ObjShotBench: Benchmarking VLMs for Object Level Scene Consistency Verification in Multi Shot Videos

Anonymous ACL submission

Abstract

Video generation is moving from isolated single shot clips toward multi shot videos with related views, scene cuts, and narrative progression. This shift introduces a cross shot consistency problem: a video can look plausible within each shot while allowing objects to disappear or appear across related shots without a plausible cause. We study *object level scene consistency verification*, which asks whether a multi shot video contains an invalid object state transition and which object is responsible. We introduce OBJSHOTBENCH, a controlled diagnostic benchmark that isolates disappearance, appearance, and normal cross shot cases through counterfactual object removal. The benchmark supports evaluation of feature based metrics and VLM based semantic verifiers across increasingly difficult verification settings. Experiments show that whole shot feature similarity is poorly aligned with local object state failures. VLMs provide stronger structured verification, but open ended responsible object grounding remains a key bottleneck.

1 Introduction

Recent video generation systems increasingly produce multi shot videos with related views, scene cuts, and events (Kara et al., 2025; Guo et al., 2025; Wang et al., 2026; Meng et al., 2025; Wang et al., 2025a). This shift raises a finer evaluation problem than per shot realism. A video can look plausible within each shot while violating scene continuity across shots: an object may disappear in a later view, or appear without a plausible physical or narrative cause. Such failures require evaluating whether an object's state transition is valid, not merely whether two shots look globally similar.

Existing consistency metrics commonly rely on proxy signals such as face embeddings, DINOv2 features, subject crops, segmented background features, or video text similarity. These signals capture resemblance, identity preservation, and text alignment, but their sensitivity is misaligned with object level failures. A normal viewpoint, scale, or framing change can substantially reduce feature similarity between two consistent shots, while removing a small object may leave global representations nearly unchanged. A metric that flags the former and misses the latter is not measuring object level scene consistency. This asymmetry, examined directly in Section 4, motivates semantic verification beyond representation similarity.

We focus on object disappearance and appearance across related shots. We define *object level scene consistency verification* as determining whether an object's presence state changes in a semantically invalid way, and identifying the responsible object when an inconsistency exists. This treats object continuity as a structured semantic judgment. VLMs are a natural candidate because, unlike feature encoders, they can condition on task instructions, compare object identity across shots, and reason about plausible causes such as occlusion, out of frame motion, or narrative progression. Yet they may still confuse disappearance with occlusion, fail under camera changes, misattribute the responsible object, or hallucinate inconsistency.

We introduce OBJSHOTBENCH, a controlled diagnostic benchmark for this problem. Rather than reproduce the full distribution of generated video artifacts, it isolates one necessary capability: judging whether an object's cross shot presence transition is semantically valid. We construct counterfactual examples from real object annotated videos instead of mining generated videos, where object state failures are often entangled with artifacts, deformation, and motion defects. Object removal creates controlled disappearance and appearance cases, while original or semantically valid shot pairs serve as hard negatives that preserve realistic cross shot variation.

OBJSHOTBENCH supports binary judgment, multiple choice answering, and open ended ver-

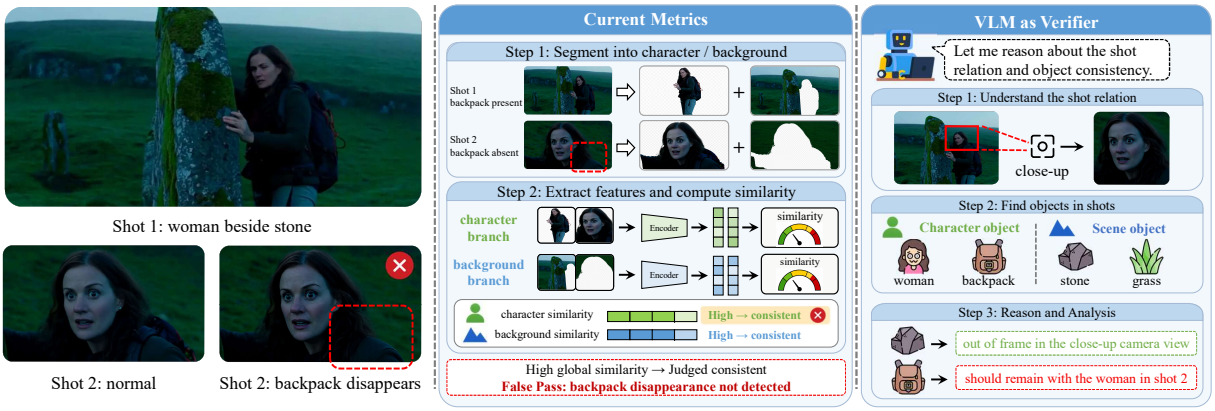

Figure 1: Comparison between feature based consistency metrics and VLM based semantic verification. Feature metrics measure representation level similarity and can miss object level appearance or disappearance, whereas a VLM verifier can inspect cross shot visual evidence, judge whether an object state transition is semantically valid, and identify the responsible object.

ification, separating detection, transition type classification, and responsible object grounding. We use these protocols to evaluate feature metrics and prompted VLM verifiers under in domain and out of domain settings, and include a lightweight GRPO adaptation to test whether verifier behavior can be improved.

Our contributions are:

- We formulate object level scene consistency verification as a diagnostic task for multi shot video generation, focusing on invalid object disappearance and appearance across related shots.
- We construct OBJSHOTBENCH, a controlled diagnostic benchmark of disappearance, appearance, and normal cross shot cases that isolates semantic object state reasoning through counterfactual object removal.
- We evaluate feature metrics and VLM verifiers across binary, multiple choice, open ended, and out of domain settings, and include a lightweight GRPO adaptation to probe the strengths and limits of semantic verification.

2 Related Work

Multi shot video generation. Video generation has progressed from short isolated clips toward longer and more structured outputs with shot changes, camera transitions, and narrative progression (Ho et al., 2022; Singer et al., 2022; Gao et al., 2025; Kara et al., 2025; Guo et al., 2025; Wang et al., 2026; Meng et al., 2025; Wang et al., 2025a). This shift makes cross shot coherence an increasingly important evaluation target. Existing work often emphasizes controllable generation, prompt to shot planning, or visual quality across generated segments. Our work studies a narrower but consequential failure mode in the resulting videos: whether an object's presence remains semantically valid across related shots.

Feature based consistency metrics. Consistency is commonly approximated with representation similarity. Face embeddings, DINO style features, subject crops, segmented background features, and CLIP or ViCLIP like video text scores can measure identity preservation, scene resemblance, or text alignment (Deng et al., 2019; Oquab et al., 2023; Radford et al., 2021; Wang et al., 2022). These metrics are useful proxies for broad visual resemblance, but they do not directly test object state validity. In multi shot videos, a normal viewpoint change can reduce feature similarity, while a small object disappearance may leave global representations nearly unchanged. This motivates our diagnostic comparison between feature metrics, which measure similarity, and semantic verification, which judges whether a presence state transition is valid.

Semantic verification with VLMs. Vision language models have been used as judges for visual quality, image text alignment, video understanding, and multimodal reasoning (Chen et al., 2024; Li et al., 2023; Maaz et al., 2023). Their language conditioned visual reasoning makes them appealing for scene consistency verification, but their judgments can depend on prompts, frame

sampling, context length, and object salience, and they may hallucinate or miss temporally distributed evidence. We therefore evaluate VLMs with controlled object level labels rather than treating them as automatic judges. Our construction uses object masks and descriptions from video object segmentation resources, including MeViS v2 and Cut-VOS, together with ROSE style object removal to create controlled disappearance and appearance cases (Ding et al., 2023; Hu et al., 2026; Miao et al., 2025). These resources are used as interventions for diagnostic evaluation; the target problem is semantic consistency verification, not segmentation or editing.

3 Method

We study object level scene consistency verification in multi shot videos. The verifier determines whether an object's presence state changes across related shots without a plausible physical or narrative cause. This section defines the task, describes the OBJSHOTBENCH construction, and presents a lightweight GRPO based verifier adaptation.

3.1 Problem Definition

Given a multi shot video $V = \{S_1, S_2, \ldots, S_T\}$, where each S_t is a shot or segment, we consider shot pairs that are temporally, spatially, or narratively related. Such shots may vary in viewpoint, scale, background, framing, or camera position, but object state changes should be explainable by motion, occlusion, framing, or narrative progression.

For a target object o, we define its observed presence state in shot S_t as

$$s_t(o) \in \{\text{present}, \text{absent}\}. \tag{1}$$

Here, *present* means that the object is visually observable, while *absent* means that it is not visible. Absence alone does not imply inconsistency: occlusion, out of frame motion, close up crops, or narrative transitions may all be valid. An inconsistency occurs only when the inferred transition is semantically invalid across related shots.

We use three labels. *Disappearance* means that an object is visible in an earlier related shot but becomes absent later without a plausible cause. *Appearance* means that an absent object appears later without explanation. *Normal* covers semantically valid cases, including natural movement, occlusion, viewpoint changes, out of frame motion, and reasonable object exits. Thus, the verifier must judge state validity, not just visibility, and identify the inconsistency type and responsible object when needed.

3.2 Benchmark Construction

OBJSHOTBENCH is a controlled diagnostic benchmark for object level scene consistency verification. As shown in Figure 2, we construct it from object annotated videos rather than directly mining generated videos, whose inconsistencies are often entangled with artifacts, deformation, and motion errors. Real videos with object masks let us isolate whether a model can judge object state transitions under realistic cross shot variation.

We use Cut-VOS and SeCVOS as source datasets (Hu et al., 2026; Zhang et al., 2026), which provide object masks, descriptions, and editable target regions. We select multi shot videos where the target object remains identifiable across related shots despite viewpoint, scale, background, or framing changes, and exclude targets that are too small, heavily occluded, ambiguous, or hard to associate before editing.

We apply object removal to the target region and compose original and removed shots into three example types. *Disappearance* keeps the earlier shot original and removes the target from a later shot, creating an unexplained transition from present to absent. *Appearance* removes the target from an earlier shot and keeps the later shot original, creating an unexplained transition from absent to present. *Normal* uses semantically consistent shot pairs without introduced object level inconsistency.

After composition, we perform quality control to ensure that samples test object state reasoning rather than artifact detection. We check four conditions: the target object is visible in the source shot, object removal succeeds, the edited region does not contain obvious visual artifacts, and the original shots form a valid cross shot relation. Samples failing any check are removed.

We generate question answer instances in three formats: binary judgment, multiple choice, and open ended question answering. Binary questions test detection of a specified abnormal transition. Multiple choice questions test discriminative grounding with correct, normal, and distractor options. OpenQA asks whether any object level inconsistency exists and, if so, which type and object. These formats separate detection, type classification, and responsible object grounding.

Each instance contains the video, target object

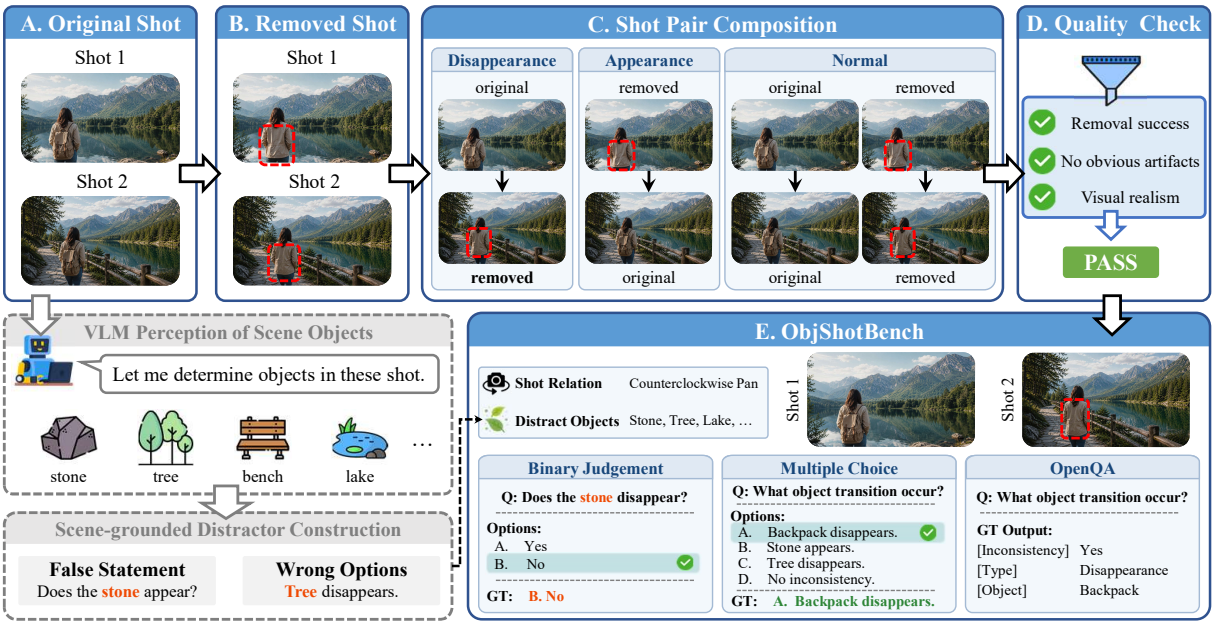

Figure 2: Overview of the ObjShotBench construction pipeline. Starting from original shots, we remove target objects to compose disappearance, appearance, and normal shot pairs, filter samples through visual quality checks, and generate benchmark questions for binary judgment, multiple choice answering, and OpenQA evaluation. Distractor objects and candidate statements are included to test whether VLMs can ground the target object rather than rely on superficial cues.

description, shot pair composition, question format, ground truth label, and responsible object when applicable. Table 1 summarizes the benchmark scale. We use Cut-VOS for training and in domain evaluation, and hold out SeCVOS as an out of domain split to test transfer across source distributions.

3.3 Verifier Optimization

We further optimize a verifier with Group Relative Policy Optimization (GRPO) to test whether the protocols can improve object state verification. The model generates responses in the `<thinking><answer>` format: `<thinking>` analyzes shot relations, object permanence, occlusion, out of frame cases, and motion, while `<answer>` gives the final judgment, type, and responsible object.

We initialize from Qwen3-VL-8B-Instruct without an SFT cold start stage and optimize on the Cut-VOS training split. Cut-VOS is used for in domain validation, while SeCVOS remains unseen for OOD validation. This provides a lightweight probe of whether reward optimization improves detection, transition classification, and responsible object grounding.

Reward Design. We use a composite reward with a format reward r_{format} and an accuracy reward r_{acc}. The format reward checks the required `<thinking><answer>` structure. The accuracy reward evaluates whether `<answer>` matches the ground truth: normal examples require no inconsistency, while abnormal examples require the correct type and responsible object. Object matching uses object descriptions and normalized aliases to avoid rewarding generic names. The final reward is

$$R = r_{\text{format}} + r_{\text{acc}}.$$

This reward design encourages structured outputs and discrimination among disappearance, appearance, and normal cases. We report the adapted verifier as ObjShot-GRPO-8B in Section 5.

4 Feature Metric Diagnostic Study

Before evaluating VLM based verifiers, we first test whether feature similarity is aligned with the target failure mode at all. This diagnostic study bridges the motivation and the verifier evaluation: if whole shot similarity already captured object state validity, a semantic verifier would be less necessary. The key issue is that normal cross shot variation, such as changes in viewpoint, scale, framing, and background, can strongly affect global visual features, while object disappearance or appearance often changes only a local region.

Source	Videos	QA	Pos.	Neg.	Normal	Shots/video	Shot len.	Video len.
Cut-VOS [11]	47	1,071	423	423	225	2–4	4.3s	15.5s
SeCVOS [28]	27	948	260	260	428	2–4	2.1s	25.9s

Table 1: Dataset statistics for OBJSHOTBENCH by source dataset. Pos. and Neg. denote positive and negative QA examples. Normal denotes QA pairs labeled with no object level inconsistency. Shot len. and Video len. denote average shot and video duration.

We study this issue from two perspectives on OBJSHOTBENCH. First, we compare feature sensitivity to normal cross shot variation and object removal. Second, we convert feature similarity into a binary Normal/Inconsistent decision and compare it with VLM based OpenQA results.

4.1 Sensitivity Asymmetry Protocol

We first compare two similarity scores. S_{view} denotes the average similarity between two semantically consistent shots. It measures how much normal cross shot variation changes the feature representation. S_{obj} denotes the average similarity before and after removing the target object from the same shot. It measures how much object removal changes the feature representation.

For each visual encoder, we extract shot level features and compute cosine similarity. For image level encoders, including DINOv2, SigLIP, and EVA-CLIP, we average frame level features over sampled frames. For ViCLIP, we use video level shot features.

Encoder	$S_{\text{view}} \uparrow$	$S_{\text{obj}} \uparrow$	Δ
DINOv2 [18]	0.601	0.833	0.231
SigLIP [27]	**0.872**	**0.914**	0.042
EVA-CLIP [21]	0.783	0.876	0.093
ViCLIP [25]	0.741	0.907	0.166

Table 2: Sensitivity asymmetry of feature based metrics on OBJSHOTBENCH. S_{view} is the average similarity of normal cross shot pairs, S_{obj} is the average similarity before and after same shot object removal, and Δ denotes the gap between them.

The results in Table 2 show that all encoders have higher S_{obj} than S_{view}. In other words, the same shot remains highly similar after the target object is removed, while normal cross shot pairs receive lower similarity because of camera and scene changes. This indicates that whole shot feature similarity is more sensitive to cross shot visual variation than to local object state changes.

This result shows why feature similarity is not a direct measure of object level scene consistency. It can capture coarse visual resemblance, but it does not reliably reflect whether an object undergoes an invalid disappearance or appearance.

4.2 Pairwise Accuracy Protocol

We next test whether feature similarity can be used as a binary object level inconsistency detector. For each encoder, we set a similarity threshold. If the similarity of a shot pair is above the threshold, the pair is predicted as Normal. Otherwise, it is predicted as Inconsistent.

The threshold is chosen for a conservative operating point comparison rather than as a claim about the best possible feature metric detector. We observe that Qwen3-VL-8B-Instruct has a high Normal accuracy in OpenQA, around 87%. Therefore, we calibrate each feature metric to a similar operating point by choosing the threshold that makes Normal accuracy close to 85%. We then evaluate how well the metric detects Appearance and Disappearance examples under this high Normal accuracy regime.

For Qwen3-VL-8B-Instruct, we report two OpenQA results. The strict accuracy counts a prediction as correct only when both the inconsistency type and the responsible object are correct. The relaxed binary accuracy only checks whether the model correctly distinguishes Normal from Inconsistent. The relaxed setting is more comparable to feature based binary detection.

Method	τ	Appear $\uparrow$	Disappear $\uparrow$
Feature Encoders			
DINOv2 [18]	0.333	0.207	0.207
SigLIP [27]	0.779	0.320	**0.311**
EVA-CLIP [21]	0.640	0.248	0.279
ViCLIP [25]	0.565	0.234	0.239
Qwen3-VL-8B-Instruct [3]			
Strict OpenQA	–	0.152	0.165
Relaxed binary	–	**0.330**	0.272

Table 3: Threshold based feature metric accuracy on object level inconsistency detection, calibrated to a conservative Normal accuracy operating point.

As shown in Table 3, feature metrics remain weak detectors of object level inconsistency at

Model	Binary				MCQ			
	Disapp.	Appear.	Normal	Acc	Disapp.	Appear.	Normal	Acc
Proprietary Models								
Claude Sonnet 4.6 [2]	**96.3**	92.6	85.2	**91.4**	**92.6**	88.9	70.4	84.0
Gemini 3 Flash [8]	92.6	**96.3**	81.5	90.1	88.9	**92.6**	**74.1**	**85.2**
GPT-5.4 [17]	92.6	92.6	**88.9**	**91.4**	**92.6**	88.9	66.7	82.7
Qwen3.6-Plus [1]	88.9	85.2	77.8	84.0	85.2	85.2	63.0	77.8
Open source VLMs								
Qwen3-VL-4B-Instruct [3]	81.5	70.4	63.0	71.6	59.3	77.8	33.3	56.8
Qwen3-VL-8B-Instruct [3]	85.2	74.1	63.0	74.1	63.0	**81.5**	**37.0**	60.5
Qwen3-VL-30B-A3B-Instruct [3]	88.9	**77.8**	**66.7**	**77.8**	74.1	**81.5**	**37.0**	64.2
InternVL3.5-4B [24]	77.8	66.7	55.6	66.7	51.9	70.4	29.6	50.6
InternVL3.5-8B [24]	81.5	70.4	59.3	70.4	59.3	74.1	29.6	54.3
InternVL3.5-14B [24]	85.2	74.1	63.0	74.1	66.7	77.8	33.3	59.3
GRPO Trained Models								
OBJSHOT-GRPO-8B (Ours) [3]	**92.6**	**77.8**	63.0	**77.8**	**88.9**	**81.5**	33.3	**67.9**

Table 4: Binary classification and multiple choice question answering results on OBJSHOTBENCH. We report accuracy for disappearance, appearance, and normal examples, together with overall accuracy.

this conservative operating point. Most encoders achieve only around 20%–30% accuracy on Appearance and Disappearance. SigLIP performs best among feature metrics, but its accuracy is still low.

Qwen3-VL-8B-Instruct has lower strict OpenQA accuracy because it must identify both the inconsistency type and the responsible object. Under the relaxed binary setting, its Appearance accuracy increases to 0.3304 and its Disappearance accuracy increases to 0.2718. This setting is closer to feature based binary detection and shows that VLM verification provides some semantic benefit, but still misses many inconsistent cases when the answer is unconstrained.

Overall, feature similarity is not well aligned with object level scene consistency. It is sensitive to normal cross shot variation and weakly sensitive to local object state changes. This motivates the following experiments on VLM based semantic verification, where the protocols explicitly test detection, transition classification, and responsible object grounding.

5 Experiments

We evaluate whether VLMs can serve as semantic verifiers for object level scene consistency. Section 4 shows that feature similarity is poorly aligned with object disappearance and appearance, so the central question here is whether instruction following VLMs can make the required cross shot judgment and identify the responsible object. The three protocols form a difficulty ladder: binary classification tests detection for a specified target, MCQ adds discriminative grounding among candidate object state transitions, and OpenQA removes candidates entirely, requiring free form type and object grounding. We also evaluate out of domain generalization to test whether the learned or prompted verifier transfers beyond the benchmark construction distribution.

5.1 Experimental Setup

Evaluation data. OBJSHOTBENCH contains both Cut-VOS and SeCVOS examples, but we assign them different experimental roles. Cut-VOS provides the training split for GRPO and the in domain validation split used for the main benchmark results. SeCVOS is held out as an out of domain validation split, so it is part of OBJSHOTBENCH's source coverage but not used for verifier optimization. The experiments cover three task formats: binary classification, multiple choice question answering (MCQ), and open ended question answering (OpenQA). Binary classification asks the model to determine whether a specified object undergoes abnormal disappearance or appearance. MCQ asks the model to select the correct object state transition from candidate options. OpenQA asks the model to directly analyze whether object level inconsistency exists in the video, and to report the inconsistency type and the corresponding object.

Models. We compare proprietary VLMs, open source VLMs, and our GRPO adapted verifier. The proprietary group includes Claude Sonnet 4.6, Gemini 3 Flash, GPT-5.4, and Qwen3.6-Plus. The open source group includes Qwen3-VL and InternVL3.5 family models with different parameter

Model	Type Acc	Object Acc	Acc	Disapp. P	Disapp. R	Appear. P	Appear. R
Proprietary Models							
Claude Sonnet 4.6 [2]	64.2	**61.7**	51.9	**96.3**	18.5	88.9	**22.2**
Gemini 3 Flash [8]	**65.4**	60.5	**53.1**	92.6	**22.2**	**96.3**	18.5
GPT-5.4 [17]	63.0	**61.7**	51.9	**96.3**	18.5	92.6	**22.2**
Qwen3.6-Plus [1]	60.5	58.0	48.1	88.9	14.8	88.9	14.8
Open source VLMs							
Qwen3-VL-4B-Instruct [3]	59.3	56.8	44.4	66.7	22.2	63.0	22.2
Qwen3-VL-8B-Instruct [3]	60.5	58.0	45.7	66.7	25.9	63.0	22.2
Qwen3-VL-30B-A3B-Instruct [3]	64.2	60.5	48.1	**70.4**	25.9	66.7	25.9
InternVL3.5-4B [24]	59.3	56.8	44.4	63.0	22.2	63.0	22.2
InternVL3.5-8B [24]	58.0	55.6	43.2	63.0	22.2	59.3	22.2
InternVL3.5-14B [24]	63.0	59.3	46.9	66.7	25.9	66.7	25.9
GRPO Trained Models							
OBJSHOT-GRPO-8B (Ours) [3]	**65.4**	**61.7**	**49.4**	**70.4**	**29.6**	**70.4**	**29.6**

Table 5: OpenQA results on OBJSHOTBENCH. Type Accuracy measures whether the model predicts the correct inconsistency type. Object Accuracy measures whether the model identifies the responsible object. P and R denote precision and recall.

Model	Binary				MCQ				OpenQA			
	Disapp.	Appear.	Normal	Acc	Disapp.	Appear.	Normal	Acc	Disapp.	Appear.	Normal	Acc
Proprietary Models												
Claude Sonnet 4.6 [2]	**84.2**	85.9	89.0	**86.7**	**76.3**	68.2	77.1	74.2	**36.8**	36.5	83.1	56.3
Gemini 3 Flash [8]	80.3	**88.2**	88.1	86.0	73.7	**70.6**	**78.8**	**74.9**	34.2	**40.0**	**83.9**	**57.0**
GPT-5.4 [17]	82.9	84.7	**89.8**	86.4	**76.3**	67.1	76.3	73.5	35.5	35.3	83.1	55.6
Qwen3.6-Plus [1]	72.4	77.6	79.7	77.1	65.8	58.8	67.8	64.5	26.3	27.1	76.3	47.7
Open source VLMs												
Qwen3-VL-4B-Instruct [3]	55.3	65.9	68.6	64.2	39.5	41.2	55.1	46.6	21.1	16.5	50.8	32.3
Qwen3-VL-8B-Instruct [3]	**59.2**	70.6	70.3	67.4	43.4	**43.5**	**59.3**	**50.2**	7.9	15.3	**88.1**	44.1
Qwen3-VL-30B-A3B-Instruct [3]	**59.2**	72.9	75.4	70.3	46.1	42.4	58.5	**50.2**	18.4	14.1	78.8	42.7
InternVL3.5-4B [24]	53.9	64.7	66.1	62.4	36.8	36.5	51.7	43.0	**39.5**	10.6	56.8	38.0
InternVL3.5-8B [24]	55.3	67.1	66.9	63.8	39.5	38.8	53.4	45.2	19.7	14.1	59.3	34.8
InternVL3.5-14B [24]	56.6	69.4	68.6	65.6	42.1	40.0	55.9	47.3	25.0	12.9	64.4	38.0
GRPO Trained Models												
OBJSHOT-GRPO-8B (Ours) [3]	**59.2**	**75.3**	**79.7**	**72.8**	**50.0**	35.3	**59.3**	49.5	22.4	**22.4**	84.7	**48.7**

Table 6: Out of domain generalization on cross shot object level consistency verification. All models are evaluated with the same prompting protocols as the in domain benchmark.

scales. OBJSHOT-GRPO-8B is initialized from Qwen3-VL-8B-Instruct and optimized with the reward design in Section 3.3.

Evaluation metrics. For binary classification and MCQ, we report accuracy on disappearance, appearance, and normal examples, together with overall accuracy. For OpenQA, we report Type Accuracy, Object Accuracy, overall accuracy, and precision/recall for disappearance and appearance. Type Accuracy measures whether the model predicts the correct inconsistency type. Object Accuracy measures whether the model identifies the corresponding inconsistent object. Overall accuracy requires the final structured prediction to be completely correct. This metric design separates three failure modes: missing abnormal transitions, over predicting inconsistency on normal shot pairs, and naming the wrong responsible object after detecting the correct type.

5.2 VLM Verification Results

5.2.1 Binary Classification and Multiple Choice

Table 4 evaluates two constrained protocols: binary detection for a specified object and MCQ selection among candidate object transitions. Proprietary models are strong but do not form a single ranking. Claude Sonnet 4.6 and GPT-5.4 tie for the best binary accuracy at 91.4%, while Gemini 3 Flash obtains the best MCQ accuracy at 85.2%.

Open source VLMs are clearly weaker, especially on normal examples. The best open source normal accuracy drops from 66.7% in binary classification to 37.0% in MCQ, indicating that candidate transitions make normal case rejection much harder.

OBJSHOT-GRPO-8B improves the constrained

results relative to its base model. It reaches 92.6% binary disappearance accuracy and 77.8% overall binary accuracy, matching the best open source binary result, and obtains the best open source MCQ accuracy at 67.9%. The gains are selective: binary normal accuracy remains 63.0%, below Qwen3-VL-30B-A3B-Instruct at 66.7%, and MCQ normal accuracy remains 33.3%. Thus, GRPO improves sensitivity to object state changes, but does not fully solve normal case calibration.

5.2.2 OpenQA

Table 5 removes candidate answers and requires free form type and object grounding. This setting exposes a different bottleneck. Proprietary models make precise positive predictions, but recall is low: Claude Sonnet 4.6 and GPT-5.4 reach 96.3% disappearance precision, and Gemini 3 Flash reaches 96.3% appearance precision, yet positive recall remains no higher than 22.2%.

Open source VLMs show the complementary weakness. Their recall is comparable, but precision is much lower: the best open source disappearance precision is 70.4%, and the best appearance precision is 66.7%. Several models obtain Type Accuracy near the proprietary range, but overall accuracy remains lower because the responsible object must also be grounded correctly.

GRPO improves OpenQA only modestly. OBJSHOT-GRPO-8B reaches 49.4% overall accuracy, above Qwen3-VL-8B-Instruct at 45.7%, and raises both positive recalls to 29.6%. However, its precision remains far below proprietary models. OpenQA therefore confirms the main distinction: structured semantic verification is more reliable than open ended object grounding.

5.3 Out of Domain Generalization

To test whether models generalize beyond the benchmark construction distribution, we evaluate them on an out of domain set of cross shot examples. These examples follow the same object level consistency verification format but differ in source domain, object categories, visual appearance, or shot composition. We use the same prompting protocols and evaluation rules as the in domain setting.

Table 6 reports out of domain performance under binary classification, MCQ, and OpenQA. For each protocol, we report accuracy on disappearance, appearance, and normal examples, together with overall accuracy. The goal is to measure whether a model has learned a general semantic verification ability or is overfitted to the visual distribution and object removal patterns of the benchmark.

The OOD results preserve the main in domain pattern while lowering absolute performance. Proprietary models remain strongest overall, but their advantages differ by protocol. Claude Sonnet 4.6 leads binary disappearance at 84.2% and OpenQA disappearance at 36.8%, Gemini 3 Flash leads binary appearance at 88.2%, and GPT-5.4 has the strongest binary normal accuracy at 89.8%. This suggests that disappearance detection, appearance detection, and normal calibration transfer differently.

Among open source systems, OBJSHOT-GRPO-8B gives the strongest OOD binary result, reaching 72.8% overall accuracy. MCQ transfer is more mixed: it leads open source disappearance and normal accuracy, but appearance accuracy drops to 35.3%, leaving overall MCQ accuracy slightly below Qwen3-VL-8B-Instruct and Qwen3-VL-30B-A3B-Instruct.

OOD OpenQA is the weakest setting for all model groups. Proprietary models keep normal accuracy above 83%, but disappearance and appearance accuracies remain below 40.0%. OBJSHOT-GRPO-8B improves over its base model on both positive classes, reaching 22.4%, but normal accuracy decreases from 88.1% to 84.7%. This tradeoff reinforces that the benchmark tests calibrated object state reasoning, not only detection of edited objects.

6 Conclusion

We introduce OBJSHOTBENCH, a controlled diagnostic benchmark for object-level scene consistency in multi-shot videos. Built from object-annotated videos, it constructs disappearance, appearance, and normal cross-shot cases to isolate local consistency failures missed by broad visual similarity metrics.

Our results show that object-level consistency should be evaluated as structured semantic verification rather than a single similarity score. Constrained protocols test whether models can detect invalid object state transitions, while OpenQA further requires grounding the responsible object without candidates. This makes OBJSHOTBENCH a diagnostic tool for current VLM verifiers and a testbed for future methods combining calibrated consistency judgment with robust object grounding.

Limitations

This work focuses on object level scene consistency and does not cover all forms of scene consistency, such as lighting, color, material, layout, character identity, or complex narrative causality. The benchmark uses controlled object removal to synthesize disappearance and appearance. This provides a clean diagnostic setting for testing object state reasoning, but it does not reproduce the full distribution of generated video errors, which may be more diverse and less controllable. The benchmark also inherits the domains, object categories, and visual distributions of its source datasets. ROSE based removal may introduce artifacts despite quality control, so we apply filtering and artifact checks during construction. Verifier performance may also depend on prompt design, frame sampling, video resolution, context length, and model version. Although GRPO improves several binary and multiple choice results, it does not solve open ended object grounding, and its gains should be interpreted as verifier specific rather than as evidence that the benchmark is saturated.

AI Assistance Statement

AI tools were used only for writing assistance, including language polishing, grammar correction, and improving clarity. All research ideas, benchmark construction, experimental design, analyses, and conclusions were developed and verified by the authors.

Ethical Considerations

The benchmark is built from existing video datasets, so its use should follow the licenses and redistribution rules of the source datasets. Video object removal can be misused for misleading editing or manipulation. In this paper, it is used to construct diagnostic evaluation data for consistency verification and safety oriented analysis. If source videos contain people, privacy and consent constraints from the original datasets should be respected. The benchmark should not be presented as a tool for deceptive editing, but as a controlled resource for detecting and understanding object level inconsistencies.

References

Alibaba Cloud. 2026. Alibaba Cloud Model Studio: Models. https://www.alibabacloud.com/help/en/model-studio/text-generation-model. Accessed: May 25, 2026.

Anthropic. 2026. Introducing Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6. Accessed: May 25, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Liu, Yinuo Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. MLLM-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. 2023. MeViS: A large-scale benchmark for video segmentation with motion expressions. *arXiv preprint arXiv:2308.08544*.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, Xunsong Li, Yifu Li, Shanchuan Lin, Zhijie Lin, Jiawei Liu, Shu Liu, Xiaonan Nie, Zhiwu Qing, Yuxi Ren, and 4 others. 2025. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*.

Google Cloud. 2025. Gemini 3 Flash. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-flash. Accessed: May 25, 2026.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. 2025. Long context tuning for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17281–17291.

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

Hengrui Hu, Kaining Ying, and Henghui Ding. 2026. Segment Anything Across Shots: A method and Benchmark. In *AAAI*.

Ozgur Kara, Krishna Kumar Singh, Feng Liu, Duygu Ceylan, James M. Rehg, and Tobias Hinz. 2025. Shotadapter: Text-to-multi-shot video generation with diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28405–28415.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-ChatGPT: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Yihao Meng, Hao Ouyang, Yue Yu, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Hanlin Wang, Yixuan Li, Cheng Chen, Yanhong Zeng, and 1 others. 2025. Holocine: Holistic generation of cinematic multi-shot long video narratives. *arXiv preprint arXiv:2510.20822*.

Chenxuan Miao, Yutong Feng, Jianshu Zeng, Zixiang Gao, Hantang Liu, Yunfeng Yan, Donglian Qi, Xi Chen, Bin Wang, and Hengshuang Zhao. 2025. ROSE: Remove objects with side effects in videos. *arXiv preprint arXiv:2508.18633*.

OpenAI. 2026. GPT-5.4 model. https://developers.openai.com/api/docs/models/gpt-5.4. Accessed: May 25, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2023. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. EVA-CLIP: Improved training techniques for CLIP at scale. *arXiv preprint arXiv:2303.15389*.

Jiahao Wang, Hualian Sheng, Sijia Cai, Weizhan Zhang, Caixia Yan, Yachuang Feng, Bing Deng, and Jieping Ye. 2026. Echoshot: Multi-shot portrait video generation. *Advances in Neural Information Processing Systems*, 38:22058–22090.

Qinghe Wang, Xiaoyu Shi, Baolu Li, Weikang Bian, Quande Liu, Huchuan Lu, Xintao Wang, Pengfei Wan, Kun Gai, and Xu Jia. 2025a. Multishotmaster: A controllable multi-shot video generation framework. *arXiv preprint arXiv:2512.03041*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. 2023. InternVid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. 2022. InternVideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Zhixiong Zhang, Shuangrui Ding, Xiaoyi Dong, Songxin He, Jianfan Lin, Junsong Tang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2026. Advancing complex video object segmentation via progressive concept construction. In *The Fourteenth International Conference on Learning Representations*.

A Dataset Construction Details

Table 7 reports the split level composition of ObjShotBench. Cut-VOS is used for verifier optimization and in domain validation, while SeCVOS is reserved for out of domain evaluation. The normal class corresponds to semantically valid shot pairs; appearance and disappearance correspond to counterfactual object state inconsistencies. Object category counts exclude normal examples whose inconsistent object is *None*.

Table 8 summarizes the object category distribution after aggregating question answer samples. The distribution covers people, animals, food or organic objects, plants and natural structures, transparent or reflective objects, tools and instruments, furniture and indoor rigid objects, transportation or machines, and architectural or background structures.

Source	Train	Val/Test	Total	Normal	Appear	Disappear	Obj. cats.
Cut-VOS	990	81	1,071	225	423	423	37
SeCVOS	669	279	948	428	260	260	20

Table 7: Split and label statistics for OBJSHOTBENCH. SeCVOS test examples are used only for out of domain evaluation. Obj. cats. denotes the number of object categories excluding *None*.

Object category	Cut-VOS	SeCVOS	Total
Human	140	98	238
Animal	46	44	90
Food / organic object	188	22	210
Plant / fine natural structure	80	0	80
Transparent / reflective / container / kitchenware	112	134	246
Tool / instrument / slender object	102	164	266
Furniture / indoor rigid object	34	30	64
Transportation / machine / robot	68	28	96
Architecture / ground / background structure	76	0	76
Total, excluding *None*	846	520	1,366

Table 8: Object category distribution by source dataset. Counts are computed over non normal QA samples and therefore exclude the *None* object category.

B Proprietary Model Versions

Table 9 reports the proprietary model identifiers used in our experiments. We queried the latest public API version available to us at evaluation time and record the model identifier, pinned snapshot or release identifier when provided by the vendor, supported visual input modality, and documentation access date for reproducibility.

Documentation sources are the OpenAI GPT-5.4 model page (https://developers.openai.com/api/docs/models/gpt-5.4), Anthropic Claude model overview and model ID documentation (https://platform.claude.com/docs/en/about-claude/models/overview, https://platform.claude.com/docs/en/about-claude/models/model-ids-and-versions), Google Gemini API model documentation (https://ai.google.dev/gemini-api/docs/models), and Alibaba Cloud Model Studio vision model documentation (https://help.aliyun.com/zh/model-studio/vision-model).

C GRPO Training Details

Table 10 summarizes the data and visual input configuration used for verifier optimization. The GRPO verifier is trained on the Cut-VOS training split and selected on the Cut-VOS validation split. SeCVOS is not used for optimization and is reserved for out of domain evaluation. During training, we sample two frames from each shot because each shot is short and usually has limited within shot visual change.

Table 11 reports the optimization hyperparameters. We initialize from Qwen/Qwen3-VL-8B-Instruct and train with LoRA adapters rather than full parameter fine tuning.

Table 12 lists the rollout and decoding settings. Training rollouts use stochastic decoding, while validation uses deterministic decoding.

The reward is task format dependent. For binary and multiple choice training, we use a format reward of 0.1 and a correct answer reward of 1.0. For OpenQA training, we use a format reward of 0.1, a type correct reward of 0.5, and an object correct reward of 0.5. The format reward checks whether the response follows the required structured output format. The correctness rewards compare the final answer against the ground truth inconsistency label and, when applicable, the responsible object.

D Feature Based Metric Details

Implementation details for DINOv2 similarity, ViCLIP similarity, object crop similarity, and background similarity follow the diagnostic baselines in Section 4.

Display name	Provider	API model ID	Pinned version / release	Notes
GPT-5.4	OpenAI	`gpt-5.4`	`gpt-5.4-2026-03-05`	Supports text and image input through the Responses and Chat Completions APIs.
Claude Sonnet 4.6	Anthropic	`claude-sonnet-4-6`	`claude-sonnet-4-6`	Anthropic documents 4.6 generation dateless IDs as pinned snapshots rather than evergreen aliases.
Gemini 3 Flash	Google	`gemini-3-flash-preview`	`gemini-3-flash-preview`	Public preview model supporting text, code, image, audio, video, and PDF inputs.
Qwen3.6-Plus	Alibaba Cloud	`qwen3.6-plus`	`qwen3.6-plus-2026-04-02`	Multimodal Model Studio model supporting text, image, and video inputs.

Table 9: Proprietary model identifiers used in the experiments. Documentation was accessed on May 25, 2026.

Item	Value
Training split	Cut-VOS train
Training samples	990
Validation split	Cut-VOS val
Validation samples	81
Training visual input	2 sampled frames per shot
`min_pixels`	200704
`max_pixels`	345600

Table 10: Data and visual input configuration for GRPO verifier training.

Item	Value
Base model	`Qwen/Qwen3-VL-8B-Instruct`
LoRA rank	64
LoRA alpha	128
KL coefficient	1.0×10^{-2}
Optimizer	AdamW
Learning rate	2.0×10^{-5}
Weight decay	1.0×10^{-2}
LR warmup ratio	0.05

Table 11: Optimization hyperparameters for GRPO verifier training.

Item	Value
Rollout group size n	8
Rollout batch size	16
Training temperature	0.7
Training top-p	0.8
Training top-k	20
Repetition penalty	1.05
Presence penalty	1.5
Validation temperature	0.0
Validation top-p	1.0
Validation top-k	-1

Table 12: Rollout and decoding settings for GRPO verifier training and validation.