Beyond Perceptual Shortcuts: Causal-Inspired Debiasing Optimization for Generalizable Video Reasoning in Lightweight MLLMs

Supplementary Material

In this document, we provide additional details on the implementation and benchmarks to complement the main paper. Further experiments are also incorporated. And we provided all the code in the supplement, including the JSON files for bias and VideoThinker-R1 training. Specifically, we introduce the details of the training setting in Sec. A. And in Sec. B, we illustrate how to filter the inferential and observational data, including the algorithm and samples, and the details of the setups. Finally, the discussion of the limitations is provided in Sec. C.

A. Detailed Experimental Setup

A.1. Training Setup

Prompt. For training, we use the simple prompting strategy like TW-GRPO [6], the prompt is: *"Output the thinking process in <think></think> and the final answer (letters separated by commas, if multiple) in <answer></answer> tags."*

Reward. VideoThinker-R1 uses two types of rewards:

- **Format Reward.** Similar to other existing MLLM-R1 [4, 5], we introduce format rewards to ensure the model outputs responses in the desired format. For example, we expect the model to enclose its thought process within `<think>...</think>` and the answer within `<answer>...</answer>`. We design a format reward R_{format} for each task and use regular expression matching to determine whether the model adheres to the specified format.
- **Multi-Level Soft Reward.** To address the high reward variance in complex reasoning tasks with multiple correct answers, we choose a soft reward [6] to provide a more granular learning signal. This reward is designed to assign partial credit for incomplete yet correct predictions while strictly penalizing any false positives. Specifically, the reward R_{soft} is calculated as the ratio of correctly predicted items to the total ground truth items ($|P|/|G|$) if and only if the predicted set P is a subset of the ground truth set G. If any prediction is outside the ground truth set, the reward is zero, thus promoting precision. This fine-grained feedback on accuracy leads to more stable gradient estimation and policy optimization.

Bias Model Training. The first step in our VideoThinker framework is to create a specialized Bias Model (π_{bias}) that explicitly learns and embodies the "perceptual shortcut" behavior. To construct its training data, we begin with the counterfactual subset of the CLEVRER [9] dataset. Using the filtering method detailed in Section B.3, we curate a set of 12,191 (out of 18,473) "perceptual" samples, where the correct answer can be directly observed from the video. Crucially, to compel the model to adopt a shortcut, we programmatically simplify these questions into purely observational tasks by removing the counterfactual condition. For instance, the question *"Which event will happen if the cylinder is removed?"* is transformed into the simpler *"Which event will happen?"*. This modification forces the model to ignore the reasoning premise and instead learn a policy that describes only visual events, thereby intentionally instilling the desired perceptual bias.

To train the bias model, we randomly selected 500 samples from this curated dataset and fine-tuned the base model for 500 steps. We employed the GRPO [1] but deliberately removed its KL-divergence constraint to accelerate the model's convergence to the biased policy [38]. Other training parameters, such as a learning rate of 10^{-6}, follow established work. On a single NVIDIA RTX A6000 GPU with 48GB VRAM, this fine-tuning process takes approximately 4.5 hours to complete.

VideoThinker-R1 Training. With the frozen Bias Model (π_{bias}), we proceed to fine-tune our primary reasoning model, VideoThinker-R1. For this stage, we randomly sample 1,000 examples from the original CLEVRER [9] counterfactual training set. The model is trained for 500 steps using CDPO, as defined in Equation 6. In this objective, the crucial hyperparameter β, which controls the strength of the repulsive force against the bias model, is set to 0.001. On two NVIDIA RTX A6000 GPUs with 48GB VRAM, this fine-tuning process takes approximately 6 hours to complete.

A.2. Evaluating Setup

Our model's performance is assessed across six diverse video benchmarks, which we categorize into two groups to ensure a comprehensive evaluation. The first group, focused on general video understanding, includes MVBench [49], TempCompass [50], and VideoMME [14]. These benchmarks primarily test core visual perception and temporal comprehension abilities. The second group is designed to evaluate complex reasoning, featuring CLEVRER [9], Video-Holmes [48], and MMVU [47]. These datasets as-

sess sophisticated spatiotemporal and multimodal reasoning over dynamic video content.

For all evaluations, we follow the experimental setup used in Video-RFT [8], using identical prompts, sampling temperature (0.01), top_p (0.001), and batch size to ensure consistency. For CLEVRER [9], following the work [6], we evaluate exclusively on its most challenging counterfactual subset. To maintain a fair comparison with models that do not support a multiple-answer format, we adopt a single-answer evaluation subset. For other benchmarks, we align with the setup in Video-RFT [8], conducting experiments on excluding subtitles from VideoMME [14] and the multiple-choice split of MMVU [47].

B. Details of Diagnostic Experiment

To better understand the root causes of performance degradation in smaller models fine-tuned under perceptual biases, we constructed a diagnostic experiment that disentangles two qualitatively different reasoning types: **observational** and **inferential**. This diagnostic task was motivated by the observation that perceptual shortcuts, heuristics that exploit surface-level correlations in visual outputs, often suffice for solving a subset of questions, while failing for others that require causal or counterfactual inference. To concretely illustrate this distinction, we curated visual question-answering datasets from the CLEVRER benchmark [9] and manually annotated questions into two categories:

- **Observational questions**, where a model succeeds by detecting and describing what visibly occurred in the video.
- **Inferential questions**, where solving the task requires reasoning about hypothetical interventions.

In the following sections, we present detailed examples from each category to clarify their defining characteristics and implications for model behavior. We also provide a formal description of our problem filtering strategy, which employs a rule-based classification algorithm to automatically distinguish between observational and inferential questions, thereby enabling large-scale analysis of capability conflicts under different fine-tuning regimes.

B.1. Examples of the Observational Problem

Observational questions are characterized by the fact that the correct answer can be derived directly from what is visually present in the video. Figure A1 presents a representative example of an *observational* question from the CLEVRER dataset. In this case, the question asks what event will not happen if the rubber object is removed. Among the candidate options, "the sphere and the blue cylinder collide" corresponds to a visible event in the original video, and this event remains unaffected by the hypothetical removal. The annotations confirm that this collision occurs regardless of the intervention. Solving such questions does not require reasoning about alternative outcomes or hypothetical changes. Instead, perceptual matching between the video content and the answer options is sufficient.

B.2. Examples of the Inferential Problem

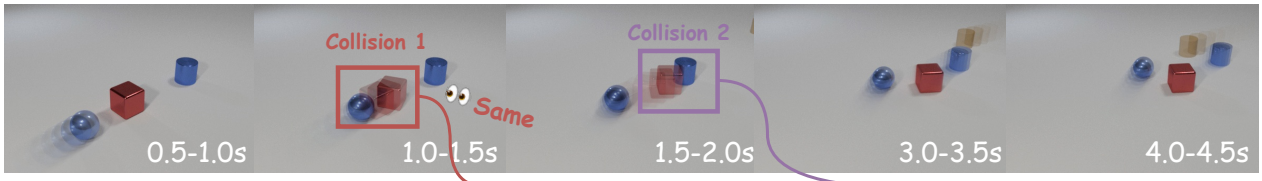
Figure A2 illustrates a representative example of an *inferential* question from the CLEVRER dataset. The question asks what event would occur if a specific object, the green cube, were removed. In this case, answering correctly requires reasoning beyond the directly observable sequence. The correct answer involves predicting a new collision that is not present in the original annotated video, namely the interaction between the yellow cylinder and the metal sphere.

This type of question cannot be resolved through direct observation alone. In the original video, the green cube initiates a chain of interactions, and its removal would alter the subsequent trajectory of the remaining objects. As such, solving the question demands counterfactual reasoning about how the physical system would evolve under a hypothetical intervention. This makes the problem fundamentally different from observational tasks and places greater demands on the model's causal understanding.

B.3. Problem Filtering Strategy Explanation

To determine whether a visual reasoning question is *observational* or *interventional*, we employ a rule-based classification algorithm, detailed in Algorithm 1, to distinguish whether a given visual reasoning question is *observational* or *interventional*. The procedure takes as input a natural language `problem` and a candidate `option`, and proceeds in four main stages. First, the system extracts the core physical event from the option using the `ExtractBaseEvent` function. This typically corresponds to an interaction predicate such as "the sphere and the cube collide." Second, the problem text is examined for linguistic negation (e.g., "not," "never") using the `ContainsNegation` function. If negation is detected, the base event is logically inverted via the `NegateEvent` function (e.g., "do not collide"); otherwise, the base event remains unchanged. Third, the system queries the target event—either the base event or its negation—against structured video annotations using `SearchInAnnotations`. If the event is found in the annotated data, the option is considered grounded in actual observations and the problem is classified as *observational*. Otherwise, the event must be inferred under a hypothetical intervention (e.g., object removal), and the problem is labeled *interventional*.

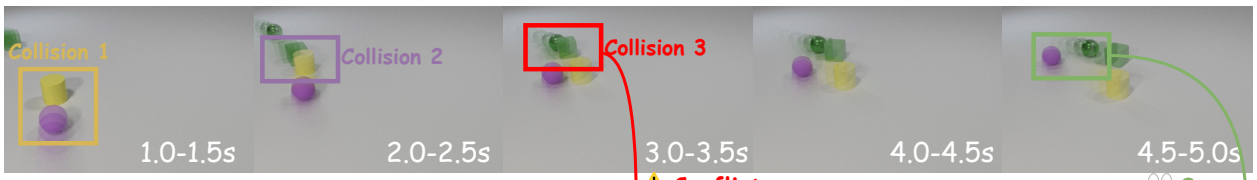
To illustrate, consider the problem: "If the rubber object is removed, what will **not** happen?" with the option "The sphere and the cube collide." Because the question is negative, the event is negated to "The sphere and the cube do not collide," and this negated event is searched for in the annotations. If it is not found, the problem requires reasoning about a counterfactual scenario and is thus classified as



Problem: If the rubber object is removed, what will not happen?
Options: A. The sphere and the cube collide ✗ B. The sphere and the blue cylinder collide ✓ (Same)
Annotations (Collision Event in Video):

Collision_ID	Frame_ID	Object 1	Object 2
1	31	"blue", "metal", "sphere"	"red", "metal", "cube"
2	50	"red", "metal", "cube"	"blue", "metal", "cylinder"

Figure A1. The example of the Observational Problem in CLEVRER.



Problem: Which of the following will happen if the cube is removed?
Options: A. The cylinder collides with the metal sphere ✓ B. The purple object and the metal sphere collide ✗
Annotations (Collision Event in Video):

Collision_ID	Frame_ID	Object 1	Object 2
1	44	"purple", "rubber", "sphere"	"yellow", "rubber", "cylinder"
2	63	"yellow", "rubber", "cylinder"	"green", "rubber", "cube"
3	81	"green", "rubber", "cube"	"green", "metal", "sphere"

Figure A2. The example of the Inferential Problem in CLEVRER.

interventional. In contrast, for the problem "Which of the following will happen if the cube is removed?" and the option "The cylinder collides with the metal sphere," the base event is used directly. If it is present in the annotations, the problem is classified as observational. This pipeline ensures that natural language cues, video-grounded evidence, and causal structure are systematically integrated to support robust problem categorization.

B.4. Diagnostic Experimental Setup

We adopt the same evaluation setup as described in Section A.2. To eliminate the potential influence of unfamiliar answer formats on baseline model performance, we restrict our evaluation to the single-answer questions within the counterfactual subset of CLEVRER. This avoids degradation caused by exposure mismatches with multiple-answer formats. Our implementation follows Video-RFT [8], using identical prompts, a sampling temperature of 0.1, top_p of 0.001, and a batch size of 64. For each video, we sample 32 frames and upscale them to a resolution of $256 \times 28 \times 28$. The full counterfactual subset contains 9,238 questions, among which 3,945 are single-answer. To obtain fine-grained accuracy estimates, especially for inferential generalization, we evaluate model predictions at the *option level* rather than only at the question level. This is necessary because some inferential questions include distractor options that are observational in nature (e.g., option B in Figure A2). We therefore compute accuracy separately over individual options, resulting in a total of 11,524 observational options and 1,224 inferential options. This setup enables a

Algorithm 1 Data Filtering Strategy

Input: `problem`, `option`
Output: “Observational" or “Interference"
{Step 1: Extract the base event}
baseEvent ← ExtractBaseEvent(option)
{Step 2: Determine if the problem contains negation}
if problem contains negation **then**
    eventToQuery ← NegateEvent(baseEvent)
else
    eventToQuery ← baseEvent
end if
{Step 3: Search the event in annotations}
found ← SearchInAnnotations(eventToQuery)
{Step 4: Classify based on search result}
if found is true **then**
    return “Observational"
else
    return “Inferential"
end if

more precise measurement of the model’s reasoning degradation and perceptual bias under fine-tuning.

C. Discussion

C.1. Limitations

Despite its promising results, our work has several limitations that offer avenues for future research. First, the effectiveness of VideoThinker is primarily validated on VQA tasks that align well with our underlying causal assumptions; its generalization to less structured tasks like video captioning remains an open question. Furthermore, our framework employs a practical, gradient-based approximation (CDPO) of a causal intervention, prioritizing computational efficiency over theoretical exactness. Finally, our analysis is centered on a 3B lightweight model where the fine-tuning degradation was most pronounced, and a more comprehensive study is needed to understand how perceptual bias and our intervention scale with larger models.