A. Related Work

A.1. Multimodal Large Reasoning Models

Multimodal reasoning with MLLMs, namely multimodal large reasoning models (MLRMs) have evolved from *modular prompting* to *rule-based RL*. Early approaches decouple perception from inference—visual facts are distilled into textual hints, then consumed by a separate reasoning stage (Zhang et al., 2024c; Zheng et al., 2023; Li et al., 2024b); video counterparts further impose stage-wise templates and external memory/tool use to scaffold multi-step inference (Shi et al., 2024; Fei et al., 2024; Qiu et al., 2025; Yang et al., 2024c). This structured supervision stabilizes training but tends to overfit handcrafted stages, yielding brittle generalization to temporal and causal patterns. More recently, rule-based RL (Guo et al., 2025; Jaech et al., 2024; Team et al., 2025; Qu et al., 2025) has shifted the focus from hand-crafted pipelines to *programmatic, verifiable rewards*, encouraging CoT (chain-of-thought)-style behavior in code (Robeyns & Aitchison, 2025; Pennino et al., 2025), math (Yan et al., 2025; Zhan et al., 2025), and images (Liu et al., 2025b; Zhang et al., 2025d; Tan et al., 2025; Huang et al., 2025b; Yang et al., 2025c), with extensions to videos (Feng et al., 2025; Wang et al., 2025a; Li et al., 2025c; Zhang et al., 2025f). In this Reinforcement Learning with Verifiable Rewards (RLVR) paradigm, the model outputs a canonicalized answer; a rule/regex validator exact-matches it to a gold label to yield a binary reward that deters reward hacking and directly drives subsequent advantage estimation for scalable training. For video tasks, recent variants (Chen et al., 2025b; Wang et al., 2025d; Meng et al., 2025; Zhang et al., 2025b) broaden subtasks coverage and data, refine credit assignment (*e.g.*,, difficulty-aware (Park et al., 2025) and token-level importance-based (Dang et al., 2025)), and scale test-time compute (Wang et al., 2025e) to better steer single-model trajectories. Yet more signals or compute hardly internalize *correct reasoning dependencies* (Yue et al., 2025; Zhao et al., 2025a). Thus, we introduce a teacher model that debiases spurious dependencies in the student's reasoning, turning guidance into robust, causally grounded gains.

A.2. On-Policy and Off-Policy RL for Video QA

Reinforcement learning (RL) for Video QA has evolved into two primary paradigms based on how experiences are utilized during policy updates. *On-policy* methods rely on trajectories generated by the current policy, updating immediately. This approach ensures stable improvements, especially when rewards are verifiable and closely tied to the answer formats, enabling effective exploration. However, it can be data and compute intensive. Several large video models (Feng et al., 2025; Wang et al., 2025a; Zhang et al., 2025a; Wang et al., 2025d) follow this route and have demonstrated consistent performance gains. In contrast, *off-policy* methods, such as DPO (Rafailov et al., 2023) and its variants, leverage historical trajectories or preferences generated by older or broader policies, improving sample efficiency and accelerating performance gains (Zhang et al., 2024b; Huang et al., 2025a; Ahn et al., 2025; Dahal et al., 2025). However, off-policy learning introduces the challenge of distribution shift (Zhan et al., 2025; Yan et al., 2025).

Building on progress in mathematical reasoning, several video off-policy RL frameworks adopt hybrid policies that mix replay buffers with on-policy samples, using past stable traces to aid current learning (Agarwal et al., 2024; Yan et al., 2025; Zhang et al., 2025e; Kulkarni & Fazli, 2025). Yet replay-based schemes depend on careful replay management and mixed-policy regularizer. Without them, exploration shrinks and reward hacking emerges. A parallel line zooms into dynamically acquired, dependency-focused context to boost reasoning, but typically requires curated pretraining and two-stage fine-tuning (He et al., 2025; Zhang et al., 2025c; Wang et al., 2025b), yet they still rely on context retrieved within a small model's capability envelope, limiting the potential to overcome the performance bottleneck. To overcome these limits, we introduce a larger, tool-integrated teacher that detects and challenges spurious or missing dependencies, delivering targeted evidence and prompts that steer exploration and refine the student's internal reasoning.

B. Teacher Model Prompts

This section provides the complete prompts used for the teacher model in our experiments. These prompts were carefully designed through extensive preliminary testing to ensure optimal compatibility with the teacher model's capabilities, minimal tool invocation errors, and consistent formatted output generation. The fixed prompts presented here represent the best-performing versions selected from multiple iterations based on empirical validation.

B.1. Teacher Error Analysis Prompt

The `TEACHER_ERROR_ANALYSIS_PROMPT` is specifically designed for analyzing incorrect responses from the student model while maintaining strict no-leakage constraints. As described in Section 2, the teacher provides guidance regardless of

whether it has access to the ground truth answer, ensuring that the evidence patch never directly reveals the correct solution.

Listing 1. TEACHER_ERROR_ANALYSIS_PROMPT

```
You are an expert teacher model analyzing incorrect responses from a student
model's video reasoning.

{input_information}

## Instructions

The student's answer is incorrect. Your task is to:

1. First, think through your analysis in <think> tags:
   - Identify what the student misunderstood
   - Determine the specific type of error
   - Identify minimal evidence that would correct the error

2. Then provide structured output in <answer> tags.

## Error Categories (choose one):

- `temporal`: Misunderstood temporal sequences or event ordering
- `spatial`: Incorrect interpretation of specific frame(s)
- `misconception`: Misinterpretation of task requirements or question intent

## Output Format

<think>
[Your analysis of the student's error]
</think>

<answer>
{
    "error_classification": "one of: temporal, spatial, misconception",
    "evidence_patch": {
        "content": "Minimal guidance to correct error WITHOUT revealing answer",
        "key_frames": [list of important frame indices],
        "temporal_markers": ["key timestamps or temporal relationships"],
        "spatial_regions": ["important regions in specific frames"]
    }
}
</answer>

## Critical Constraints:
- NEVER directly reveal the correct answer in the evidence patch
- Provide MINIMAL necessary information to guide correction
- Focus on highlighting what was missed, not stating the answer
- The student must still reason independently to reach the solution

## Leakage Prevention for Directly-Observable Queries:
Some queries have answers that are directly observable (e.g., color, count,
object presence/absence, or a single discriminative frame). For these cases,
apply STRICTER constraints:

- Color/Appearance: Do NOT mention the target color or visual attribute.
  Instead: "Re-examine the object's appearance in frame X."
- Object Presence/Absence: Do NOT confirm or deny existence.
  Instead: "Look more carefully at the specified region across frames X-Y."
- Counting: Do NOT state the count.
  Instead: "Recount the items in the specified area, frame by frame."
- Single Discriminative Frame: Do NOT describe the frame content.
  Instead: "Pay closer attention to the transition around frame X."
- Action Identity: Do NOT name the action.
  Instead: "Track the subject's movement trajectory between frames X and Y."

If the answer can be inferred from ANY single field of the evidence patch
alone (key_frames + temporal_markers + spatial_regions + content), the patch
is TOO revealing. Reduce specificity until the student must combine the
patch with their own visual re-examination to arrive at the answer.

Remember: Your evidence must help the student identify their error without
leaking the answer. Guide discovery, don't provide solutions.
```

B.2. Teacher Negative Prompt

A critical challenge in designing the teacher model is ensuring it provides meaningful guidance without revealing the correct answer. Our preliminary experiments revealed that even state-of-the-art language models tend to inadvertently leak answer

information when attempting to help correct errors. To address this, we developed a comprehensive negative prompt that explicitly constrains the teacher's behavior through clear prohibitions and concrete examples.

The TEACHER_NEGATIVE_PROMPT serves as a crucial safeguard against answer leakage by establishing strict boundaries on what information the teacher can provide. Rather than directly stating what occurs in the video or providing specific details that would shortcut the student's reasoning process, the teacher is instructed to guide the student toward discovering errors independently. This approach ensures that the learning signal remains meaningful - the student must still perform visual analysis and reasoning to arrive at the correct answer, with the teacher merely highlighting where attention should be focused.

Listing 2. TEACHER_NEGATIVE_PROMPT

```
## Critical Constraints - DO NOT VIOLATE

You MUST NOT:
- State or imply the correct answer directly
- Describe what happens in the video beyond what the student already observed
- Provide specific counts, colors, or object identities unless already mentioned by student
- Complete the student's reasoning for them
- Use phrases like "the answer is", "you should select", or "the correct choice"

## Examples of Violations to Avoid

**Counting Dynamics**
* **BAD:** "There are exactly 3 people in frame 15." (Directly reveals the count)
* **GOOD:** "Recount the subjects within the frame range [13, 17]."

**Dynamics**
* **BAD:** "The person moves from left to right." (Directly describes the action/direction)
* **GOOD:** "Track and analyze the subject's movement direction across the provided clip."

**Temporal**
* **BAD:** "The event happens before the person sits down." (Directly reveals temporal order/causality)
* **GOOD:** "Examine the temporal relationship and causal sequence between the two identified events."

**Spatial**
* **BAD:** "The answer is C / the yellow sphere." (Directly points to the answer/object)
* **GOOD:** "Identify paths intersected by the static cylinder to determine potential collisions."

**Attribute**
* **BAD:** "The object being picked up is a red metal cube." (Directly reveals visual properties)
* **GOOD:** "Observe the visual features (color/material) of the object involved in the interaction."

**Logic**
* **BAD:** "The ball stops because it hits the wall." (Directly reveals the causal explanation)
* **GOOD:** "Analyze the interaction between the moving object and its surrounding environment."

## Filtering Rules for Directly-Observable Answers

When the question asks about a directly observable attribute:
1. NEVER name the attribute value (color, count, object identity, direction).
2. NEVER describe frame content that would make the answer self-evident.
3. ALWAYS redirect to a region or temporal window, requiring the student
   to RE-OBSERVE the visual evidence independently.
4. If unsure whether guidance leaks the answer, apply the "blind test":
   Could someone who has NOT seen the video determine the answer from
   your patch alone? If yes, the patch is too revealing.

Remember: Your role is to help students discover their errors through guided
exploration, not to provide answers. Every piece of evidence should require
the student to observe, analyze, and conclude independently.
```

This negative prompt component is integrated with the main teacher prompt to ensure consistent behavior across all interactions. The examples provided illustrate the distinction between revealing information (BAD) and guiding discovery (GOOD), helping the teacher model understand the appropriate level of assistance.

Handling Directly-Observable Queries. A particular challenge arises when the answer is directly observable—*e.g.,*, a color, an object's presence or absence, or a count visible in a single frame. In such cases, even minimal content guidance risks inadvertently revealing the answer. Our prompt design addresses this through three layers: ❶ The error analysis prompt includes *query-type-specific* constraints (see Listing 1) that explicitly prohibit naming colors, counts, or attribute values; ❷ The negative prompt enforces a "blind test" rule: if the patch alone (without seeing the video) would allow someone to infer the answer, it must be made less specific; ❸ The structured output format forces the teacher to decompose guidance

into key_frames, temporal_markers, and spatial_regions—none of which individually encode the answer. Together, these constraints ensure that the student must re-examine the video to derive the answer, even for queries with directly observable solutions.

B.3. Answer Leakage Prevention Analysis

To validate the effectiveness of our prompt design, we conducted extensive empirical evaluation with manual verification. As shown in Table 5 (main text), our final prompt design combining structured output constraints with explicit negative prompts achieves zero answer leakage across 200 manually verified teacher-student interactions. The negative prompt component proved particularly effective, reducing leakage from 39.5% to 5.5% when used alone, and eliminating it entirely when combined with format constraints. This validation confirms that our prompt design successfully maintains the integrity of the learning process while providing meaningful corrective feedback.

B.4. Teacher Tool Use Prompt

The TEACHER_TOOL_USE_PROMPT defines the tools available to the teacher model for detailed video analysis.

Listing 3. TEACHER_TOOL_USE_PROMPT

```
## Available Tools

You have access to the following tools to examine the video more closely:

### 1. get_frame(frame_index: int)
Retrieves a specific frame from the video for detailed analysis.

### 2. zoom_region(frame_index: int, x1: float, y1: float, x2: float, y2: float)
Zooms into a specific region of a frame. Coordinates are normalized (0-1).

### 3. get_temporal_segment(start_frame: int, end_frame: int, stride: int = 1)
Retrieves a sequence of frames to analyze temporal patterns.

Use these tools when you need to:
- Verify specific visual details mentioned in the student's response
- Identify missed temporal dependencies
- Locate spatial regions containing key evidence
- Validate or refute the student's observations

Call tools in the following format:
TOOL_CALL: tool_name(parameters)

Example:
TOOL_CALL: get_frame(45)
TOOL_CALL: zoom_region(45, 0.3, 0.2, 0.7, 0.6)
```

B.5. Error Classification Summary

Table 6 summarizes the error classification system used by the teacher model, along with typical leakage risks and the corresponding mitigation strategies applied in the evidence patch.

Table 6. Error Classification Categories with Leakage Mitigation Strategies

Category	Description	Leakage Risk	Mitigation
temporal	Wrong temporal order or event sequence	Naming the correct order directly	Redirect to temporal window; use "before/after" without specifying the events
spatial	Wrong interpretation of frame content	Describing the correct visual content	Point to region coordinates; avoid naming objects/colors/attributes
misconception	Misinterpretation of question intent	Restating the question with implicit answer	Clarify the task focus (e.g., "which action" vs. "which object") without revealing the target

C. Teacher Model Analysis

Table 7 presents a comprehensive evaluation of how different teacher models influence student performance across video reasoning and general understanding benchmarks. This analysis reveals several key insights about teacher-student dynamics

Table 7. Performance comparison of student models using different teacher models. All experiments use Video-R1-SFT-7B as the student model.

Model	Video Reasoning Benchmark				Video General Benchmark			
	MMVU↑	VSI-Bench↑	VideoMMMU↑	Video-Holmes↑	LongVideoBench↑	LVBench↑	MVBench↑	TempCompass↑
Video-R1-SFT	61.3	31.8	47.4	34.6	47.6	30.7	59.4	69.2
FFR-Qwen3-32B	67.9	38.5	50.5	47.8	52.6	34.2	68.3	72.2
FFR-Qwen3-235B	68.2	38.1	**56.5**	51.6	54.2	33.9	68.8	75.2
FFR-GPT-4o	**69.0**	38.5	55.4	49.2	53.9	36.3	**70.0**	74.9
FFR-GLM4.5-V	68.5	**38.9**	54.6	**52.3**	**55.3**	**38.1**	68.8	**75.6**

Table 8. Training Hyperparameters

Hyperparameter	Value	Hyperparameter	Value
Rollouts (G)	8	Weight decay	0.01
Max prompt len.	16,384	Batch size (per GPU)	1
Max completion len.	1,024	Grad. accum. steps	1
Temperature	1.0	Max grad. norm	5.0
Top-p	0.95	Training epochs	1
KL coeff. (β)	0.04	Precision	BF16
Patch tax (κ)	0.3	Attention	FlashAttn-2
Learning rate	5e-6	Max image res.	401,408 px
LR scheduler	Cosine	Video frames	16

in the FFR framework.

Teacher Model Selection Matters. While all teacher models improve upon the baseline Video-R1-SFT student (61.3 MMVU without teacher), the choice of teacher significantly impacts final performance. GLM-4.5V emerges as the most effective teacher, achieving the best results on 5 out of 8 benchmarks. This superiority likely stems from its strong instruction-following capabilities and robust tool-use integration, enabling more precise error diagnosis and targeted evidence generation.

Task-Specific Teacher Advantages. Different teachers excel at different types of reasoning tasks. GPT-4o shows particular strength in MMVU (69.0), suggesting superior multi-modal understanding, while GLM-4.5V dominates in Video-Holmes (52.3), indicating better temporal reasoning capabilities. This task-specific variation suggests that teacher selection could be optimized based on the target domain.

Scaling Effects. The comparison between Qwen3-32B and Qwen3-235B teachers reveals that simply scaling model size doesn't guarantee better teaching effectiveness. While the 235B model achieves higher VideoMMMU scores (56.5 vs 50.5), the smaller 32B variant performs comparably or better on several other benchmarks. This suggests that teaching ability depends more on model architecture and training methodology than raw parameter count.

D. Experiment Setting Details

This section provides detailed descriptions of the benchmarks, baselines, and implementation details referenced in Section 3.

D.1. Benchmarks

In this work, we evaluate our method on several video reasoning and general video understanding benchmarks to assess its effectiveness and generalization.

Video Reasoning Benchmarks. We use MMVU (Zhao et al., 2025b), VSI-Bench (Yang et al., 2025a), VideoMMMU (Hu et al., 2025b), and Video-Holmes (Cheng et al., 2025). These benchmarks cover a variety of tasks, testing models on their ability to answer complex questions by leveraging spatiotemporal context. MMVU focuses on expert-level multi-discipline understanding, VSI-Bench on visual–spatial reasoning, VideoMMMU on model's ability to acquire and apply domain–knowledge, and Video-Holmes on causal and narrative reasoning from multi-segment video clues.

General Video Understanding Benchmarks. We incorporate LongVideo-Bench (Wu et al., 2024), LVBench (Wang et al., 2025c), MVBench (Li et al., 2024a), and TempCompass (Liu et al., 2024). These benchmarks evaluate models' generalization across diverse video tasks, from long-form video understanding to temporal consistency and motion-based

reasoning. LongVideo-Bench tests long-duration video comprehension, LVBench assesses large-scale visual understanding, MVBench challenges multi-modal video understanding and TempCompass examines temporal alignment.

D.2. Baselines

We evaluate our method against a range of baselines, categorized as on-policy methods, mixed-policy methods, and tool-use reasoners, as shown in Figure 1. Specifically, on-policy methods (Figure 1 (*a*)) include Qwen2.5-VL-7B (Bai et al., 2025), Video-R1-7B (Feng et al., 2025), Video-RFT-7B (Wang et al., 2025a), and Video-ChatR1-7B (Li et al., 2025c). Mixed-policy methods (Figure 1 (*b*)) consider AVATAR-7B (Kulkarni & Fazli, 2025). Tool-use reasoners (Figure 1 (*c*)) include Pixel-Reasoner (Su et al., 2025a) and Video-Thinker (Wang et al., 2025b). Specifically, Pixel-Reasoner adopts a two-stage training (SFT-RFT) to learn tool-use abilities, while Video-Thinker is trained for multi-tasking, specifically captioning and temporal grounding.

D.3. Implementation Details

We use Video-R1-7B-SFT and VideoRFT-7B-SFT as our base models and conduct training on 8 NVIDIA A100 GPUs with 80GB memory each. For training efficiency, video inputs are limited to 16 frames, with each frame processed at a resolution of $128\times28\times28$. We train with the FFR framework for 1 epoch, using a learning rate of 5e-6 and generating 8 samples per rollout. The training dataset contains 4,000 samples (corresponding to 1K steps of RL training), sourced from Video-R1, Video-Thinker, and Video-Holmes. Considering computational costs, we select GLM-4.5V as the teacher model. The entire experimental framework is implemented based on R1-V (Chen et al., 2025a). During inference, we maintain the same configuration as training, using 16-frame inputs and a maximum pixel resolution of $128\times28\times28$ for model evaluation.

E. Algorithm Implementation Details

This section provides implementation details of the FFR framework integrated with GRPO training.

E.1. Training Pipeline

Our implementation builds upon the R1-V framework (Chen et al., 2025a) with modifications to support multi-modal inputs and teacher-guided evidence patches. The training pipeline consists of three main components:

1. Rollout Generation. For each training sample, the student model generates G rollouts (typically $G = 8$) using temperature sampling. Each rollout τ_i consists of the model's reasoning trajectory and final answer. During generation, we maintain visual context (images or videos) alongside text prompts to ensure proper multi-modal reasoning.

2. Teacher Intervention. When a rollout produces an incorrect answer (verified through rule-based matching), the frozen teacher model analyzes the error and provides a minimal evidence patch c_i. The teacher uses predefined tools to examine specific frames, temporal segments, or spatial regions, generating targeted guidance without revealing the answer. This intervention is triggered only for incorrect rollouts, creating an adaptive curriculum.

3. Second-Round Generation with Evidence. For incorrect rollouts, the student generates a new response τ_i' conditioned on both the original input and the teacher's evidence patch. This second-round generation allows the student to correct its reasoning while maintaining on-policy exploration.

E.2. More Implementation Details

Our implementation integrates the FFR framework into the standard GRPO training loop with minimal modifications. The key implementation choices include:

Teacher API Architecture. The teacher model runs as a separate service to ensure stable performance during distributed training. This decoupling allows the teacher to use different hardware resources and avoids memory conflicts with the student model training.

Rollout Selection Strategy. Among the $G = 8$ rollouts per batch, we select the rollout with highest reward as the chosen trajectory $\widehat{\tau}_i$ for policy update. When multiple rollouts achieve the same reward, we prefer those that succeeded without teacher assistance to encourage autonomous reasoning.

Synchronization in Distributed Training. With DeepSpeed ZeRO-3, model parameters are sharded across GPUs. To avoid deadlocks, all ranks must participate in generation even if only some ranks need teacher intervention. We use collective communication primitives to coordinate which rollouts require second-round generation across all ranks.

E.3. Multi-Modal Processing

For video inputs, we sample frames at regular intervals and process them through the vision encoder. Both teacher and student models process 16 frames per video to balance computational cost and temporal coverage. The maximum image resolution is constrained to 401,408 pixels. Visual features are projected and concatenated with text embeddings before the language model backbone.

E.4. Hyperparameters

Table 8 summarizes the key hyperparameters used in our experiments.

F. Additional Experimental Results

F.1. SFT-Teacher vs FFR Comparison

Table 9 presents the full comparison between direct teacher distillation (SFT-Teacher) and FFR across all eight benchmarks (a summary on video reasoning benchmarks is provided in Table 4 in the main text). We use teacher-generated reasoning traces from Qwen3-32B and Qwen3-235B for SFT baselines. The results show that while SFT-Teacher improves over the baseline, FFR significantly outperforms direct distillation, particularly on complex reasoning tasks (+8.4 on VideoMMMU, +5.2 on Video-Holmes compared to SFT-235B). FFR with a 32B teacher already matches SFT with a 235B teacher on overall average (both 54.0), and FFR with GLM-4.5V achieves the best overall performance (56.5). This demonstrates that FFR's targeted intervention mechanism provides more effective learning signals than wholesale knowledge transfer from the teacher.

Table 9. Full comparison of SFT-Teacher vs. FFR under matched teacher models across all benchmarks. FFR consistently outperforms SFT at every scale; FFR with a 32B teacher already matches SFT with a 235B teacher (both Avg: 54.0).

Method	Video Reasoning				Video General				Avg.
	MMVU	VSI-Bench	VideoMMMU	Video-Holmes	LongVideoBench	LVBench	MVBench	TempCompass	
Video-R1-SFT (Baseline)	61.3	31.8	47.4	34.6	47.6	30.7	59.4	69.2	47.8
SFT (Qwen3-32B)	63.9	39.1	42.0	43.3	50.4	36.7	57.3	73.0	50.7
SFT (Qwen3-235B)	67.4	**41.9**	46.2	47.1	54.5	**39.9**	60.2	75.1	54.0
FFR (Qwen3-32B)	67.9	38.5	50.5	47.8	52.6	34.2	68.3	72.2	54.0
FFR (Qwen3-235B)	68.2	38.1	**56.5**	51.6	54.2	33.9	**68.8**	75.2	55.8
FFR (GLM-4.5V, Ours)	**68.5**	38.9	54.6	**52.3**	**55.3**	38.1	**68.8**	**75.6**	**56.5**

F.2. Patch Tax κ Sensitivity Analysis

Table 10 presents comprehensive results for different κ values across all benchmarks.

Table 10. Sensitivity analysis of patch tax κ across all benchmarks.

κ	Video Reasoning				Video General				Avg.
	MMVU	VSI-Bench	VideoMMMU	Video-Holmes	LongVideoBench	LVBench	MVBench	TempCompass	
0.1	**72.7**	38.7	49.0	44.3	53.6	**42.0**	61.2	73.6	54.4
0.3	68.5	38.9	**54.6**	**52.3**	55.3	38.1	**68.8**	**75.6**	**56.5**
0.5	69.3	**42.8**	47.3	44.2	52.4	37.6	60.6	73.8	53.5
0.7	68.5	39.3	47.0	45.2	54.8	37.2	61.8	73.5	53.4
1.0	70.6	40.4	47.9	43.0	**57.8**	35.8	60.8	75.4	54.0

The results demonstrate that FFR is robust to different κ values, with all configurations outperforming the baseline. $\kappa = 0.3$ achieves the best overall performance (56.5%), particularly on complex reasoning benchmarks. Too small a penalty (κ=0.1) leads to teacher over-reliance, while larger values ($\kappa \geq 0.5$) excessively discount teacher contributions.

F.3. Detailed Case Study

Table 11 provides a detailed breakdown of the case study shown in Figure 3, illustrating how FFR's evidence patches enable reasoning correction without answer leakage. This case is drawn from a real training sample in the STAR dataset.

Table 11. Detailed case study of FFR intervention on a temporal reasoning task from STAR dataset.

Component	**Content**
Source	STAR Dataset, Video: 5QW1X.mp4
Question	Which object was put down by the person?
Options	A. The clothes B. The book C. The shoe D. The dish
Ground Truth	B (The book)
Student's Original Response	✗ Answer: A (The clothes) *Reasoning:* "I see clothes in the scene..." → Focused on visible clothes, missed the actual put-down action
Error Classification	`temporal_error`, `spatial_error`
Evidence Patch	
Key Frames	[13, 14, 15]
Temporal Markers	"when the person's hand releases the object", "moment of placement on surface"
Guidance	"Track hand movements and which object is placed down."
Student's Corrected Response	✓ Answer: B (The book) *Reasoning:* "At frames 8–10, the person's hand releases the book onto the surface. Clothes remain in place throughout."
Why This Patch Works	
✓ Minimal & Sufficient	Points to specific frames [13, 14, 15]; provides temporal cues (hand release); guides attention without revealing answer
✗ Does NOT	State "the book was put down"; reveal the correct answer is B; describe exact frame contents

This case exemplifies the core design principle of FFR: the evidence patch redirects the student's attention to the relevant visual evidence without short-circuiting the reasoning process. The student must still observe the specified frames, track the hand movements, and reason about which object transitions from being held to being placed down.