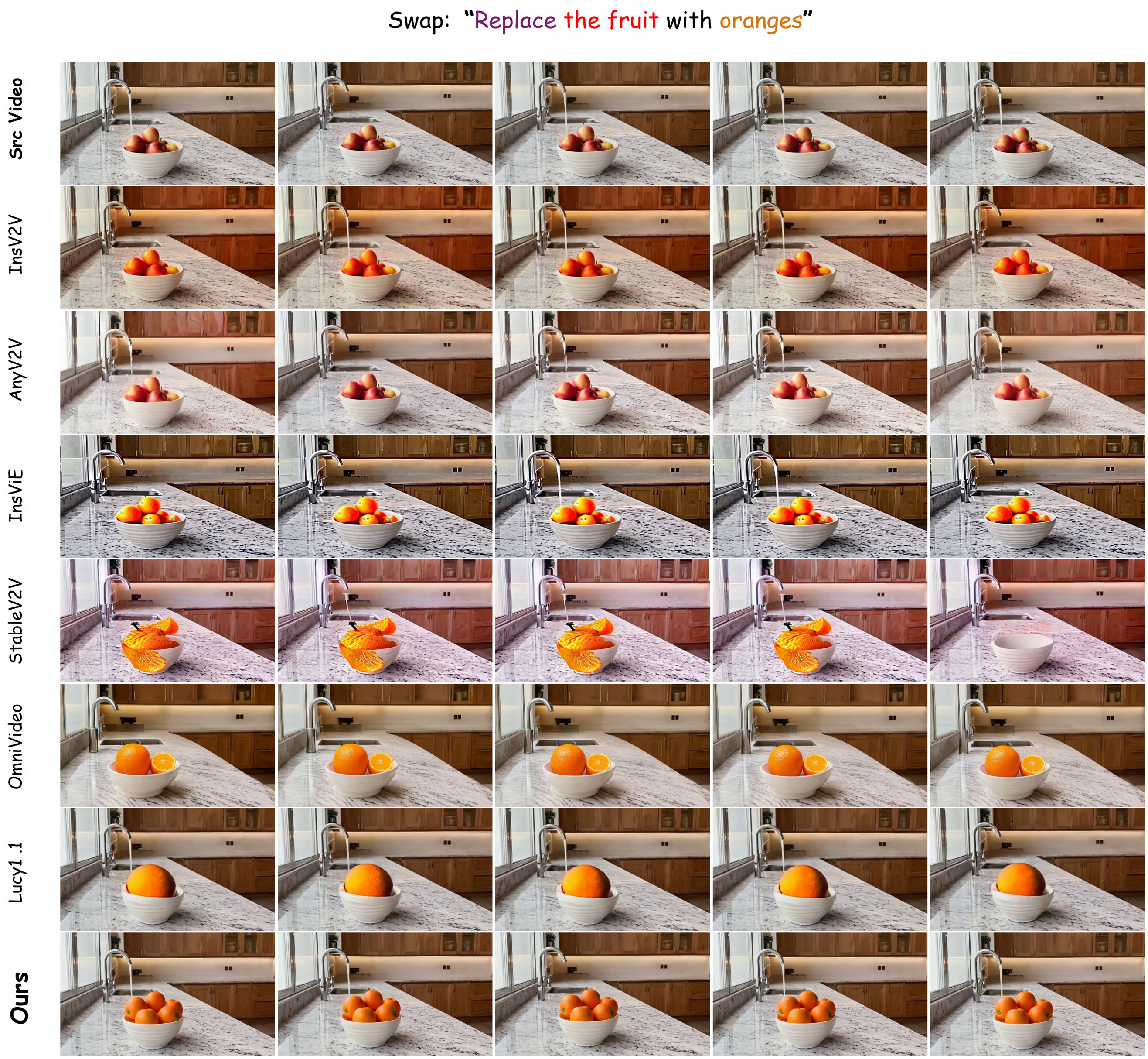

Figure 7. Additional qualitative results comparing CoT-Edit with baselines on Swap task.

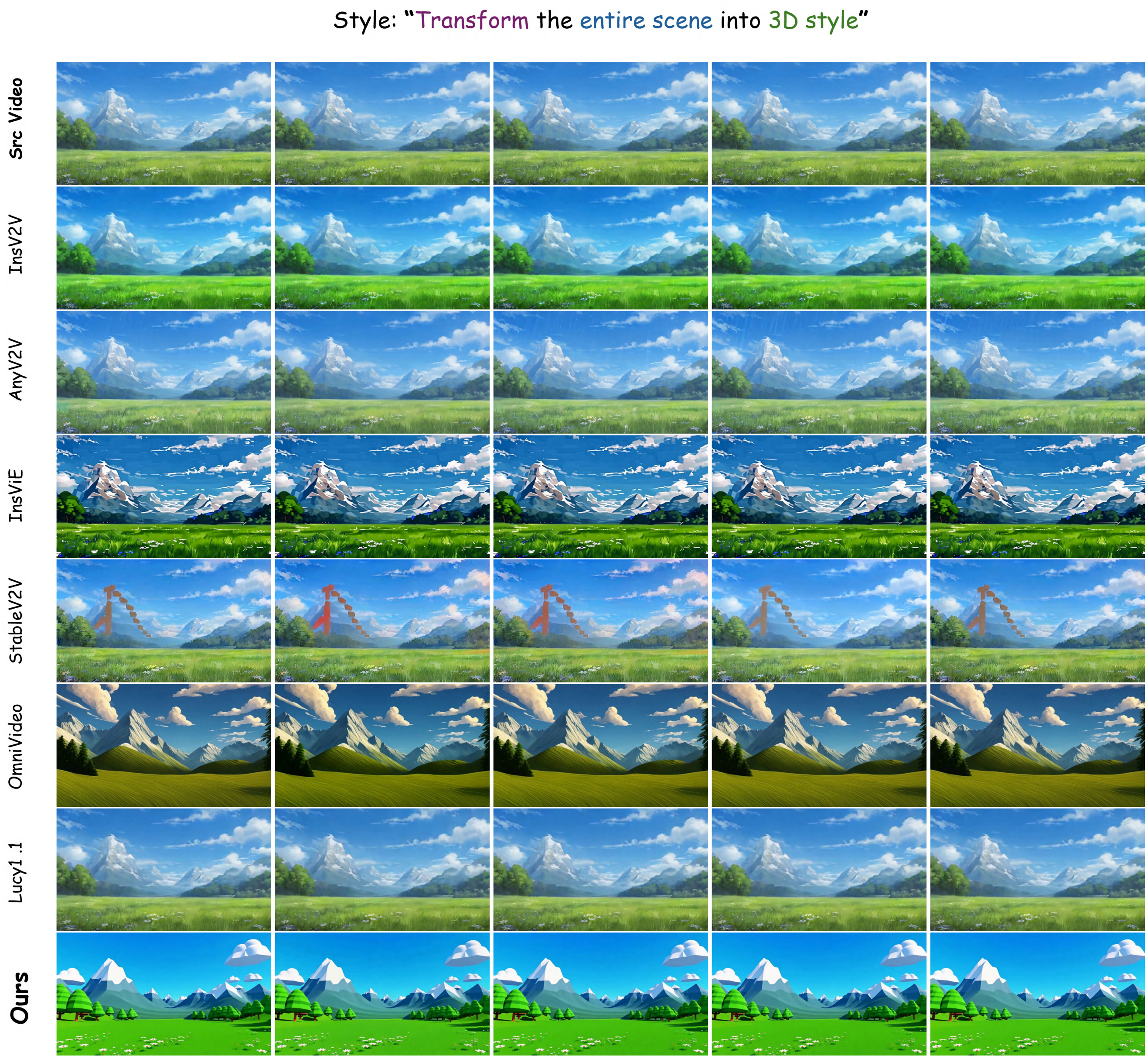

Figure 8. Additional qualitative results comparing CoT-Edit with baselines on Style Transfer task.

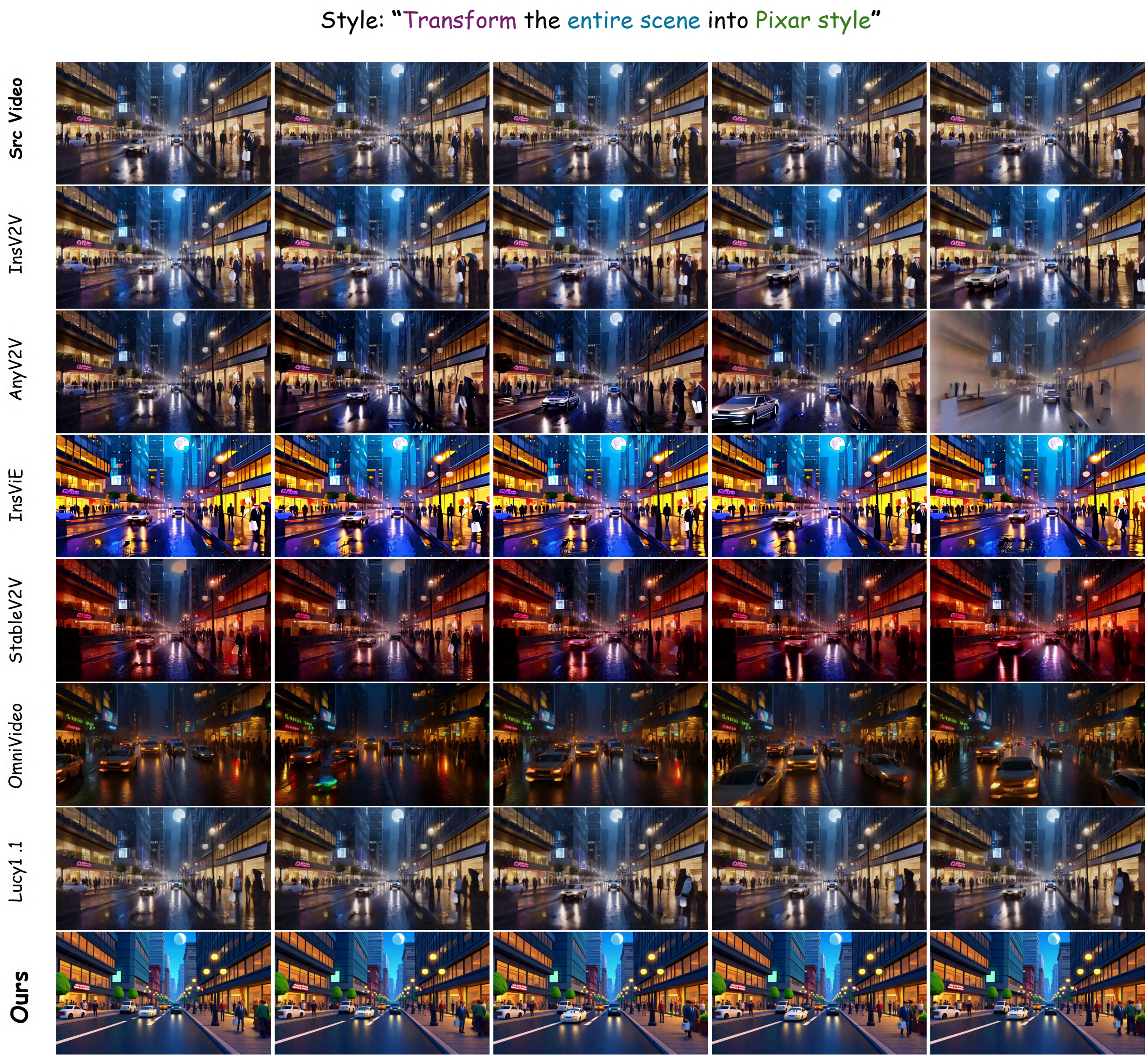

Figure 9. Additional qualitative results comparing CoT-Edit with baselines on Style Transfer task.

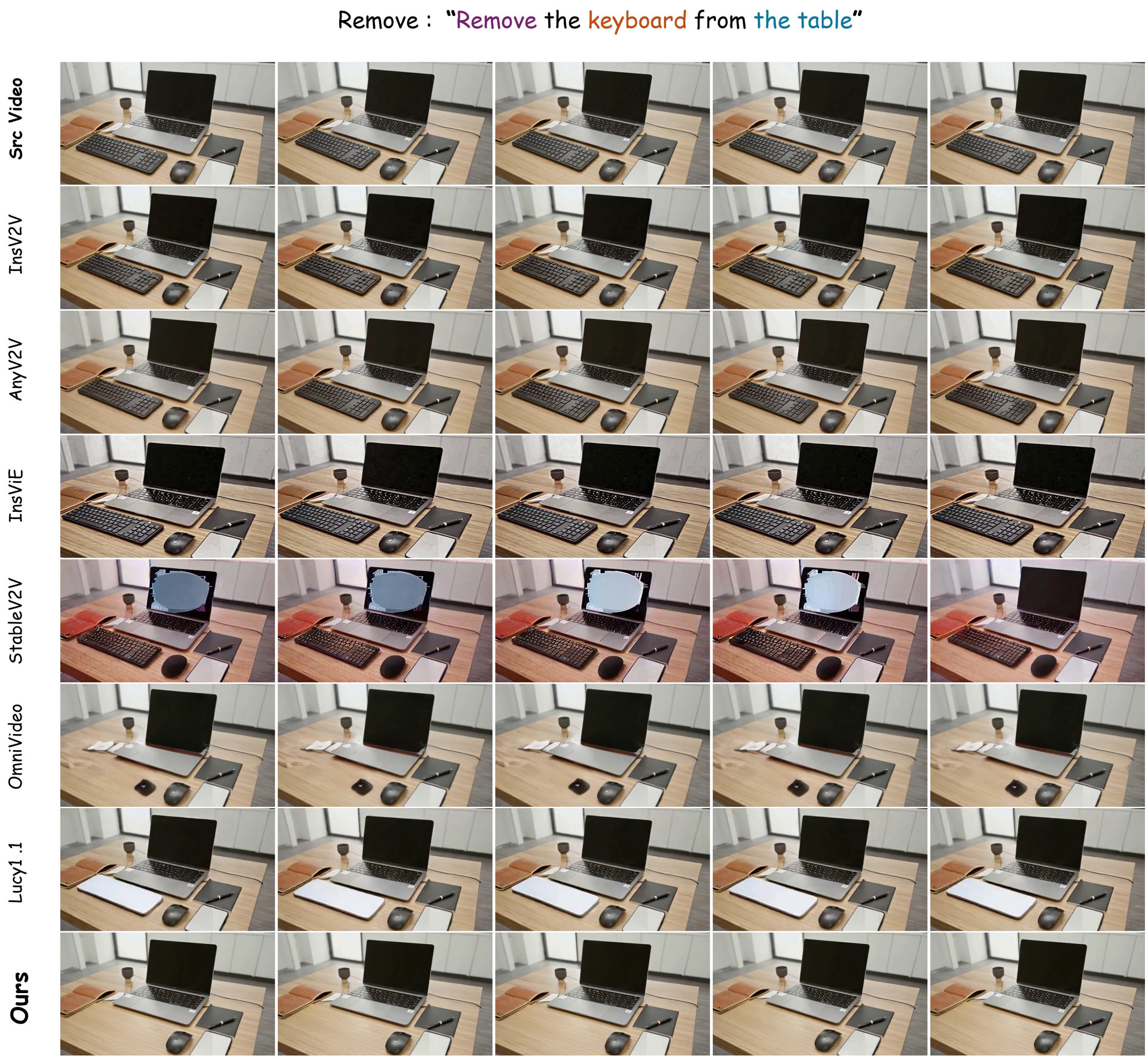

Figure 10. Additional qualitative results comparing CoT-Edit with baselines on Remove task.

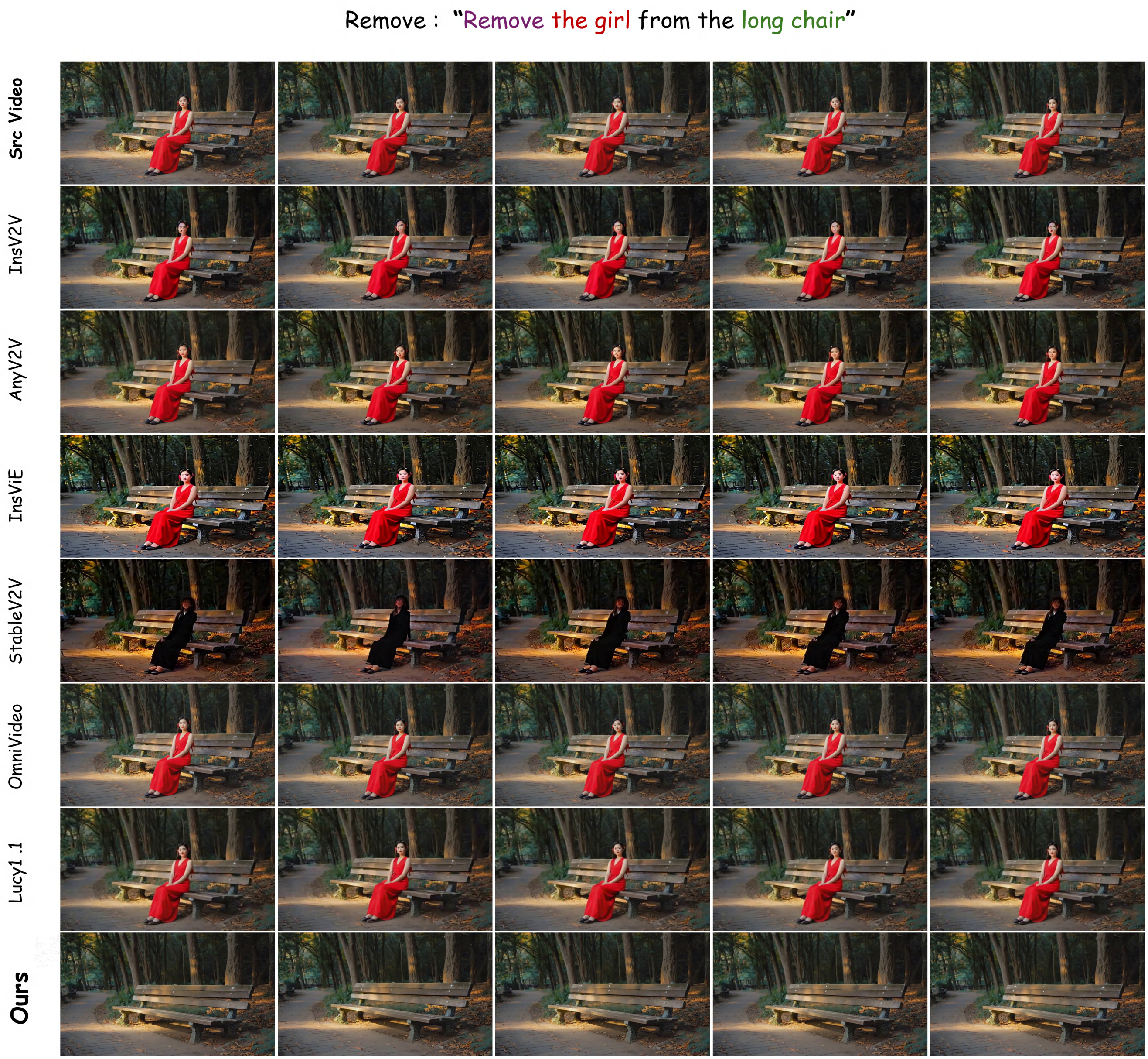

Figure 11. Additional qualitative results comparing CoT-Edit with baselines on Remove task.

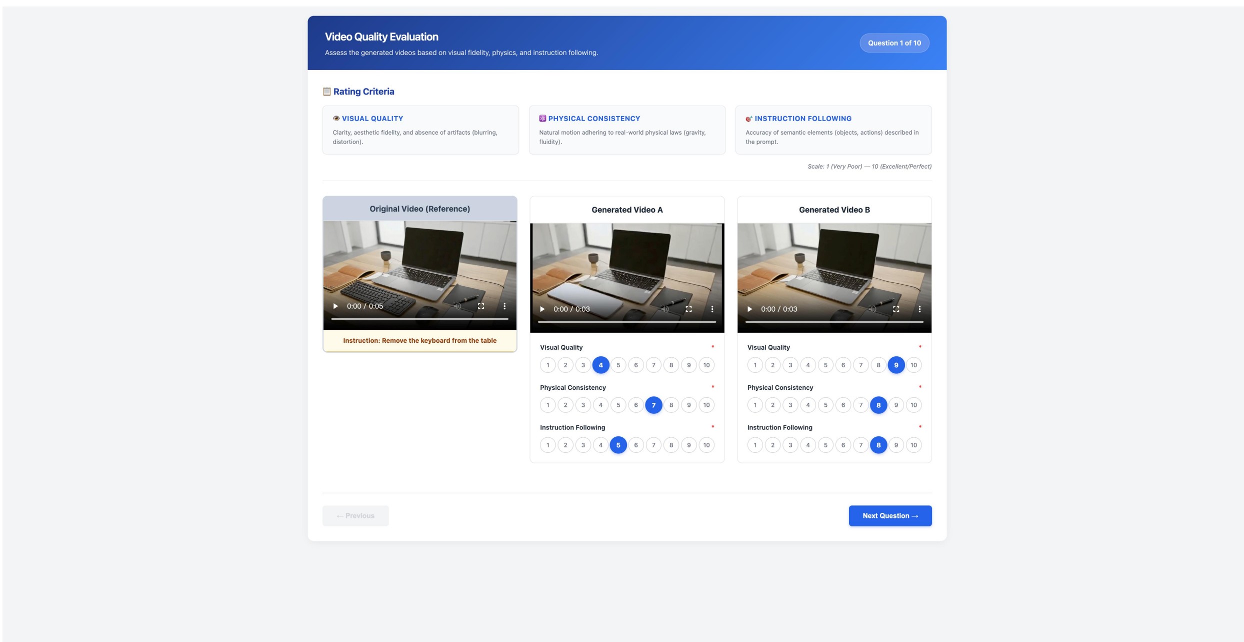

Figure 12. **User Study Interface.** A screenshot of the evaluation platform showing the side-by-side comparison setup. The instruction is displayed below the original video along with the scoring criteria, and rating sliders for Visual Quality, Physical Consistency, and Instruction Following are provided below the videos.